

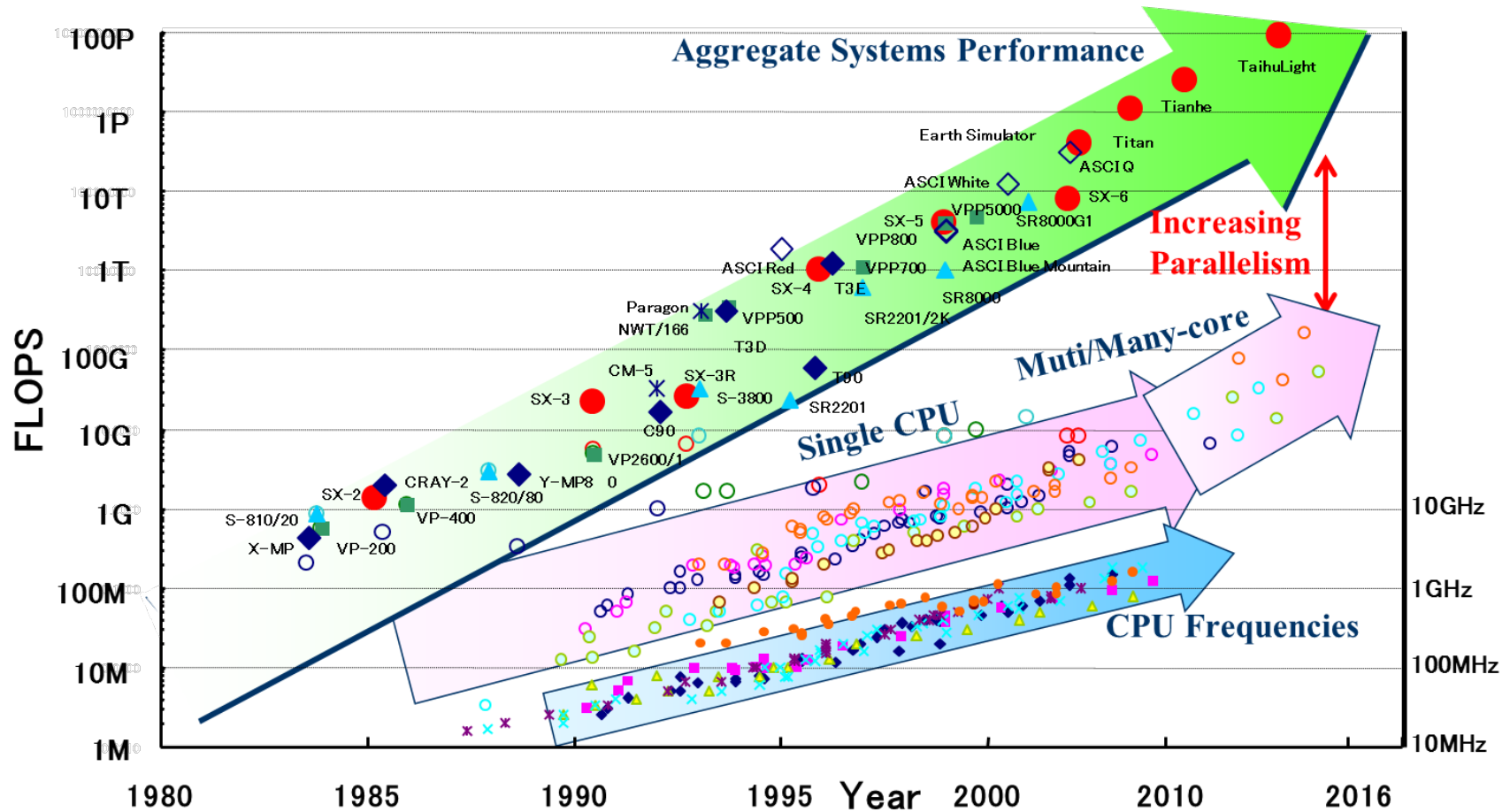
Performance Optimization on High Performance Computers

Guangming Tan

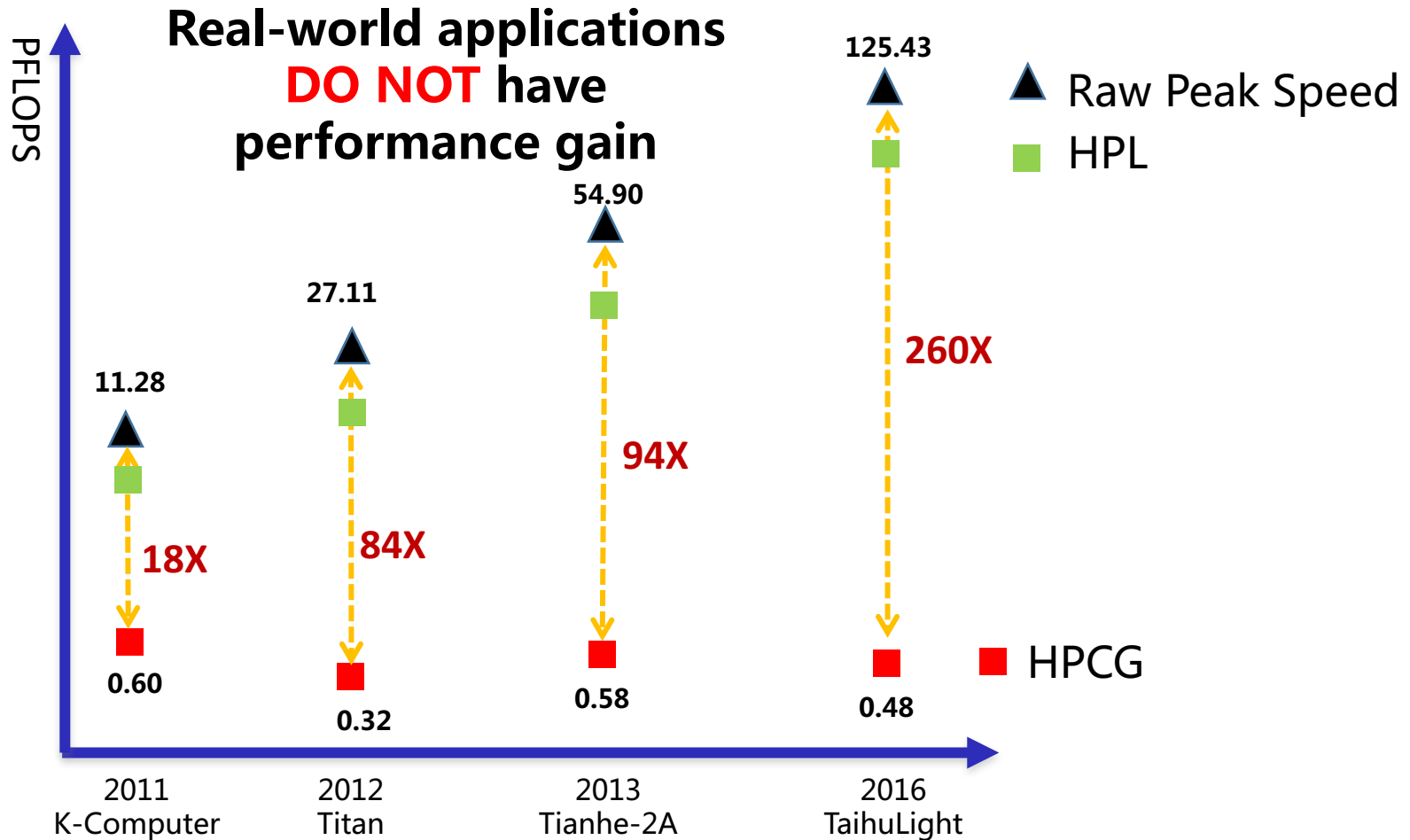
Institute of Computing Technology, Chinese Academy of Sciences

2018.5.29

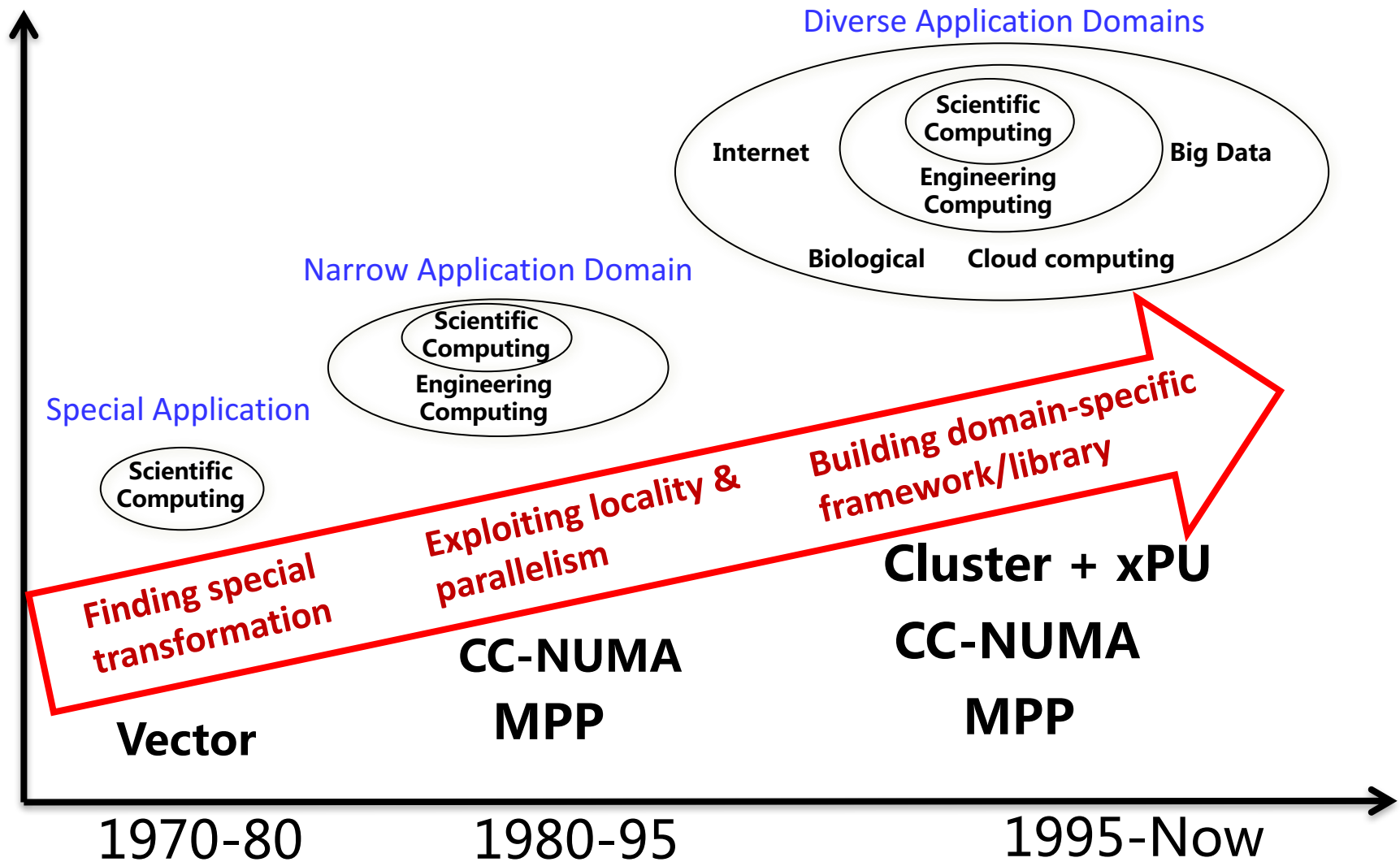
High Parallelism = High Speed



High Speed ? High Performance



A Retrospective to HPC Optimization



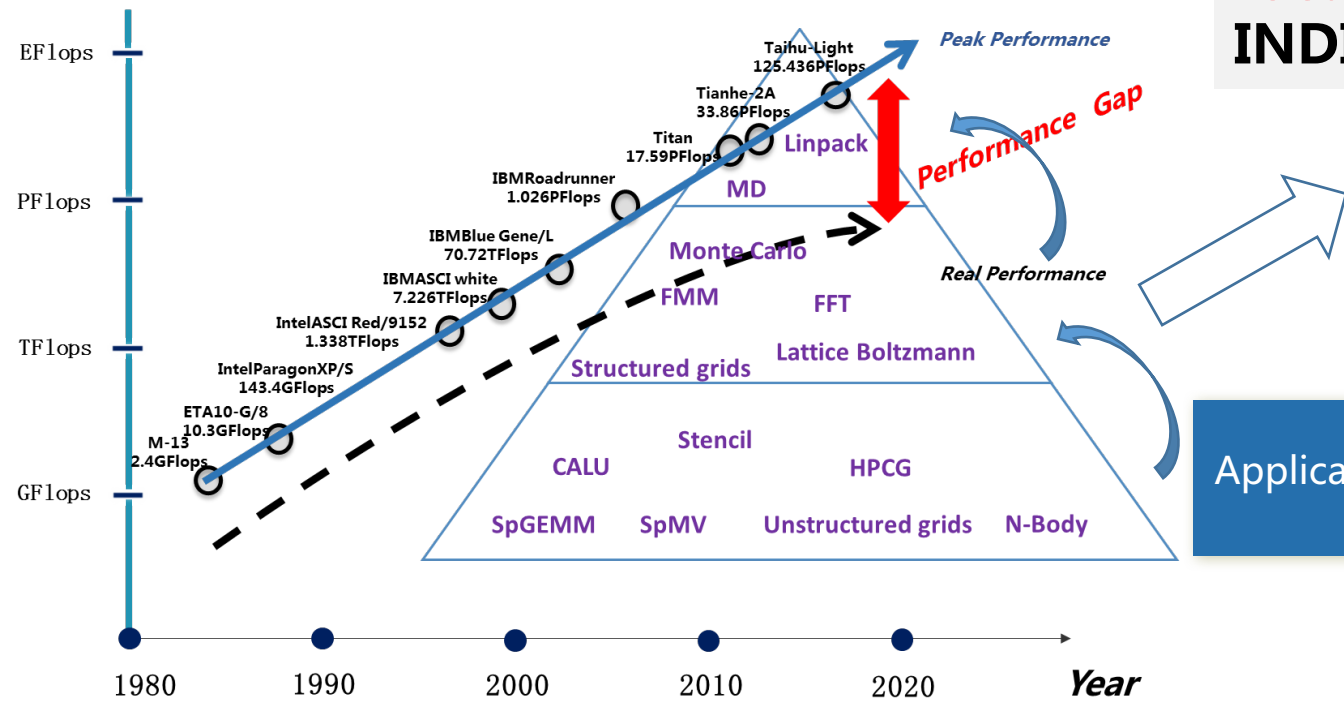
From Hero to Masses: A System Perspective is Desired

System

More computing units
Slow memory access

A system perspective
way to improve **MOST**
of application, Not only
focusing on
INDIVIDUAL application

Performance

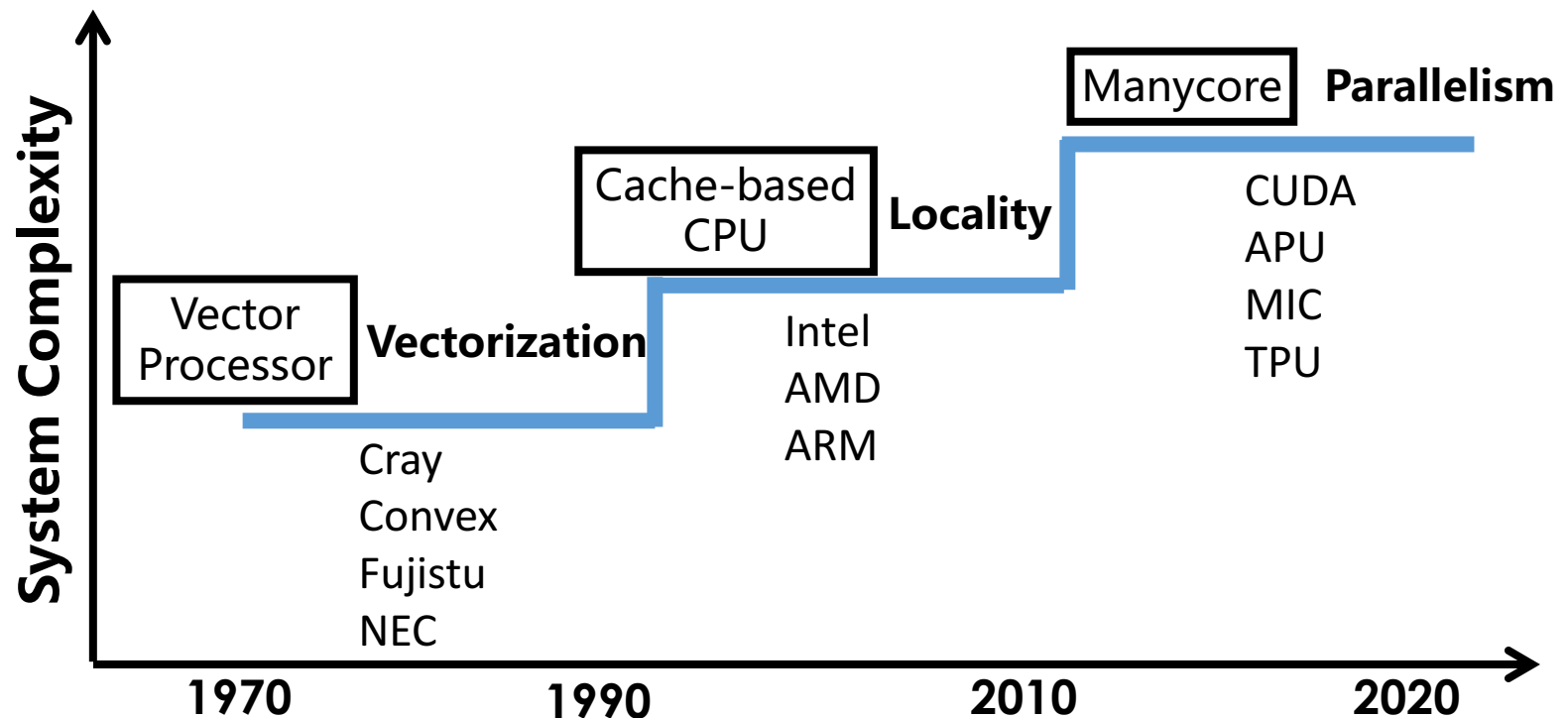


Application

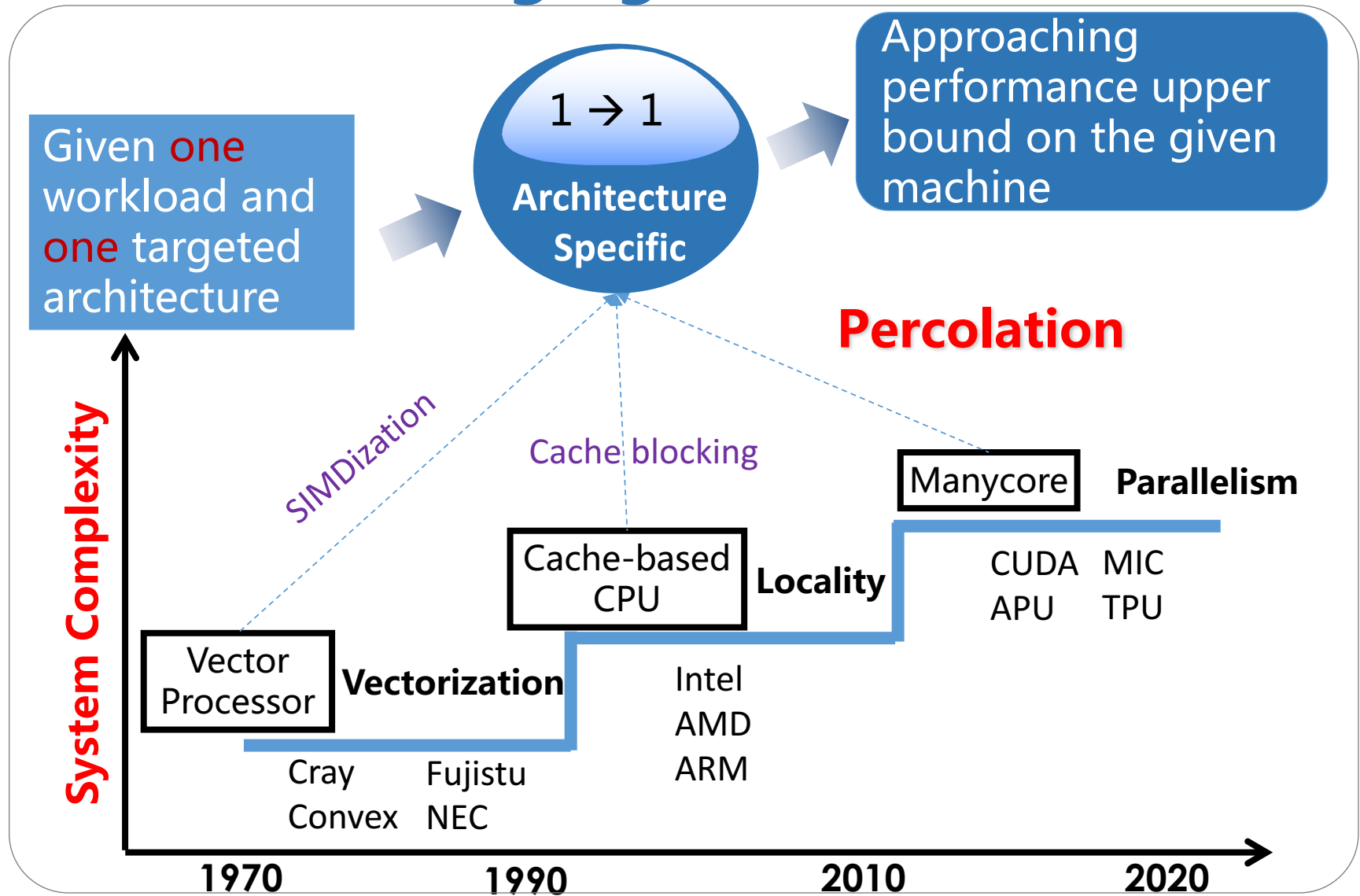
More data
Diverse access pattern

Architecture-Driven Optimization Methodology

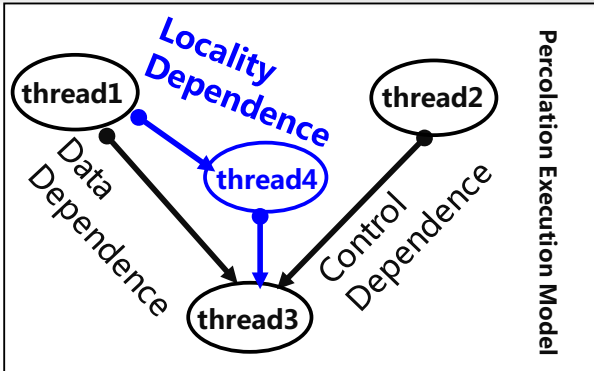
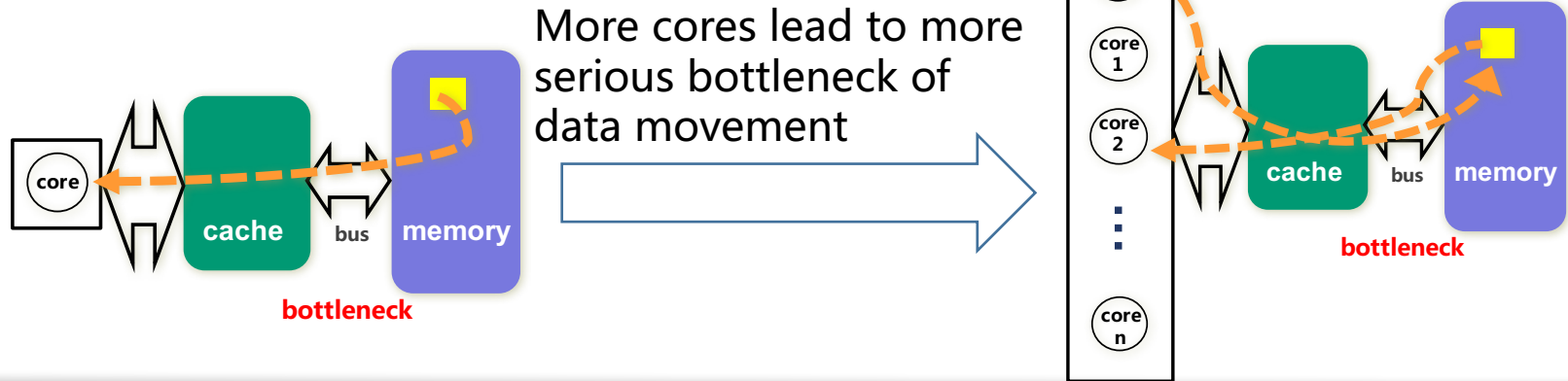
Promoting application performance by improving
vectorization, locality and **parallelism**



Issue 1 – Performance Bound on Emerging Architecture



Percolation Execution Model

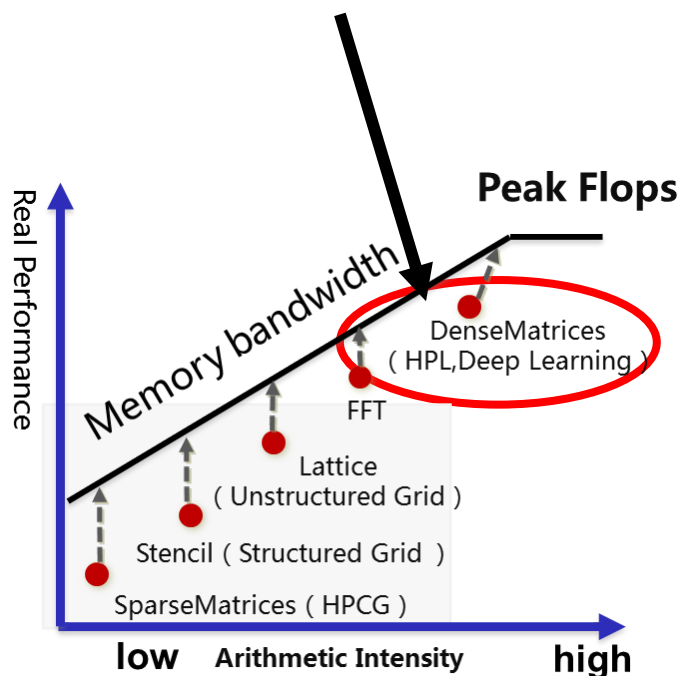


Guangming Tan, Ninghui Sun, and Guang R. Gao, Improving Performance of Dynamic Programming via Parallelism and Locality on Multicore Architecture, IEEE Transactions on Parallel and Distributed Systems, 2009.

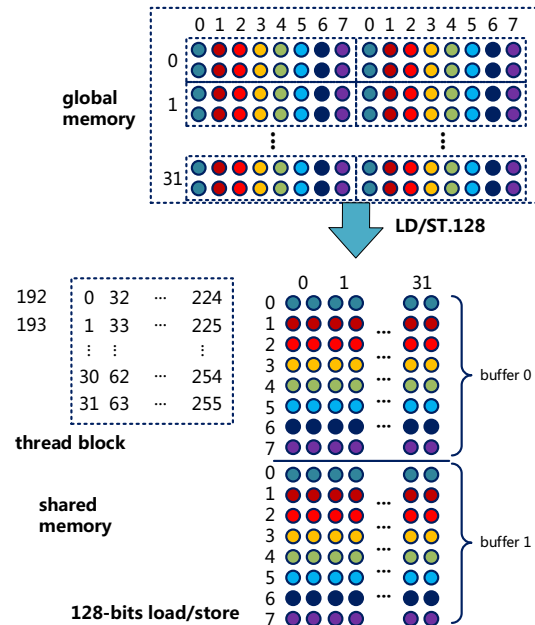
- The massive parallelism is leveraged to create just-in-time locality, where on-demand data movement happens in parallel
 - Transform static dependence to dynamic one
 - Couple dynamic parallelism with data movement

Applying to Numerical Computing Kernels

- The best algorithm that may approach machine's raw performance
- The kernel determining deep learning's performance



Matrices computation



- ✓ generating the optimal instruction execution order
- ✓ hiding data movement by multiple pipelines

Improving Performance of Math Library on Dawning6000 Supercomputer

Guangming Tan, Linchuan Li, Sean Triechler, Everett Phillips,
Yungang Bao, Ninghui Sun, Fast Implementation of DGEMM
on Fermi GPU, ACM/IEEE Supercomputing (SC), 2011

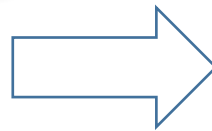
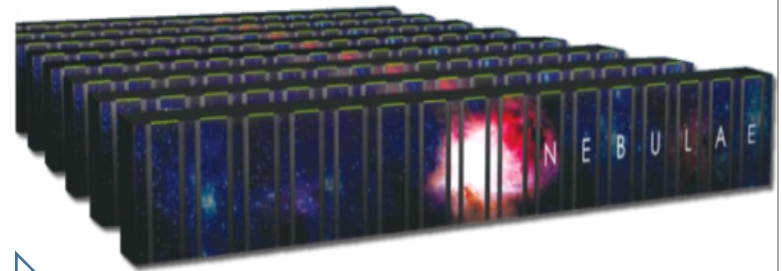
Ours	362GFlops
CUBLAS4.0 *	160GFlops

*NVIDIA GPU

Jiajia Li, Xingjian Li, Guangming Tan, Mingyu Chen, Ninghui
Sun, An Optimized Large-Scale Hybrid DGEMM Design for
CPUs and ATI GPUs, The 26th ACM International
Conference on Supercomputing (ICS), pp.377-386, 2012.

Ours	758GFlops
ACML-GPU1.0 *	392GFlops

*AMD GPU

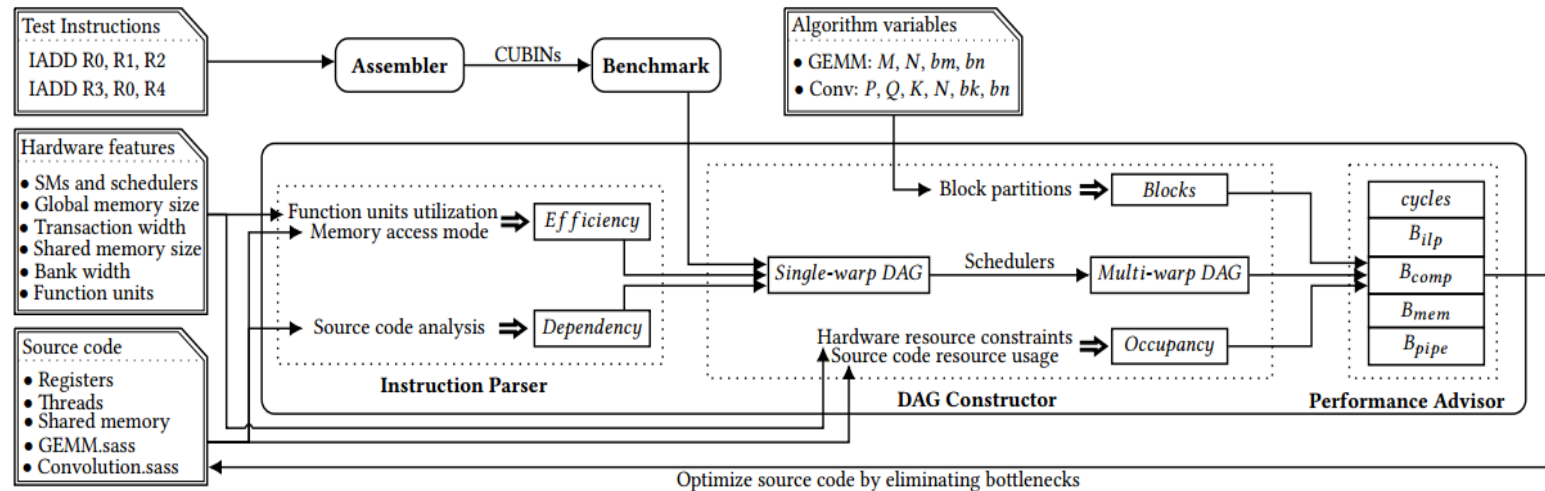


Dawning 6000 (Nebulae)

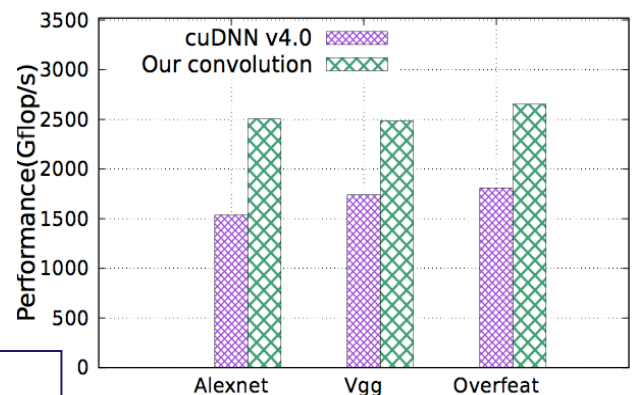
No.2, Top500, 2010.6
Shenzhen Supercomputing Center

Improving Performance of Deep Learning Library

<https://github.com/PAA-NCIC/DeepPerf>



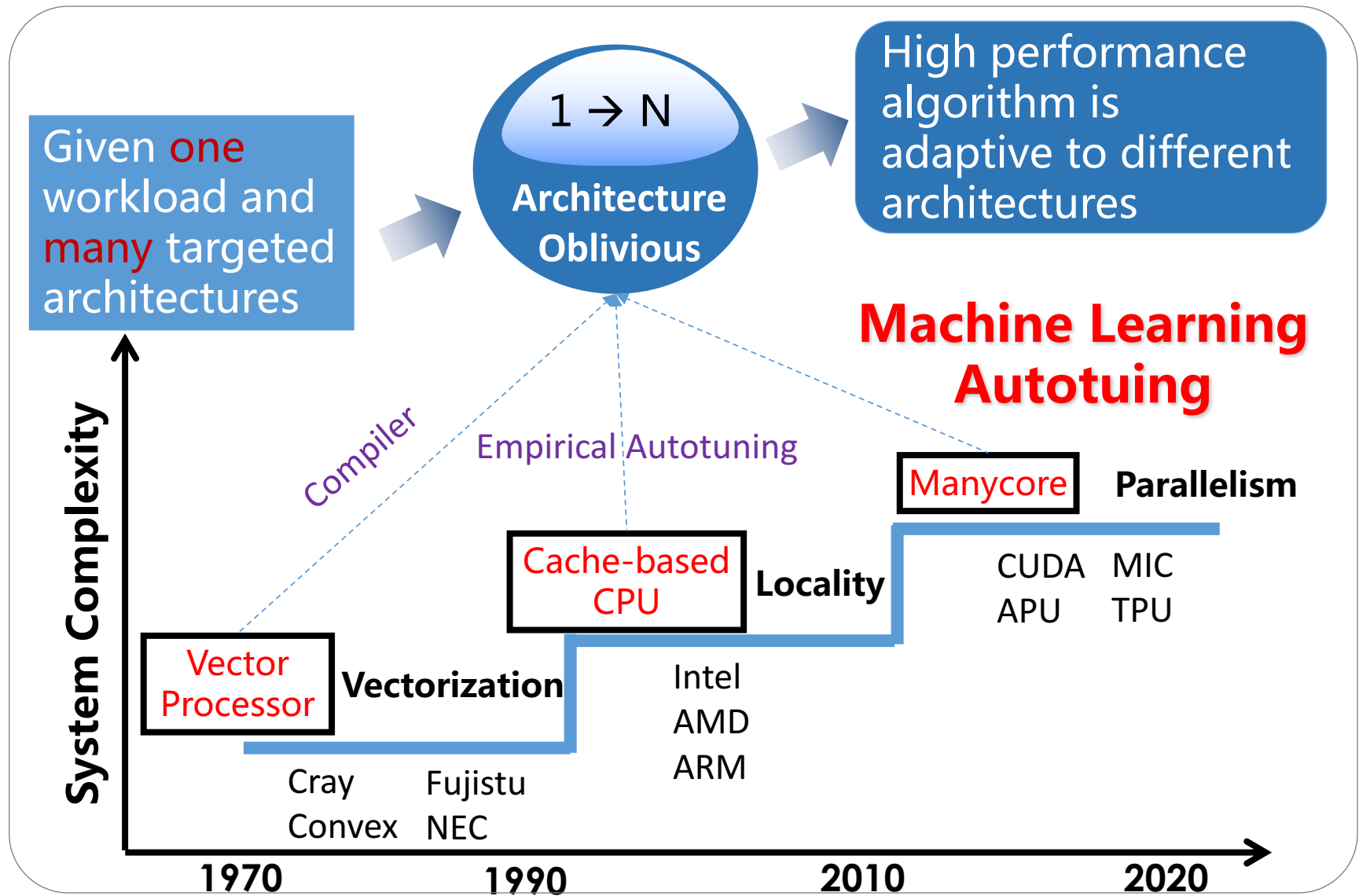
- ✓ A performance tuning framework to explore GPU micro-architectural features
- ✓ fully utilize fused multiply-add (FMA) dual-issue mechanism



20-60% higher than CuDNN

Xiuxia Zhang, Guangming Tan, Shuangbai Xue, Jiajia Li, Keren Zhou, Mingyu Chen, Understanding the GPU Microarchitecture to Achieve Bare-Metal Performance Tuning. ACM PPOPP 2017: 31-43

Issue 2 – Performance Portability on Different Architectures

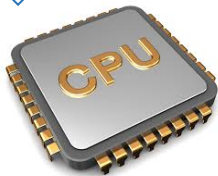


Performance Autotuning

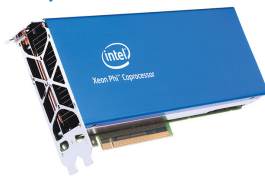
Some kernel algorithm (e.g., SpMV, Stencil)



- Pentium(MMX)
- Pentium III(SSE)
- Sandy Bridge(AVX)
- Skylake(AVX512)



- Knights Ferry
- Knights Corner
- Knights Landing
- Knights Mill



- Fermi
- Kepler
- Maxwell
- Pascal



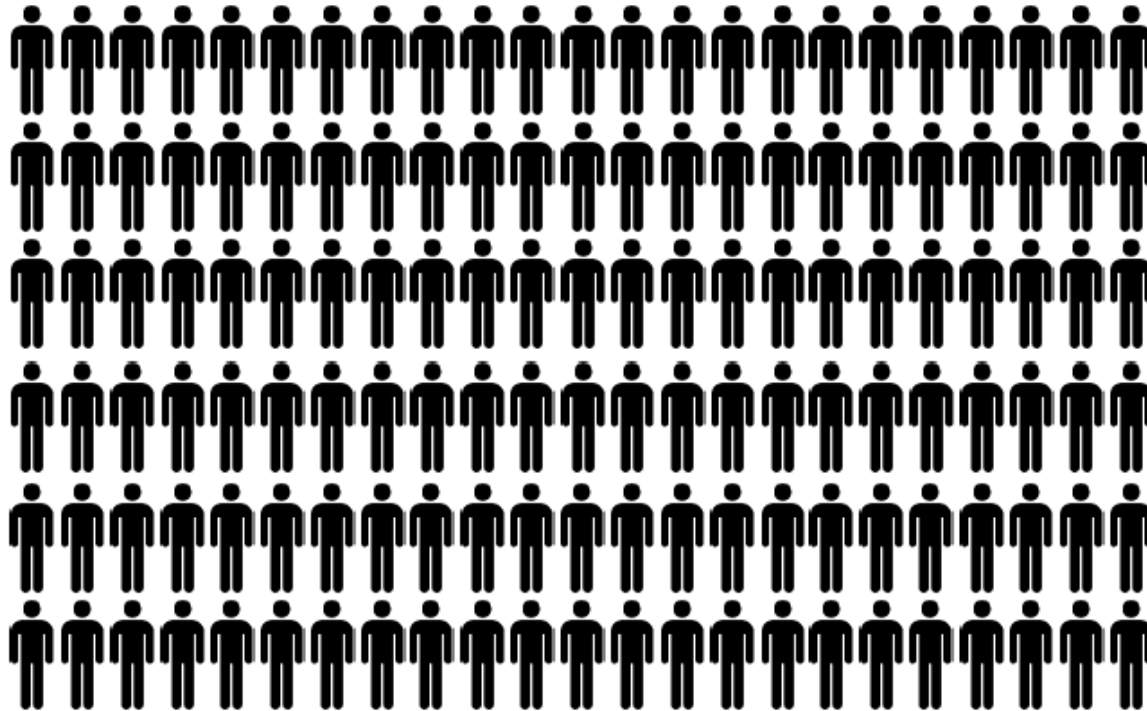
...

Autotuning
Library
with **ONE**
interface



**N implementations of performance
optimization by hand**

Performance Autotuning



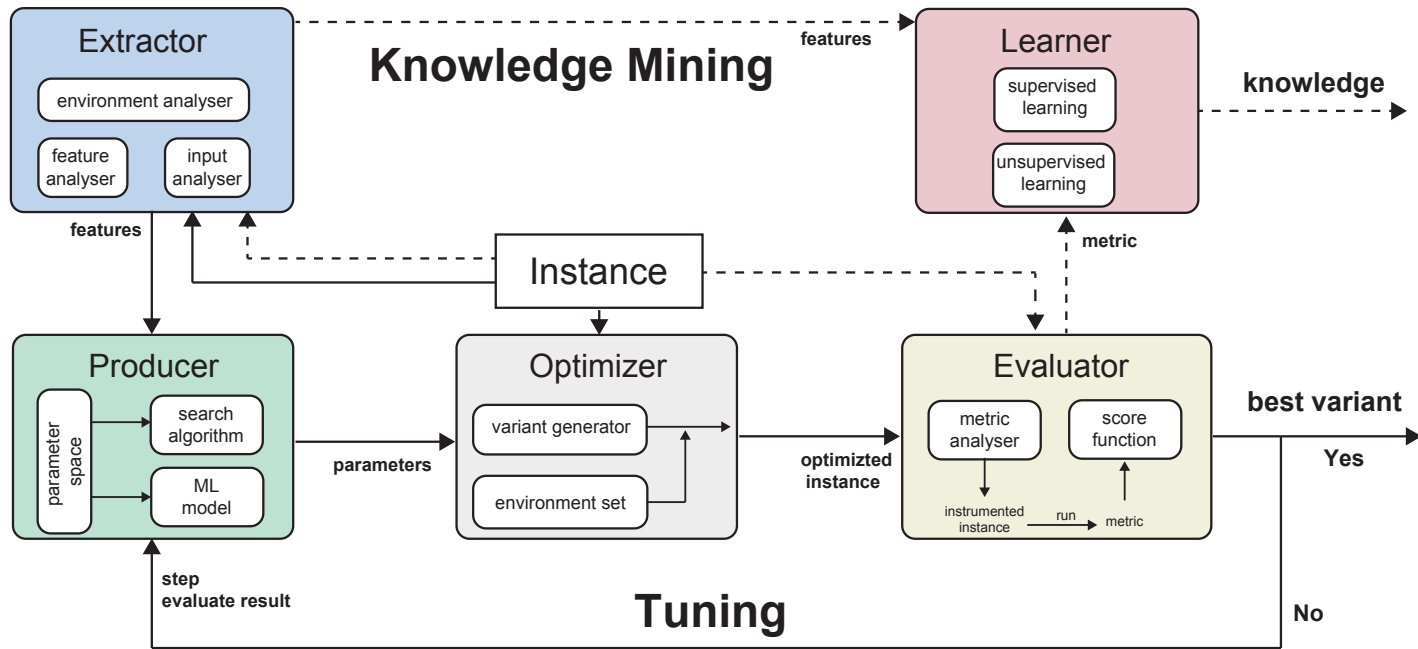
N implementations of performance optimization by hand

**Autotuning
Library
with ONE
interface**



Machine Learning Based Autotuning

<https://github.com/PAA-NCIC/PAK>



- framework to provide basic tools to build autotuning library with machine learning
- Composable & Resuable for 5 modules

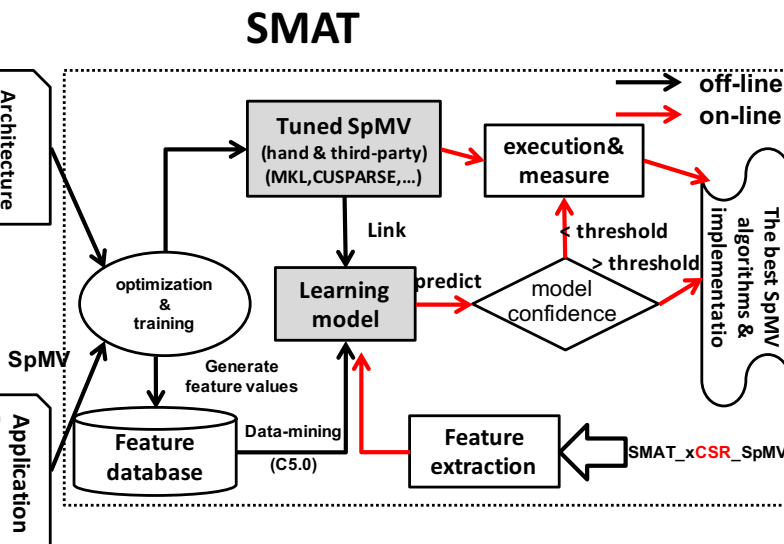
Case study : SpMV Autotuning

Example

- TLB
- Cache
- Register
- prefetch
- SIMDize
- Branch
- multithreading

Example

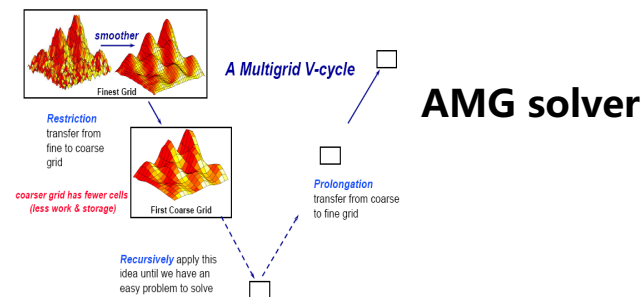
- Dimension
- Dialog
- #Non-zero
- Distribution of non-zero
- Power-law



J. Li, G. Tan, M. Chen, N. Sun, SMAT: An Input Adaptive Auto-Tuner for Sparse Matrix-Vector Multiplication, ACM SIGPLAN conference on Programming Language Design and Implementation (PLDI), 2013

Ours	32GFlops
MKL	10GFlops

- ✓ Extracting both application features and architecture features with data mining
- ✓ Learning a predict model to generate optimal codes



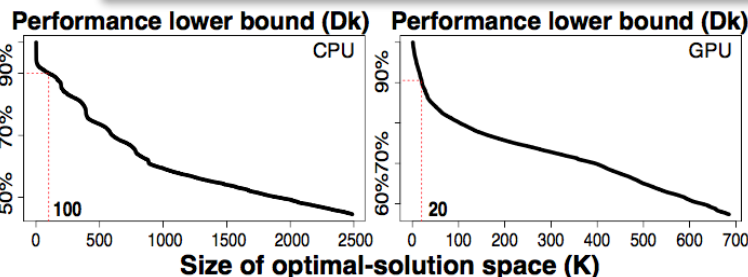
Reduce execution time by 20%

Case study: Stencil Autotuning

Explosive Search Space

PATUS	8 hours
SDSL	> 33 hours
PARTANS	2.5h-1month
Halide	2h-2days

Optimal Space Solution (OSS) Model



$$OSS^K = \{v | M(v) \geq M(v_K^*), v, v_K^* \in R\}$$

The optimal variant
contained in a small
OSS

Two running instances
have the same optimal
variant if their
features are similar

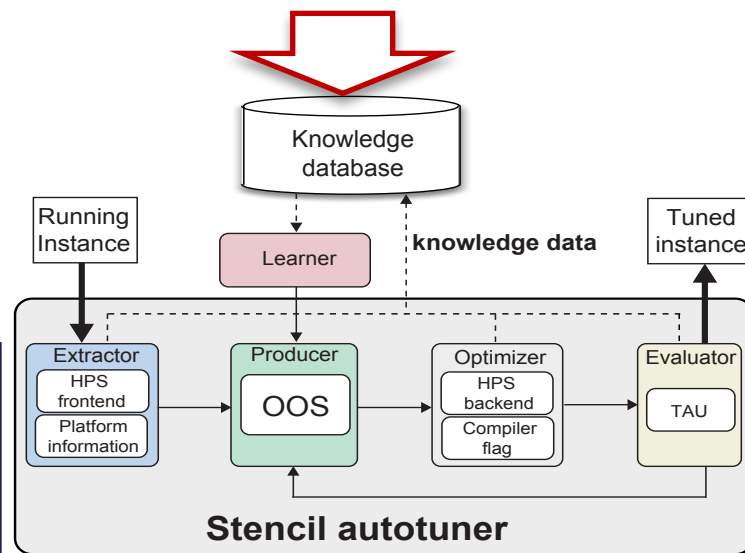
$$OR = \frac{|OSS_x^K \cap OSS_y^K|}{K} \rightarrow OR^* = \underset{x}{\operatorname{argmax}} q(y|\vec{x})$$

Algorithm Similarity → Reduce Search Time

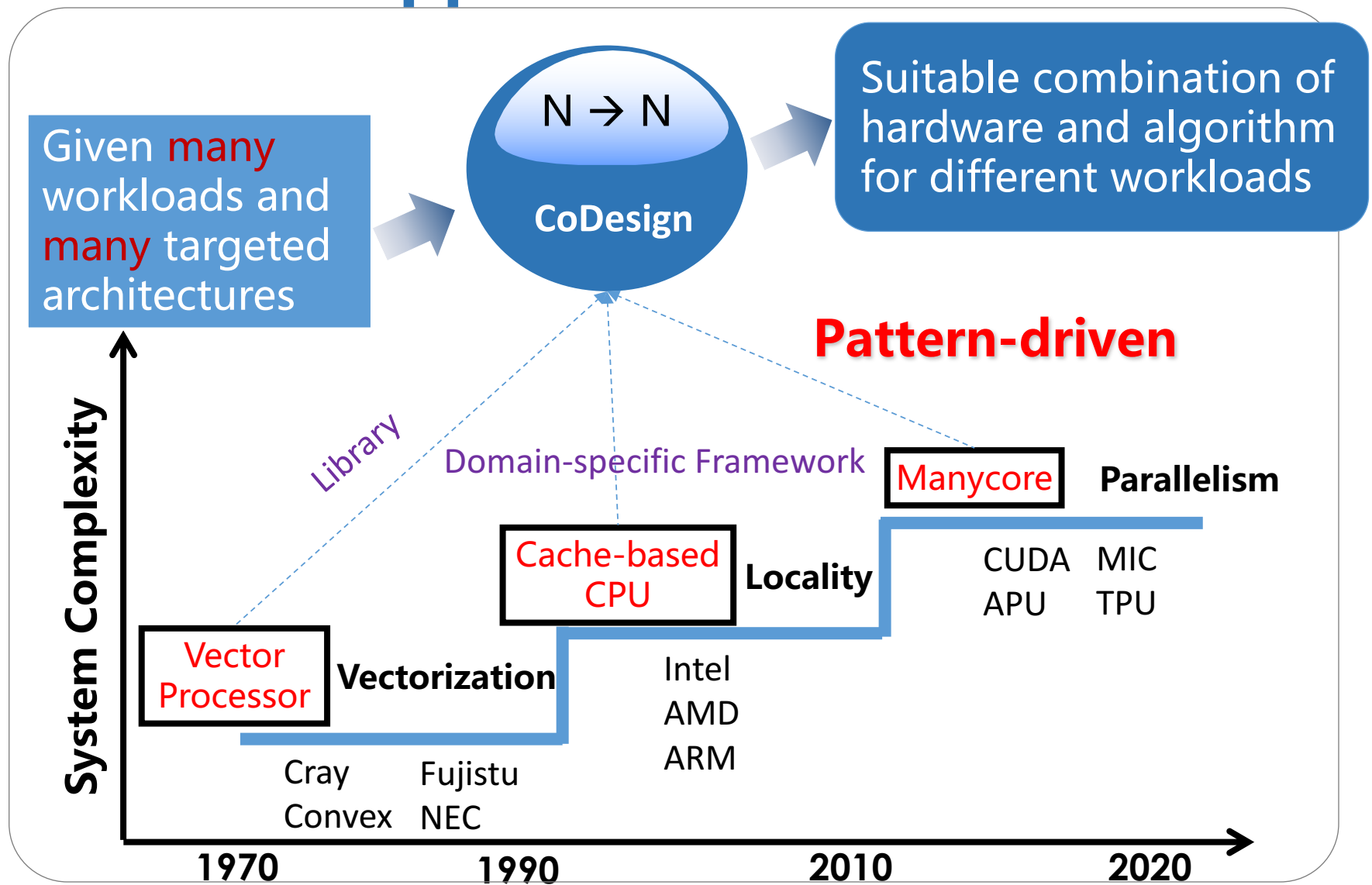
Autotuning Speed

Ours	9 steps
PATUS, SDSL	2725 steps

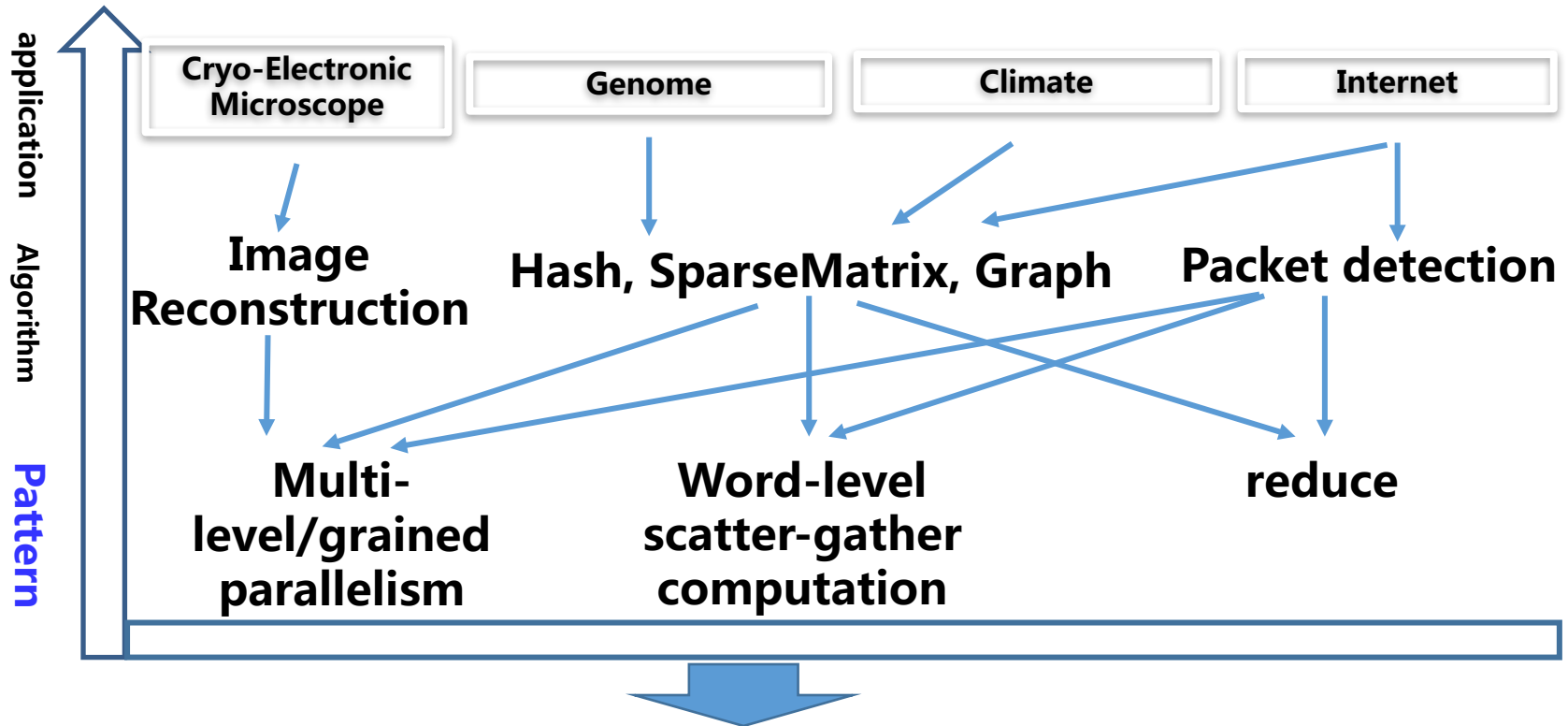
Yulong Luo, Guangming Tan, Zeyao Mo, Ninghui Sun,
FAST: A Fast Stencil Autotuning Framework Based On
An Optimal-solution Space Model, *Proceedings of the
29th ACM on International Conference on
Supercomputing (ICS)*, June 08 - 11, 2015.



Issue 3 – Performance Adaptive between Application and Architecture



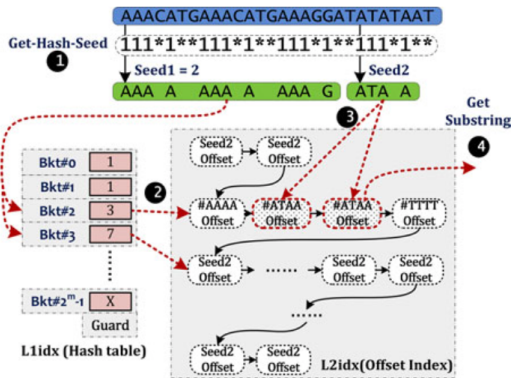
Pattern-Driven Software-Hardware Codesign



- Specific design for the abstracted patterns (**NOT for algorithms**), for example:
 - Heterogeneous system combining data-level parallelism and task-level parallelism
 - Specialized computing units for common operations
 - Customized hardware units for irregular memory access

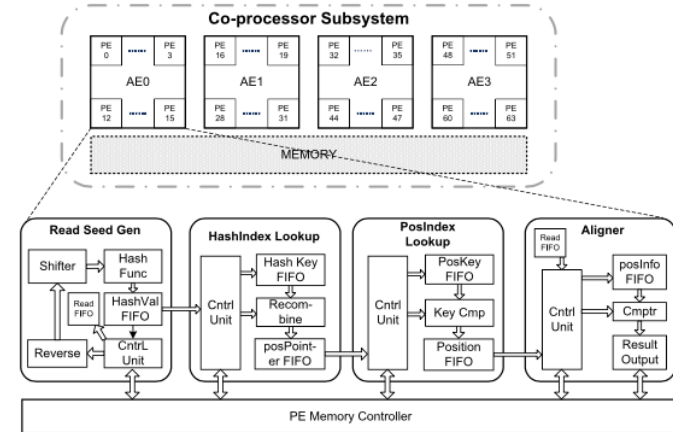
Case study: Accelerating Irregular Computation in Genomic Data Analysis

Hash index lookup

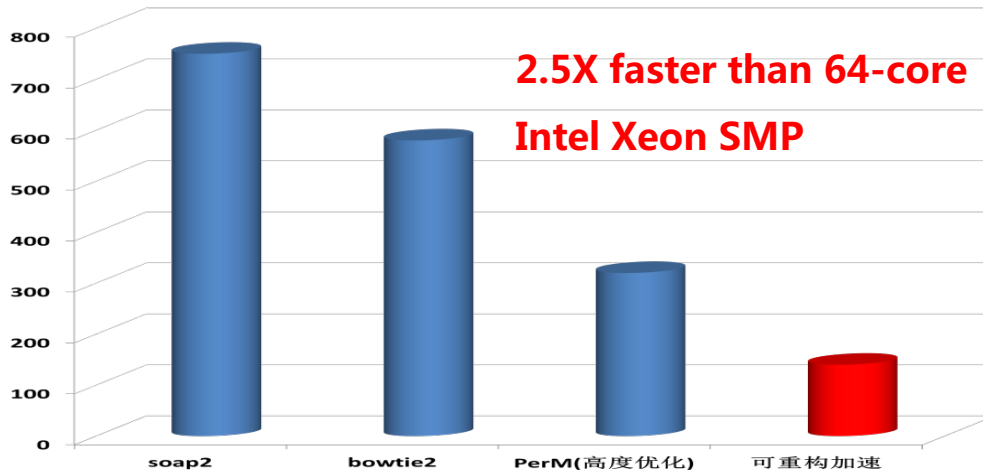


Specific Memory controller
for word-level scatter-
gather memory access

Co-processor to exploit
massive fine-grained memory
level parallelism

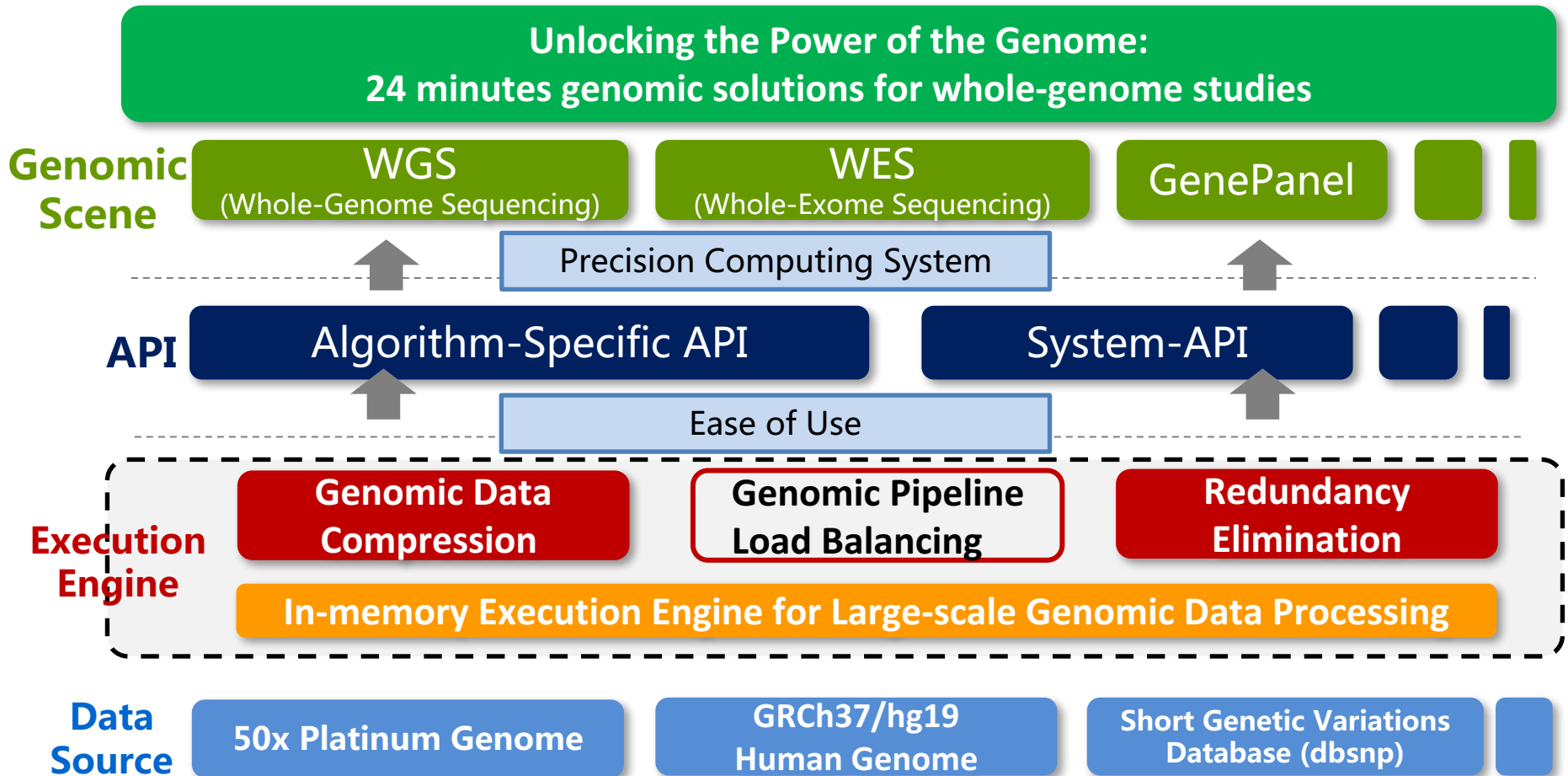


generate memory requests from
every AE to every memory
controller port on every cycle



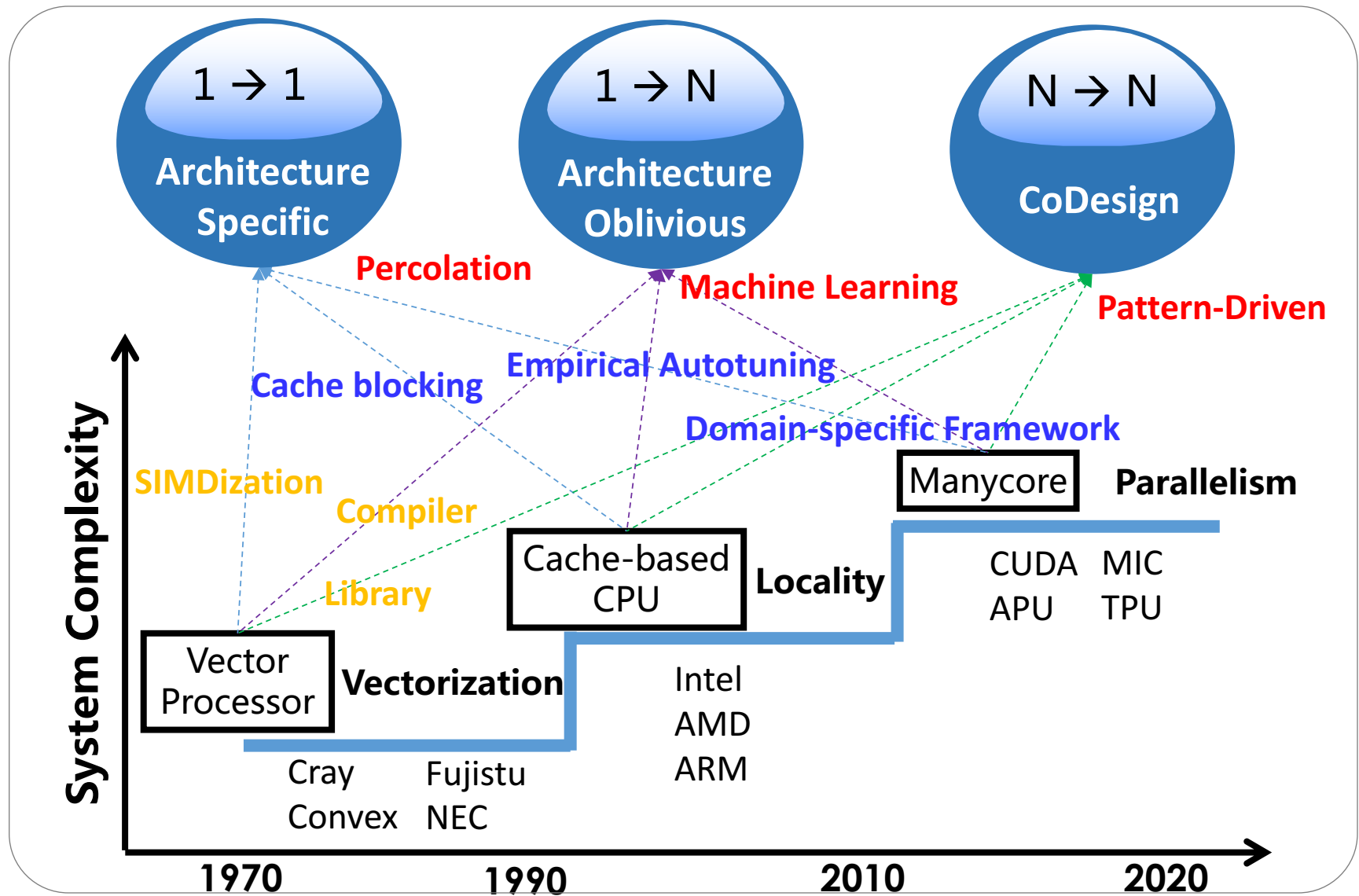
Guangming Tan, Chunming Zhang, Wen Tang, Peiheng Zhang, Ninghui Sun, Accelerating Irregular Computation in Massive Short Reads Mapping on FPGA Co-processor, IEEE Transactions on Parallel and Distributed Systems, Vol.27, No.5, 2016.

Case study: Accelerating In-Memory Computing for Genomic Data Analysis

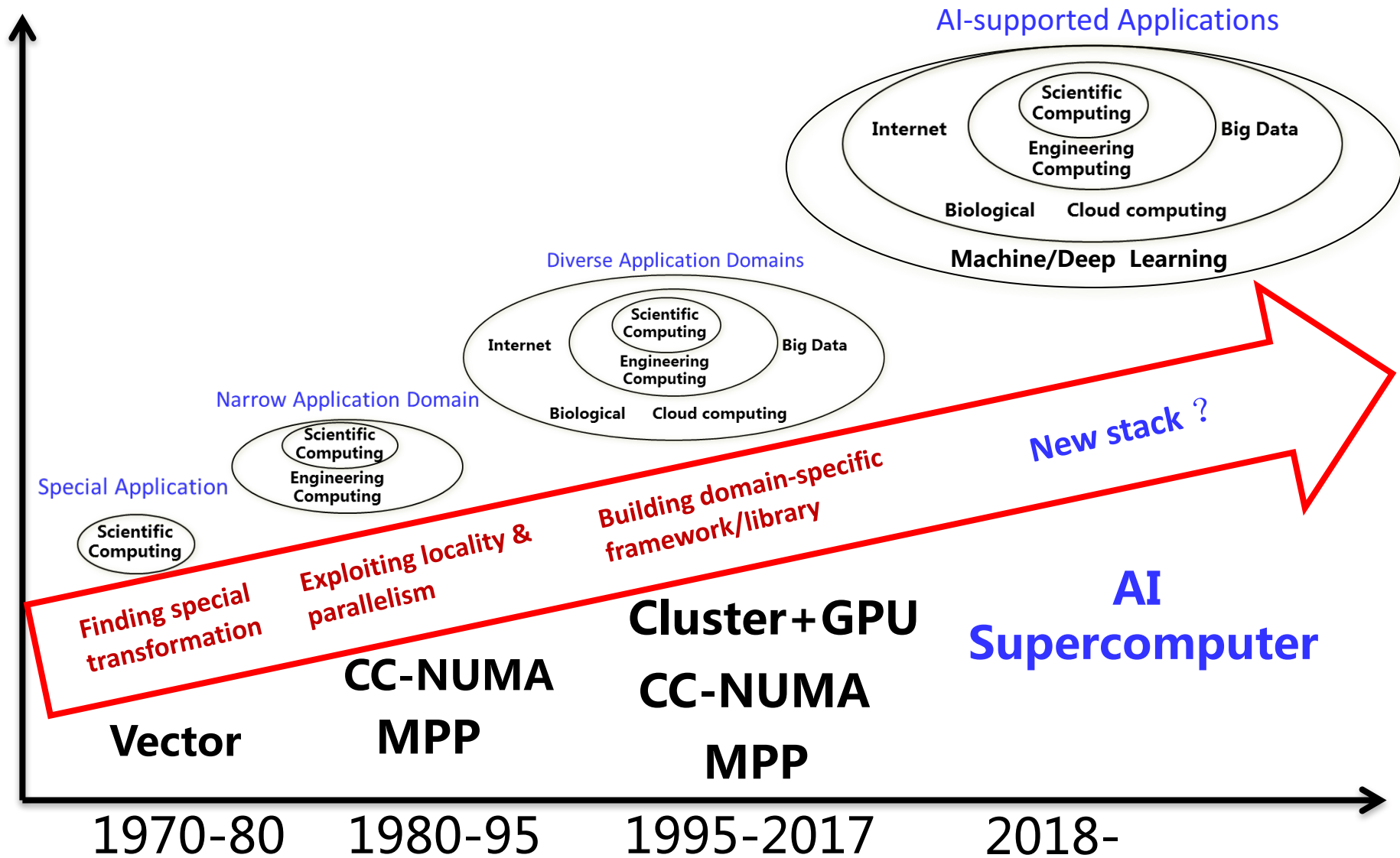


Xueqi Li, Guangming Tan, Bingchen Wang, Ninghui Sun: High-performance genomic analysis framework with in-memory computing. ACM PPOPP2018: 317-328

Wrap-up



What's Next



Optimization on AI Supercomputer



- Architecture support
 - Variable numerical precision
 - Interconnection customized for asynchronous communication
- Library support
 - Abstract programming interface
 - Adaptive to diverse accelerators
- Algorithm support
 - Exploit billion-level parallelism in model
 - Tradeoff between speed and accuracy

Thanks !