High Performance Computing for DNA Sequence Alignment and Assembly

Michael C. Schatz

April 22, 2010
CMSC858W: Algorithms for Biosequence Analysis

Outline

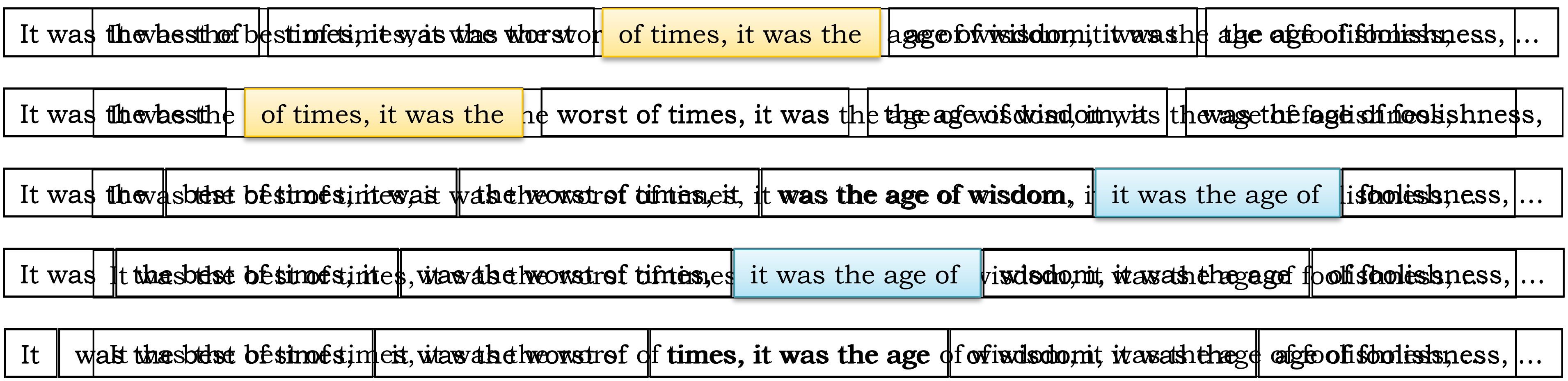
1. Genome Assembly by Analogy

2. DNA Sequencing and Genomics

3. High Performance Sequence Analysis
    1. Read Mapping
    2. Mapping & Genotyping
    3. Genome Assembly

Shredded Book Reconstruction

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



- 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

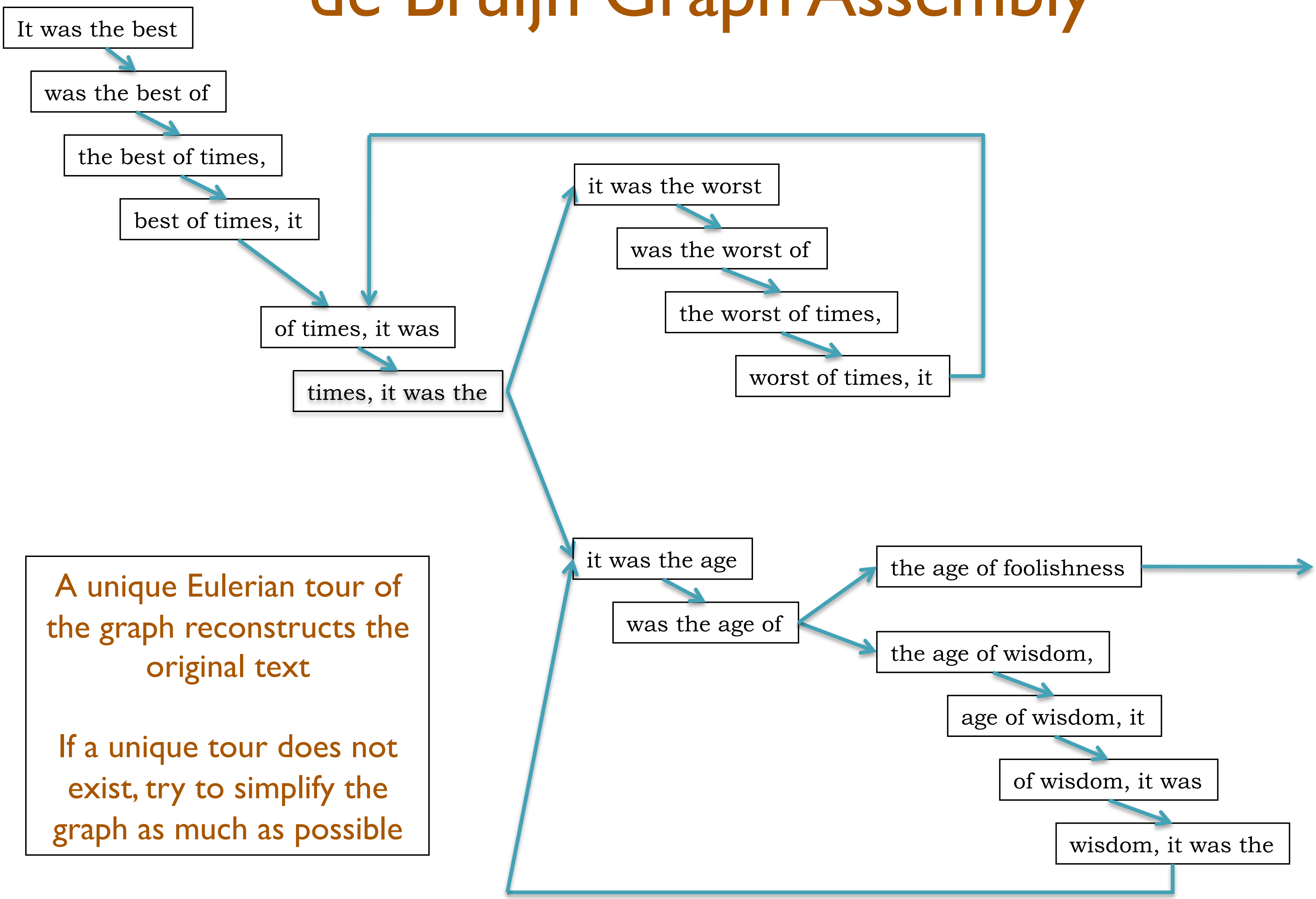
Directed Edge

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

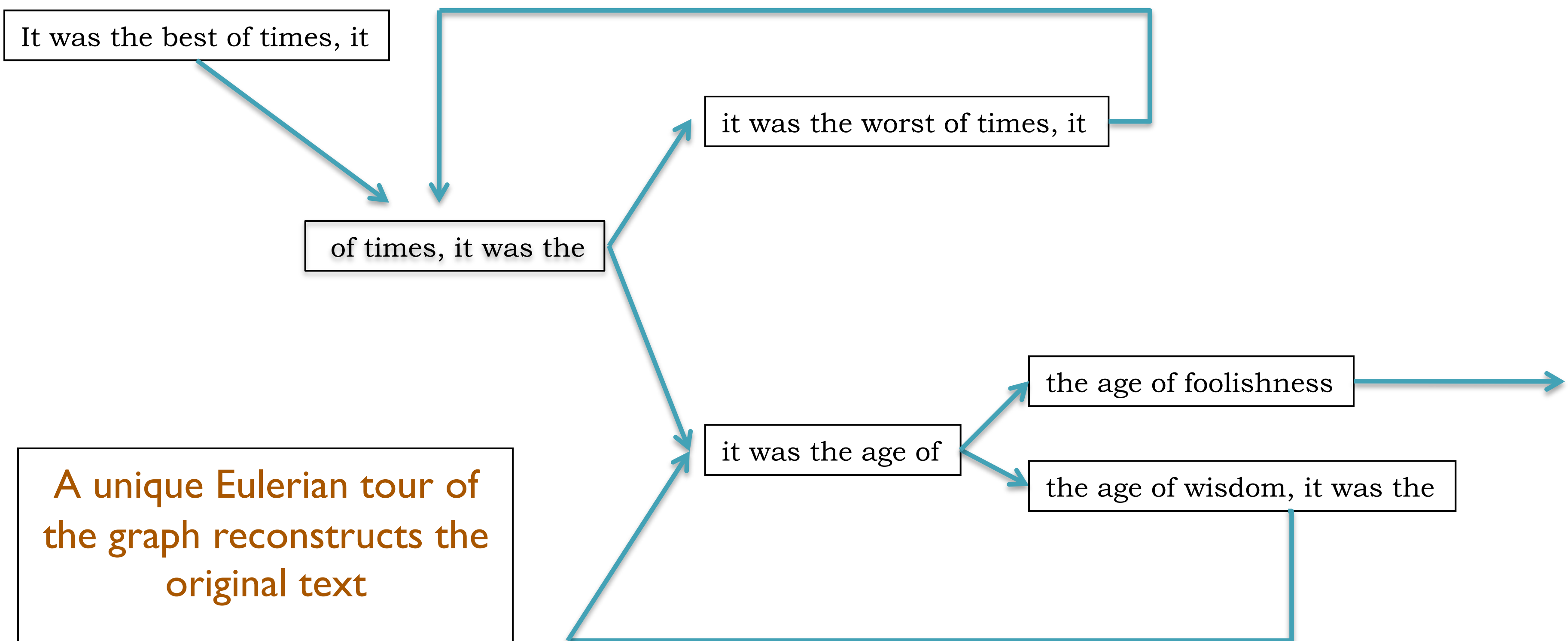
de Bruijn Graph Assembly



A unique Eulerian tour of the graph reconstructs the original text

If a unique tour does not exist, try to simplify the graph as much as possible

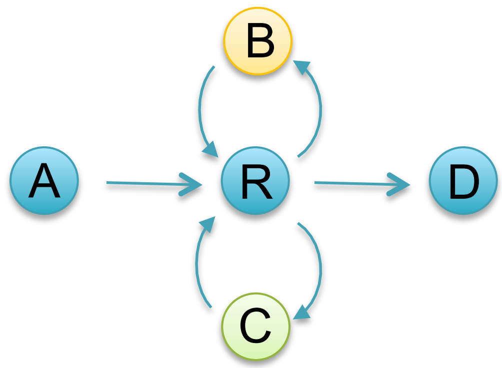
de Bruijn Graph Assembly



A unique Eulerian tour of the graph reconstructs the original text

If a unique tour does not exist, try to simplify the graph as much as possible

Counting Eulerian Tours



ARBRCRD
or
ARCRBRD

Generally an exponential number of compatible sequences

– Value computed by application of the BEST theorem (Hutchinson, 1975)

$$\mathcal{W}(G,t) = (\det L)\left\{\prod_{u \in V}(r_u - 1)!\right\}\left\{\prod_{(u,v)\in E} a_{uv}!\right\}^{-1}$$

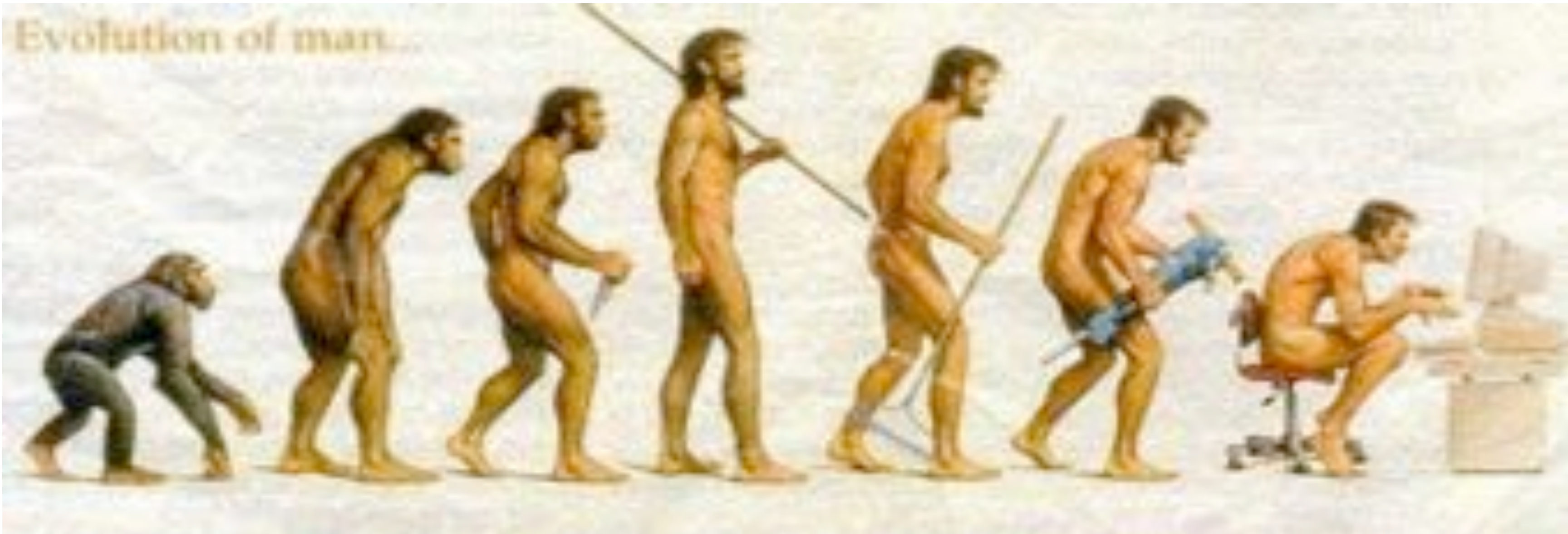
L = $n \times n$ matrix with r_u-a_{uu} along the diagonal and -a_{uv} in entry uv

$r_u = d^+(u)+1$ if $u=t$, or $d^+(u)$ otherwise

a_{uv} = multiplicity of edge from u to v

Assembly Complexity of Prokaryotic Genomes using Short Reads.
Kingsford C, Schatz MC, Pop M (2010) *BMC Bioinformatics*.

Genomics and Evolution



Your genome influences (almost) all aspects of your life

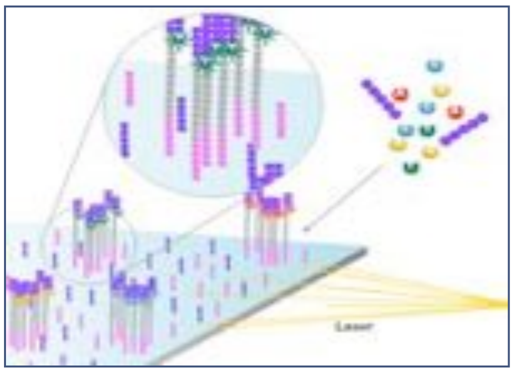
- Anatomy & Physiology: 10 fingers & 10 toes, organs, neurons
- Diseases: Sickle Cell Anemia, Down Syndrome, Cancer
- Psychological: Intelligence, Personality, Bad Driving
- Genome as a recipe, not a blueprint

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

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 can generate 1-2 Gbp of sequence per day, in millions of short reads

- Per-base error rate estimated at 1-2% (Simpson *et al*, 2009)
- Sequences originate from random positions of the genome

ATCTGATAAGTCCCAGGACTTCAGT

GCAAGGCAAACCCGAGCCCAGTTT

TCCAGTTCTAGAGTTTCACATGATC

GGAGTTAGTAAAAGTCCACATTGAG

Recent studies of entire human genomes analyzed 3.3B (Wang, et al., 2008) & 4.0B (Bentley, et al., 2008) 36bp reads

- ~100 GB of compressed sequence data

The Evolution of DNA Sequencing

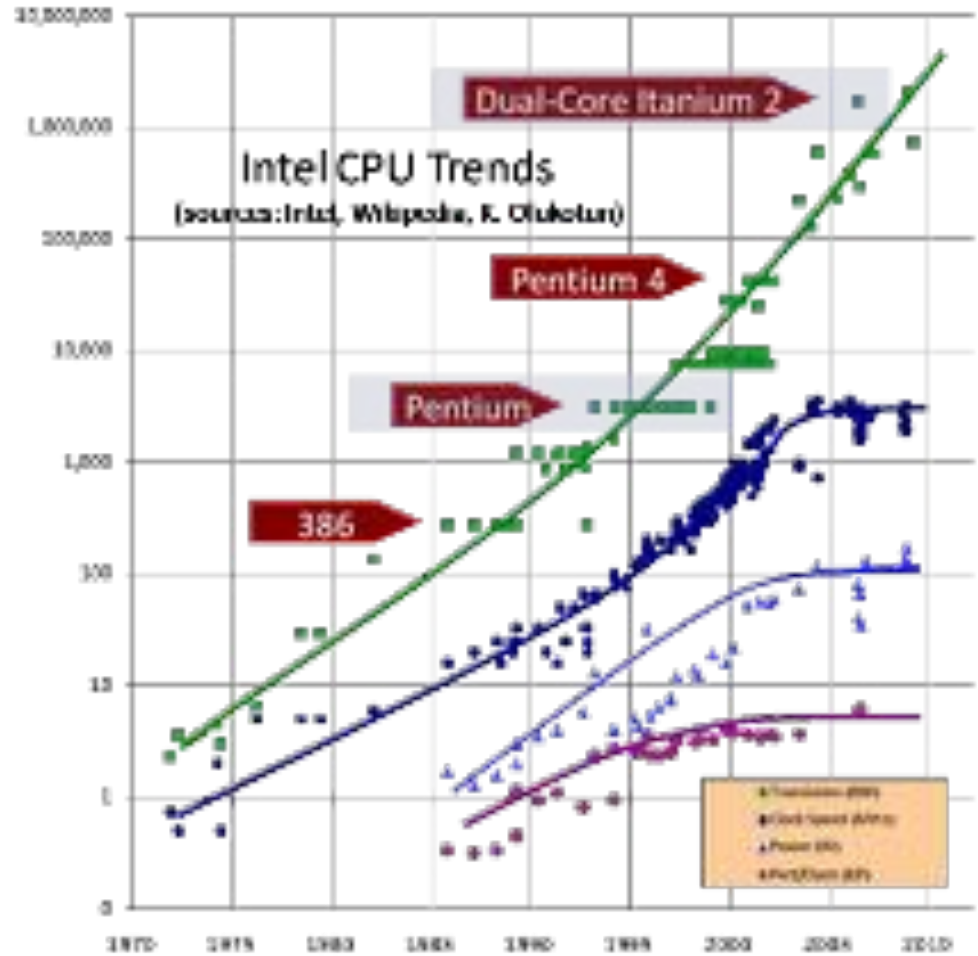
Year	Genome	Technology	Cost
2001	Venter *et al.*	Sanger (ABI)	$300,000,000
2007	Levy *et al.*	Sanger (ABI)	$10,000,000
2008	Wheeler *et al.*	Roche (454)	$2,000,000
2008	Ley *et al.*	Illumina	$1,000,000
2008	Bentley *et al.*	Illumina	$250,000
2009	Pushkarev *et al.*	Helicos	$48,000
2009	Drmanac *et al.*	Complete Genomics	$4,400

(Pushkarev *et al.*, 2009)

Critical Computational Challenges: Alignment and Assembly of Huge Datasets

Why HPC?

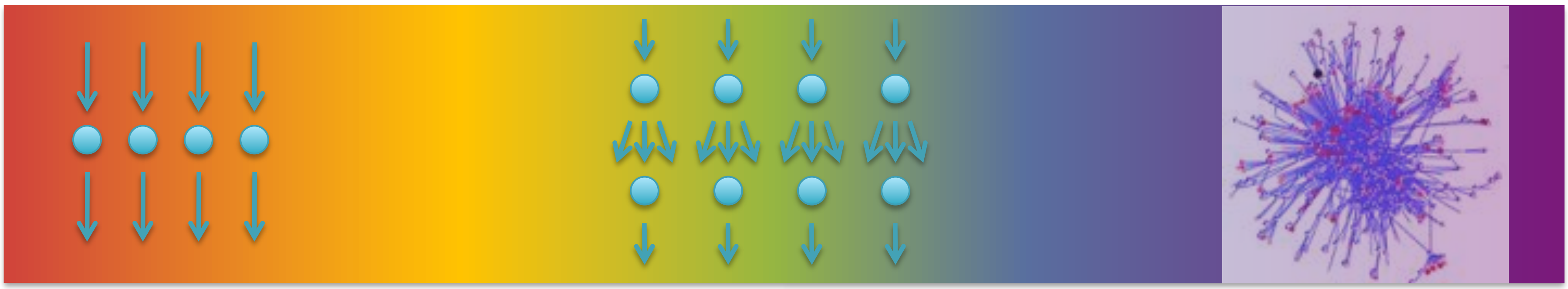
- Moore's Law is valid in 2010
  - But CPU speed is flat
  - Vendors adopting parallel solutions instead
- Parallel Environments
  - Many cores, including GPUs
  - Many computers
  - Many disks
- Why parallel
  - Need results faster
  - Doesn't fit on one machine



The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software
Herb Sutter, http://www.gotw.ca/publications/concurrency-ddj.htm

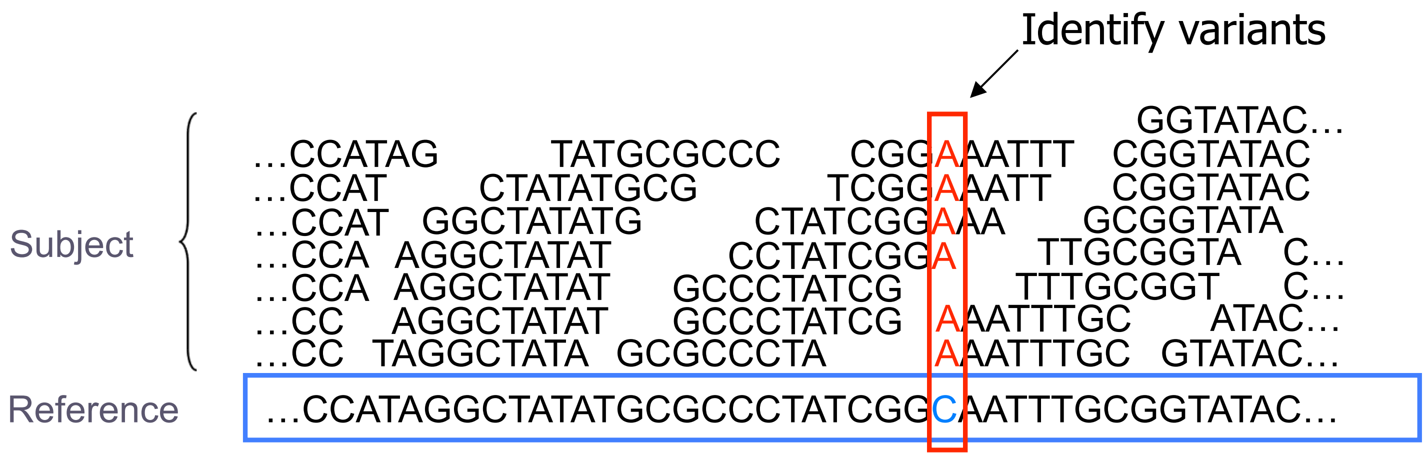
Parallel Computing Spectrum

Embarrassingly Parallel	Regularly Parallel	Deeply Parallel



Batch Computing	MapReduce/DryadLINQ	MPI/MapReduce/Pregel
Alignment HMM Scoring	Genotyping K-mer Counting	Graph Analysis Genome Assembly
Scheduling + Load Balance	Embarrassingly Parallel + Parallel Communication	Regular Parallel + Parallel Algorithm Design

Short Read Mapping



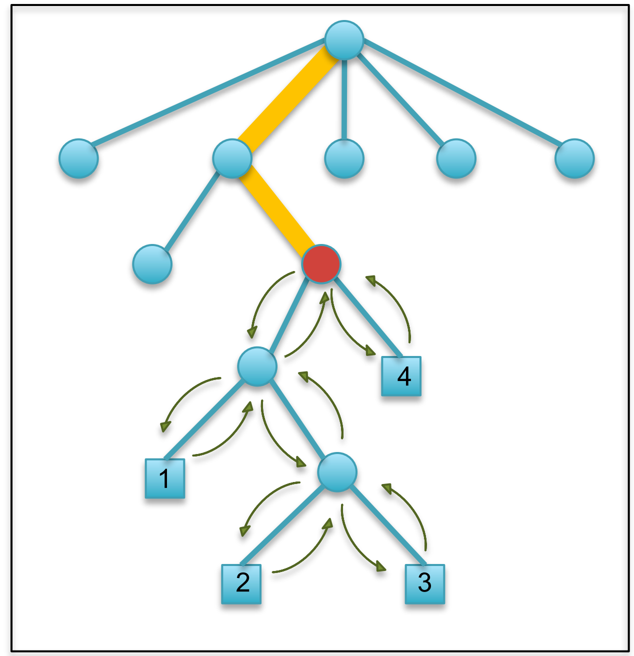
- Given a reference and many subject reads, report one or more “good” end-to-end alignments per alignable read
  - Find where the read most likely originated
  - Fundamental computation for many assays
    - Genotyping RNA-Seq Methyl-Seq
    - Structural Variations Chip-Seq Hi-C-Seq
- Desperate need for scalable solutions
  - Single human requires >1,000 CPU hours / genome

MUMmerGPU

http://mummergpu.sourceforge.net

- Map many reads simultaneously on a GPU
  - Index reference using a suffix tree
  - Find matches by walking the tree
  - Find coordinates with depth first search

- Performance on nVidia GTX 8800
  - Match kernel was ~10x faster than CPU
  - Print kernel was ~4x faster than CPU
  - End-to-end runtime ~4x faster than CPU



High-throughput sequence alignment using Graphics Processing Units.
Schatz, MC*, Trapnell, C*, Delcher, AL, Varshney, A. (2007) *BMC Bioinformatics* 8:474.

Optimizing data intensive GPGPU computations for DNA sequence alignment.
Trapnell C*, Schatz MC*. (2009) *Parallel Computing.* 35(8-9):429-440.

Elementary School Dance

Hadoop MapReduce

- MapReduce is the parallel distributed framework invented by Google for large data computations.
  - Data and computations are spread over thousands of computers, processing petabytes of data each day (Dean and Ghemawat, 2004)
  - Indexing the Internet, PageRank, Machine Learning, etc…
  - Hadoop is the leading open source implementation

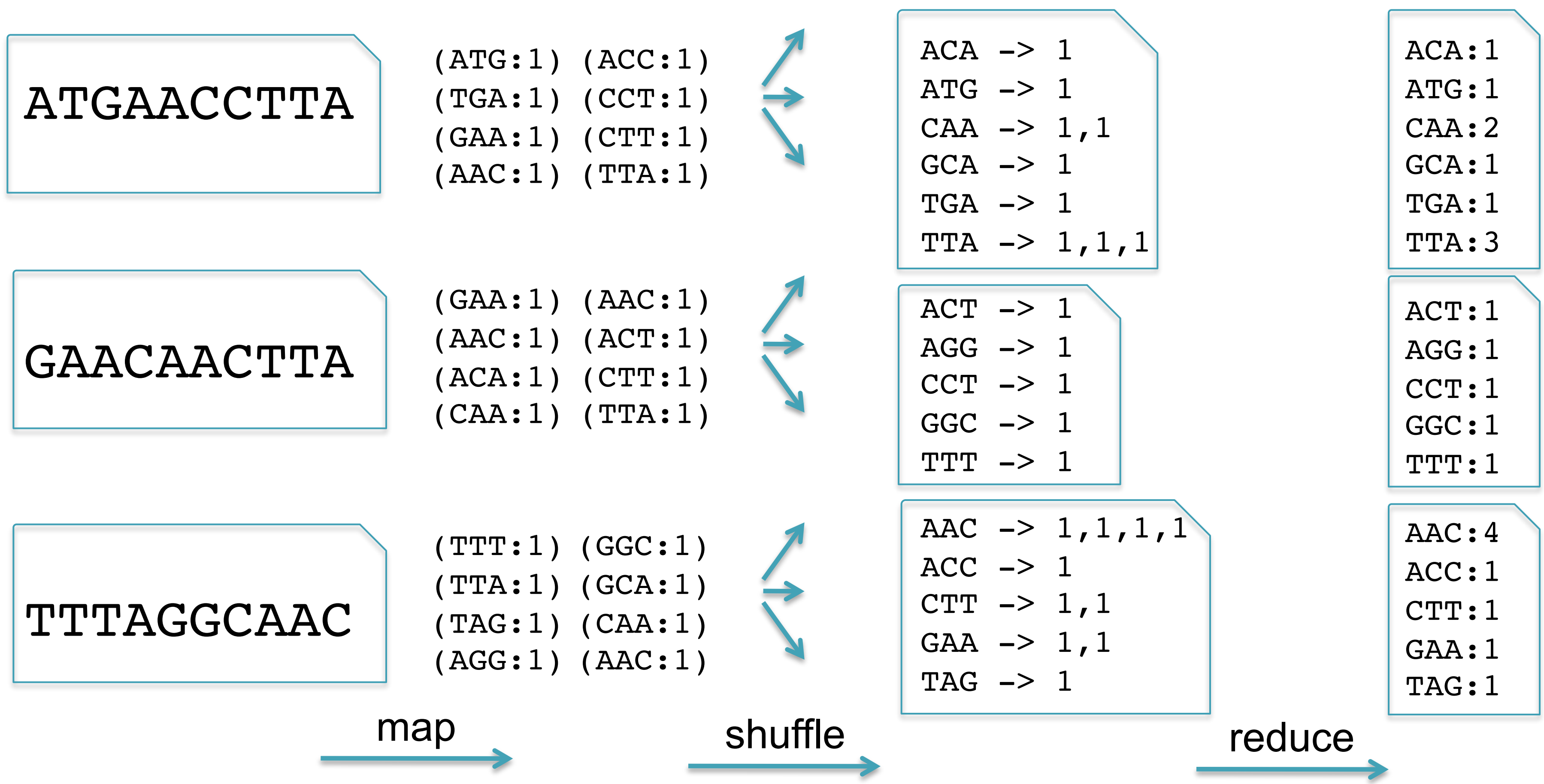
- Benefits
  - Scalable, Efficient, Reliable
  - Easy to Program
  - Runs on commodity computers

- Challenges
  - Redesigning / Retooling applications
    - Not Condor, Not MPI
    - Everything in MapReduce

K-mer Counting

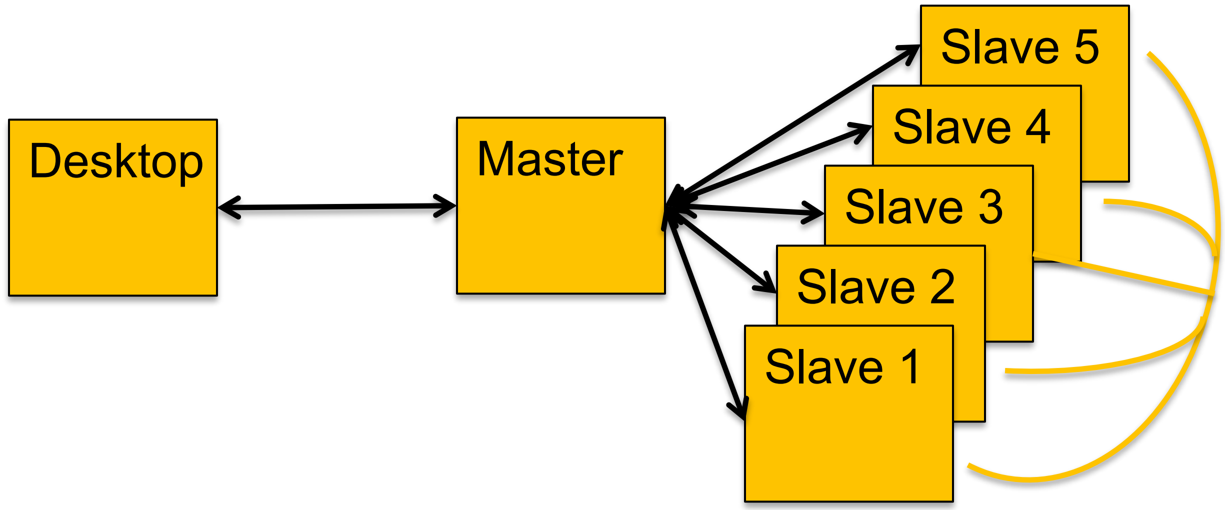
- Application developers focus on 2 (+1 internal) functions
  - Map: input ➔ key:value pairs
  - Shuffle: Group together pairs with same key
  - Reduce: key, value-lists ➔ output

Map, Shuffle & Reduce
All Run in Parallel



Junior High Dance

Hadoop Architecture

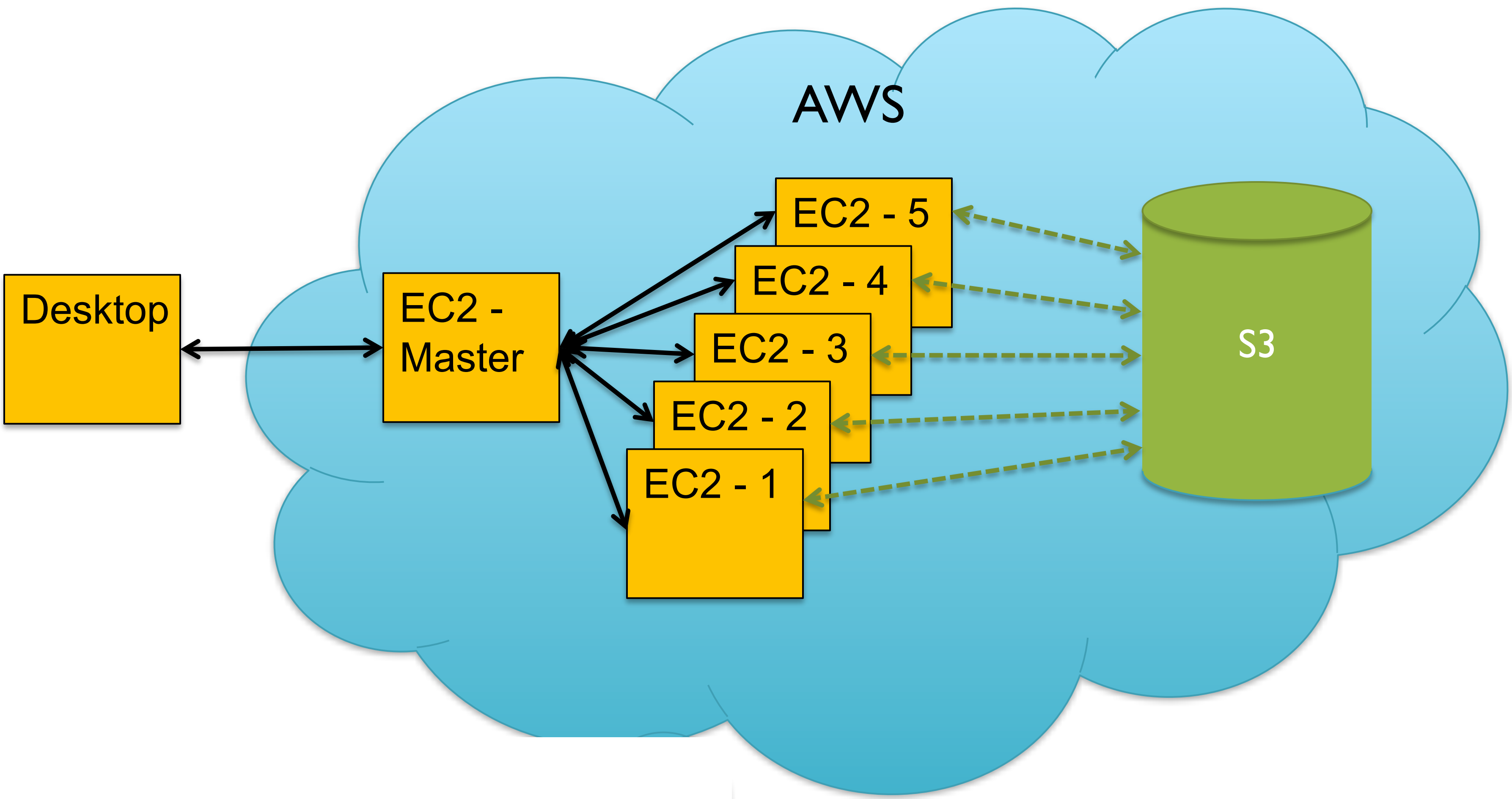


- Hadoop Distributed File System (HDFS)
  - Data files partitioned into large chunks (64MB), replicated on multiple nodes
  - NameNode stores metadata information (block locations, directory structure)

- Master node (JobTracker) schedules and monitors work on slaves
  - Computation moves to the data, rack-aware scheduling

- Hadoop MapReduce system won the 2009 GreySort Challenge
  - Sorted 100 TB in 173 min (578 GB/min) using 3452 nodes and 4x3452 disks

Hadoop on AWS



- If you don't have 1000s of machines, you can rent them from Amazon
  - After machines spool up, ssh to master as if it was a local machine.
  - Use S3 for persistent data storage, with very fast interconnect to EC2.

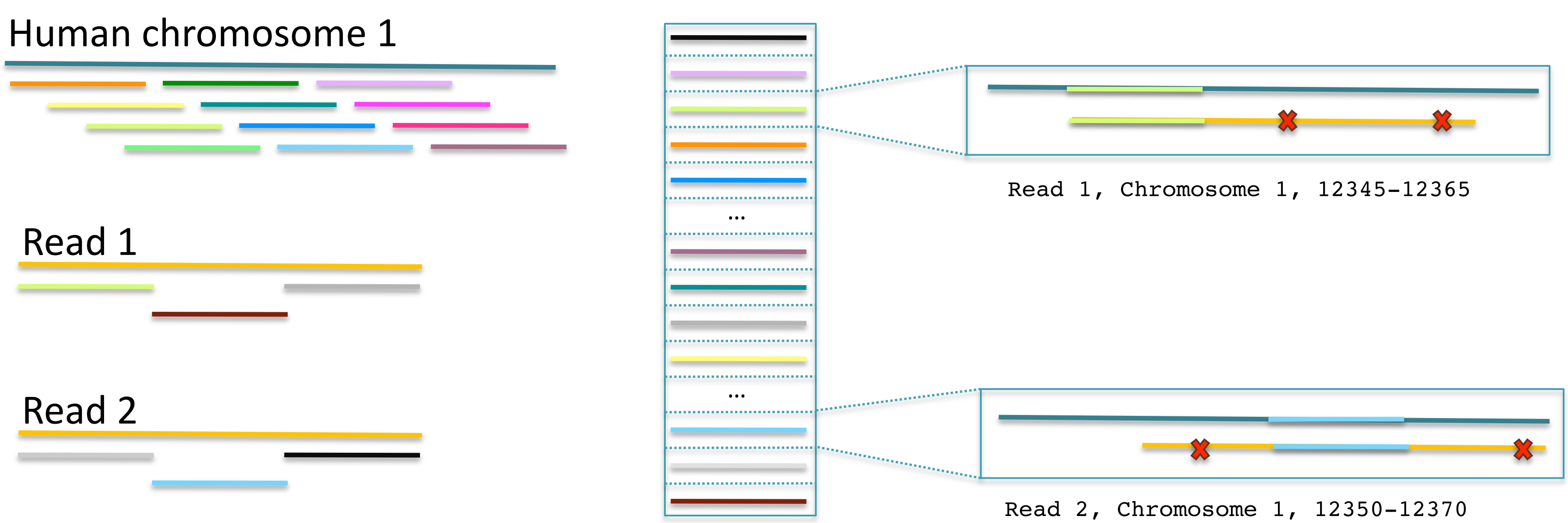
CloudBurst

http://cloudburst-bio.sourceforge.net

1. Map: Catalog K-mers
   - Emit k-mers in the genome and reads

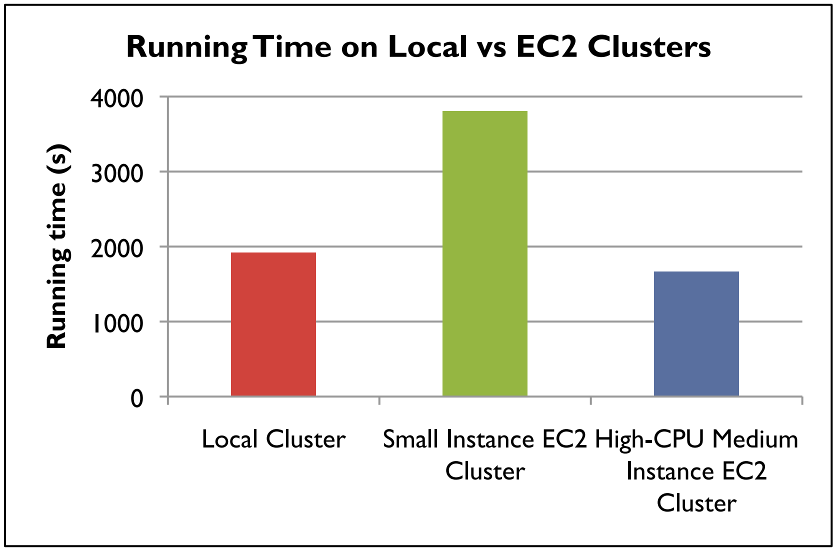
2. Shuffle: Collect Seeds
   - Conceptually build a inverted index of k-mers

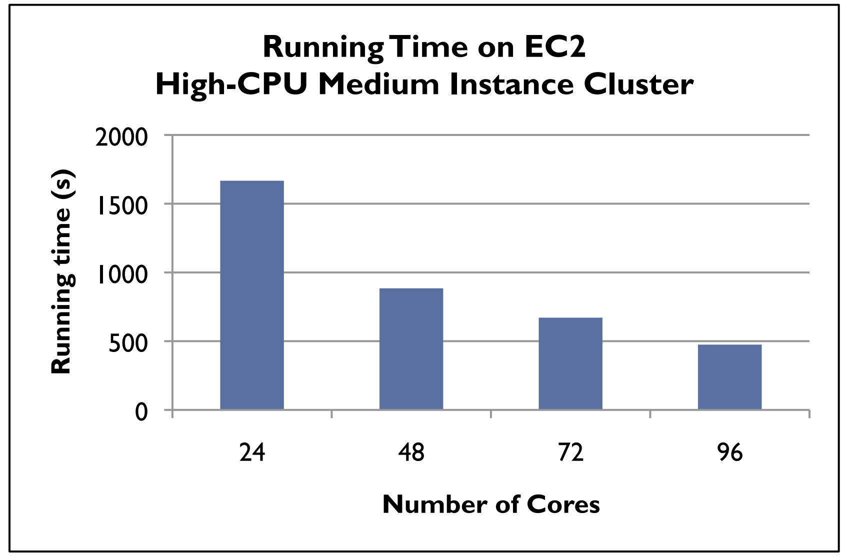
3. Reduce: End-to-end alignment
   - If read aligns end-to-end with ≤ k errors, record the alignment



EC2 Evaluation

http://cloudburst-bio.sourceforge.net





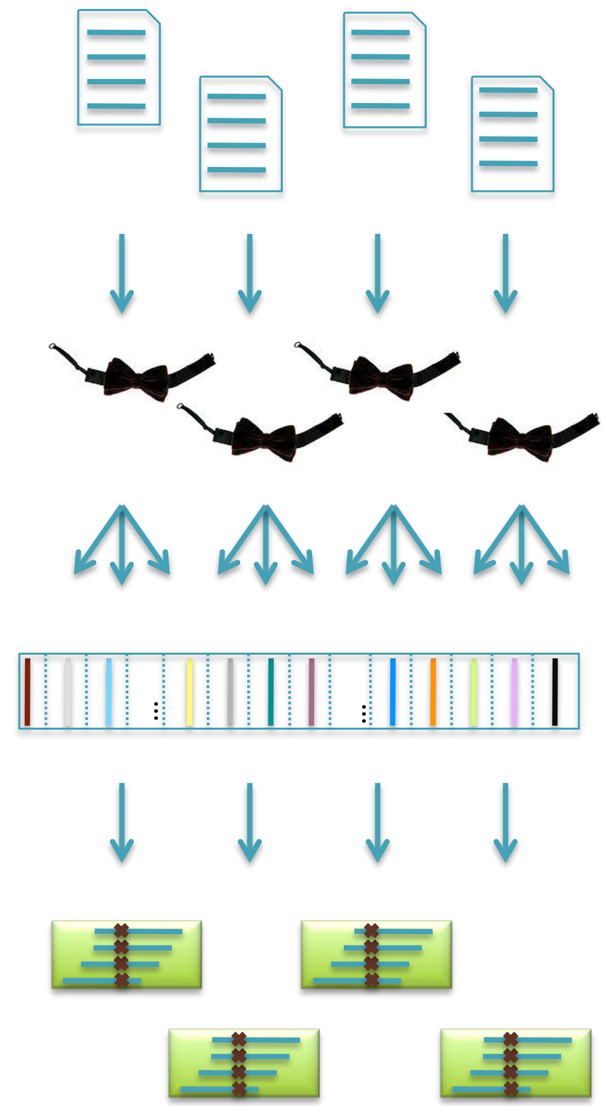
Evaluate mapping 7M reads to human chromosome 22 with at most 4 mismatches on a local and 2 EC2 clusters.

– 24-core High-CPU Medium Instance EC2 cluster is faster than 24-core local cluster.
– 96-core cluster is 3.5x faster than the 24-core, and 100x faster than serial RMAP.

CloudBurst: Highly Sensitive Read Mapping with MapReduce.
Schatz MC (2009) *Bioinformatics*. 25:1363-1369

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

Analyze an entire human genome for ~$100 in an afternoon.
Accuracy validated at >99%

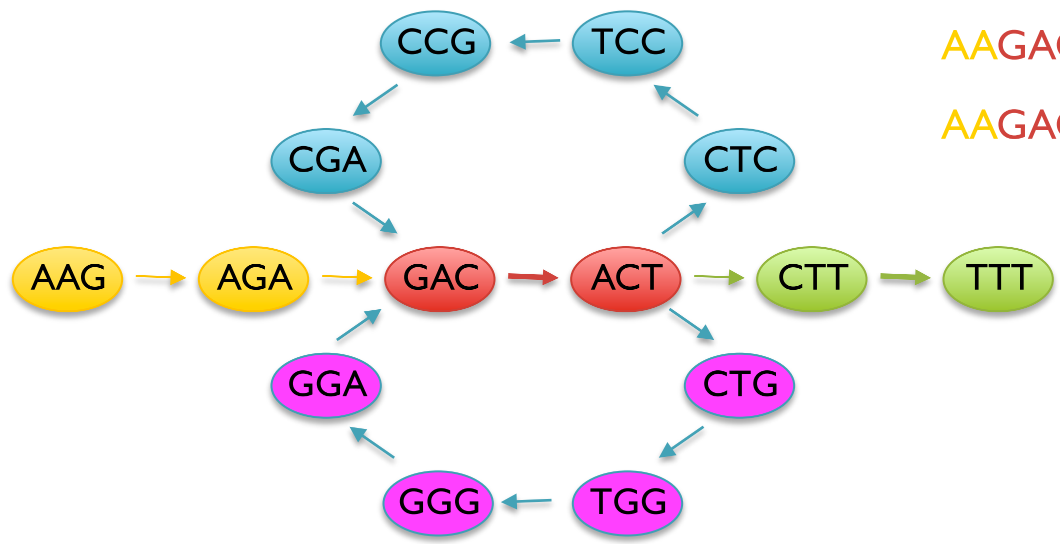
Searching for SNPs with Cloud Computing.
Langmead B, Schatz MC, Lin J, Pop M, Salzberg SL (2009) *Genome Biology*.

Short Read Assembly

<u>Reads</u>

AAGA
ACTT
ACTC
ACTG
AGAG
CCGA
CGAC
CTCC
CTGG
CTTT
…

<u>de Bruijn Graph</u>



<u>Potential Genomes</u>

AAGACTCCGACTGGGACTTT

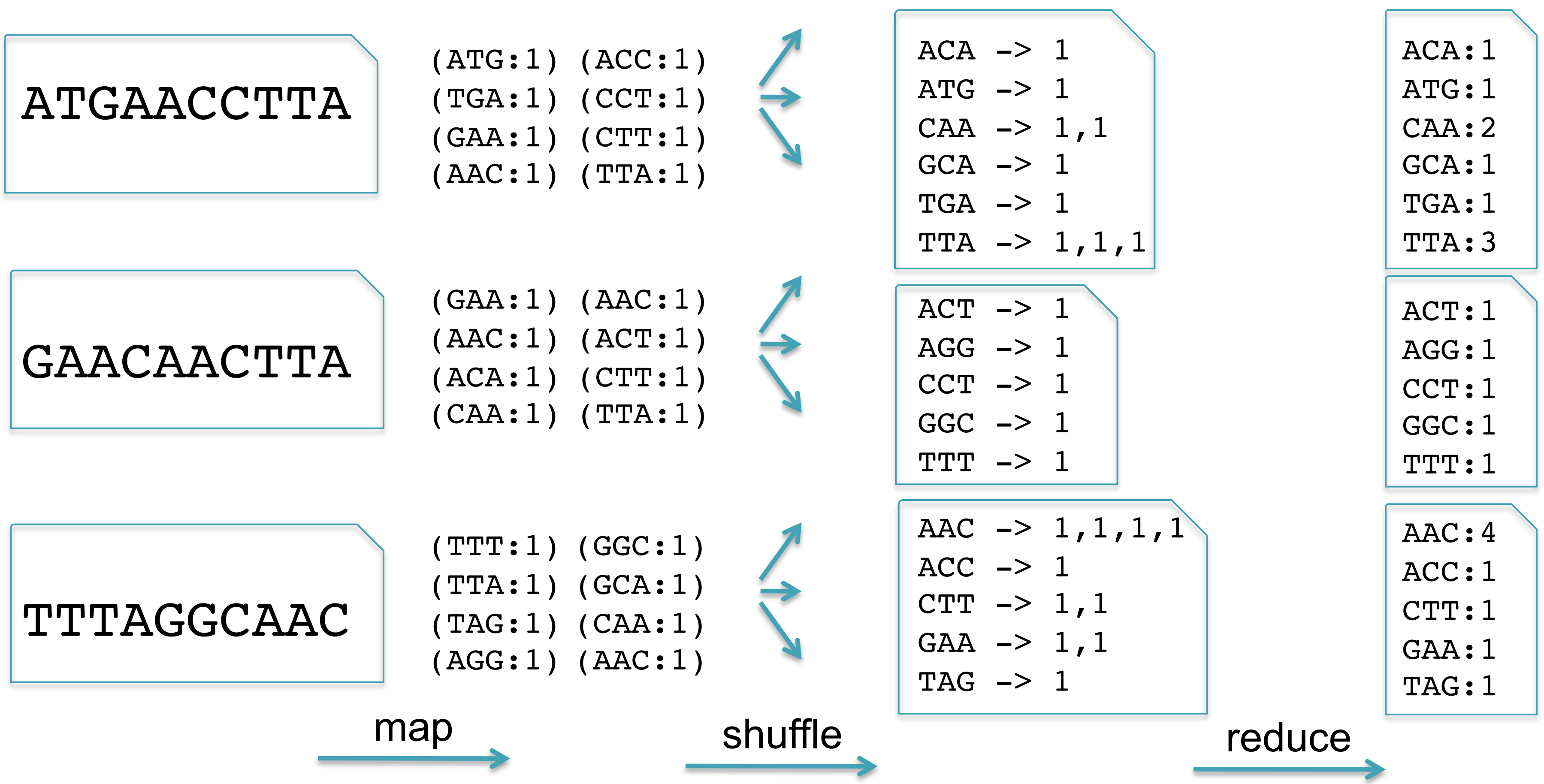
AAGACTGGGACTCCGACTTT

- Genome assembly as finding an Eulerian tour of the de Bruijn graph
  - Human genome: >3B nodes, >10B edges
- The new short read assemblers require tremendous computation
  - Velvet (Zerbino & Birney, 2008) serial: > 2TB of RAM
  - ABySS (Simpson *et al.*, 2009) MPI: 168 cores x ~96 hours
  - SOAPdenovo (Li *et al.*, 2010) pthreads: 40 cores x 40 hours, >140 GB RAM

K-mer Counting

- Application developers focus on 2 (+1 internal) functions
  - Map: input ➔ key:value pairs
  - Shuffle: Group together pairs with same key
  - Reduce: key, value-lists ➔ output

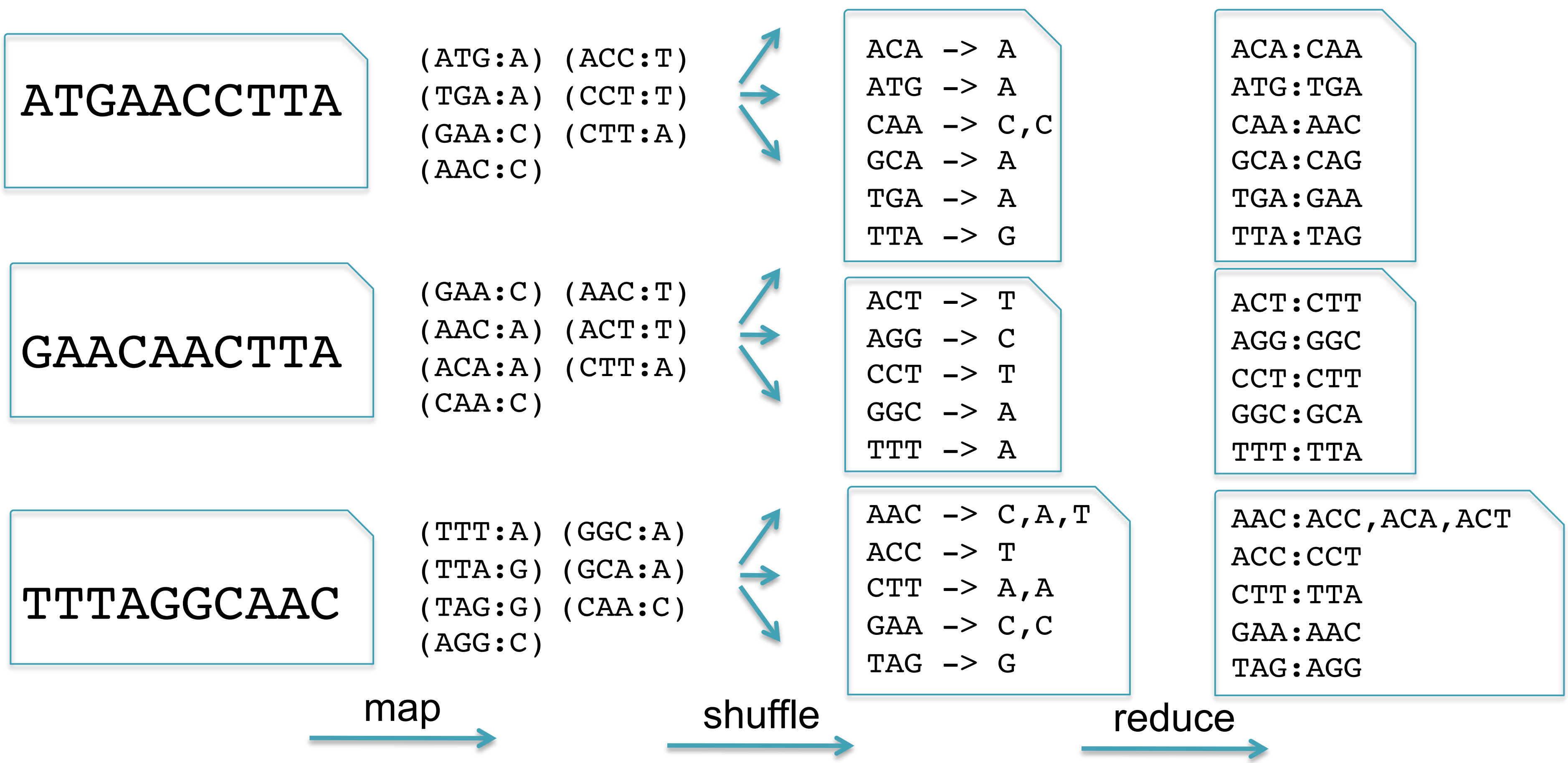
Map, Shuffle & Reduce
All Run in Parallel



Graph Construction

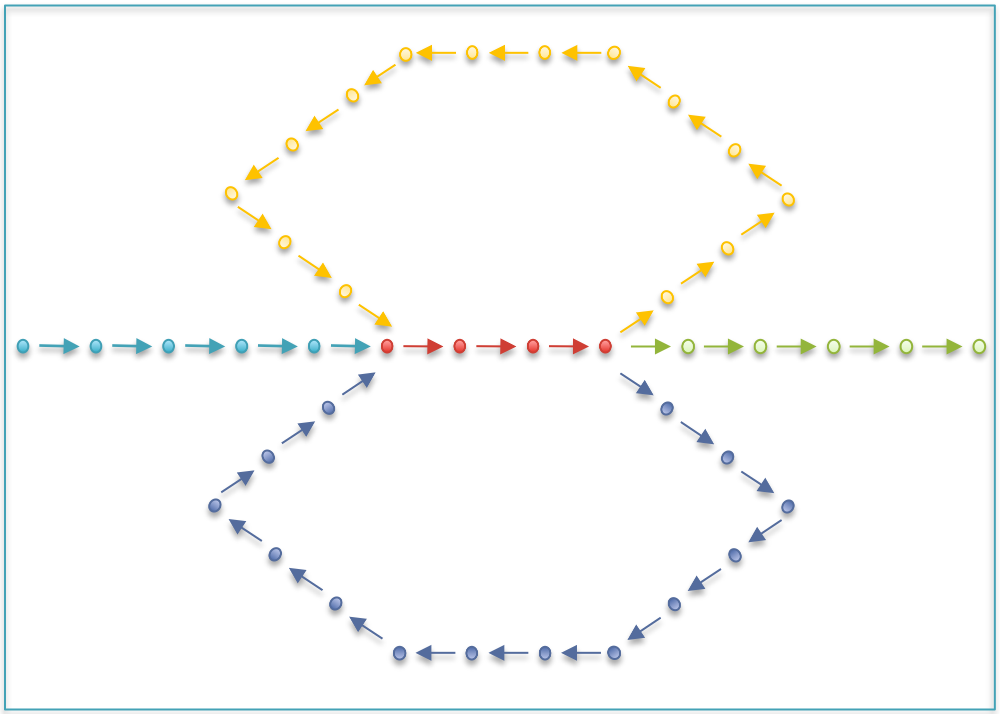
- Application developers focus on 2 (+1 internal) functions
  - Map: input ➔ key:value pairs
  - Shuffle: Group together pairs with same key
  - Reduce: key, value-lists ➔ output

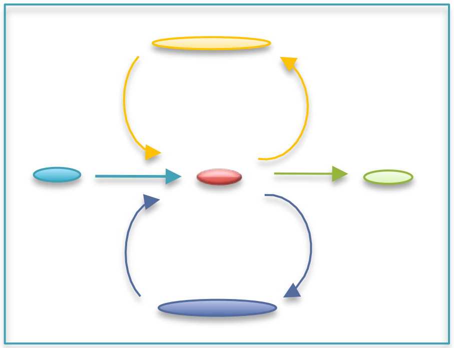
Map, Shuffle & Reduce
All Run in Parallel



Graph Compression

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

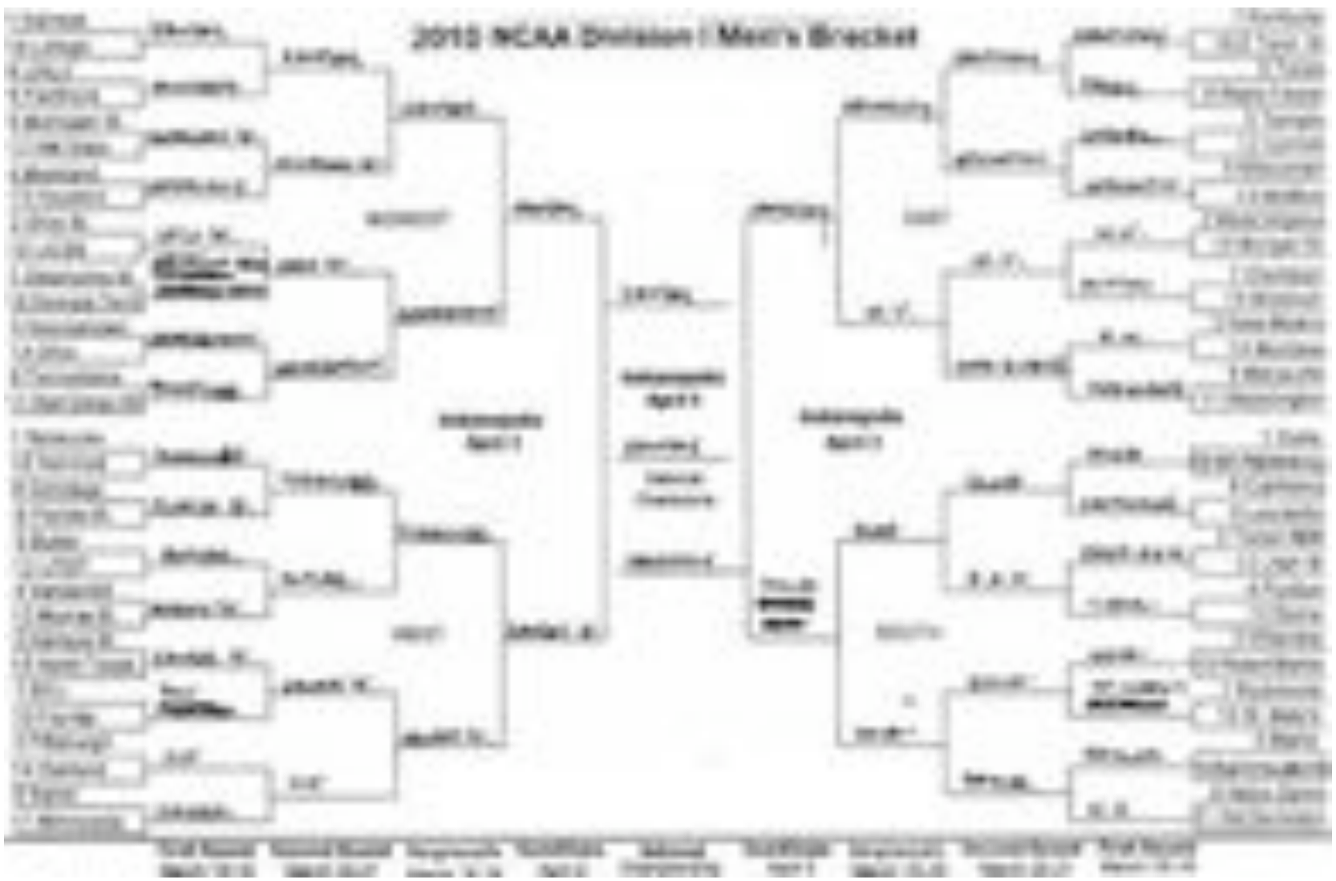




High School Dance

Warmup Exercise

- Who here was born closest to April 22?
  - You can only compare to 1 other person at a time



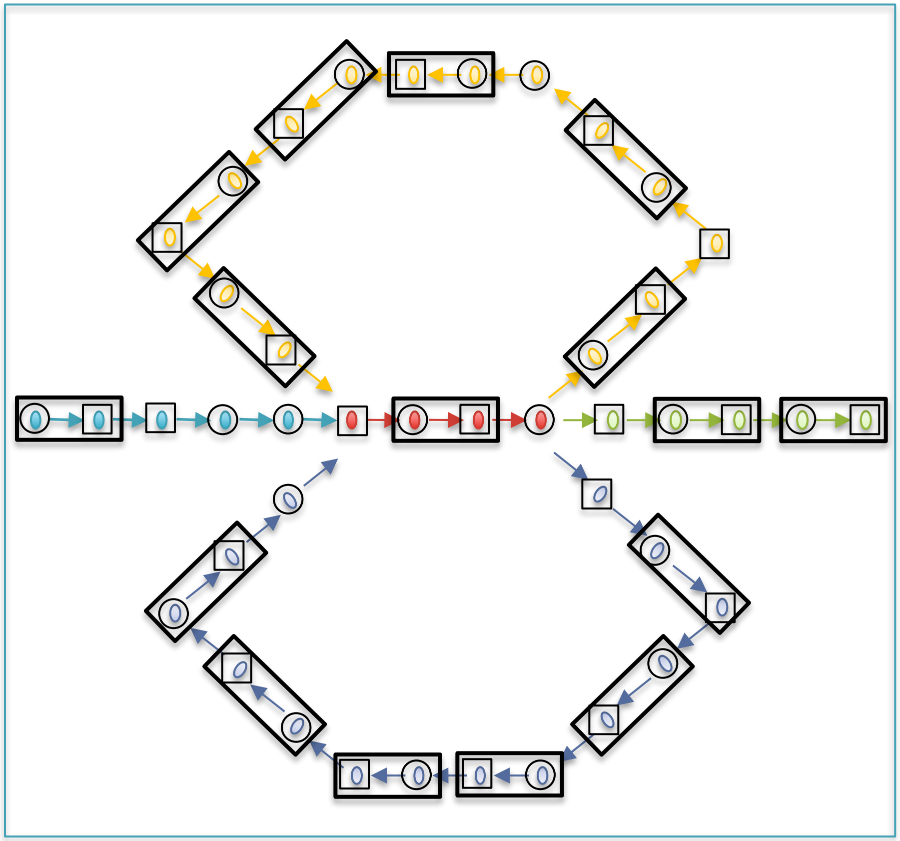
Fast Path Compression

Challenges

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

Randomized List Ranking

- Randomly assign H / T to each compressible node
- Compress H → T links



Initial Graph: 42 nodes

Randomized Speed-ups in Parallel Computation.
Vishkin U. (1984) *ACM Symposium on Theory of Computation. 230-239.*

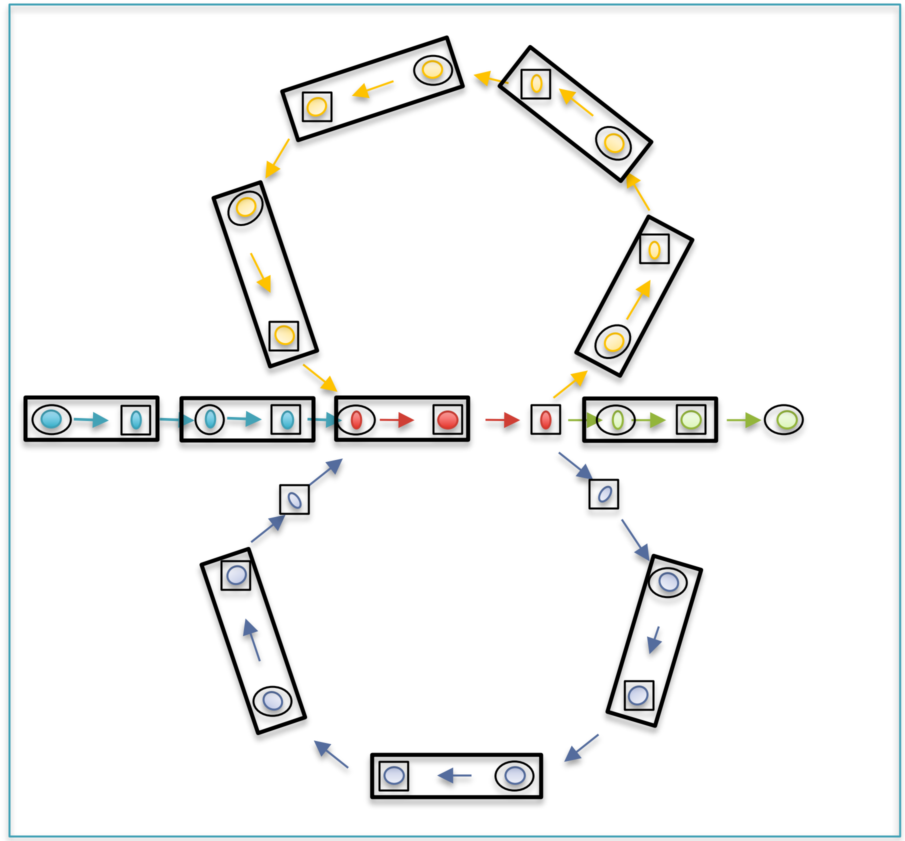
Fast Path Compression

Challenges

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

Randomized List Ranking

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



Round 1: 26 nodes (38% savings)

Randomized Speed-ups in Parallel Computation.
Vishkin U. (1984) *ACM Symposium on Theory of Computation. 230-239.*

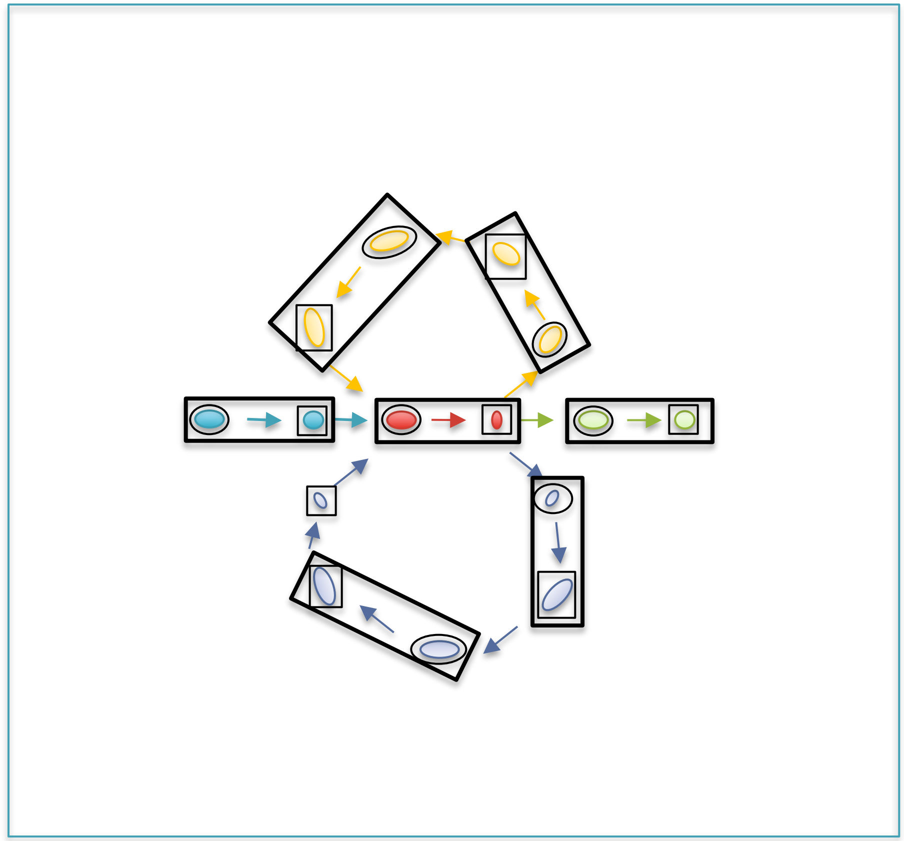
Fast Path Compression

Challenges

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

Randomized List Ranking

- Randomly assign H / T to each compressible node
- Compress H ➔ 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.*

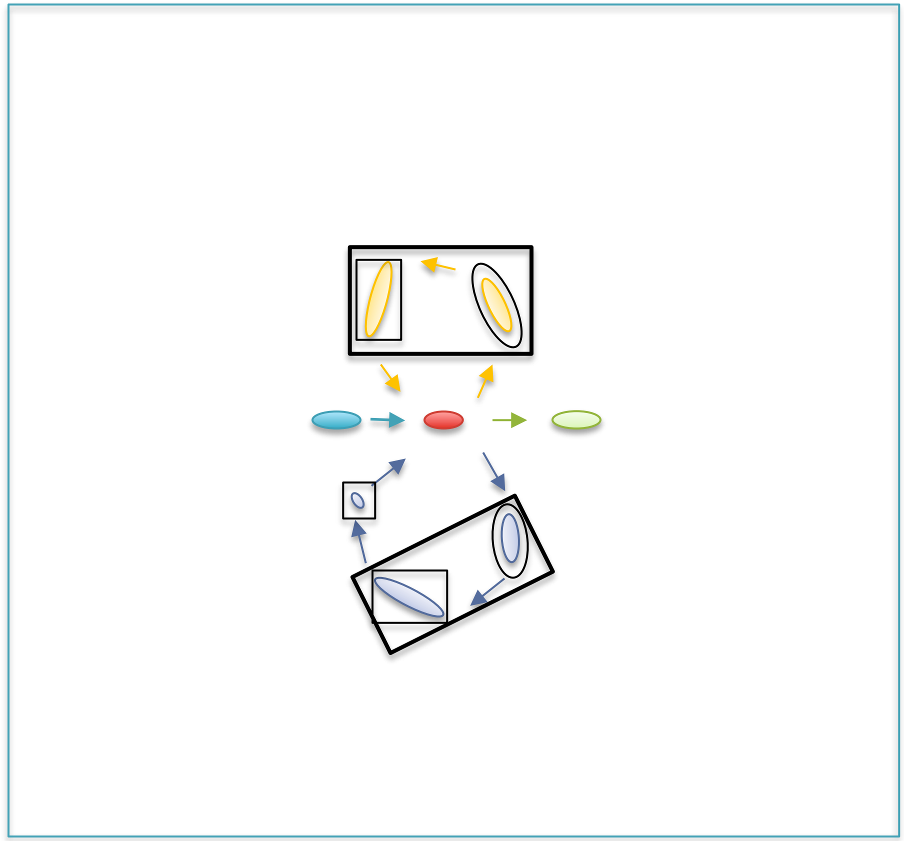
Fast Path Compression

Challenges

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

Randomized List Ranking

– Randomly assign H / $\boxed{T}$ to each compressible node
– Compress H → $\boxed{T}$ links



Round 2: 8 nodes (81% savings)

Randomized Speed-ups in Parallel Computation.
Vishkin U. (1984) *ACM Symposium on Theory of Computation. 230-239.*

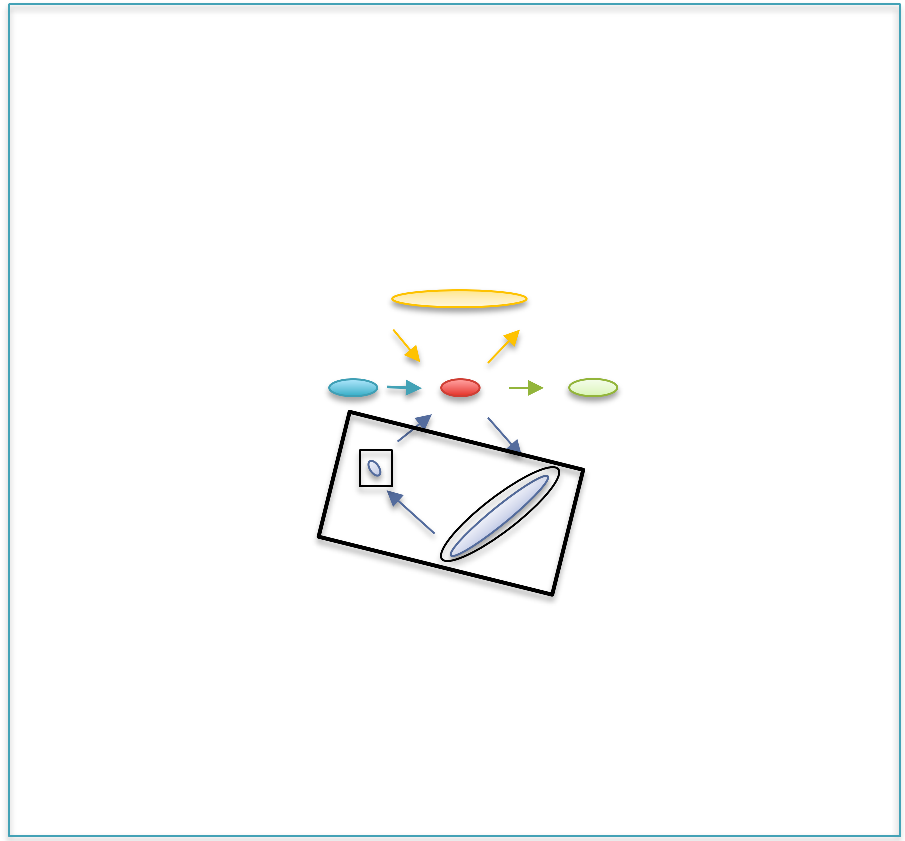
Fast Path Compression

Challenges

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

Randomized List Ranking

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

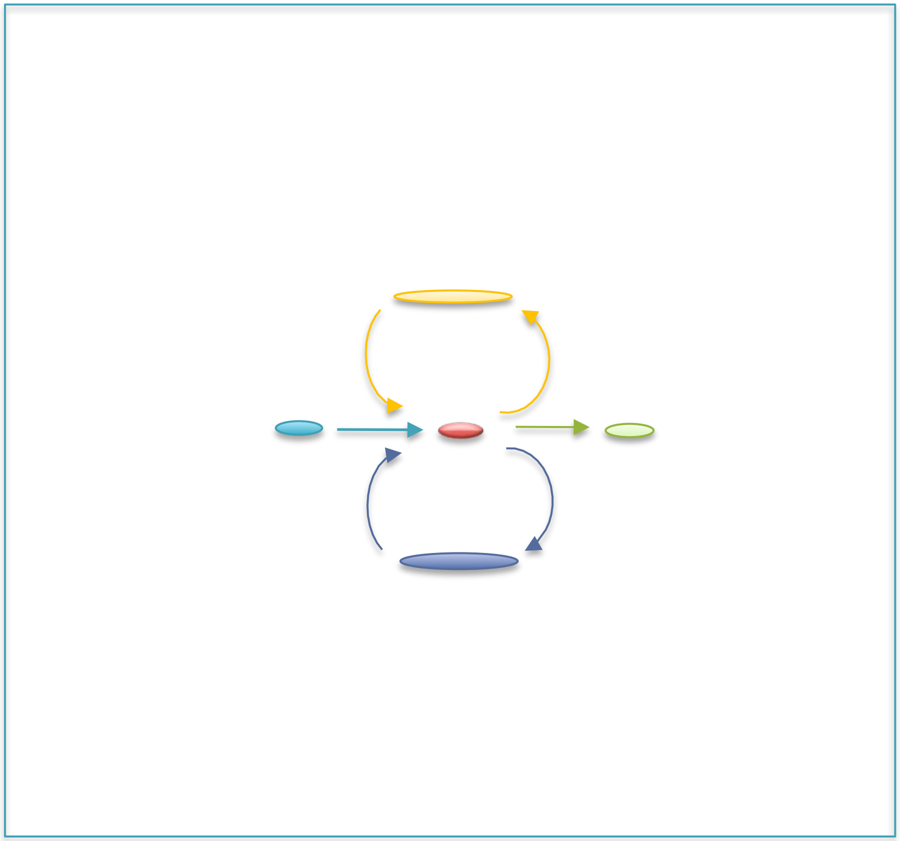
- Nodes stored on different computers
- Nodes can only access direct neighbors

Randomized List Ranking

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

Performance

- Compress all chains in log(S) rounds
- If <1024 nodes to compress then assign them all to the same reducer
  - Save last 10 rounds

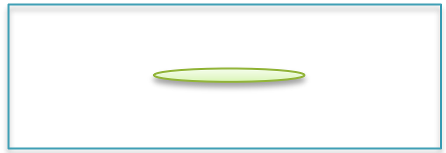


Round 4: 5 nodes (88% savings)

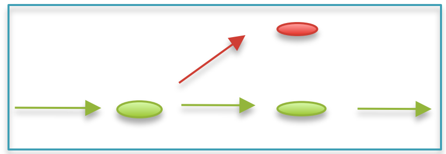
Randomized Speed-ups in Parallel Computation.
Vishkin U. (1984) *ACM Symposium on Theory of Computation. 230-239.*

Node Types

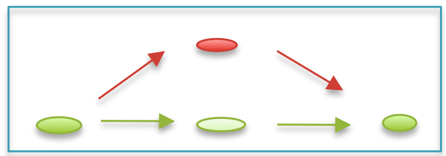
(Chaisson, 2009)



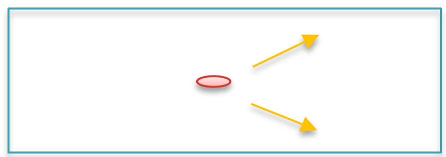
Isolated nodes (10%)



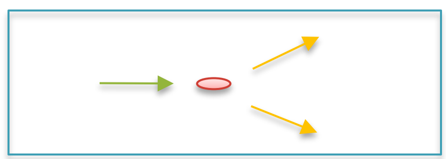
Tips (46%)



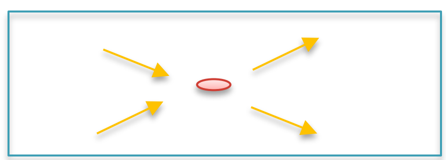
Bubbles/Non-branch (9%)



Dead Ends (.2%)



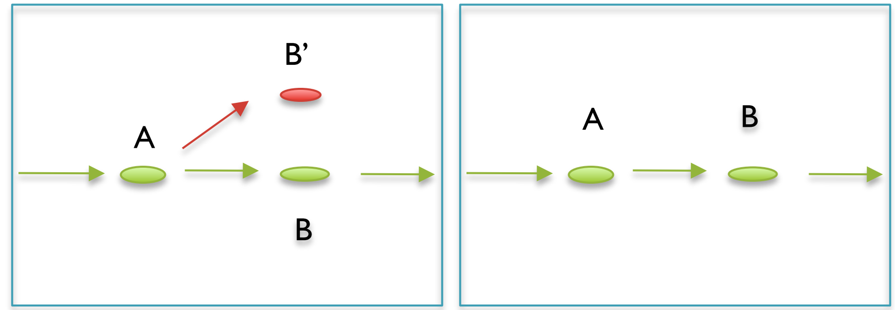
Half Branch (25%)



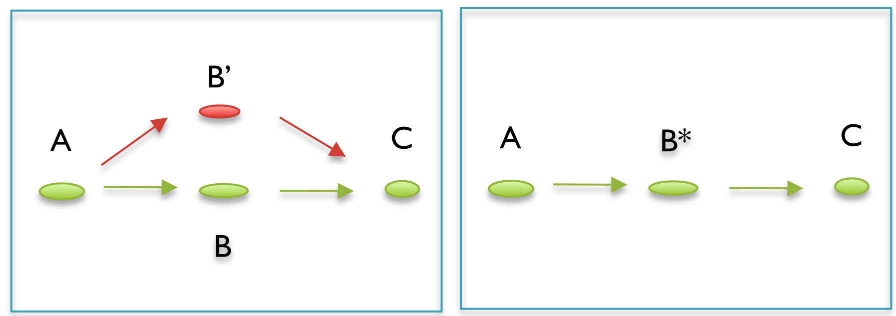
Full Branch (10%)

Error Correction

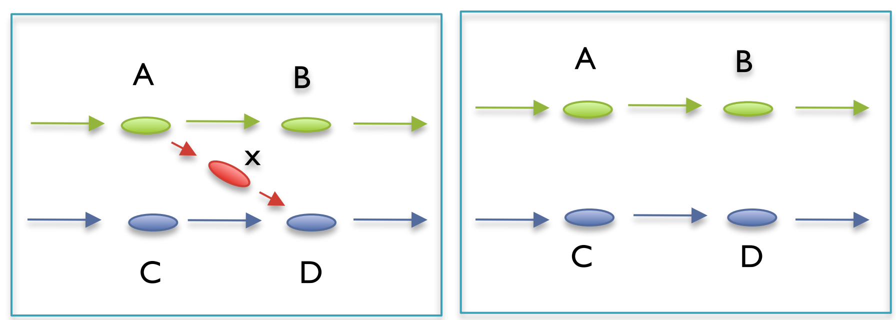
- Errors at end of read
  - Trim off ‘dead-end’ tips



- Errors in middle of read
  - Pop Bubbles



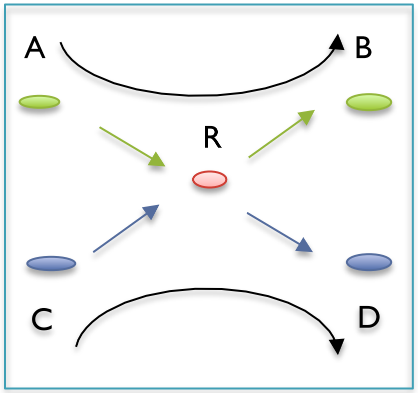
- Chimeric Edges
  - Clip short, low coverage nodes

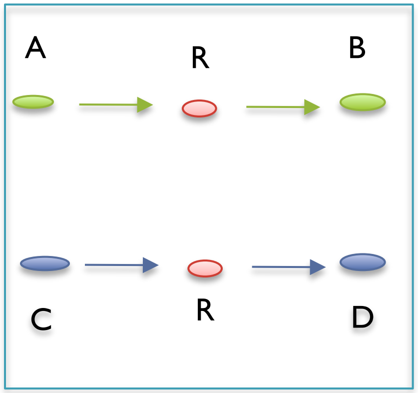


Parallel Network Motif Finding

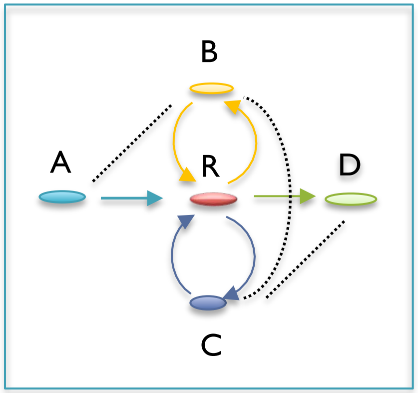
Repeat Analysis

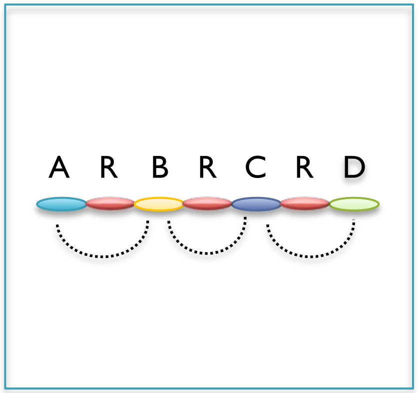
- X-cut
  - Annotate edges with spanning reads
  - Separate fully spanned nodes
    - (Pevzner *et al.*, 2001)





- Scaffolding
  - If mate pairs are available search for a path consistent with mate distance
  - Use message passing to iteratively collect linked and neighboring nodes





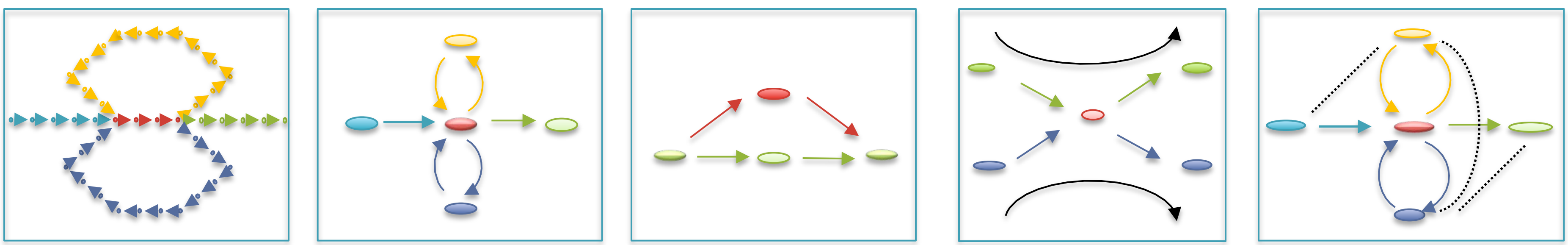
Parallel Frontier Search

Contrail

http://contrail-bio.sourceforge.net

Scalable Genome Assembly with MapReduce

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



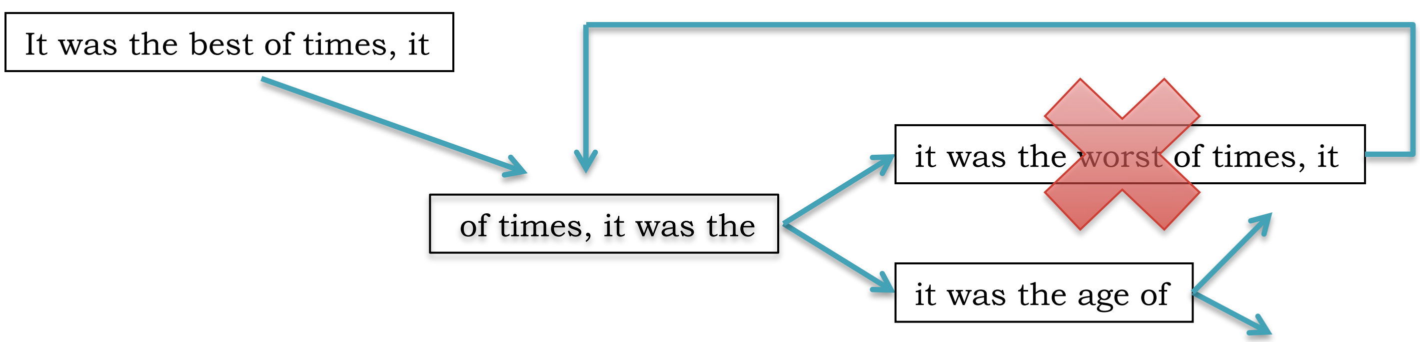
	Initial	Compressed	Error Correction	Resolve Repeats	Cloud Surfing
N	5.1 M	245,131	2,769	1,909	300
Max	27 bp	1,079 bp	70,725 bp	90,088 bp	149,006 bp
N50	27 bp	156 bp	15,023 bp	20,062 bp	54,807 bp

Assembly of Large Genomes with Cloud Computing.
Schatz MC, Sommer D, Kelley D, Pop M, *et al. In Preparation.*

E. coli Assembly Quality

Incorrect contigs: Align at < 95% identity or < 95% of their length

Assembler	Contigs ≥ 100bp	N50 (bp)	Incorrect contigs
Contrail PE	300	54,807	4
Contrail SE	529	20,062	0
SOAPdenovo PE	182	89,000	5
ABySS PE	233	45,362	13
Velvet PE	286	54,459	9
EULER-SR PE	216	57,497	26
SSAKE SE	931	11,450	38
Edena SE	680	16,430	6



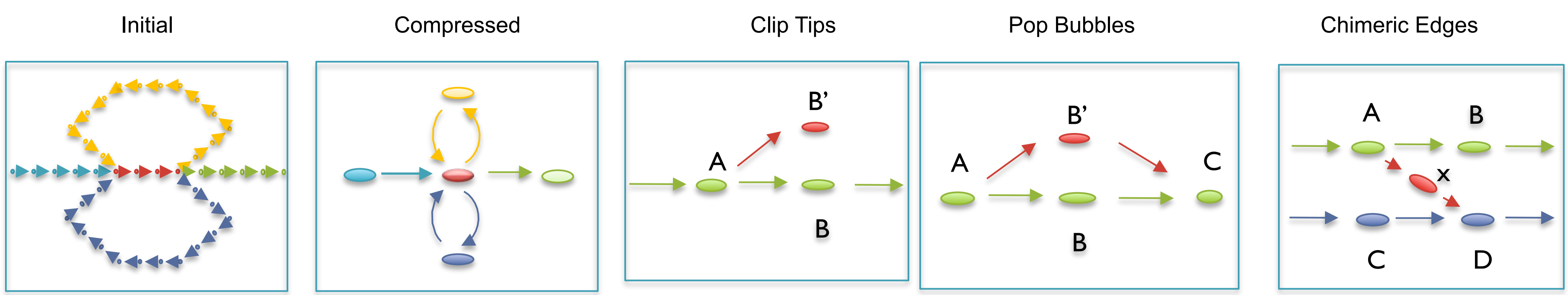

One more thing...

Contrail

http://contrail-bio.sourceforge.net

De Novo Assembly of the Human Genome

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



	Initial	Compressed	Clip Tips	Pop Bubbles	Chimeric Edges
N	>7 B	>1 B	5.0 M	4.2 M	
Max	27 bp	303 bp	14,007 bp	20,594 bp	In progress
N50	27 bp	< 100 bp	650 bp	923 bp	

Assembly of Large Genomes with Cloud Computing.
Schatz MC, Sommer D, Kelley D, Pop M, *et al. In Preparation.*

Summary

"NextGen sequencing has completely outrun the ability of good bioinformatics people to keep up with the data and use it well… We need a MASSIVE effort in the development of tools for "normal" biologists to make better use of massive sequence databases."

Jonathan Eisen – JGI Users Meeting – 3/28/09

- Surviving the data deluge means computing in parallel
  - Good solutions for "easy" parallel problems, but gets fundamentally more difficult as dependencies get deeper
- Emerging technologies are a great start, but we need continued research integrating computational biology with research in HPC
  - A word of caution: new technologies are new

Acknowledgements

Advisor

Steven Salzberg

UMD Faculty

Mihai Pop, Art Delcher, Amitabh Varshney, Carl Kingsford, Ben Shneiderman, James Yorke, Jimmy Lin, Dan Sommer

CBCB Students

Adam Phillippy, Cole Trapnell, Saket Navlakha, Ben Langmead, James White, David Kelley

Thank You!

http://www.cbcb.umd.edu/~mschatz