

Cloud Computing and the DNA Data Race

Michael Schatz

April 14, 2011

Data-Intensive Analysis, Analytics, and Informatics



Outline

1. Genome Assembly by Analogy
2. DNA Sequencing and Genomics
3. Large Scale Sequence Analysis
 1. Mapping & Genotyping
 2. Genome Assembly



Shredded Book Reconstruction

- Dickens accidentally shreds the first printing of A Tale of Two Cities
 - Text printed on 5 long spools

It was	the	best	of	times,	it	was	the	worst	of	times,	it	was	the	age	of	wisdom,	it	was	the	age	of	foolishness,	...	
It was	the	best	of	times,	it	was	the	worst	of	times,	it	was	the	age	of	wisdom,	it	was	the	age	of	foolishness,	...	
It was	the	best	of	times,	it	was	the	worst	of	times,	it	was	the	age	of	wisdom,	it	was	the	age	of	foolishness,	...	
It was	the	best	of	times,	it	was	the	worst	of	times,	it	was	the	age	of	wisdom,	it	was	the	age	of	foolishness,	...	
It	was	the	best	of	times,	it	was	the	worst	of	times,	it	was	the	age	of	wisdom,	it	was	the	age	of	foolishness,	...

- How can he reconstruct the text?
 - 5 copies x 138,656 words / 5 words per fragment = 138k fragments
 - The short fragments from every copy are mixed together
 - Some fragments are identical

Greedy Reconstruction

It was the best of
age of wisdom, it was
best of times, it was
it was the age of
it was the age of
it was the worst of
of times, it was the
of times, it was the
of wisdom, it was the
the age of wisdom, it
the best of times, it
the worst of times, it
times, it was the age
times, it was the worst
was the age of wisdom,
was the age of foolishness,
was the best of times,
was the worst of times,
wisdom, it was the age
worst of times, it was

It was the best of
was the best of times,
the best of times, it
best of times, it was
of times, it was the
of times, it was the
times, it was the worst
times, it was the age

The repeated sequence make the correct reconstruction ambiguous

- It was the best of times, it was the [worst/age]

Model sequence reconstruction as a graph problem.

de Bruijn Graph Construction

- $D_k = (V, E)$
 - $V =$ All length- k subfragments ($k < l$)
 - $E =$ Directed edges between consecutive subfragments
 - Nodes overlap by $k-1$ words

Original Fragment

It was the best of

Directed Edge

It was the best → was the best of

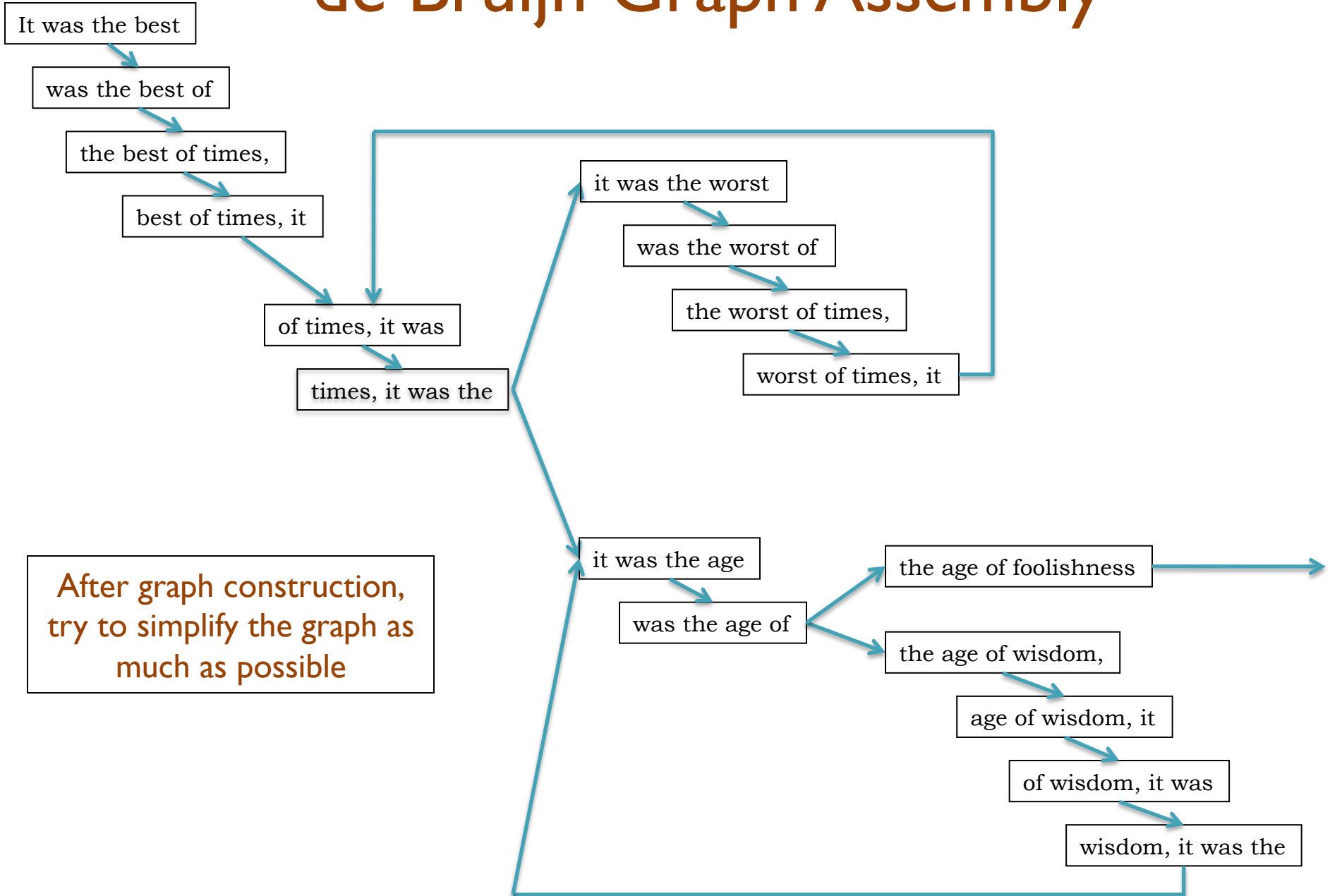
- Locally constructed graph reveals the global sequence structure
 - Overlaps between sequences implicitly computed

de Bruijn, 1946

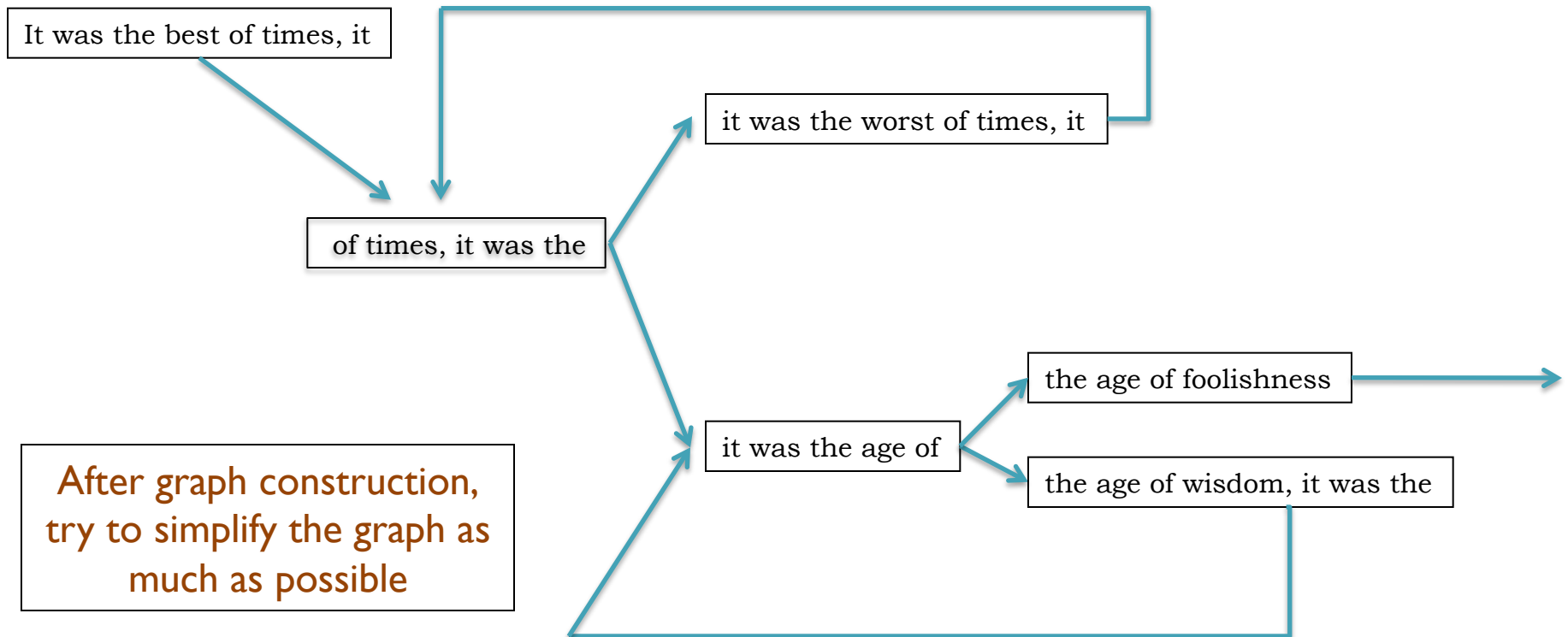
Idury and Waterman, 1995

Pevzner, Tang, Waterman, 2001

de Bruijn Graph Assembly



de Bruijn Graph Assembly

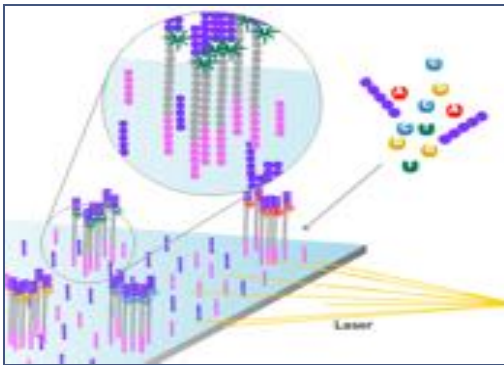


Dickens & DNA Sequencing



Genome of an organism encodes the genetic information in long sequence of 4 DNA nucleotides:ACGT

- Bacteria: ~3 million bp
- Humans: ~3 billion bp



Current DNA sequencing machines sequence hundreds of millions of short (25-500bp) reads from random positions of the genome

- ~25 GB / day / machine
- Per-base error rate estimated at 1-2% (Simpson *et al*, 2009)

ATCTGATAAGTCCCAGGACTTCAGT

GCAAGGCAAACCCGAGCCCAGTTT

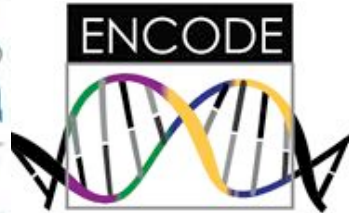
TCCAGTTCTAGAGTTTCACATGATC

GGAGTTAGTAAAAGTCCACATTGAG

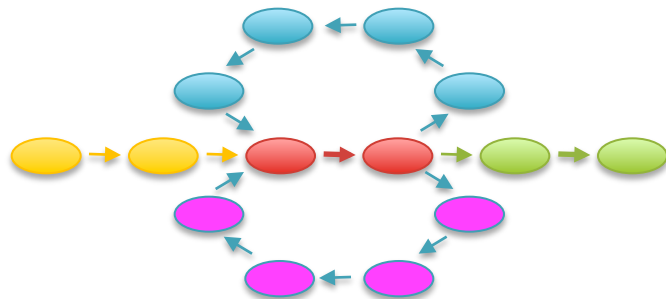
Like Dickens, we can only sequence small fragments of the genome at once.

- Must substantially oversample each genome
- A single human genome requires ~150 GB of raw data

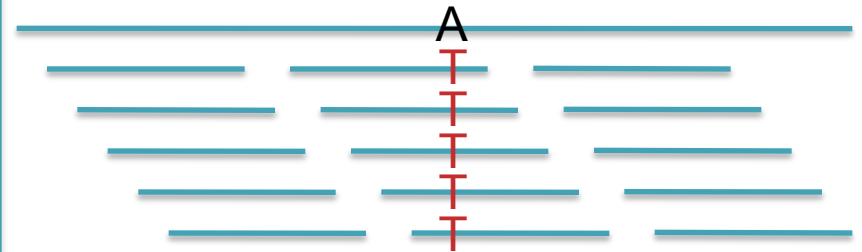
Sequencing Applications



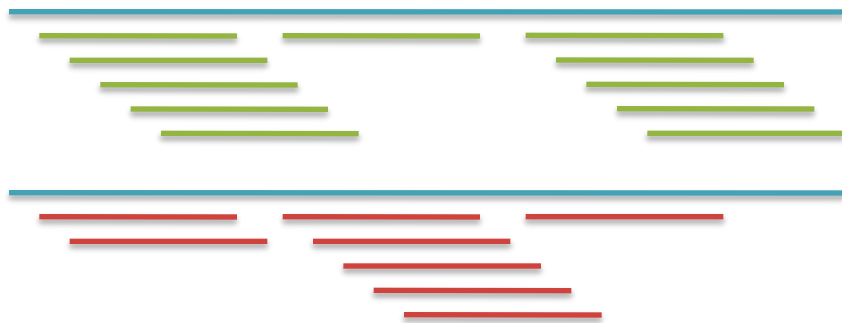
De novo Assembly



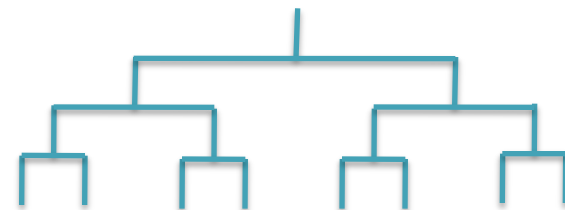
Alignment & Variations



Differential Analysis

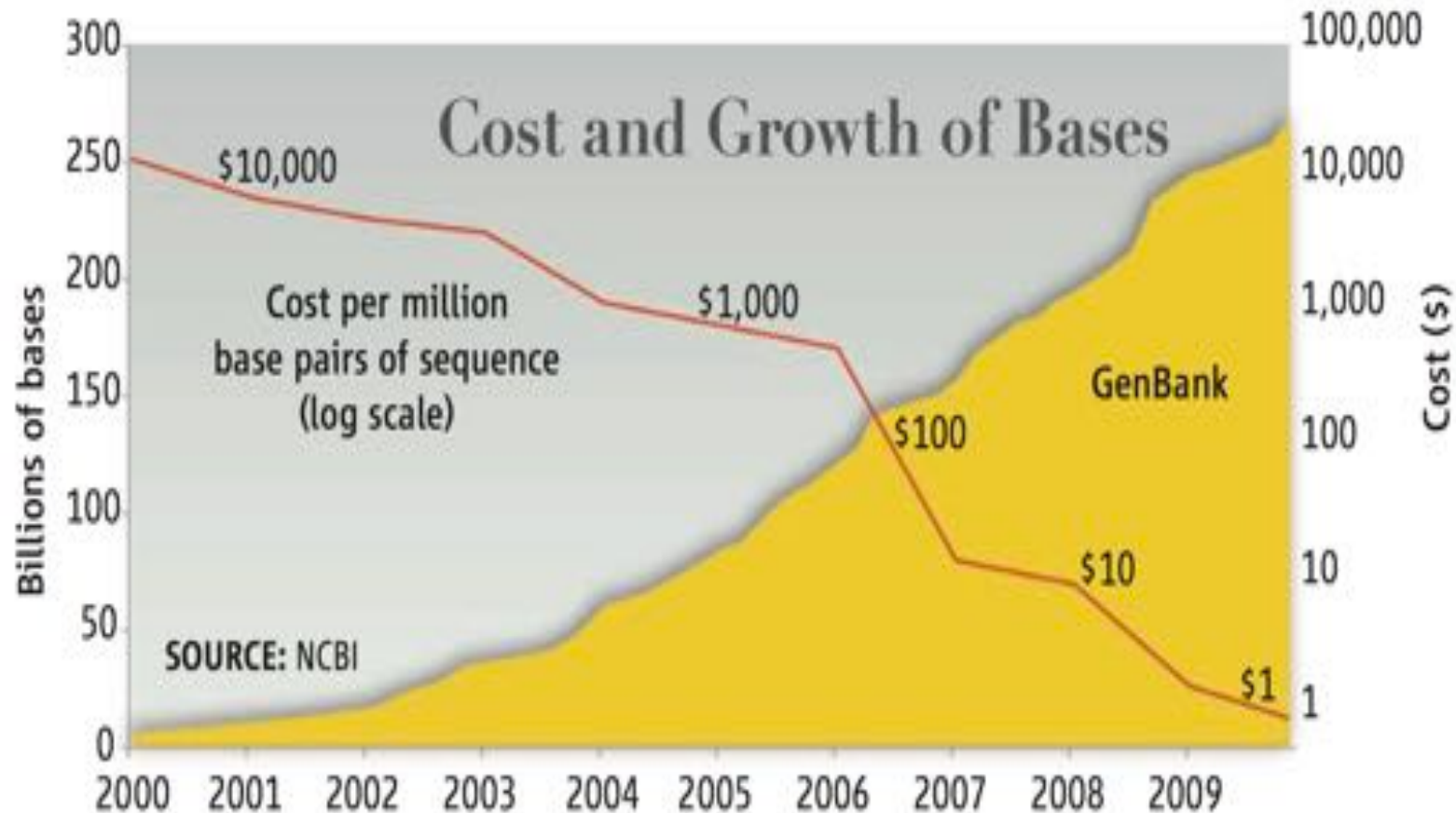


Phylogeny & Evolution



The DNA Data Tsunami

Current world-wide sequencing capacity exceeds 10Tbp/day (3.6Pbp/year) and is growing at 5x per year!



"Will Computers Crash Genomics?"

Elizabeth Pennisi (2011) *Science*. 331(6018): 666-668.

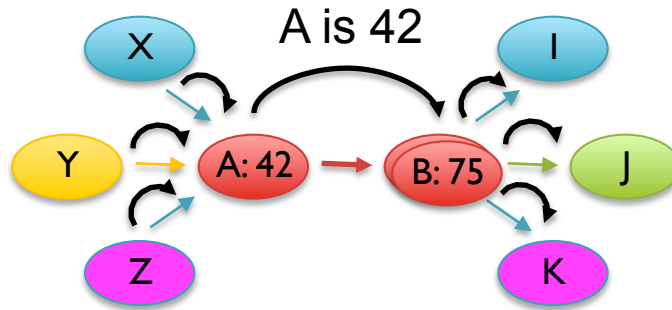
Hadoop MapReduce

<http://hadoop.apache.org>

- MapReduce is Google's framework for large data computations
 - Data and computations are spread over thousands of computers
 - Indexing the Internet, PageRank, Machine Learning, etc... (Dean and Ghemawat, 2004)
 - 946 PB processed in May 2010 (Jeff Dean at Stanford, 11.10.2010)
 - Hadoop is the leading open source implementation
 - Developed and used by Yahoo, Facebook, Twitter, Amazon, etc
- Benefits
 - Scalable, Efficient, Reliable
 - Easy to Program
 - Runs on commodity computers
- Challenges
 - Redesigning / Retooling applications
 - Not Condor, Not MPI
 - Everything in MapReduce



Distributed Graph Processing



MapReduce
Message Passing

Input:

- Graph stored as node tuples

A: (N E: B W: 42)
B: (N E: I, J, K W: 33)

Map

- For all nodes, re-emit node tuple
- For all neighbors, emit value tuple

A: (N E: B W: 42)
B: (V A 42)
B: (N E: I, J, K W: 33)
...

Shuffle

- Collect tuples with same key

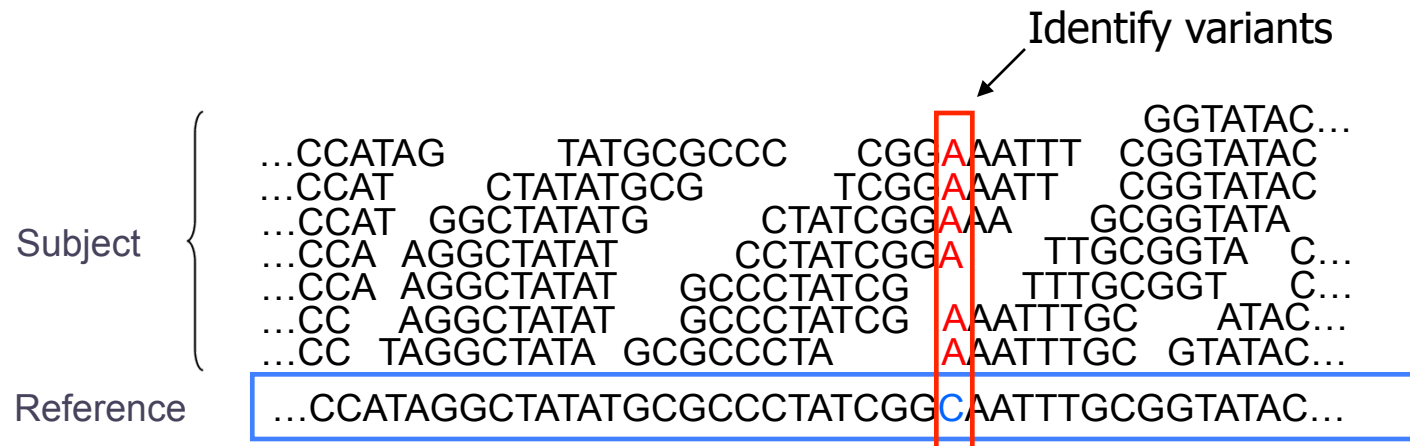
B: (N E: I, J, K W: 33)
B: (V A 42)

Reduce

- Add together values, save updated node tuple

B: (N E: I, J, K W: 75)

Short Read Mapping



- Given a reference and many subject reads, report one or more “good” end-to-end alignments per alignable read
 - Find where the read most likely originated
 - Fundamental computation for many assays
 - Genotyping RNA-Seq Methyl-Seq
 - Structural Variations Chip-Seq Hi-C-Seq

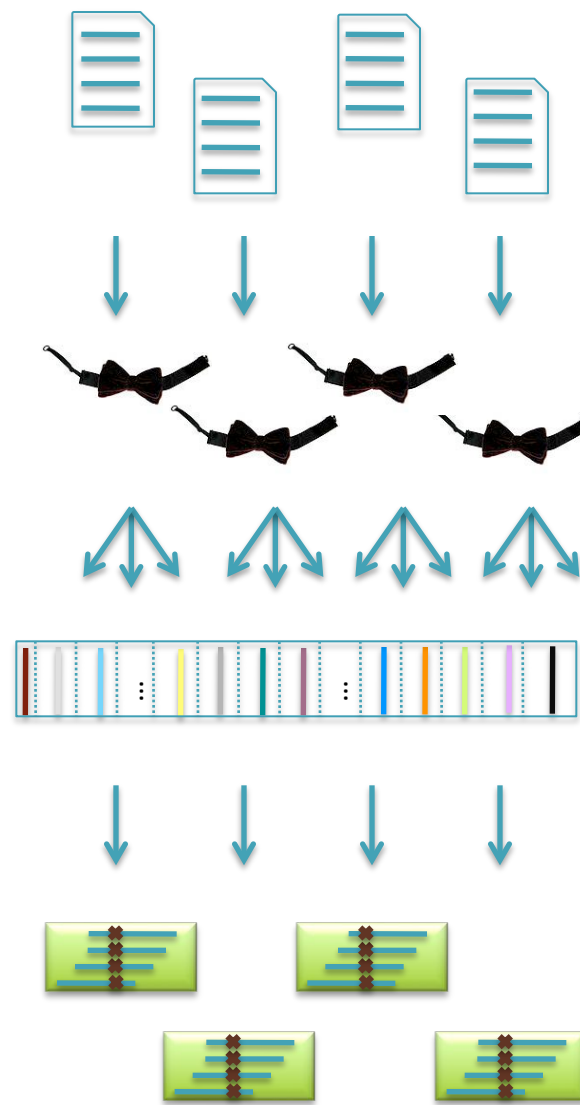
- Desperate need for scalable solutions
 - Single human requires >1,000 CPU hours / genome



Crossbow

<http://bowtie-bio.sourceforge.net/crossbow>

- Align billions of reads and find SNPs
 - Reuse software components: Hadoop Streaming
- Map: Bowtie (Langmead *et al.*, 2009)
 - Find best alignment for each read
 - Emit (chromosome region, alignment)
- Shuffle: Hadoop
 - Group and sort alignments by region
- Reduce: SOAPsnp (Li *et al.*, 2009)
 - Scan alignments for divergent columns
 - Accounts for sequencing error, known SNPs



Performance in Amazon EC2

<http://bowtie-bio.sourceforge.net/crossbow>

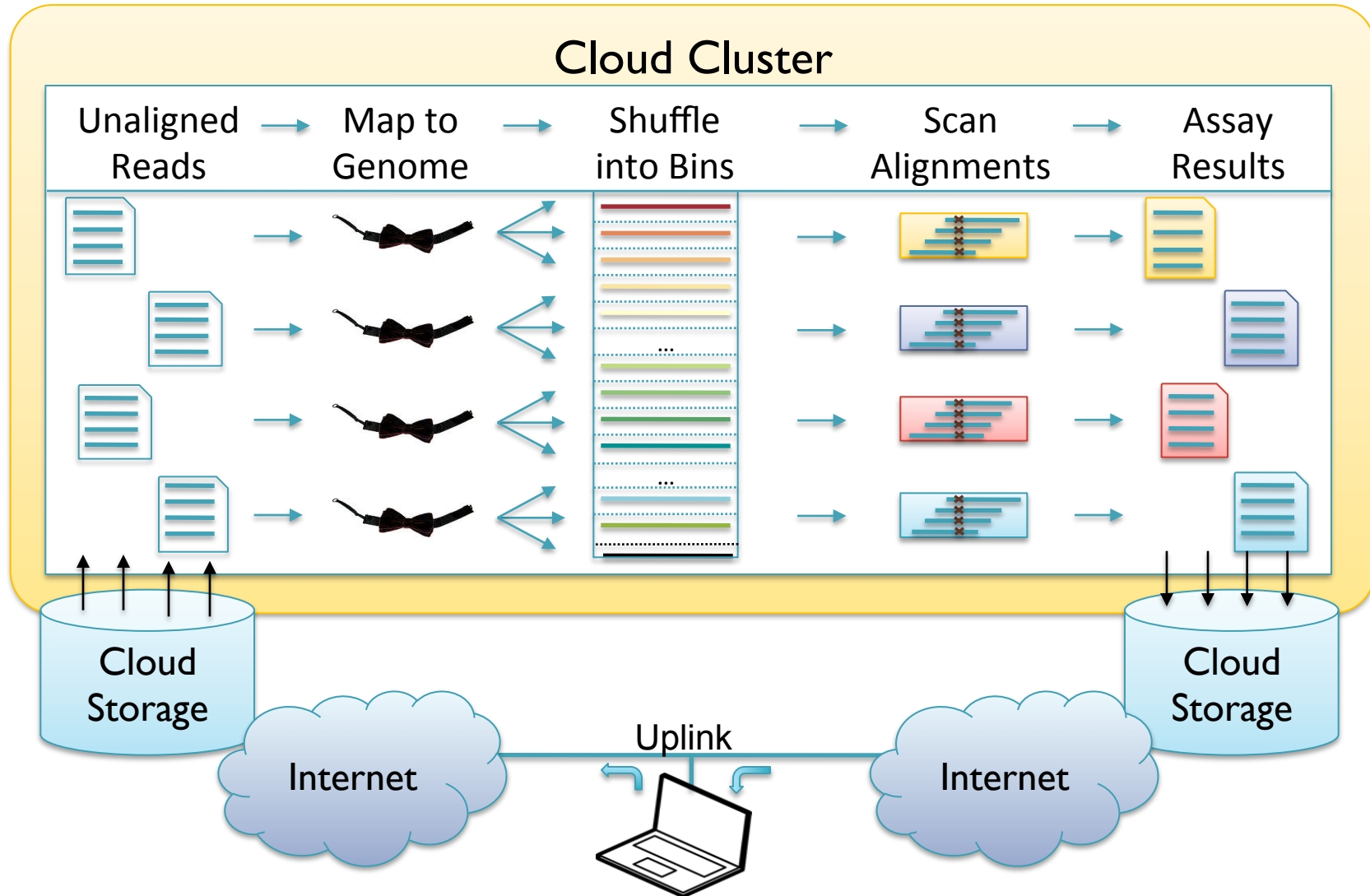
	Asian Individual Genome		
Data Loading	3.3 B reads	106.5 GB	\$10.65
Data Transfer	1h :15m	40 cores	\$3.40
Setup	0h : 15m	320 cores	\$13.94
Alignment	1h : 30m	320 cores	\$41.82
Variant Calling	1h : 00m	320 cores	\$27.88
End-to-end	4h : 00m		\$97.69

Discovered 3.7M SNPs in one human genome for ~\$100 in an afternoon.
Accuracy validated at >99%

Searching for SNPs with Cloud Computing.

Langmead B, Schatz MC, Lin J, Pop M, Salzberg SL (2009) *Genome Biology*. **10**:R134

Map-Shuffle-Scan for Genomics

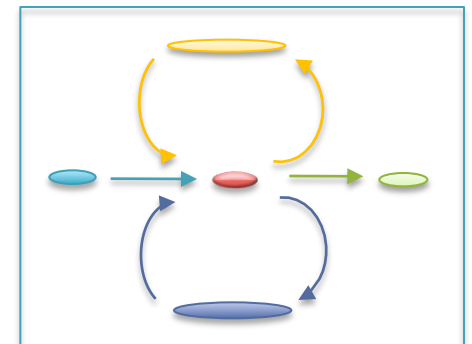
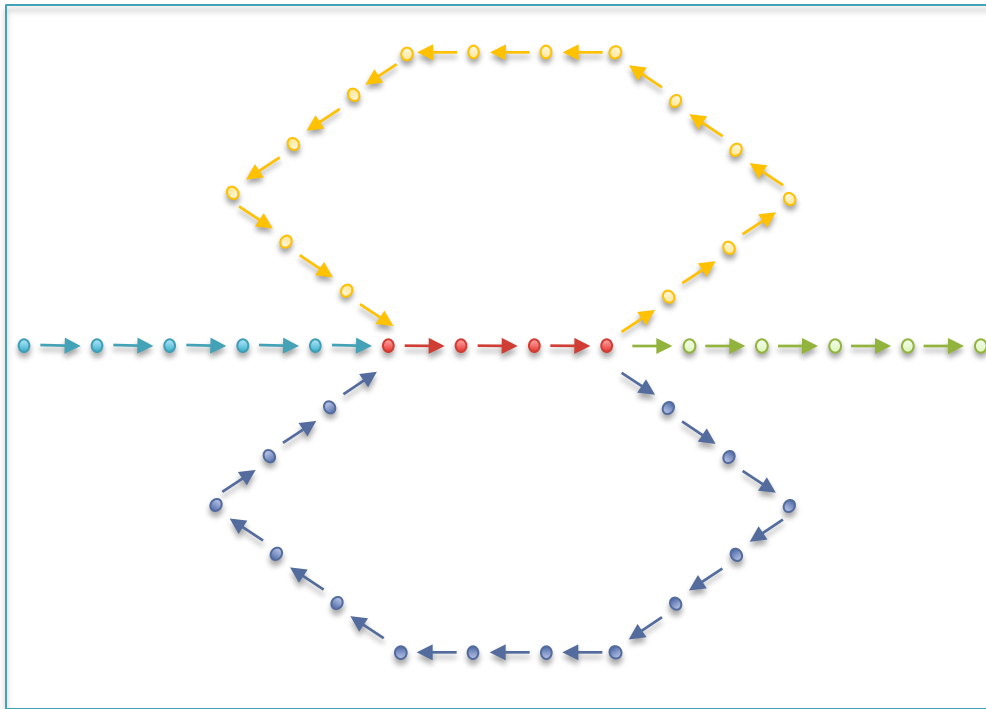


Cloud Computing and the DNA Data Race.

Schatz, MC, Langmead B, Salzberg SL (2010) *Nature Biotechnology*. **28**:691-693

Graph Compression

- After construction, many edges are unambiguous
 - Merge together compressible nodes
 - Graph physically distributed over hundreds of computers



Warmup Exercise

- Who here was born closest to April 14?
 - You can only compare to 1 other person at a time



Find winner among 64 teams in just 6 rounds

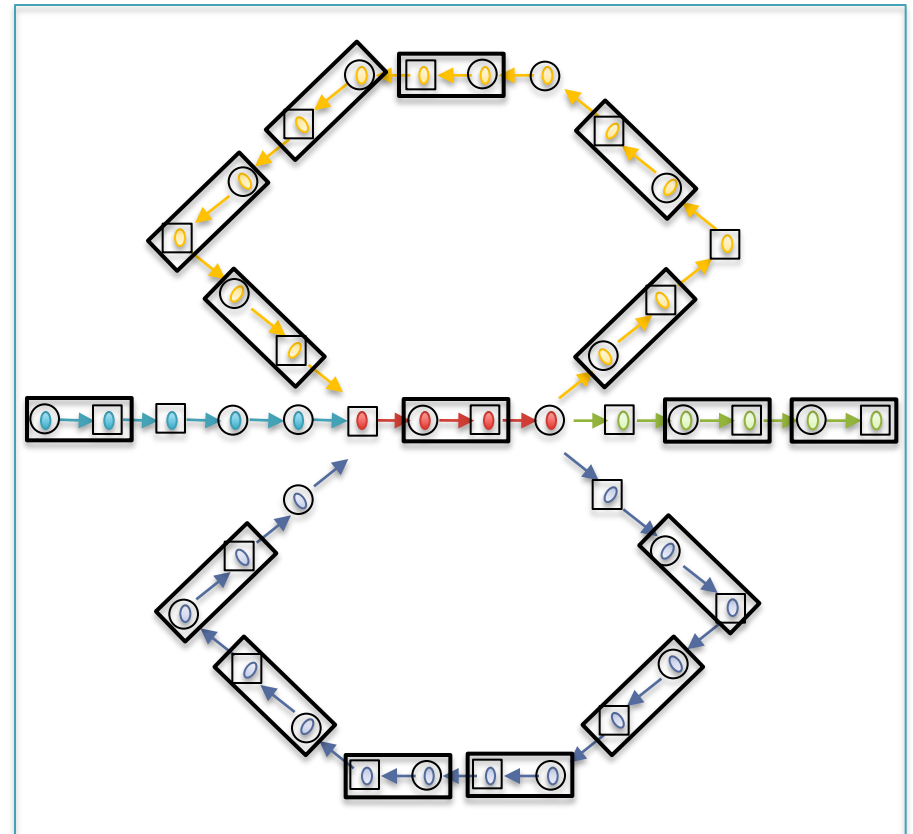
Fast Path Compression

Challenges

- Nodes stored on different computers
- Nodes can only access direct neighbors

Randomized List Ranking

- Randomly assign \textcircled{H} / $\square T$ to each compressible node
- Compress $\textcircled{H} \rightarrow \square T$ links



Initial Graph: 42 nodes

Randomized Speed-ups in Parallel Computation.

Vishkin U. (1984) *ACM Symposium on Theory of Computation*. 230-239.

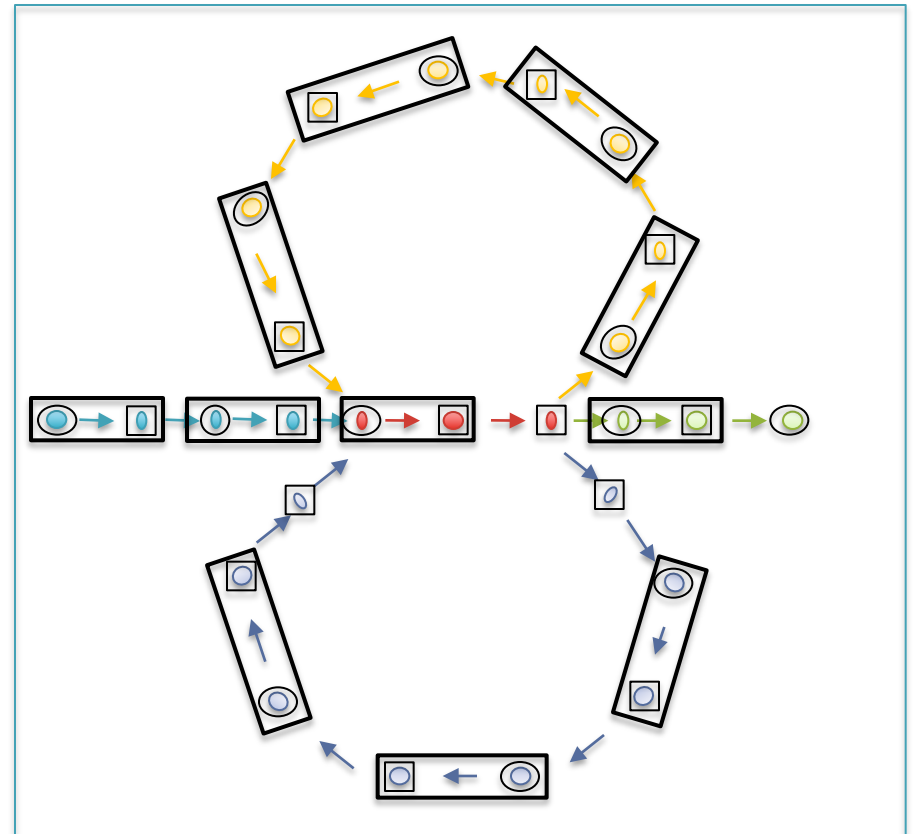
Fast Path Compression

Challenges

- Nodes stored on different computers
- Nodes can only access direct neighbors

Randomized List Ranking

- Randomly assign \textcircled{H} / $\square T$ to each compressible node
- Compress $\textcircled{H} \rightarrow \square T$ links



Round 1: 26 nodes (38% savings)

Randomized Speed-ups in Parallel Computation.

Vishkin U. (1984) *ACM Symposium on Theory of Computation*. 230-239.

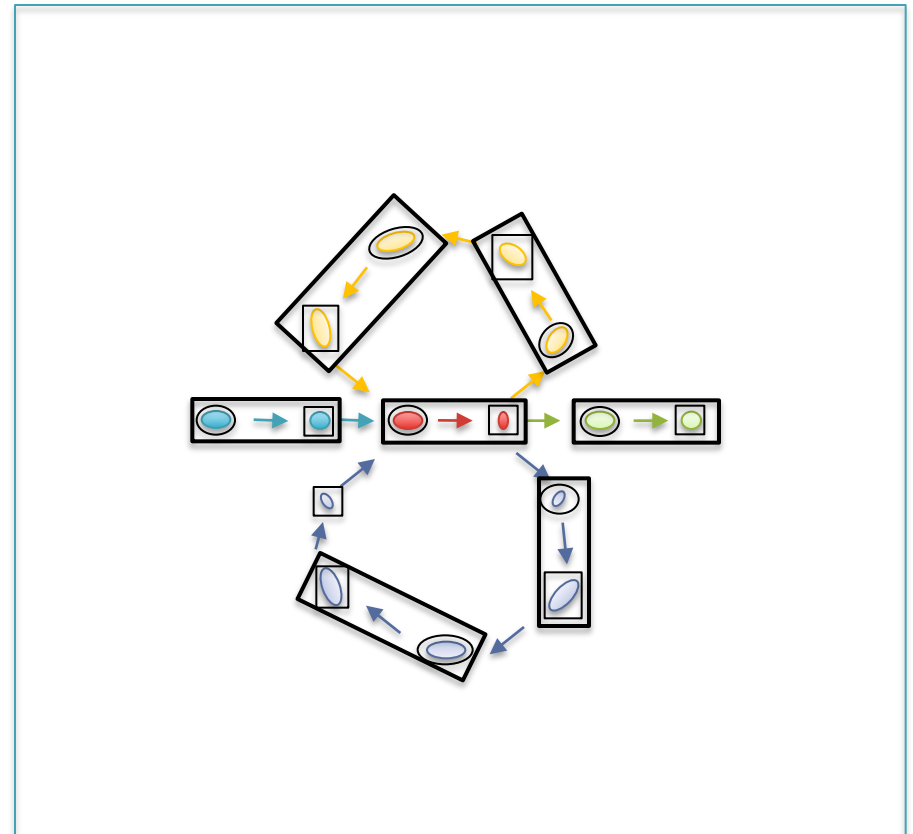
Fast Path Compression

Challenges

- Nodes stored on different computers
- Nodes can only access direct neighbors

Randomized List Ranking

- Randomly assign \textcircled{H} / \boxed{T} to each compressible node
- Compress $\textcircled{H} \rightarrow \boxed{T}$ links



Round 2: 15 nodes (64% savings)

Randomized Speed-ups in Parallel Computation.

Vishkin U. (1984) *ACM Symposium on Theory of Computation*. 230-239.

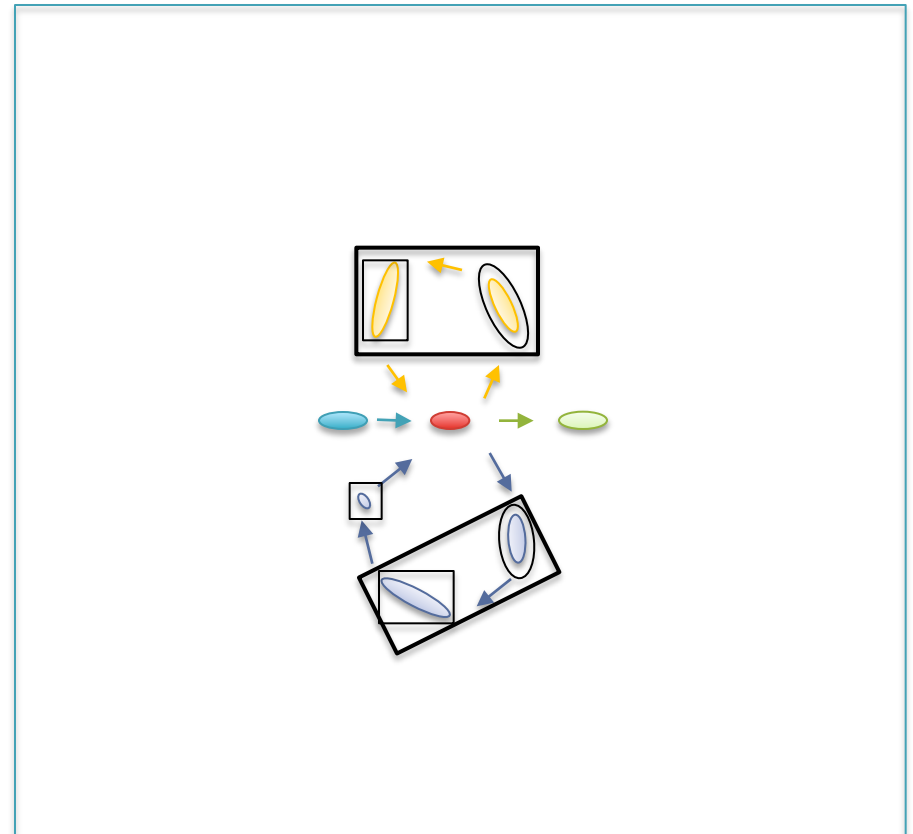
Fast Path Compression

Challenges

- Nodes stored on different computers
- Nodes can only access direct neighbors

Randomized List Ranking

- Randomly assign $\textcircled{\text{H}}$ / $\boxed{\text{T}}$ to each compressible node
- Compress $\textcircled{\text{H}} \rightarrow \boxed{\text{T}}$ links



Round 2: 8 nodes (81% savings)

Randomized Speed-ups in Parallel Computation.

Vishkin U. (1984) *ACM Symposium on Theory of Computation*. 230-239.

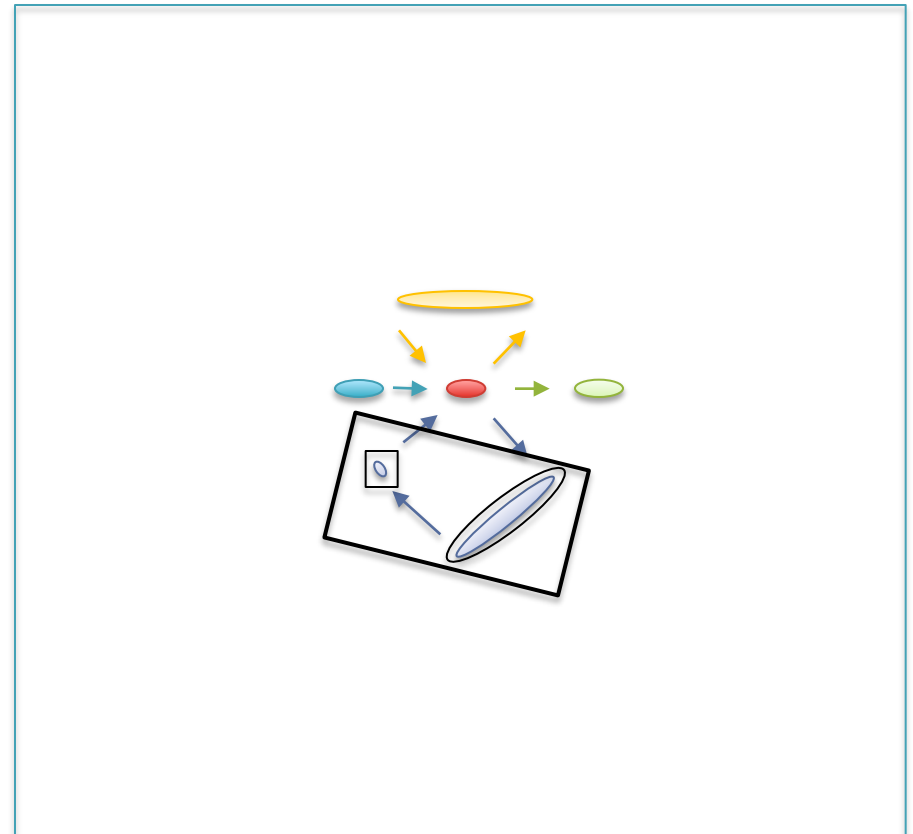
Fast Path Compression

Challenges

- Nodes stored on different computers
- Nodes can only access direct neighbors

Randomized List Ranking

- Randomly assign $\textcircled{\text{H}}$ / $\boxed{\text{T}}$ to each compressible node
- Compress $\textcircled{\text{H}} \rightarrow \boxed{\text{T}}$ links



Round 3: 6 nodes (86% savings)

Randomized Speed-ups in Parallel Computation.

Vishkin U. (1984) *ACM Symposium on Theory of Computation*. 230-239.

Fast Path Compression

Challenges

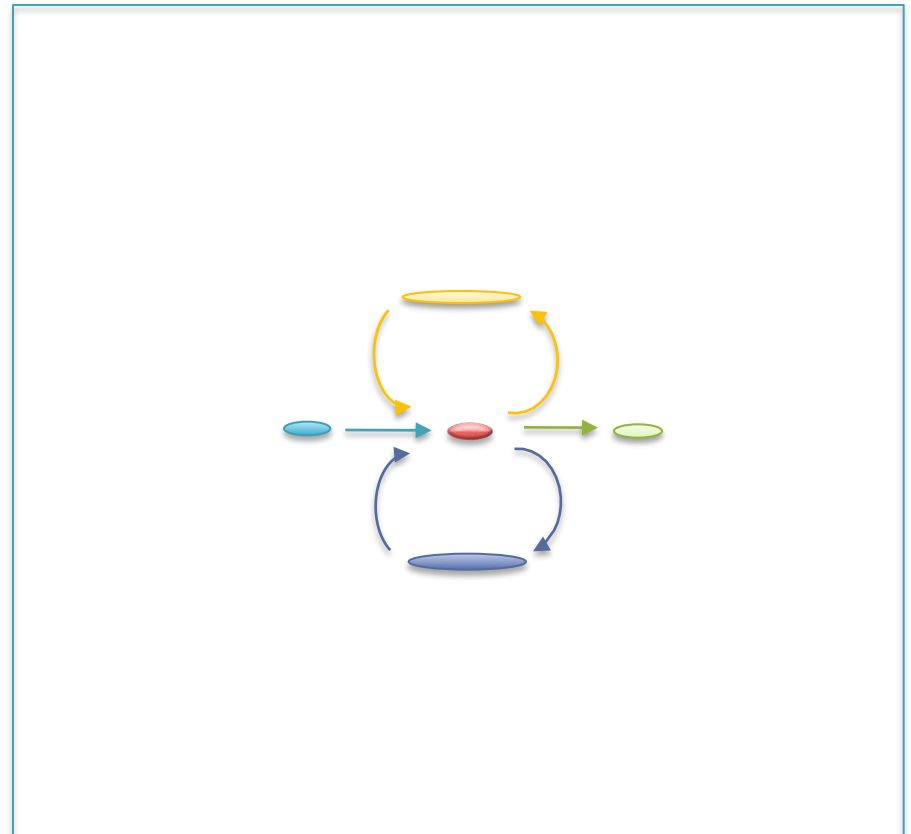
- Nodes stored on different computers
- Nodes can only access direct neighbors

Randomized List Ranking

- Randomly assign $\textcircled{\text{H}}$ / $\boxed{\text{T}}$ to each compressible node
- Compress $\textcircled{\text{H}} \rightarrow \boxed{\text{T}}$ links

Performance

- Compress all chains in $\log(S)$ rounds
- If < 1024 nodes to compress (from any number of chains), assign them all to the same reducer (save 10 rounds)

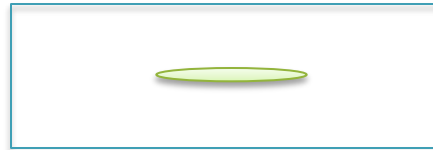


Round 4: 5 nodes (88% savings)

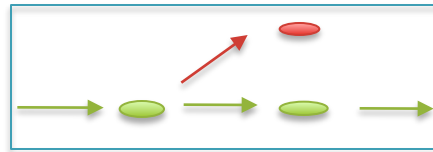
Randomized Speed-ups in Parallel Computation.

Vishkin U. (1984) *ACM Symposium on Theory of Computation*. 230-239.

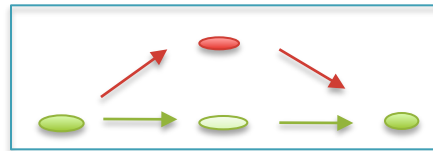
Node Types



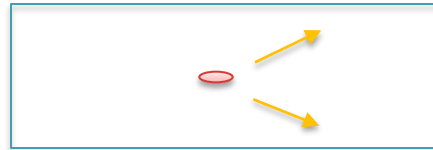
Isolated nodes (10%)



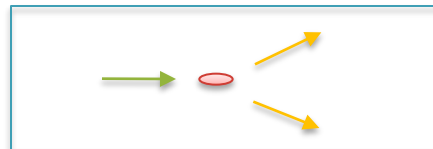
Tips (46%)



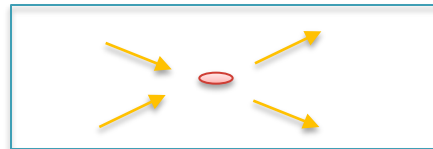
Bubbles/Non-branch (9%)



Dead Ends (.2%)



Half Branch (25%)



Full Branch (10%)

(Chaisson, 2009)

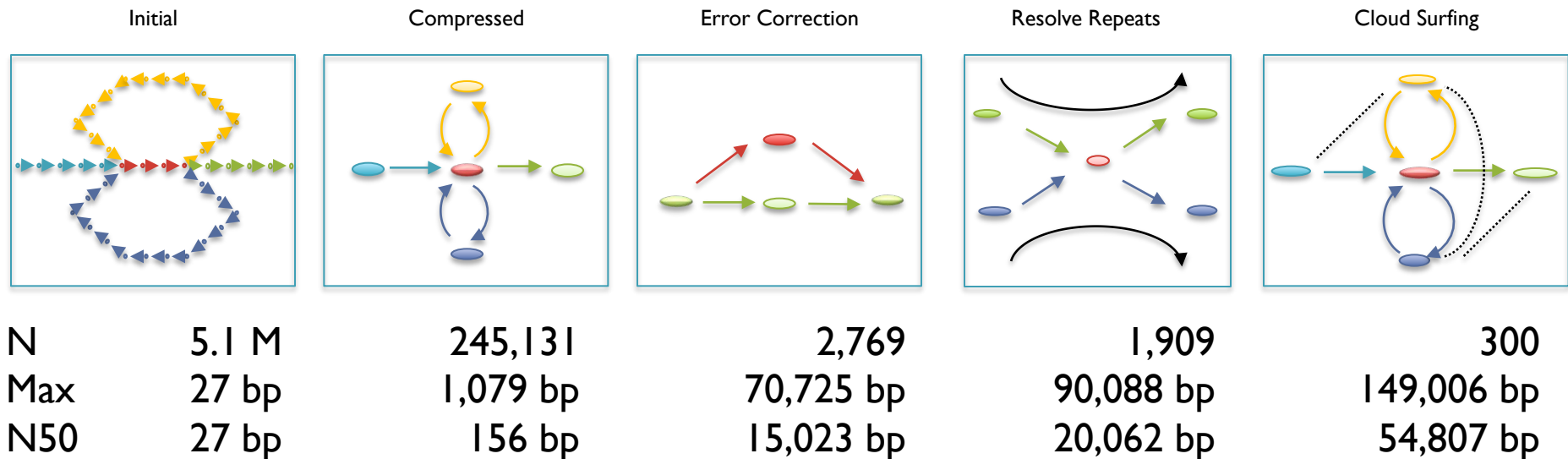
Contrail

<http://contrail-bio.sourceforge.net>



De novo bacterial assembly

- *Genome: E. coli* K12 MG1655, 4.6Mbp
- *Input: 20.8M* 36bp reads, 200bp insert (~150x coverage)
- *Preprocessor: Quake* Error Correction



Assembly of Large Genomes with Cloud Computing.

Schatz MC, Sommer D, Kelley D, Pop M, et al. *In Preparation.*

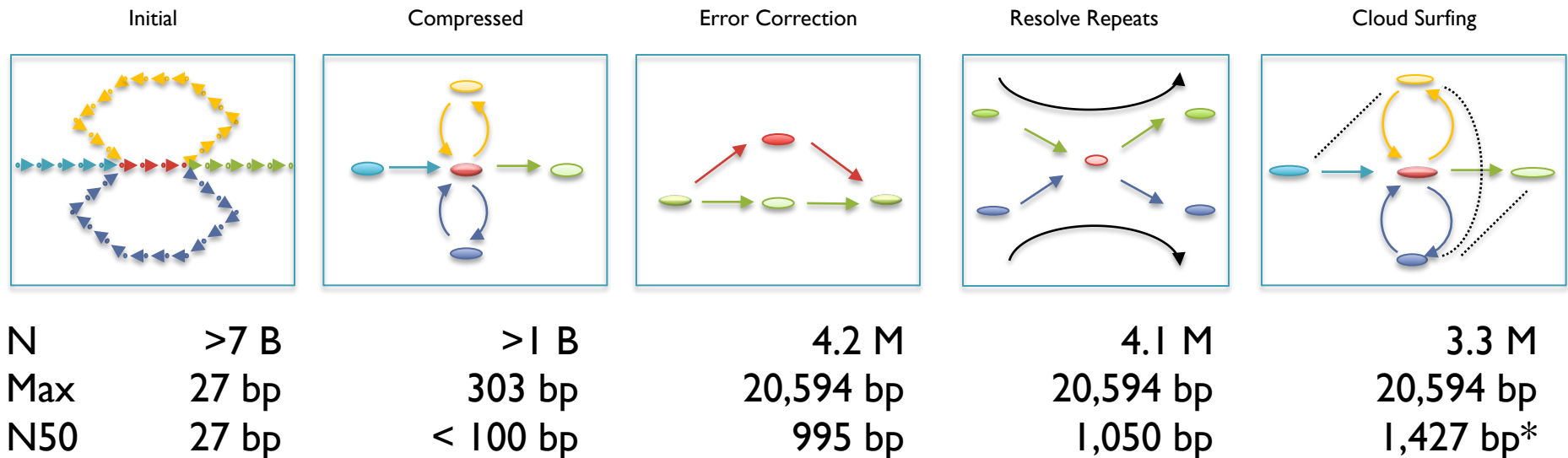
Contrail

<http://contrail-bio.sourceforge.net>



De novo assembly of a human genome

- *Genome*: African male NA18507 (SRA000271, Bentley *et al.*, 2008)
- *Input*: 3.5B 36bp reads, 210bp insert (~40x coverage)



Assembly of Large Genomes with Cloud Computing.

Schatz MC, Sommer D, Kelley D, Pop M, *et al.* *In Preparation.*

Hadoop for NGS Analysis



CloudBurst

Highly Sensitive Short Read Mapping with MapReduce

100x speedup mapping on 96 cores @ Amazon

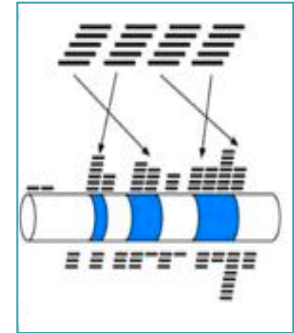
<http://cloudburst-bio.sf.net>

(Schatz, 2009)

Myrna

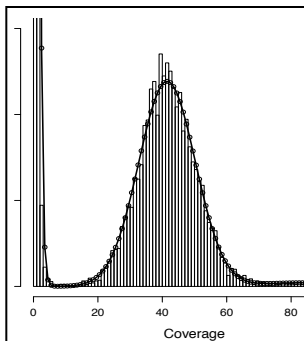
Cloud-scale differential gene expression for RNA-seq

Expression of 1.1 billion RNA-Seq reads in ~2 hours for ~\$66



(Langmead, Hansen, Leek, 2010)

<http://bowtie-bio.sf.net/myrna/>



Quake

Quality-aware error correction of short reads

Correct 97.9% of errors with 99.9% accuracy

<http://www.cbcb.umd.edu/software/quake/>

(Kelley, Schatz, Salzberg, 2010)

Genome Indexing

Rapid Parallel Construction of Genome Index

Construct the BWT of the human genome in 9 minutes

```
$GATTACA  
A$GATTAC  
ACA$GATT  
ATTACA$G  
CA$GATTA  
GATTACA£  
TACA$GAT  
TTACA$GA
```

(Menon, Bhat, Schatz, 2011*)

<http://genome-indexing.googlecode.com>



Summary

- Staying afloat in the data deluge means computing in parallel
 - Hadoop + Cloud computing is an attractive platform for large scale sequence analysis and data intensive computation
- Significant obstacles ahead
 - Bandwidth & Storage
 - Diverse applications, complex workflows
 - Rapidly changing data types
 - Time and expertise required for development
- Emerging technologies are a great start, but we need continued research
 - Need integration across disciplines

Acknowledgements

CSHL

Mike Wigler

Zach Lippman

Dick McCombie

Doreen Ware

Mitch Bekritsky

SBU

Steve Skiena

Matt Titmus

Rohith Menon

Goutham Bhat

Hayan Lee

JHU

Ben Langmead

Jeff Leek

Univ. of Maryland

Steven Salzberg

Mihai Pop

Art Delcher

Jimmy Lin

Adam Phillippy

David Kelley

Dan Sommer



Thank You!

Want to help?

<http://schatzlab.cshl.edu/apply/>

@mike_schatz