# Computational Architecture of Cloud Environments

## Michael Schatz

# Cloud Architecture

→

**Input**

Computation

→

**Output**
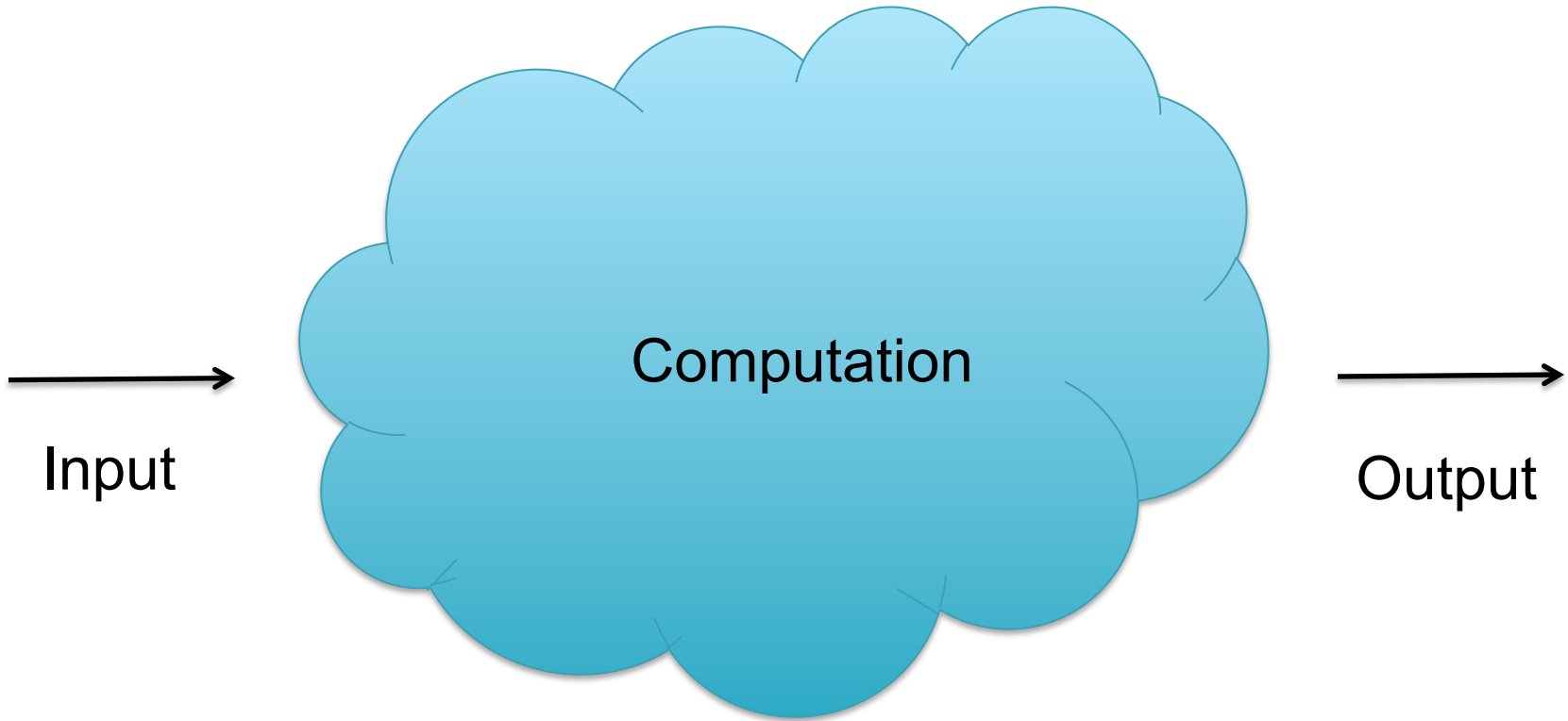
Nebulous question:

Cloud computing = Utility computing + Enabling Computational Technologies

Goal: Many computers working together to analyze huge datasets

Challenge: 100x processors rarely means 100x faster

# Why Parallel?

- Moore's Law is valid in 2010
  - But CPU speed is flat
  - Vendors switching to multicore solutions instead

- Why parallel
  - Need it done faster
  - Doesn't fit on one machine
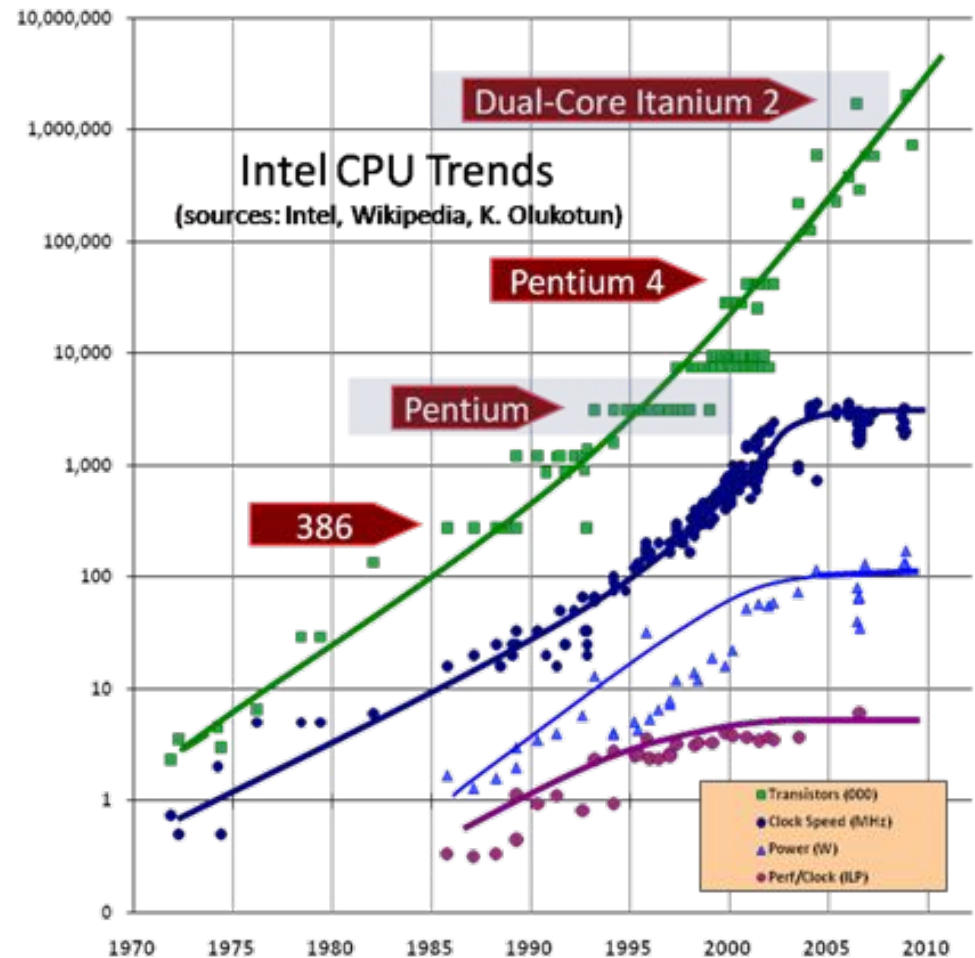


The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software
Herb Sutter, http://www.gotw.ca/publications/concurrency-ddj.htm

# Parallel Computing Spectrum

| Embarrassingly Parallel | Regularly Parallel | Deeply Parallel |
|---|---|---|



| Batch Computing Many Good Solutions | MapReduce Enabling Technologies Emerging | MPI & PRAM Open Research Area |
|---|---|---|

| BLAST HMM Scoring Parameter Sweep | Align-Shuffle-Scan Genotyping K-mer Counting | Graph Analysis Genome Assembly MD Simulations |
|---|---|---|

# Embarrassingly Parallel

- Batch computing
  - Each item is independent
  - Split input into many chunks
  - Process each chunk separately on a different computer

- Challenges
  - Distributing work, load balancing, monitoring & restart

- Technologies
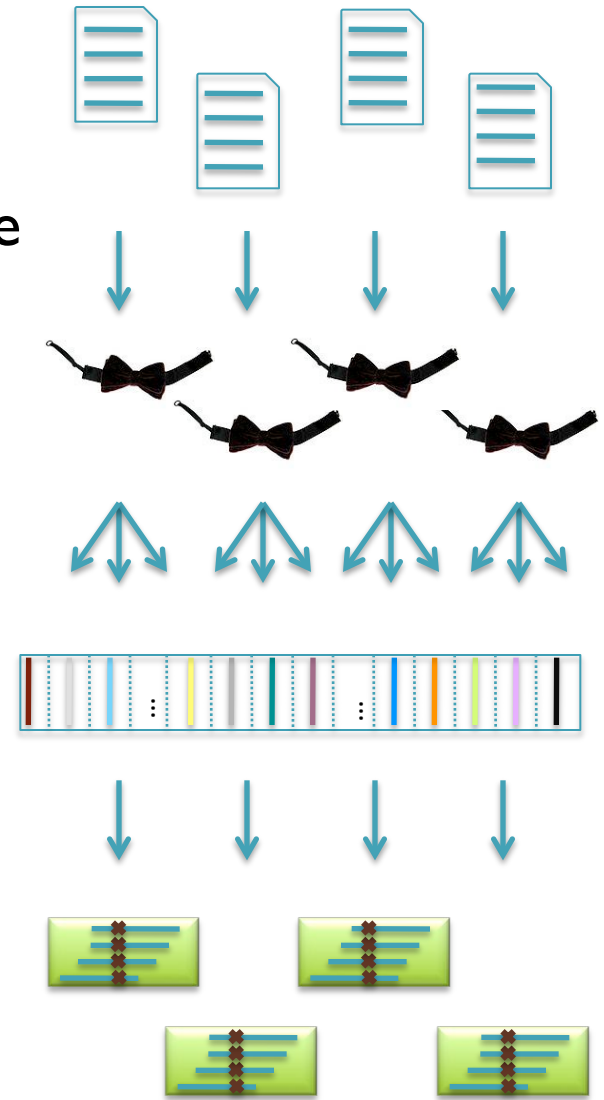  - Condor, Sun Grid Engine
  - Amazon Simple Queue
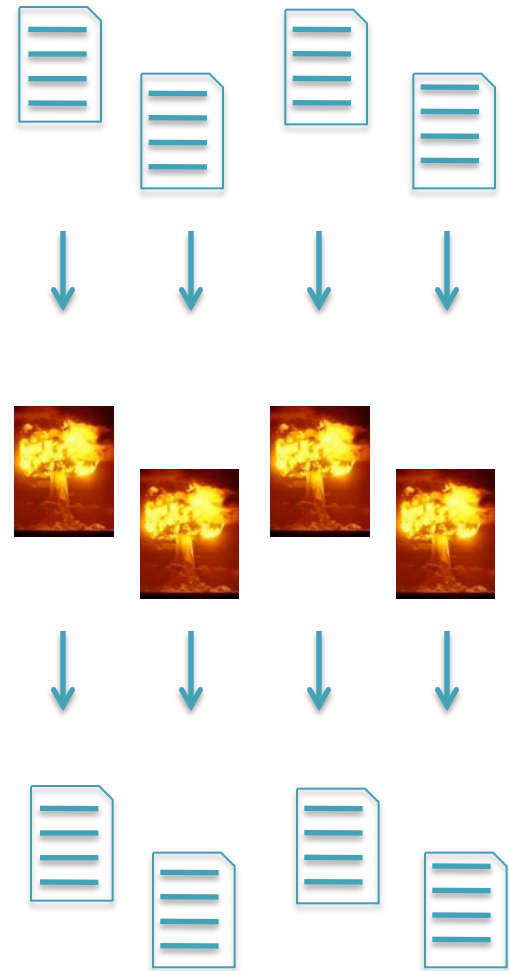
# Elementary School Dance
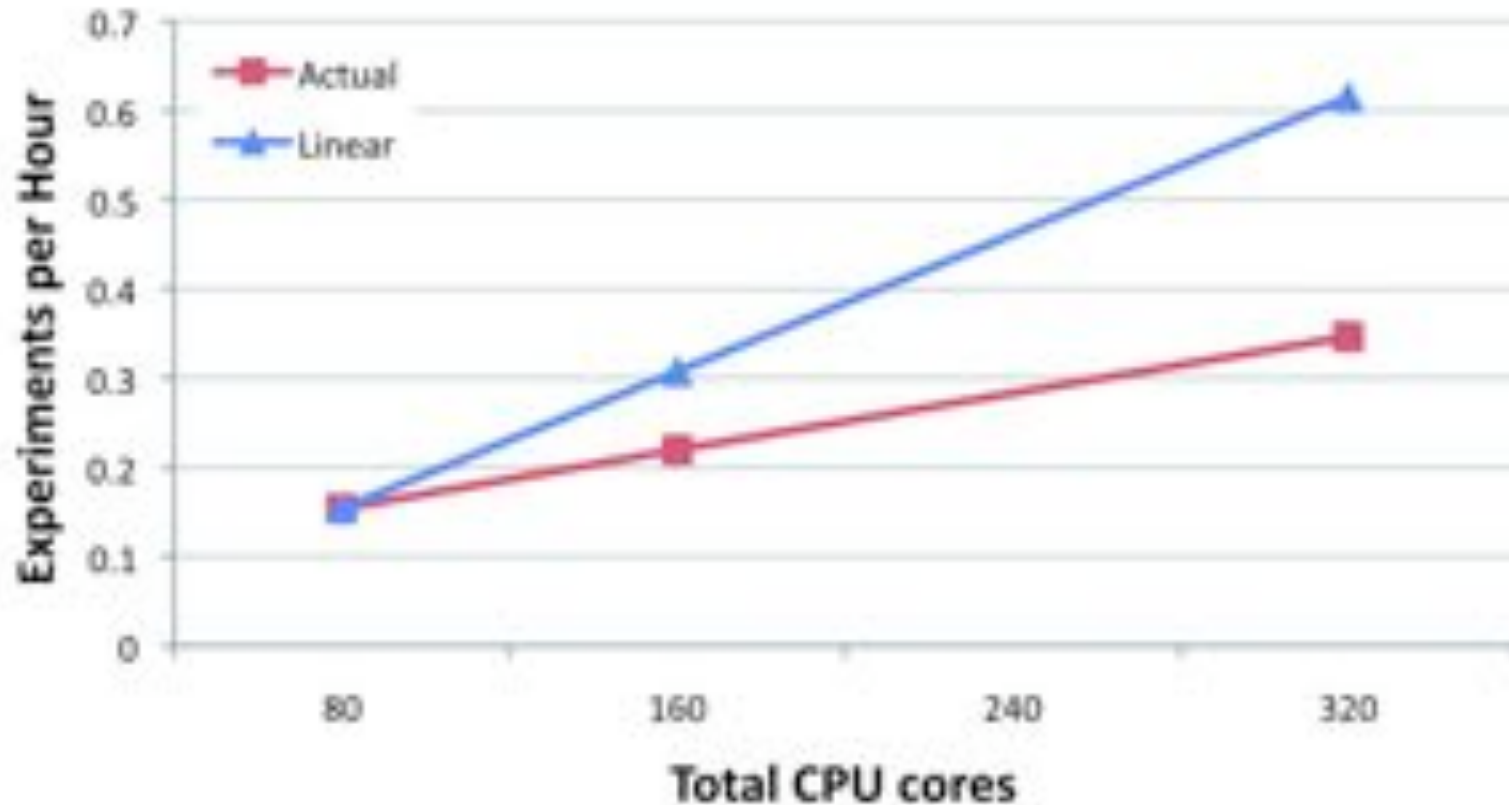
# Regularly Parallel

- Align-Shuffle-Scan in MapReduce
  - Align a large set of reads
  - Shuffle to group and sort by chromosome
  - Scan alignments for SNPs

- Challenges
  - Batch computing challenges
  - + Shuffling of huge datasets

- Technologies
  - Hadoop, Elastic MapReduce, Dryad
  - Parallel Databases

# Junior High Dance

# Crossbow Scaling



- Even with this relatively simple parallel application, we do not achieve perfect efficiency.
  - Interesting tradeoffs in time vs. money

# Deeply Parallel

- Computation that cannot be partitioned
  - Graph Analysis
  - Molecular Dynamics
  - Population simulations

- Challenges
  - Regular parallel challenges
  - + Parallel algorithms design

- Technologies
  - MPI
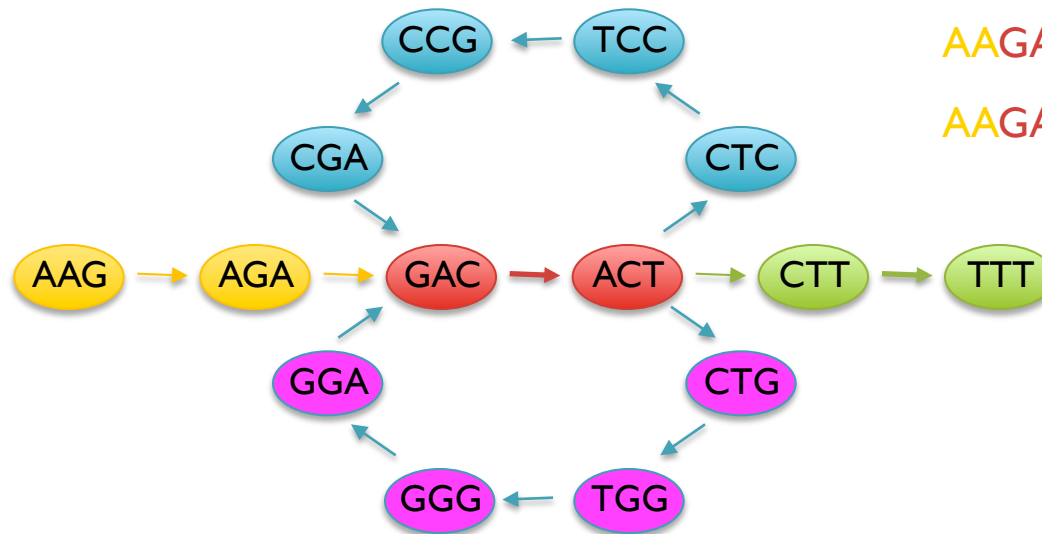  - MapReduce, Dryad
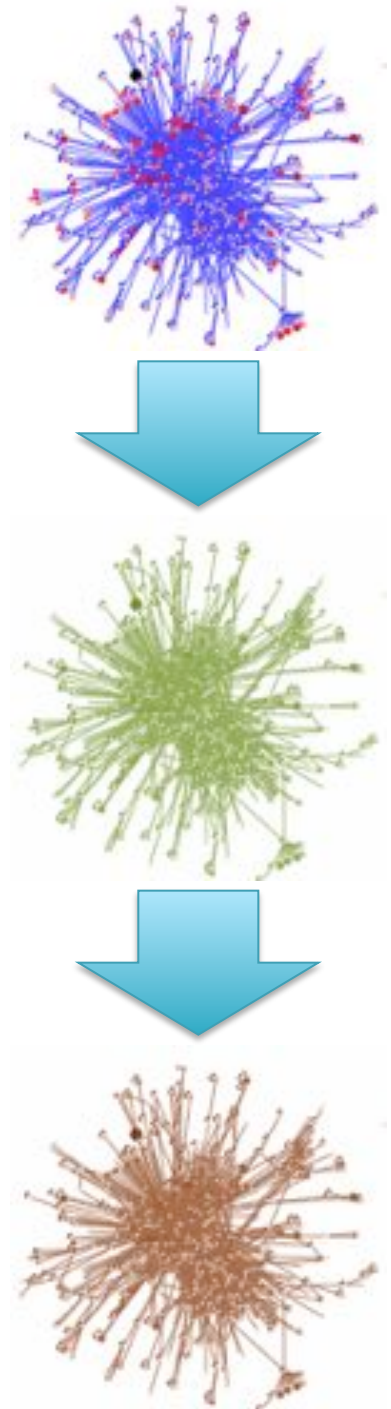
# High School Dance

# Short Read Assembly

**Reads**

AAGA
ACTT
ACTC
ACTG
AGAG
CCGA
CGAC
CTCC
CTGG
CTTT
...

**de Bruijn Graph**

**Potential Genomes**

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

# Graph Compression

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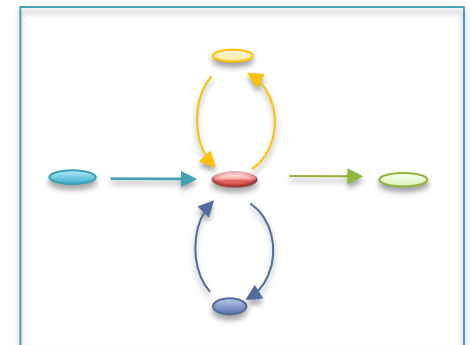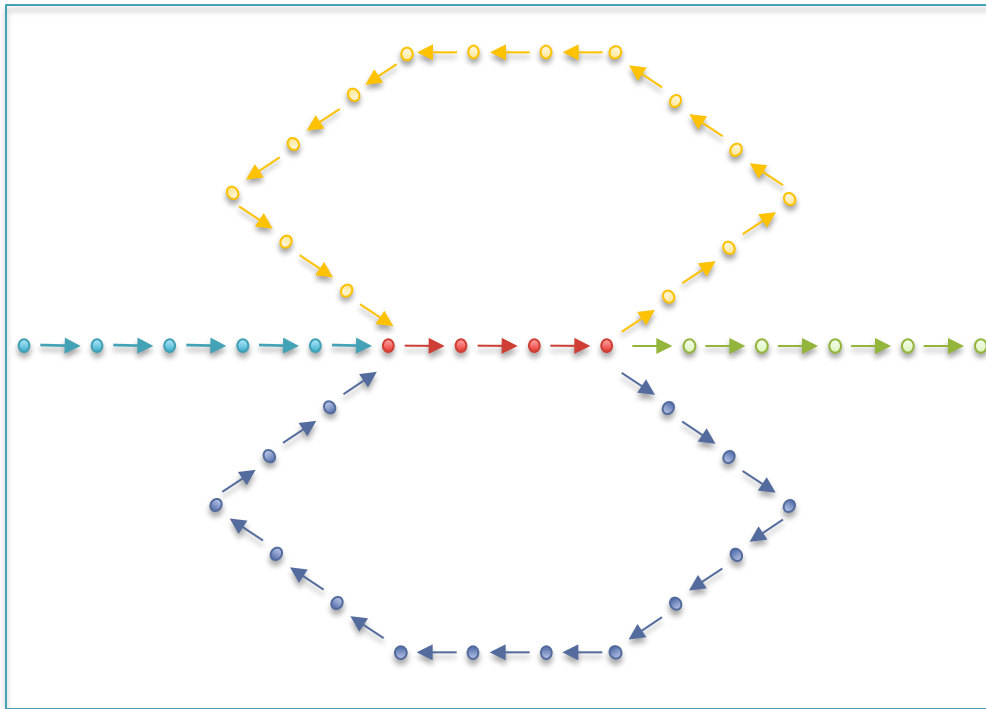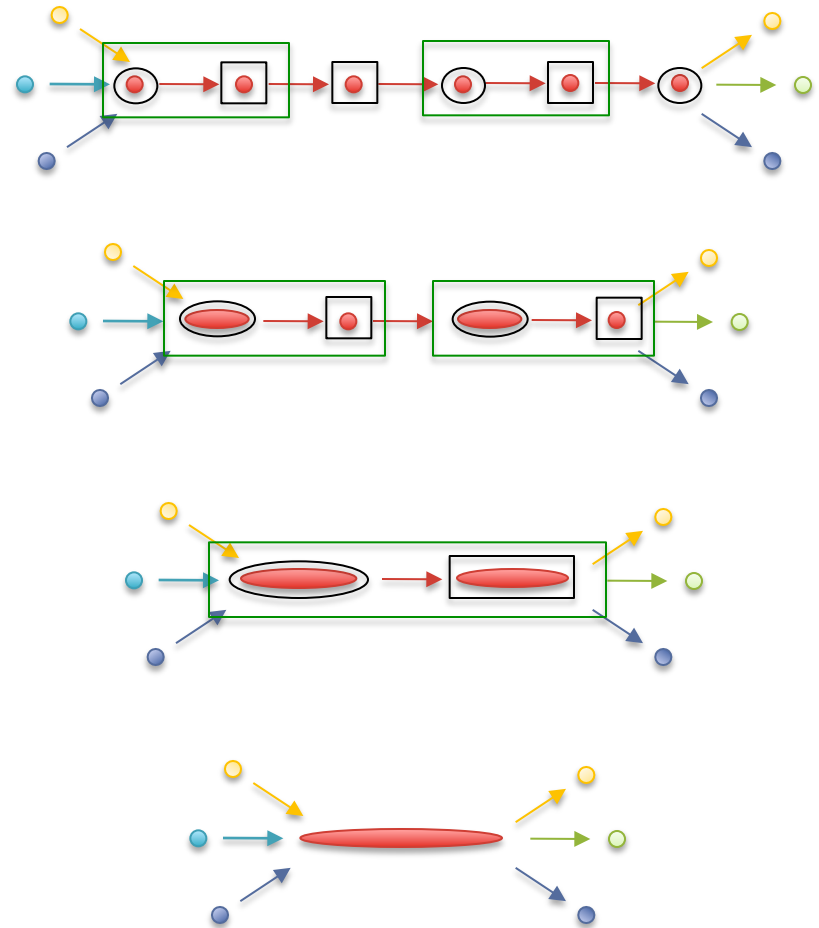
# Fast Path Compression

## Challenges

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

## Randomized List Ranking

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

## Performance

- Compress all chains in log(S) rounds

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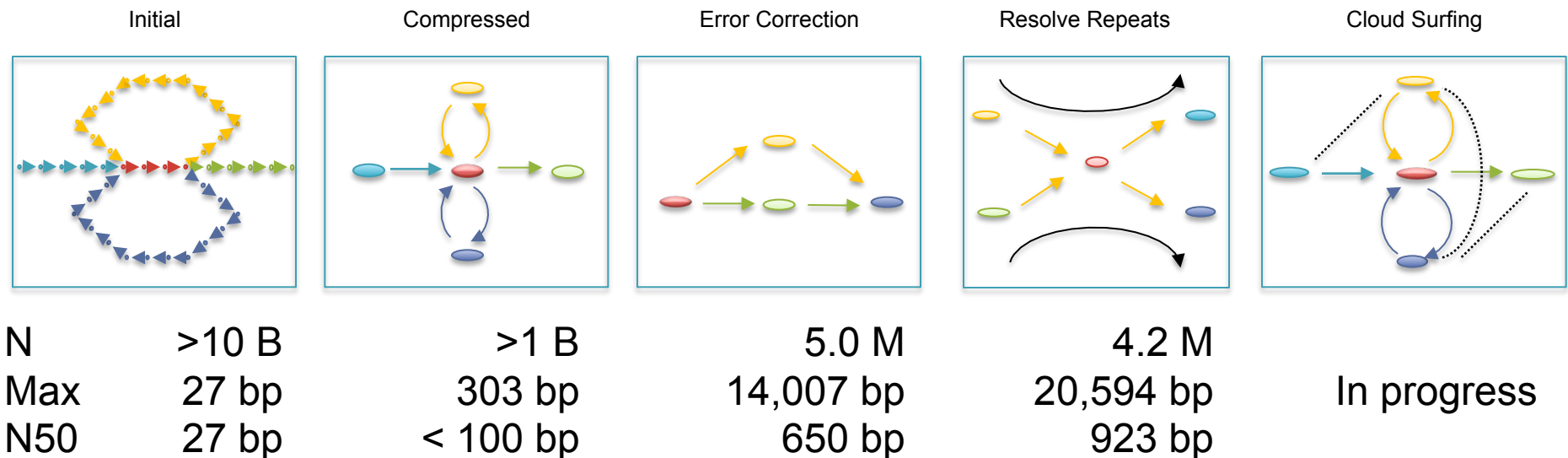Vishkin U. (1984) *ACM Symposium on Theory of Computation. 230-239.*

# Contrail

http://contrail-bio.sourceforge.net

## Scalable Genome Assembly with MapReduce

- *Genome:* African male NA18507 (Bentley *et al.*, 2008)
- *Input:* 3.5B 36bp reads, 210bp insert (SRA000271)
- *Preprocessor:* Quality-Aware Error Correction



| | Initial | Compressed | Error Correction | Resolve Repeats | Cloud Surfing |
|---|---|---|---|---|---|
| N | >10 B | >1 B | 5.0 M | 4.2 M | In progress |
| Max | 27 bp | 303 bp | 14,007 bp | 20,594 bp | |
| 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

- Surviving the data deluge means computing in parallel
  - Good solutions for "easy" parallel problems, but gets fundamentally more difficult as dependencies get deeper

- Parallel systems require connecting many components
  - We can get started by agreeing on common input and output formats, open source software
  - Move the computation to the data

- Emerging technologies are a great start, but we need continued research integrating computational biology with research in HPC
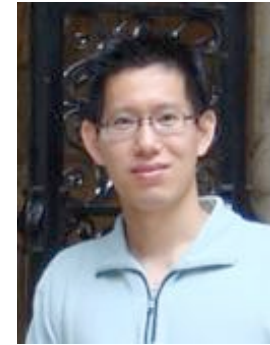  - A word of caution: new technologies are new

# Acknowledgements
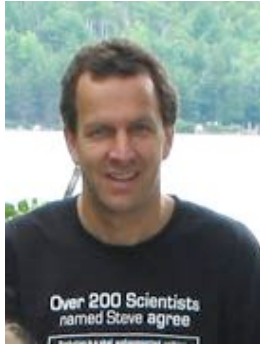
Steven Salzberg

Mihai Pop

Jimmy Lin

Ben Langmead

Dan Sommer

David Kelley

amazon
web services™

# Thank You!

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

@mike_schatz