SEVENTH FRAMEWORK PROGRAMME
Research Infrastructures

INFRA-2011-2.3.5 – Second Implementation Phase of the European High Performance Computing (HPC) service PRACE

PRACE-2IP

PRACE Second Implementation Project

Grant Agreement Number: RI-283493

D4.1
Training and Education Survey

Final

Version: 1.00
Author(s): Simon Wong, NUI Galway; Danica Stojiljkovic, IPB; Stelios Erotokritou, CaSToRC; George Tsouloupas, CaSToRC; Pekka Manninen, CSC; David Horak, VSB-TUO; Georgi Prangov, NCSA
Date: 23.12.2011

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: RI-283493**	
	Project Title: PRACE Second Implementation Project	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: D4.1	
	Deliverable Nature: Report	
	Deliverable Level: PU	**Contractual Date of Delivery:** 31 / 12 / 2011
		Actual Date of Delivery: 31 / 12 / 2011
	EC Project Officer: Bernhard Fabianek	

* - The dissemination level are indicated as follows: **PU** – Public, **PP** – Restricted to other participants (including the Commission Services), **RE** – Restricted to a group specified by the consortium (including the Commission Services). **CO** – Confidential, only for members of the consortium (including the Commission Services).

Document Control Sheet

Document	**Title: Training and Education Survey**	
	ID: D4.1	
	Version: 1.00	**Status:** ***Final***
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2007	
	File(s): D4.1.docx	
Authorship	**Written by:**	Simon Wong, NUI Galway Danica Stojiljkovic, IPB Stelios Erotokritou, CaSToRC George Tsouloupas, CaSToRC Pekka Manninen, CSC David Horak, VSB-TUO Georgi Prangov, NCSA
	Contributors:	Ondrej Jakl, VSB-TUO
	Reviewed by:	Timothy Robinson, ETHZ
	Approved by:	MB/TB

Document Status Sheet

Version	**Date**	**Status**	**Comments**
0.1	01/12/2011	Draft	
0.2	04/12/2011	Contributions added	
1.0	31/12/2011	Final version	

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure, HPC Training, User Training Survey, HPC Trainer Survey

Disclaimer

This deliverable has been prepared by Work Package 4 of the Project in accordance with the Consortium Agreement and the Grant Agreement n° RI-283493. It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the Project and to the extent foreseen in such agreements. Please note that even though all participants to the Project are members of PRACE AISBL, this deliverable has not been approved by the Council of PRACE AISBL and therefore does not emanate from it nor should it be considered to reflect PRACE AISBL's individual opinion.

Table of Contents

List of Figures

List of Tables

References and Applicable Documents

[1] PRACE RI Official Website: http://www.prace-ri.eu
[2] T. Stitt and T. Robinson (2008) PRACE-PP D3.3.1: "A survey on training and education needs for petascale computing".
[3] P. Manninen and T. Robinson (2010) PRACE-1IP D3.2.1: "Training Plan".
[4] T. Robinson, J.D. Poznanovic and N.D. Stringfellow (2011) PRACE-1IP D3.1.5: "HPC Training Portal".
[5] P. Manninen (2011) PRACE-1IP D3.2.2: "First Training Report".
[6] Top500 Website: http://www.top500.org/
[7] Python Snakes Its Way Into HPC: http://www.hpcwire.com/hpcwire/2010-11-17/python_snakes_its_way_into_hpc.html
[8] J. Planas, R.M. Badia, E. Ayguadé, J. Labarta (2009) Hierarchical Task-Based Programming with StarSs. *International Journal of High Performance Computing Applications* 23(3): 284-299.
[9] P. Manninen and S. Wong (2011) PRACE-1IP D3.2.3: "PRACE Advanced Training Centres".

List of Acronyms and Abbreviations

ARC	Advanced Resource Connector (grid middleware)
BSC	Barcelona Supercomputing Center (Spain)
CAF	Co-Array Fortran
CaSToRC	Computation-based Science and Technology Research Center (Cyprus)
CEA	Commissariat à l'Energie Atomique (represented in PRACE by GENCI, France)
CFD	Computational fluid dynamics
CINECA	Consorzio Interuniversitario, the largest Italian computing centre (Italy)
CPMD	Car-Parrinello Molecular Dynamics
CPU	Central Processing Unit
CRM	Customer Relationship Management
CSC	Finnish IT Centre for Science (Finland)
CSCS	The Swiss National Supercomputing Centre (represented in PRACE by ETHZ, Switzerland)
CUDA	Compute Unified Device Architecture (NVIDIA)
DEISA	Distributed European Infrastructure for Supercomputing Applications. EU project by leading national HPC centres.
EC	European Community
EPCC	Edinburg Parallel Computing Centre (represented in PRACE by EPSRC, United Kingdom)
EPSRC	The Engineering and Physical Sciences Research Council (United Kingdom)
ETHZ	Eidgenössische Technische Hochschule Zuerich, ETH Zurich (Switzerland)
FETI	Finite element tear and interconnect
FPGA	Field Programmable Gate Array
FZJ	Forschungszentrum Jülich (Germany)
GCS	Gauss Centre for Supercomputing (Germany)
GPAW	Grid-based projector-augmented wave method

GPGPU	General Purpose GPU
GPU	Graphic Processing Unit
HDF5	Hierarchical Data Format
HMPP	Hybrid Multi-core Parallel Programming (CAPS enterprise)
HLRS	High Performance Computing Center Stuttgart (Germany)
HPC	High Performance Computing; Computing at a high performance level at any given time; often used synonym with Supercomputing
HPF	High Performance Fortran
HP-SEE	High-Performance Computing Infrastructure for South East Europe’s Research Communities
IBM	Formerly known as International Business Machines
ICHEC	Irish Centre for High-End Computing (Ireland)
I/O	Input/Output
IPB	Institute of Physics Belgrade (Serbia)
ISC	International Supercomputing Conference; European equivalent to the US based SC0x conference. Held annually in Germany.
JKU	Johannes Kepler Universität Linz (Austria)
JSC	Jülich Supercomputing Centre (FZJ, Germany)
KTH	Kungliga Tekniska Högskolan (represented in PRACE by SNIC, Sweden)
LAPACK	Linear Algebra PACKage
LRZ	Leibniz Supercomputing Centre (Garching, Germany)
MAGMA	Matrix Algebra on GPU and Multicore Architectures
MKL	Math Kernel Library (Intel)
MPI	Message Passing Interface
MPP	Massively Parallel Processing (or Processor)
NAMD	Not (just) Another Molecular Dynamics program
NCSA	National Centre for Supercomputing Applications (Bulgaria)
NetCDF	Network Common Data Format
NUI Galway	National University of Ireland, Galway (Ireland)
OpenCL	Open Computing Language
OpenGL	Open Graphic Library
Open MP	Open Multi-Processing
PATC	PRACE Advanced Training Centre
PETSc	Portable, Extensible Toolkit for Scientific Computation
PGAS	Partitioned Global Address Space
PGI	Portland Group, Inc.
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
PSNC	Poznan Supercomputing and Networking Centre (Poland)
PVM	Parallel Virtual Machine
SARA	Stichting Academisch Rekencentrum Amsterdam (Netherlands)
SGI	Silicon Graphics, Inc.
SMP	Symmetric MultiProcessing
SNIC	Swedish National Infrastructure for Computing (Sweden)
Tier-0	Denotes the apex of a conceptual pyramid of HPC systems. In this context the Supercomputing Research Infrastructure would host the Tier-0 systems; national or topical HPC centres would constitute Tier-1
UNICORE	Uniform Interface to Computing Resources. Grid software for seamless access to distributed resources.
UPC	Unified Parallel C
UYBHM	Ulusal Yüksek Başarımlı Hesaplama Merkezi (Turkey)

VASP	Vienna Ab-initio Simulation Package

Executive Summary

This document describes the work carried out to capture the training requirements of existing and candidate PRACE users. Its primary objective is to assist the planning of the PRACE training programme for the intended target audience, by complementing local training activities as well as taking into account HPC trainer input and availability. The views and training requirements of the trainers themselves have also been captured for a "train the trainers" programme to be implemented by PRACE. In essence, this report describes the implementation and analysis of the results from three surveys.

There were 416 (330 fully completed) responses to the User Training Survey from users distributed all around Europe. The results gave valuable insight into the HPC subjects that are being employed by users and those where training is in demand. For many of these subjects, PRACE is ideally placed to complement local training activities to address user needs. The results have also enabled the examination of training requirements among different user segments and provided an insight into the evolving PRACE user training landscape since 2008. User preferences on training organisation and delivery have also been collected to guide future PRACE training events.

The HPC Trainer Survey collected 205 (166 fully completed) responses from trainers distributed all around Europe, with a few worldwide. The results provided an overview of the HPC training expertise that are currently available and professional opinions on areas where training may be needed. While the range and scope of trainers' expertise appear to meet user demand in some cases, there are gaps where HPC trainers will benefit from a "train the trainers" programme that includes pedagogical training. Significantly, this work has gathered valuable information on the expertise of a large subset of HPC trainers who may be leveraged in future, through the PRACE CRM system, to source the best equipped trainers for PRACE training events.

Information on local training activities was collected through an initial web-based research study using partner websites followed by complementary information provided by respective partners. In all, the training activities in 2011 and 2012 were collected from 14 partners which gave an indication as to where PRACE is best placed to complement training for subjects that are in demand among the target audience.

Taken together, analysis of the demand and supply of HPC training have enabled the identification of several groups of HPC subjects to be addressed by PRACE at varying levels of priority. The subjects of top, immediate priority include performance analysis and optimisation, debugging tools and techniques, and advanced MPI; subjects with a significant high priority include GPU computing, hybrid OpenMP-MPI, parallel program design, architecture and compiler specific optimisation, software engineering, scientific visualisation, OpenMP and Python. Subjects for which introductory courses should be provided to increase uptake include parallel I/O, high-level numerical libraries, PGAS and next-generation languages.

There is evidence that PRACE should disseminate future training events through its own mailing list. While online documentation on the PRACE website and the PRACE Training Portal continues to be a rich source of information and distance learning for users, face-to-face training remains to be the most effective. With the establishment of the PRACE Advanced Training Centres, PRACE is in ideal position to provide highly targeted training for users at convenient locations.

1 Introduction

PRACE has been running a series of HPC training events in its preparatory phase (PRACE-PP) and implementation phase (PRACE-1IP and PRACE-2IP) projects. In 2008, PRACE-PP carried out a survey to establish the training requirements of potential Tier-0 users from PRACE partner countries [2]. The results from this survey led to a number of recommendations to address the immediate, short and long term training requirements for the PRACE Tier-0 user community. These recommendations have subsequently been used to guide the themes of PRACE training events that are organised by PRACE-1IP and beyond.

A number of factors, including the length of time since the last survey, the expansion of PRACE after the preparatory phase, the integration of Tier-1 systems access, and the anticipated elevation of PRACE training activities, have all contributed to the need to carry out an updated survey to examine the training requirements of a larger target audience, i.e. the demands of potential PRACE Tier-0 and Tier-1 users.

Apart from the survey on training demand, knowledge of the existing supply of HPC training was also gained in this work through a survey targeted at HPC trainers and collection of data on local training activities. The comparison between training demand and supply should indicate where PRACE can complement existing training activities. Useful information on the repertoire of HPC trainer expertise was also collected which will assist with the planning and implementation of future PRACE training activities.

Hence this document describes the planning, implementation and outcome of three surveys, with the overall objective to capture a snapshot of the European HPC training landscape that is relevant to PRACE. The surveys are the following:

1. HPC User Training Survey
2. HPC Trainer Survey
3. Local Training Activities

The report is structured in the following manner: Section 2 provides a general background overview of the surveys, their objectives and target audiences. The raw results and some summary statistics are presented in Section 3. Section 4 discusses the results and provides some cross-survey comparisons. Section 5 contains some concluding remarks and a list of recommendations for future PRACE training activities.

2 Background

2.1 User Training Survey

The User Survey was designed primarily to determine the key HPC competencies of existing and potential PRACE Tier-0 and Tier-1 users and their training requirements. While the previous PRACE Training Survey (conducted in 2008, see [2]) had provided a valuable source of questions on different HPC competencies, these questions have been re-factored and new questions were added with the emphasis on whether the respondent (a) knows of a certain competency, (b) whether he/she uses it, and (c) requires some training on the topic. Apart from the key questions on competencies, other questions were asked to guide the development of more effective PRACE training events, focussing on organisation and methods of delivery.

2.2 HPC Trainer Survey

The HPC Trainer Survey was designed to examine the range and scope of HPC competencies that are taught by individual trainers at partner sites, many of whom have the potential to contribute towards PRACE training. One key outcome from this would be knowledge of specific types of expertise that may be leveraged for future PRACE events; respondents were asked if they would like to be invited to participate in future. The results may also provide an indication of areas where there may be a paucity of trainers when compared to training demand from the User Training Survey.

Questions have also been designed to gauge the preferences on the part of trainers with respect to training methodologies, delivery and organisation. Trainers were also asked of their professional opinions on areas where training is needed most to improve the community of HPC users. Lastly, trainers were asked of their own training requirements. The key outcome from this will be to direct future PRACE engagement in “train the trainers” activities.

2.3 Local Training Activities

This is an information gathering process to create a snapshot of the extent and range of local training activities that are carried out by PRACE partner sites. The primary objective is that future PRACE-organised training events will aim to complement local training while taking into consideration user/trainer demands along with trainer resources.

Rather than carrying out a full survey, each partner was requested to provide a list of their local training courses and their delivery in 2011 where available.

3 Results

3.1 User Training Survey

The User Training Survey is composed of 41 questions divided into four main sections:

- Background
  Academic status, country, native language, level of HPC experience, scientific domains, own code and/or third party applications, architectures, maximum core count used, accumulated core hours.
- General HPC training
  Adequacy of training, courses attended and awareness of events organised by PRACE

or others, availability/range/quality/local availability of training, modes of training, language of oral presentations/slides, accessibility, time of training, duration of training, participation in serial/parallel programming courses in university.

- Specific competencies and applications
  Competency, usage and training requirements on different areas such as parallel programming paradigms, programming languages, libraries and code development tools, and other topics.
- Final remarks
  Free-form feedback.

3.1.1 Summary statistics and demographics

A total of 416 responses (330 fully completed) were collected from a target audience of potential PRACE Tier-0 and Tier-1 users, represented by all PRACE partner countries apart from Turkey. Among these are 23 responses from non-PRACE countries and 3 who did not specify their country of origin. A significant proportion of respondents were from Germany (46%) and Italy (9%), due to the blanket dissemination of the survey invite to a large number of past training participants and users rather than a more targeted approach adopted by other partners (i.e. selection of local users/groups who consume the largest amount of resources). However, in most cases there were no important differences between sub-samples consisting of responses with or without Germany/Italy.

The academic status of the respondents is composed largely of graduate students (33%), staff scientists (18%), postdoctoral researchers (17%) and professors (13%).

A combined 72% of respondents have three or more years of experience in HPC, the other 28% with two years or less. The top three domains of expertise represented by respondents are physics (32%), computational fluid dynamics (24%) and computer science (23%)[1].

3.1.2 Profiles of HPC users

62% of respondents mainly run in-house codes, 14% mainly use third-party scientific applications and 19% constitutes a mix of both. Where respondents provided details of the scientific applications used, the packages which are represented most are Gaussian (16), CPMD (14), Quantum ESPRESSO (11), Gromacs (10), NAMD (10) and VASP (10).

Expectedly, a large proportion of respondents are familiar or proficient in the use of HPC clusters (combined 85%) and shared memory systems (combined 77%). Experience with massively parallel architectures (e.g. BlueGene) and novel architectures (e.g. GPU, FPGA, Cell) is less widespread: 51% and 37% of respondents claimed some level of proficiency with these two categories, respectively.

Figure 1 provides a breakdown of the maximum number of cores used to provide an indication of the level of scaling achieved by the respondents. Respondents were also asked to estimate their requirements for accumulated core hours in 2012, results of which are shown in Figure 2.

[1] Respondents may specify multiple scientific domains of expertise.

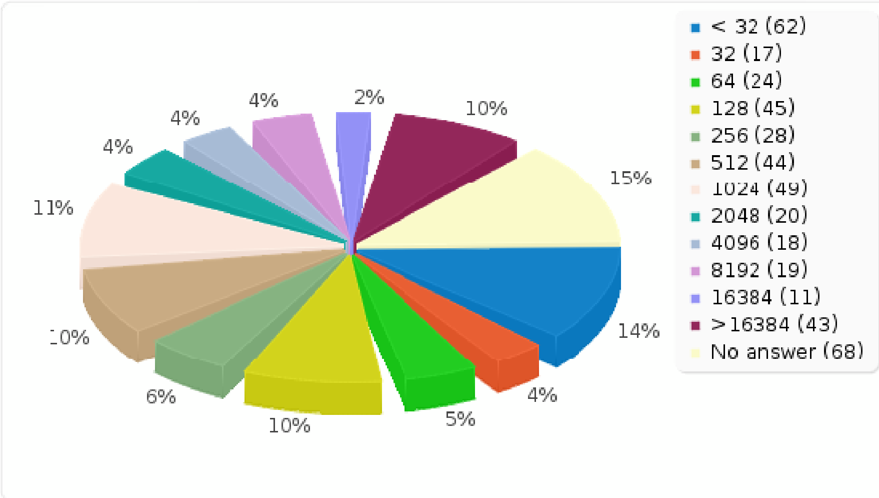


Figure 1: Maximum number of cores that respondents claimed to have achieved scaling, either using own code, or via a third-party application

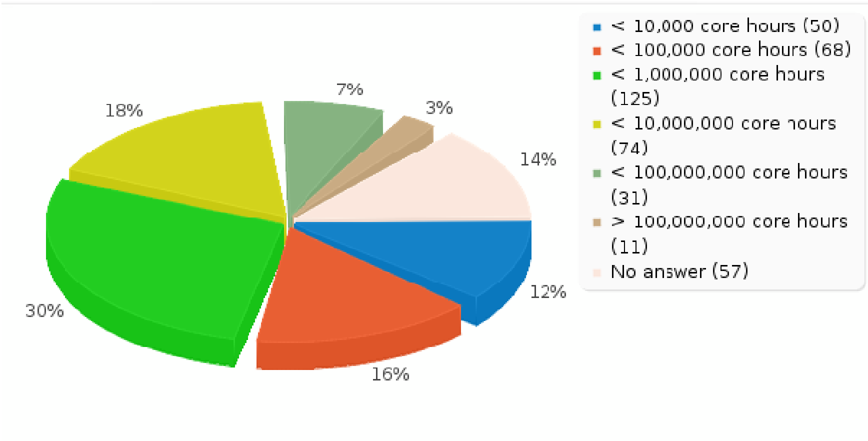


Figure 2: Respondents' estimated requirements of accumulated core hours for 2012

3.1.3 Training methodologies and organisation

Respondents were asked if they have received sufficient training to maximise usage and efficiency of HPC resources available to them. 45% believe that they have, while 44% believe they have not. 21% of respondents have never participated in HPC training courses; 46% attended one to two events; and 30% have attended three or more events. Out of 273 respondents who had participated in training events, 27% (75/273) have attended PRACE training events; 82% (225/273) have attended those that were not organised by PRACE.

Regarding the awareness of HPC training activities that are available, 80% of respondents claimed to have been kept informed of upcoming events; conversely only 13% did not claim any awareness. Mailing lists represent the predominant channel of communication where 64% of respondents use them to be kept informed of upcoming training events. Respondents also use web sites (19%) and word of mouth (16%), but only a small proportion are kept informed through printed advertisements or RSS feeds (combined 7%).

When the same awareness questions were asked but with the focus switched to PRACE, a relatively smaller audience (50%) are kept informed of PRACE training events. Mailing lists still represent the predominant dissemination channel followed by web sites and word of mouth.

A series of statements were then posed to examine the availability, range, quality and local availability of training. 51% of respondents agree that training is available when required, while 13% disagree. Similarly 50% of respondents are satisfied with the range of HPC topics that training is currently provided for, with only 14% in disagreement. 49% are satisfied with the overall quality of HPC training currently available, while only 8% expressed dissatisfaction. 39% of respondents believe that their local HPC centre provides adequate training for their needs, while 25% do not concur.

Respondents were asked to provide their opinions on the effectiveness of various modes of training delivery. The results are shown in Table 1, ranked in order of perceived usefulness.

Mode of training delivery	Not useful	Somewhat useful	Very useful	No answer
Combining lectures with hands-on sessions	1%	11%	81%	7%
Consultation with an expert	2%	17%	70%	11%
Face-to-face classes	2%	30%	57%	11%
Online documentation and tutorials	1%	39%	52%	7%
Printed material	7%	52%	32%	9%
Online discussion forum	15%	47%	28%	10%
Computer-based interactive training courses	22%	45%	19%	14%
Recorded material	27%	48%	13%	12%
Live interactive broadcasts	29%	42%	12%	17%

Table 1: User opinions on the usefulness of different methods of training delivery

Respondents were then asked for their language preference when attending oral presentations and when viewing course slides, the results of which are summarised in Table 2. There is a notable proportion (21%) of non-English native speakers who prefer oral presentations in their own native language. Closer analysis reveals that the majority of this group indicated German as their native language.

Medium	% representing English speakers among respondents, who prefers English	% representing non-English speakers among respondents, who prefers English	% representing non-English speakers among respondents, who prefers own native language	Other
Oral presentations	5%	66%	21%	3%
Course slides	6%	80%	7%	2%

Table 2: Users' language preference for oral presentations and course slides

Respondents were asked whether they had difficulties in attending past training events; 40% have encountered instances when they were unable to attend HPC training events of interest. Within this group, 71% have cited work commitments as the reason, while 41% cited insufficient funding for travel and 32% cited distance of travel as other barriers to attend training.

On the timing and duration of training events, 51% of the audience have no particular preference for which time of year training events are held during. For the rest, summer (17%) and spring (13%) are preferential to autumn (9%) and winter (8%). A large proportion of respondents prefer courses that lasts for 2-4 days (combined 71%), with smaller numbers favouring one-day courses (8%) and those lasting five days or more (11%).

Topics	Know the topic?				Use it?			Require training?			
	Don't know it	Basic	Proficient	Expert / Guru	Never	Occasionally	Frequently or extensively	Immediate	Short-term	Longer-term	Never
Parallel Programming Paradigms											
OpenMP	13%	60%	25%	--	22%	49%	24%	6%	26%	24%	23%
Basic MPI (point-to-point, collective communication)	7%	33%	57%	--	9%	25%	61%	8%	20%	16%	32%
Advanced MPI (MPI-I/O, one-sided communication...)	26%	48%	21%	--	33%	38%	23%	14%	27%	24%	16%
Mixed-mode (hybrid) OpenMP-MPI	37%	45%	13%	--	45%	32%	15%	10%	28%	25%	15%
PGAS languages (CAF, UPC)	80%	13%	3%	--	85%	7%	1%	6%	13%	23%	26%
Next-gen languages (Chapel, X10, Fortress)	89%	5%	1%	--	90%	2%	1%	5%	10%	28%	25%
GPU computing (OpenCL, CUDA)	53%	35%	8%	--	63%	25%	6%	12%	30%	29%	10%
PVM	81%	10%	3%	--	87%	5%	1%	4%	4%	13%	40%
HPF	80%	13%	1%	--	86%	5%	0%	5%	6%	15%	35%
Programming Languages											
C / C++	6%	30%	42%	19%	12%	31%	53%	6%	19%	21%	33%
Fortran 77	16%	34%	32%	16%	33%	33%	31%	3%	5%	7%	58%
Fortran 90, 95...	15%	31%	37%	14%	26%	25%	46%	7%	11%	15%	43%
Java	46%	34%	12%	4%	64%	25%	5%	1%	8%	22%	45%
Python	38%	42%	13%	3%	44%	37%	15%	8%	23%	22%	30%
Matlab / R	33%	43%	17%	4%	47%	33%	15%	2%	14%	22%	40%
Libraries and Code Development Tools											
Basic numerical libraries (LAPACK, EISPACK, …)	15%	50%	25%	4%	24%	42%	26%	4%	20%	26%	28%
High-level numerical libraries (PETSc, Trilinos, …)	53%	32%	7%	1%	63%	19%	8%	7%	17%	30%	24%
Parallel I/O libraries (HDF5, Parallel NetCDF)	50%	32%	10%	2%	50%	26%	13%	10%	20%	29%	22%
General compiler usage and optimisation	6%	36%	42%	11%	9%	35%	50%	12%	25%	23%	20%
Architecture-specific optimisation and tuning	20%	44%	23%	7%	24%	40%	26%	10%	28%	26%	16%
Debugging tools and techniques	12%	49%	29%	4%	13%	48%	32%	14%	29%	25%	14%
Performance analysis/optimisation tools and techniques	19%	43%	26%	5%	23%	45%	23%	16%	31%	23%	13%
Software engineering tools and techniques	28%	40%	18%	6%	40%	31%	18%	11%	26%	30%	15%
Other topics											
Basic UNIX skills	0%	18%	55%	25%	1%	11%	86%	3%	12%	12%	52%
Scripting (shell, PERL, etc)	4%	35%	46%	13%	4%	33%	59%	6%	17%	17%	42%
Batch job systems (job submission and management)	2%	37%	49%	9%	5%	29%	63%	6%	17%	19%	39%
Scientific visualisation tools (e.g. VisIt, Paraview)	33%	35%	25%	3%	39%	31%	25%	12%	23%	26%	22%
Grid interfaces (e.g. Globus toolkit)	59%	27%	6%	2%	73%	15%	5%	6%	12%	20%	38%

Table 3: HPC subjects and their usage, level of competency among respondents and training requirements

Respondents were then asked if they had been taught serial/parallel programming as part of their university education. 71% have received formal teaching in serial programming at university; but for parallel programming, only 20% have been taught this subject while at university.

3.1.4 HPC user competencies and training requirements

This main section of the survey poses to the respondent a number of HPC-related subjects and queries whether the respondent (a) uses it, (b) is competent in it, and (c) requires training on the subject. The results are summarised in Table 3.

Before the end of the survey, users were also asked to provide additional free form feedback on other types of training that should be provided (see Annex 6.2.1 Question 40 for full details). A significant number of responses have requested that training in algorithm design should be provided. In addition, personal comments were also sought with regards to any aspect of HPC training that will be beneficial to respondent's own research and these are available in the Annex 6.2.1 Question 41.

3.2 HPC Trainer Survey

The HPC trainer survey was split into four sections:

- Demographic information and HPC background
  Name, e-mail, affiliation, country, academic status, level of HPC experience, field of specialisation.
- Parallel programming paradigms and languages
  Taught matter, level of trainers' proficiency in various subjects, perceived level of students' knowledge.
- Parallel programming methodologies
  Focus on specified architectures, HPC principles, availability of training materials, importance of software engineering usage, skills in need of development.
- Training organisation
  Preference for different types of training methods, optimal time and duration of training events, preferred language of teaching as well as training demands from the trainers themselves to become better trainers and to improve competencies in HPC subjects.

3.2.1 Summary statistics and demographics

A total of 205 responses (166 fully completed) were collected from HPC trainers represented by all PRACE partner countries (apart from Turkey) but also a small number from non-partner European countries (e.g. Moldova, Slovakia) and further afield (e.g. USA, Australia). Over half (53%) of the trainers have six or more years of experience in conducting HPC training, with 21% having less than two years of experience. The domain backgrounds of the trainers are predominantly in computer science (49%), physics (27%) and mathematics (24%).

3.2.2 Trainer Competencies

A substantial proportion (77%) of trainers stated that their programming language skills have been self-taught. 74% of respondents teach code development, 10% teach third party scientific applications (three packages mentioned most were CP2K, NAMD and Gromacs), with 12% conducting training in both.

Table 4 summarises the competency of those surveyed at providing training under a number of headings. Under each heading, indicated in bold, a list of associated subjects (mirroring those from the User Training Survey) are then ranked in order of decreasing training proficiency, i.e. as one goes down the list under each heading, there are generally less trainers who are sufficiently prepared to teach that particular topic.

3.2.3 Perception of User Competencies

This part of the survey first asked at what level (beginner/advanced) should HPC training focus on, for a variety of HPC architectures (see Annex 6.2.2 Question 20). The responses for symmetric multiprocessing, massively parallel architectures and heterogeneous (mixed architecture) systems gave similar results: trainers indicate that there should be equal focus on both beginner and advanced level training. However, for vector processors and novel architectures (e.g. Cell, FPGA, GPU), the indication is that slightly higher emphasis should be placed at the beginner level. When asked if a sound theoretical background (e.g. in mutual exclusion, computer architecture) is important for the success of HPC training, a combined 63% have responded to say that such background is important or essential.

In a similar manner to the HPC architectures question above, respondents were asked to indicate whether beginner/advanced training should be emphasised for a variety of fundamental HPC principles (e.g. load balancing, data/task decomposition). Here the different principles gave similar results but there are slightly larger numbers who indicated that training in scalability, efficiency, load balancing and code optimisation should be focused at a more advanced level; whereas those in data/task decomposition and check-pointing should focus slightly more towards the beginner level (see Annex 6.2.2 Question 21).

Trainers were asked what skill-sets require the most development. Here the most significant result is that advanced and parallel programming was perceived to require a lot of development (73% of respondents). 72% of trainers also consider important or essential that software engineering techniques to be taught to lead to better practices in developing code.

Subject	Little to no knowledge	Some knowledge	Proficient
Programming paradigms			
Basic MPI	5%	23%	67%
OpenMP	12%	40%	44%
Advanced MPI (MPI-I/O, one-sided communication)	19%	45%	31%
Mixed-mode (hybrid) MPI-OpenMP	21%	41%	34%
GPU computing (OpenCL, CUDA)	40%	38%	18%
PGAS languages (CAF, UPC)	66%	22%	7%
Next-generation languages (Chapel, X10, Fortress)	80%	14%	1%
Programming languages			
C/C++	12%	27%	56%
Fortran 77	15%	31%	49%
Fortran 90, 95, onwards	17%	36%	42%
Java	43%	36%	15%
Matlab / R	48%	35%	12%
Python	50%	36%	9%
Libraries and code development tools			
General compiler usage and optimisation	6%	32%	56%
Debugging tools and techniques	9%	43%	42%
Performance analysis/optimisation tools and techniques	11%	38%	45%
Basic numerical libraries (e.g. LAPACK, EISPACK)	23%	42%	29%
Parallel I/O libraries (HDF5, parallel NetCDF)	47%	38%	9%
High-level numerical libraries (PETSc, Trilinos)	52%	33%	10%
HPC basics, software engineering and visualisation			
Basic UNIX skills	3%	15%	77%
Batch job systems (job submission & management)	3%	26%	64%
Scripting (shell, PERL, etc.)	7%	32%	55%
Version control software (subversion, cvs, git)	10%	45%	39%
Code documentation tools	27%	51%	16%
Check-point/restart implementation	39%	43%	13%
Scientific visualisation tools (VisIt, Paraview)	50%	33%	11%
Grid middleware stacks			
Globus	64%	23%	7%
UNICORE	71%	19%	4%
gLite	74%	14%	7%
ARC	83%	10%	1%

Table 4: HPC trainer competencies

3.2.4 Training methodology & organisation

Similar to what was carried out in the User Training Survey, trainers were asked to provide their opinions on the effectiveness of various training methods. Results are summarised in Table 5 where trainers were asked to rank different methods on a scale of one to five for importance (one means little or no importance; five means very important). Here, combining lectures with hands-on sessions and face-to-face classes represent the most important delivery methods as perceived by the trainers.

Mode of training delivery	Importance
Combining lectures with hands-on sessions	4.66
Face-to-Face classes	4.28
User Guides	3.83
Electronic slides	3.6
Books	3.51
Online web tutorials	3.25
Interactive computer-based training courses	3.29
Journals	2.68
Live web-broadcast	2.54

Table 5: Trainer opinions on different training methods ranked by importance. Respondents were asked to rate importance on the scale of 1-5 and the scores above represent a weighted sum of the ratings.

Among the trainers there is a slight preference to hold events in either spring (51%) or autumn (41%), compared to summer (38%) and winter (34%). Trainers were also asked of their preference for duration of courses aimed at different levels (see Question 28 in Annex 6.2.2). But similar to the User Training Survey, trainers generally prefer courses held over 2-4 days at any level.

A large proportion of trainers either seldom (67%) or have never (23%) been required to teach subjects that they are uncomfortable in teaching. The majority of the respondents (89%) feel comfortable teaching in English. When asked if course prerequisites are used and enforced, 44% do set course prerequisites which are then enforced. However, 32% have taught courses where prerequisites are set but not enforced and 23% do not set prerequisites at all.

3.2.5 Trainers' personal development

In order to understand where PRACE may assist in the development of HPC trainers, possible pedagogical training courses were posed to respondents to assess their usefulness. The results are summarised in Table 19: and discussed in Section 4.1.7, where there seem to be sizeable interest for pedagogical training to be provided. In particular, 51% of respondents considered receiving training on training methodologies to be somewhat useful and 36% considered it very useful.

Lastly, trainers were asked to indicate if they require training themselves on a range of HPC subjects, similar to the list used by the User Training Survey. The results are summarised in Annex 6.2.2 Question 33 and further discussed upon in Section 4.1.7.

3.3 Local Training Activities

Information on local PRACE partner training activities was collected through partner web sites and request for additional input (see Section 6.1.3).

3.3.1 Summary statistics

Table 6 provides an overview of local training events carried out at PRACE partner sites. Information is shown only for those partners for whom information was obtained.

Country and institution	Number of events	Teaching language	Average duration of training events
Austria: JKU	5	English	Semester-long courses
Bulgaria: NCSA	7	English	One-day workshops
Cyprus: CaSToRC	3	English	3-4 day events
Finland: CSC	17	English	2-4 day events and one 14 day training school
France: GENCI	67	English, French	2-4 day events
Germany: GCS	52	Mostly German, some English	1-5 day events
Ireland: ICHEC	16	English	1-3 day events
Italy: CINECA	20	Mostly Italian, some English	1-3 day events and three 5/10 day schools
Poland: PSNC	9	Polish	3-7 hour events
Spain: BSC	4	English or Spanish	1-4 day events
Sweden: SNIC	5	English	2 weeks – 1 month
Switzerland: ETH	11	English	2-3 day events
Turkey: UYBHM	1	Turkish	14 day training school
UK: EPSRC	21	English	1-3 day events and 30 hour lecture courses

Table 6: Local training events at partner sites and training delivery in 2011

It is apparent that each partner focuses on different range of HPC subjects in the training events they host, further details of which can be found in Section 6.1.3. As an indication of the HPC subjects these events cover, Table 7 maps the various events which occurred at PRACE partner sites in 2011 to the subjects that were posed in the User Training Survey based on data collected and where the mapping is feasible.

Subject	No. local training events
Parallel Programming Paradigms	
OpenMP	28
Basic MPI (point-to-point, collective communication)	31
Advanced MPI (MPI-I/O, one-sided communication...)	3
Mixed-mode (hybrid) OpenMP-MPI	6
PGAS languages (CAF, UPC)	7
Next-gen languages (Chapel, X10, Fortress)	4
GPU computing (OpenCL, CUDA)	30
PVM	0
HPF	0
Programming Languages	
C / C++	8
Fortran 77	0
Fortran 90, 95...	10
Java	0
Python	8
Matlab / R	1
Libraries and Code Development Tools	
Basic numerical libraries (LAPACK, EISPACK, …)	3
High-level numerical libraries (PETSc, Trilinos, …)	8
Parallel I/O libraries (HDF5, Parallel NetCDF)	2
General compiler usage and optimisation	4
Architecture-specific optimisation and tuning	5
Debugging tools and techniques	5
Performance analysis/optimisation tools and techniques	8
Software engineering tools and techniques	2
Other topics	
Basic UNIX skills	2
Scripting (shell, PERL, etc)	3
Batch job systems (job submission and management)	0
Scientific visualisation tools (e.g. VisIt, Paraview)	7
Grid interfaces	2

Table 7: Number of local training events held in 2011 that covers HPC subjects corresponding to those in the User Training Survey

3.3.2 Planned future events

On request by the PRACE director Maria Ramalho, partners were asked to provide information on future local training events that would occur in 2012. The information for those partners who provided such information is displayed in Table 8.

Country and Institution	Number of Events	Teaching Language	Average duration of Training Events
Cyprus: CaSToRC	1	English	3 days
Scope of Training: MPI/OpenMP and GPU Parallel Programming. Scalasca and UNICORE training.			
Finland: CSC	11	English	1-5 days 14 day school
Scope of Training: Scientific and High-Performance computing, CUDA, Advanced Parallel Programming, Intro. to MPI, Fortran 95/2003, UNIX, Elmer FEM/intensive course, Scientific Visualisation, Computational chemistry, Bioinformatics with large datasets.			
France: GENCI	67	English, French	2 days
Scope of Training: Includes but not limited to parallel programming (MPI, OpenMP), HPC applications, programming languages (Fortran 95, C, C++), GPU programming and usage of Tier-0/Tier-1 systems.			
Germany: GCS	44	German or English	1-5 days
Scope of Training: Programming in Fortran (various levels). Parallel Programming (MPI, OpenMP, PETSc, Cilk). GPU programming. Parallel I/O and PGAS. Various aspects of R programming language. UPC, CAF. Cray XE6 Optimization. Scripting languages. SimLab porting. Debugging applications, performance analysis. Blue Gene Scaling. Tools and Tuning. Eclipse Parallel Tools Platform. Introduction to CFD. Iterative Solvers and Parallelization. ArBB Training. Iterative Solvers and Parallelization. Visualisation of Large Data Sets. Molecular modelling and systems. Scientific 3D Animation.			
Ireland: ICHEC	8	English	1-3 days
Scope of Training: Introduction to HPC, Introduction to MPI, Introduction to OpenMP, Introduction to R, Advanced MPI, Introduction to CUDA, Software design & Carpentry for Scientists, Introduction to Modern Fortran.			
Italy: CINECA	23	Italian	1-3 days 5/10 day Schools
Scope of Training: Scientific Visualization, Parallel Computing, Introduction to OpenMP/MPI/Fortran90/C/GPGPU/C++/CUDA. Python for computational science.			
Poland: PSNC	10	Polish	3 – 8 hours
Scope of Training: Using LINUX, Programming with CUDA/OpenCL/OpenMP.			
Serbia: IPB	6	Serbian	1/2 day Ecents 14 day school
Scope of Training: EGI training for grid site administrators, EGI training for grid users, Advanced HPC tools and techniques, Introduction to parallel programming, parallel programming, code optimization, porting and benchmarking.			
Turkey: UYBHM	3	Turkish	2-3 days 14 day school
Scope of Training: Linux, Programming in MPI/OpenMP/CUDA/OpenCL, Computational Chemistry, Computational Nanoscience, Parallel Debuggers.			
UK:EPSRC	14	English	2-3/10 days 11 lecture courses

Country and Institution	Number of Events	Teaching Language	Average duration of Training Events
Scope of Training: Advanced OpenMP, Performance Programming, Parallel Design Patterns, Software Development, HPC Ecosystem, HPC Architectures, Message-Passing Programming, Threaded Programming, Parallel Numerical Algorithms, GPU Programming			

Table 8: Planned events in 2012 by local training centres

4 Discussion

4.1 Training requirements

4.1.1 Usage level and competency

To ascertain the level of usage of various HPC methodologies and tools among users, respondents to the User Training Survey were asked to identify various subjects they use in their work and their competency in these subjects. Table 9 shows the key subjects that are used occasionally or extensively by more than 50% of respondents from the User Training Survey, as well as the general competency on these subjects. Overall, there is good correlation between the level of usage and competency among users.

Subject	Usage by respondents (Occasional, Extensive)	Level of competency (Basic, Proficient/Expert)
Parallel programming paradigms		
OpenMP	73% (49%, 24%)	85% (60%, 25%)
Basic MPI	86% (25%, 61%)	90% (33%, 57%)
Advanced MPI	60% (38%, 22%)	69% (48%, 21%)
Programming languages		
C / C++	84% (31%, 53%)	91% (30%, 61%)
Fortran 77	64% (33%, 31%)	82% (34%, 48%)
Fortran 90, 95	71% (25%, 46%)	82% (31%, 51%)
Python	52% (37%, 15%)	58% (42%, 16%)
Libraries and code development tools		
Basic numerical libraries	68% (42%, 26%)	79% (50%, 29%)
General compiler usage and optimisation	85% (35%, 50%)	88% (36%, 52%)
Architecture-specific optimisation/tuning	66% (40%, 26%)	74% (44%, 30%)
Debugging tools and techniques	80% (48%, 32%)	82% (49%, 33%)
Performance analysis/optimisation	68% (45%, 23%)	74% (43%, 31%)
Other topics		
Basic UNIX skills	97% (11%, 86%)	98% (18%, 80%)
Scripting	92% (33%, 59%)	94% (35%, 59%)
Batch job systems	92% (29%, 63%)	96% (37%, 59%)
Scientific visualisation tools	56% (31%, 25%)	63% (35%, 28%)

Table 9: Key subjects that are in use by HPC user respondents and level of competency in each subject

Classical HPC programming languages, such as C/C++ and Fortran, are widely employed by respondents. Parallel programming paradigms such as OpenMP and MPI are also widely used as expected. For these subjects, the level of user competency appears to be sufficient.

For subjects where both usage and competency are relatively low, it can be the case that users simply do not have the necessary knowledge to use it in their work. For example, advanced aspects of MPI are not as widely used and the level of competency amongst users is relatively low. But this is a subject where demand for training is high (Table 3). Furthermore, there are relatively low number of training events that cover this subject (Table 7).

Table 9 also shows some topics where the level of usage and competency are lower, but where demand for training for some are among the highest (Table 3). These include Python, architecture-specific optimisation/tuning, performance analysis/optimisation, as well as scientific visualisation tools. The increasing need for skills in these areas should be taken into account by PRACE for future training events.

Not shown in Table 9 are other subjects with relative low usage levels among users:

- High-level numerical libraries; 27% of respondents have used them (19% occasionally, 8% extensively), 63% have never used them or have no knowledge of their use
- Parallel I/O libraries; 39% of respondents have used them (26% occasionally, 13% extensively), 50% have never used them or have no knowledge of their use.
- Software engineering tools and techniques: 48% of respondents have used them (31% occasionally, 18% extensively), 40% have never used these tools and techniques.

The levels of competency among users in these subjects are also among the lowest where 53%, 50% and 28%, respectively, claimed not to know about them (Table 3). Hence it may be important to raise the level of awareness and competency to increase uptake of these libraries and tools.

It is also notable that day-to-day usage is not very pronounced in the case of GPU programming (63% non-usage) and hybrid OpenMP-MPI programming (45% non-usage). In the context of PRACE, the usage levels of these programming paradigms should be improved. Hybrid OpenMP-MPI codes are important to maximise performance on PRACE Tier-0 architectures with up to 32 cores per node (e.g. CURIE, HERMIT; see [1]). GPU programming is also gaining importance as both Tier-0 and Tier-1 systems incorporate GPU hardware such as the hybrid CPU-GPU partition on CURIE. It is somewhat encouraging that user training demands for these programming paradigms are relatively high, as discussed in Section 4.1.2.

4.1.2 User training demand

The different levels of user demands for HPC training from Table 3 are summarised in Table 10. Here, in the different categories, the subjects are ranked according to the highest immediate and short-term training demand as indicated by users. It also shows the relative proportion of respondents to the HPC Trainer Survey who are proficient at teaching each of the subjects (from Table 4), as well as coverage of each subject by the number of local training events (from Table 7) and PRACE training events held in 2011 (from [5]). Hence Table 10 not only shows the areas of high training demand, but it also provides indication of the level of proficiency held by trainers to meet this demand, and highlights areas where PRACE have begun to address in training in complement to local training events.

Parallel Programming Paradigms

For the various parallel programming paradigms, there are high levels of demand for GPU computing, advanced MPI, hybrid OpenMP-MPI and OpenMP. Such demands are not surprising, perhaps because of the following:

- GPU computing is a relatively novel technology in HPC that is gaining in relevance and popularity, where some of the top supercomputers in the world are equipped with GPU hardware [5]. With many who are new to the technology (only 31% of users use it), it is not surprising that a relatively large number of local training events in 2011 have taken place, but the training demand among users remains. GPU programming has featured in PRACE seasonal schools [6], and it is a subject that PRACE training should continue to cover. It is worth noting that many of the current GPU programming courses are focused towards CUDA; other aspects such as OpenCL, pyCUDA, pyOpenCL and MAGMA could also be considered in our opinion.

- Users are generally comfortable with basic MPI (as already highlighted in Section 4.1.1) but are looking to learn the more advanced features of MPI. In addition, there were relatively few local training events, at least in 2011, which taught advanced MPI compared to the numerous introductory MPI events. The proportion of trainers surveyed who are proficient in teaching this subject is just below average.
- Similarly for hybrid OpenMP-MPI, training demand is high with relatively few local events in 2011 that taught this subject and the proportion of proficient trainers is similar to that for advanced MPI.

Subject	Usage (occasional & extensive)	Training demand (immediate & short-term)	Proficient trainers surveyed	No. local training events (2011)	PRACE-1IP training events (2011)
Parallel programming paradigms					
GPU computing (OpenCL, CUDA)	31%	**42%**	18%	30	4
Advanced MPI	61%	**41%**	31%	3	1
Mixed-mode (hybrid) OpenMP-MPI	47%	38%	34%	6	2
OpenMP	73%	32%	44%	28	0
Basic MPI	86%	28%	67%	31	1
PGAS languages (CAF, UPC)	8%	19%	7%	7	1
Next-gen languages (Chapel, X10, Fortress)	3%	15%	1%	4	0
HPF	5%	11%	--	0	0
PVM	6%	8%	--	0	0
Programming languages					
Python	52%	31%	12%	8	0
C / C++	84%	25%	56%	3	0
Fortran 90, 95...	71%	18%	42%	10	0
Matlab / R	48%	16%	9%	1	0
Java	30%	9%	15%	0	0
Fortran 77	64%	8%	48%	0	0
Libraries and code development tools					
Performance analysis/optimisation tools and techniques	68%	**47%**	45%	8	2
Debugging tools and techniques	80%	**43%**	42%	5	2
Architecture-specific optimisation/tuning	66%	38%	--	5	0
General compiler usage and optimisation	85%	37%	56%	4	0
Software engineering tools and techniques	49%	37%	--	2	0
Parallel I/O libraries (HDF5, Parallel NetCDF)	39%	30%	9%	2	0
Basic numerical libraries (LAPACK, EISPACK, …)	68%	24%	29%	3	0
High-level numerical libraries (PETSc, Trilinos, …)	27%	24%	10%	8	0
Other topics					
Scientific visualisation tools (e.g. VisIt, Paraview)	56%	35%	11%	7	1
Scripting (shell, PERL, etc)	92%	23%	55%	3	0
Batch job systems (job submission and management)	92%	23%	64%	0	0
Grid interfaces (e.g. Globus toolkit)	20%	18%	--	2	0
Basic UNIX skills	97%	15%	77%	2	0

Table 10: Training demands for various HPC subjects in different categories. Dark to light blue shading indicates highest to gradually lower demands for training. Also shown are usage levels for same subjects among users, proportions of proficient trainers from the HPC Trainer Survey, as well as the level of coverage by local training events and PRACE-1IP training events in 2011.

There are still appreciable training demands for basic MPI and OpenMP, even though a relatively large number of local events do cover these topics. However, these subjects are foundations for HPC programming; PRACE should continue supporting such training but with reduced priority compared to other programming paradigms.

The role of the PRACE training programme has traditionally put more emphasis on the advanced parallel programming skills, since these types of events do not appear as often in the local training offering. In passing, a gap between existing local training offerings and the need for advanced skills transfer was one of the key expectations outlined in the PRACE-1IP training plan [3] and this survey corroborates that expectation.

The low demand, relative to other parallel programming paradigms, of PGAS languages, next-generation languages, High Performance Fortran (HPF) and Parallel Virtual Machine (PVM) can be reasoned as follows:

- PGAS languages require high programmer capability as well as platform support to be employed efficiently which are barriers for uptake.
- The next-generation languages are so novel compared to other programming paradigms that they have not been widely adopted by both users and HPC providers for mainstream use.
- HPF and PVM represent legacy technologies and thus there is little interest in them.

While there is insufficient justification for providing training on HPF or PVM, some attention should be given to PGAS and next-generation languages due to their potential importance in programming future HPC systems. It may well be that more basic introductory lectures could increase uptake leading to future demand. It is also important to note that only some HPC centres have the required infrastructure and expertise to carry out this type of training at present.

It is noteworthy to mention here, that from the free-form responses collected as part of the User Training Survey, a significant proportion took the extra time to indicate that training on parallel algorithm design be provided (Annex 6.2.1 Question 40). This should also be considered by PRACE to be included in its training programme.

Programming Languages

It is worth noting that the surveyed users have a more “formal” background education in sequential programming languages when compared to that of parallel programming paradigms. This is illustrated in Table 11 showing the responses to the question of whether users were formally taught sequential and parallel programming as part of their primary university degree, (e.g. if they received ECTS credits for formal courses on the following). With 71% of respondents having received training in sequential programming as part of their university education, this goes some way to explain the lower training demand for sequential programming.

Of all the sequential programming languages, the highest demand for training was observed for Python with 8% and 23% of respondents indicating a need for immediate and short-term training respectively (Table 3). This is higher than the demand for Fortran 90/95 (7% immediate; 11% short-term) and for C/C++ (6% immediate; 19% short-term).

This higher demand for Python training could be due to its ease of use or that Python is not usually taught as a programming language in undergraduate courses in PRACE countries. There is also an increase in the use of Python in HPC and scientific computing in general [7].

There is very little demand for the Java programming language with only 1% of respondents needing immediate training and 8% needing short-term training.

On sequential programming at least, PRACE should include Python in its training plans. An example of such a course is already being offered by CSC (see Section 6.1.4). Intensive courses for HPC programming in Fortran and C would probably be useful, as well as advanced courses for writing efficient sequential code.

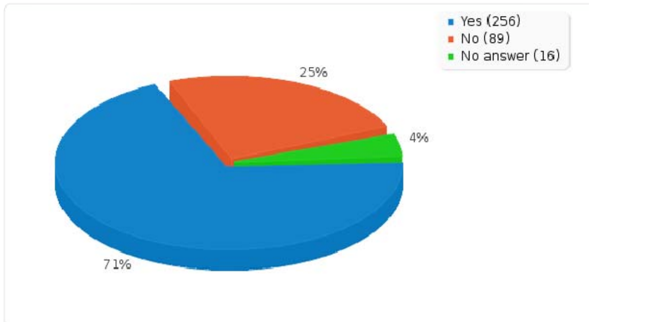
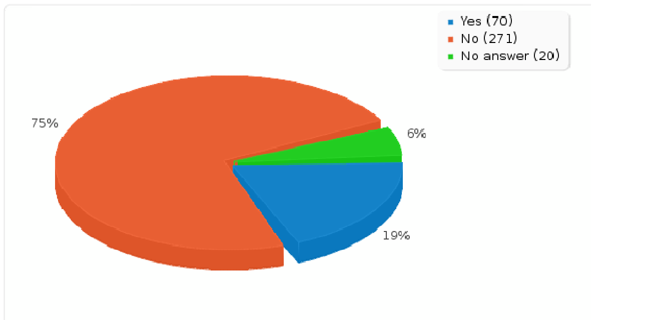
Subject area	Formally taught at university?
Sequential programming (in C, Fortran, etc.)	
Parallel programming (OpenMP, MPI, etc.)	

Table 11: Proportion of users who were taught sequential/parallel programming as part of university education

Libraries and code development tools

There are high training demands across the board for this category. In particular, tools and techniques in performance analysis/optimisation and debugging represent the strongest demand among users for immediate and short-term training (Table 3 and Table 10). The level of local training covering these topics is probably insufficient to meet demand. There are significant opportunities for PRACE to offer training on these. Similar could be said for optimisation techniques for specific architectures or compilers, with the level of training demands relatively less pronounced (Table 10).

For the remaining subjects in this category, it should be noted that some of those with relatively lower levels of training demands (e.g. software engineering tools and techniques, parallel I/O libraries and high-level numerical libraries) also correspond to low levels of usage and knowledge among users (see Section 4.1.1). They are also not as well represented in the number of proficient trainers (Table 10), who identify the same subjects as some of the most in need of development among users (see Section 3.2.3). Hence more could be done for these topics to increase awareness and basic competency.

Other topics

For topics such as basic Unix skills, scripting, batch job systems, scientific visualisation tools, grid interfaces, the demands for training are relatively low in general (Table 10); in most cases only 6% or less of respondents have indicated a need for immediate training (Table 3). The exception is in the case of scientific visualisation tools where the demand is highest within this category at 23% (immediate and short-term); this demand is also relatively strong compared to subjects in other categories (Table 10). It is thus advisable that more emphasis be given to training in scientific visualisation tools and their use in specific fields of science.

The demand for basic Unix skills is the lowest in comparison to other topics in this area, with 52% stating that training would never be required in this area. This is unsurprising as the target audience of the survey are existing HPC users who are likely to be skilled or trained in this area.

Nearly a third of the respondents use third party scientific applications on HPC systems almost exclusively or in combination with their own code (Section 3.1.2); training focused towards scientific applications, or towards specific scientific domains, should also be taken into account by the PRACE training programme.

4.1.3 Comparisons with previous PRACE training survey

In this section, we discuss how demand for training has changed over time by comparing the results from this survey to that carried out in the Preparatory Phase of PRACE in 2008 [2]. Before arriving at definite conclusions, one should bear in mind that the target audience of the current survey comprise Tier-0 and Tier-1 PRACE uses while the 2008 survey focused on Tier-0 users only. There is also a difference in the number of responses (330 in this survey versus 119 in the 2008 survey) and that the current survey covers a broader geographic spread.

In general, the training demand for common HPC subjects has not changed significantly. A more pronounced change can be observed in the urgency for this training to be carried out. Reasons for the reduction in training demand for subjects discussed below could be a result of increased availability of training that have alleviated some of the most urgent demands in the intervening time between the two surveys. Furthermore, as mentioned earlier, the speculations here must be taken in light of the statistical uncertainties between the two surveys.

An example of a difference in training demand is the comparison of demands for hybrid OpenMP-MPI programming. In 2008, two-thirds of survey responses indicated that they had no competency in hybrid programming and thus one of the recommendations of the 2008 survey was to provision immediate training for this topic. According to the current survey, the demand and urgency for hybrid programming is quite different. While the demand for hybrid programming still remain high – and as stated earlier PRACE should keep this subject in its future training plans – the urgency is now less pronounced with just 9% of users requesting immediate training. The PRACE training effort, together with local activities, carried out after the 2008 survey may have met that demand, as can be shown in Table 10.

Similar to the above, the demand for training in the following areas has also decreased:

- Sequential programming, most notably for Fortran 95/2003 and C/C++
- Multi-core programming
    - There is now a greater level of competency among users.
- Visualisation tools
    - Despite current demand remaining high, it has dropped compared to 2008. The difference is either a statistical artefact, or the sporadic training opportunities have not been able to sufficiently address this need.

- Libraries, code and compiler optimization and debugging tools
    - Similar to the above, even though current demands remain (fairly) high, they are much lower compared to results in 2008.

It should be recognised, that demand for training for Unix has decreased dramatically. In 2008, 44% of respondents identified Unix as a field that they would like to receive training on. In the current survey just 15% of users wanted training in Unix (3% in the immediate terms and 12% in the short-term).

Demand has also decreased for training on grid interfaces. The results of the 2008 survey suggested that demand for grid-related training was greater than the current demand. It is possible that grid training activities have addressed some of that demand, or that the target audience of the two surveys are somewhat different.

However some topics have seen an increase in training demand. While GPU programming was not raised in the 2008 survey, it has become the field within parallel programming that is in most demand for training. This can be explained by the dramatic increase in the importance of GPUs for HPC between 2008 and 2011. The first prominent GPU accelerated HPC system featured in the Top500 list of supercomputers in November 2008 [6]. By November 2011, GPUs featured in 37 systems in the Top500 list and three of the top five systems use a mix of both CPUs and GPUs.

A slight increase in demand can be observed in PGAS and next generation languages. As stated in Section 4.1.2, more should be done to increase the awareness and usage of both types of languages due to their potential benefits to HPC.

The demand for training in Parallel I/O is high in both surveys, even though the numbers are lower in the 2011 survey. It seems that there are still relatively few events which cover this topic, at least in 2011. Therefore it is advisable that more should be done for training in this important aspect.

In general, users seem to be more satisfied with training events when compared to 2008 where the quality of events was raised as an issue. Despite this, as shown in Figure 3, the current 51% satisfaction with the overall quality of HPC training currently available remains to be lower than ideal and PRACE should strive to improve the quality of training in the future.

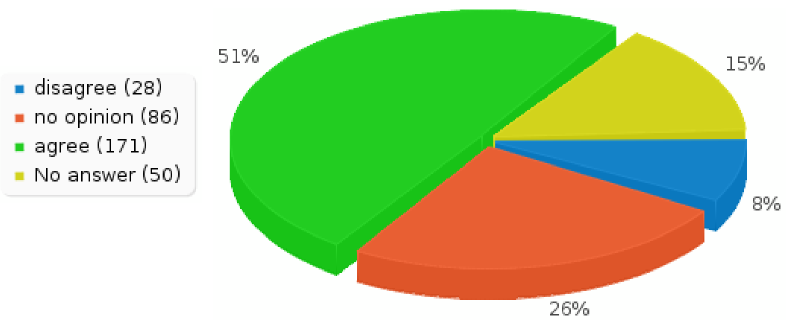


Figure 3: Satisfaction with the overall quality of HPC training currently available

4.1.4 Demands of different classes of users

In this section we identify different classes of users and comment on their respective immediate and short-term training demands. The respondents were classified by proficiency in developing/implementing code (Table 12), accumulated core hours (Table 13), level of HPC experience (Table 14) and types of code they run (Table 15).

Architecture	Number of respondents	Key used in Table 16
HPC clusters advanced proficiency users	160	HPCC Adv
HPC clusters beginner proficiency users	128	HPCC Beg
Shared memory systems advanced proficiency users	99	SMS Adv
Shared memory systems beginner proficiency users	156	SMS Beg
Massively parallel architectures advanced proficiency users	60	MPA Adv
Massively parallel architectures beginner proficiency users	120	MPA Beg
Novel architectures advanced proficiency users	26	NA Adv
Novel architectures beginner proficiency users	97	NA Beg

Table 12: Classification of users based on their proficiency on different architectures

Estimated accumulated core hours	Number of respondents	Key used in Table 16
< 10,000 core hours	35	<10^4 h
> 10,000 and < 100,000 core hours	52	<10^5 h
> 100,000 and < 1,000,000 core hours	107	<10^6 h
> 1,000,000 and < 10,000,000 core hours	65	<10^7 h
> 10,000,000 and < 100,000,000 core hours	27	<10^8 h
> 100,000,000 core hours	9	>10^8 h

Table 13: Classification of users based on their estimated accumulated core hour requirements in 2012

Years of experience in HPC	Number of respondents	Key used in Table 16
< 1 year	36	<1 y
1 - 2 years	53	1-2 y
3 - 5 years	99	3-5 y
6 - 10 years	70	6-10 y
> 10 years	71	> 10 y

Table 14: Classification of users based on years of HPC experience

Types of code	Number of respondents	Key used in Table 16
Mainly run codes developed by themselves and/or their research project	208	Self Code
Mainly use third-party scientific applications	45	3rd Party
A mix of both of the above	66	Mix

Table 15: Classification of users based on the type of code deployed on HPC systems

Table 16 shows an overview of training demand from the different categories of users. As expected, the more experienced users of a category have less training demand in terms of the range of subjects; they also tend to demand training on more advanced and specialised subjects.

It is important to note the number of times performance analysis/optimisation tools and techniques appears as the most in-demand subjects. Similarly, training in general compiler usage and optimisation, as well as debugging tools and techniques, are also in high demand in most user categories. It is also important to note the number of times software engineering tools and techniques has been requested amongst the categories.

It is important for PRACE to consider the demands of different user classes in planning more targeted training activities.

	HPCC Adv	HPCC Beg	SMS Adv	SMS Beg	MPA Adv	MPA Beg	NA Adv	NA Beg	<10^4 h	<10^5 h	<10^6 h	<10^7 h	<10^8 h	<10^9 h	<1 y	1-2 y	3-5 y	6-10 y	> 10 y	Self Code	3rd Party	Mix
OpenMP																						
Basic MPI																						
Advanced MPI																						
Hybrid OpenMP-MPI																						
PGAS languages																						
Next-gen languages																						
GPU computing																						
C / C++																						
Python																						
Basic numerical libraries																						
High-level numerical libraries																						
Parallel I/O libraries																						
General compiler usage and opt.																						
Architecture-specific opt. and tuning																						
Debugging tools and techniques																						
Perf. analysis or op.tools and techs																						
Software engineering tools and techniques																						
Basic UNIX skills																						
Scripting																						
Batch job systems																						
Scientific visualisation tools																						

Table 16: Training demands for different classes of users. Subjects that are deemed to be in-demand by at least 33% of respondents in each category are highlighted with cells in light grey. The top three in-demand subjects within each category are highlighted by dark grey cells.

4.1.5 Supply of trainers

In the trainer survey, trainers were asked on their proficiency on various subjects similar to those in the user survey. Furthermore, they were also asked if they would be open to an invitation to carry out training in future PRACE training events. From the responses, there does not seem to be a deficiency of trainers able to carry out training in future events for most subjects.

Subject	Number of proficient trainers (%)
Next-generation languages	3 (1%)
PGAS languages	15 (7%)
Python	18 (9%)
Parallel I/O libraries	19 (9%)
High-level numerical libraries	20 (10%)
Scientific visualisation tools	23 (11%)
Matlab / R	17 (12%)
Check-point/restart implementation	26 (13%)
Java	27 (15%)
Code documentation tools	33 (16%)
GPU computing	37 (18%)
Basic numerical libraries	60 (29%)
Advanced MPI	64 (31%)
Mixed-mode (hybrid)	69 (34%)

Table 17: Subjects that are the among the least represented by proficient trainers (about a third or less)

Table 17 identifies the areas where less than about a third of the trainers deemed themselves to be proficient trainers. For some subjects there may be insufficient trainers to meet demand. For instance, trainers of next generation languages or new programming paradigms may be largely confined to staff members of the organisation developing such a language (e.g. the StarSs programming model developed by BSC [8]). Based on the shortage of trainers in the areas highlighted in Table 17, there is a strong need for PRACE to continue funding international experts to contribute towards PRACE training events; this has already been done in many of the seasonal schools within the PRACE-1IP training programme. Other subjects in Table 17 that have considerable training demand include Python and scientific visualisation.

Relatively few trainers feel proficient at teaching high-level numerical libraries. This aspect of HPC programming is very important for scientists who do not have a computational/computer science background. Such libraries can potentially implement most of the computation that are required without the user having to “re-invent the wheel”; furthermore considerable effort would have been invested to highly optimise such libraries for maximum efficiency. As indicated in Table 3, 53% of users surveyed do not know this subject, hence its awareness could be improved.

Despite the above and the general coverage of demand by training events, it is important that PRACE should recognise that its training events should not be driven by demand alone. Users do not necessarily know the right content they should be learning. Users are not aware of new paradigms nor are they aware of the potential of techniques/languages they are not familiar with. It is human nature for someone to get by with what they know best, but this may not necessarily be the best option available to them. Based on this, it is important to ensure that potentially important aspects of (future) HPC are not neglected in training events due to the lack of demand in this survey. On the contrary, more effort should be made to ensure that enough training events on such aspects are carried out and that users are encouraged to attend these.

4.1.6 Trainers' perspective on users' training demands

One of the objectives of the trainer survey was to examine how trainers perceived the level of competency amongst users on various subjects. Trainers were asked to assess whether a particular competency was perceived to be sufficient among users, and whether it needs improvement or if it needs a lot of improvement[2].

The aims of this objective are twofold:

- The trainers' opinion on the educational needs of users is first-hand advice on the possible direction and scope of future training events.
- Trainers have advanced foresight on emerging and beneficial developments in HPC that have not yet been made aware or adopted by users.

Shown in Table 18 are subjects where at least two thirds of proficient trainers on that subject considered that knowledge amongst users needed improvement (some improvement or a lot of improvement). The results show that many of the highest ranked subjects that trainers considered improvement is needed also correspond to the ones that users are demanding training (e.g. performance analysis and optimisation, advanced MPI, hybrid OpenMP-MPI programming, GPU computing).

However, there are cases where training on particular subjects is deemed as important by trainers but it is in relatively low demand among users. For instance, there is a strong message from a relatively small number of trainers in PGAS and next-generation programming paradigms that users require a lot of improvements in these areas. As highlighted in Section 4.1.2, awareness and basic knowledge of these paradigms, which may play an important role in the future of HPC, could be improved through more introductory training.

Similarly, there is also an indication from trainers that user knowledge on areas such as software engineering, parallel I/O libraries and high-level numerical libraries requires significant improvement. As mentioned in Section 4.1.1, the level of usage of these libraries, techniques and tools are not high among users, along with few proficient trainers and few local training events that cover these areas (Table 10). Hence PRACE is in an ideal position to provide cover in such areas.

Therefore, while PRACE should focus on user training demands and prioritise accordingly, it is also vital that opinions of expert trainers be taken into account.

[2] It is important to note that trainers were asked if they teach the subject in question. The results that are presented are only for those trainers who teach the subjects in question.

Subject	% of trainers who considered subject requires (some, a lot of) improvement
PGAS languages[3]	91% (17%, 74%)
Parallel I/O libraries	94% (27%, 67%)
Checkpoint/Restart implementation	95% (31%, 63%)
Next-generation languages[4]	94% (33%, 61%)
Performance analysis/optimisation tools and techniques	94% (40%, 53%)
Advanced MPI	90% (36%, 54%)
High-level numerical libraries	95% (49%, 46%)
Hybrid OpenMP-MPI	89% (37%, 52%)
GPU computing	91% (42%, 49%)
Scientific visualisation tools	86% (33%, 53%)
Code Documentation Tools	87% (40%, 47%)
Debugging tools and techniques	94% (53%, 40%)
Grid interfaces	88% (60%, 28%)
Basic numerical libraries	83% (51%, 32%)
General compiler usage and optimisation	86% (63%, 23%)
OpenMP	81% (56%, 25%)
Version control software	76% (48%, 28%)
Python	83% (64%, 19%)
Scripting	76% (52%, 23%)
Fortran 90, 95	75% (53%, 22%)
Basic MPI	66% (43%, 23%)
Batch job systems	71% (54%, 16%)

Table 18: Subjects that proficient trainers consider require development, ranked by scoring each subject where "a lot of improvement" is weighted by a factor of two compared to "some improvement".

4.1.7 Training the trainers

One of the objectives of the trainer survey was to identify ways to improve future PRACE training events. This was done through identifying potential issues in existing training delivery and by identifying potential didactic traits which may require improvement amongst trainers. Furthermore, HPC trainers were asked if they require pedagogical training in order to improve as trainers. Improvements in this regard will potentially lead to higher quality training within PRACE and elsewhere.

To identify potential issues with existing training, trainers were asked about the courses they teach and whether they had to teach something they did not feel completely comfortable with teaching. The results of the trainer responses are displayed in Figure 4. It appears that most trainers have never (23%) or seldom (67%) been asked to do teach a subject outside their comfort zone. However, the percentage of the "never" category could have been greater. For higher quality training events PRACE should strive to deploy trainers who are fully comfortable with the subjects that are being taught.

[3] Note that the number of proficient PGAS trainers is relatively low (23 out of 166 trainers).
[4] Note that the number of proficient next-generation language trainers is relatively low (15 out of 166 trainers).

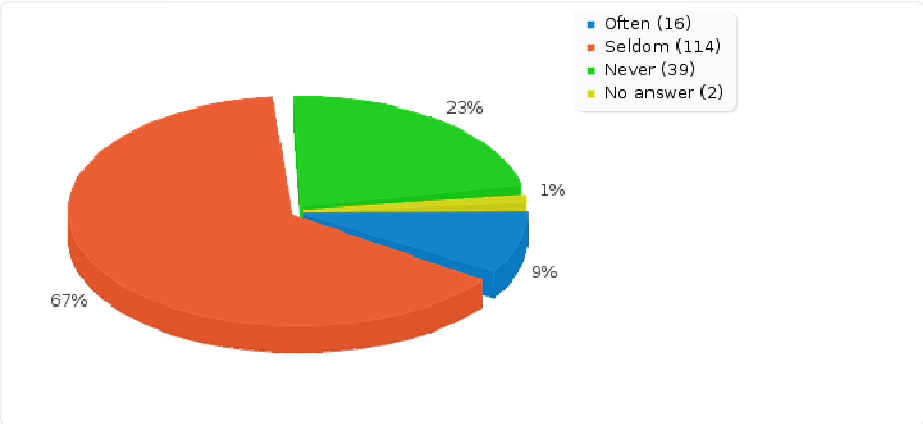


Figure 4: Trainer responses to whether they had been asked to teach a subject that they did not feel comfortable with teaching

Trainers were also asked about the specification and enforcement of prerequisites in the courses they teach. The responses are shown in Figure 5 where the results are far from ideal. Course prerequisites are in most cases essential and should be set and enforced more often. PRACE should ensure that prerequisites are defined and enforced for events it organises.

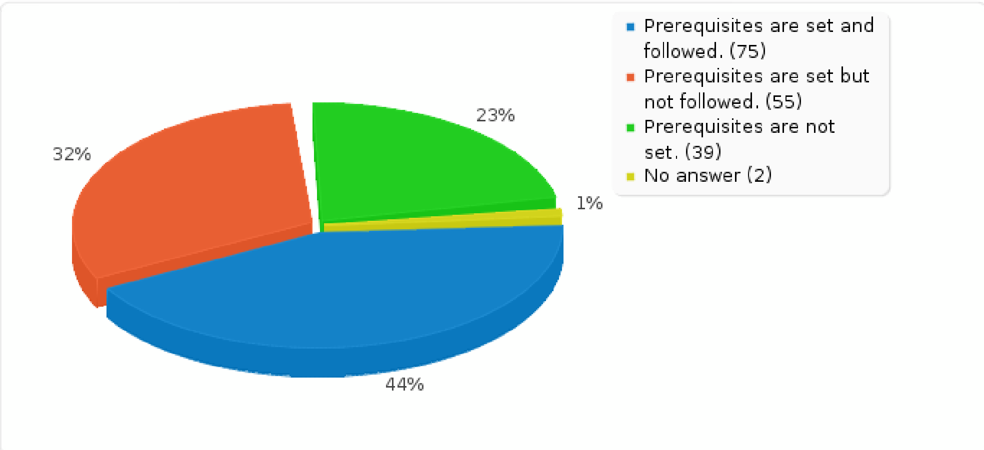


Figure 5: Specification and enforcement of course prerequisites

Not setting or not enforcing prerequisites could greatly reduce the quality of training events. A large number of participants in a training event who do not have the right background could prevent the full curriculum of an event from being covered, as the pace of training would have to be reduced so that the course can be followed by everyone. Another option is to stick with the planned agenda and pace, which would then yield frustrated and unhappy attendees who may not able to follow.

The danger of not setting/enforcing prerequisites may often occur because of intentions to reach out to a large audience, less than expected enrolments, or a lack of training events for everyone to find suitable training for their needs. In the case of the former two cases, it may actually be best to accept a lower number of trainees to an event rather than reducing the level of training to accommodate larger audiences. In conjunction with tackling the latter case, a greater number of events, aimed at different levels to cater for the training requirements of those from different backgrounds, should be made available; something PRACE aims to achieve with the establishment of the PRACE Advanced Training Centres (PATCs) [9].

From the responses of the trainers it is encouraging to note that 91% of trainers who responded are comfortable with conducting training in English, as it is the language used for training in PRACE events and at the PATCs. It is also encouraging that 78% of trainers have agreed to be invited to future PRACE training events to teach the subjects of their expertise. This information will be invaluable for the planning and organisation of future PRACE training in sourcing the most appropriate trainers for different themed events. The details of the trainers will be stored in the PRACE CRM system for this purpose.

Pedagogical skills	**Percentage from 166 responses**		
	Not useful	Somewhat useful	Very useful
Training methodologies	12%	52%	36%
Creating effective slides/handouts	20%	53%	27%
Improving oratory skills	22%	48%	30%
Organizing training events	31%	54%	15%
English technical presentation skills	30%	40%	30%

Table 19: Pedagogical skills and training demands from trainers

Trainers were also asked whether training to improve pedagogical skills would be useful for themselves. Table 19 shows a large proportion (over two thirds) of trainers responding to the survey stated that they would find it useful to pursue training in the listed pedagogical skills. Based on this, it is highly recommended that PRACE take the initiative to initiate a programme of “train the trainers” events to improve HPC trainers as soon as possible. This is likely to improve the quality of training carried out in PRACE events.

Trainers where also asked to identify HPC-related subjects on which they would like to attend courses to improve as trainers. The most requested subjects for training, i.e. those reflecting more than 20% demand, are shown in Table 20. For the purpose of improving training events it is advisable that PRACE address such training demand from the trainers. It is interesting to note that the subjects requested by trainers are similar, for the most part, to the subjects requested for immediate training by HPC users.

HPC subjects where trainers have requested training	**Percentage**
GPU computing (OpenCL, CUDA)	35%
Mixed-mode (hybrid) OpenMP-MPI	32%
Advanced MPI (MPI-I/O, one-sided communication)	30%
Next-generation languages (Chapel, X10, Fortress)	24%
Parallel I/O libraries (HDF5, Parallel NetCDF)	24%
Performance analysis/optimisation tools and techniques	24%
PGAS languages (CAF, UPC)	23%
High-level numerical libraries (e.g. PETSc, Trilinos)	21%
Debugging tools and techniques	20%

Table 20: HPC subjects where trainers have indicated demand for own training, ranked by descending levels of demand

4.2 Training Delivery

Identifying the most effective channels to raise awareness of available HPC training activities is of great importance to reach the target audience. The majority (75%) of respondents to the User Training Survey had attended HPC training courses with 18% having been to training events organised by PRACE; the other events being organised mainly by local HPC centres, universities or other HPC-related EU projects such as HP-SEE or HPC-Europa.

The level of awareness of available HPC training is quite high (80%), with about half of the user survey respondents being aware of PRACE training events. This is an area where PRACE can improve upon. Mailing lists constitute the primary dissemination channel where users receive information about upcoming training (Section 3.1.3). While it has always been requested that partner sites disseminate PRACE training events through local websites/mailing lists, it should be considered that PRACE maintains its own mailing list of interested parties, on an opt-in basis and leveraging the PRACE CRM system, that informs users directly of training events.

However other channels for training dissemination should also be well maintained. Websites have also been identified as the next most important channel (Section 3.1.3). In this regard, the established PRACE Training Portal [4] should have up-to-date information on PRACE as well as partners' local training events.

For delivery of training, both users and trainers have congruent views on the most effective means (see Table 1 and Table 5). By far the most effective means as identified by both groups is face-to-face events that combines a mixture of lectures and hands-on sessions. PRACE is well positioned to deliver on this front with an expansion of its training programme, through the PATCs, to provide a parallel and geographically-distributed series of face-to-face events. Many of these events should also have relatively small class sizes (e.g. 20) compared to PRACE seasonal schools, which should translate to more focused attention to individual participants, where one-to-one consultation with an expert is also highly-valued by user respondents (second most effective means of training delivery in Table 1).

It is also important that other means of training delivery are also addressed by PRACE. Around 45% of user respondents have indicated past instances where there were barriers to attend training events (e.g. work commitments, insufficient funding for travel, distance of travel to training location, insufficient capacity, visa issues or disabilities, see Annex 6.2.1 Question 24). While some of these can be alleviated (e.g. provision of limited travel subsidies, the PATCs are to be geographically distributed to reduce distance of travel for many), online and printed material are in some cases the only means where users may access training. Here PRACE already publishes user documentation and best practice guides on its website, and the PRACE training portal does not only hold static training material (e.g. slides), but also recorded video material as well as a discussion forum where users have indicated a fair degree of usefulness (Table 1).

With regards to the duration of training events, the ideal duration appears to be 2-4 days as reflected by both users (Section 3.1.3) and trainers (Section 3.2.4). Most users have no preference for the time of the year where training events are held, with perhaps slight preferences for spring and summer. The preference of trainers is almost equally distributed among the seasons with spring slightly preferred for holding training events.

Lastly, PRACE is an international association and as such it aims to make HPC training courses and materials accessible to wide range of users across Europe. Therefore the selection of language in which the courses are given is of significance. The participants of the User Training Survey have responded that they prefer oral presentations to be in English in 71% of cases (the sum of 66% of respondents who are non-English native speakers and 5% of respondents who are native English speakers). 21% of respondents who are non-English native speakers preferred the presentations to be delivered in their own native language. A small percentage of non-English speakers stated that they have no preference. Few respondents highlighted that one of the reasons they could not attend the HPC course they wanted was due to a language barrier (event was not carried out in English). The situation is slightly different when it comes to the language of the HPC course slides. The majority preferred English (85%) while only 7% prefer course slides in a non-English language. It is also encouraging that 89% of the available trainers who participated in the PRACE Trainer Survey stated that they feel comfortable with teaching in English.

5 Conclusions & Recommendation

In this document, the planning, implementation and outcome of three surveys – the HPC User Training Survey, the HPC Trainer Survey, and Local Training Activities – have been described. It primarily captures the training demands of existing and candidate PRACE users, with a view towards addressing such demands through PRACE in the context of local training activities and the availability of trainers. The following are some concluding remarks and recommendations for the PRACE training programme.

1. Areas that can be considered as top priority for PRACE training:
    a. Performance analysis and optimisation tools and techniques.
    b. Debugging tools and techniques
    c. Advanced MPI
2. Areas of high priority for PRACE training:
    a. GPU computing (e.g. OpenCL, CUDA)
    b. Mixed-mode (hybrid) OpenMP-MPI programming
    c. Parallel algorithm design
    d. Architecture-specific optimisation and tuning
    e. General compiler usage and optimisation
    f. Software engineering tools and techniques
    g. Scientific visualisation tools (e.g. VisIt, Paraview)
    h. OpenMP
    i. Python
3. Areas where introductory courses should be provided as more “forward-looking” types of training:
    a. Parallel I/O libraries (e.g. HDF5, parallel NetCDF)
    b. High-level numerical libraries (e.g. PETSc, Trilinos)
    c. PGAS languages (e.g. CAF, UPC)
    d. Next-generation languages (e.g. Chapel, X10, Fortress)
4. Areas of lower priority but should be maintained to some level in the PRACE training programme:
    a. Basic MPI
    b. Advanced C/C++ and Fortran (90, 95…) programming
    c. Third-party scientific applications or domain-specific training (e.g. tackling particular problems)

5. Some areas for training can be targeted across many user segments (advanced MPI, debugging techniques, performance analysis, GPU computing); some may be targeted to particular user segments (visualisation tools & techniques, software engineering practices).
6. There is some indication that PRACE and local training events have made inroads into addressing user training demands since the last PRACE training survey was carried out in 2008 (Section 4.1.3). However as indicated above, there are still remaining areas to be addressed.
7. It is important to recognise that the PRACE training programme should not be driven by user demand alone (as documented in this survey or based on other surveys). Expert PRACE trainers should also have an input, especially to ensure user awareness of technologies and paradigms that are potentially important in the future of HPC.
8. The HPC Trainer Survey has gathered and assessed the expertise of 132 HPC trainers who have agreed to be invited to teach at future PRACE training events. Their individual expertise and contact details, easily accessible by PRACE staff using the PRACE CRM system, will provide an invaluable resource for organising training events with specific themes, i.e. finding the best equipped trainers.
9. There is a need for trainers to be trained in both training delivery skills (training methodologies, communication skills) as well as in technical subjects, the range and profile of which are similar to areas of highest user training demand. The PRACE training programme (e.g. the PATC network) should organise a programme of "training the trainers" events in the near future.
10. As a majority of respondents indicated that English is a suitable language for HPC courses, there is insufficient impetus for any localisation of PRACE training. On the contrary, it is important that all PRACE events are accessible to anyone from the PRACE region, and English has been the *lingua franca* of the PRACE community. PRACE training material should also be in English as they can readily be made available through the PRACE Training Portal for dissemination to all users.
11. Face-to-face training events, relatively smaller in size and more targeted at specific user segments, should remain to be the priority to deliver training. However, online training delivery, such as the PRACE training portal, remains to be vital for many PRACE users. PRACE should consider maintaining a mailing list for interested HPC users as the primary dissemination channel for training events.

6 Annex

6.1 Survey implementation and supplementary material

6.1.1 Implementation of User Training Survey

The objective of the User Training Survey is primarily to determine the training requirements of both existing and potential PRACE users that encompass access to both Tier-1 and Tier-0 systems. Questions from the previous training survey conducted by PRACE during the preparatory phase [2] provided the initial material for the design of questions in this deliverable, even though the former was targeted at Tier-0 users. They were re-factored and new questions added to form the final 41 questions (Section 6.2.1).

The User Training Survey is implemented using a survey tool that is set up and maintained by the Institute of Physics Belgrade. It uses the LimeSurvey open source application which provides a vast number of features with user-friendly interfaces for both survey administrators and end users (screenshot shown in Figure 6). This survey tool has also been deployed previously for other surveys, as well as to collect feedback from the participants of PRACE trainings, workshops, symposiums, etc. It is also capable of generating basic statistics and graphs which is convenient for subsequent data analysis.

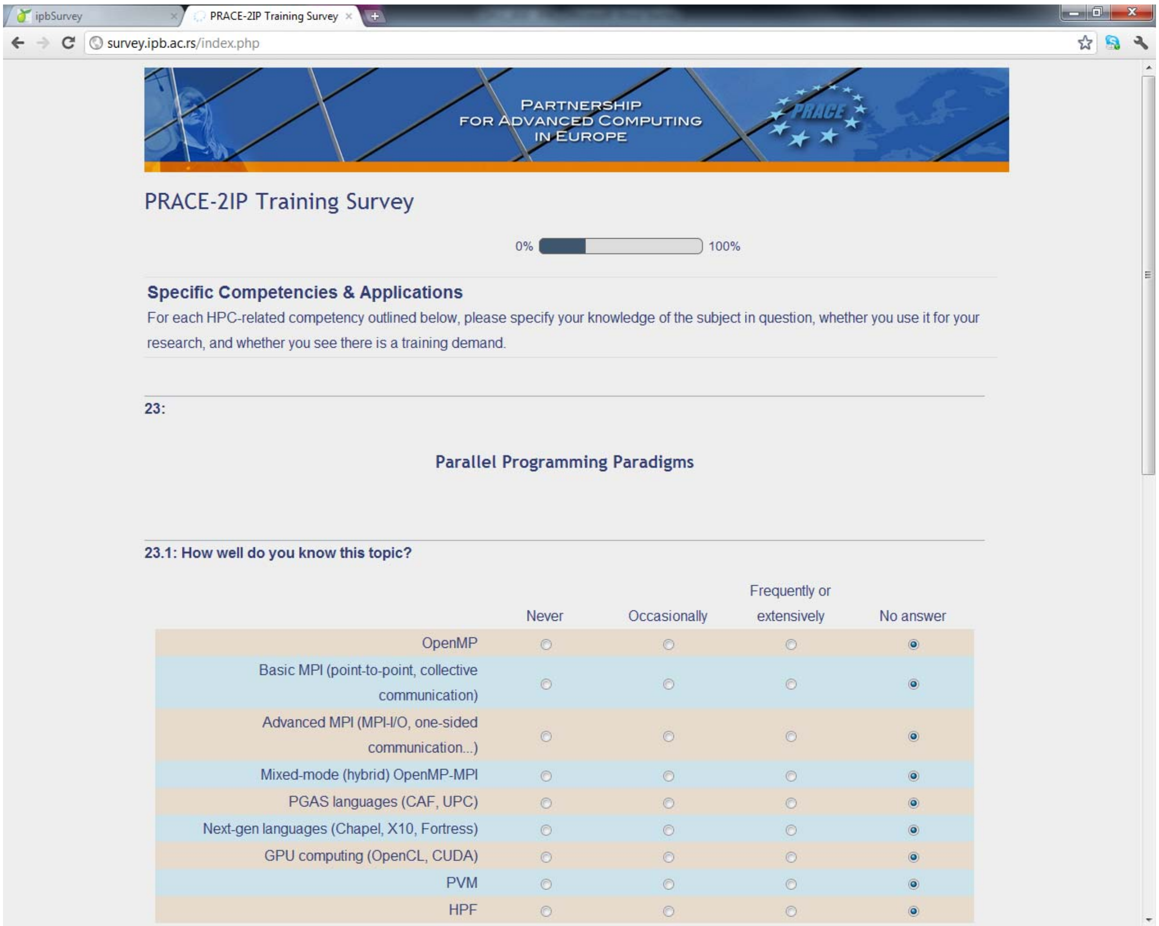


Figure 6: Screenshot of User Training Survey implemented using LimeSurvey

LimeSurvey offers the options for surveys to be anonymous or personalised. For the User Training Survey the anonymous option was used and hence there was no way to trace back and connect survey participants with their responses. No timestamps, tokens or IP addresses are kept in the database and all participants accessed the survey through the same URL. None of the survey questions were made to be mandatory and each question had the “No Answer” option. Questions were divided into several sections displayed in a sequence of separate web pages. If a survey participant gives up before reaching the end of a survey, his/hers responses will be recorded only for the visited sections (responses are sent to the server when user click on the link to the next page). Therefore the total number of answers in the later sections can decrease.

The survey was opened to users from 4th November 2011 until 2nd December 2011. PRACE partners were asked to distribute the survey invite containing the survey URL to their users who are existing or candidate PRACE users, i.e. the groups which consume the most compute resources. The invite also requests that the recipients, some of whom may be non-technical members of research groups, to forward it to other technical members of their group who are developing and/or using applications on high-end HPC resources. This “fan-out” invitation process to target existing/potential PRACE users was similar to what was carried out in the PRACE 2008 training survey [2].

In order to motivate participation, we offered three Amazon Kindle E-book readers as prizes to be awarded to three participants who provided the closest guesses to the final number of “fully-completed” respondents. Only users who fully completed the survey were eligible to enter the competition and PRACE staff members were ineligible to enter.

6.1.2 Implementation of HPC Trainer Survey

In the design phase of the survey it was decided that it was important for the survey to be as complete and concise as possible but at the same time to be kept as short as possible. It was thus imperative to select survey questions that would provide qualitative answers from trainers which would reflect the true judgments of those responding to the survey request. To ensure this, any quantitative or factually based questions, such as information which could be obtained from the web, were thus excluded. This not only allowed for a shorter survey, but a lower amount of effort by the participants.

The questions that comprise the survey were carefully selected so as to ensure the objectives of the trainer survey would be achieved. Additionally through close coordination with all partners involved with implementation of surveys described in this deliverable, some questions were designed to mirror those from the user survey. One purpose of this was to ascertain whether the training requirements of users were similar to that of what trainers consider the requirements to be. This was important to discover, as a difference in opinions between the two sets of parties could possibly identify a false judgment on the abilities of users by the trainers and because of this, improper training techniques could possibly be occurring in some training events.

Some of the questions used in the survey were adopted from a previous trainer survey carried out in the past. These questions were adapted to reflect on new technologies and languages used in HPC systems. Furthermore, new questions were added to cover a broader scope including computer science and software development that are relevant to HPC users. Additional questions were also added to investigate the training requirements of the trainers themselves.

The platform on which the survey was implemented was the LimeSurvey tool hosted at IPB that was also used for the User Training Survey (6.1.1). This tool was chosen as it provides a powerful, flexible question builder interface, automatic collector and analyser modules, as well as a reliable and easy-to-use dissemination and contact reminder capability. Furthermore, the tool is user-friendly both for those designing the survey and for the participants.

The survey invite URL was sent to 281 different trainers with a personalised message explaining the purpose of the survey and respectfully asking trainers to participate. The contact details of these trainers were obtained by an initial search through the websites of all PRACE partners and were later supplemented through a request for more information to all PRACE partners via email. The distribution of the survey invites was carried out using a feature of the LimeSurvey tool which was used to implement the survey.

The HPC Trainer Survey was opened to participants from 3rd of November 2011 until 1st December 2011. Reminder e-mails were sent to trainers on 17th of November 2011 and again on 28th of November 2011.

6.1.3 Implementation of Local Training Activities

The initial collection of local training information was carried out by manually exploring the websites of all PRACE partners. Of the 21 PRACE partners, information on local training events was found for 10 partners. For some of these partners the information on training events was found in a structured and complete manner, for other partners an extensive search of the website was required.

As the collection of information on local training activities can prove to be difficult and those collected from websites may be incomplete, partner sites were asked to provide input. A request to all partners was thus sent requesting validation of the information collection, as well as any additional complementary information that may be missing. Partners were also asked to provide information on future local training events, i.e. those that have been planned for 2012.

6.1.4 Additional information on local training events

In this section we present the areas of training and level of competencies covered by each partner in the past year. We also remark on whether the events of each partner are pre-arranged or if they are delivered on demand. We present information only for those partners for whom information was obtained.

Austria: JKU

The training events of Johannes Kepler University of Linz in Austria are mainly lecture courses which routinely take place every year. They focus more on the architectural aspects of computing systems covering aspects such digital circuits and computer architecture. Further to this, there is a course in parallel computing where parallel programming with Cilk and Cuda is taught.

Bulgaria: NCSA

The training events of the National Centre for Supercomputing Applications in Bulgaria are pre-arranged workshops which mainly focus on BlueGene systems such as Blue Gene/P system software and porting applications to Blue Gene/P.

Cyprus: CaSToRC

The training events of Computation-based Science and Technology Research Centre in Cyprus are arranged training schools mainly focussing on parallel programming methods.

Subjects taught include MPI, OpenMP, Hybrid, CUDA and OpenCL programming. Furthermore, other programming aspects such as parallel programming strategies, benchmarking, debugging and optimisation are explored. Additionally through various programs CaSToRC is involved with it hosts various workshops where HPC can be used – such as in climate research.

Finland: CSC

The IT Centre for Science in Finland has a structured training program with various programs which occur on an annual basis. The centre offers various introductory courses such as introduction to parallel programming with MPI, Fortran 95/2003 and an OpenFOAM introductory course. Python based events include introduction to Python, introduction to Biopython, Python for Scientific Computing and Python in Numerical Computing. Other courses include linkage and association analysis, electronic structure calculations with GPAW, working with Unix and the Elmer FETI workshop. Furthermore CSC hosts an advanced parallel programming event which occurs every six months. Two training events on an introduction to CUDA and an introduction to PETSc were held after demand for these subjects requested for a course of this nature to take place. Additionally, CSC hosted a 14 day Summer School in Scientific and High-Performance computing where training in various subjects in the field was carried out.

France: GENCI

GENCI relies on the three French national computing centres (CEA/TGCC-CCRT, CINES and CNRS/IDRIS) as well as on the Maison de la Simulation and INRIA to provide HPC training.

All the partners provide training in a very wide range of applications ranging from basic programing and HPC skills to advanced state-of-the-art numerical methods. Additionally, they have strong collaborations with external trainers for specific training (e.g. CAPS Enterprise, NVIDIA, Allinea). For example, in 2011, the main training events covered topics ranging from parallel programming (MPI, OpenMP), HPC applications, programming languages (Fortran 95, C, C++), GPU programming and usage of Tier-0/Tier-1 systems.

Germany: GCS

The GAUSS Centre for Supercomputing in Germany has the most comprehensive training program amongst all PRACE partners. In 2010 it hosted 47 different courses and provided training for around 1200 students. Similarly, in 2011 its training program was complete and full of events. 52 different courses took place and close to 1100 students took part in these events. The number of days for training events at GCS are between 1-5 with an average of 2.8 days overall. Most of the training events at GCS are taught in German with the rest of them taught in English.

The range of courses taught at GCS is great. Like most PRACE partners, GCS also delivers training in parallel programming paradigms such as GPU programming with CUDA, programming with MPI, OpenMP, PETSc, PGAS, unified parallel C, Co-Array Fortran amongst many others. Furthermore, specialised training events exploring scripting languages (such as Python, R and Matlab) in HPC are also offered as well as providing programming courses using sequential languages in HPC.

Furthermore, GCS offers specialised courses such as training in the Eclipse Parallel Tools Platform, Intel array building blocks training, Iterative solvers and parallelization, Cray XE6 Optimization amongst many other subjects.

Additionally, GCS provides training events which introduce attendees to various fields of science where HPC is used, such as an Introduction to CFD amongst others.

Ireland: ICHEC

The Irish Centre for High-End Computing in Ireland operates on an "on demand" basis, delivering various training events to institutions whenever such training is required. These events cover various aspects of HPC such as four introductory courses in MPI, OpenMP, modern Fortran and CUDA as well as a course covering an introduction to HPC. Other training events hosted by ICHEC include software engineering and carpentry for scientists as well as an advanced course in MPI.

Italy: CINECA

CINECA in Italy mainly operates on an arranged schedule of training events delivering most of events in Italian. The scope of the training events are mainly introductory lectures of 1-3 days length covering various programming aspects such as OpenMP, MPI, Fortran90, C, GPGPU, object orientation in C++ and CUDA. Two of these introductory events were on an on-demand basis. Additionally CINECA carried out events exploring tools and techniques in HPC environments as well as Python for computational science. Furthermore, CINECA hosted three schools where different areas of scientific computing were explored.

Poland: PSNC

The Institute of Bioorganic Chemistry at the Poznan Supercomputing and Networking Centre in Poland mainly carries out training events for Polish scientist as all their events are in Polish. The events that occur are pre-arranged and train attendees in subjects such as CUDA, OpenMP and Grid computing.

Serbia: IPB

The Institute of Physics Belgrade in Serbia hosted four one day events which were hosted as a result of on demand requirements by users. These events focused on various aspects such as input preparation, job submission and monitoring, PARADOX Cluster job management and PARADOX software stack, grid site administration and ways to utilise grid resources for the AEGIS user community. It also hosted a fourteen day event focusing on various aspects such as implicit parallelism, core performance optimization, MPI and mixed OpenMP/MPI programming, scientific data management, NetCDF, parallel filesystems, GPU programming, CPU, GPU benchmarking and the IBM BlueGene architecture amongst others.

Spain: BSC

The Barcelona Supercomputing Centre in Spain hosted two different training schools with programming aspects of HPC being the main focus of the events. Aspects covered in these events included programming models, MPI, OpenMP, Chapel and accelerator programming. Within the scope of programming HPC systems, debugging tools, parallel debugging, parallel file systems, performance analysis and performance tools were also covered. Additionally, BSC hosted more specific training events focusing on HPC aspects in physics, biology, bioinformatics and cosmological simulations.

Sweden: SNIC

The Swedish National Infrastructure for Computing in Sweden has an arranged semester long course which focuses on algorithms and applications of high performance computing. Additionally, two mini courses in computational methods in statistics with applications and programming in science and technology of time length 2.5 weeks and 4 weeks respectively are also offered. SNIC also hosted two schools in the past year the first of which was an introductory school in High Performance Computing and the second one focusing on more advanced aspects of High Performance and Grid Computing.

Switzerland: ETH

The Swiss Federal Institute of Technology in Switzerland hosted a fair number of 2-3 day training events. Some of these events were specific to one subject such as scalable performance analysis tools for HPC applications, multi-threaded programming, tuning and optimization on multi-core MPP platforms, GPU programming with CUDA Fortran and the PGI accelerator programming model, a "Hands-On" introduction to PETSc, lectures on parallel programming and HPC storage.

Turkey: UYBHM

The National Centre for High Performance Computing at Istanbul Technical University in Turkey hosted an annual High Performance Computing and Parallel Programming summer school with Turkish being the training language. Various training tracks took place covering the subjects of basic Linux, parallel programming with MPI/OpenMP, computational Chemistry, computational Nanoscience and GPGPU (CUDA and OpenCL) Programming.

UK: EPSRC

The Engineering and Physical Sciences Research Council in the UK represented by EPCC at The University of Edinburgh has a wide-ranging number of HPC related lecture courses it teaches its students. These include object oriented programming for HPC, parallel decomposition, computer simulation: Techniques and applications, Message-Passing programming, Parallel numerical algorithms, HPC Architectures amongst others. Furthermore, EPCC hosts other 1-2 day events in which other scientists (non-EPCC students) can attend which mainly focus on various aspects of HPC programming. Examples of such events include GPU programming with CUDA, parallel programming with Co-Array Fortran, the Scalasca performance analysis toolset, advanced OpenMP programming, hybrid MPI/OpenMP programming.

6.2 Complete Survey Statistics

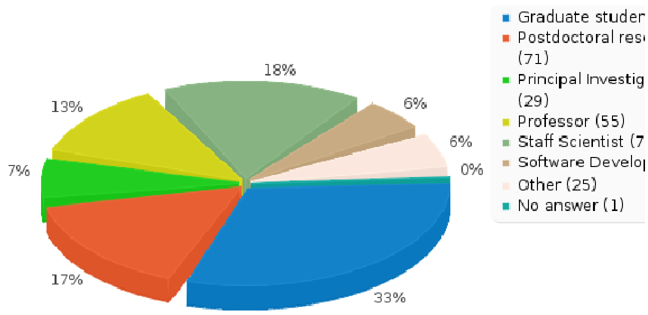
6.2.1 User Training Survey

Background

1. What is your academic status?

Answer	Count	Percentage
Graduate student	139	33.41%
Postdoctoral researcher	71	17.07%
Principal Investigator	29	6.97%
Professor	55	13.22%
Staff Scientist	73	17.55%
Software Developer	23	5.53%
Other	25	6.01%
No answer	1	0.24%

Chart 6.2.1-1

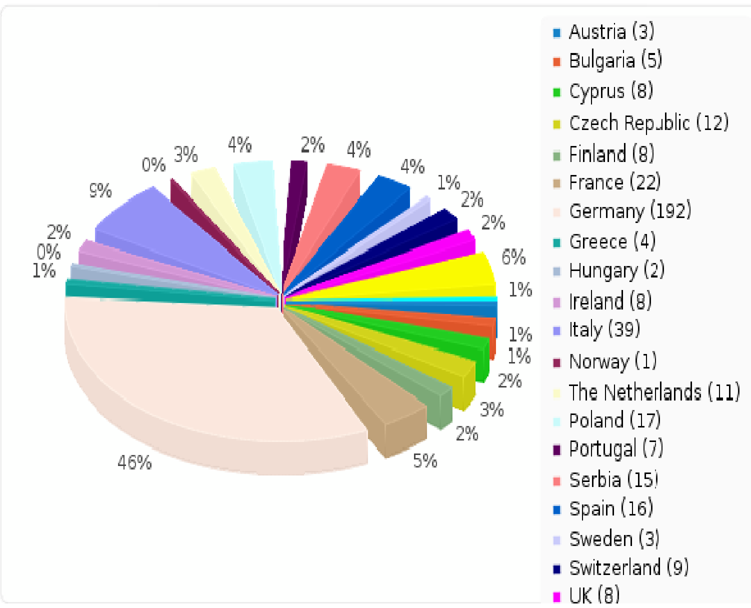


Other: Doctoral Student; Grid site manager; Industry / SW Solution Architect; Manager; 2x Master student; onsite application support; PhD Candidate; 8xPhD student; Research associate; 3xResearch Engineer; Research Engineer; 2xResearcher; Senior Scientist; Staff engineer; System Administrator

2. In which country do you mainly live and work?

Answer	Count	Percentage
Austria	3	0.72%
Bulgaria	5	1.20%
Cyprus	8	1.92%
Czech Republic	12	2.88%
Finland	8	1.92%
France	22	5.29%
Germany	192	46.15%
Greece	4	0.96%
Hungary	2	0.48%
Ireland	8	1.92%
Italy	39	9.38%
Norway	1	0.24%
The Netherlands	11	2.64%
Poland	17	4.09%
Portugal	7	1.68%
Serbia	15	3.61%
Spain	16	3.85%
Sweden	3	0.72%
Switzerland	9	2.16%
Turkey	0	0.00%
UK	8	1.92%
Other	23	5.53%
No answer	3	0.72%

Chart 6.2.1-2

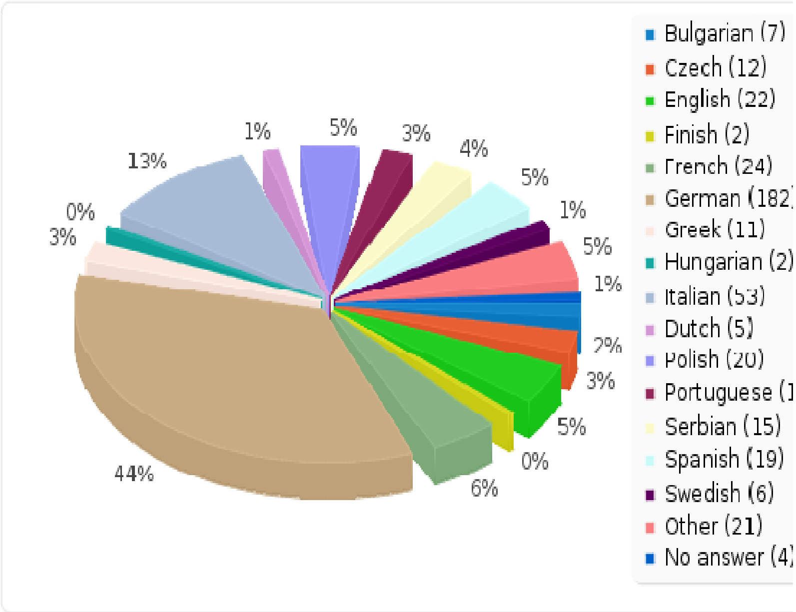


Other: 11xBelgium; Brazil; Canada; India; México; Pakistan; Russia; Slovakia; 5xUSA

3. What is your native language?

Answer	Count	Percentage
Bulgarian	7	1.68%
Czech	12	2.88%
English	22	5.29%
Finish	2	0.48%
French	24	5.77%
German	182	43.75%
Greek	11	2.64%
Hungarian	2	0.48%
Irish	0	0.00%
Italian	53	12.74%
Norwegian	0	0.00%
Dutch	5	1.20%
Polish	20	4.81%
Portuguese	11	2.64%
Serbian	15	3.61%
Spanish	19	4.57%
Swedish	6	1.44%
Turkish	0	0.00%
Other	21	5.05%
No answer	4	0.96%

Chart 6.2.1-3

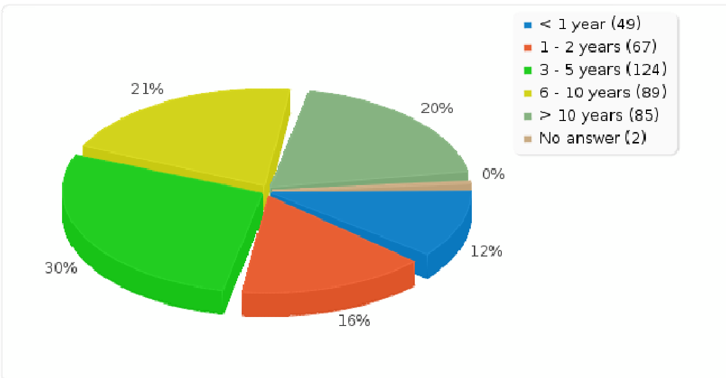


Other: 2xArabic; Belarusian and Russian; Bengali; 3xChinese; Indonesia; Luxemburgish; Persian; 3xRomanian; 5xRussian; Slovak; 2xUrdu;

4. How many years of experience do you have in High Performance Computing (HPC)?

Chart 6.2.1-4

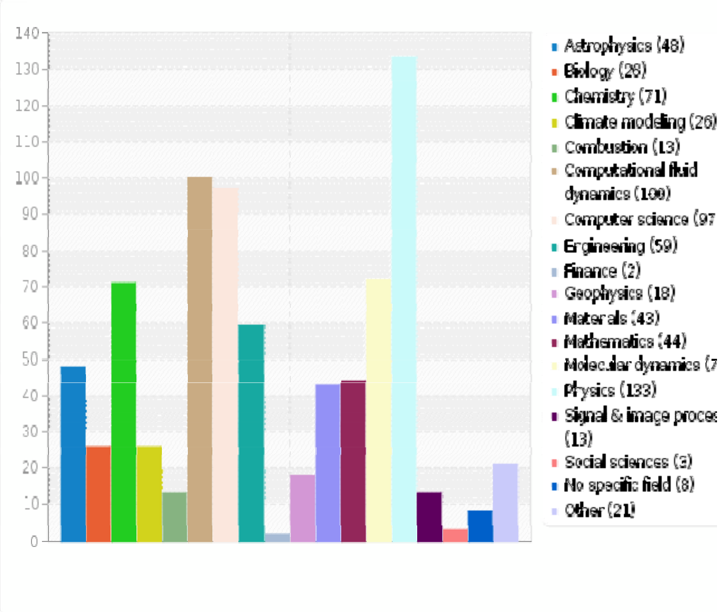
Answer	Count	Percentage
< 1 year	49	11.78%
1 - 2 years	67	16.11%
3 - 5 years	124	29.81%
6 - 10 years	89	21.39%
> 10 years	85	20.43%
No answer	2	0.48%



5. In which scientific domain do you utilise HPC?

Answer	Count	Percentage
Astrophysics	48	11.54%
Biology	26	6.25%
Chemistry	71	17.07%
Climate modelling	26	6.25%
Combustion	13	3.12%
Computational fluid dynamics	100	24.04%
Computer science	97	23.32%
Engineering	59	14.18%
Finance	2	0.48%
Geophysics	18	4.33%
Materials	43	10.34%
Mathematics	44	10.58%
Molecular dynamics	72	17.31%
Physics	133	31.97%
Signal & image processing	13	3.12%
Social sciences	3	0.72%
No specific field	8	1.92%
Other	21	5.05%

Chart 6.2.1-5

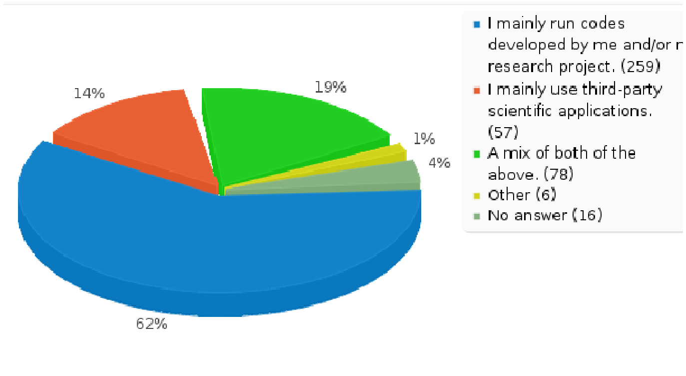


Other: Astronomy, orbits dynamics; bioinformatics; Celestial Mechanics; Climatology; computational aeroacoustics; Computational electromagnetics; Cryptography; Earth System Modeling; electromagnetics; Environmental Science; Flow Simulationgeotechnics; Individual-based population models; IT; machine learning; Oceanographyplasma physics; Power Systems; Power Systems; Robotics; Statistical physics

6. What type of codes do you run on HPC systems?

Answer	Count	Percentage
I mainly run codes developed by me and/or my research project.	259	62.26%
I mainly use third-party scientific applications.	57	13.70%
A mix of both of the above.	78	18.75%
Other	6	1.44%
No answer	16	3.85%

Chart 6.2.1-6



Other: MD software: Gromacs; code maintained by own institute; CST MWS; optimization and scaling support for existing sw; Applications developed by our company, based on commercially available solvers; Operation System, runtime, middlware

7. Please specify the third party scientific application(s) that you run on a regular basis.

Only answer this question if the following conditions are met:

° Answer was 'I mainly use third-party scientific applications.' or 'A mix of both of the above.' at question '6 [6]' (What type of codes do you run on HPC systems?)

- *Please write your answer here:*
- *Abaqus, CFX, Fluent, Nastran, Openfoam, Pamcrash, Powerflow, StarCCM+, StarCD*
- *ABINIT, CRYSTAL, VASP*
- *aces2, amber, gamess, gaussian, molpro, mopac, qchem*
- *ALADIN-Climate model, MESO-NH model*
- *ALPS (alps.comp-phys.org)*

- *amber gromacsgamess*
- *AMBER, CHARMM*
- *AMBER,CHARMM,VMD*
- *at the moment only: ANSYS CFX in the near future also: OpenFOAM*
- *Atmosphere, ocean models and coupler*
- *Bioinformatics: RAxML, MrBayes Scientific: Benchmarks from PARSEC*
- *CAMx by ENVIRON (an air quality model)*
- *CASTEP (www.castep.org) LAMMPS (lammps.sandia.gov) DL_POLY (www.cse.scitech.ac.uk/ccg/software/DL_POLY/) elk (elk.sourceforge.net)*
- *Chroma (http://usqcd.jlab.org/usqcd-docs/chroma/)*
- *Climate models developped by other groups in Europe*
- *COMSOL, MATLAB*
- *cp2k*
- *CP2K, Gromacs, CPMD, DL_POLY*
- *CPMD PWSCF (Quantum Espresso)*
- *CPMD VASP Gaussian Molpro NWCHEM*
- *CPMD*
- *CPMD*
- *CPMD*
- *cpmd (Car-Parrinello Molecular Dynamicd cpmd.org)*
- *CPMD, CP2k*
- *Cpmd, cp2k, lammps*
- *CPMD, CP2k, Turbomole, Gaussian*
- *Density-functional theory code*
- *Earth System Models (EC-Earth) and Ocean Model (NEMO)*
- *EC-Earth (ESM) and NEMO (Global Ocean Model)*
- *ECHAM climate model MPI Hamburg*
- *Electromagnetism, seismic imaging, material physics (crack propagation, dislocation dynamics)*
- *EMAC ECHAM5-MESSY (Modular Earth Submodel System)*
- *EMAC (ECHAM/MESSy), WRF*
- *EMAC atmospheric chimestry and climate model Weather Research and forecast (WRF) model.*
- *Enzo cosmological code*
- *finite volume LES*
- *FLASH PLUTO*
- *FLOWer DLR*
- *Fluent of ANSYS*
- *fluent, abaqus, alps, quest, matlab,*
- *Gadget-2*
- *GADGET2 GalactICS*
- *Gadget2*
- *gadget3 tree-Nbody + sph code*
- *Gaussian ADF*
- *Gaussian NAMD*
- *Gaussian Turbomole*
- *Gaussian TurboMole*
- *Gaussian 09 Molcas 7.6*
- *Gaussian, Amber*
- *Gaussian, Gamess-US, NWChem, Molden*
- *Gaussian, GAMESS, molpro, DLPOLY*
- *gaussian, vasp, cpmd, molpro, turbomole*
- *Gromacs*
- *Gromacs*
- *Gromacs, Amber, NAMD*
- *gromacs, espresso, charmm*
- *GROMACS, Gaussian*
- *GROMACS, NAMD, MAESTRO SCHRODINGER, GAUSSIAN, VMD*
- *I mainly run codes developed by me which are based on PETSc*
- *LAMMPS DL-POLY Q Espresso CPMD GROMACS*
- *LAMMPS*
- *LAMMPS, CPMD*
- *LAMMPS, NAMD, AMBER, Materials Studio, VMD*
- *LAMMPS, PEtot, ESCAN, VASP, NWChem, Quantum Espresso*
- *LESOCC2 - KIT (Institute of Hydormechanics) PARCOMB3D - University of Magdeburg*
- *MATLAB, COMSOL*
- *MD - amber QM - gaussian, molprofluidynamis - fluent, comsol*
- *Meteorological Models Chemistry Transport Models*
- *MOLCAS - program package for quantum chemistry CP2K - program package for molecular dynamics*
- *Molecular dynamics simulation applications, molecular docking applications*
- *MOPAC, TURBOMOLE, VASP, BigDFT, Gromacs*
- *MPI implementations: Open MPI, MPICH2, MVAPICH2, Intel MPI, etc. Benchmarks: NAS Parallel Benchmarks, Intel MB, Linpack, etc. Applications: Own developed*
- *NAMD*
- *NAMD (2.8), AutoDock-Vina (1.0)*
- *NAMD, Amber (sander, pmemd), gaussian*
- *NAMD, BLAST, Clustal, R*
- *NAMD, DLPOLY*
- *NAMD, Gromacs, Amber*
- *NAMD, GROMACS, AMBER, Gaussian*
- *NAMD, GROMACS, Rosetta, DOCK 6, Scalasca*
- *nek5000*
- *NEMO, WRF, Direct Simulation Toolkit.*
- *Network Simulator 3 (ns3)*
- *Network simulator NS3, R*
- *numerical climate/earth system models*
- *octopus (http://www.tddft.org/programs/octopus/wiki/index.php/Main_Page) cpmd (http://www.cpmd.org/)*
- *octopus, gpaw*
- *OpenFOAM*
- *OpenFoam, Palabos, Ansys CFX, Fluent*
- *OpenFoam, WRF*
- *packages like Trilinos*

- *PARSEC, SPEC2000*
- *PETSc*
- *PETSc*
- *PETSc (http://www.mcs.anl.gov/petsc/petsc-as/) , Getdp (http://geuz.org/getdp), Gmsh (http://geuz.org/gmsh)*
- *PLUTO (http://plutocode.ph.unito.it/)*
- *PWSCF Yambo GWW CASINO*
- *Quantum chemical program packages to calculacte molecular energies and geometries*
- *Quantum Espresso*
- *Quantum espresso, CP2K*
- *QUANTUM ESPRESSO, CPMD, CP2K, TINKER, DL_POLY*
- *Quantum Espresso, Yambo, NWChem*
- *quantum-espresso abinit siesta yambo*
- *Quantum-Espresso CPMD CP2K SIESTA Crystal09*
- *R statistical software Octave*
- *SIESTA*
- *SIESTA, NAMD, DFTB+, gDFTB*
- *siesta, octopus*
- *Sofi3D (gephysical code from Karlsruhe) SPARC (Code also from Karlsruhe)*
- *TAU (Finite Volume Sovler) developed by the German Aerospace Agency (DLR) to simulate transonic flow problems*
- *TAU CFD code from DLR*
- *The Chroma library for lattice field theory, http://usqcd.jlab.org/usqcd-docs/chroma/*
- *TINKER, WebKit*
- *tmLQCD, Chroma*
- *Usage of availibleneutronic codes, deterministic and montecarlo*
- *VASP*
- *VASP*
- *VASP*
- *VASP*
- *VASP, DL_POLY, GULP, WIEN2K*
- *VASP, Qbox*
- *Vienna ab initio simulation package (VASP) CPMD Octopus GULP*
- *Weather Research and Forecasting model*

8. Please describe your proficiency in developing/implementing code on the following architectures:

	None		Basic		Advanced		No Answer	
HPC clusters	38	9.13%	162	38.94%	195	46.88%	21	5.05%
Shared memory systems	73	17.55%	198	47.60%	121	29.09%	24	5.77%
Massively parallel architectures (e.g. BlueGene)	170	40.87%	135	32.45%	78	18.75%	33	7.93%
Novel architectures (e.g. GPU, FPGA, Cell)	232	55.77%	121	29.09%	31	7.45%	32	7.69%

Chart 6.2.1-7: HPC Clusters

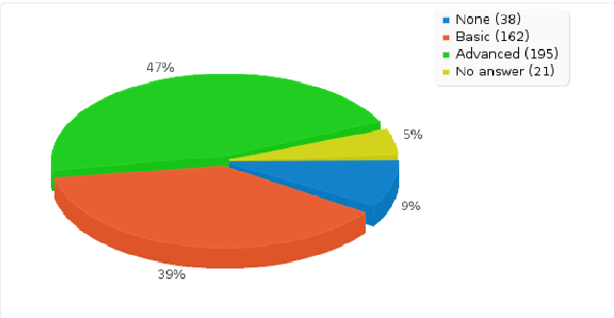


Chart 6.2.1-8: Shared memory systems

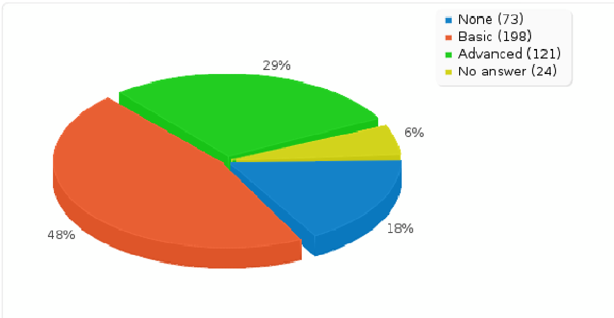


Chart 6.2.1-9: Massively parallel architectures

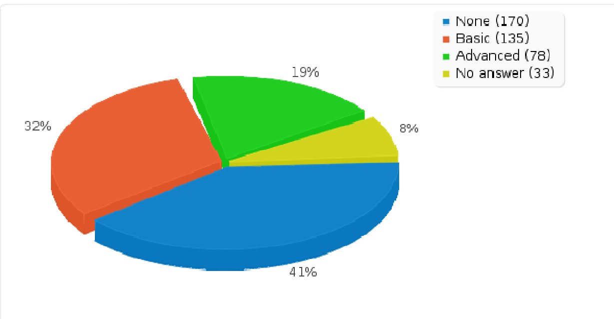
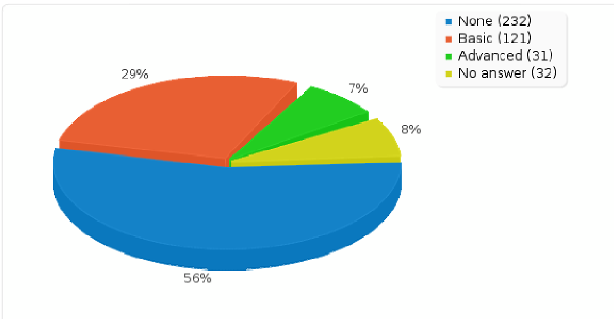


Chart 6.2.1-10: Novel architectures



9. If you are experienced with developing/implementing code on other architectures, please specify and describe your proficiency.

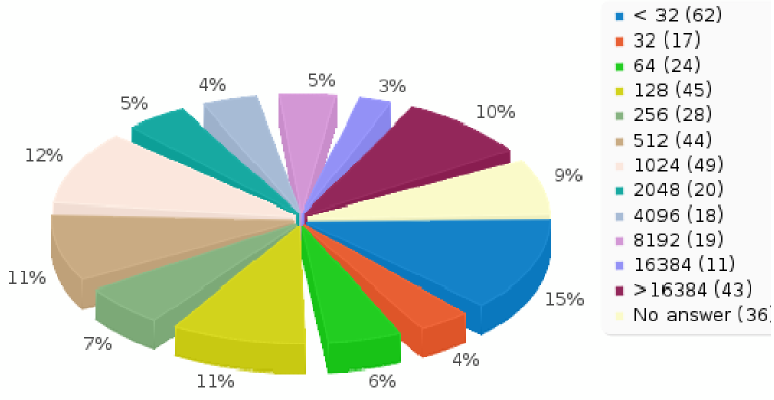
Please write your answer here:

- *-- the group has used MPI fortran90 codes for production and post-processing on: Cray T3E (Cineca, Bologna), Ibm Sp4/Sp6 (Cineca, Bologna, BSC Barcelona Mare Nostrum) and various Linux clusters (e.g. Bcx at Cineca, Bologna; Matrix at Caspur, Rome). -- actually uses SP6 CINECA -- we are thinking to pass to the BlueGene architecture*
- *(desktop) grids*
- *Advanced experience with vector computers*
- *Advanced for Service Grid middleware*
- *advanced on old-fashioned vector systems (from NEC)*
- *ARM, experienced*
- *Cloud Computing*
- *clssical vector systems*
- *Course at the HLRS Stuttgart*
- *CUDA, HMPP, PGI Compiler*
- *Developing parallel codes to run linear algebra and image processing routines.*
- *Digital Signal Processors*
- *experience to work with shared memory multi processor + parallelization through vector facility (e.g. Convex) in the olden days. used to work with shared memory / tightly coupled multi-processor (making use of coherent address space, fine granulyrity, so message passing not applicable).*
- *Fortran90 and IDL coding on standard desktop machines.*
- *gLite Grids*
- *Good knowledge of CUDA/OpenCL, MPI and shared-memory programming using threads. Some knowledge of OpenMP. Limited expererience on mixed environments.*
- *GPGPU Programming using CUDA, HMPP & PGI*
- *have implemented code which utilize openmp and MPI.*
- *Highly experienced in developing/implementing serial code, but not I do not consider myself an expert in optimising performance.*
- *I also have experiences with Vector architectures like the NEC SX.*
- *I also have long experiences in Vector Computing Architectures like the NEC SX systems.*
- *I and my group have developing large scale algorithms for the past 10+ years, and e.g. are among the first to do complex multiphysics simulations on heterogeneosuclausters using both CPU and CPUs is a massively parallel application.*
- *I develop code for single user machines and use distribution frameworks to distribute the execution of replications over multiple machines. My HPC usage is also of similar nature, i.e., run a large number of replications of the same simulation to explore parameter space or to improve result accuracy. HPC serves mainly the purpose of enabling automatic parallel replication of the program, and not of threads within the program.*
- *I have advanced proficiency in developing codes for widely distributed architectures.*
- *I have done some basic experimentation with implementing some of the very computationally intensive functions in our group's code on GPGPUs using CUDA.*
- *I wrote MPI fortran90 codes for production and post-processing. I used them on Cray T3E (Cineca, Bologna. 10000 cpu-hours), Ibm Sp4/Sp6 (Cineca, Bologna, 300000 cpu-hours), MareNostrum (BSC, Barcelona, 5000 cpu-hours) various Linux clusters (e.g. Bcx at Cineca, Bologna; Matrix at Caspur, Rome, 80000 hours). I wrote also few openmp post-processing codes.*
- *Intel SCC: Advanced*
- *large experience with vector codes.*
- *Linux workstations/cluster*
- *Low-level optimizations, performance analysis and modelling*
- *Moderate proficiency with GPGPU programming (CUDA).*
- *Multi FPGA Cores Invasic Computing*
- *my research interest lies in computational sciences with some emphasis on algorithme design for the solution of large sparse linear systems with hierachical numerical techniques capable to exploit the structure hierachy of parallel platforms.*
- *NA*
- *none*
- *None*
- *Old vector supercomputers like Cray X1E (vectorization& parallelization).*
- *Parallel vector machines.*
- *Sun solaris, basic knowledge*
- *until mid-2009: NEC SX vector architecture*
- *We have worked a bit on Many Integrated Core architectures. (Basic)*
- *x86 ;-)*

10. What is the maximum number of CPU cores you have managed to scale a code to? For third-party scientific application users, what is the largest number of CPU cores you have used? Please specify the application in question in the comment box.

Answer	Count	Percentage
< 32	62	14.90%
32	17	4.09%
64	24	5.77%
128	45	10.82%
256	28	6.73%
512	44	10.58%
1024	49	11.78%
2048	20	4.81%
4096	18	4.33%
8192	19	4.57%
16384	11	2.64%
>16384	43	10.34%
No answer	36	8.65%

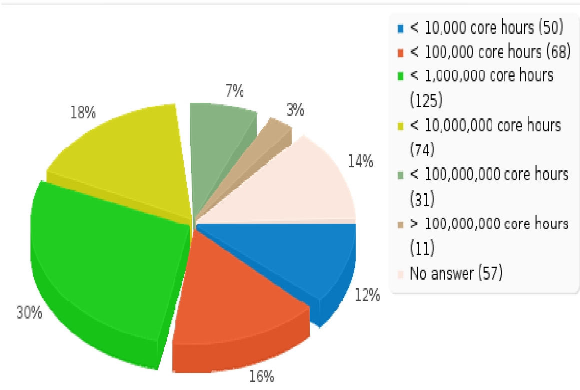
Chart 6.2.1-11



11. Please make estimation on how many accumulated core hours you like to use in 2012?

Answer	Count	Percentage
< 10,000 core hours (e.g., = 16 cores * 8 h * 80 runs)	50	12.02%
< 100,000 core hours (e.g., = 64 cores * 20 h * 80 runs)	68	16.35%
< 1,000,000 core hours (e.g., = 512 cores *24 h * 80 runs)	125	30.05%
< 10,000,000 core hours (e.g., = 4096 cores * 24 h * 100 runs)	74	17.79%
< 100,000,000 core hours (e.g., = 16284 cores *24 h * 250 runs, or 1000 cores * 24 h * 50 jobs in parallel * 80 such runs)	31	7.45%
> 100,000,000 core hours	11	2.64%
No answer	57	13.70%

Chart 6.2.1-12

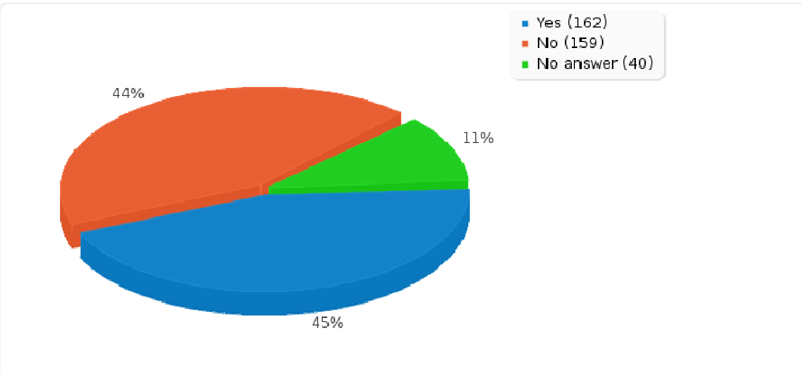


General High Performance Computing Training

12. Do you believe you have received sufficient training to maximise usage and efficiency of the HPC resources available to you?

Chart 6.2.1-13

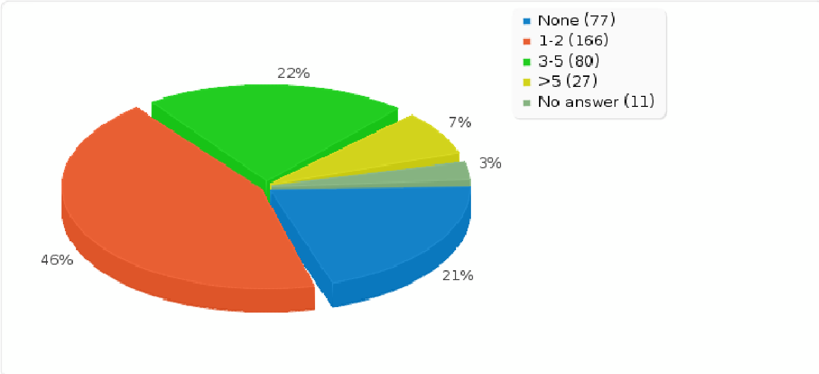
Answer	Count	Percentage
Yes	162	44.88%
No	159	44.04%
No answer	40	11.08%



13. How many HPC training courses have you attended in the past?

Chart 6.2.1-14

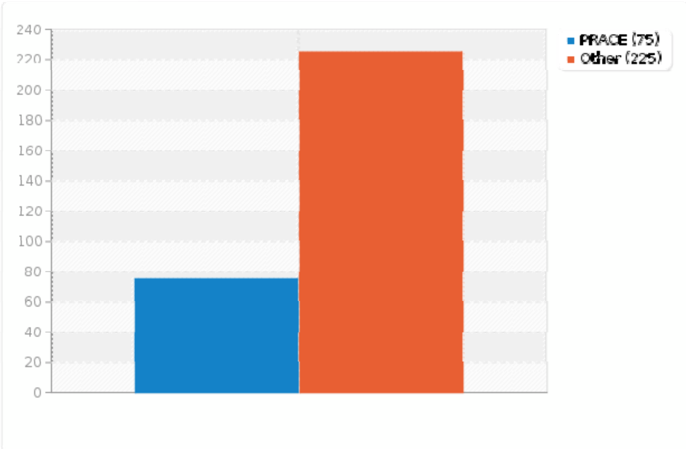
Answer	Count	Percentage
None	77	21.33%
1-2	166	45.98%
3-5	80	22.16%
>5	27	7.48%
No answer	11	3.05%



14. Who has organised these training events?

Chart 6.2.1-15

Answer	Count	Percentage
PRACE	75	18.03%
Other	225	54.09%



Other:

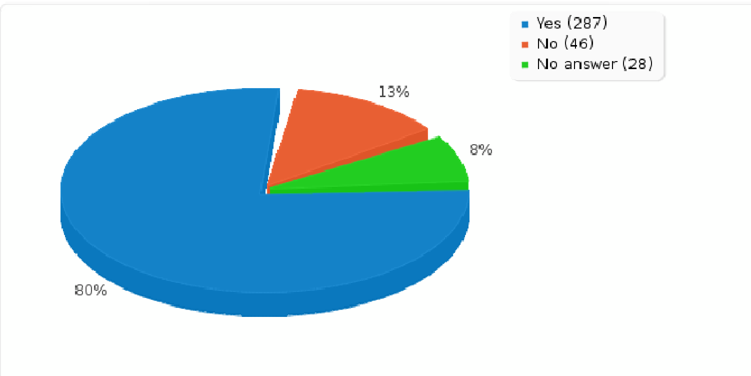
- *Academia*
- *3xBSC*
- *BSC, CESCA*
- *BSC, myself*
- *2xCASPUR*
- *CECAM/ICHEC/EPCC*
- *10xCINECA*
- *CINECA, CASPUR, and others*
- *Cineca, HPC programming*
- *CISM*
- *CISM UCL Belgium*
- *computer vendor*
- *2xComputing Centers*
- *Cray, SGI, IBM*
- *6xCSC*
- *CSCS*
- *CSCS Manno, ETH Zurich*
- *CSCS, HPC-CH*
- *CSIM*
- *DEISA, HLRS*
- *DEISA, Internal training*
- *DEISA/DECI HPC centers (HRZ , LRZ, etc.)*
- *Different international institutions*
- *2xDKRZ*
- *EGI, BSC, SARA*
- *Employer, NAG Ltd, Conferences*
- *3xEPCC*
- *EPCC, BSC, Univ.Malaga*
- *EPCC, JSC*
- *EPCC, LIP*
- *EPCC, NAG, EPSRC*
- *ERASMUS*
- *Euforia*
- *EUGrid, IBM*
- *Forgot*
- *French reserach organisations*
- *FZ Jülich*
- *FZ Jülich, HLRS*
- *FZJ Juelich, HLRS Stuttgart, others*
- *Garchin*
- *German Research School for Simulation Science, Aachen/Jülich*
- *Hellasgrid*
- *High Performance Computing Center (HLRS)*
- *HLRN, HLRS, JSC*
- *37xHLRS*
- *HLRS and internal training*
- *HLRS Stuttgart, LRZ Muenchen*
- *HLRS Stuttgart, LRZ Munich, NIC Juelich*
- *HLRS Stuttgart, RZ RWTH Aachen*
- *HLRS Stuttgart, Universität Karlsruhe*
- *HLRS Stuttgart, ZIH Dresden*
- *HLRS, CIMEC Santa Fe Argentina*
- *HLRS, DLR*
- *HLRS, LRZ, NEC*
- *HLRS, NEC*
- *HLRS, TEXT*
- *HLRS, Uni Heidelberg*
- *HLRS,ICM*
- *7xHP-SEE*
- *HPC Companies*
- *HPC conferences/tutorials*
- *2xHPC Europa*
- *HU Berlin, DESY Zeuthen*
- *IBM, Cray, Univ Heidelberg, HLRS*
- *IBM, ISC events*
- *3xICHEC*
- *ICHEC,CINECA, CASPUR,ITALIAN GRID*

- *ICM*
- *ICM, POWIEW*
- *ICM, University of Warsaw*
- *4xICTP (Trieste, Italy)*
- *Idris*
- *in house*
- *INCITE, SCIDAC*
- *JRC, ECMWF*
- *JSC @ FZ-Jülich*
- *JSC Jülich*
- *JSC, HLRS*
- *JSC, HLRS, DEISA*
- *JSC/NIC*
- *Jülich, RWTH*
- *KU Leuven*
- *LinkSCEEM*
- *Local cluster manager*
- *Local computcenter*
- *Local Computercenter, HLRS, FZ-Juelich*
- *2xLocal HPC centre*
- *LRS*
- *6xLRZ Munich*
- *LRZ, HLRS*
- *LRZ, NCSA*
- *LRZ, RRZE, HLRN*
- *n.a.*
- *National HPC initiative*
- *3xNCSA*
- *NCSA, JSC (Juelich)*
- *NCSA, NERSC (USA)*
- *nordita*
- *NSCA*
- *Palermo University, COMETA consortium*
- *2xPCSS, Poznan, Poland*
- *PDC at KTH*
- *Pennsylvania State University, ACCRE supercomputing center Vanderbilt, USA*
- *Portuguese universities*
- *Prof. Barry Topping*
- *regional centers*
- *RWTH Aachen, FZ Juelich*
- *RWTH Aachen, other Universities, tutorials at conferences*
- *RWTH Aachen, TU Dresden*
- *RWTH, FZ Jülich*
- *RWTH,HLRS*
- *RZ RWTH*
- *SARA & TU Delft*
- *SCC-KIT*
- *School*
- *SCL*
- *SNIC, HPC2N, IBM, Intel*
- *Stuttgart, tutorials at Supercomputing*
- *supercomputer centers*
- *TRACS*
- *Tracs, Nvidia, Universities*
- *Trinity college TCHPC, University of Cambridge*
- *2xTU Dresden*
- *TU Dresden, Uni Stuttgart*
- *TUHH coop with HLRS*
- *UK HPCx and HECToR services*
- *Uni Heidelberg*
- *Univ + CEA*
- *4xUniversity*
- *University Bordeaux 1, ENSEIRB Engineering School*
- *University of Illinois at Urbana-Champaign*
- *University of Louvain (Belgium)*
- *University of Toulouse*
- *various*
- *various national centers*
- *Various national organisations and universities.*
- *vendors, computing centers*
- *VI-HPS / Intel / Bull*
- *VIHPS*
- *VSB-Technical University of Ostrava*

15. Are you aware/kept informed of HPC training courses that are available to you?

Chart 6.2.1-16

Answer	Count	Percentage
Yes	287	79.50%
No	46	12.74%
No answer	28	7.76%



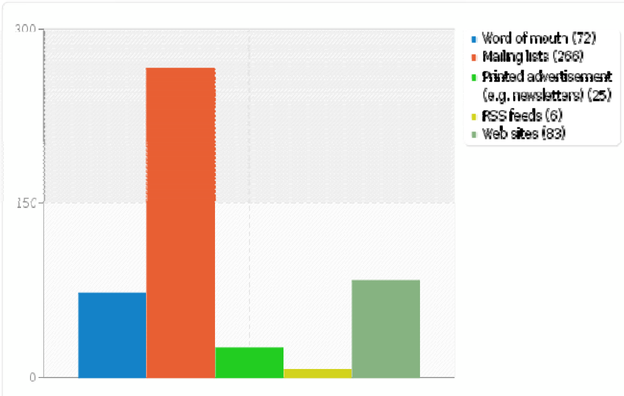
16. If your answer above is "Yes", please specify how?

Only answer this question if the following conditions are met:

° Answer was 'Yes' at question '15 [14]' (Are you aware/kept informed of HPC training courses that are available to you?)

Chart 6.2.1-17

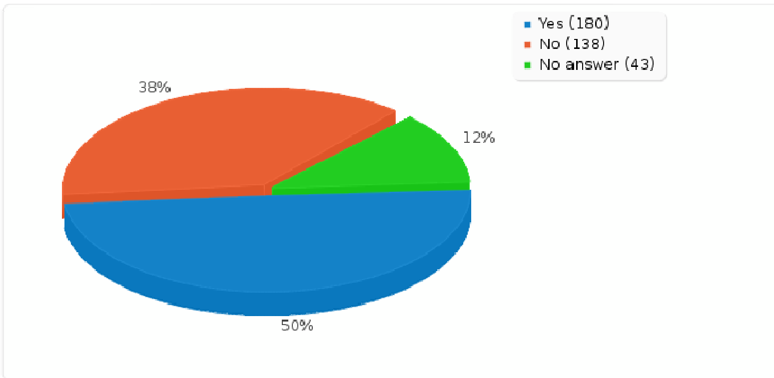
Answer	Count	Percentage
Word of mouth	72	17.31%
Mailing lists	266	63.94%
Printed advertisement (e.g. newsletters)	25	6.01%
RSS feeds	6	1.44%
Web sites	83	19.95%



17. Are you aware/kept informed of training events organised by PRACE?

Chart 6.2.1-18

Answer	Count	Percentage
Yes	180	49.86%
No	138	38.23%
No answer	43	11.91%



hart 6.2.1-19Chart 6.2.1-20

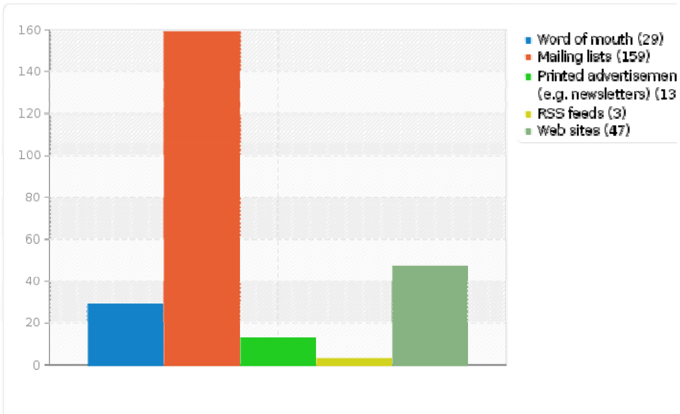
18. If your answer above is "Yes", please specify how?

Only answer this question if the following conditions are met:

° Answer was 'Yes' at question '17 [15]' (Are you aware/kept informed of training events organised by PRACE?)

Chart 6.2.1-21

Answer	Count	Percentage
Word of mouth	29	6.97%
Mailing lists	159	38.22%
Printed advertisement (e.g. newsletters)	13	3.12%
RSS feeds	3	0.72%
Web sites	47	11.30%



19. Please provide your opinions on the following statements.

	disagree		no opinion		agree		no answer	
HPC courses that I'd wish to attend are available when I need them, given I have the time and money to actually attend.	46	12.74%	83	22.99%	185	51.25%	47	13.02%
I am satisfied with the range of HPC topics on which training is available currently.	52	14.40%	74	20.50%	181	50.14%	54	14.96%
I am satisfied with the overall quality of	29	8.03%	95	26.32%	178	49.31%	59	16.34%

HPC training currently available.								
My local HPC centre provides adequate HPC training for most of my needs.	89	24.65%	64	17.73%	142	39.34%	66	18.28%

Chart 6.2.1-22: HPC courses that I'd wish to attend are available when I need them, given I have the time and money to actually attend.

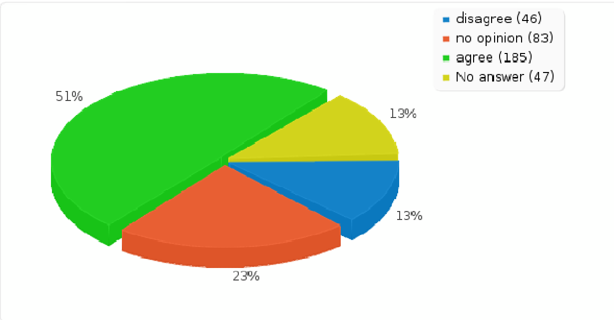


Chart 6.2.1-23: I am satisfied with the range of HPC topics on which training is available currently.

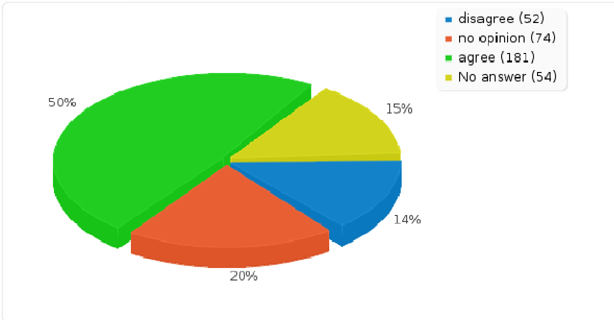


Chart 6.2.1-24: I am satisfied with the overall quality of HPC training currently available.

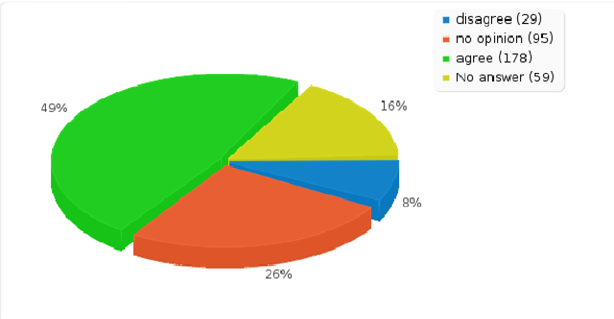
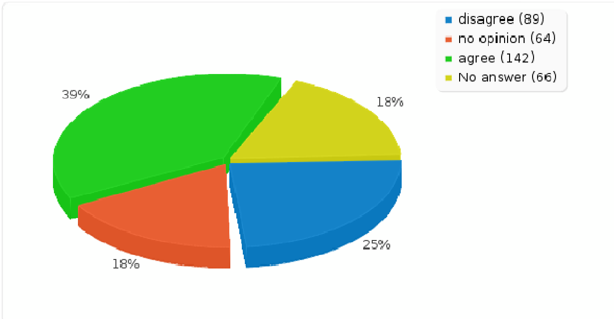


Chart 6.2.1-25: My local HPC centre provides adequate HPC training for most of my needs



20. How would you rate the following modes of training for learning HPC techniques and methodologies:

	not useful		somewhat useful		very useful		No answer	
Face-to-face classes	9	2.49	108	29.92	205	56.79	39	10.80
Combining lectures with hands-on sessions / practical	2	0.55	38	10.53	294	81.44	27	7.48
One-to-one consultation with an expert	6	1.66	63	17.45	252	69.81	40	11.08
Printed material (e.g. books, journals)	25	6.93	189	52.35	114	31.58	33	9.14
Online documentation and tutorials	5	1.39%	142	39.34%	187	51.80%	27	7.48%
Online discussion forum	55	15.24%	170	47.09%	102	28.25%	34	9.42%
Recorded material: videos, podcasts, narrated screencasts	99	27.42%	172	47.65%	48	13.30%	42	11.63%
Live interactive broadcasts (e.g. webinars, multi-cast remote training)	104	28.81%	153	42.38%	44	12.19%	60	16.62%
Computer-based interactive training courses (programme of lectures, material, quizzes and exercises implemented in one environment)	79	21.88%	162	44.88%	67	18.56%	53	14.68%

Chart 6.2.1-26: Face-to-face classes

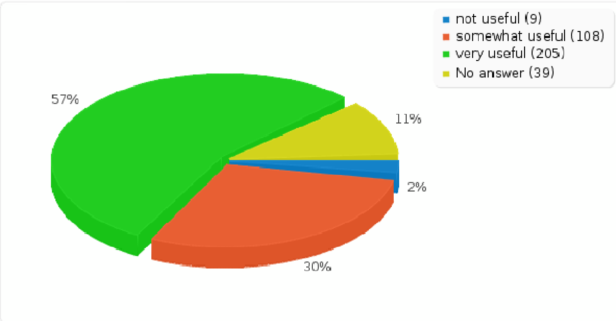


Chart 6.2.1-27: Combining lectures with hands-on sessions / practical

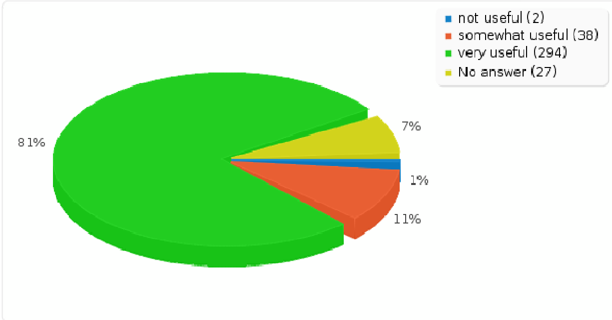


Chart 6.2.1-28: One-to-one consultation with an expert

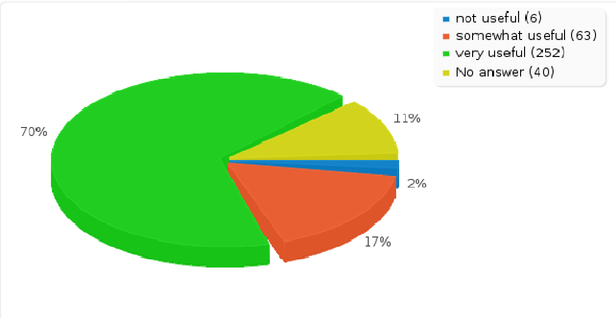


Chart 6.2.1-29: Printed material

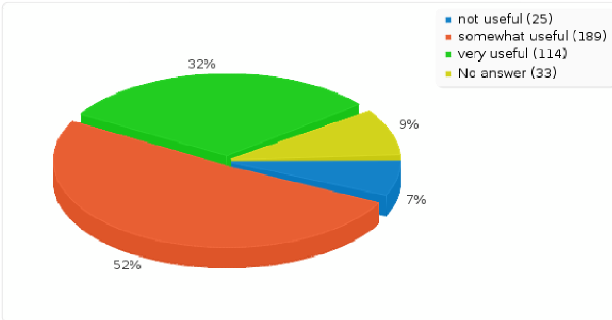


Chart 6.2.1-30 Online documentation and tutorials

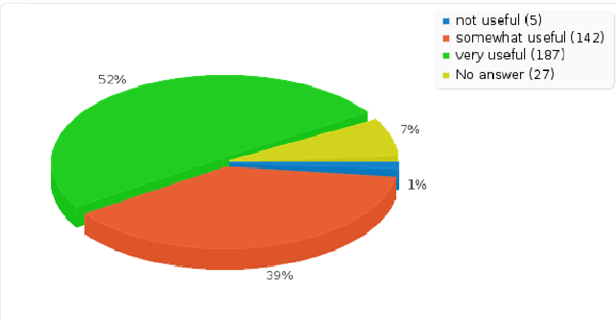


Chart 6.2.1-31 Online discussion forum

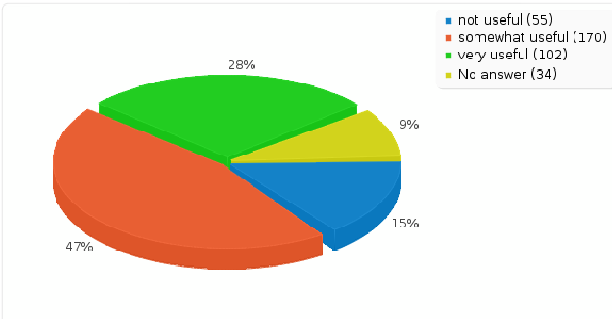


Chart 6.2.1-32 Recorded material: videos, podcasts, narrated screencasts

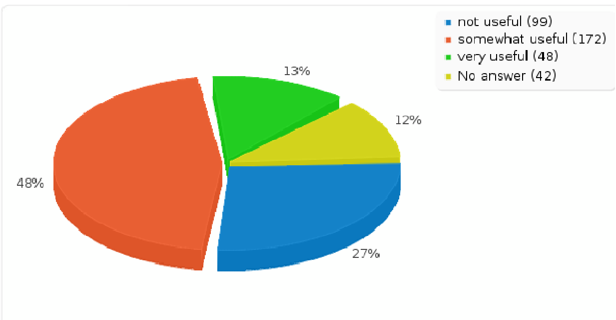


Chart 6.2.1-33 Live interactive broadcasts

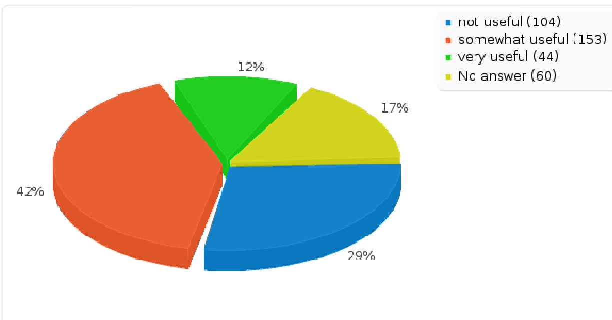
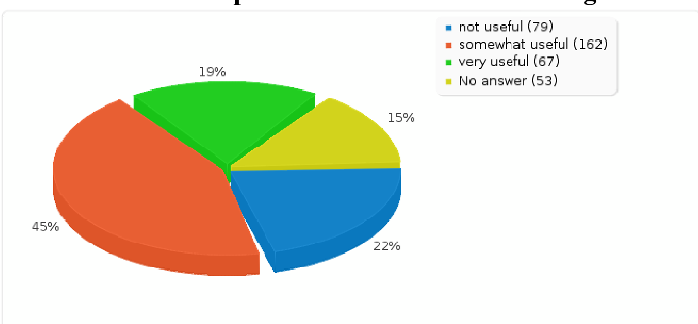


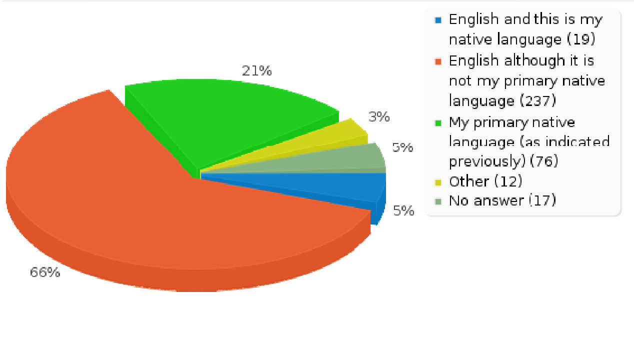
Chart 6.2.1-34: Computer-based interactive training courses



21. Which language do you prefer for oral presentation during HPC courses?

Chart 6.2.1-35

Answer	Count	Percentage
English and this is my native language	19	5.26%
English although it is not my primary native language	237	65.65%
My primary native language (as indicated previously)	76	21.05%
Other	12	3.32%
No answer	17	4.71%

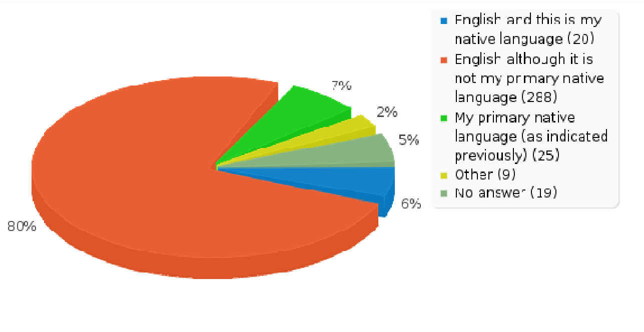


Other: English or German; English or German; English or Native; English or native language - I don't care; English or native language (no preference); English or primary native language - both ok; German; German or English; German or English; I don't really care as long as I understand what's being said. I like when all the material (slides and written) is in the same language though; in different English/native language; Prefer French, but can use English

22. Which language do you prefer for HPC course slides?

Chart 6.2.1-36

Answer	Count	Percentage
English and this is my native language	20	5.54%
English although it is not my primary native language	288	79.78%
My primary native language (as indicated previously)	25	6.93%
Other	9	2.49%
No answer	19	5.26%

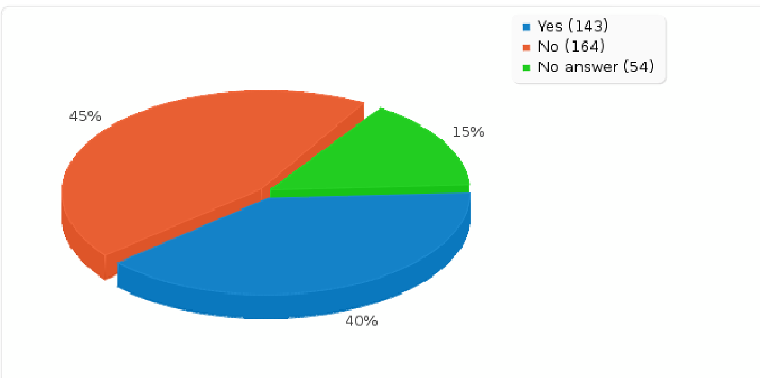


Other: English or primary native language - both ok; English & primary native language are both all right; English or native, I don't mind; English or native language - I don't care; in different English/native language; Same !; German or English; English or German;

23. Have there been cases where you were not able to attend HPC training events that you had wished to participate?

Chart 6.2.1-37

Answer	Count	Percentage
Yes	143	39.61%
No	164	45.43%
No answer	54	14.96%

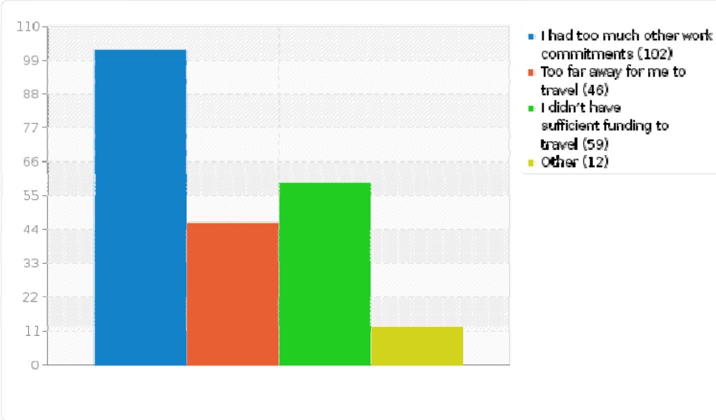


24. If the above answer is "Yes", please specify ALL past reasons

Only answer this question if the following conditions are met:
° Answer was 'Yes' at question '23 [19]' (Have there been cases where you were not able to attend HPC training events that you had wished to participate?)

Chart 6.2.1-38

Answer	Count	Percentage
I had too much other work commitments	102	24.52%
Too far away for me to travel	46	11.06%
I didn't have sufficient funding to travel	59	14.18%
Other	12	2.88%



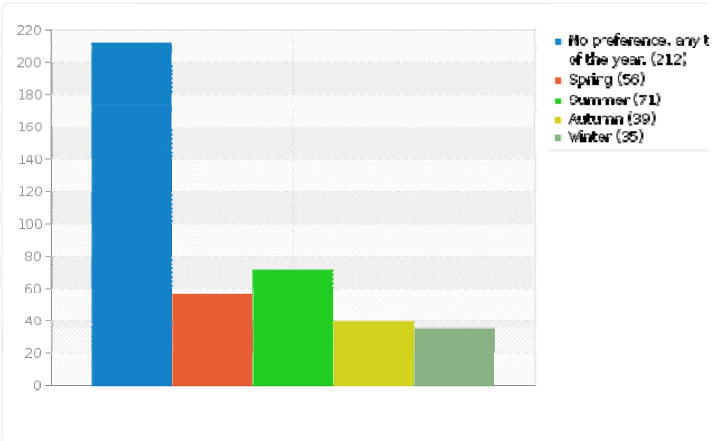
Other:

- *booked up.*
- *course was full*
- *Course was too expensive*
- *due to disability*
- *I didn't get the VISA in the right time to attend the event*
- *I didn't understand the language for presentation*
- *I was in a conference when the local HPC offered a HPC course that I wanted to attend.*
- *Interested in only one topic of the schedule and was afraid that it would just recycle things I already know without going deeper (even though the title would be "Advanced")*
- *It was announced on a very short notice.*
- *Language Barrier*
- *Language: the course was given in German*
- *lectures/slides/course language was not English*

25. What is your preferred time of the year for attending training events?

Chart 6.2.1-39

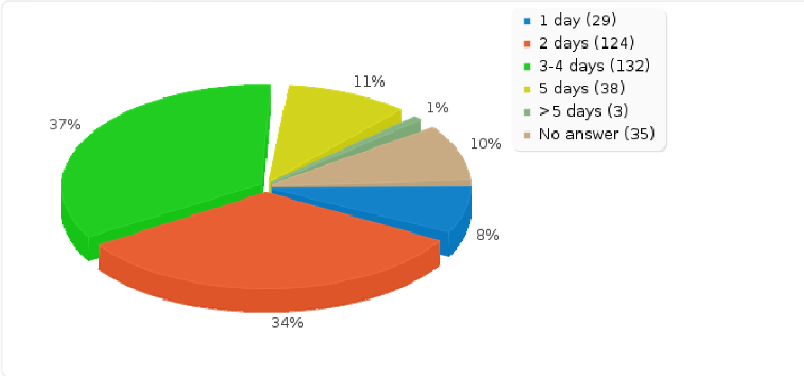
Answer	Count	Percentage
No preference, any time of the year	212	50.96%
Spring	56	13.46%
Summer	71	17.07%
Autumn	39	9.38%
Winter	35	8.41%



26. What is your preferred duration for face-to-face training courses?

Answer	Count	Percentage
1 day (1)	29	8.03%
2 days (2)	124	34.35%
3-4 days (3)	132	36.57%
5 days (4)	38	10.53%
>5 days (5)	3	0.83%
No answer	35	9.70%

Chart 6.2.1-40



27. Were you formally taught the following subjects as part of your primary degree in university? (e.g. received ECTS credits for formal courses on the following)

	Yes		No		No answer	
Sequential programming (in C, Fortran, etc)	256	70.91%	89	24.65%	16	4.43%
Parallel programming (OpenMP, MPI, etc)	70	19.39%	271	75.07%	20	5.54%

Chart 6.2.1-41: Sequential programming (C, Fortran…)

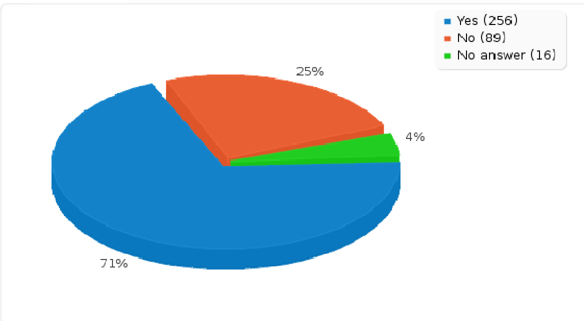
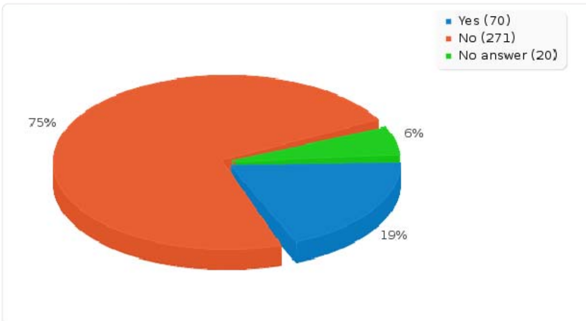


Chart 6.2.1-42: Parallel programming (OpenMP, MPI, etc)



Specific Competencies & Applications

For each HPC-related competency outlined below, please specify your knowledge of the subject in question, whether you use it for your research, and whether you see there is a training demand.

Parallel Programming paradigms

28. How well do you know this topic?

	Never		Occasionally		Frequently or extensively		No answer	
OpenMP	43	12.50%	206	59.88%	85	24.71%	10	2.91%
Basic MPI (point-to-point, collective communication)	24	6.98%	112	32.56%	196	56.98%	12	3.49%
Advanced MPI (MPI-I/O, one-sided communication...)	89	25.87%	165	47.97%	72	20.93%	18	5.23%
Mixed-mode (hybrid) OpenMP-MPI	128	37.21%	155	45.06%	46	13.37%	15	4.36%
PGAS languages (CAF, UPC)	275	79.94%	45	13.08%	10	2.91%	14	4.07%
Next-gen languages (Chapel, X10, Fortress)	307	89.24%	16	4.65%	4	1.16%	17	4.94%
GPU computing (OpenCL, CUDA)	182	52.91%	122	35.47%	28	8.14%	12	3.49%
PVM	280	81.40%	35	10.17%	11	3.20%	18	5.23%
HPF	274	79.65%	46	13.37%	4	1.16%	20	5.81%

Chart 6.2.1-43: OpenMP

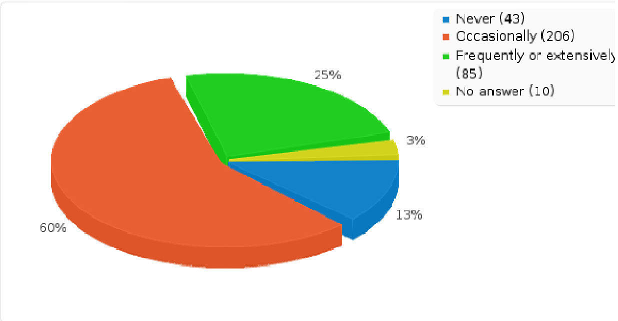


Chart 6.2.1-44: Basic MPI

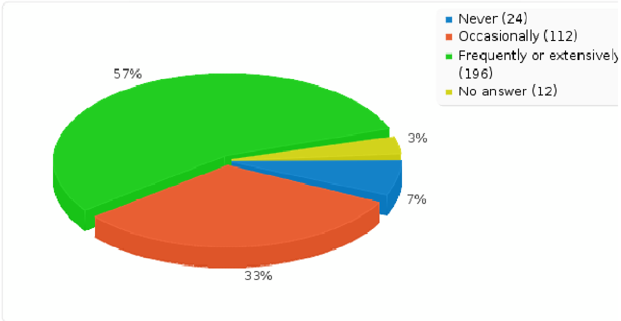


Chart 6.2.1-45: Advanced MPI

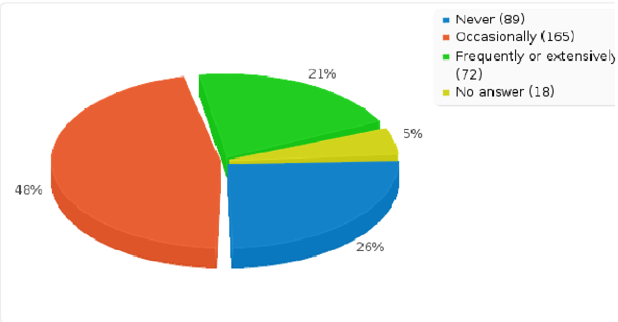


Chart 6.2.1-46: Mixed mode (hybrid) OpenMP - MPI

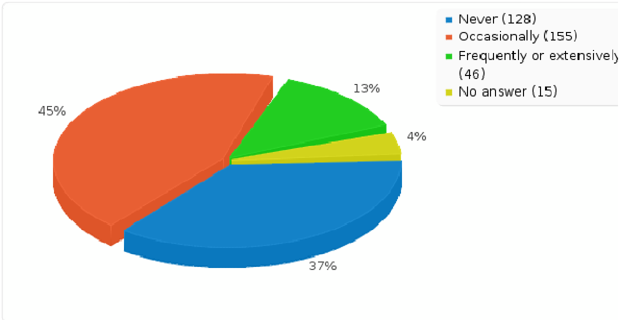


Chart 6.2.1-47: PGAS languages (CAF, UPC)

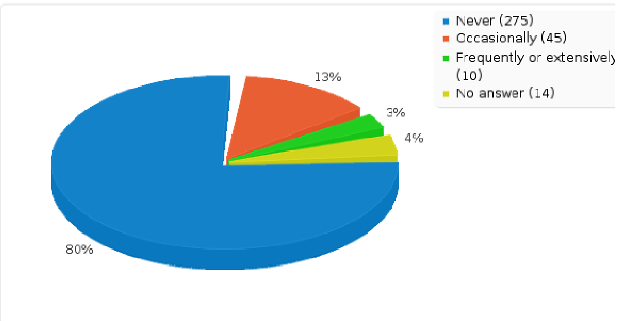


Chart 6.2.1-48: Next-gen languages

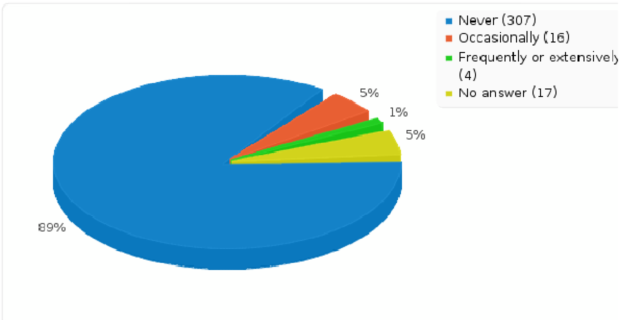


Chart 6.2.1-49: GPU computing

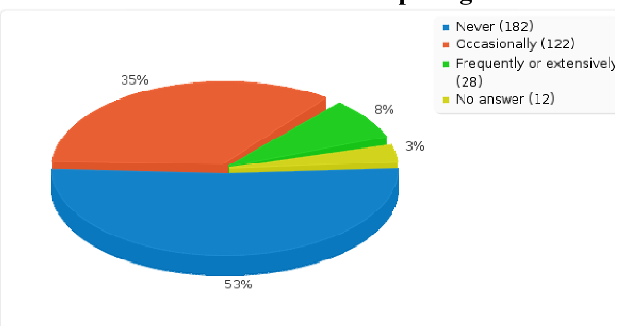


Chart 6.2.1-50: PVM

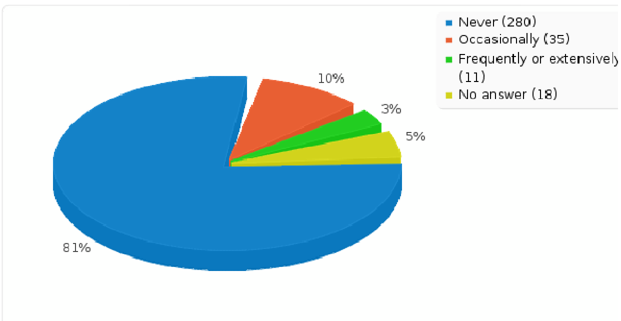
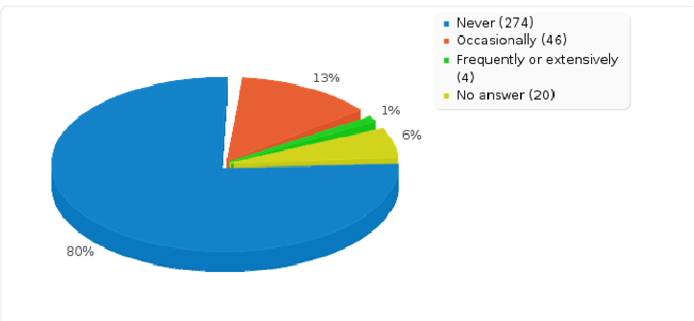


Chart 6.2.1-51: HPF



29. Do you use it?

	Never		Occasionally		Frequently or extensively		No answer	
OpenMP	75	21.80%	167	48.55%	84	24.42%	18	5.23%
Basic MPI (point-to-point, collective communication)	32	9.30%	87	25.29%	211	61.34%	14	4.07%
Advanced MPI (MPI-I/O, one-sided communication...)	112	32.56%	130	37.79%	78	22.67%	24	6.98%
Mixed-mode (hybrid) OpenMP-MPI	154	44.77%	110	31.98%	51	14.83%	29	8.43%
PGAS languages (CAF, UPC)	291	84.59%	24	6.98%	4	1.16%	25	7.27%
Next-gen languages (Chapel, X10, Fortress)	311	90.41%	6	1.74%	3	0.87%	24	6.98%
GPU computing (OpenCL, CUDA)	216	62.79%	85	24.71%	20	5.81%	23	6.69%
PVM	300	87.21%	16	4.65%	2	0.58%	26	7.56%
HPF	296	86.05%	17	4.94%	1	0.29%	30	8.72%

Chart 6.2.1-52: OpenMP

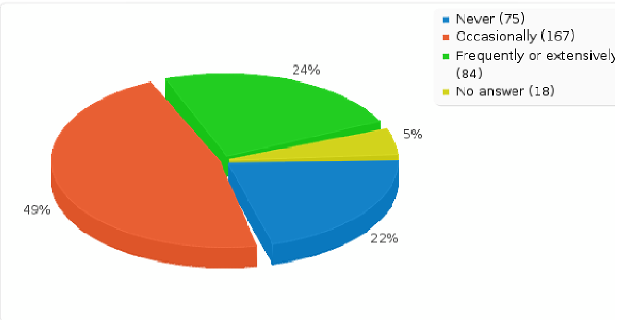


Chart 6.2.1-53: Basic MPI

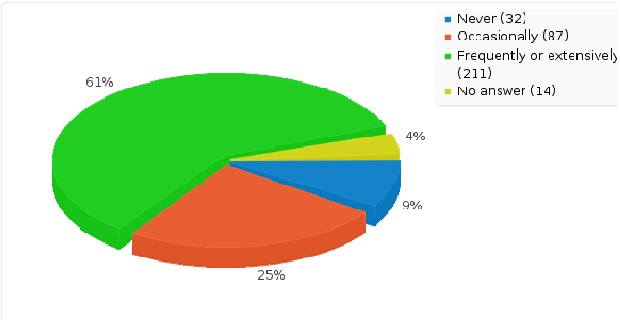


Chart 6.2.1-54: Advanced MPI

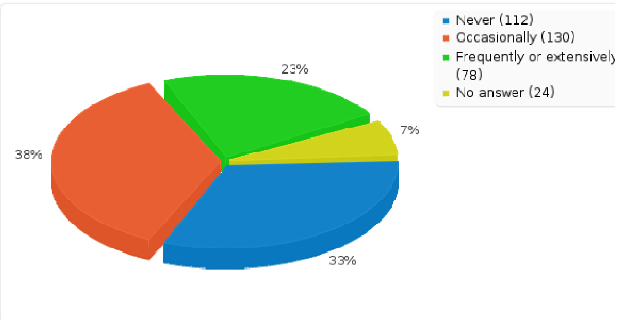


Chart 6.2.1-55: Mixed-mode (hybrid) OpenMP-MPI

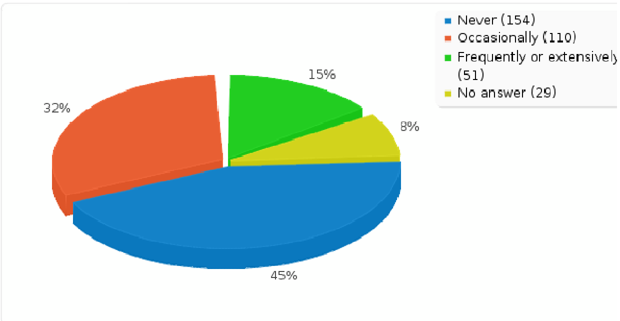


Chart 6.2.1-56: PGAS languages

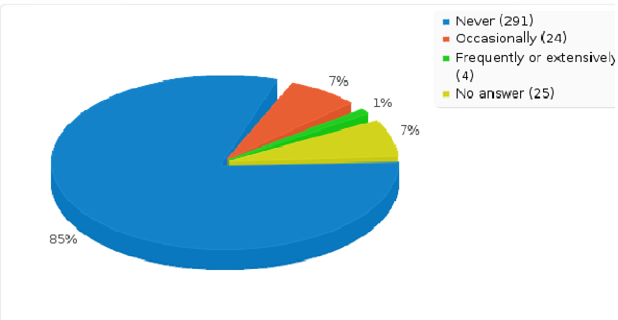


Chart 6.2.1-57: Next-gen languages

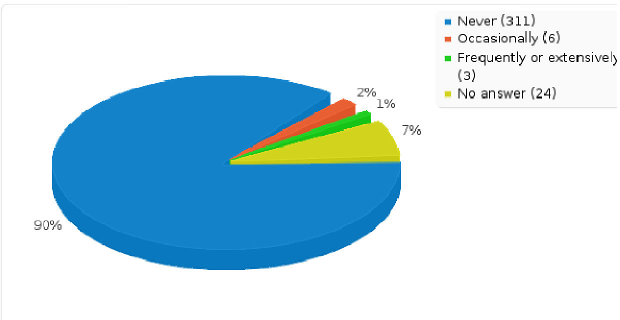


Chart 6.2.1-58: GPU computing

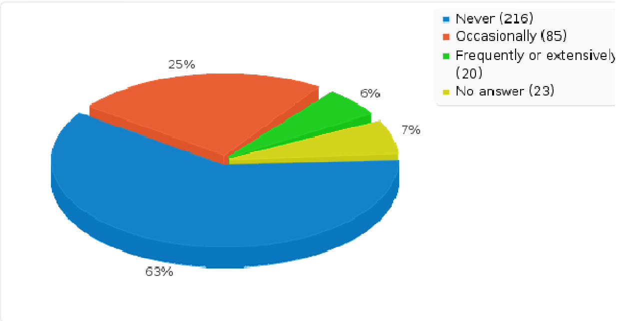


Chart 6.2.1-59: PVM

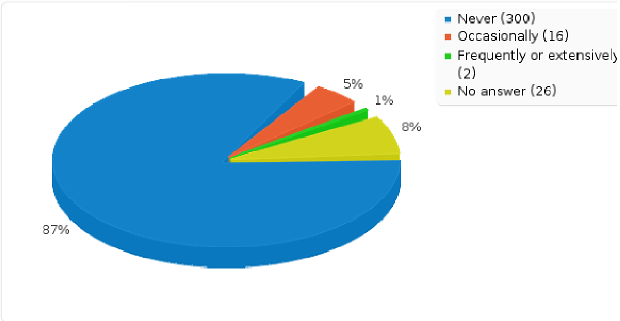
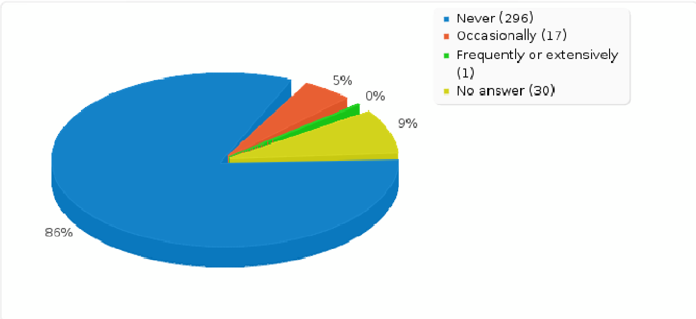


Chart 6.2.1-60: HPF



30. Would you like training on this?

	ASAP		Within the next year		Longer term		Never		No answer	
OpenMP	20	5.81%	88	25.58%	83	24.13%	79	22.97%	74	21.51%
Basic MPI (point-to-point, collective communication)	28	8.14%	68	19.77%	55	15.99%	109	31.69%	84	24.42%
Advanced MPI (MPI-I/O, one-sided communication...)	49	14.24%	93	27.03%	83	24.13%	55	15.99%	64	18.60%
Mixed-mode (hybrid) OpenMP-MPI	33	9.59%	98	28.49%	87	25.29%	52	15.12%	74	21.51%
PGAS languages (CAF, UPC)	21	6.10%	44	12.79%	78	22.67%	89	25.87%	112	32.56%
Next-gen languages (Chapel, X10, Fortress)	17	4.94%	35	10.17%	95	27.62%	87	25.29%	110	31.98%
GPU computing (OpenCL, CUDA)	42	12.21%	104	30.23%	101	29.36%	35	10.17%	62	18.02%
PVM	14	4.07%	14	4.07%	45	13.08%	137	39.83%	134	38.95%
HPF	16	4.65%	22	6.40%	52	15.12%	122	35.47%	132	38.37%

Chart 6.2.1-61: OpenMP

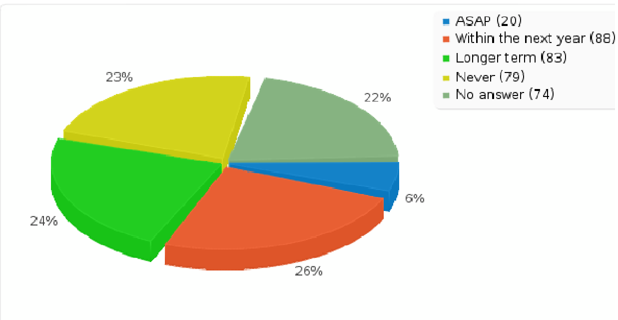


Chart 6.2.1-62: Basic MPI

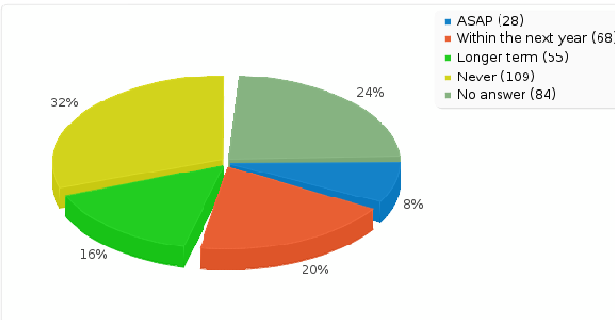


Chart 6.2.1-63: Advanced MPI

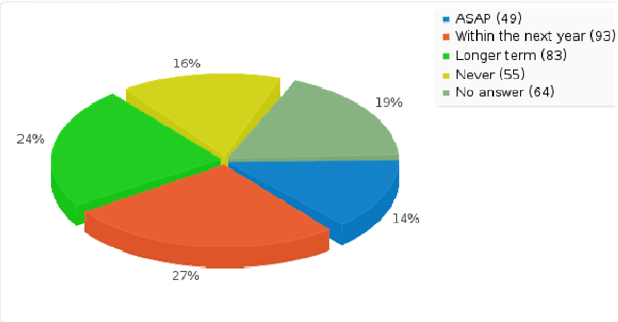


Chart 6.2.1-64: Mixed-mode (hybrid) OpenMP-MPI

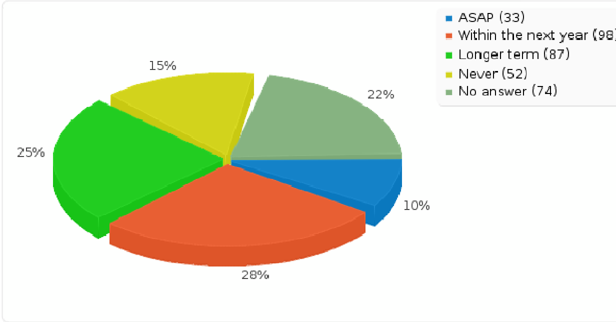


Chart 6.2.1-65: PGAS languages (CAF, UPC)

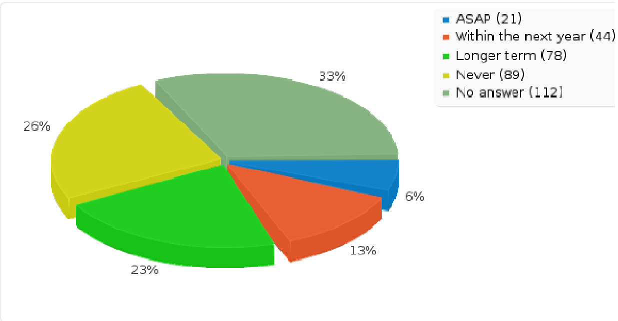


Chart 6.2.1-66: Next-gen languages

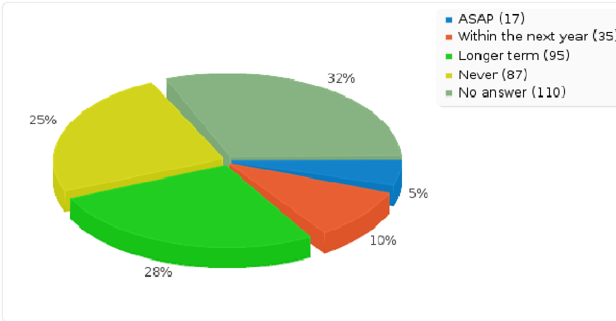


Chart 6.2.1-67: GPU computing

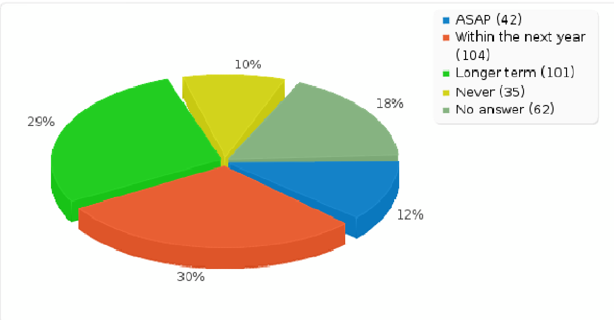


Chart 6.2.1-68: PVM

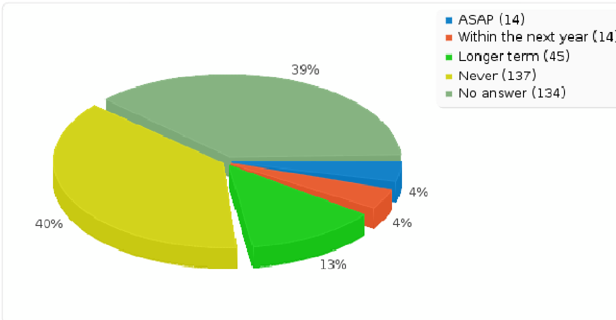
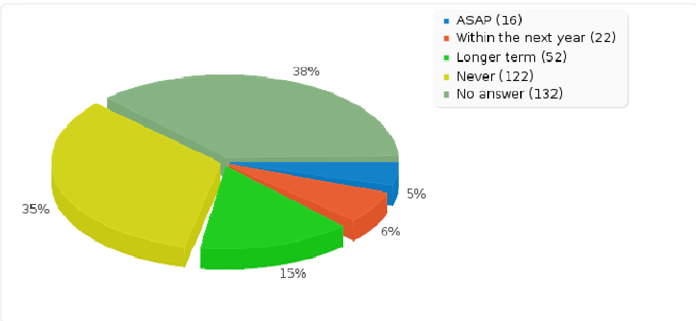


Chart 6.2.1-69: HPF



Programming Languages

31. How well do you know this topic?

	Don't know it		Basic		Proficient		Expert/Guru		No answer	
C / C++	22	6.45%	102	29.91%	142	41.64%	66	19.35%	9	2.64%
Fortran 77	53	15.54%	115	33.72%	110	32.26%	55	16.13%	8	2.35%
Fortran 90, 95...	52	15.25%	105	30.79%	125	36.66%	48	14.08%	11	3.23%
Java	157	46.04%	115	33.72%	40	11.73%	14	4.11%	15	4.40%
Python	131	38.42%	142	41.64%	45	13.20%	11	3.23%	12	3.52%
Matlab / R	113	33.14%	145	42.52%	57	16.72%	12	3.52%	14	4.11%

Chart 6.2.1-70: C / C++

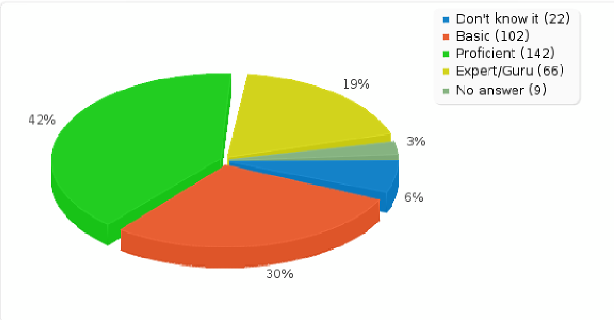


Chart 6.2.1-71: Fortran 77

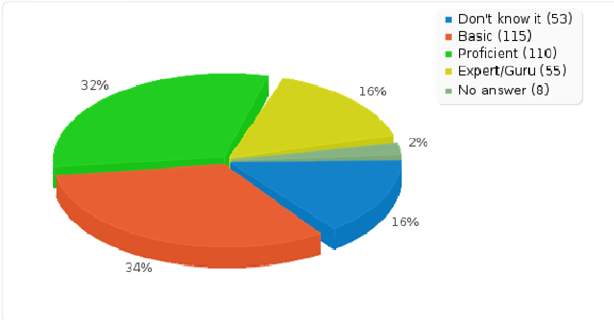


Chart 6.2.1-72: Fortran 90,95

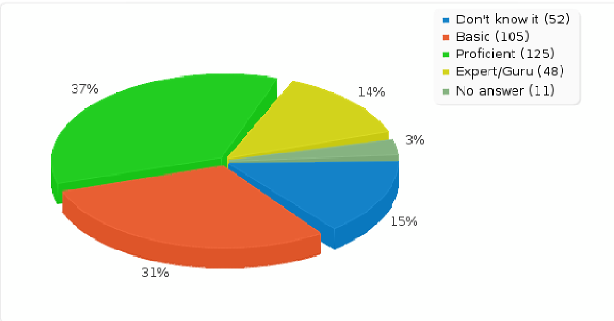


Chart 6.2.1-73: Java

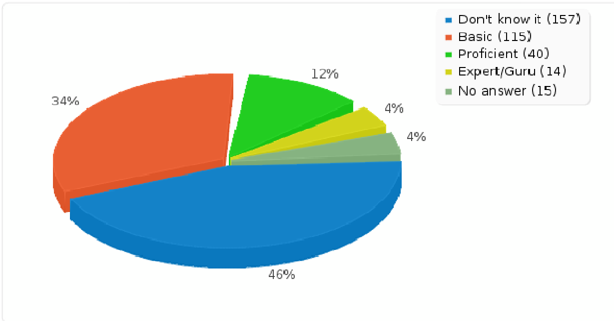


Chart 6.2.1-74: Python

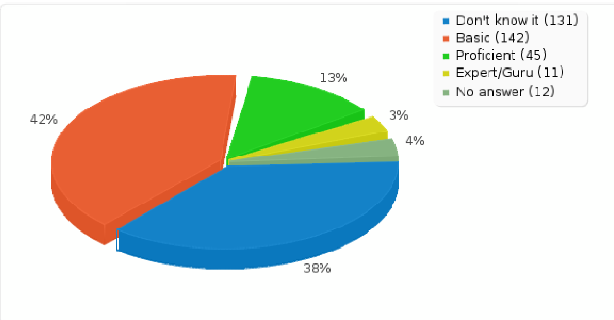
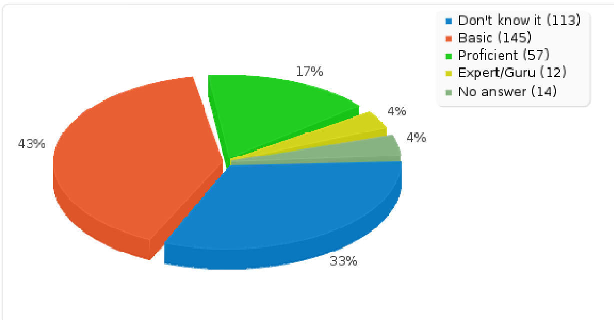


Chart 6.2.1-75: Matlab / R



32. Do you use it?

	Never		Occasionally		Frequently/Extensively		No answer	
C / C++	40	11.73%	107	31.38%	182	53.37%	12	3.52%
Fortran 77	112	32.84%	111	32.55%	105	30.79%	13	3.81%
Fortran 90, 95...	87	25.51%	84	24.63%	156	45.75%	14	4.11%
Java	218	63.93%	85	24.93%	16	4.69%	22	6.45%
Python	149	43.70%	127	37.24%	50	14.66%	15	4.40%
Matlab / R	159	46.63%	114	33.43%	51	14.96%	17	4.99%

Chart 6.2.1-76: C / C++

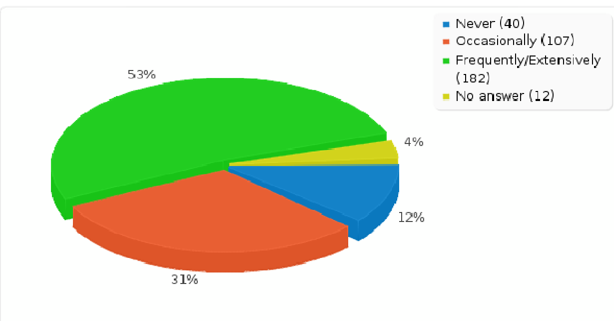


Chart 6.2.1-77: Fortran 77

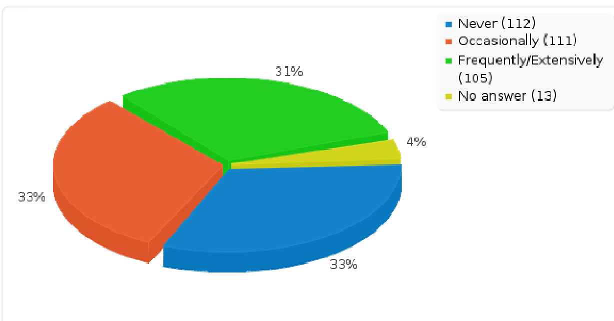


Chart 6.2.1-78: Fortran 90,95

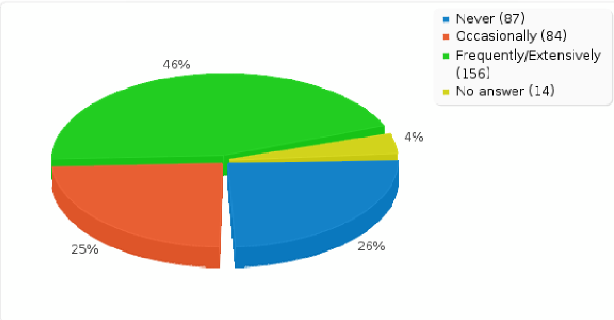


Chart 6.2.1-79: Java

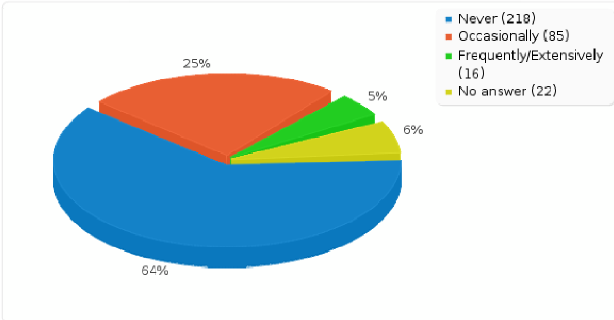


Chart 6.2.1-80: Python

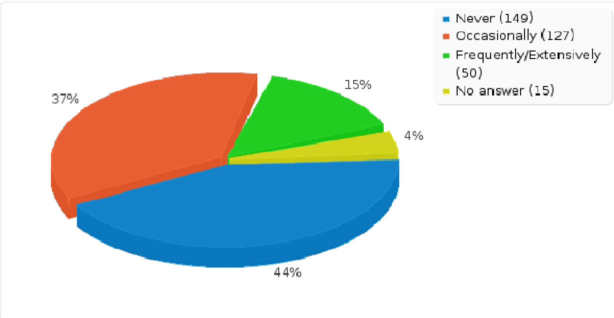
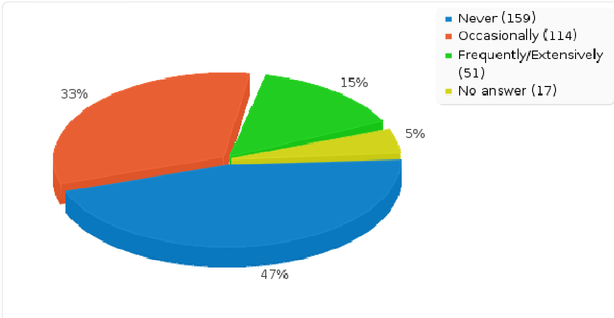


Chart 6.2.1-81: Matlab / R



33. Would you like training on this?

	ASAP		Within next year		Long term		Never		No answer	
C / C++	20	5.87%	66	19.35%	73	21.41%	113	33.14%	69	20.23%
Fortran 77	10	2.93%	18	5.28%	24	7.04%	198	58.06%	91	26.69%
Fortran 90, 95...	23	6.74%	39	11.44%	50	14.66%	145	42.52%	84	24.63%
Java	4	1.17%	28	8.21%	76	22.29%	153	44.87%	80	23.46%
Python	28	8.21%	79	23.17%	76	22.29%	101	29.62%	57	16.72%
Matlab / R	8	2.35%	47	13.78%	74	21.70%	138	40.47%	74	21.70%

Chart 6.2.1-82: C / C++

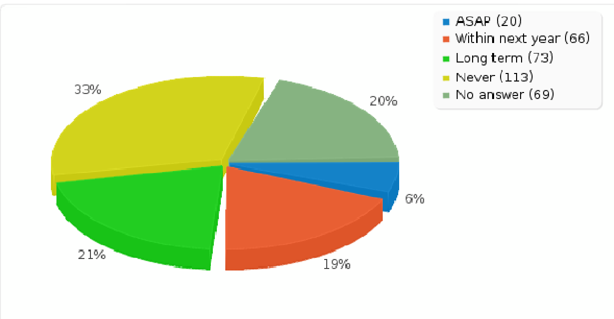


Chart 6.2.1-83: Fortran 77

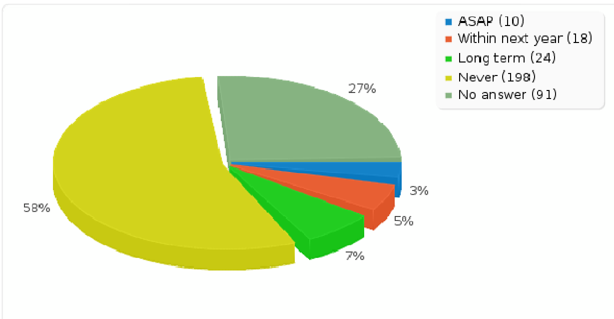


Chart 6.2.1-84: Fortran 90, 95

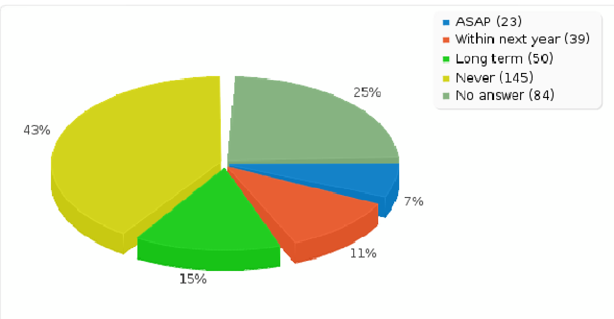


Chart 6.2.1-85: Java

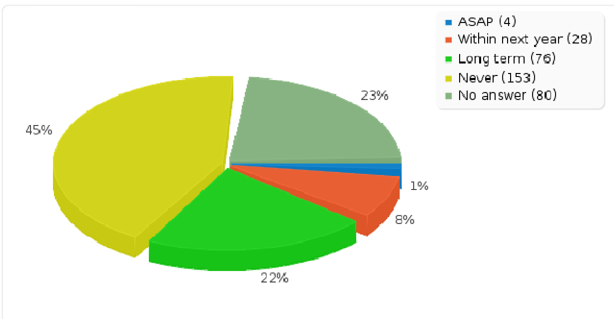


Chart 6.2.1-86: Python

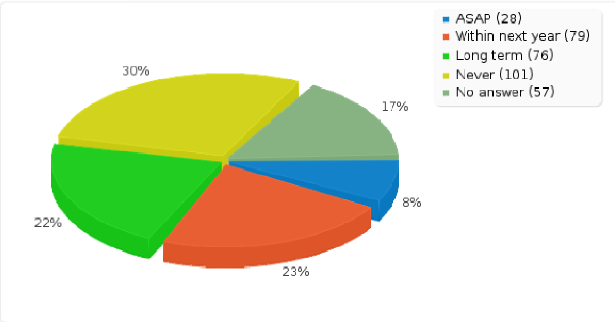
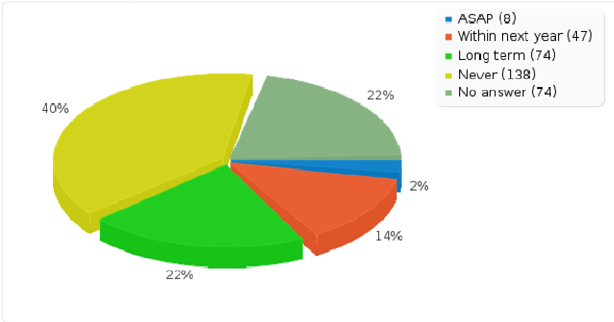


Chart 6.2.1-87: Matlab / R



Libraries and code development tools

34. How well do you know this topic?

	Don't know it		Basic		Proficient		Expert/Guru		No answer	
Basic numerical libraries (LAPACK, EISPACK,...)	52	15.43%	169	50.15%	84	24.93%	13	3.86%	19	5.64%
High-level numerical libraries (PETSc, Trilinos,...)	179	53.12%	107	31.75%	22	6.53%	5	1.48%	24	7.12%
Parallel I/O libraries (HDF5, Parallel NetCDF)	168	49.85%	107	31.75%	32	9.50%	6	1.78%	24	7.12%
General compiler usage and optimisation	20	5.93%	121	35.91%	141	41.84%	36	10.68%	19	5.64%
Architecture-specific optimisation and tuning	66	19.58%	148	43.92%	77	22.85%	24	7.12%	22	6.53%
Debugging tools and techniques	42	12.46%	164	48.66%	97	28.78%	13	3.86%	21	6.23%
Performance analysis/optimisation tools and techniques	63	18.69%	145	43.03%	88	26.11%	17	5.04%	24	7.12%
Software engineering tools and techniques (e.g. code design, maintainability, extensibility)	94	27.89%	136	40.36%	61	18.10%	19	5.64%	27	8.01%

Chart 6.2.1-88: Basic numerical libraries (LAPACK, EISPACK,...)

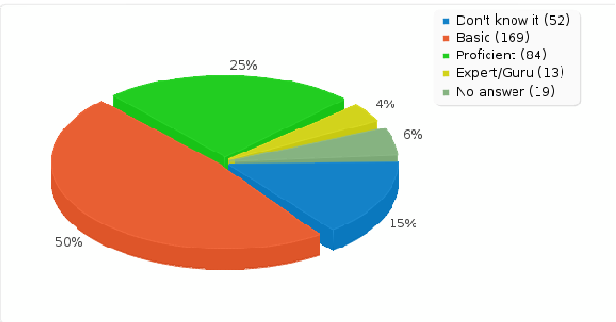


Chart 6.2.1-89: High-level numerical libraries (PETSc, Trilinos,...)

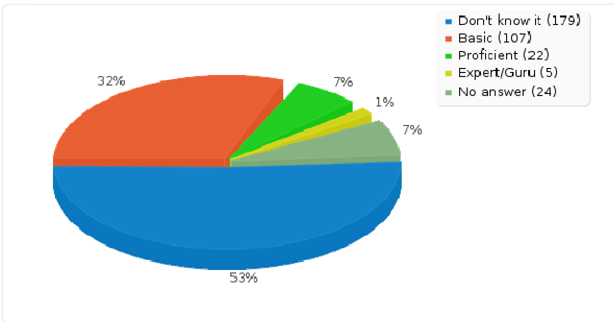


Chart 6.2.1-90: Parallel I/O libraries (HDF5, Parallel NetCDF

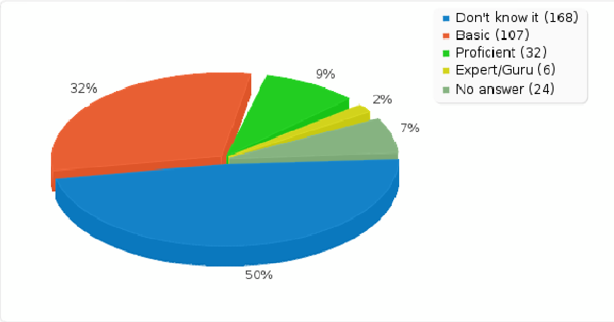


Chart 6.2.1-91: General compiler usage and optimisation

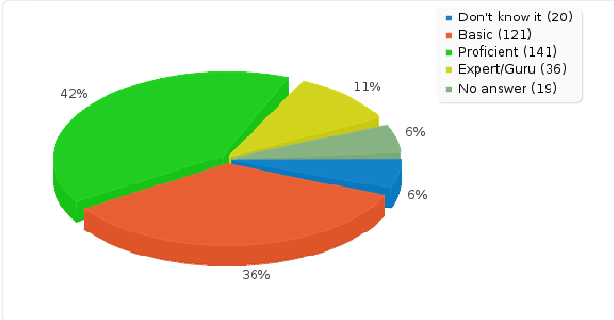


Chart 6.2.1-92: Architecture-specific optimisation and tuning

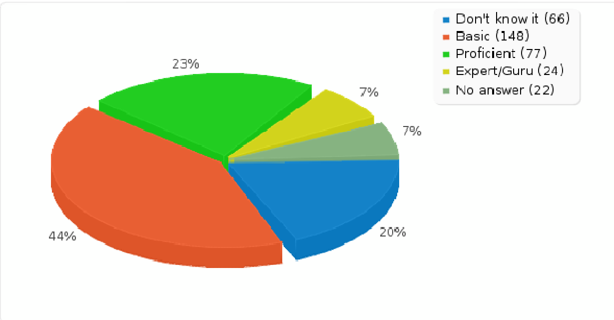


Chart 6.2.1-93: Debugging tools and techniques

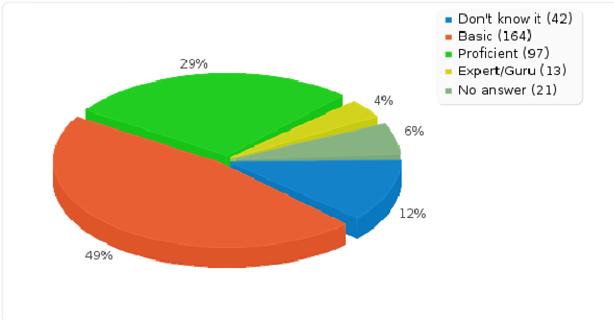


Chart 6.2.1-94: Performance analysis/optimisation tools and techniques

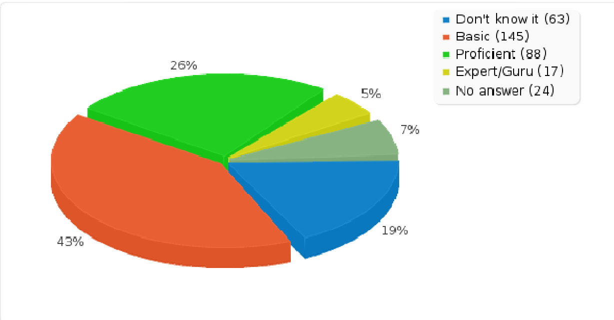
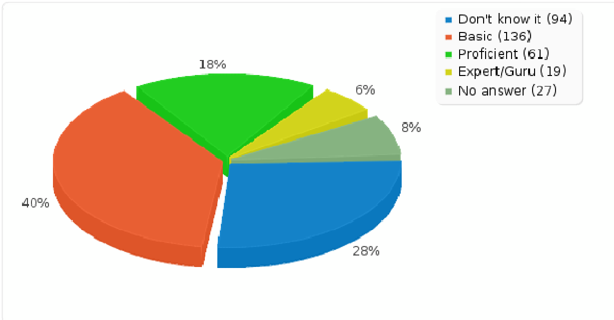


Chart 6.2.1-95: Software engineering tools and techniques (e.g. code design, maintainability, extensibility)



35. Do you use it?

	Never		Occasionally		Frequently or Extensively		No answer	
Basic numerical libraries (LAPACK, EISPACK,...)	80	23.74%	143	42.43%	88	26.11%	26	7.72%
High-level numerical libraries (PETSc, Trilinos,...)	211	62.61%	64	18.99%	27	8.01%	35	10.39%
Parallel I/O libraries (HDF5, Parallel NetCDF)	167	49.55%	89	26.41%	43	12.76%	38	11.28%
General compiler usage and optimisation	29	8.61%	118	35.01%	169	50.15%	21	6.23%
Architecture-specific optimisation and tuning	80	23.74%	136	40.36%	89	26.41%	32	9.50%
Debugging tools and techniques	43	12.76%	161	47.77%	109	32.34%	24	7.12%
Performance analysis/optimisation tools and techniques	77	22.85%	153	45.40%	76	22.55%	31	9.20%
Software engineering tools and techniques (e.g. code design, maintainability, extensibility)	136	40.36%	103	30.56%	62	18.40%	36	10.68%

Chart 6.2.1-96: Basic numerical libraries (LAPACK, EISPACK,...)

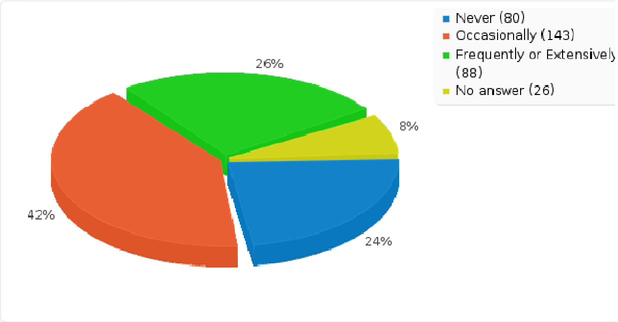


Chart 6.2.1-97: High-level numerical libraries (PETSc, Trilinos,...)

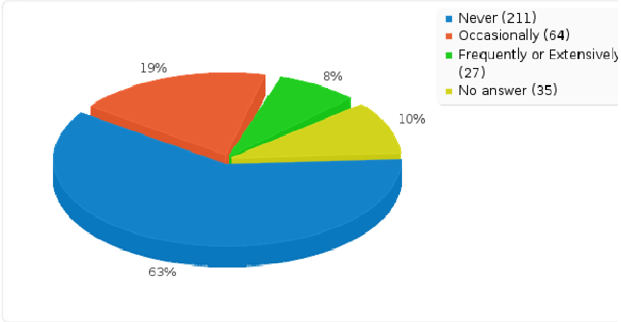


Chart 6.2.1-98: Parallel I/O libraries (HDF5, Parallel NetCDF)

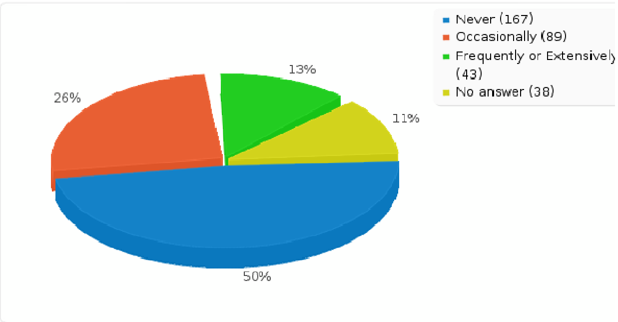


Chart 6.2.1-99: General compiler usage and optimisation

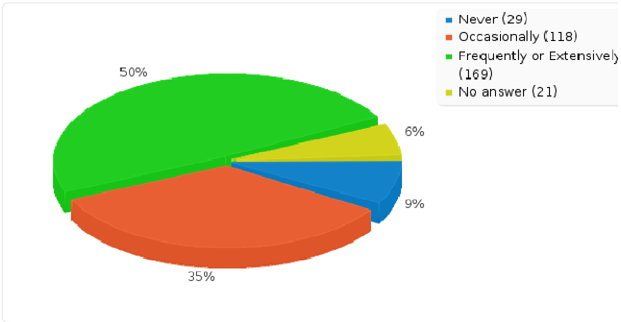


Chart 6.2.1-100: Architecture-specific optimisation and tuning

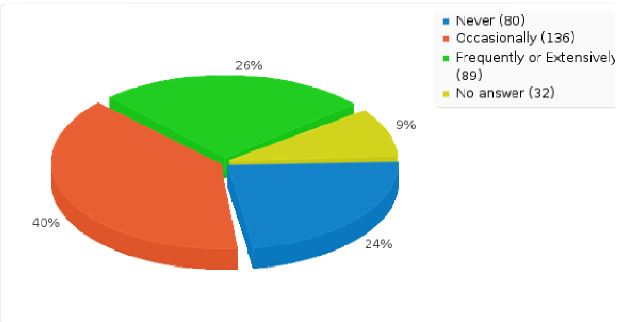


Chart 6.2.1-101: Debugging tools and techniques

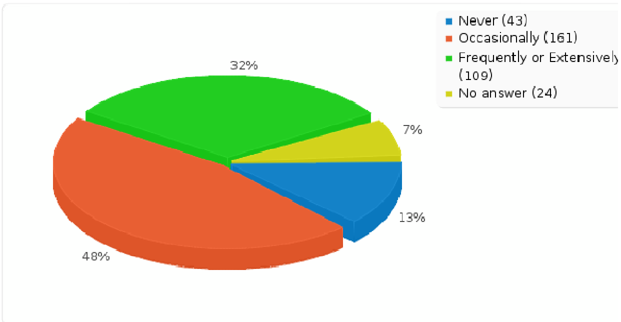


Chart 6.2.1-102: Performance analysis/optimisation tools and techniques

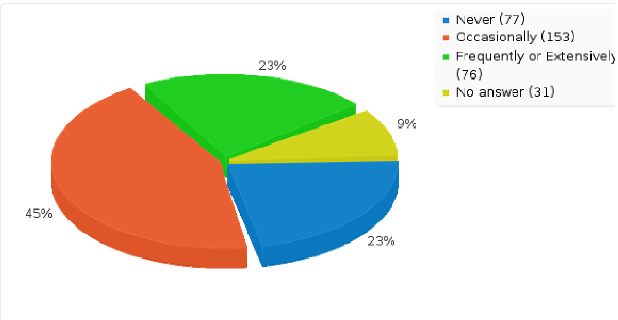
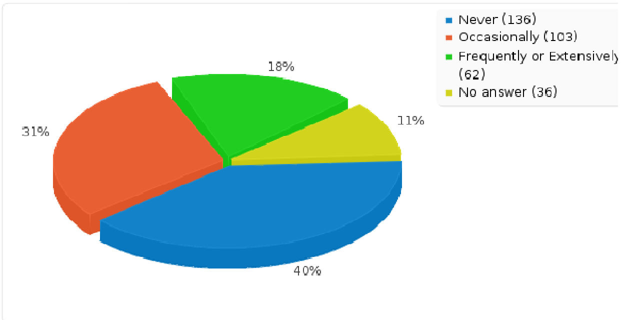


Chart 6.2.1-103: Software engineering tools and techniques (e.g. code design, maintainability, extensibility)



36. Would you like training on this?

	ASAP		Within next year		Long term		Never		No answer	
Basic numerical libraries (LAPACK, EISPACK,...)	14	4.15%	69	20.47%	89	26.41%	94	27.89%	71	21.07%
High-level numerical libraries (PETSc, Trilinos,...)	22	6.53%	58	17.21%	100	29.67%	81	24.04%	76	22.55%

Parallel I/O libraries (HDF5, Parallel NetCDF)	32	9.50%	69	20.47%	99	29.38%	73	21.66%	64	18.99%
General compiler usage and optimisation	39	11.57%	83	24.63%	78	23.15%	68	20.18%	69	20.47%
Architecture-specific optimisation and tuning	35	10.39%	94	27.89%	86	25.52%	54	16.02%	68	20.18%
Debugging tools and techniques	47	13.95%	99	29.38%	83	24.63%	47	13.95%	61	18.10%
Performance analysis/optimisation tools and techniques	53	15.73%	105	31.16%	79	23.44%	43	12.76%	57	16.91%
Software engineering tools and techniques (e.g. code design, maintainability, extensibility)	37	10.98%	88	26.11%	101	29.97%	52	15.43%	59	17.51%

Chart 6.2.1-104: Basic numerical libraries (LAPACK, EISPACK,...)

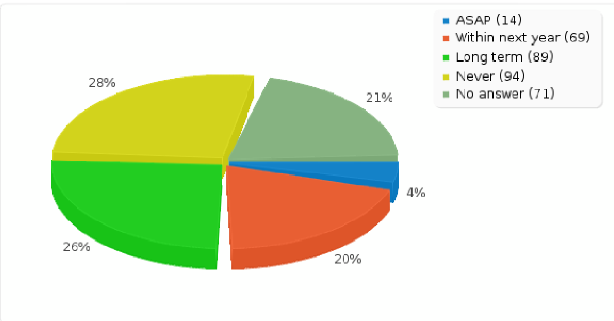


Chart 6.2.1-105: High-level numerical libraries (PETSc, Trilinos,...)

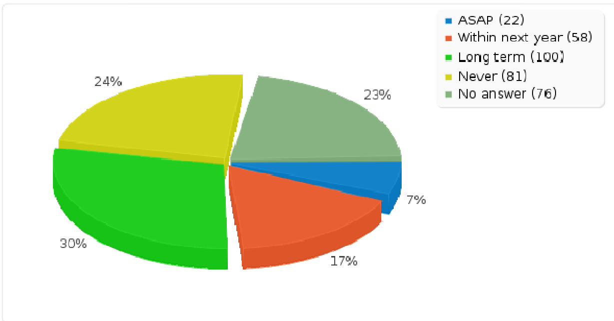


Chart 6.2.1-106: Parallel I/O libraries (HDF5, Parallel NetCDF)

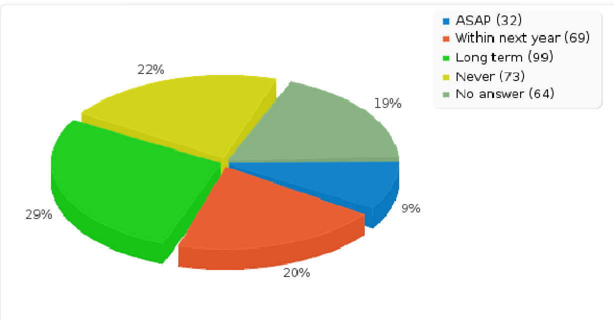


Chart 6.2.1-107: General compiler usage and optimisation

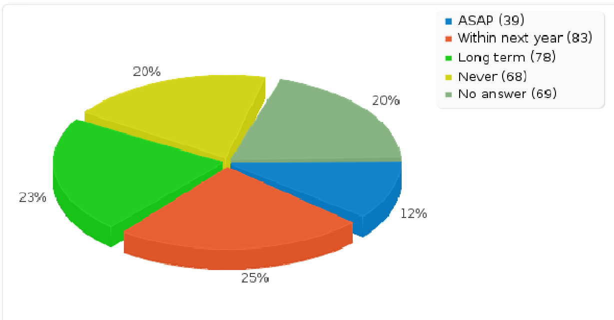


Chart 6.2.1-108: Architecture-specific optimisation and tuning

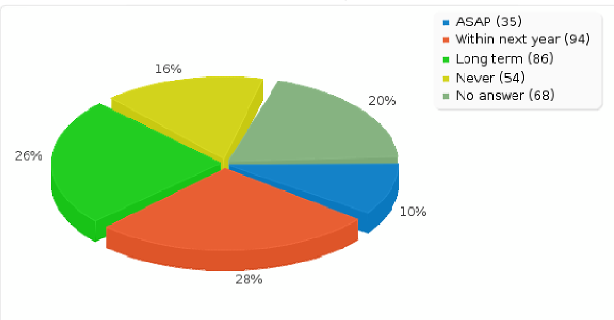


Chart 6.2.1-109: Debugging tools and techniques

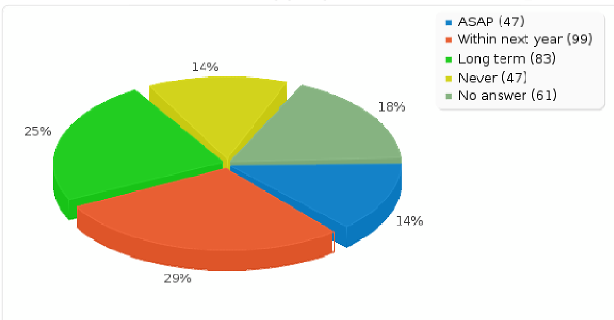


Chart 6.2.1-110: Performance analysis/optimisation tools and techniques

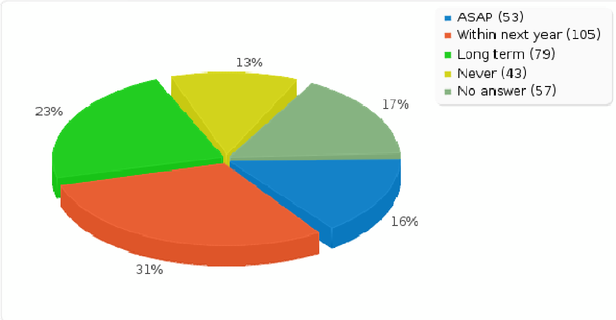
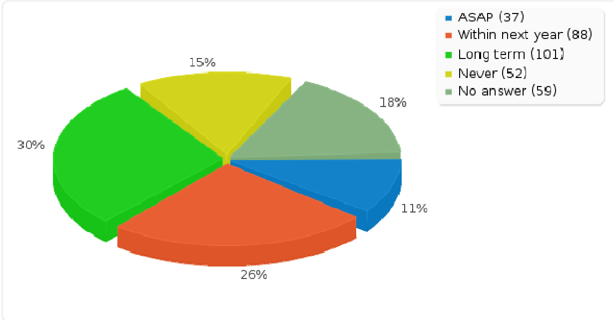


Chart 6.2.1-111: Software engineering tools and techniques (e.g. code design, maintainability, extensibility)



Other topics

37. How well do you know this topic?

	Don't know it		Basic		Proficient		Expert/Guru		No answer	
Basic UNIX skills	1	0.30%	59	17.72%	184	55.26%	83	24.92%	6	1.80%
Scripting (shell, PERL, etc)	14	4.20%	115	34.53%	154	46.25%	42	12.61%	8	2.40%
Batch job systems (job submission and management)	7	2.10%	124	37.24%	164	49.25%	31	9.31%	7	2.10%
Scientific visualisation tools (e.g. VISIT, Paraview)	109	32.73%	117	35.14%	83	24.92%	9	2.70%	15	4.50%
Grid interfaces (e.g. Globus toolkit)	198	59.46%	89	26.73%	19	5.71%	7	2.10%	20	6.01%

Chart 6.2.1-112: Basic UNIX skills

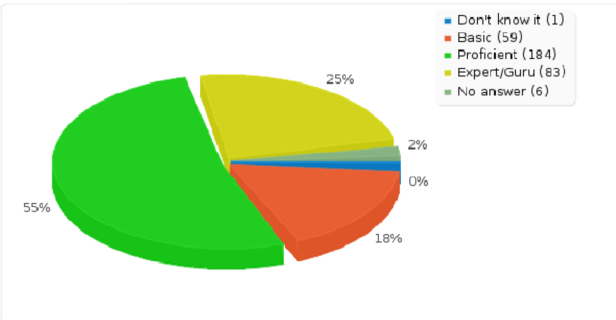


Chart 6.2.1-113: Scripting (shell, PERL, etc)

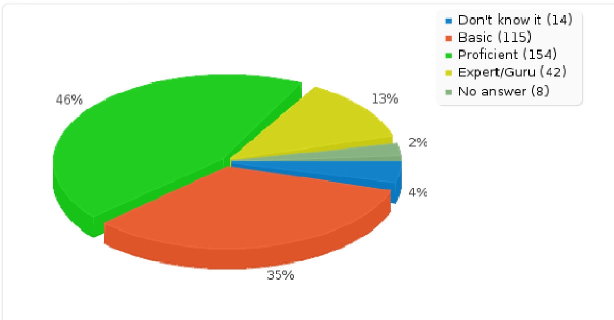


Chart 6.2.1-114: Batch job systems (job submission and management)

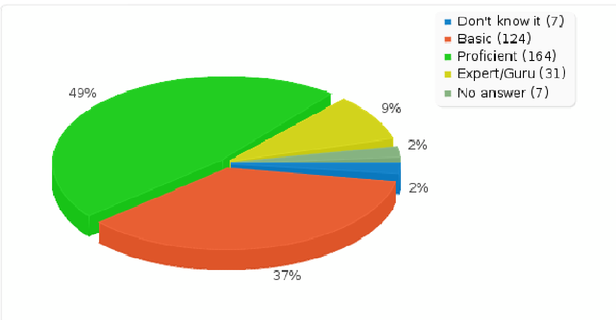


Chart 6.2.1-115: Scientific visualisation tools (e.g. VISIT, Paraview)

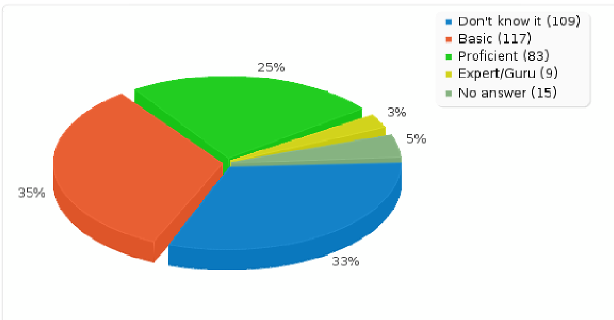
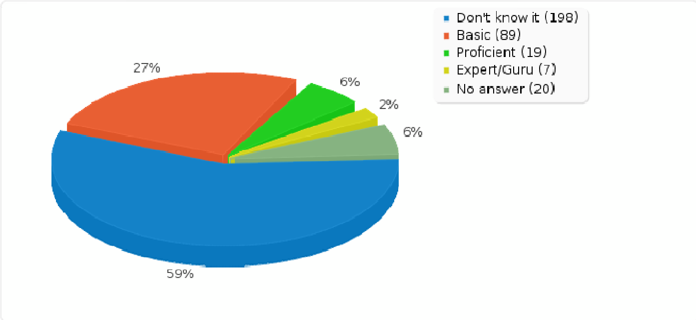


Chart 6.2.1-116: Grid interfaces (e.g. Globus toolkit)



38. Do you use it?

	Never		Occasionally		Frequently or Extensively		No answer	
Basic UNIX skills	2	0.60%	35	10.51%	287	86.19%	9	2.70%
Scripting (shell, PERL, etc)	13	3.90%	110	33.03%	198	59.46%	12	3.60%
Batch job systems (job submission and management)	15	4.50%	97	29.13%	209	62.76%	12	3.60%
Scientific visualisation tools (e.g. VISIT, Paraview)	130	39.04%	103	30.93%	83	24.92%	17	5.11%
Grid interfaces (e.g. Globus toolkit)	244	73.27%	50	15.02%	17	5.11%	22	6.61%

Chart 6.2.1-117: Basic UNIX skills

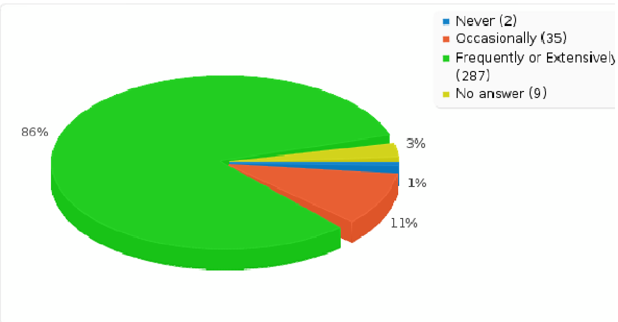


Chart 6.2.1-118: Scripting (shell, PERL, etc)

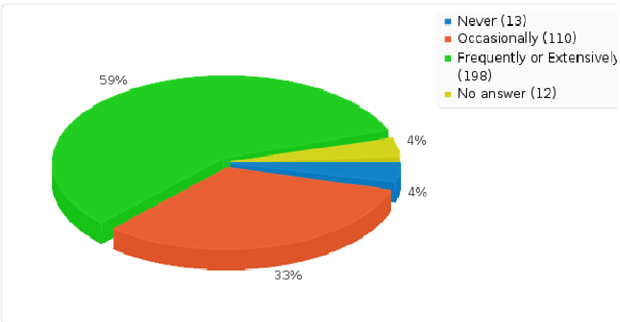


Chart 6.2.1-119: Batch job systems (job submission and management)

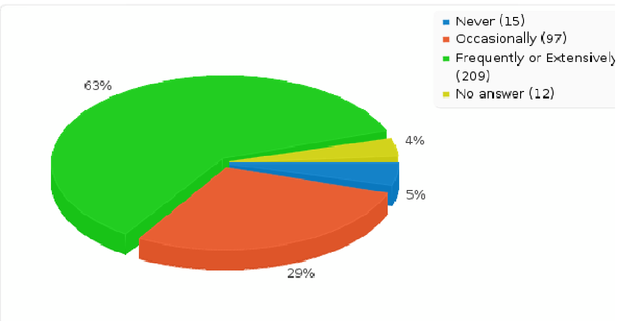


Chart 6.2.1-120: Scientific visualisation tools (e.g. VISIT, Paraview)

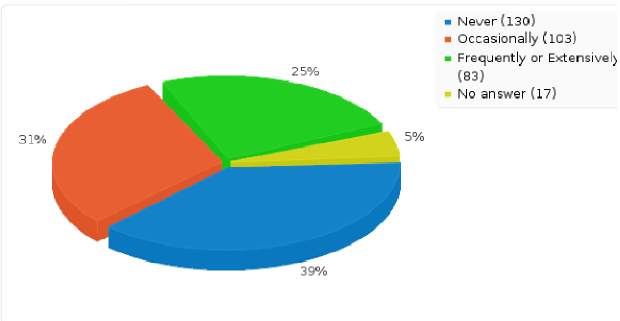
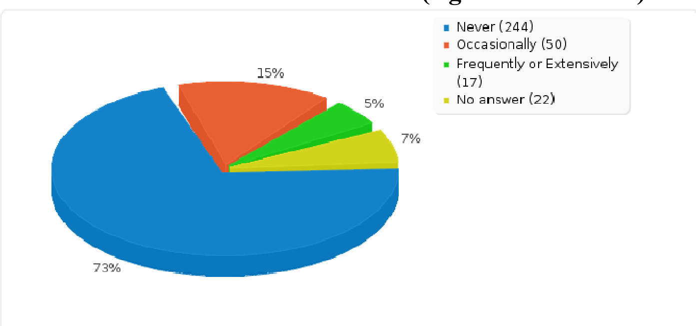


Chart 6.2.1-121: Grid interfaces (e.g. Globus toolkit)



39. Would you like training on this?

	ASAP		Within next year		Long term		Never		No answer	
Basic UNIX skills	11	3.30%	39	11.71%	39	11.71%	174	52.25%	70	21.02%
Scripting (shell, PERL, etc)	20	6.01%	56	16.82%	57	17.12%	139	41.74%	61	18.32%
Batch job systems (job submission and management)	19	5.71%	57	17.12%	63	18.92%	131	39.34%	63	18.92%
Scientific visualisation tools (e.g. VISIT, Paraview)	41	12.31%	76	22.82%	88	26.43%	74	22.22%	54	16.22%
Grid interfaces (e.g. Globus toolkit)	21	6.31%	41	12.31%	67	20.12%	128	38.44%	76	22.82%

Chart 6.2.1-122: Basic UNIX skills

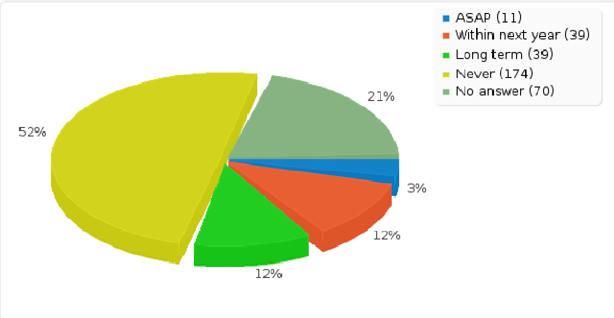


Chart 6.2.1-123: Scripting (shell, PERL, etc)

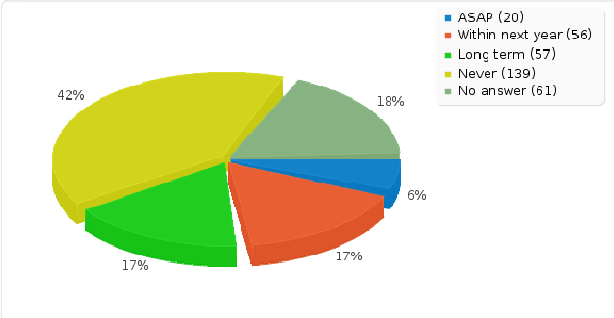


Chart 6.2.1-124: Batch job systems (job submission and management)

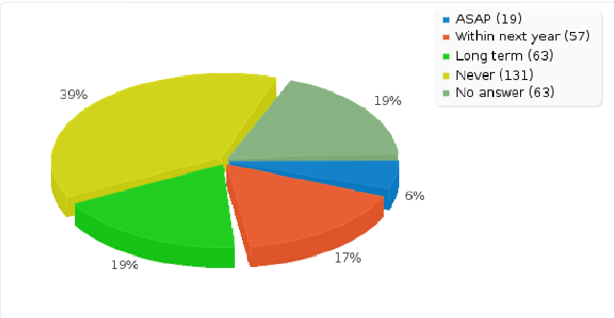


Chart 6.2.1-125: Scientific visualisation tools (e.g. VISIT, Paraview)

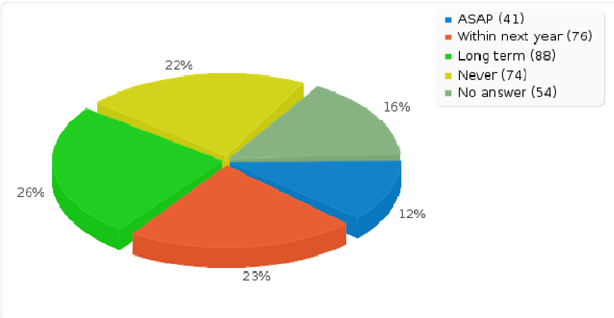
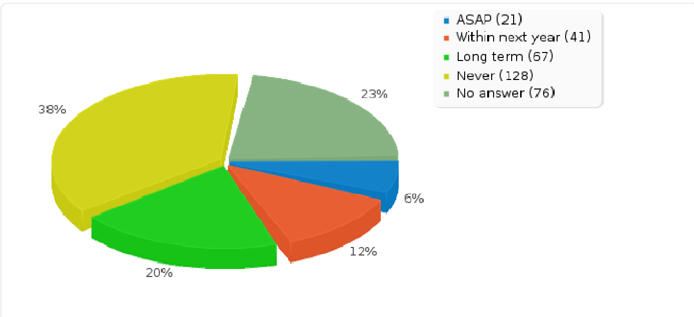


Chart 6.2.1-126: Grid interfaces (e.g. Globus toolkit)



40. Please comment on additional topics where you feel training is required and the type of training that should be provided (e.g. algorithm design, specific third party scientific applications)

Please write your answer here:

- *- algorithm design - performance oriented algorithm design - how to reduce memory consumption*
- *- algorithm design for hybrid openmp/mpi - openmp scaling optimization / cache and memory access optimization - usage of portable 3rd party tools (not vendor specific) for performance analysis*
- *- Optimizing memory access patterns (maximizing cache usage, etc.) - Knowing which optimizations are done by compilers and how to write code that can be optimized easily*

- *1) fundamentals of parallelization and efficiency 2) virtualization in general and methods and the impact on HPC and on the scaling of parallel codes 3) get some hardware experience on infiniband and 10G ethernet 4) big storage devices / SANs / NASs and corresponding file systems like GlusterFS / Lustre*
- *advanced scripting in COMSOL and MATLAB*
- *algorithm*
- *algorithm design implementation of numerical methods*
- *algorithm design*
- *algorithm design*
- *algorithm design*
- *algorithm design for molecular dynamics and statistical mechanics*
- *algorithm design, architecture-specific optimisations*
- *algorithm design, especially for upcoming hypbrid/many-core architectures in a workshop-based approach (i.e. handling (close to) real world examples) test concepts for HPC applications*
- *algorithm design, python*
- *Algorithm design.*
- *best practices for good scaling on Tier-0 systems*
- *Code optimization for C++ with respect to an efficient usage of cache based architectures.*
- *Debugging in Linux/Windows for a Fortran code using softwares like Eclipse.*
- *efficient use of parallel filesystems (especially alternatives to HDF5) load balancing of hybrid algorithms (OpenMP+MPI, CUDA/OpenCL+OpenMP+MPI) fine tunning MPI based programs on infiniband interconnect OpenCL basic and advanced course*
- *evtl.: domain decomposition/load balancing strategies for different machine architectures, especially "hybrid" (mixed shared and distributed memory requiring MPI and OpenMP - for example cluster with 16 compute nodes, 4 processors each and 4 cores per processor). rem.: "No answer" in some of the questions above usually means I already attended a course/got some training about the concerned topic.*
- *Gromacs for molecular dynamics*
- *HPC on Multi-GPU with MPI+CUDA*
- *I co-develop tools, compilers, and high performance libraries*
- *I feel that algorithm design should be taught.*
- *I find that many students are unfamiliar with the basic concepts around the mechanics of using libraries and are confused by the "magic incantations" needed to link them. Training on linking models, library compatibility, what to do when code doesn't compile etc would be helpful. Note that the HECToR UK service already provides such training as part of a course on core algorithms and libraries. This is run jointly with the University of Warwick Centre for Scientific Computing. It is often oversubscribed.*
- *I would be interested in hearing about successful methods for designing numerical codes that retain conventional good software engineering practice. I am also interesting in advanced training for extreme scalability (i.e. large parts of Tier 0 machines)*
- *Information on future hardware technologies and their implications for scientific programming (basic level, long term); parallelization strategies (advanced, within a year); optimization strategies (in addition to the basic handling of the tools, advanced, long term)*
- *MKL Vtune*
- *Not just visualisation tools, but scientific visualisation per se: tech savvy people usually can figure out how to use e.g. VISIT, but some training on what makes a great visualisation and what tells a good/honest scientific story would be helpful.*
- *optimization and parallelization strategies*
- *Other parallel libraries (e.g. Zoltan).*
- *Parallel algorithmic design, especially with respective to collective operations, and there efficient deployment would be nice.*
- *Parallel algorithms and data structures*
- *Parallelization methodologies and strategies*
- *scalasca , gdb*
- *scientific computing - scalable algorithms design*
- *source code or general documentation; version control for collective software/application development*
- *Specific third party scientific applications optimization, in particular, VASP*
- *team-based software engineering in HPC, auxiliary tools - e.g. versioning systems*
- *The absolute top-level courses are often missing.*
- *The survey is one-sided / biased in several respects and ignorant to important areas of (potential) use of parallelism. It's all geared towards the conventional numerical HPC and the scientific research community. You completely leave out the area of mathematical optimisation / OR. There are important and demanding*

industrial applications in this area, and usage of parallel processing paradigms in this area is largely underdeveloped with relevant software tools / base library packages practically missing (they exist, but not with sufficiently well developed support for parallelism). However, there is, in principle, a strong need for it, in various problem domains; but note that, here, we are not talking about days of CPU time on, say, 32 k processors / cores, but acceleration of run-times for ab-initio solution of large problem domains from 1 day to several minutes, or for partial (re-)optimization after problem modifications from few minutes to few seconds. With the present state of the art, one is practically limited to a few CPU's (considered to be single core); there is a lack of awareness / support / tools (maybe they exist, but are not known of) to employ (the upcoming / future) many-multi-core CPU's in combination with many CPU's ("many" >> 8, << 10k). If thresholds in front of employing parallelism in these application areas were substantially lowered, there could be and would be, in parctice, a much larger adoption of HPC / parallel approaches.

- *Tools like Vampir, Paraver, etc.*
- *Versioning (GIT) Software licensing / protection Zoltan*
- *Workshops on third party scientific applications in chemistry; includes VASP, Quatum Espresso, CPMD.*

Final Remarks

41. Please provide any additional personal comments on HPC training that will be relevant and beneficial to your research.

Please write your answer here:

- *My role on the HPC computing is to provide scientific guidance. My coding is on the "science codes" which are in C, and on the overall system design. There are many details of the HPC computing that I'm not familiar with, but people on my team must be (or must become) expert with. I strongly support your efforts to create a robust training environment!*
- *Strongly support online training, interactive courses and all chances to avoid having to travel. That would save time, money and help the environment too. Also, would suggest courses to be given in English, as well as keeping English as language for the slides and online material. Thank you for the opportunity to provide my feedback, keep up the excellent work !*
- *+: well-designed modular course program (programming languages/FORTRAN -> parallelization/MPI,OpenMP -> UPC/CAF/CUDA) +: courses are offered periodically - can be attended later in case of a "time slot collision" +: very experienced teachers - each course part/chapter is presented by the best specialist in the concerned area*
- *A course on optimisation and on code design would be very helpful (or would have been when I began implementing my code). Especially nice would be a guided bottleneck search regarding our own code to avoid the worst deficiencies.*
- *Although in this survey I expressed little personal interest in various training areas, I nevertheless would consider most of them usefull and important for younger HPC users.*
- *Am i bit puzzled by this question as I am convinced, based on my experience, that the only way to learn is to (I get informed of a tool to satisfy your needs (ii work hard on your own (or follow long formal courses at the university, if it is a broad scope subject) (iii exchange opinions with experts when you have already formed the basic competences via (ii. In other words, I am not at all convinced that the standard training offered by computing centres can change the attitude of users with respect or computing (...I speak these words having given courses of the type I was talking about).*
- *As principal investigator I follow the scientific part and I am not directly involved in programming, but I follow the learning and improvements on HPC knowledges of phd and post-docs. For me, thus, it is important to be informed in HPC training and code developments*
- *C++ templates*
- *Create a list of users who share the same scopes. For example, users who make use of the high performance computing for finite element simulations in Fortran.*
- *Development of robust algorithms with respect to failure of parallel CPUs in exascale computing.*
- *For any HPC system a 'Getting Started'-Manual (with additional information linked) and/or a 'How-To' containing essential news should be a default requirement!*
- *For my scientific research I think it might be useful to Teach programming languages,batch job systems for job submission and management, and OpenMPI.*
- *From my point of view, higher level libraries (Trilinos, PETSc) are quite underestimated compared to low-level parallelization tools in PRACE training activities.*
- *Hands-on assistance for improvements to our own code.*
- *High performance libraries like MKL should be taught more frequently and from basic.*

- *I am actually quite happy with the range/scope/frequency of training offered in Germany. I regularly send some of our incoming students to such courses. It is promising that such activities are extended to the European level in PRACE. For more than 10 years there was a surprisingly long-lived, simple, successful paradigm: C/C++ with MPI on clusters with fast interconnects (at least for our simulations of PDEs in physics). This is changing now, with various challenges but also opportunities with regard to heterogeneous multicore platforms. Future HPC training on GPU computing, multicore platforms (Knights Corner), and supporting software frameworks will be very important.*
- *I am fine with training courses available.*
- *I am working with huge number of data. It will be interesting if HPC training provides specific topics for this kind of "massively data processing".*
- *I believe the seminars should be taped and available on streaming for non attendants. (Podcast like access). Enphasis on system limitations would be appreciated (ionodes limits , files per user quotas , filesystems)*
- *I do not have any Idea, you questions above included all topics...*
- *I have trouble traveling an staying in hotels, due to my illness (MS)*
- *I like the training available, but lack the time to attend it regularly. For new Ph.D. students, courses should be held on a yearly basis.*
- *I really liked the Audio-CD version of the MPI-openmp lecture by Rolf Rabenseifner, together with the course material and the references it provided everything i needed to successfully use MPI and openmp.*
- *I would like to see more hands-on in the training events (or at least being something more than "change the source a little bit, compile and submit"). Remote learning could be also beneficial. A "course" with weekly lectures (webinars, video etc.) and exercises with (restricted) access to HPC infrastructure.*
- *I would love attend any courses on Harware-Sofwarecodesign or HLS (High level Sysnthese) hands-on courses..*
- *I'd like to learn to use better Totalview to debug or some fancy optimization tool (e.g. Scalasca? or equivalent/better?)*
- *Important is to have courses with concrete hands on typical research problems of the trainees.*
- *In our case, much of the special training courses is included in the regular curriculum, but I agree of course that these types of courese are necessary to educate other domain scientists who have less experience with HPC. So we use HPC courses only occasionally.*
- *It is important for me courses in English speaking*
- *It is important to separate between training of computer center staff and the end-users (aka "scientists"). It might make sense to open some courses only to HPC staff.*
- *Lack of time is the main constraint to improving my HPC skills. I hope to learn a number of the topics i am not proficient in "on the job" and/or through our local HPC centres. PRACE schools could be potentially very useful to students and young researchers.*
- *Main problems for me are: there are courses offered, but too far away and too time-consuming. Webcast would be a nice alternative.*
- *Maybe the upcoming parallel extension like OpenACC, OpenCL and C++ AMP would be of interest. Thank you for your work on this.*
- *Perhaps stimulate my scientific colleagues to learn something more than just F77 and use hardwarde more efficiently.*
- *Right now, I'm research computing support staff at University of Porto, Portugal. Besides training the users and potential users directly, I think PRACE should train a network of people that could act in close proximity with researchers at each institution, in order to filter and then support potential users of your high-end infrastructures, and I'm interested and available for that ([5]), and so is my institution.*
- *Running the program with the best efficiency.*
- *see 27.)*
- *Source code (i.e. software) specific trainings (scalability, fine tuning of performances), including user meetings within virtual community for knowledge exchange, discusion and alike.*
- *The CUDA course was very helpful and well-structured. I would do this course or another offered by HLRS again.*
- *The programming skills of students should be massively increased at the high school level, *before* entering university.*
- *Training is in general for beginners. I am missing training for experts. Most courses give recipes for standard problems without much information about the underlaying details. But details about cpu*

[5] E-mail address redacted.

architecture, auto vectorization by the compiler etc. is important to know while programming, to write code suitable to hardware and compiler.

- *Updates on the latest tools and compilers, exchange with the developers of tools, libraries, and compilers, in depth updates on hardware trends and their implications to such tools*
- *Using big machines to break cryptographic algorithms seems to be unknown, but maybe the topic is too special.*
- *What I'm often missing more than training courses is up-to-date and more complete online documentation of different supercomputing platforms accessible via the websites of the respective computing centers. This comprises for instance options for batch systems as well as hardware-specific information beneficial for optimization.*

6.2.2 Trainer Survey

General Information and HPC Background

1. Name *

Please write your answer(s) here:

First name

Family name

2. E-mail *

Please write your answer here:

3. Affiliation *

Please write your answer here:

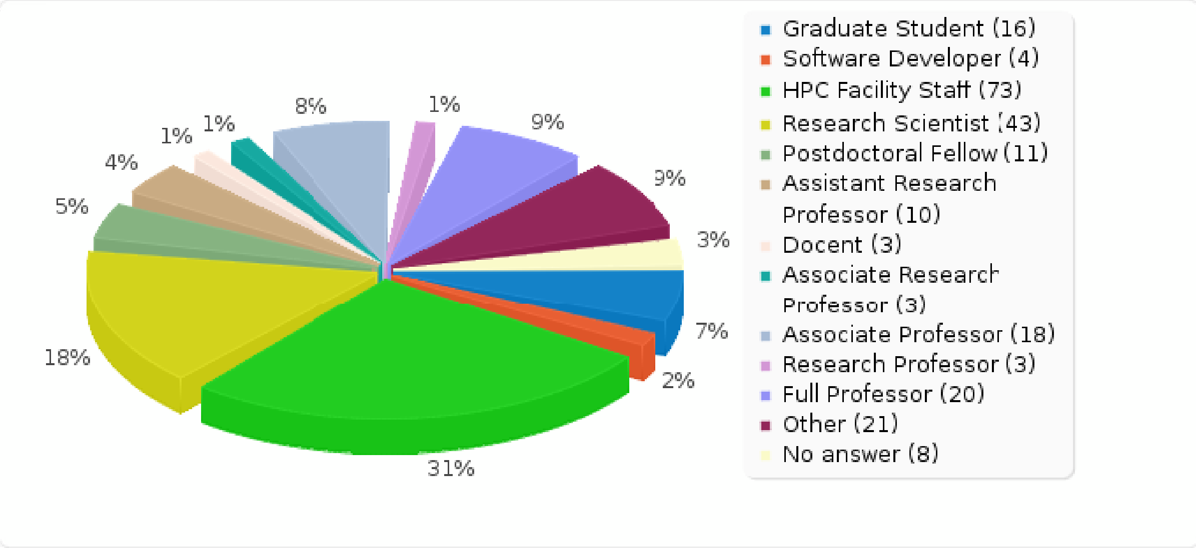
4. Country *

Please write your answer here:

5. What is your academic status? *

Answer	Count	Percentage	
Graduate Student	16	6.87%	**Other answers: IBM Employee. HPC IT Specialist; PhD student; Guest Professor; M.Sc; Research Team Leader; Post-doctoral short-term employee; Undergraduate Student; Computing Officer (Lecturer status); PhD Candidate; research engineer; Senior Scientist / Group leader; HPC company Staff;** http://survey.ipb.ac.rs/admin/admin.php?action=browse&sid=68456&subaction=id&id=127**Cray onsite application support; Director of Technology; Associate Research Scientist; HPC Specialist;** http://survey.ipb.ac.rs/admin/admin.php?action=browse&sid=68456&subaction=id&id=177**HPC Admin;** http://survey.ipb.ac.rs/admin/admin.php?action=browse&sid=68456&subaction=id&id=189**Senior Manager Industry;** http://survey.ipb.ac.rs/admin/admin.php?action=browse&sid=68456&subaction=id&id=211**HPC Facility Staff + Research assistant; PhD. Student; Senior Research Scientist, Ph.D.**
Software Developer	4	1.72%	
HPC Facility Staff	73	31.33 %	
Research Scientist	43	18.45 %	
Postdoctoral Fellow	11	4.72%	
Assistant Research Professor	10	4.29%	
Docent	3	1.29%	
Associate Research Professor	3	1.29%	
Associate Professor	18	7.73%	
Research Professor	3	1.29%	
Full Professor	20	8.58%	
Other	21	9.01%	
No answer	8	3.43%	

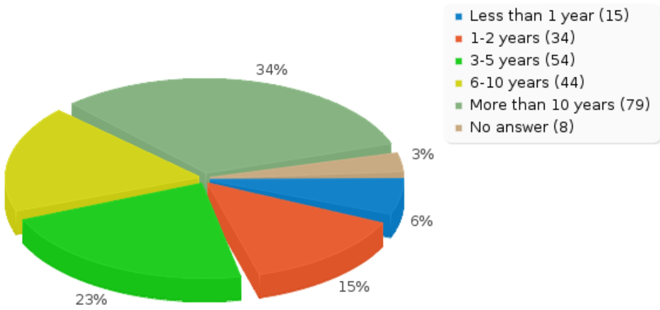
Chart 6.2.2-1


8%
1%
9%
1%
1%
4%
9%
5%
3%
7%
18%
2%
31%
Graduate Student (16)
Software Developer (4)
HPC Facility Staff (73)
Research Scientist (43)
Postdoctoral Fellow (11)
Assistant Research Professor (10)
Docent (3)
Associate Research Professor (3)
Associate Professor (18)
Research Professor (3)
Full Professor (20)
Other (21)
No answer (8)

6. How many years of experience do you have using/as a trainer for HPC systems? *

Chart 6.2.2-2

Answer	Count	Percentage
Less than 1 year	15	6.41%
1-2 years	34	14.53%
3-5 years	54	23.08%
6-10 years	44	18.80%
More than 10 years	79	33.76%
No answer	8	3.42%



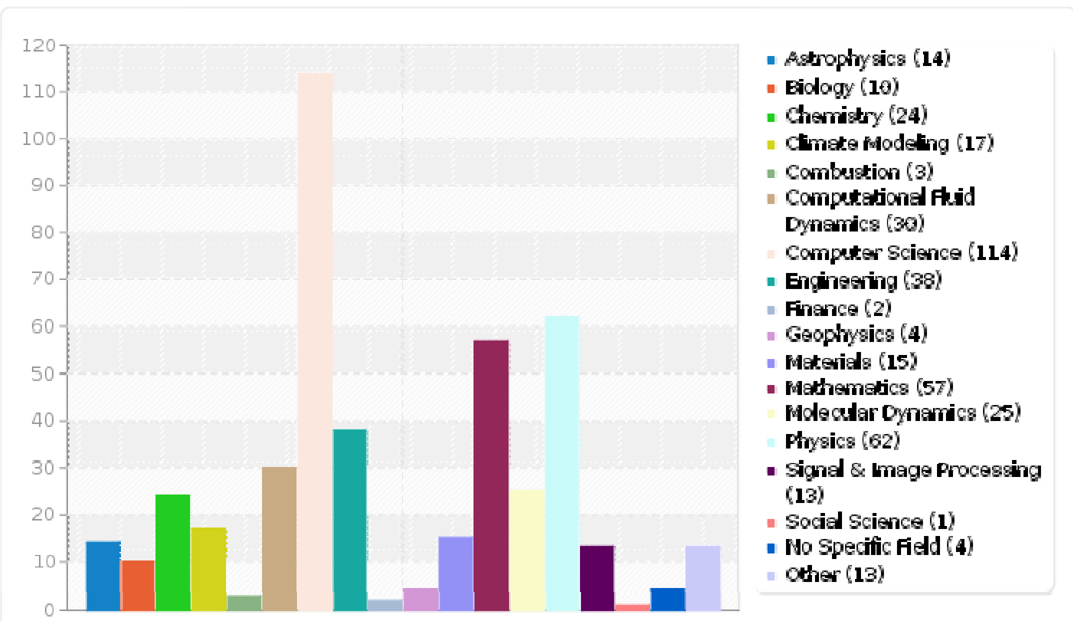
7. Is your background in any of the following domains? *

Answer	Count	Percentage
Astrophysics	14	5.98%
Biology	10	4.27%
Chemistry	24	10.26%
Climate Modeling	17	7.26%
Combustion	3	1.28%
Computational Fluid Dynamics	30	12.82%
Computer Science	114	48.72%
Engineering	38	16.24%
Finance	2	0.85%
Geophysics	4	1.71%
Materials	15	6.41%
Mathematics	57	24.36%
Molecular Dynamics	25	10.68%
Physics	62	26.50%
Signal & Image Processing	13	5.56%
Social Science	1	0.43%
No Specific Field	4	1.71%
Other	13	5.56%

Other:

**Biomechanics;
Plasma Physics;
Computational Mechanics;
Structural Mechanics;
Computer architecture;
Statistics;
Working for computer vendors for ~20 years;
Medicine;
Bioinformatics;
Numerical methods and algorithms;
Earth sciences;
Numerical analysis;
HPC, Computational Science**

Chart 6.2.2-3

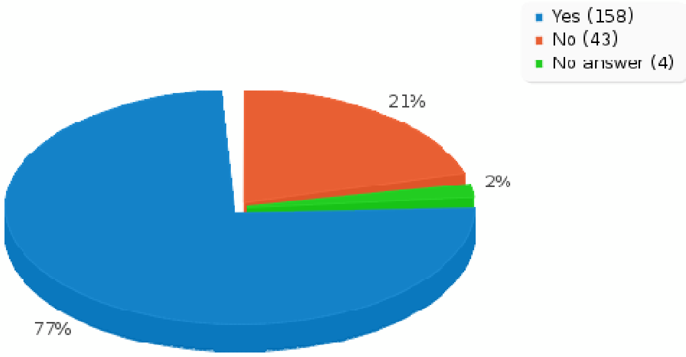


Programming Paradigms and Languages

8. Are the majority of your current programming language skills self-taught? *

Answer	Count	Percentage
Yes	158	77.07%
No	43	20.98%
No answer	4	1.95%

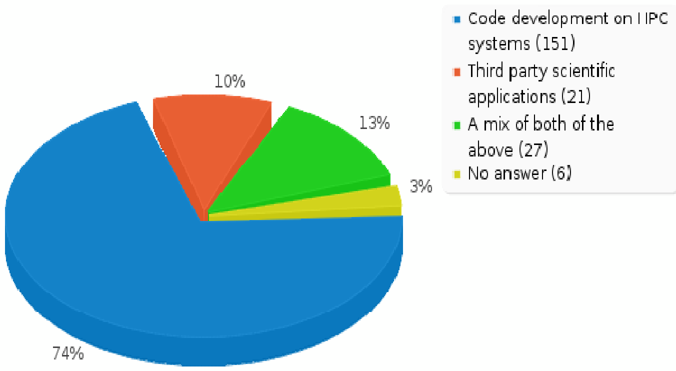
Chart 6.2.2-4



9. I mainly teach: *

Chart 6.2.2-5

Answer	Count	Percentage
Code development on HPC systems	151	73.66%
Third party scientific applications	21	10.24%
A mix of both of the above	27	13.17%
No answer	6	2.93%



10. Please specify the third party application(s) that you teach:

Only answer this question if the following conditions are met:

° Answer was 'Third party scientific applications' or 'A mix of both of the above' at question '9 [2.2]' (I mainly teach:)

Please write your answer here:

- *OpenFOAM*
- *No applications but System usage and Hardware technologies.*
- *Parallel R programming using SPRINT*
- *gLite, TORQUE ...*
- *LAMMPS*
- *Sequences pre-processors, assemblers, annotators*
- *Valgrind, Kcachegrind*
- *Lustre*
- *GPAW*
- *Globus*
- *MATLAB*
- *OpenFOAM*
- *NAMD, GROMACS*
- *Climate Modeling (REGCM4) / Molecular dynamics code (in past: DLPROTEIN)*
- *ABAQUS, Mathematica*
- *ECMWF Integrated Forecasting System (IFS)*
- *how to get the best performance of a given code on our supercomputer.*
- *VASP*
- *MPI/OpenMP/CUDA/Grid/Cloud software stacks WRF/CHIMERE/WW3/ROMS model stacks*
- *Classical molecular dynamics packages: NAMD, Gromacs, Amber CPMD CP2K Abinit*
- *PETSc*
- *WRF, CAMx, RegCM Flow-3D, SPHysics NAMD, CP2K, LAMMPS, DL-POLY, Abinit*
- *Dalton*
- *Combustion, two-phase flow, fluid mechanics*
- *Mathematica, and other proprietary codes that I cannot mention.*
- *MPI, WS-PGRADE/gUSE/P-GRADE, R, Bioperl*
- *Trilinos*
- *Computational Chemistry codes (cp2k, gromacs, namd, cpmd, gamess-us, nwchem etc)*
- *Computer modelers in Ecology and Computational Fluid Dynamics*

- *Visualization, Virtual Reality applications (COVISE/Paraview) Computer Graphics, OpenGL, ...*
- *Computational Fluid Dynamics*
- *vasp, cp2k, paragauss*
- *Air pollution computer modelling CFD*
- *Applications in virus genomics*
- *Statistical and Bioinformatics applications*
- *Bioinformatics, statistical and molecular dynamics applications*
- *Parallel Visualization softwares such as ParaView and VisIt. Plugin development for these softwares.*
- *Theoreticla and Computational Chemistry - MOLCAS program package*
- *Vampir, Vampirtrace, BenchIT*
- *Teach PG students to apply parallel programming, mathematical modeling, computational science research tools to specific scientific problems.*

11. Please indicate your current level of proficiency at providing training in the following parallel programming paradigms. *

	Proficient		**Some Knowledge**		**Little/No Knowledge**		**No answer**	
OpenMP	91	44.39%	81	39.51%	24	11.71%	9	4.39%
Basic MPI (point-to-point, collective communication)	138	67.32%	47	22.93%	11	5.37%	9	4.39%
Advanced MPI (MPI-I/O, one-sided communication)	64	31.22%	93	45.37%	39	19.02%	9	4.39%
Mixed-mode (hybrid) OpenMP-MPI	69	33.66%	84	40.98%	43	20.98%	9	4.39%
PGAS languages (CAF, UPC)	15	7.32%	46	22.44%	136	66.34%	8	3.90%
Next-generation languages (Chapel, X10, Fortress)	3	1.46%	29	14.15%	164	80.00%	9	4.39%
GPU computing (OpenCL, CUDA)	37	18.05%	77	37.56%	82	40.00%	9	4.39%

Chart 6.2.2-6: OpenMP

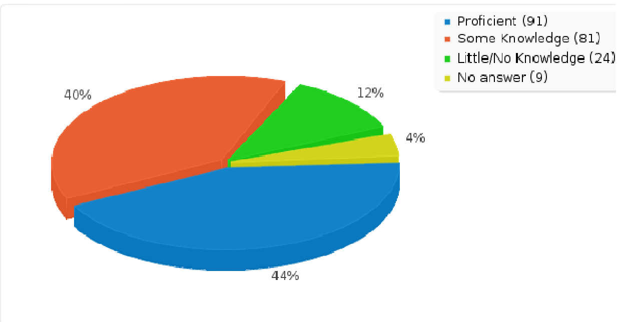


Chart 6.2.2-7: Basic MPI

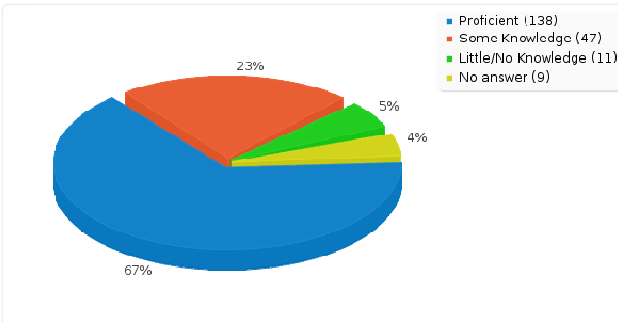


Chart 6.2.2-8: Advanced MPI

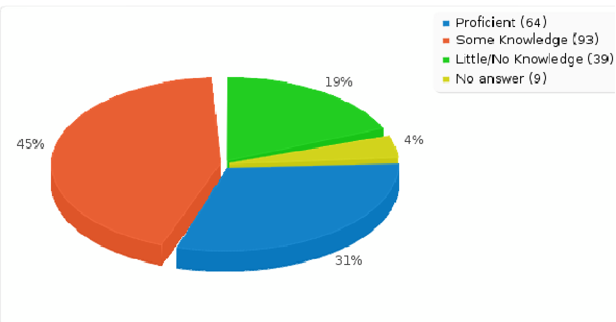


Chart 6.2.2-9: Mixed-mode (hybrid) OpenMP-MPI

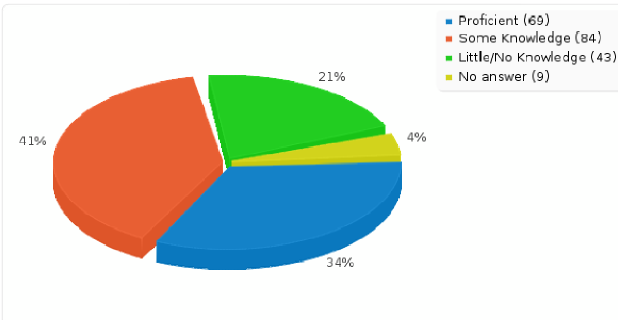


Chart 6.2.2-10: PGAS languages (CAF, UPC)

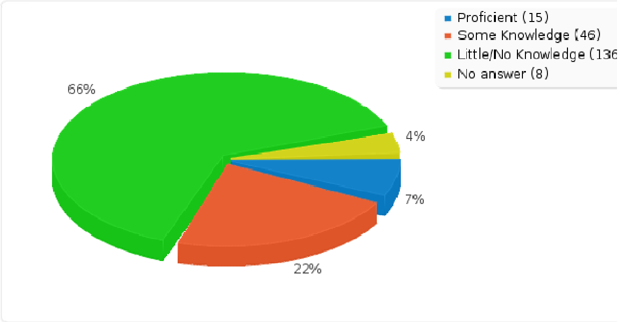


Chart 6.2.2-11: Next-generation languages (Chapel, X10, Fortress)

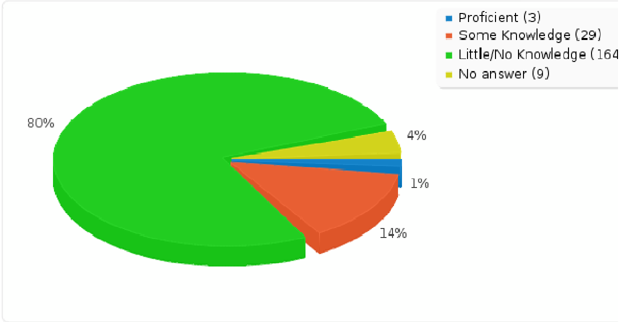
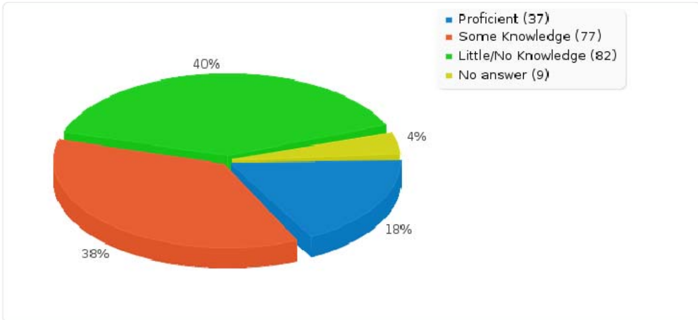


Chart 6.2.2-12: GPU computing (OpenCL, CUDA)



12. Please assess the current level of knowledge amongst students/trainees for the following parallel programming paradigms in your courses. *

	Sufficient		Needs improvement		Needs a lot improvement		I don't teach this		No answer	
OpenMP	23	11.22%	65	31.71%	28	13.66%	79	38.54%	10	4.88%
Basic MPI (point-to-point, collective communication)	42	20.49%	62	30.24%	30	14.63%	60	29.27%	11	5.37%
Advanced MPI (MPI-I/O, one-sided communication)	11	5.37%	35	17.07%	58	28.29%	90	43.90%	11	5.37%
Mixed-mode (hybrid) OpenMP-MPI	12	5.85%	41	20.00%	52	25.37%	88	42.93%	12	5.85%
PGAS languages (CAF, UPC)	2	0.98%	6	2.93%	21	10.24%	164	80.00%	12	5.85%
Next-generation languages (Chapel, X10, Fortress)	1	0.49%	7	3.41%	12	5.85%	172	83.90%	13	6.34%
GPU computing (OpenCL, CUDA)	8	3.90%	30	14.63%	32	15.61%	123	60.00%	12	5.85%

Chart 6.2.2-13: OpenMP

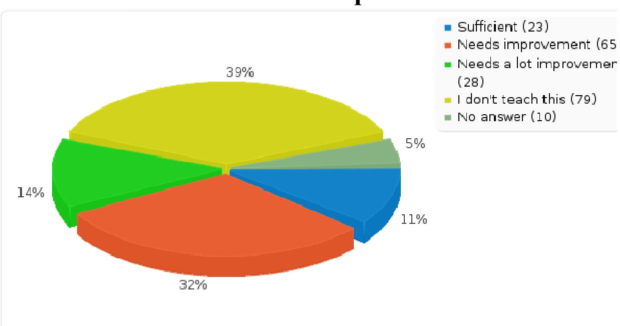


Chart 6.2.2-14: Basic MPI

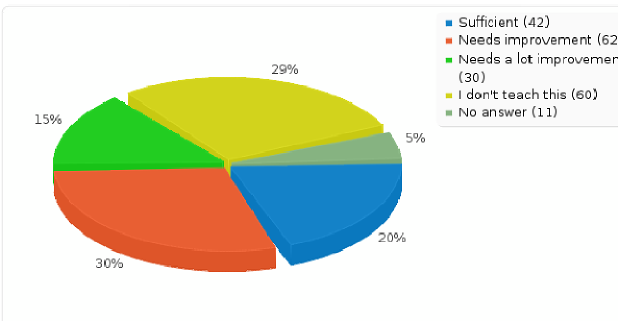


Chart 6.2.2-15: Advanced MPI

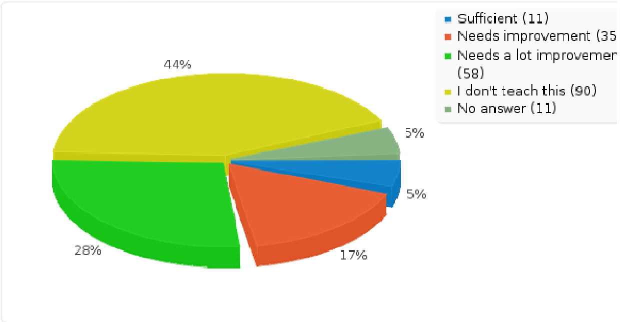


Chart 6.2.2-16: Mixed-mode (hybrid) OpenMP-MPI

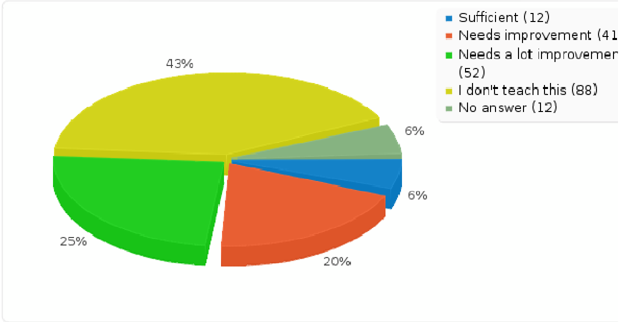


Chart 6.2.2-17: PGAS languages (CAF, UPC)

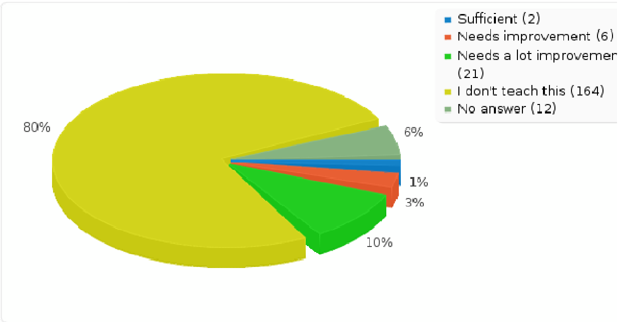


Chart 6.2.2-18: Next-generation languages (Chapel, X10, Fortress)

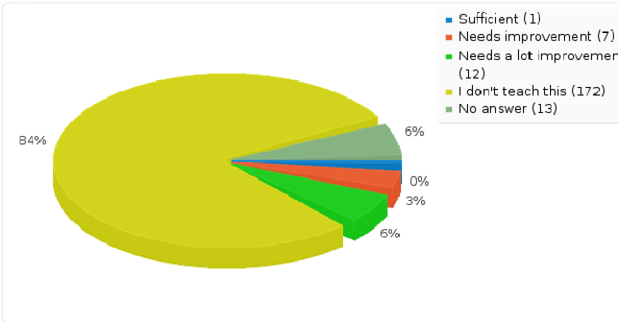
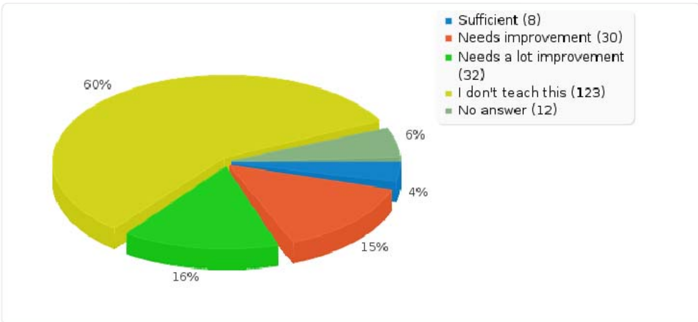


Chart 6.2.2-19: GPU computing (OpenCL, CUDA)



13. Please indicate your current level of proficiency at providing training in the following programming languages. *

	Proficient		Some Knowledge		Little/No Knowledge		No answer	
C/C++	114	55.61%	55	26.83%	24	11.71%	12	5.85%
Fortran 77	100	48.78%	63	30.73%	30	14.63%	12	5.85%
Fortran 90, 95, onwards	86	41.95%	73	35.61%	34	16.59%	12	5.85%
Java	31	15.12%	73	35.61%	89	43.41%	12	5.85%
Python	18	8.78%	73	35.61%	102	49.76%	12	5.85%
Matlab/R	24	11.71%	71	34.63%	98	47.80%	12	5.85%

Chart 6.2.2-20: C/C++

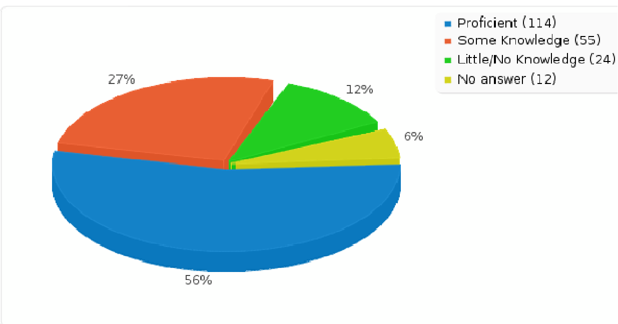


Chart 6.2.2-21: Fortran 77

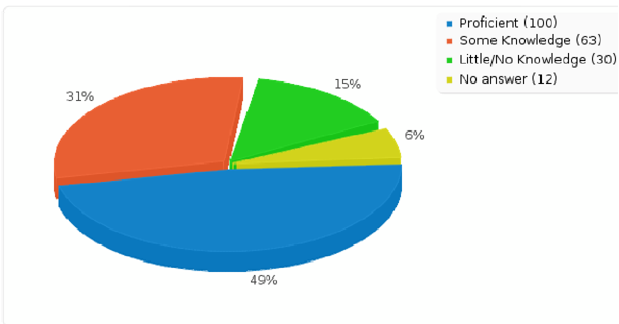


Chart 6.2.2-22: Fortran 90, 95, onwards

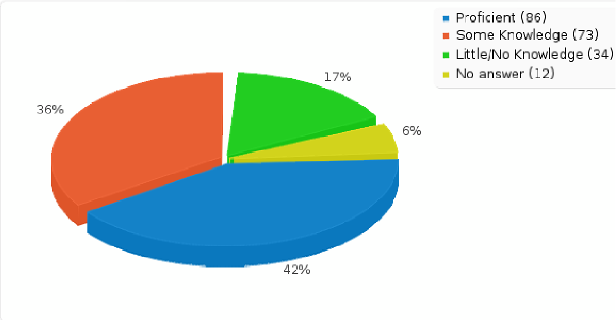


Chart 6.2.2-23: Java

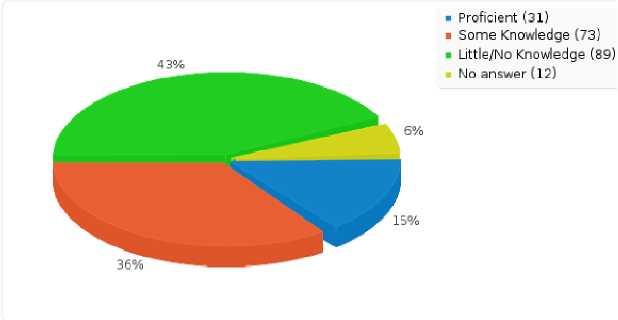


Chart 6.2.2-24: Python

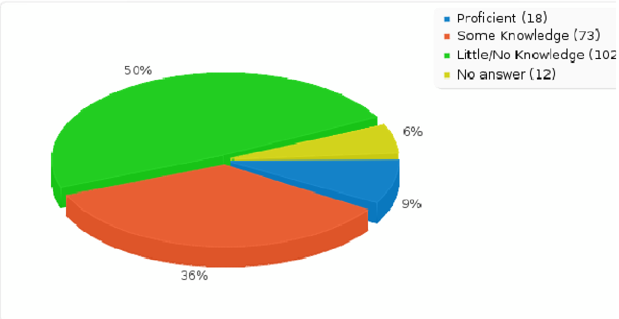
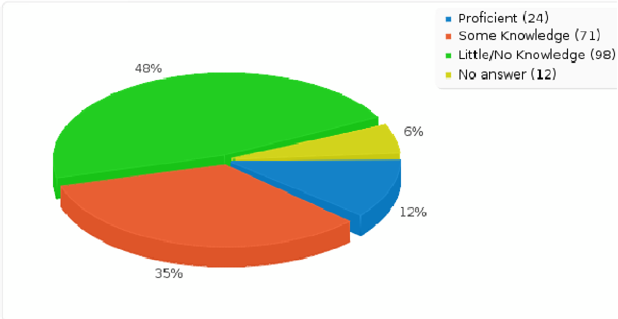


Chart 6.2.2-25: Matlab/R



14. Please assess the current level of knowledge amongst students/trainees for the following programming languages in your courses. *

	Sufficient		Needs improvement		Needs a lot improvement		I don't teach this		No answer	
C/C++	59	28.78%	59	28.78%	14	6.83%	61	29.76%	12	5.85%
Fortran 77	32	15.61%	44	21.46%	22	10.73%	95	46.34%	12	5.85%
Fortran 90, 95, onwards	28	13.66%	52	25.37%	26	12.68%	87	42.44%	12	5.85%
Java	27	13.17%	18	8.78%	2	0.98%	146	71.22%	12	5.85%
Python	10	4.88%	27	13.17%	10	4.88%	146	71.22%	12	5.85%
Matlab/R	17	8.29%	22	10.73%	13	6.34%	141	68.78%	12	5.85%

Chart 6.2.2-26: C/C++

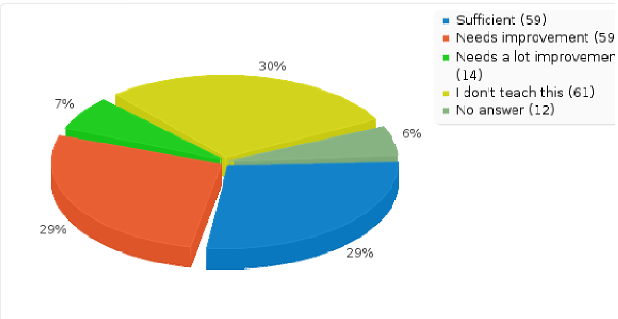


Chart 6.2.2-27: Fortran 77

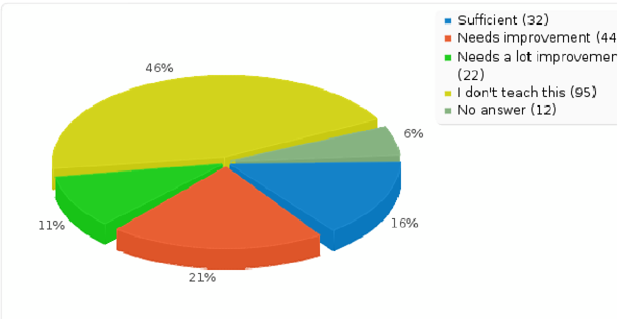


Chart 6.2.2-28: Fortran 90, 95, onwards

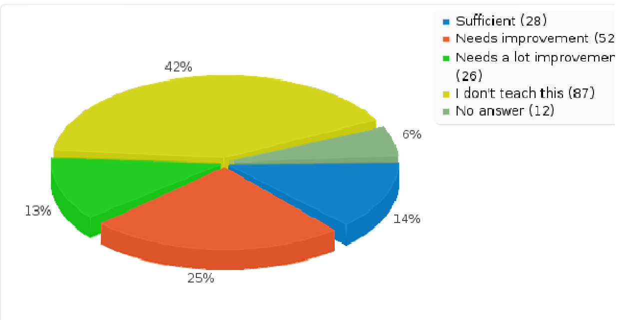


Chart 6.2.2-29: Java

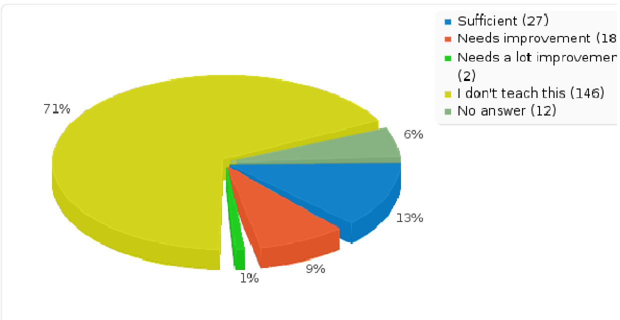


Chart 6.2.2-30: Python

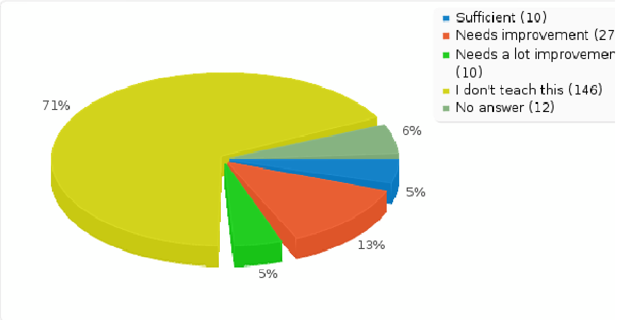
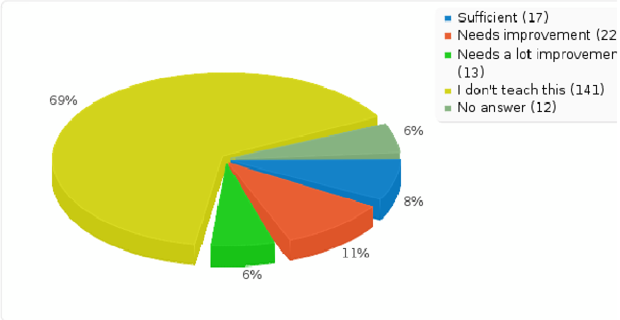


Chart 6.2.2-31: Matlab/R



15. Please indicate your current level of proficiency at providing training in the following libraries and code development tools. *

	Proficient		Some Knowledge		Little/No Knowledge		No answer	
Basic numerical libraries (e.g. LAPACK, EISPACK)	60	29.27%	86	41.95%	47	22.93%	12	5.85%
High-level numerical libraries (e.g. PETSc, Trilinos)	20	9.76%	67	32.68%	106	51.71%	12	5.85%
Parallel I/O libraries (HDF5, Parallel NetCDF)	19	9.27%	78	38.05%	96	46.83%	12	5.85%
General compiler usage and optimisation	114	55.61%	66	32.20%	13	6.34%	12	5.85%
Debugging tools and techniques	87	42.44%	88	42.93%	18	8.78%	12	5.85%
Performance analysis/optimisation tools and techniques	92	44.88%	78	38.05%	23	11.22%	12	5.85%

Chart 6.2.2-32: Basic numerical libraries (e.g. LAPACK, EISPACK)

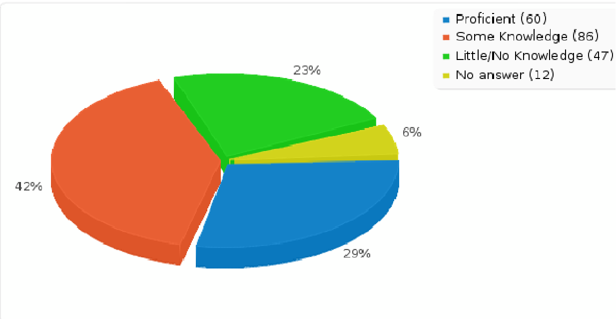


Chart 6.2.2-33: High-level numerical libraries (e.g. PETSc, Trilinos)

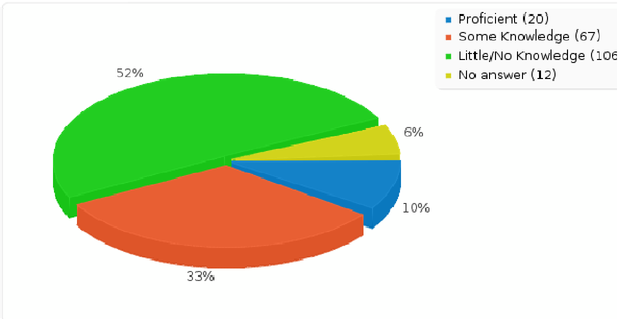


Chart 6.2.2-34: Parallel I/O libraries (HDF5, Parallel NetCDF)

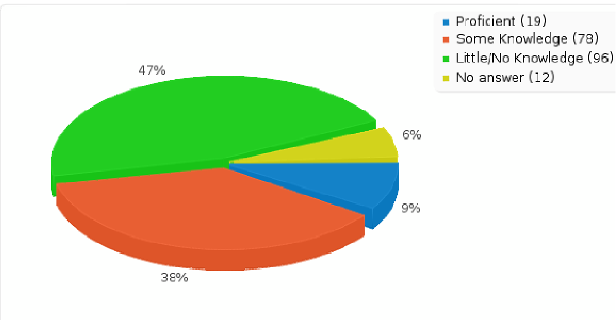


Chart 6.2.2-35: General compiler usage and optimisation

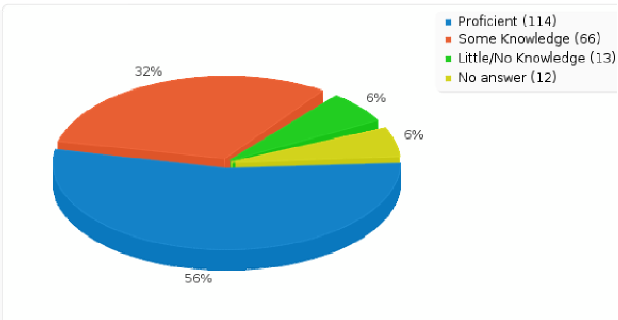


Chart 6.2.2-36: Debugging tools and techniques

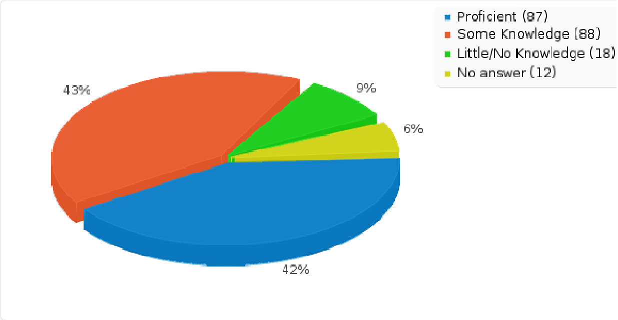
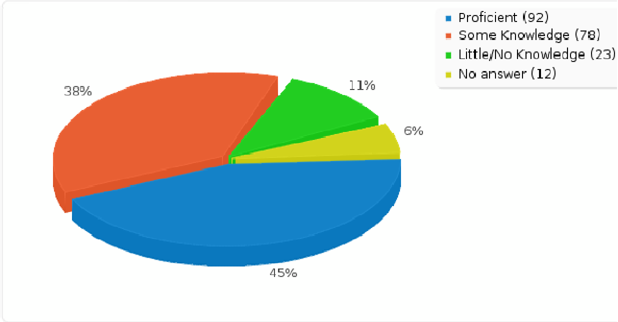


Chart 6.2.2-37: Performance analysis/optimisation tools and techniques



16. Please assess the current level of knowledge amongst students/trainees for the following libraries/code development tools in your courses. *

	Sufficient		Needs improvement		Needs a lot improvement		I don't teach this		No answer	
Basic numerical libraries (e.g. LAPACK, EISPACK)	18	8.78%	51	24.88%	31	15.12%	93	45.37%	12	5.85%
High-level numerical libraries (e.g. PETSc, Trilinos)	3	1.46%	32	15.61%	31	15.12%	127	61.95%	12	5.85%
Parallel I/O libraries (HDF5, Parallel NetCDF)	4	1.95%	16	7.80%	42	20.49%	131	63.90%	12	5.85%
General compiler usage and optimisation	22	10.73%	79	38.54%	32	15.61%	60	29.27%	12	5.85%
Debugging tools and techniques	11	5.37%	68	33.17%	56	27.32%	58	28.29%	12	5.85%
Performance analysis/optimisation tools and techniques	10	4.88%	55	26.83%	74	36.10%	54	26.34%	12	5.85%

Chart 6.2.2-38: Basic numerical libraries (e.g. LAPACK, EISPACK)

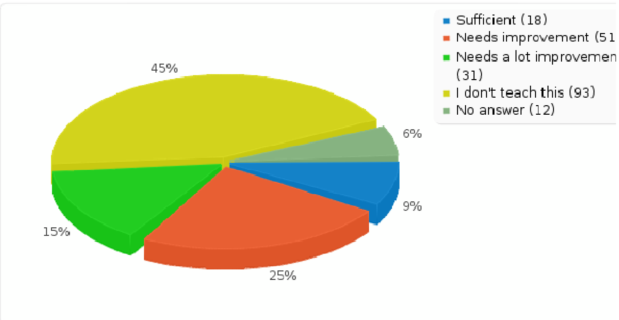


Chart 6.2.2-39: High-level numerical libraries (e.g. PETSc, Trilinos)

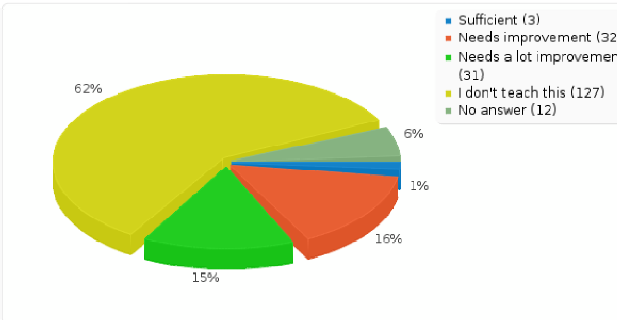


Chart 6.2.2-40: Parallel I/O libraries (HDF5, Parallel NetCDF)

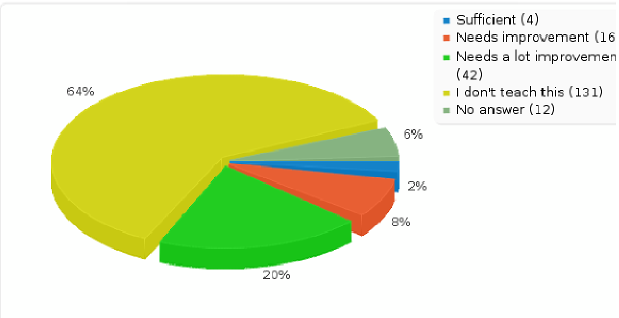


Chart 6.2.2-41: General compiler usage and optimisation

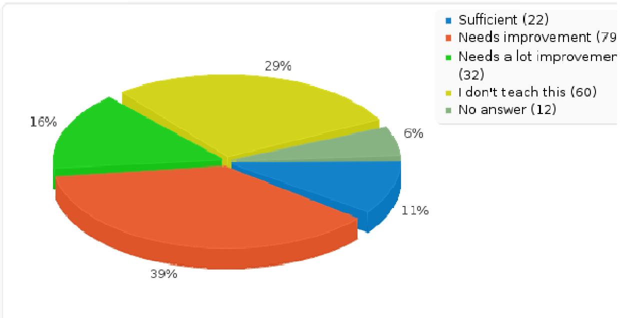


Chart 6.2.2-42: Debugging tools and techniques

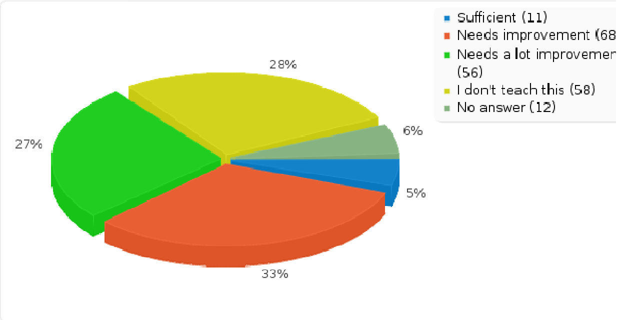
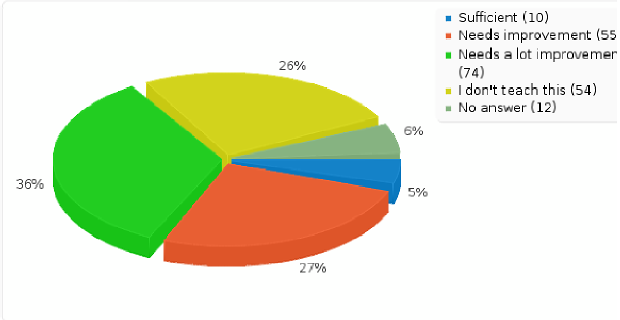


Chart 6.2.2-43: Performance analysis/optimisation tools and techniques



17. Please indicate your current level of proficiency at providing training in the following subjects. *

	Proficient		Some Knowledge		Little/No Knowledge		No answer	
Basic UNIX skills	157	76.59%	30	14.63%	6	2.93%	12	5.85%
Scripting (shell, PERL, etc)	112	54.63%	66	32.20%	14	6.83%	13	6.34%
Batch job systems (job submission and management)	132	64.39%	54	26.34%	7	3.41%	12	5.85%
Version control software (e.g. subversion, cvs, git)	79	38.54%	93	45.37%	21	10.24%	12	5.85%
Code Documentation Tools	33	16.10%	104	50.73%	56	27.32%	12	5.85%
Checkpoint/Restart implementation	26	12.68%	88	42.93%	79	38.54%	12	5.85%
Scientific visualization tools (e.g. VISIT, Paraview)	23	11.22%	67	32.68%	103	50.24%	12	5.85%
Grid interfaces (e.g. Globus toolkit)	24	11.71%	52	25.37%	117	57.07%	12	5.85%

Chart 6.2.2-44: Basic UNIX skills

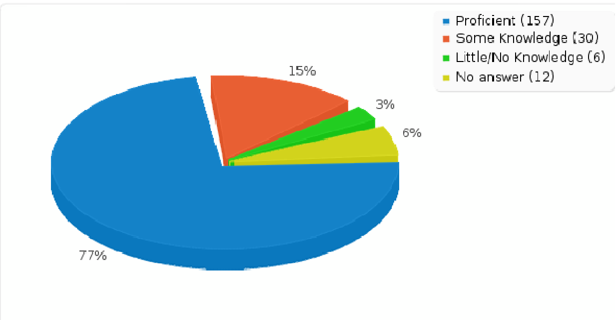


Chart 6.2.2-45: Scripting (shell, PERL, etc)

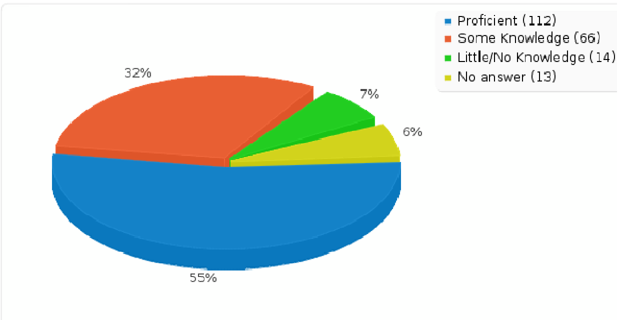


Chart 6.2.2-46: Batch job systems (job submission and management)

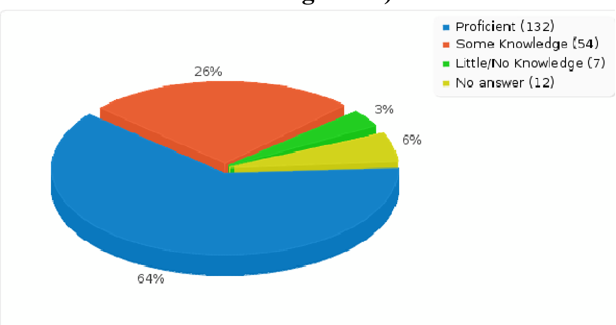


Chart 6.2.2-47: Version control software (e.g. subversion, cvs, git)

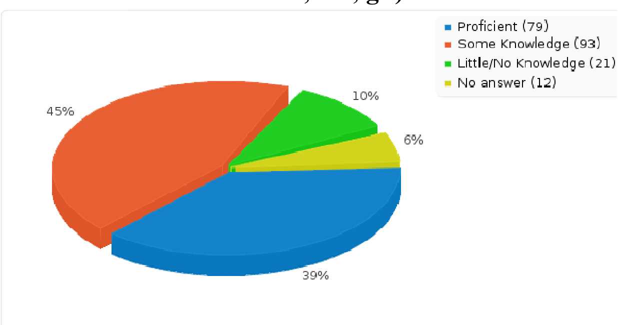


Chart 6.2.2-48: Code Documentation Tools

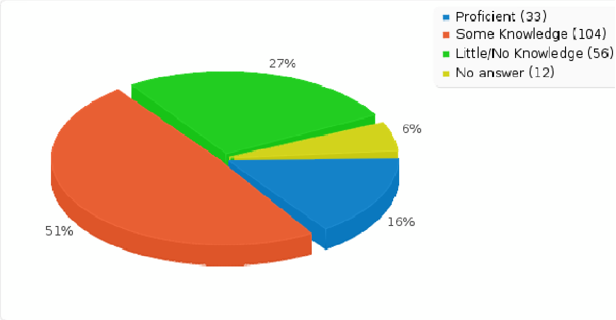


Chart 6.2.2-49: Checkpoint/Restart implementation

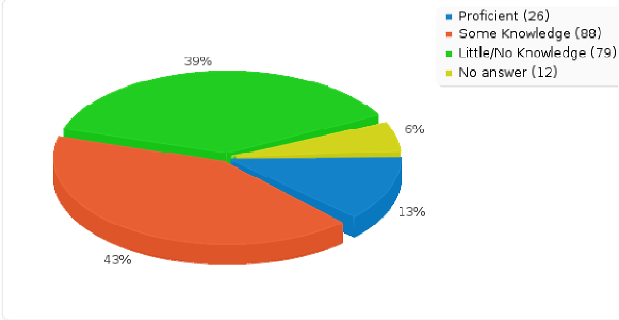


Chart 6.2.2-50: Scientific visualisation tools (e.g. VISIT, Paraview)

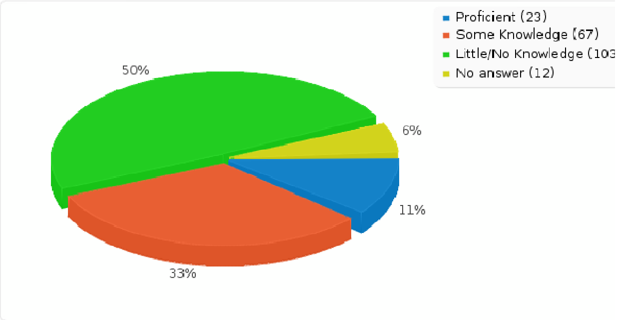
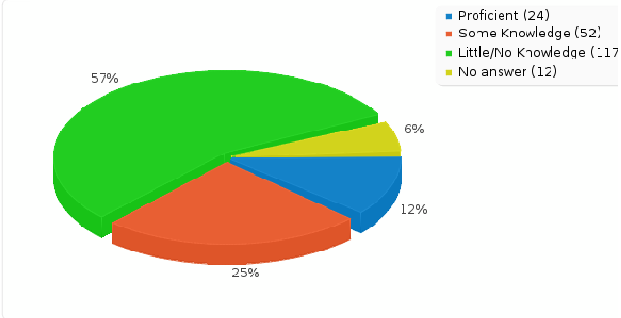


Chart 6.2.2-51: Grid interfaces (e.g. Globus toolkit)



18. Please assess the current level of knowledge amongst students/trainees for the following subjects in your courses. *

	Sufficient		Needs improvement		Needs a lot improvement		I don't teach this		No answer	
Basic UNIX skills	51	24.88%	67	32.68%	16	7.80%	59	28.78%	12	5.85%
Scripting (shell, PERL, etc)	34	16.59%	61	29.76%	27	13.17%	71	34.63%	12	5.85%
Batch job systems (job submission and management)	37	18.05%	71	34.63%	23	11.22%	62	30.24%	12	5.85%
Version control software (e.g. subversion, cvs, git)	21	10.24%	45	21.95%	24	11.71%	103	50.24%	12	5.85%
Code Documentation Tools	9	4.39%	32	15.61%	32	15.61%	120	58.54%	12	5.85%
Checkpoint/Restart implementation	3	1.46%	16	7.80%	32	15.61%	142	69.27%	12	5.85%
Scientific visualization tools (e.g. VISIT, Paraview)	8	3.90%	19	9.27%	32	15.61%	134	65.37%	12	5.85%
Grid interfaces (e.g. Globus toolkit)	4	1.95%	24	11.71%	12	5.85%	153	74.63%	12	5.85%

Chart 6.2.2-52: Basic UNIX skills

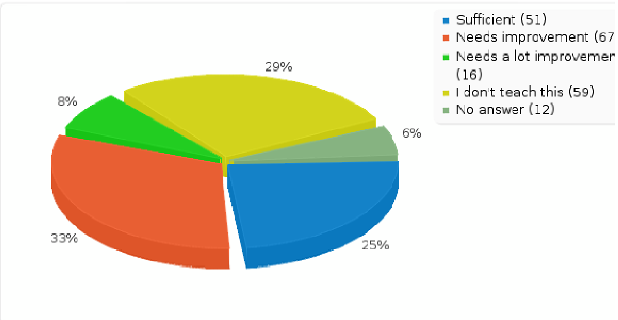


Chart 6.2.2-53: Scripting (shell, PERL, etc)

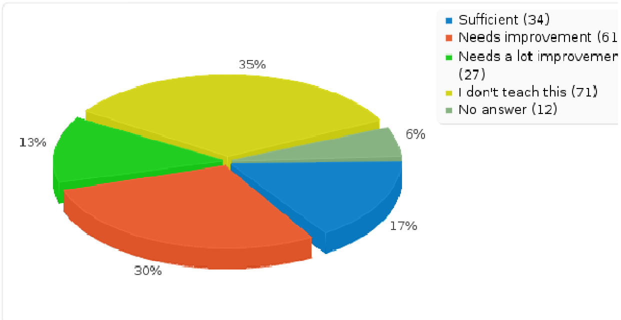


Chart 6.2.2-54: Batch job systems (job submission and management)

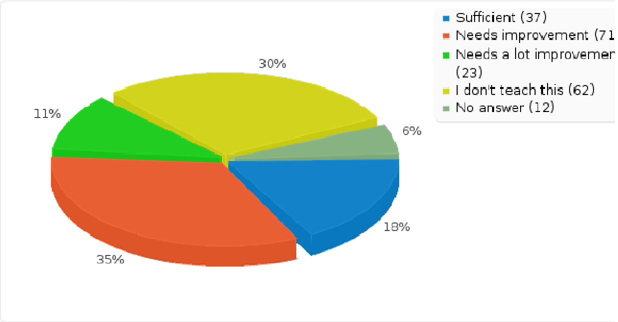


Chart 6.2.2-55: Version control software (e.g. subversion, cvs, git)

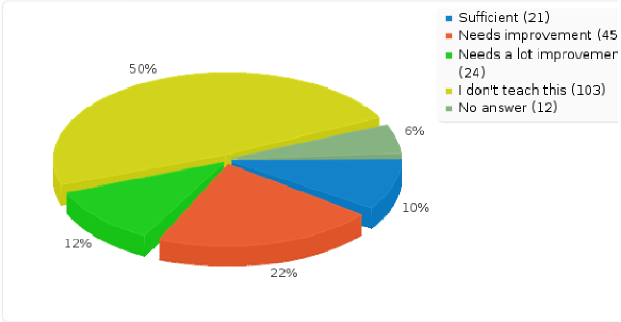


Chart 6.2.2-56: Code Documentation Tools

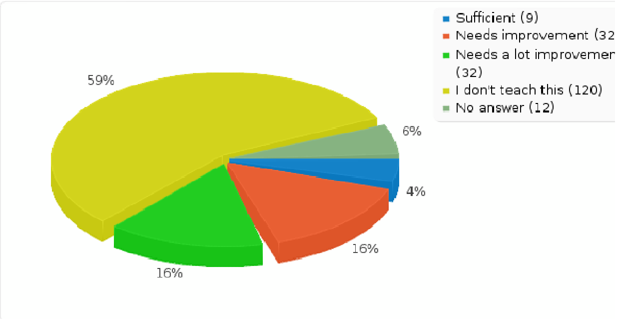


Chart 6.2.2-57: Checkpoint/Restart implementation

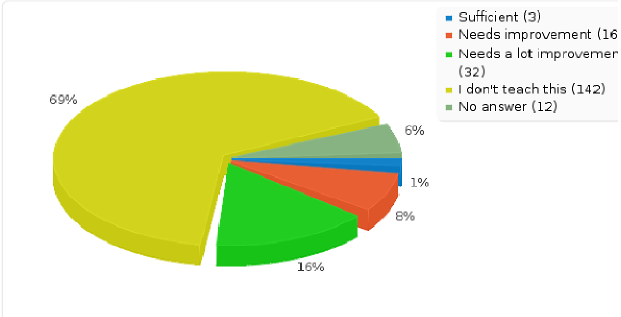


Chart 6.2.2-58: Scientific visualisation tools (e.g. VISIT, Paraview)

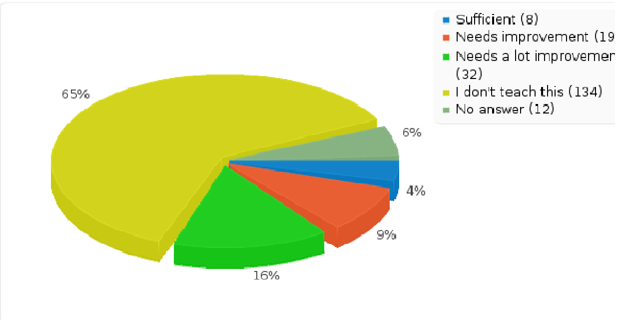
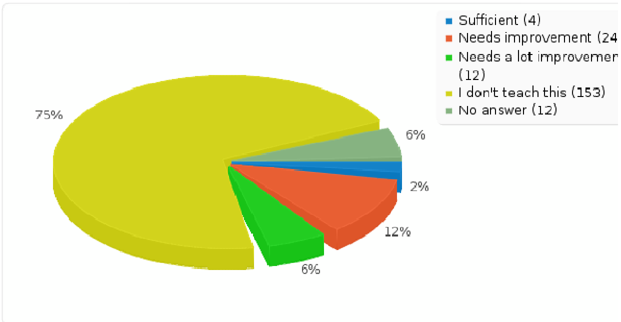


Chart 6.2.2-59: Grid interfaces (e.g. Globus toolkit)



19. Please indicate your current level of proficiency at providing training in the following Grid middleware stacks. *

	Proficient		Some Knowledge		Little/No Knowledge		No answer	
gLite	14	6.83%	28	13.66%	151	73.66%	12	5.85%
Globus	14	6.83%	48	23.41%	131	63.90%	12	5.85%
UNICORE	9	4.39%	39	19.02%	145	70.73%	12	5.85%
ARC	2	0.98%	21	10.24%	170	82.93%	12	5.85%

Chart 6.2.2-60: gLite

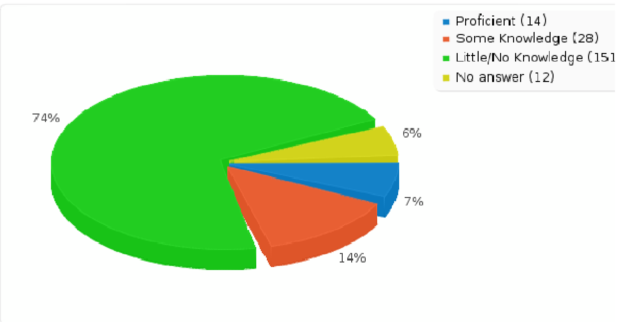


Chart 6.2.2-61: Globus

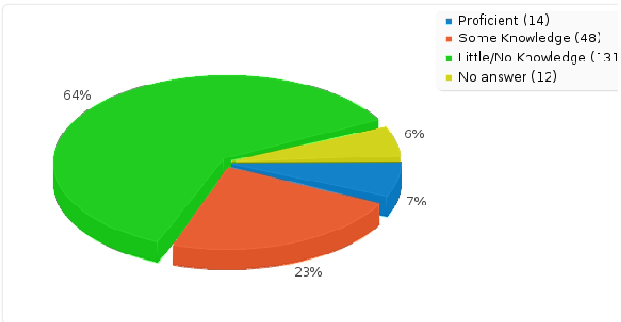


Chart 6.2.2-62: UNICORE

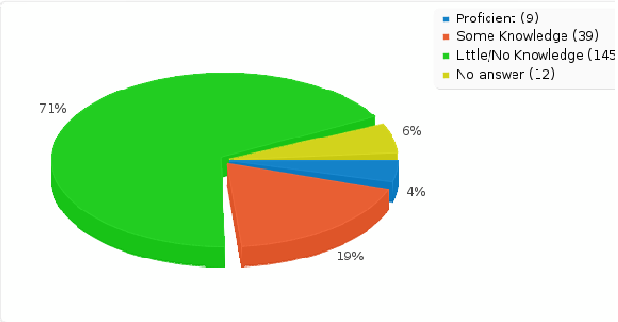
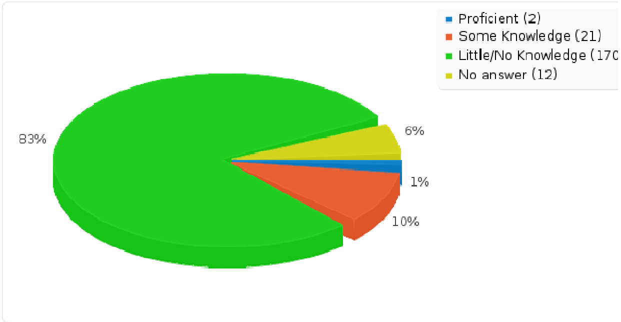


Chart 6.2.2-63: ARC



Parallel Programming Methodologies

20. At what level should HPC training focus on the following architectures? *

	None		Beginner Level		Advanced Level		No answer	
Vector Processors	41	22.04%	99	53.23%	43	23.12%	3	1.61%
Massively Parallel (MPPs)	9	4.84%	76	40.86%	98	52.69%	3	1.61%
Symmetric Multiprocessing (SMP)	7	3.76%	81	43.55%	95	51.08%	3	1.61%
Heterogeneous (mixed-architecture) system	16	8.60%	86	46.24%	81	43.55%	3	1.61%
Novel architectures (Cell, FPGA, GPU)	11	5.91%	109	58.60%	63	33.87%	3	1.61%

Chart 6.2.2-64: Vector Processors

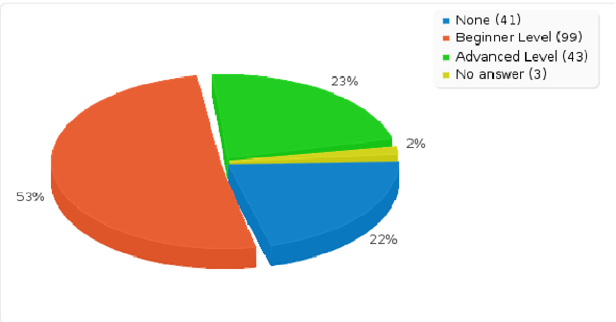


Chart 6.2.2-65: Massively Parallel (MPPs)

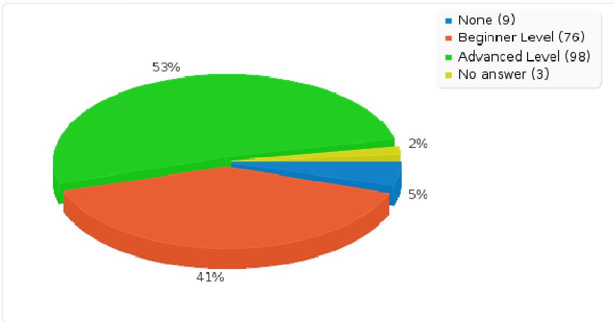


Chart 6.2.2-66: Symmetric Multiprocessing (SMP)

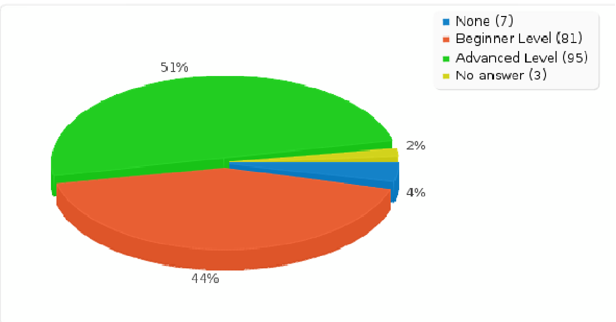


Chart 6.2.2-67: Heterogeneous (mixed-architecture) system

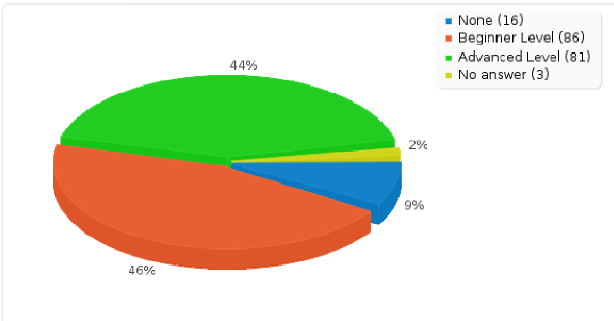
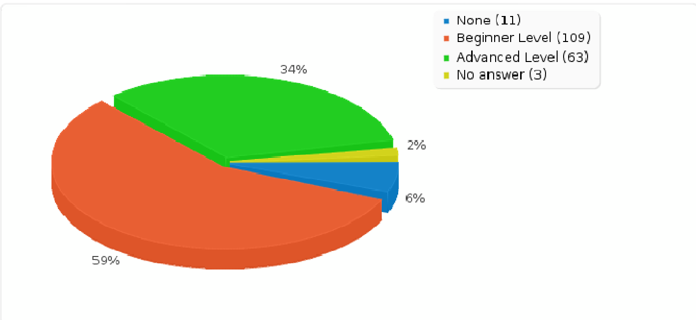


Chart 6.2.2-68: Novel architectures (Cell, FPGA, GPU)



21. At what level should HPC training focus on the following fundamental HPC principles? *

	None		Beginner Level		Advanced Level		No answer	
Scalability	2	1.08%	72	38.71%	109	58.60%	3	1.61%
Efficiency	2	1.08%	82	44.09%	99	53.23%	3	1.61%
Load balancing	2	1.08%	81	43.55%	100	53.76%	3	1.61%
Communication/Computation overlapping	6	3.23%	91	48.92%	86	46.24%	3	1.61%
Data decomposition	6	3.23%	95	51.08%	82	44.09%	3	1.61%
Task decomposition	7	3.76%	105	56.45%	71	38.17%	3	1.61%
Code optimization	5	2.69%	79	42.47%	99	53.23%	3	1.61%
Checkpointing	28	15.05%	105	56.45%	50	26.88%	3	1.61%

Chart 6.2.2-69: Scalability

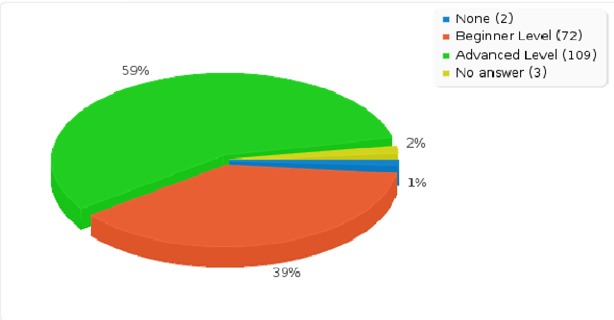


Chart 6.2.2-70: Efficiency

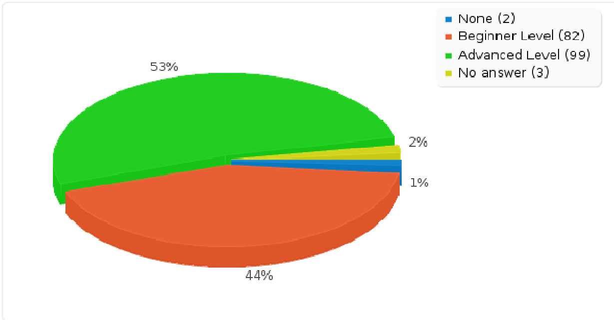


Chart 6.2.2-71: Load balancing

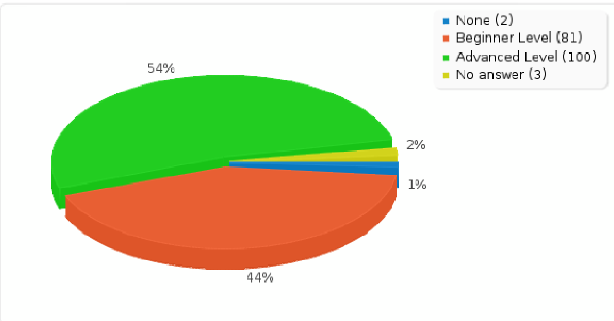


Chart 6.2.2-72: Communication/Computation overlapping

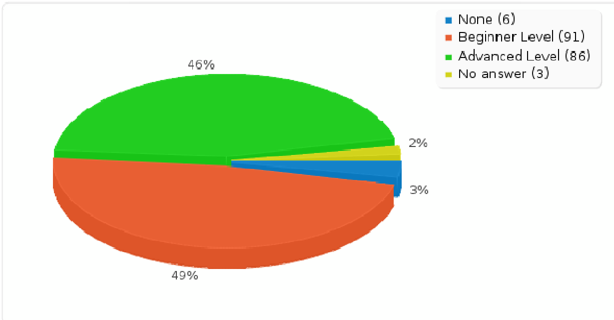


Chart 6.2.2-73: Data decomposition

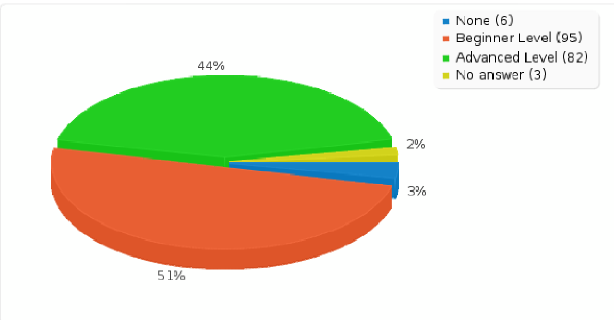


Chart 6.2.2-74: Task decomposition

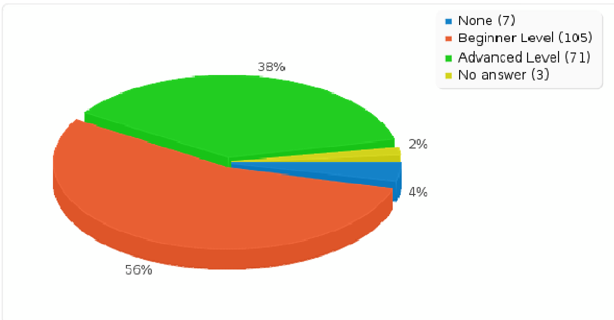


Chart 6.2.2-75: Code optimisation

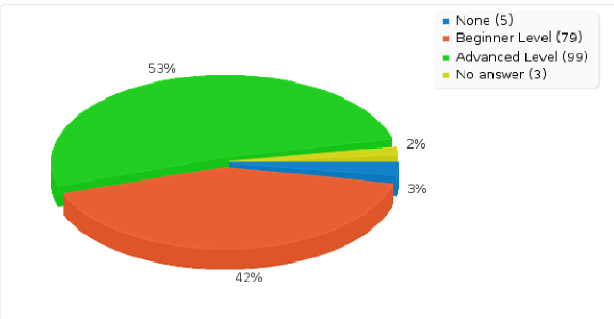
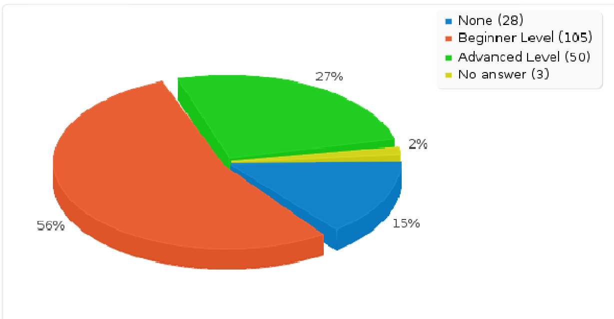


Chart 6.2.2-76: Checkpointing



22. How do you assess the availability of education and training material on the following topics?*

	Adequate		Inadequate		Don't Know		No answer	
HPC architectures	83	44.62%	65	34.95%	35	18.82%	3	1.61%
Fundamental HPC principles	94	50.54%	60	32.26%	29	15.59%	3	1.61%

Chart 6.2.2-77: HPC architectures

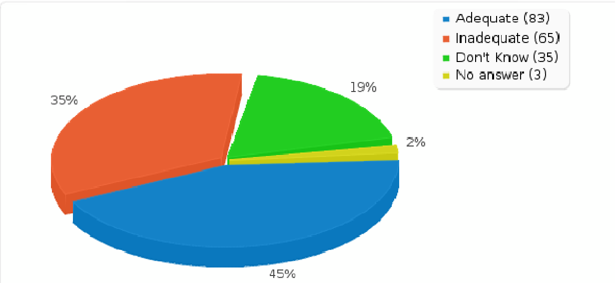
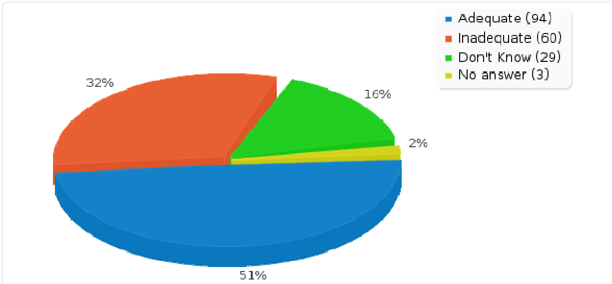


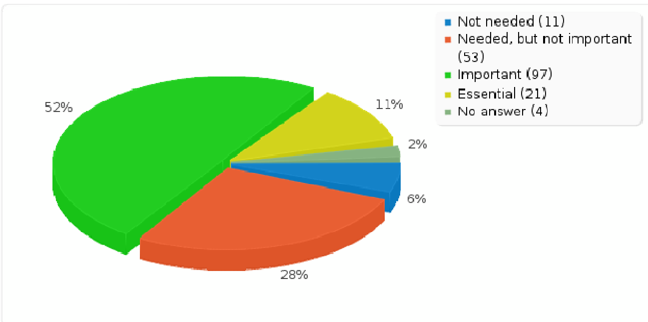
Chart 6.2.2-78: Fundamental HPC principles



23. To what extent does a sound theoretical background (e.g. mutual exclusion, computer architecture) affect the success of HPC programming training sessions? *

Answer	Count	Percentage
Not needed	11	5.91%
Needed, but not important	53	28.49%
Important	97	52.15%
Essential	21	11.29%
No answer	4	2.15%

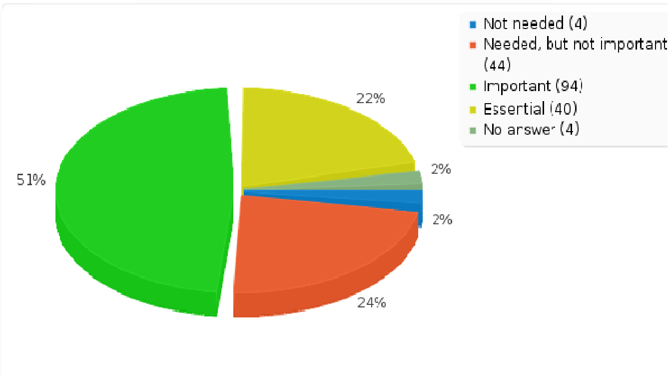
Chart 6.2.2-79



24. How important do you consider it to teach software engineering techniques as part of training for better practices and development of code? *

Chart 6.2.2-80

Answer	Count	Percentage
Not needed	4	2.15%
Needed, but not important	44	23.66%
Important	94	50.54%
Essential	40	21.51%
No answer	4	2.15%



25. What skill sets need the most development? *

	No Development		Some Development		A lot of Development		Don't Know		No answer	
Programming	9	4.84%	115	61.83%	47	25.27%	10	5.38%	5	2.69%
Advanced Programming and Parallel Programming	0	0.00%	39	20.97%	136	73.12%	6	3.23%	5	2.69%
Numerical Libraries and Algorithms	2	1.08%	91	48.92%	70	37.63%	18	9.68%	5	2.69%
Data Management	3	1.61%	88	47.31%	58	31.18%	32	17.20%	5	2.69%
Parallel I/O and Fault Tolerance	5	2.69%	61	32.80%	83	44.62%	32	17.20%	5	2.69%
Scientific	14	7.53%	80	43.01%	39	20.97%	48	25.81%	5	2.69%

Visualization										
Grid Computing	40	21.51%	53	28.49%	21	11.29%	67	36.02%	5	2.69%

Chart 6.2.2-81: Programming

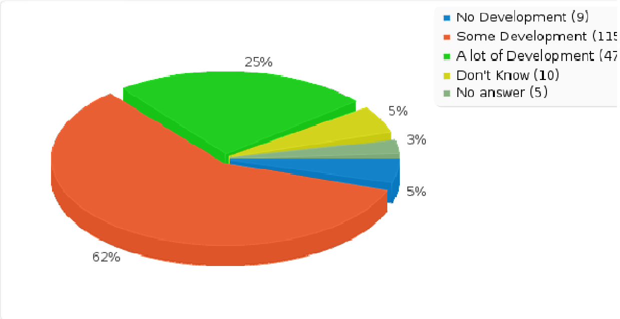


Chart 6.2.2-82: Advanced Programming and Parallel Programming

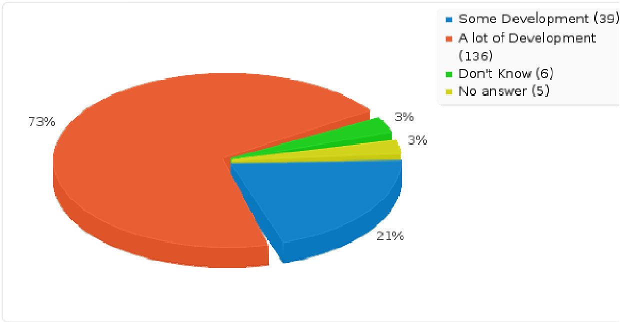


Chart 6.2.2-83: Numerical Libraries and Algorithms

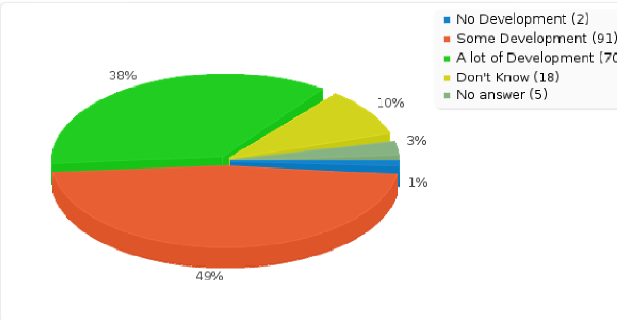


Chart 6.2.2-84: Data Management

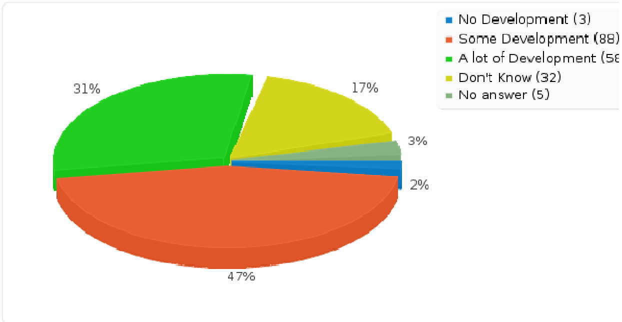


Chart 6.2.2-85: Parallel I/O and Fault Tolerance

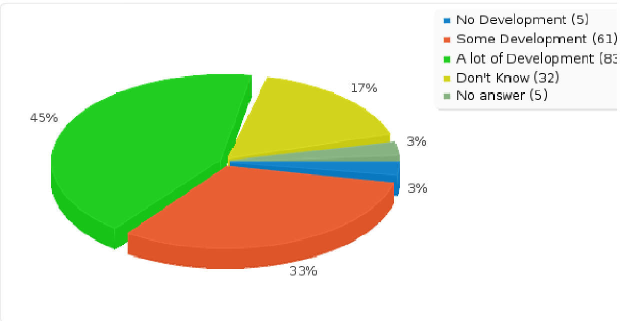


Chart 6.2.2-86: Scientific Visualization

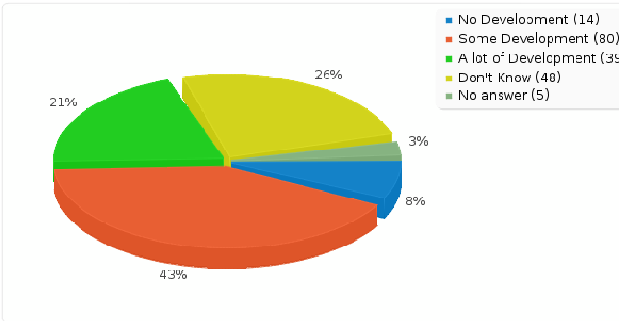
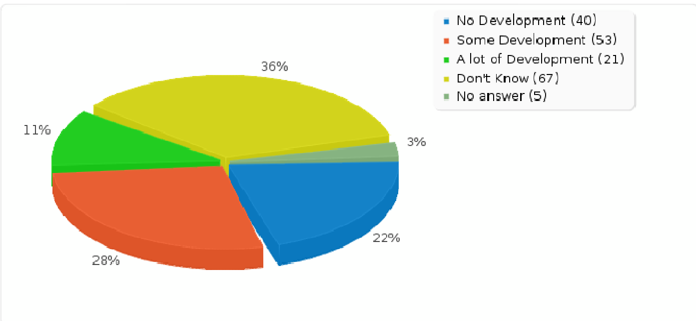


Chart 6.2.2-87: Grid Computing



Training Requirements

26. Please rank the importance of the following training methods - with 1 = "Little/No Importance" and 5 = "Very Important" *

	1		2		3		4		5		No Answer	
Face-to-Face classes	1	0.58%	7	4.09%	24	14.04%	56	32.75%	83	48.54%	0	0.00%
Combining lectures with hands-on sessions	0	0.00%	2	1.17%	8	4.68%	36	21.05%	125	73.10%	0	0.00%
Electronic slides	3	1.75%	17	9.94%	55	32.16%	57	33.33%	38	22.22%	1	0.58%
Online web tutorials	3	1.75%	28	16.37%	77	45.03%	41	23.98%	21	12.28%	1	0.58%
Interactive computer-based training courses	4	2.34%	34	19.88%	68	39.77%	39	22.81%	25	14.62%	1	0.58%
User Guides	0	0.00%	10	5.85%	56	32.75%	56	32.75%	48	28.07%	1	0.58%
Books	3	1.75%	27	15.79%	52	30.41%	56	32.75%	32	18.71%	1	0.58%
Journals	14	8.19%	72	42.11%	49	28.65%	27	15.79%	8	4.68%	1	0.58%
Virtual learning environments	21	12.28%	65	38.01%	56	32.75%	21	12.28%	6	3.51%	2	1.17%
Multi-cast remote training	27	15.79%	60	35.09%	65	38.01%	16	9.36%	2	1.17%	1	0.58%
Live web-broadcast	25	14.62%	61	35.67%	53	30.99%	27	15.79%	4	2.34%	1	0.58%
Flash documentation	65	38.01%	54	31.58%	36	21.05%	12	7.02%	3	1.75%	1	0.58%
Screen-casting	44	25.73%	57	33.33%	48	28.07%	16	9.36%	5	2.92%	1	0.58%

Chart 6.2.2-88: Face-to-Face classes

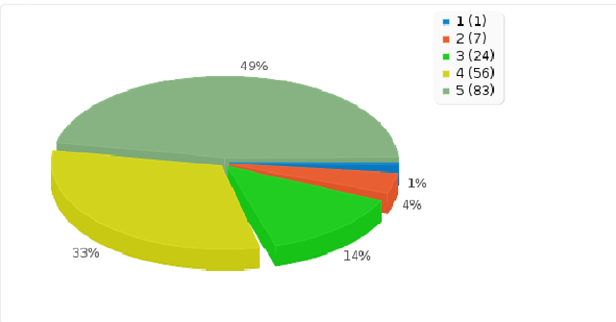


Chart 6.2.2-89: Combining lectures with hands-on sessions

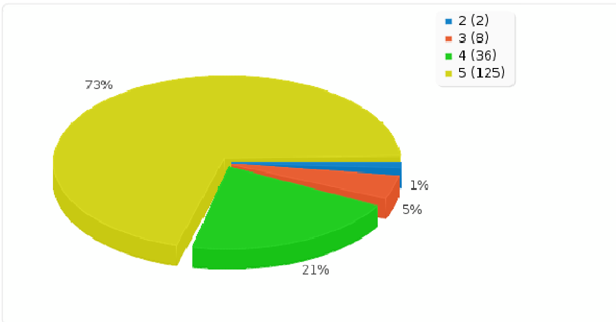


Chart 6.2.2-90: Electronic slides

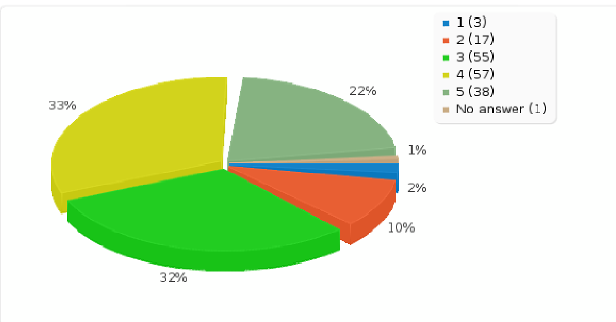


Chart 6.2.2-91: Online web tutorials

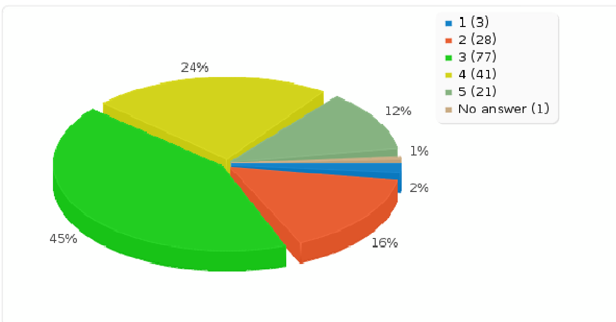


Chart 6.2.2-92: Interactive computer-based training courses

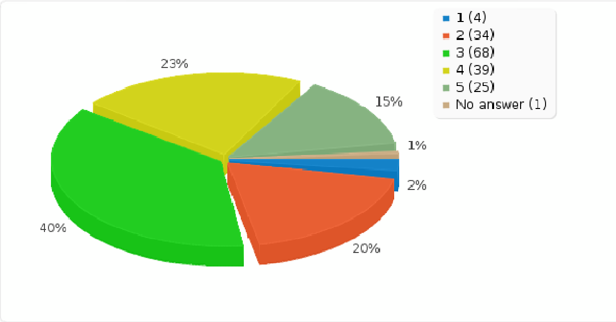


Chart 6.2.2-93: User Guides

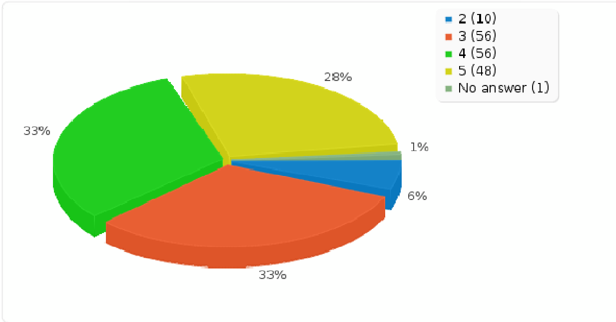


Chart 6.2.2-94: Books

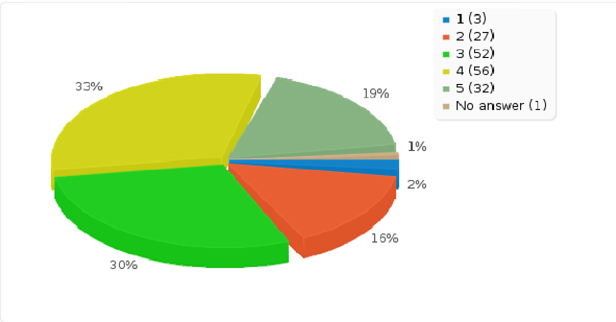


Chart 6.2.2-95: Journals

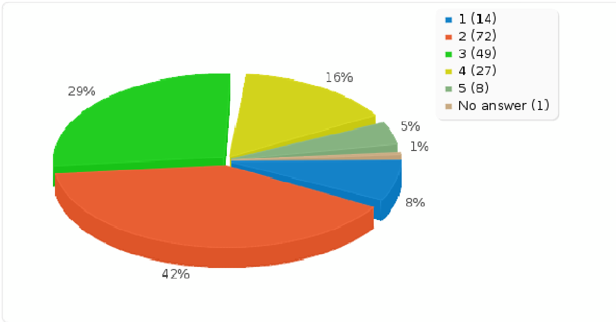


Chart 6.2.2-96: Virtual learning environments

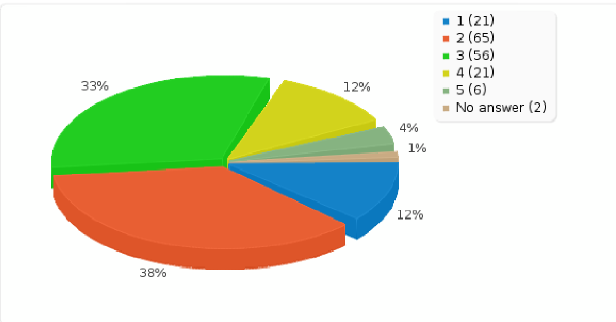


Chart 6.2.2-97: Multi-cast remote training

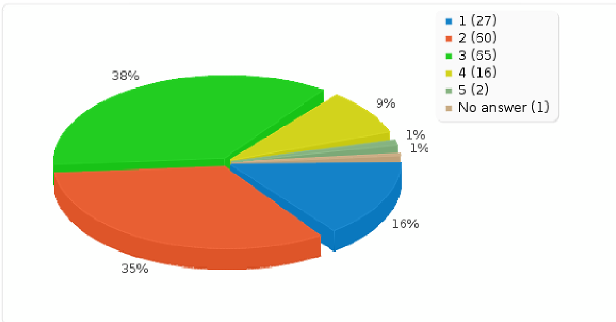


Chart 6.2.2-98: Live web-broadcast

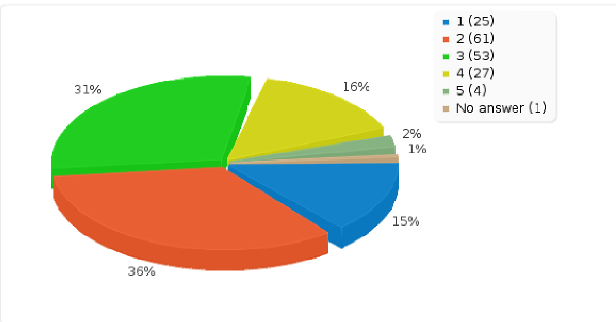


Chart 6.2.2-99: Flash documentation

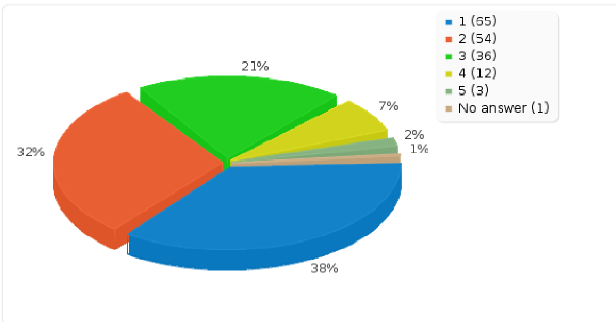
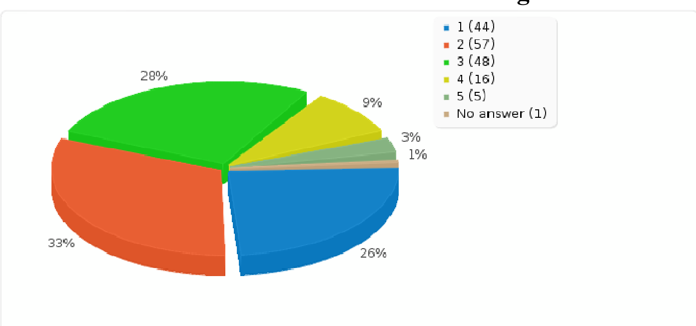


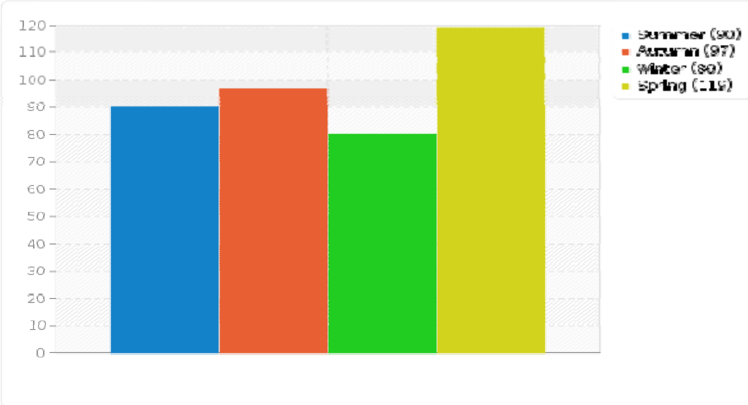
Chart 6.2.2-100: Screen-casting



27. When do you find it easiest to give lectures at a face-to-face course? *

Chart 6.2.2-101

Answer	Count	Percentage
Summer	90	38.46%
Autumn	97	41.45%
Winter	80	34.19%
Spring	119	50.85%



28. What is your preferred duration for delivering a face-to-face training course for the following levels? *

	1 day		2 days		3 - 4 days		5 days		More than 5 days		No answer	
Beginner	19	11.11%	57	33.33%	55	32.16%	21	12.28%	17	9.94%	2	1.17%
Intermediate	12	7.02%	57	33.33%	77	45.03%	15	8.77%	8	4.68%	2	1.17%
Advanced	23	13.45%	60	35.09%	57	33.33%	17	9.94%	12	7.02%	2	1.17%

Chart 6.2.2-102: Beginner

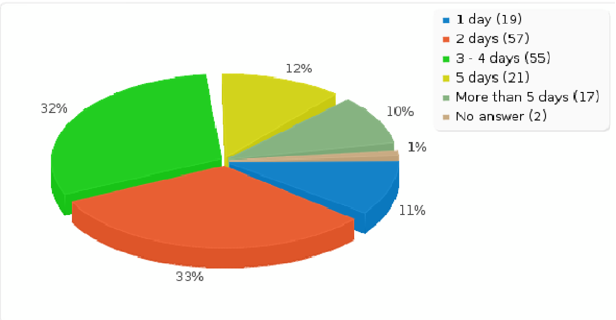


Chart 6.2.2-103: Intermediate

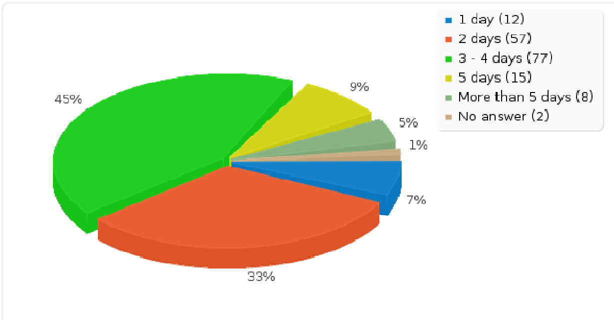
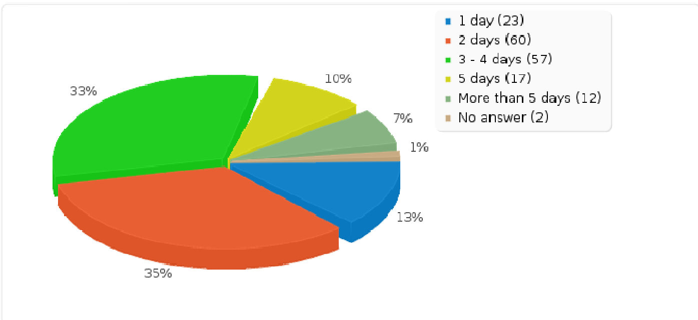


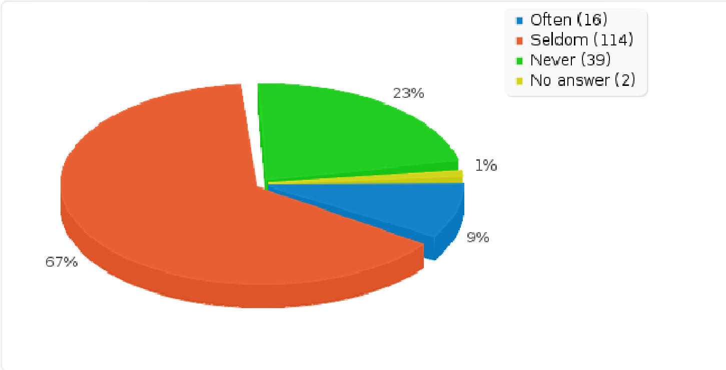
Chart 6.2.2-104: Advanced



29. Have you ever been required to teach something for which you did not feel completely comfortable teaching? *

Answer	Count	Percentage
Often	16	9.36%
Seldom	114	66.67%
Never	39	22.81%
No answer	2	1.17%

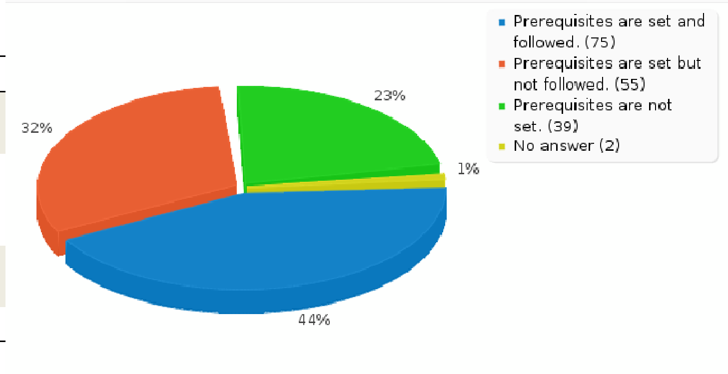
Chart 6.2.2-105



30. Are prerequisites set for classes you teach? *

Answer	Count	Percentage
Prerequisites are set and followed.	75	43.86%
Prerequisites are set but not followed	55	32.16%
Prerequisites are not set	39	22.81%
No answer	2	1.17%

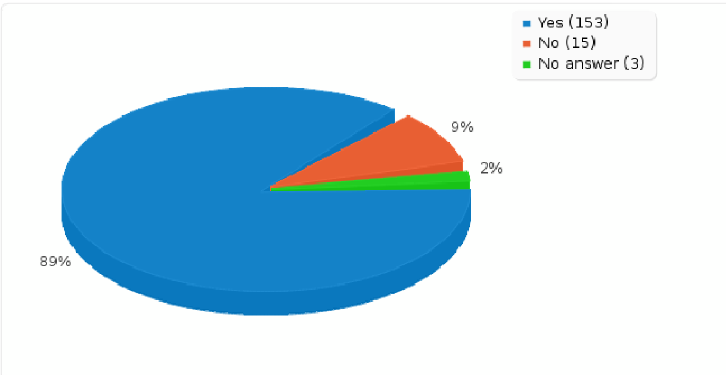
Chart 6.2.2-106



31. Do you feel comfortable with teaching in English? *

Answer	Count	Percentage
Yes	153	89.47%
No	15	8.77%
No answer	3	1.75%

Chart 6.2.2-107



32. With regards to improving as a trainer, please assess how useful training courses on the following will be for you. *

	Not useful		Somewhat useful		Very useful		No answer	
Training methodologies	20	11.70%	87	50.88%	61	35.67%	3	1.75%
Creating effective slides/handouts	33	19.30%	90	52.63%	45	26.32%	3	1.75%
Improving oratory skills	36	21.05%	81	47.37%	51	29.82%	3	1.75%
Organizing training events	51	29.82%	91	53.22%	25	14.62%	4	2.34%
English technical presentation skills	49	28.65%	70	40.94%	49	28.65%	3	1.75%

Chart 6.2.2-108: Training methodologies

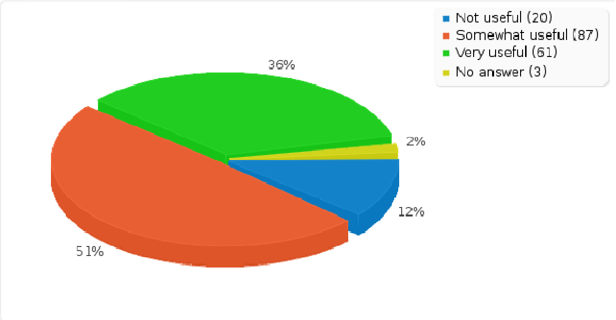


Chart 6.2.2-109: Creating effective slides/handouts

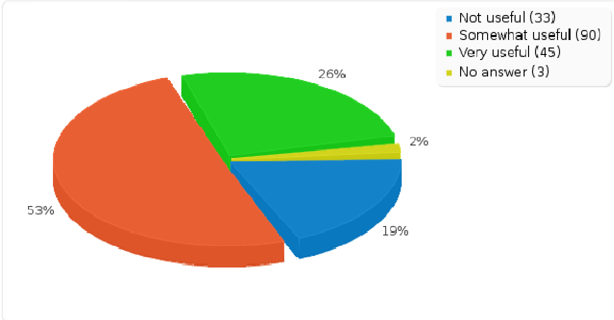


Chart 6.2.2-110: Improving oratory skills

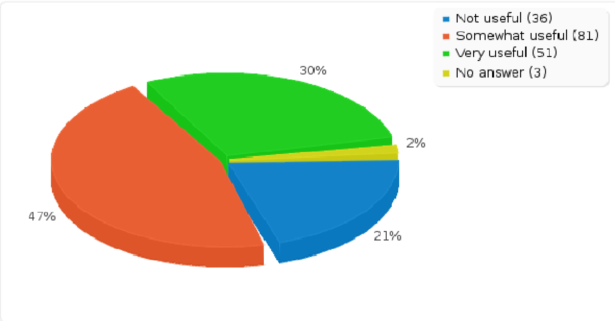


Chart 6.2.2-111: Organizing training events

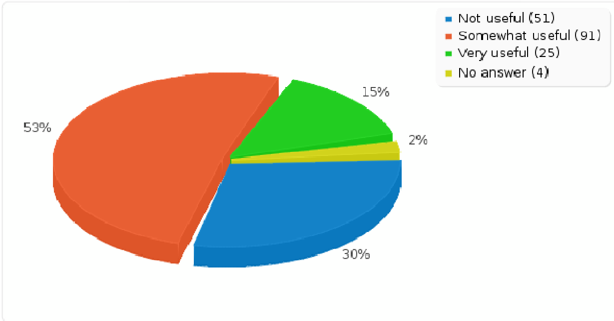
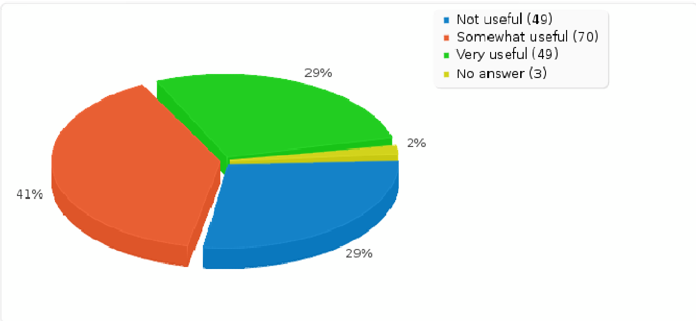


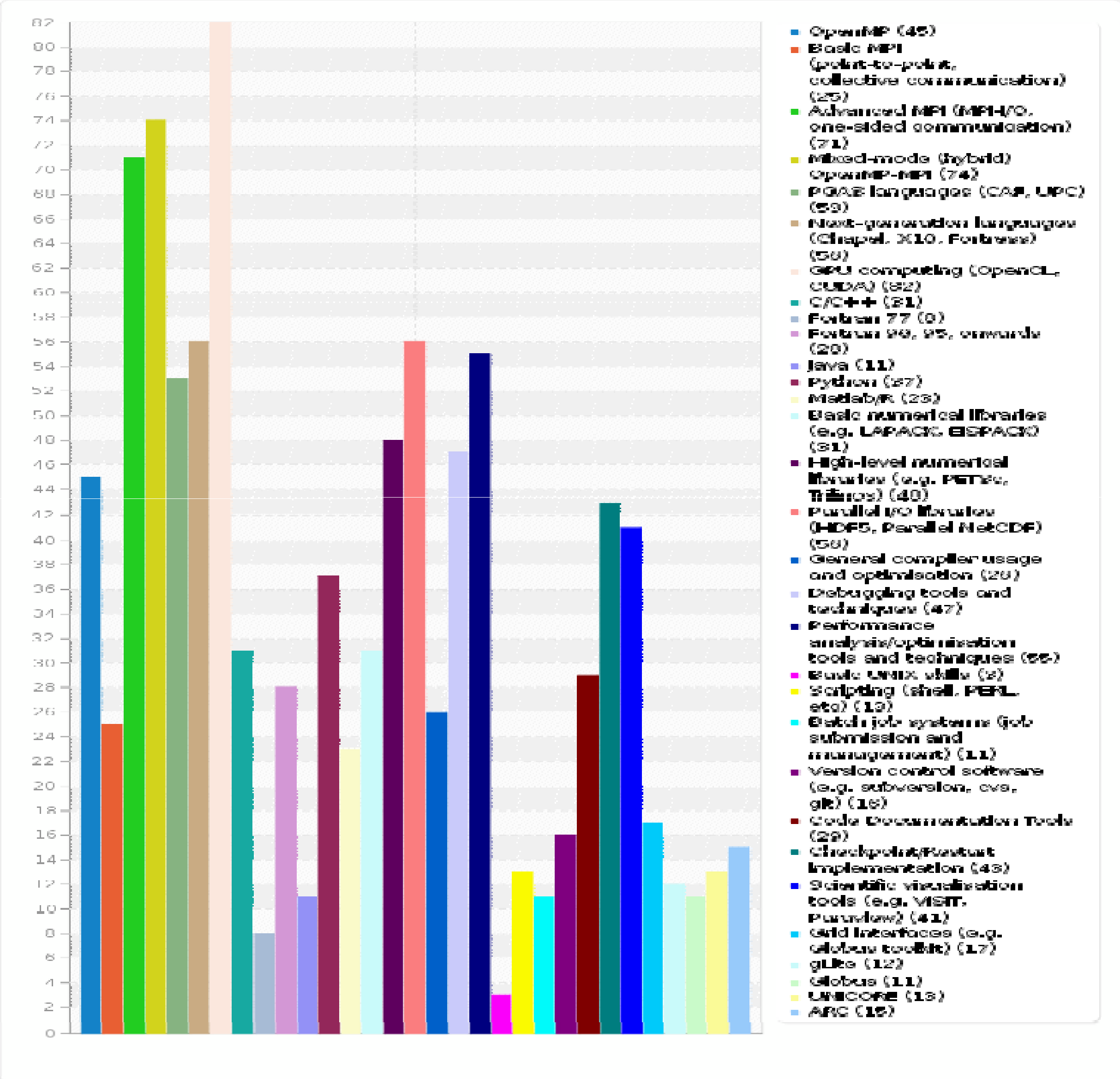
Chart 6.2.2-112: English technical presentation skills



33. From the following, are there any HPC subjects on which you would like to attend courses to improve as a trainer?

Answer	Count	Percentage
OpenMP	45	19.23%
Basic MPI (point-to-point, collective communication)	25	10.68%
Advanced MPI (MPI-I/O, one-sided communication)	71	30.34%
Mixed-mode (hybrid) OpenMP-MPI	74	31.62%
PGAS languages (CAF, UPC)	53	22.65%
Next-generation languages (Chapel, X10, Fortress)	56	23.93%
GPU computing (OpenCL, CUDA)	82	35.04%
C/C++	31	13.25%
Fortran 77	8	3.42%
Fortran 90, 95, onwards	28	11.97%
Java	11	4.70%
Python	37	15.81%
Matlab/R	23	9.83%
Basic numerical libraries (e.g. LAPACK, EISPACK)	31	13.25%
High-level numerical libraries (e.g. PETSc, Trilinos)	48	20.51%
Parallel I/O libraries (HDF5, Parallel NetCDF)	56	23.93%
General compiler usage and optimisation	26	11.11%
Debugging tools and techniques	47	20.09%
Performance analysis/optimisation tools and techniques	55	23.50%
Basic UNIX skills	3	1.28%
Scripting (shell, PERL, etc)	13	5.56%
Batch job systems (job submission and management)	11	4.70%
Version control software (e.g. subversion, cvs, git)	16	6.84%
Code Documentation Tools	29	12.39%
Checkpoint/Restart implementation	43	18.38%
Scientific visualisation tools (e.g. VISIT, Paraview)	41	17.52%
Grid interfaces (e.g. Globus toolkit)	17	7.26%
gLite	12	5.13%
Globus	11	4.70%
UNICORE	13	5.56%
ARC	15	6.41%

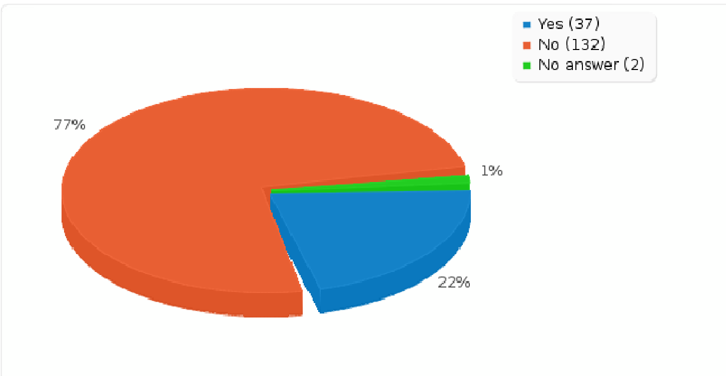
Chart 6.2.2-113



34. Have you been involved as a trainer in previous PRACE trainings? *

Chart 6.2.2-114

Answer	Count	Percentage
Yes	37	21.64%
No	132	77.19%
No answer	2	1.17%



35. Do you agree to be invited to future PRACE trainings to teach the topic of your expertise? *

Answer	Count	Percentage
Yes	132	77.19%
No	37	21.64%
No answer	2	1.17%

Chart 6.2.2-115