



## The DNA Deluge

A Parallel Computing Approach

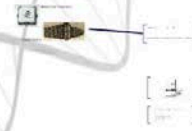
Brendan Lawlor



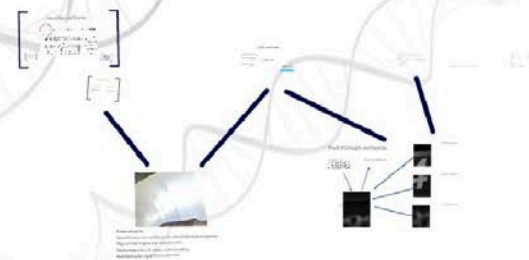
1. What is the deluge?



2. What is parallel computing?



3. How to harness parallel power?



# The DNA Deluge

A Parallel Computing Approach

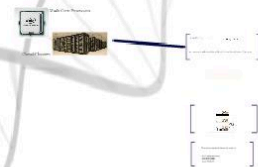
Brendan Lawlor



1. What is the deluge?



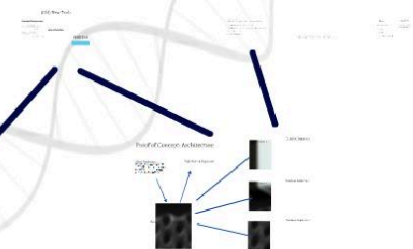
2. What is parallel computing?



3. How to harness parallel power?



Results/Insights  
Insights are the key to the system's success. They are the results of the system's operation. They are the results of the system's operation. They are the results of the system's operation.



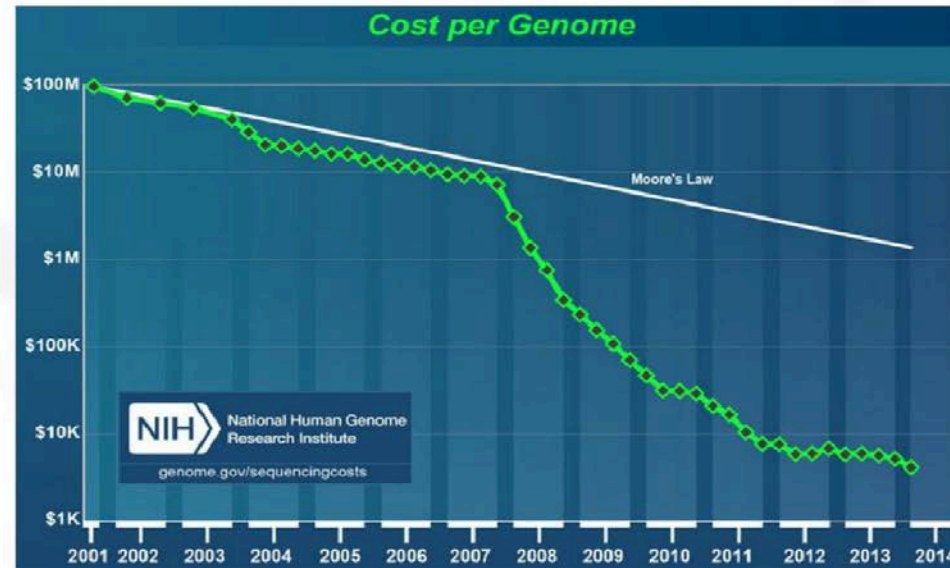
## The DNA Deluge

A Parallel Computing Approach

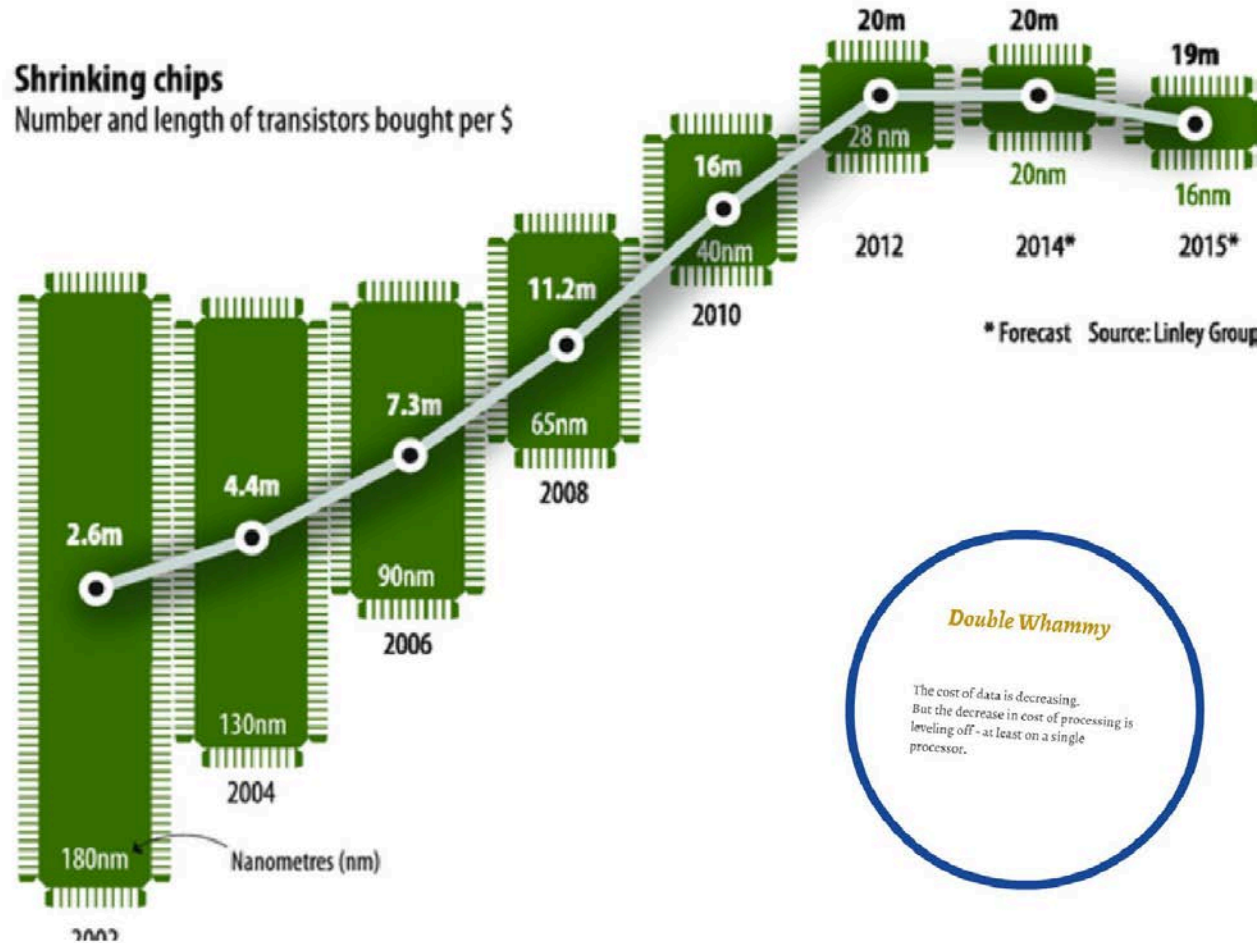
Brendan Lawlor



# The Data Deluge



# The Data Deluge



## Double Whammy

The cost of data is decreasing.  
But the decrease in cost of processing is  
leveling off - at least on a single  
processor.

## *Double Whammy*

The cost of data is decreasing.  
But the decrease in cost of processing is  
leveling off - at least on a single  
processor.

# The Data Deluge



Multi Core Processors

Cloud/Clusters





Amdahl's Law:

$$S(n) = \frac{T(1)}{T(n)} = \frac{T(1)}{T(1) \left( B + \frac{1}{n} (1 - B) \right)} = \frac{1}{B + \frac{1}{n} (1 - B)}$$

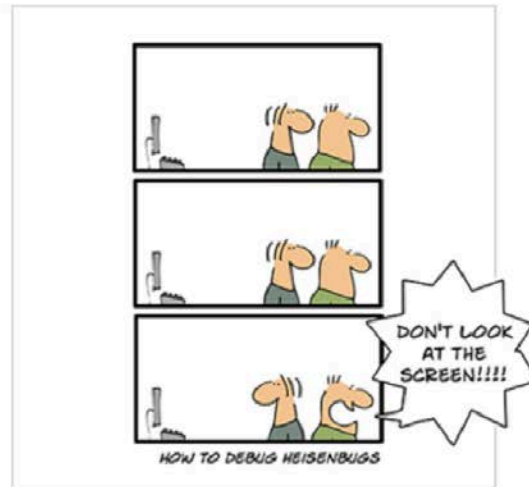
or... you can't make something faster than its slowest (= serialized) part.

*Multithreaded programming*



*Multithreaded programming*





FROM THE VOICES OF RESEARCHERS

We are going to need tools that are fit for purpose.

Functional Programming  
Actor Architecture  
Reactive Streams

## *The Smith-Waterman Algorithm*

In bioinformatics, a **sequence alignment** is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of **functional, structural, or evolutionary relationships** between the sequences.

The diagram illustrates the central dogma of molecular biology. At the top, a red circular arrow labeled "Replication" points from DNA back to DNA. Below this, a linear flow is shown: DNA is transcribed into RNA (indicated by a red arrow labeled "Transcription"), and RNA is translated into PROTEIN (indicated by a red arrow labeled "Translation").

Below the flow, a specific example is provided. The DNA sequence is shown as two strands:
   
Top strand: ... GTGCATCTGACTCCTGAGGAGAAG ...
   
Bottom strand: ... CACGTAGACTGAGGACTCCTCTTC ...
   
An arrow labeled "(transcription)" points down to the RNA sequence:
   
... GUGCAUCUGACUCUGAGGAGAAG ...
   
The RNA sequence is color-coded to match the DNA template strand (bottom strand). Brackets connect the RNA codons to the corresponding amino acids in the protein sequence:
   
... V H L T P E E K ...
   
The amino acids are also color-coded to match their respective codons. The label "protein" is at the bottom right.

[illegible]

## *The Smith-Waterman Algorithm*

S \ T		T	A	C	T	A	A
	0	1	2	3	4	5	6
S 0	0	0	0	0	0	0	0
T 1	0	1	0	0	1	0	0
A 2	0	0	2	0	0	2	1
A 3	0	0	1	1	0	1	3
T 4	0	0	0	0	2	0	1
A 5	0	0	1	0	0	3	1

## *Two Kinds of Parallel*

### *Data-Dependent*

The SW algorithm is an example of this kind: **some cells in the matrix require the results of others**. It uses intricate loops and intimate knowledge of the hardware to drive as much data as possible through a CPU in a single clock-cycle. It achieves SIMD using Intrinsics.

### *"Embarrassingly Parallel"*

Running many query sequences against many database sequences is an example of this kind. **Each run of the S-W algorithm is independent of those that go before and after.**



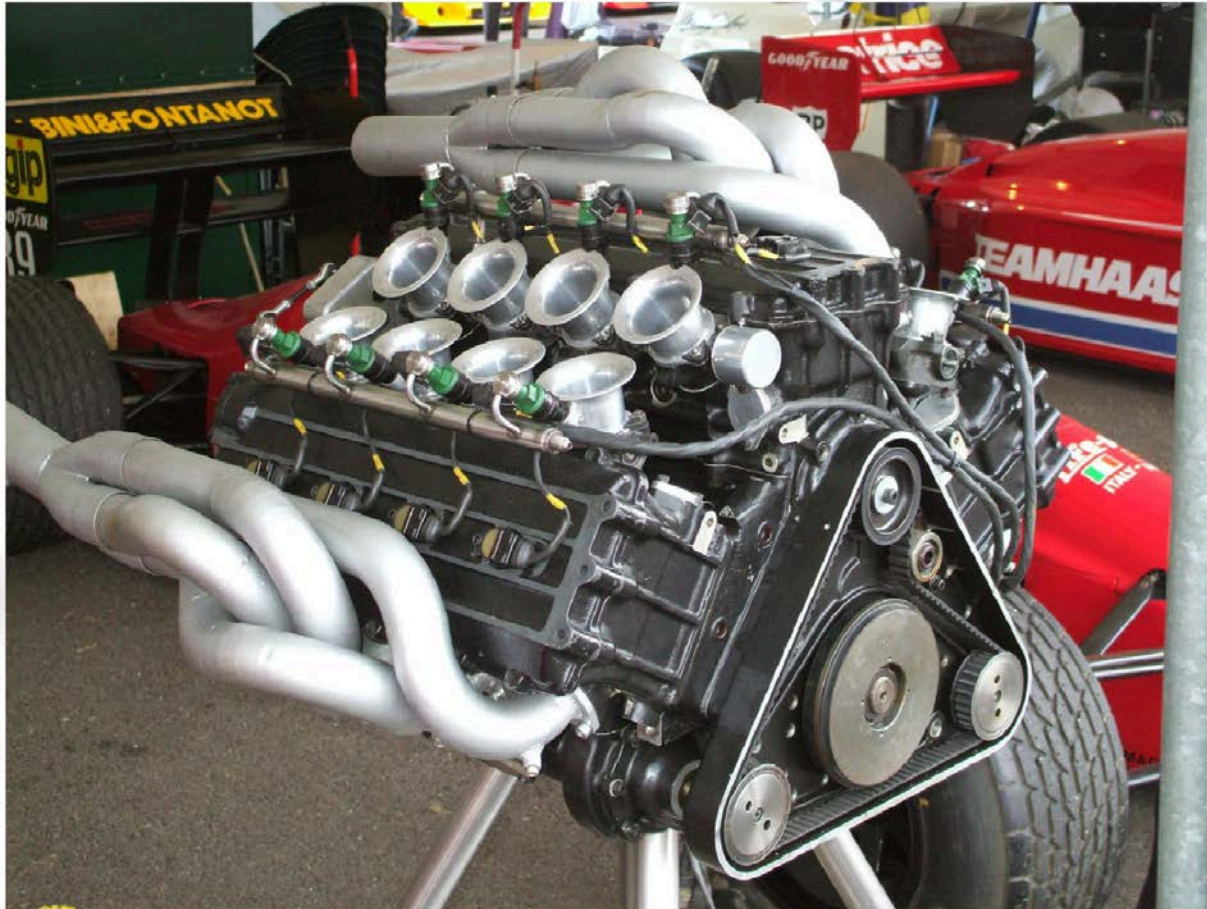
## *Data-Dependent*

The SW algorithm is an example of this kind: **some cells in the matrix require the results of others**. It uses intricate loops and intimate knowledge of the hardware to drive as much data as possible through a CPU in a single clock-cycle. It achieves SIMD using Intrinsics.

## *"Embarrassingly Parallel"*

Running many query sequences against many database sequences is an example of this kind. **Each run of the S-W algorithm is independent of those that go before and after.**

# The Data Deluge





**Create a chassis to:**

Spread the power over multiple queries and multiple database sequences

Plug in multiple engines to go faster (*scalability*)

Supply enough fuel to all engines to prevent stalling

Avoid flooding the engine with too much fuel

## (Old) New Tools

### Functional Programming

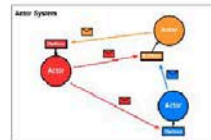
"The essence of functional programming is to concentrate on **transformations of immutable values** rather than stepwise modifications of mutable values."

- Martin Odersky, inventor of Scala programming language

In other words, with FP we can write code in an intrinsically parallel manner and let the compiler and run-time schedule on the parallel infrastructure.

- Functioing Robert Harper, Professor of Computer Science at Carnegie Mellon University

### Actor Architecture



### Reactive Streams



## Functional Programming

"The essence of functional programming is to concentrate on **transformations of immutable values** rather than stepwise modifications of mutable values."

*- Martin Odersky, inventor of Scala programming language*

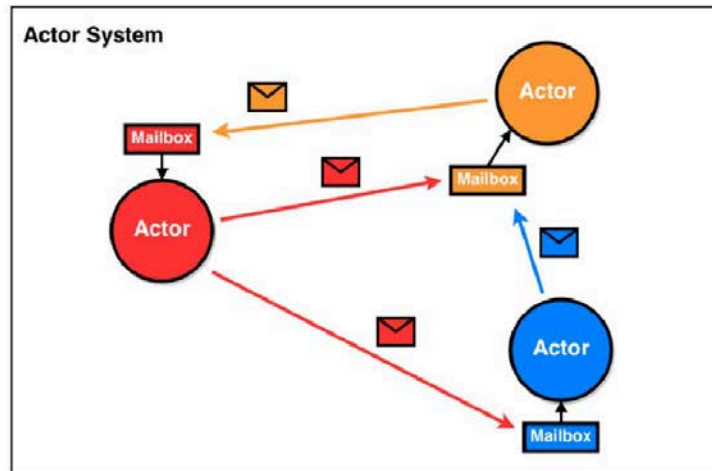
In other words, with FP we can write code in an intrinsically parallel manner and let the compiler and run-time schedule on the parallel infrastructure.

*- Paraphrasing Robert Harper, Professor of Computer Science at Carnegie Mellon University*

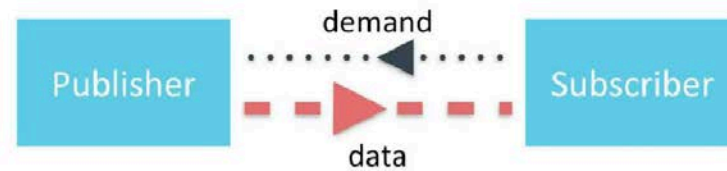


## Actor Architecture

er Science

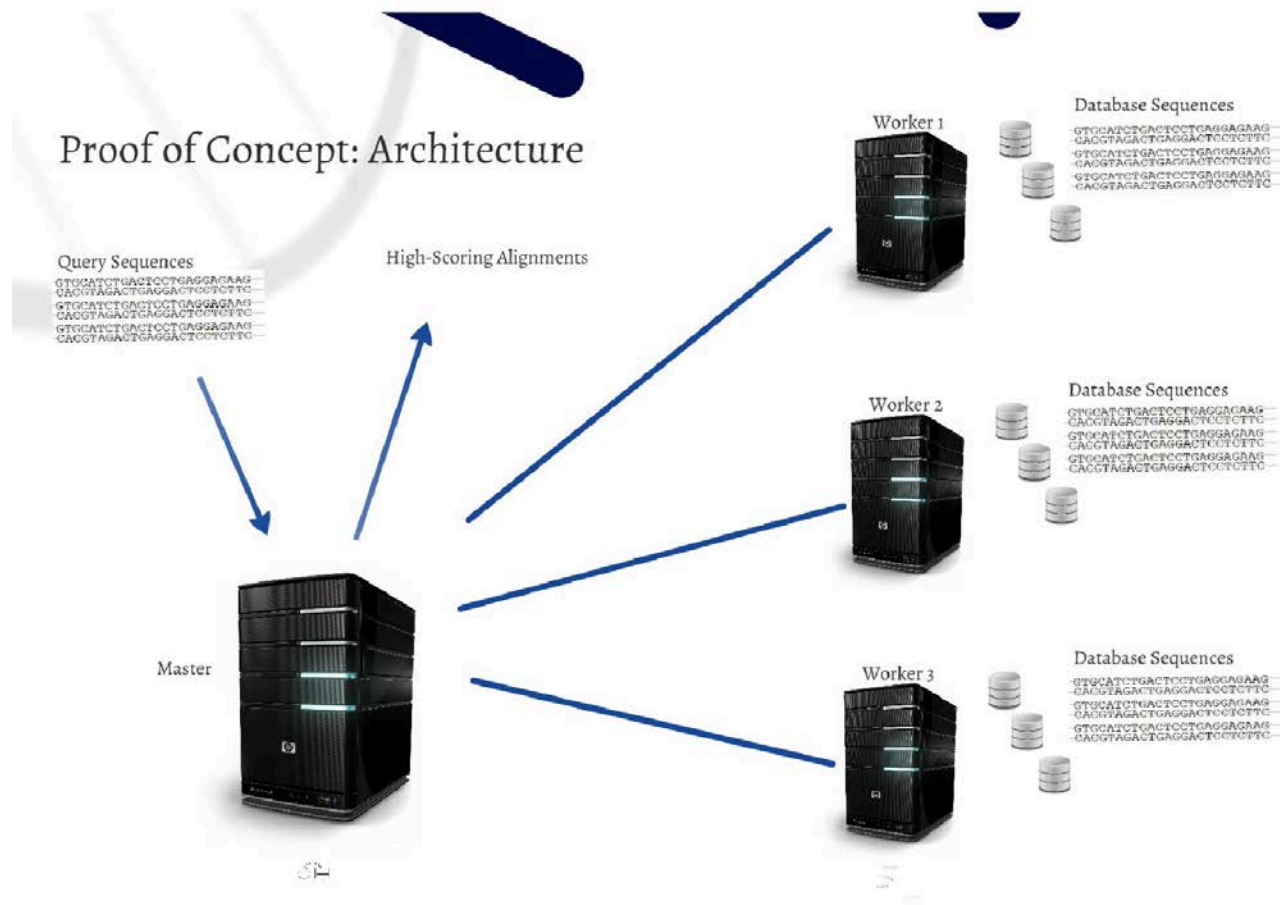


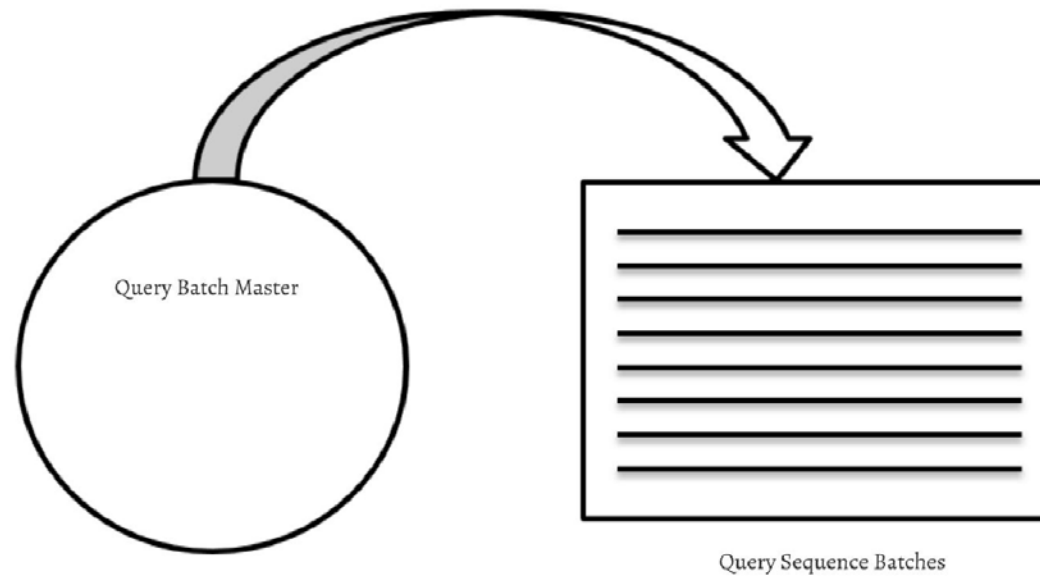
## Reactive Streams



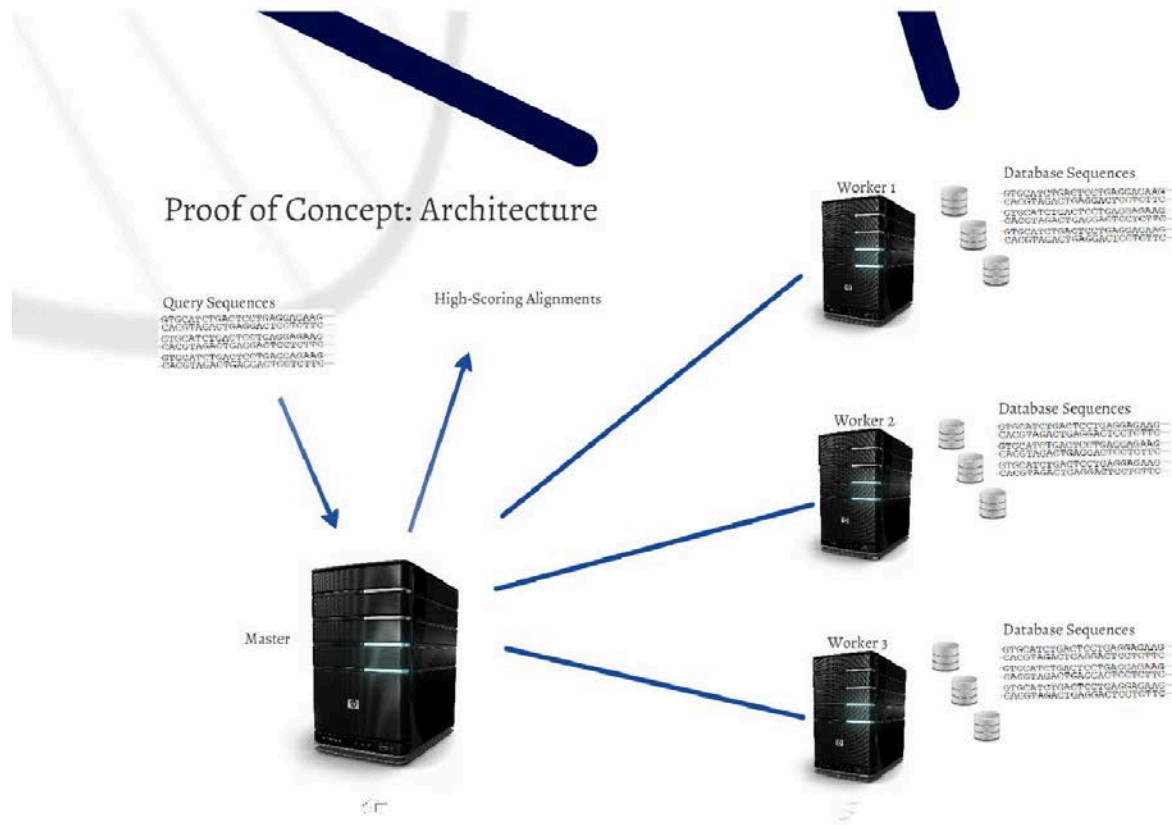


# The Data Deluge

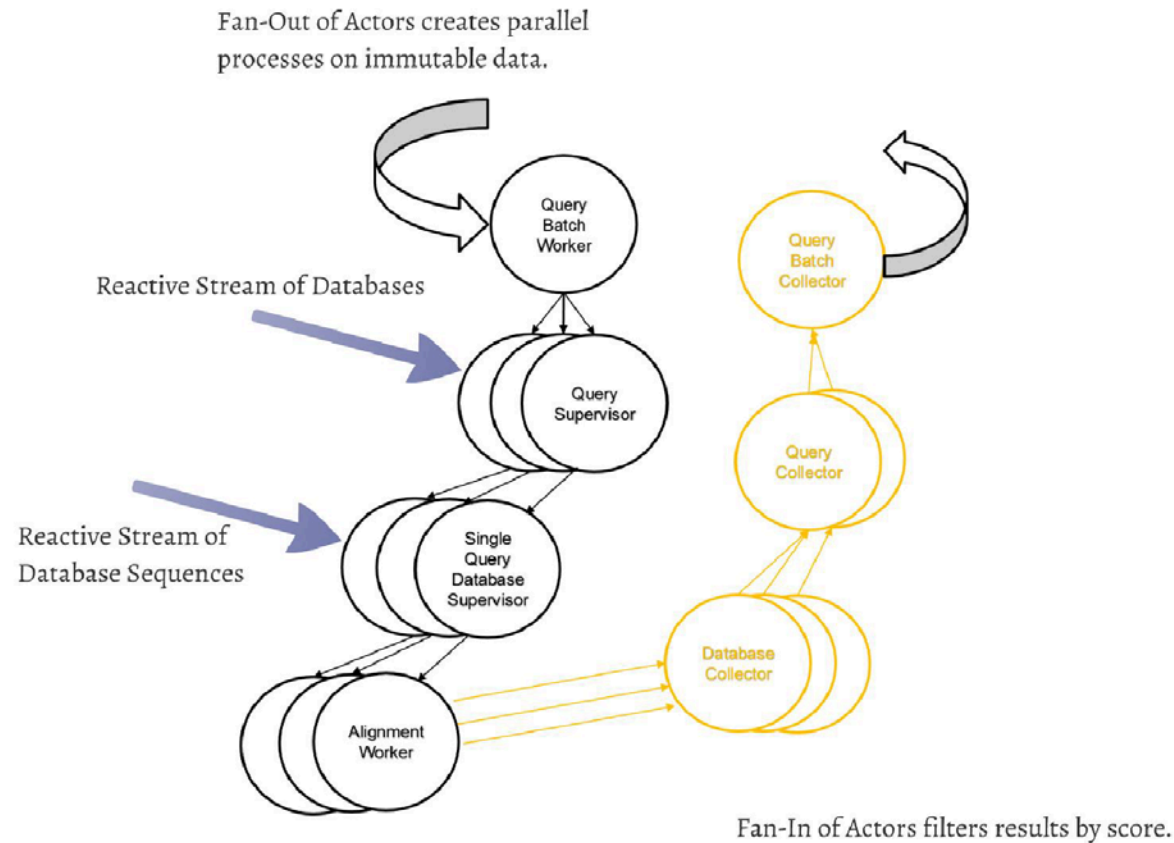




# The Data Deluge



# The Data Deluge



Proof of Concept: Results

Results on a single Node: Linux box with an i7-4770 3.4GHz processor(quad-core)

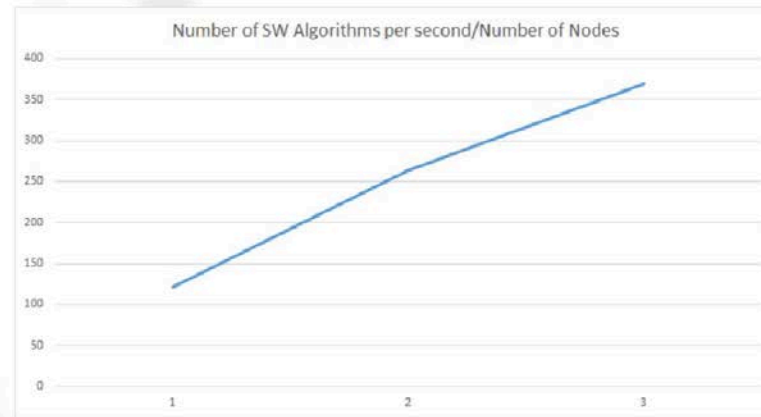
200 S-W invocations (or about 1.8 billion "cells") per second

Using a single-core baseline, this indicates ~ 15% over overhead (I/O & message passing).

~ 1000 lines of (readable!) Scala code.

What about multiple Linux boxes?

# The Data Deluge



Multiple cheap computers, with the same cheap software, multiplies the performance.

## **What's next?**

Optimization of Proof of Concept

Test on larger scale

Publish results

Consider a fully stream-centric solution.

## **(And So What!?)**

Just one of many examples:  
Clinical Microbiology

## The DNA Deluge

A Parallel Computing Approach

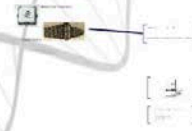
Brendan Lawlor



1. What is the deluge?



2. What is parallel computing?



3. How to harness parallel power?

