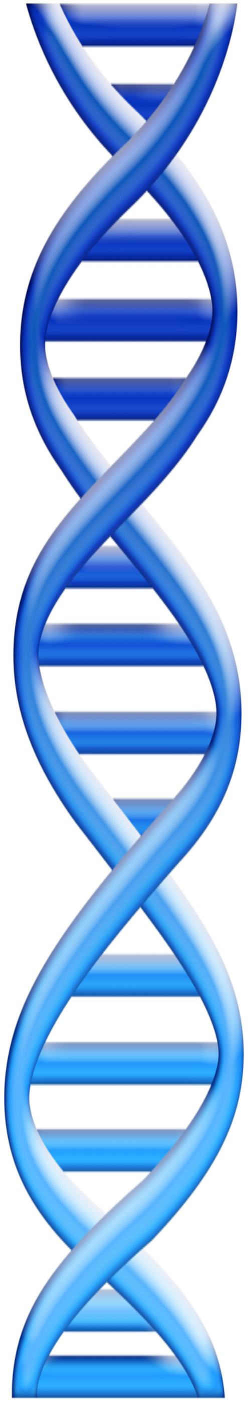
Cloud Computing and the DNA Data Race

Michael Schatz

October 22, 2010
CSHL Advanced Sequencing Course



Outline

Part 1: Theory

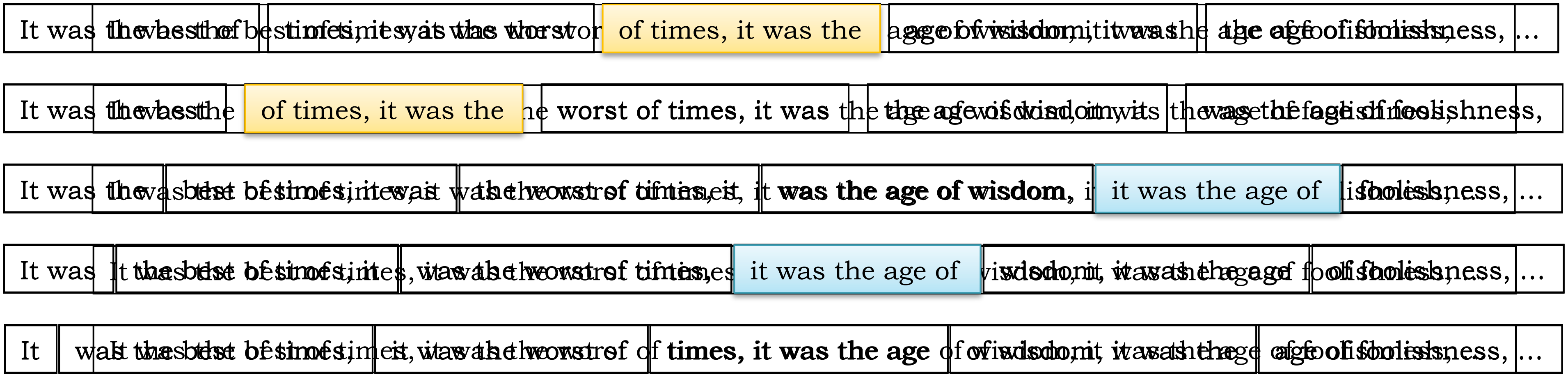
1. Genome Assembly by Analogy
2. DNA Sequencing and Genomics
3. Sequence Analysis in the Clouds
    1. Sequence Alignment
    2. Mapping & Genotyping
    3. Genome Assembly

Part 2: Practice

1. AWS Mini-Tutorial
2. Hadoop Mini-Tutorial

Shredded Book Reconstruction

- Dickens accidentally shreds the first printing of <u>A Tale of Two Cities</u>
  - 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,

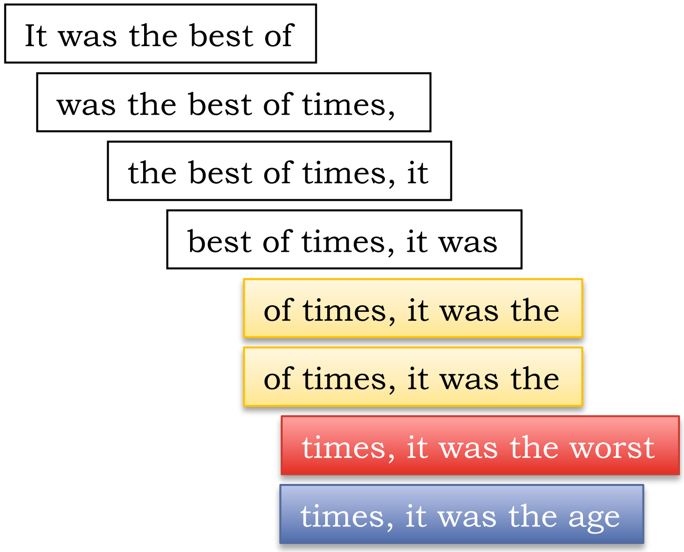
was the age of foolishness,

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

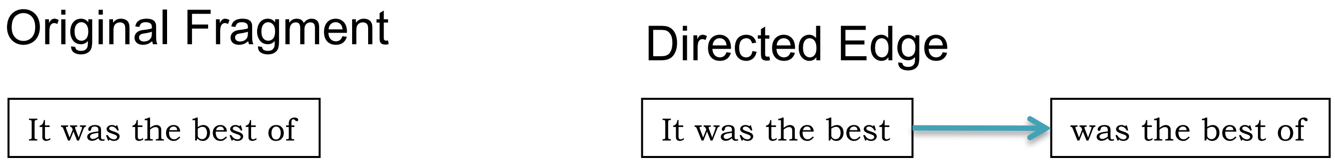
The repeated sequence make the correct reconstruction ambiguous

- 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

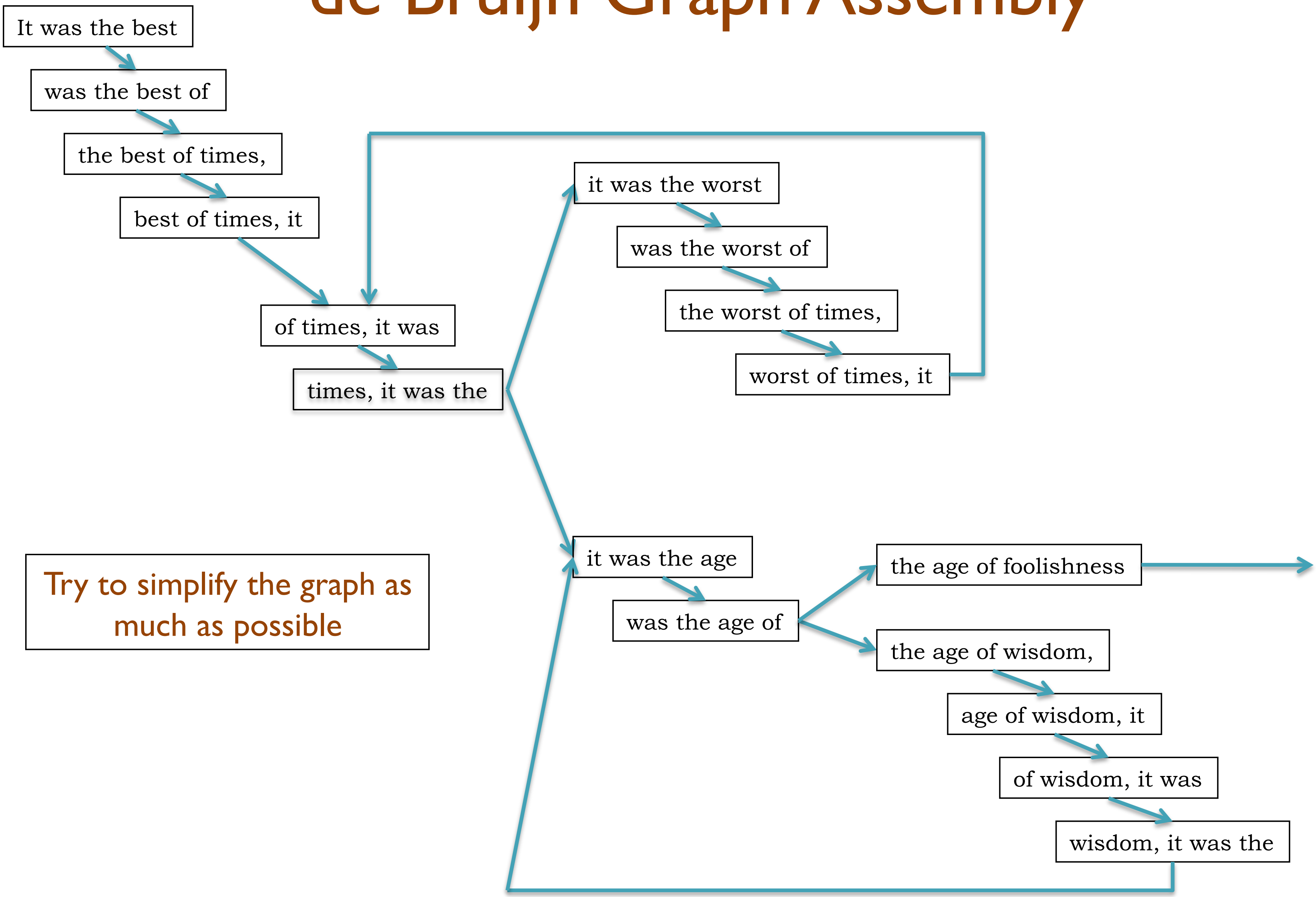
It was the best

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



It was the best

was the best of

the best of times,

best of times, it

of times, it was

times, it was the

it was the worst

was the worst of

the worst of times,

worst of times, it

it was the age

was the age of

the age of foolishness

the age of wisdom,

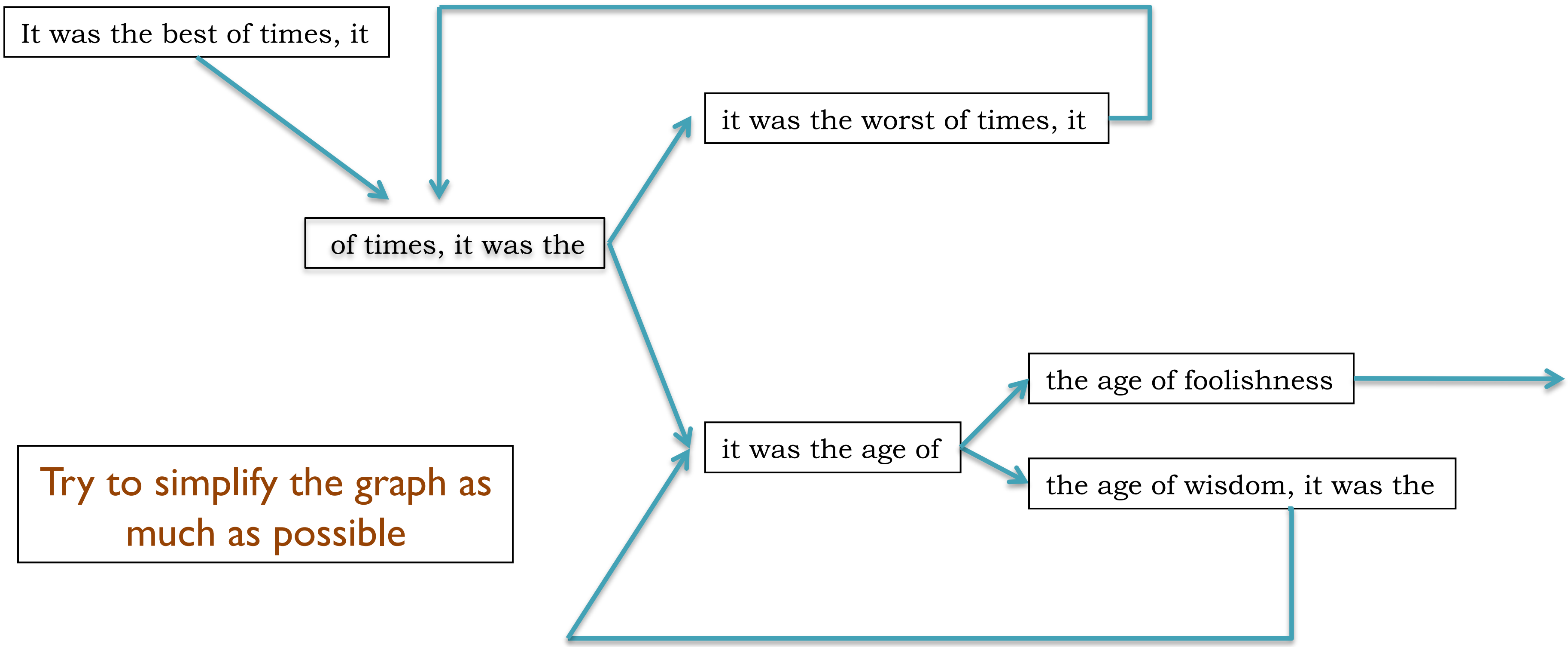
age of wisdom, it

of wisdom, it was

wisdom, it was the

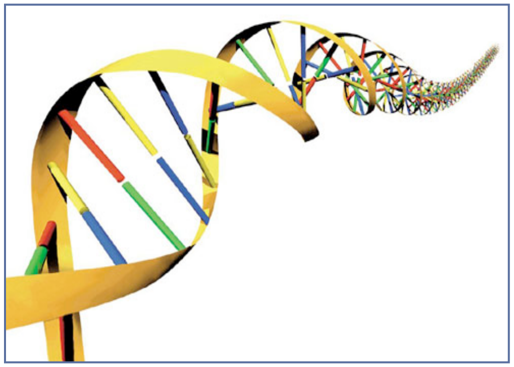
Try to simplify the graph as much as possible

de Bruijn Graph Assembly


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

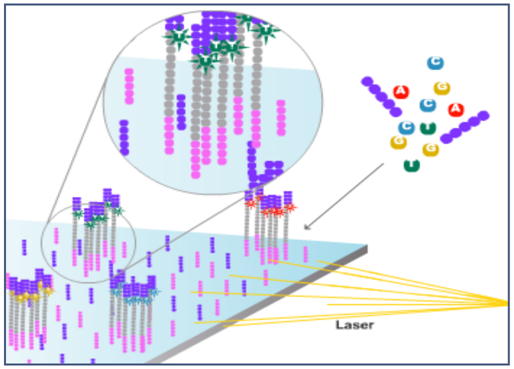
Try to simplify the graph as much as possible

Molecular Biology & 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 sequence millions of short (25-500bp) reads from random positions of the genome

- 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.

- Use software to analyze the sequences
- Modern Biology requires Computational Biology

The DNA Data Race

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)

Sequencing a single human genome uses ~100 GB of compressed sequence data in billions of short reads.

~20 DVDs / genome

The DNA Data Tsunami

Use massive amounts of sequencing to explore the genetic origins of life

Our best (only) hope is to use many computers:

- Parallel Computing aka Cloud Computing

- Now your programs will crash on 1000 computers instead of just 1 ☺

Amazon Web Services

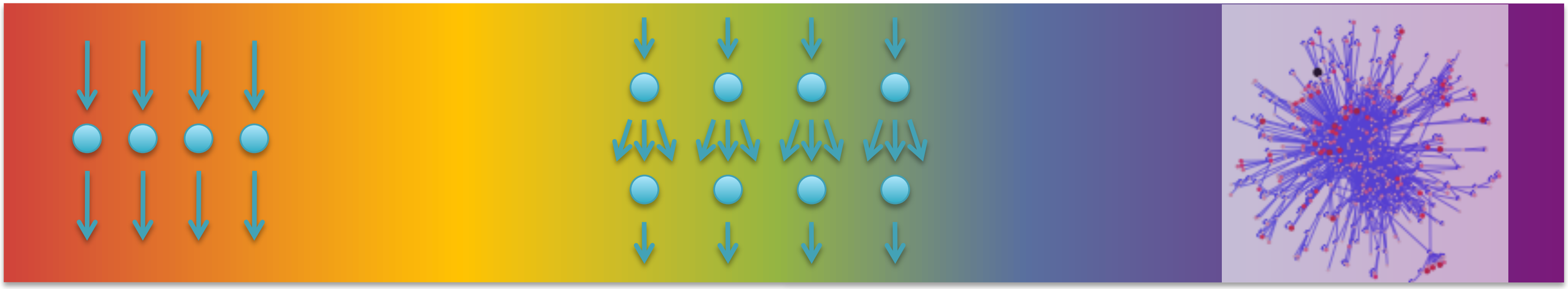
http://aws.amazon.com

- All you need is a credit card, and you can immediately start using one of the largest datacenters in the world

- Elastic Compute Cloud (EC2)
  - On demand computing power
    - Support for Windows, Linux, & OpenSolaris
    - Starting at 8.5¢ / core / hour
- Simple Storage Service (S3)
  - Scalable data storage
    - 10¢ / GB upload fee, 15¢ / GB monthly fee

Cloud Computing Spectrum

Embarrassingly Parallel	Loosely Coupled	Tightly Coupled



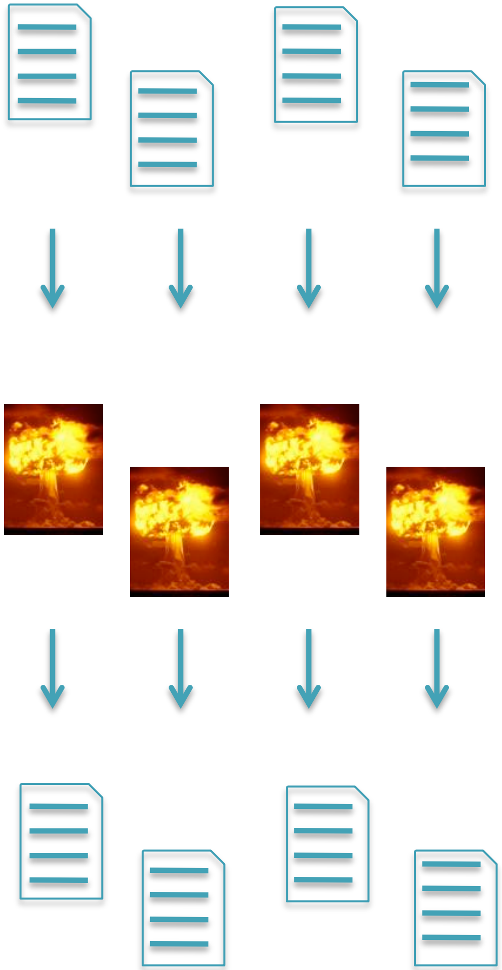
Batch Computing	MapReduce/DryadLINQ	MPI/MapReduce/Pregel
Alignment HMM Scoring	Genotyping K-mer Counting	Graph Analysis Genome Assembly
Scheduling + Load Balance	Embarrassingly Parallel + Parallel Communication	Loosely Coupled + Parallel Algorithm Design

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

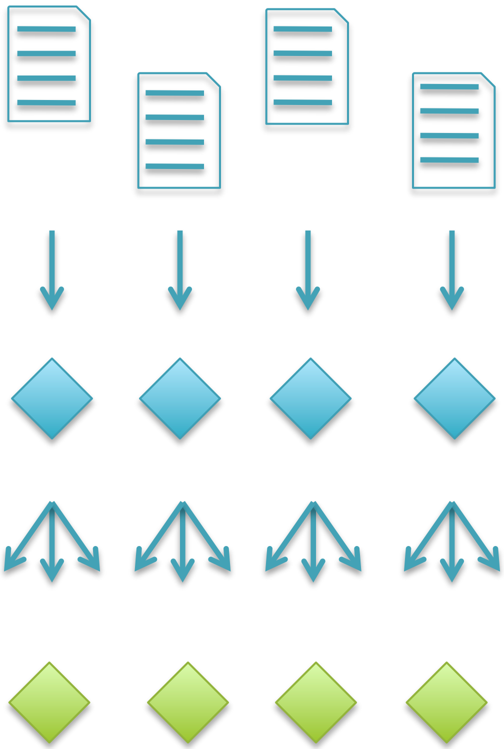
- Technologies
  - Condor, Sun Grid Engine
  - Amazon Simple Queue



Elementary School Dance

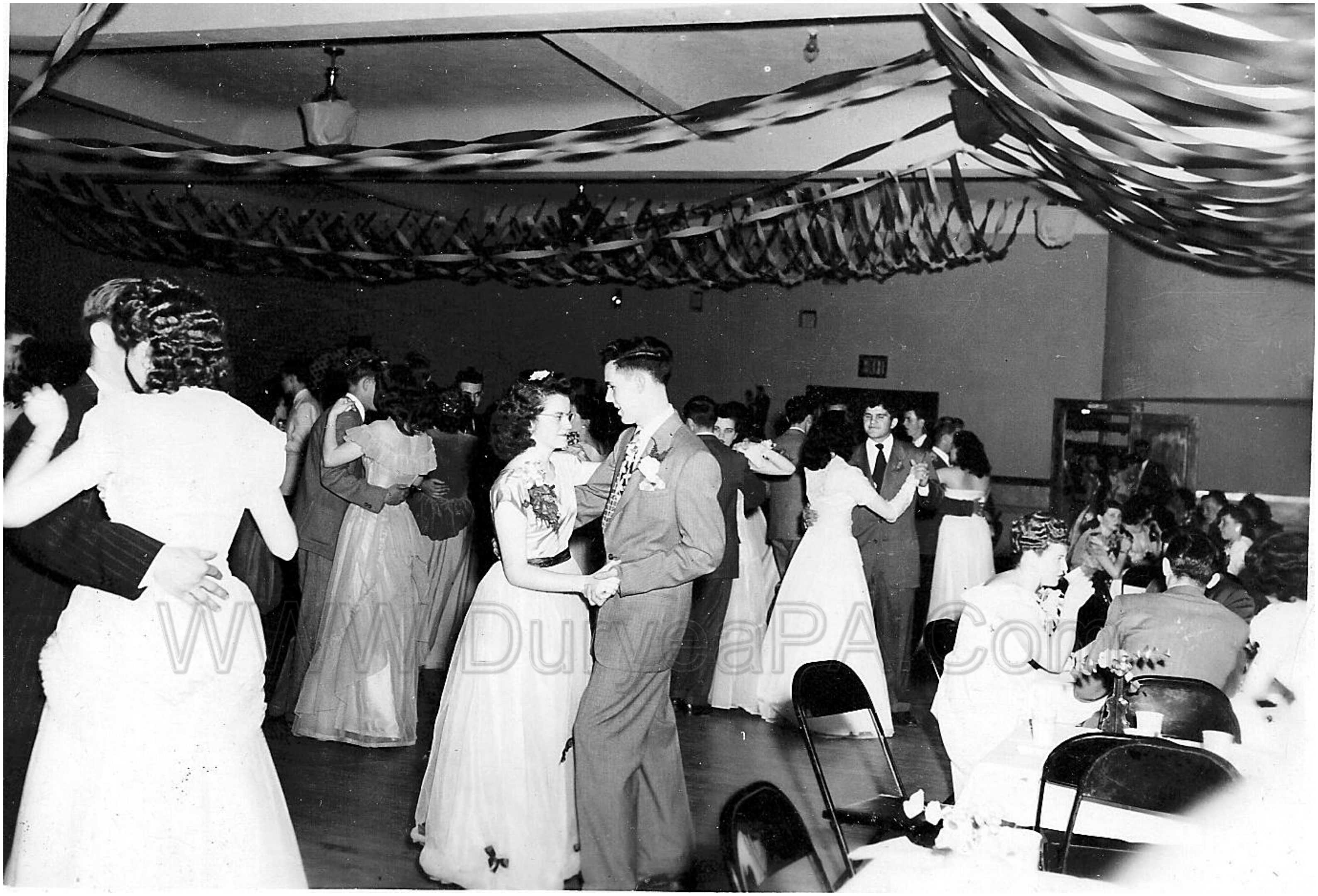
Loosely Coupled

- Divide and conquer
  - Independently process many items
  - Group partial results
  - Scan partial results into final answer



- Challenges
  - Batch computing challenges
  - + Shuffling of huge datasets
- Technologies
  - Hadoop, Elastic MapReduce, Dryad
  - Parallel Databases

Junior High Dance



Hadoop MapReduce

http://hadoop.apache.org

- 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

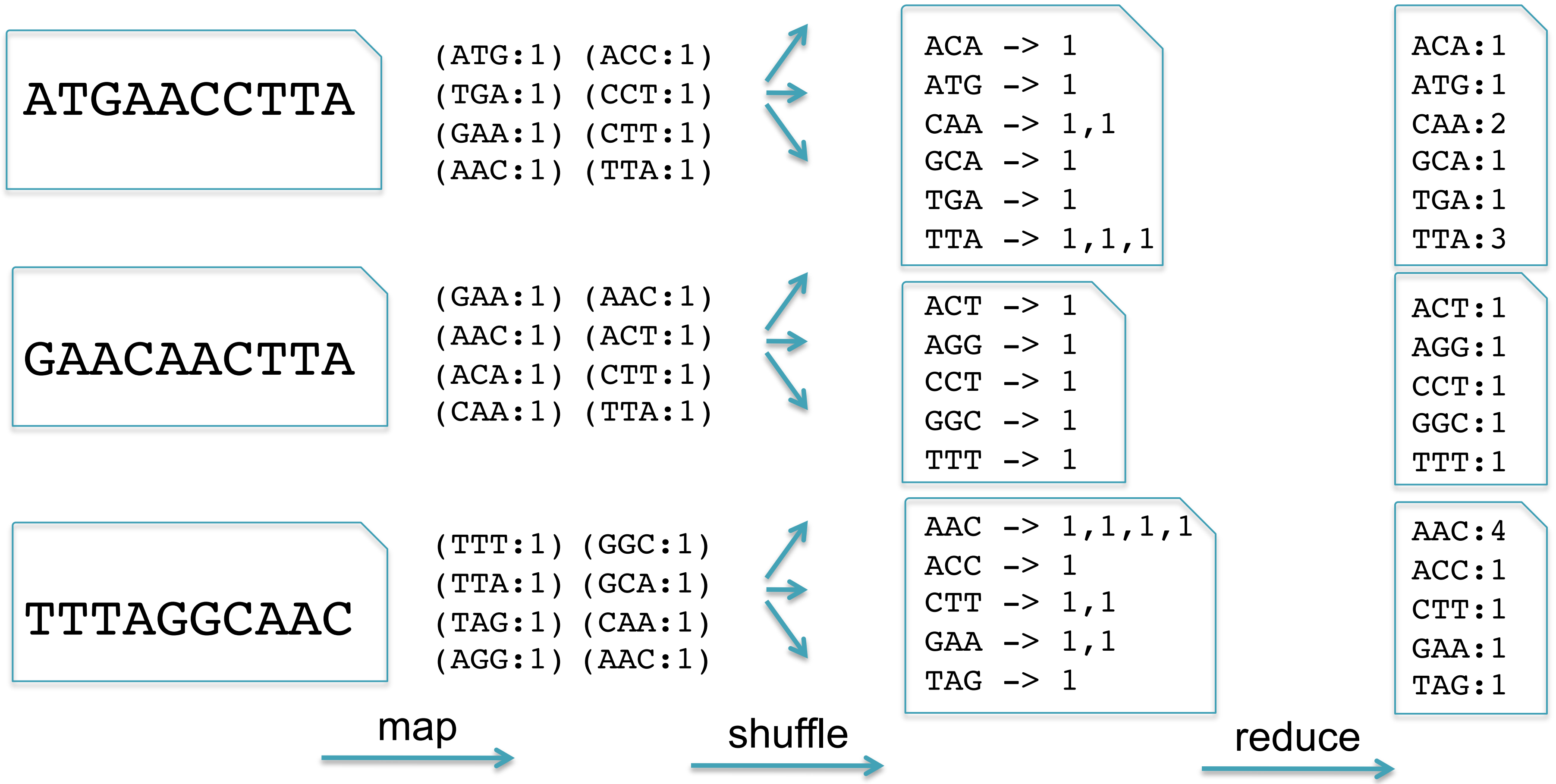
- Benefits
  - Scalable, Efficient, Reliable
  - Easy to Program
  - Runs on commodity computers

- 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



ATGAACCTTA

GAACAACTTA

TTTAGGCAAC

```
(ATG:1) (ACC:1)
(TGA:1) (CCT:1)
(GAA:1) (CTT:1)
(AAC:1) (TTA:1)
```

```
(GAA:1) (AAC:1)
(AAC:1) (ACT:1)
(ACA:1) (CTT:1)
(CAA:1) (TTA:1)
```

```
(TTT:1) (GGC:1)
(TTA:1) (GCA:1)
(TAG:1) (CAA:1)
(AGG:1) (AAC:1)
```

```
ACA -> 1
ATG -> 1
CAA -> 1,1
GCA -> 1
TGA -> 1
TTA -> 1,1,1
```

```
ACT -> 1
AGG -> 1
CCT -> 1
GGC -> 1
TTT -> 1
```

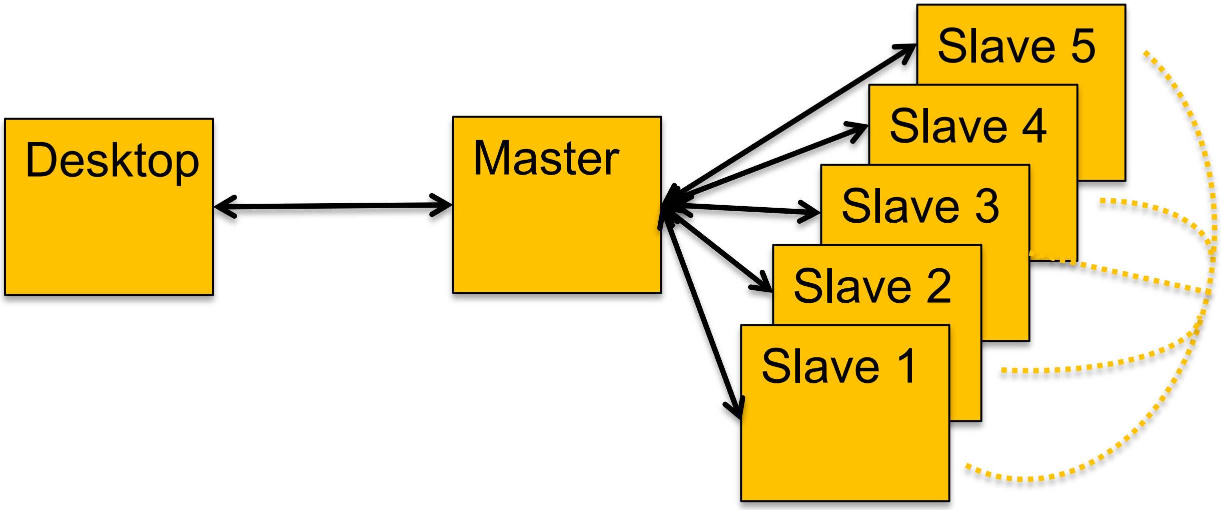
```
AAC -> 1,1,1,1
ACC -> 1
CTT -> 1,1
GAA -> 1,1
TAG -> 1
```

```
ACA:1
ATG:1
CAA:2
GCA:1
TGA:1
TTA:3
```

```
ACT:1
AGG:1
CCT:1
GGC:1
TTT:1
```

```
AAC:4
ACC:1
CTT:1
GAA:1
TAG:1
```

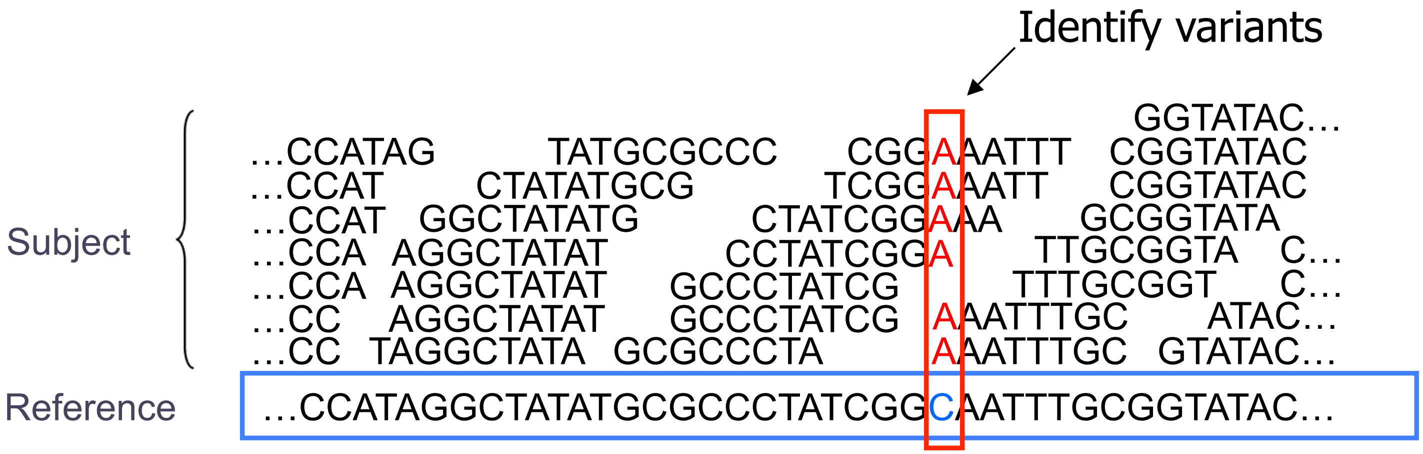
Hadoop Architecture



- Hadoop Distributed File System (HDFS)
  - Data files partitioned into large chunks (64MB), replicated on multiple nodes
  - 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

Short Read Mapping



Identify variants

```
                                                   GGTATAC...
          ...CCATAG     TATGCGCCC    CGGAAATTT  CGGTATAC
          ...CCAT     CTATATGCG     TCGGAAATT   CGGTATAC
Subject   ...CCAT  GGCTATATG     CTATCGGAAA    GCGGTATA
          ...CCA  AGGCTATAT     CCTATCGGA     TTGCGGTA   C...
          ...CCA  AGGCTATAT   GCCCTATCG      TTTGCGGT    C...
          ...CC   AGGCTATAT   GCCCTATCG   AAATTTGC     ATAC...
          ...CC  TAGGCTATA  GCGCCCTA      AAATTTGC  GTATAC...

Reference ...CCATAGGCTATATGCGCCCTATCGGCAATTTGCGGTATAC...
```

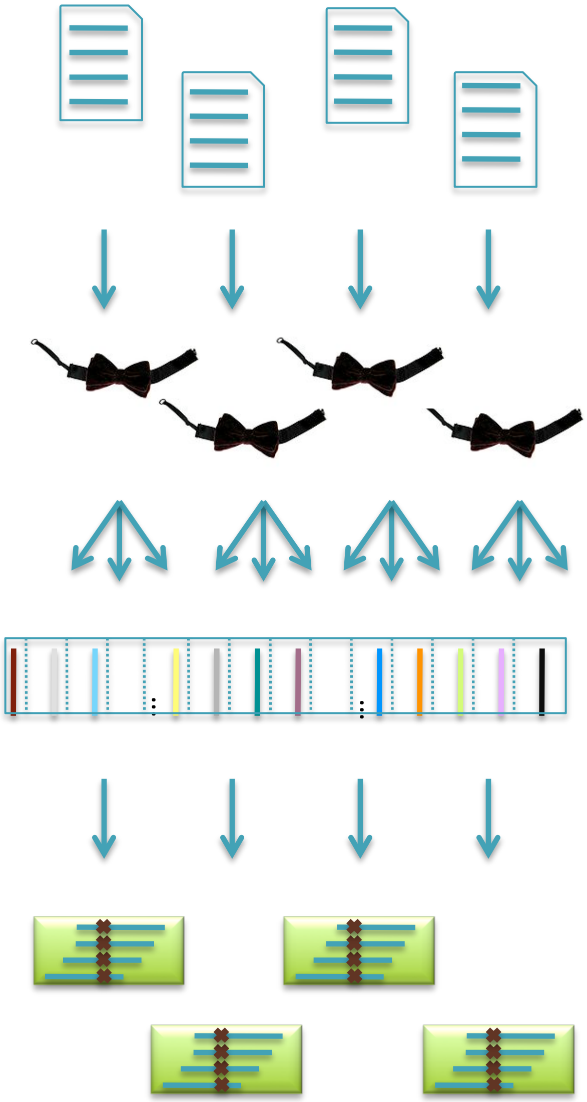
- Given a reference and many subject reads, report one or more "good" end-to-end alignments per alignable read
  - Find where the read most likely originated
  - Fundamental computation for many assays
    - Genotyping RNA-Seq Methyl-Seq
    - Structural Variations Chip-Seq Hi-C-Seq

- 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

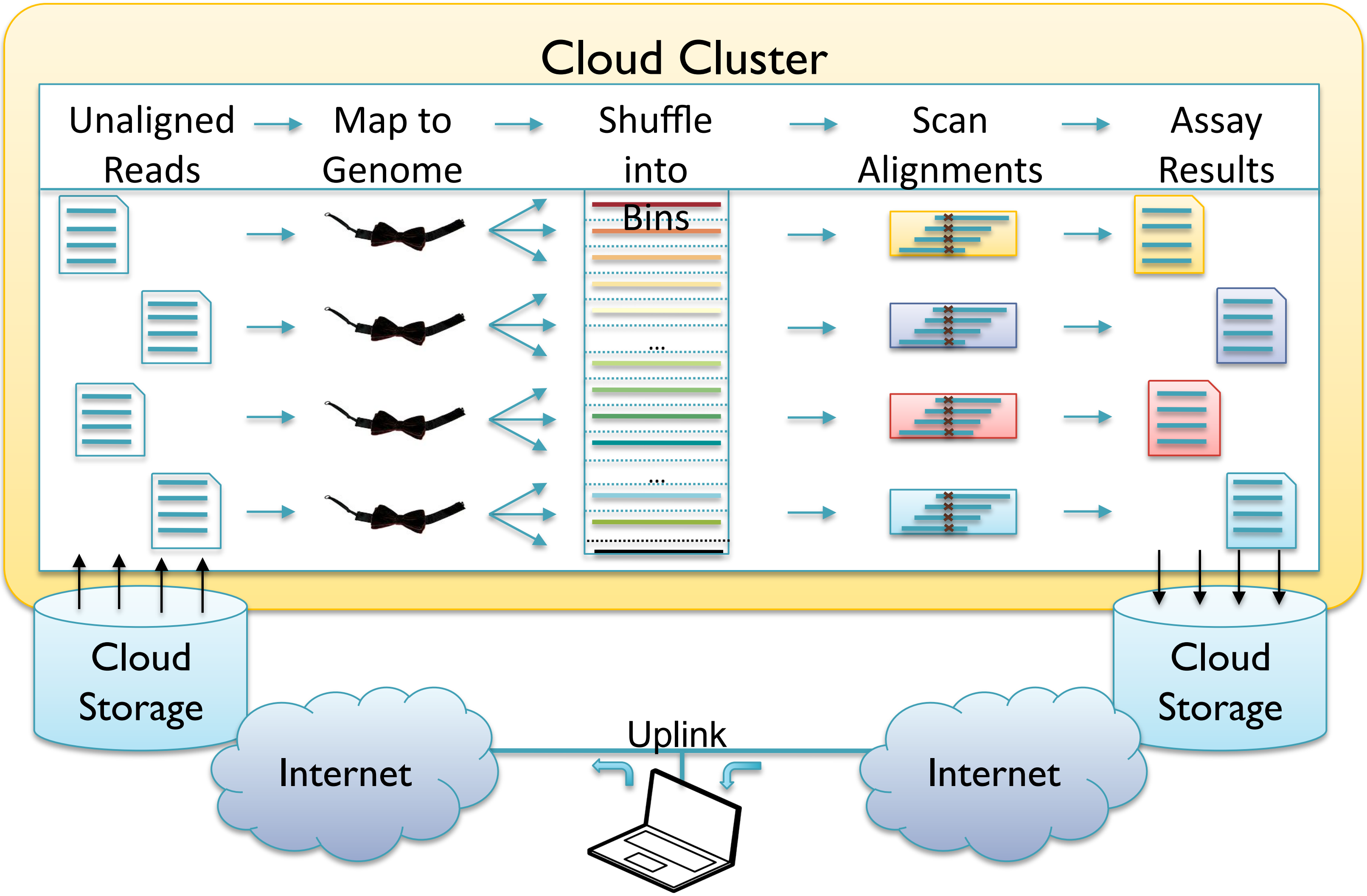
http://bowtie-bio.sourceforge.net/crossbow

	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

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*. **10**:R134

Map-Shuffle-Scan for Genomics



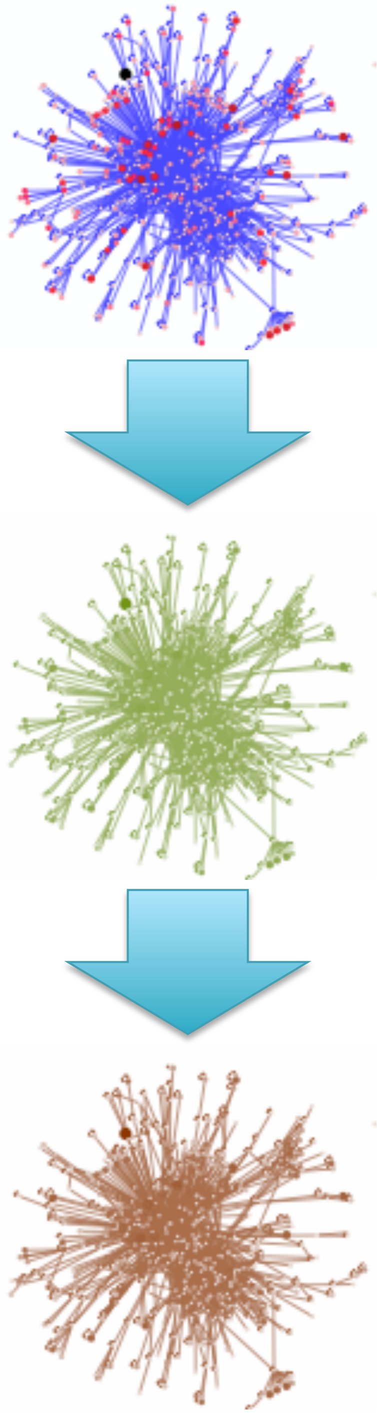
Cloud Computing and the DNA Data Race.
Schatz, MC, Langmead B, Salzberg SL (2010) *Nature Biotechnology* 28:691-693

Tightly Coupled

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

- Challenges
  - Loosely coupled challenges
  - + Parallel algorithms design

- Technologies
  - MPI
  - MapReduce, Dryad, Pregel

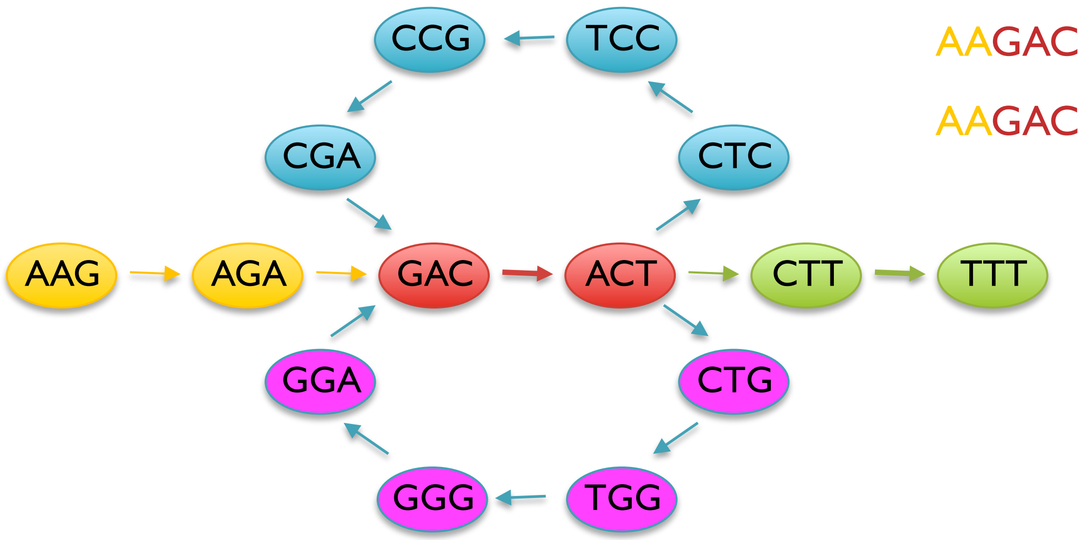


Short Read Assembly

Reads

AAGA
ACTT
ACTC
ACTG
AGAG
CCGA
CGAC
CTCC
CTGG
CTTT
…

de Bruijn Graph



Potential Genomes

AAGACTCCGACTGGGACTTT

AAGACTGGGACTCCGACTTT

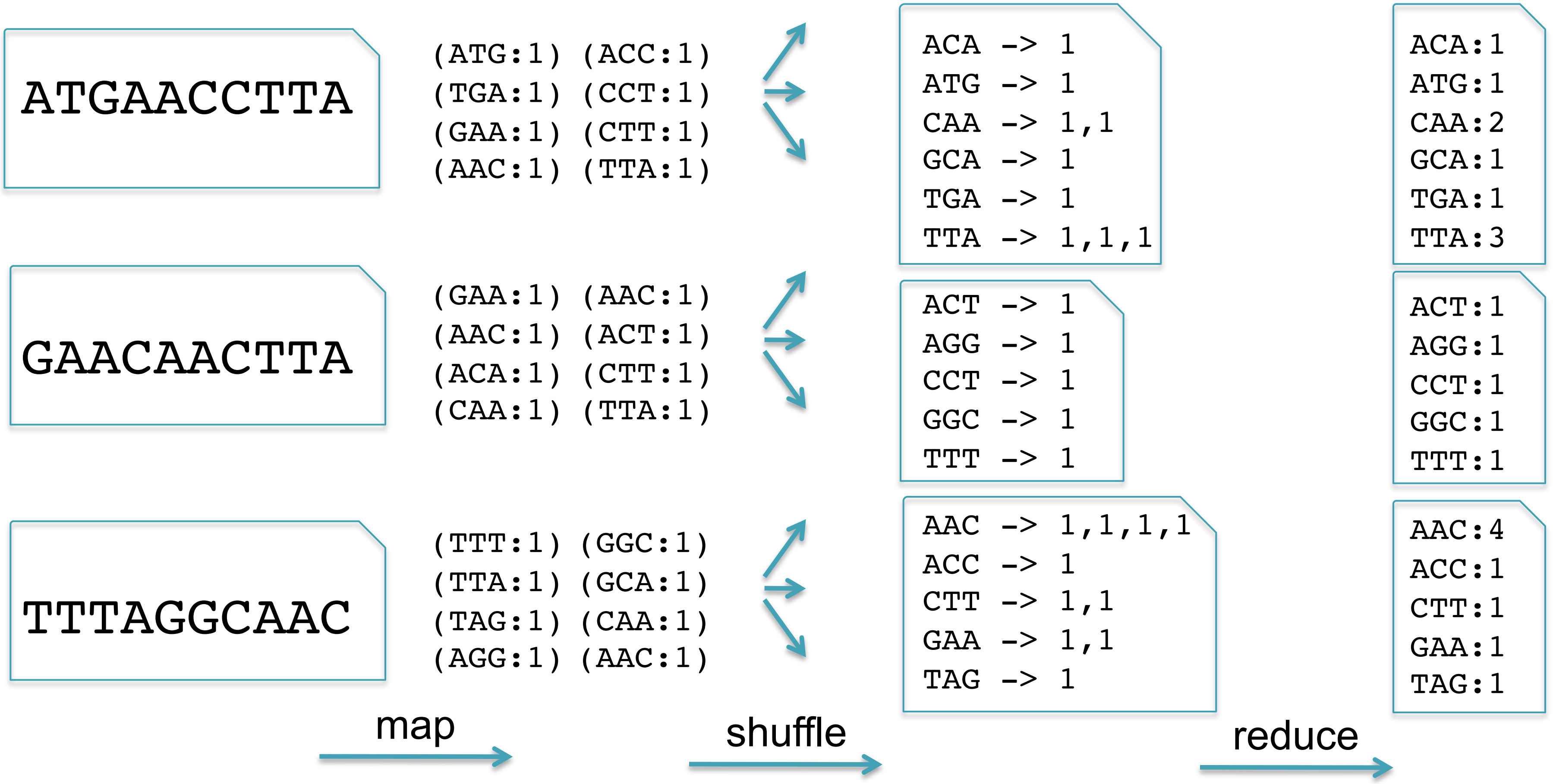
- Genome assembly as finding an Eulerian tour of the de Bruijn graph
  - Human genome: >3B nodes, >10B edges

- 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

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



```
ATGAACCTTA

GAACAACTTA

TTTAGGCAAC
```

```
(ATG:1) (ACC:1)
(TGA:1) (CCT:1)
(GAA:1) (CTT:1)
(AAC:1) (TTA:1)

(GAA:1) (AAC:1)
(AAC:1) (ACT:1)
(ACA:1) (CTT:1)
(CAA:1) (TTA:1)

(TTT:1) (GGC:1)
(TTA:1) (GCA:1)
(TAG:1) (CAA:1)
(AGG:1) (AAC:1)
```

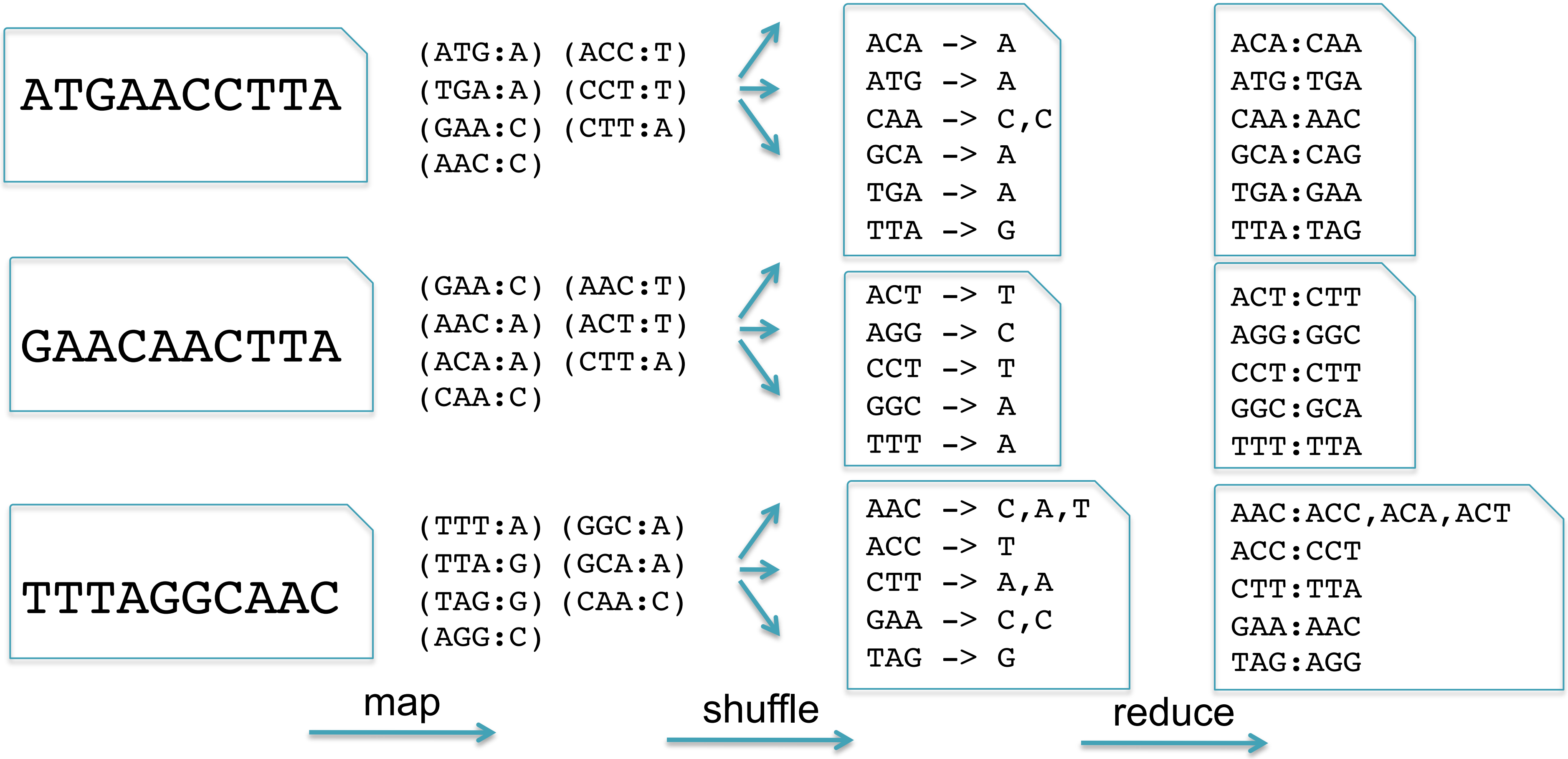
```
ACA -> 1
ATG -> 1
CAA -> 1,1
GCA -> 1
TGA -> 1
TTA -> 1,1,1

ACT -> 1
AGG -> 1
CCT -> 1
GGC -> 1
TTT -> 1

AAC -> 1,1,1,1
ACC -> 1
CTT -> 1,1
GAA -> 1,1
TAG -> 1
```

```
ACA:1
ATG:1
CAA:2
GCA:1
TGA:1
TTA:3

ACT:1
AGG:1
CCT:1
GGC:1
TTT:1

AAC:4
ACC:1
CTT:1
GAA:1
TAG:1
```

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



ATGAACCTTA

GAACAACTTA

TTTAGGCAAC

```
(ATG:A) (ACC:T)
(TGA:A) (CCT:T)
(GAA:C) (CTT:A)
(AAC:C)
```

```
(GAA:C) (AAC:T)
(AAC:A) (ACT:T)
(ACA:A) (CTT:A)
(CAA:C)
```

```
(TTT:A) (GGC:A)
(TTA:G) (GCA:A)
(TAG:G) (CAA:C)
(AGG:C)
```

```
ACA -> A
ATG -> A
CAA -> C,C
GCA -> A
TGA -> A
TTA -> G
```

```
ACT -> T
AGG -> C
CCT -> T
GGC -> A
TTT -> A
```

```
AAC -> C,A,T
ACC -> T
CTT -> A,A
GAA -> C,C
TAG -> G
```

```
ACA:CAA
ATG:TGA
CAA:AAC
GCA:CAG
TGA:GAA
TTA:TAG
```

```
ACT:CTT
AGG:GGC
CCT:CTT
GGC:GCA
TTT:TTA
```

```
AAC:ACC,ACA,ACT
ACC:CCT
CTT:TTA
GAA:AAC
TAG:AGG
```

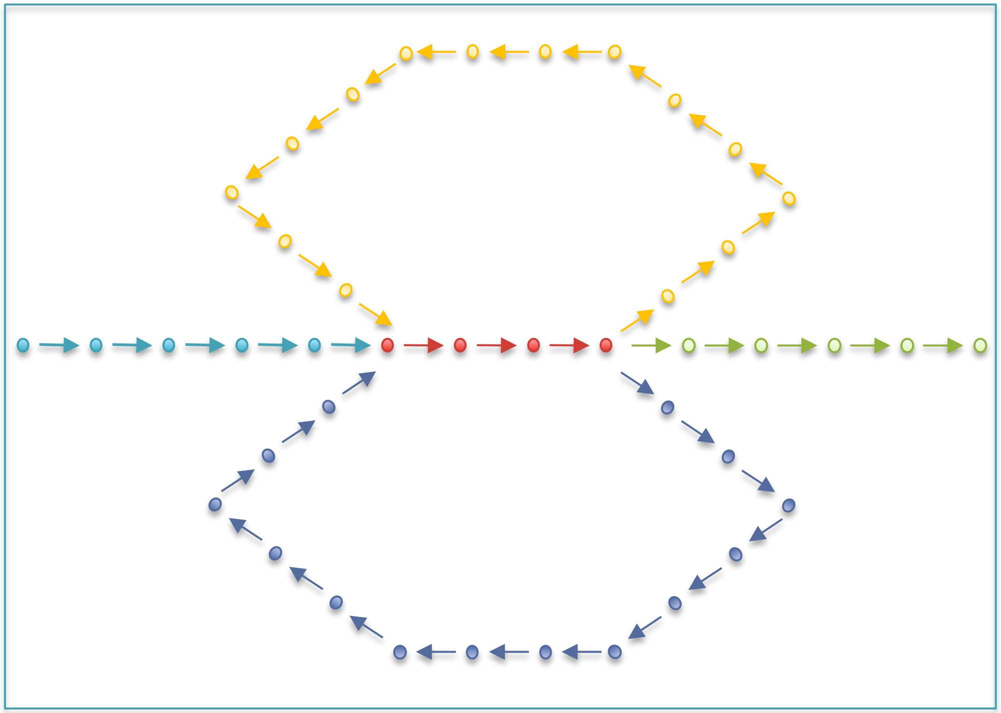
map

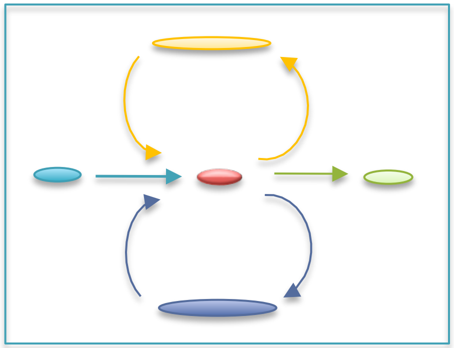
shuffle

reduce

Graph Compression

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

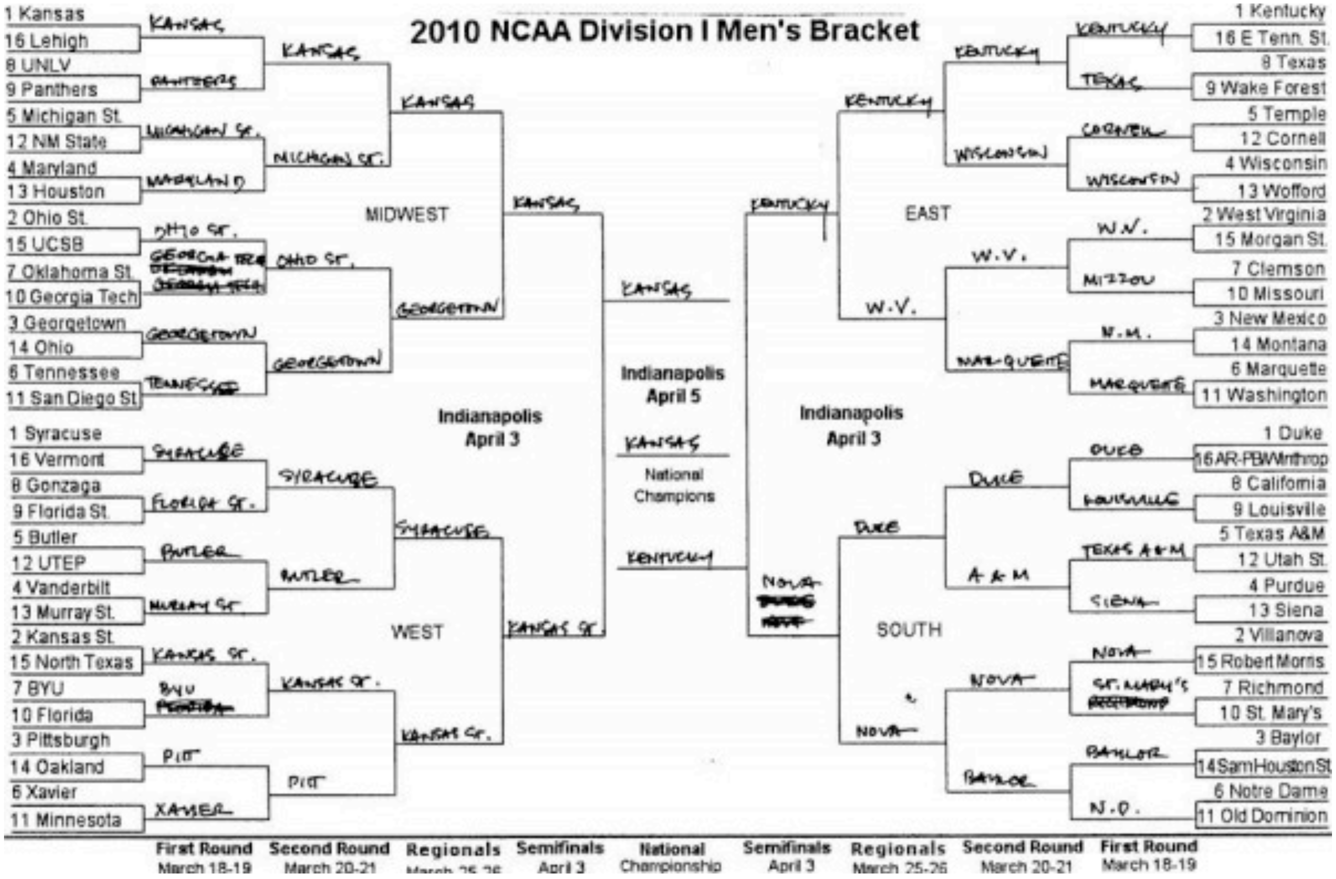




High School Dance

Warmup Exercise

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



Find winner among 64 teams in just 6 rounds

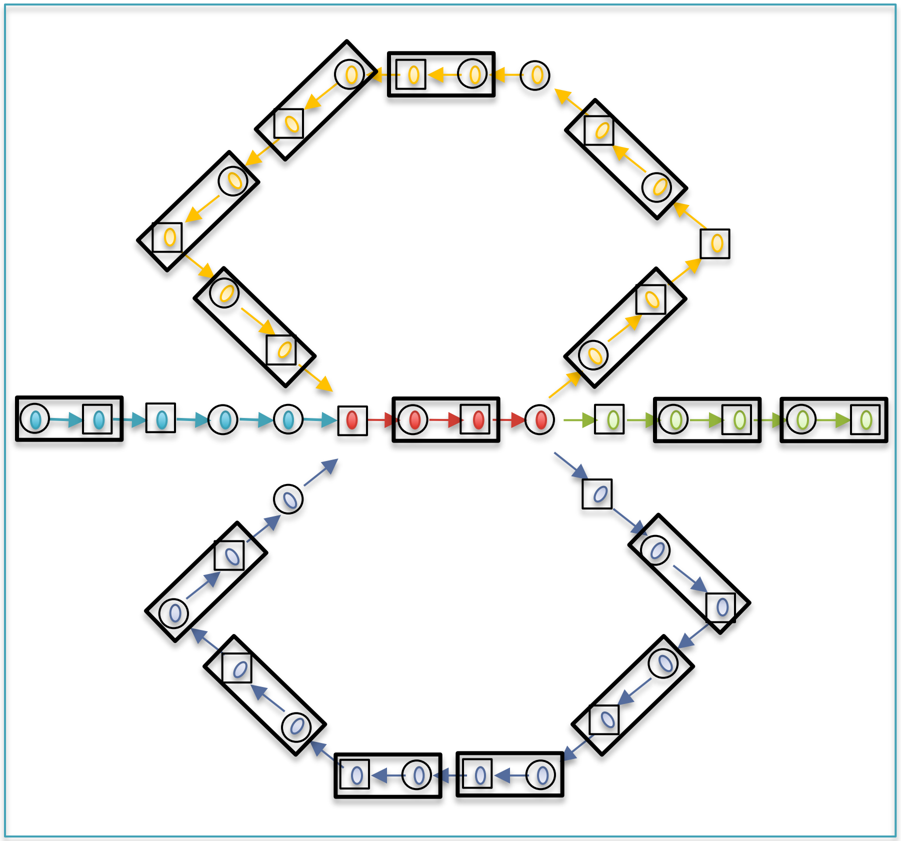
Fast Path Compression

Challenges

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

Randomized List Ranking

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



Initial Graph: 42 nodes

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

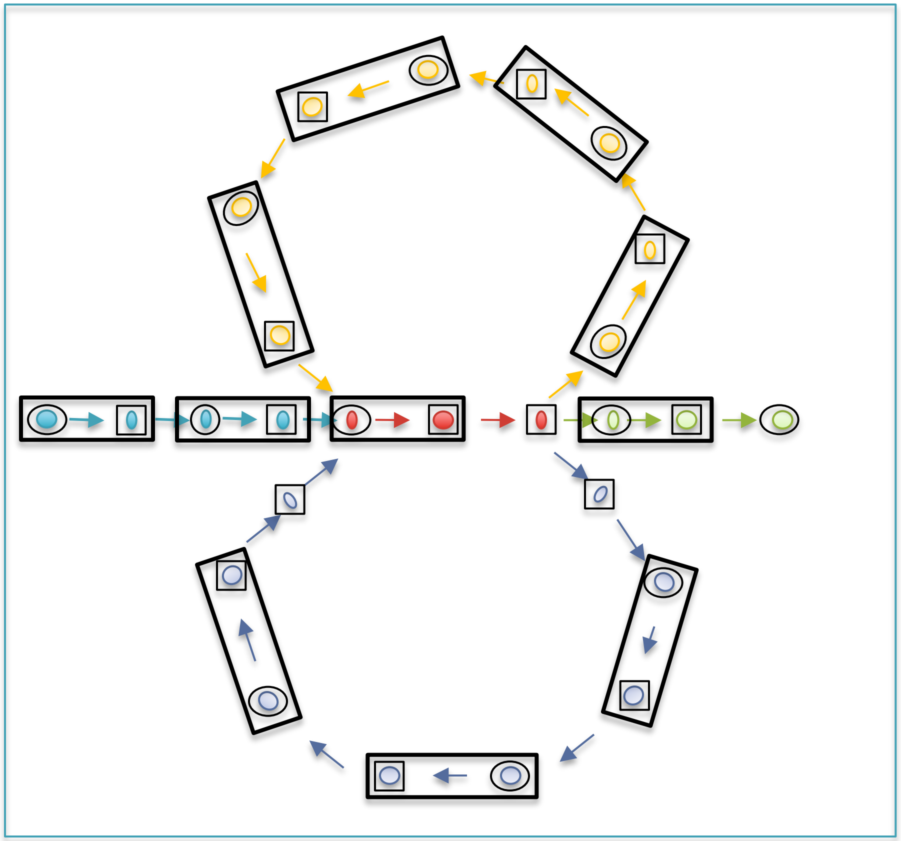
Fast Path Compression

Challenges

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

Randomized List Ranking

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



Round 1: 26 nodes (38% savings)

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

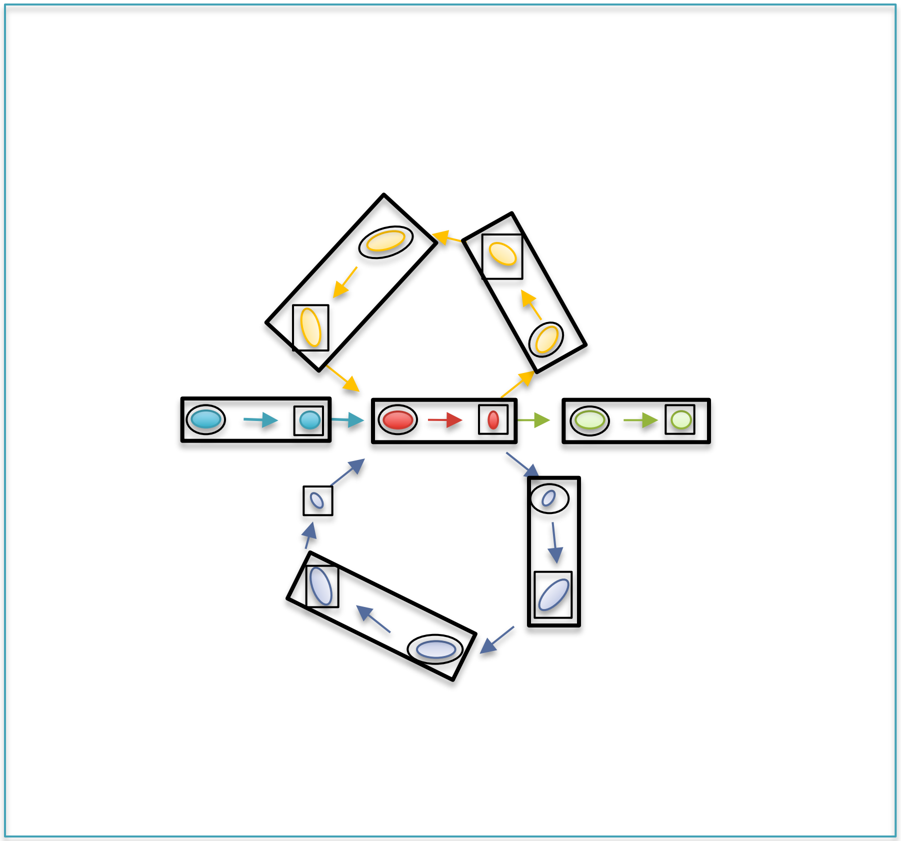
Fast Path Compression

Challenges

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

Randomized List Ranking

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



Round 2: 15 nodes (64% savings)

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

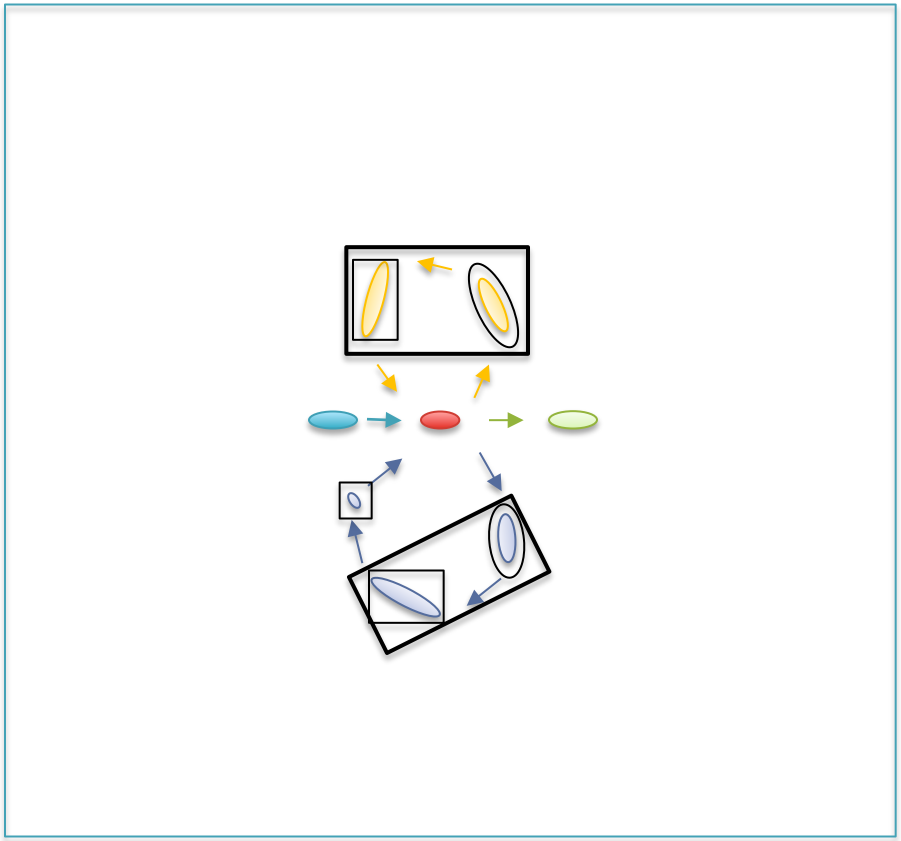
Fast Path Compression

Challenges

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

Randomized List Ranking

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



Round 2: 8 nodes (81% savings)

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

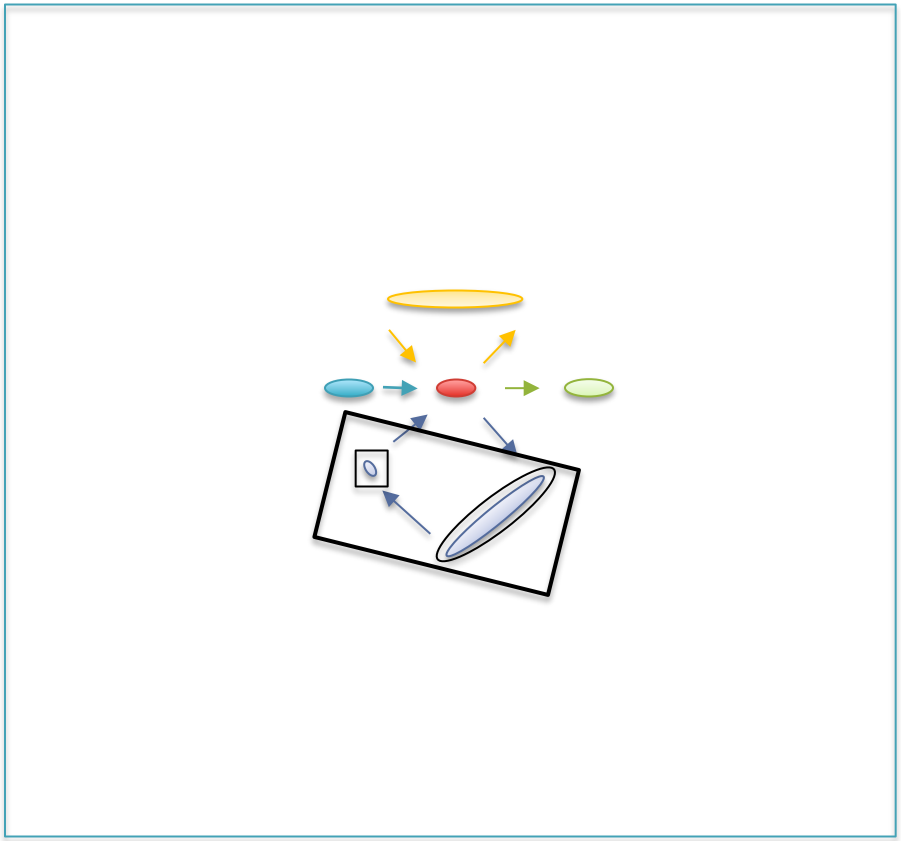
Fast Path Compression

Challenges

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

Randomized List Ranking

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

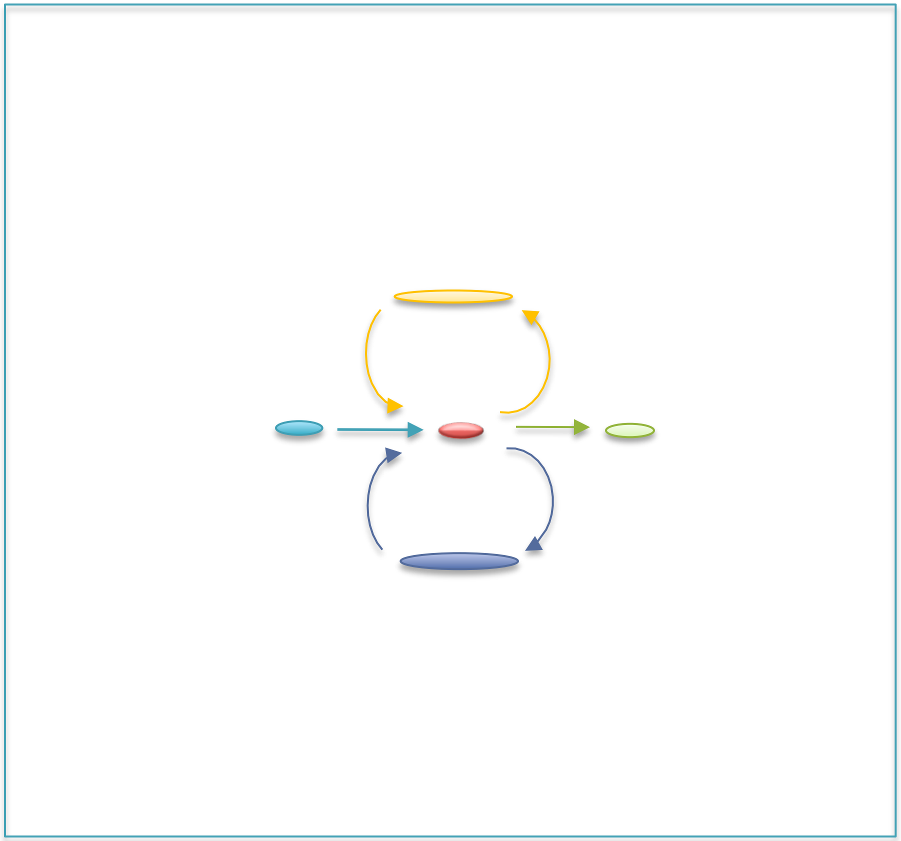
- Nodes stored on different computers
- Nodes can only access direct neighbors

Randomized List Ranking

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

Performance

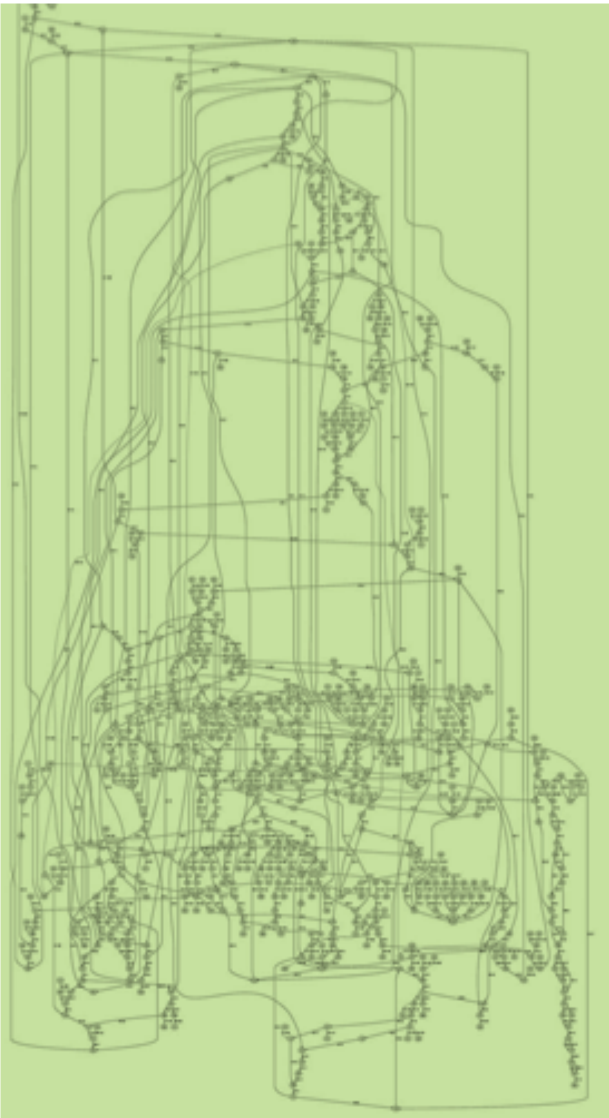
- Compress all chains in log(S) rounds



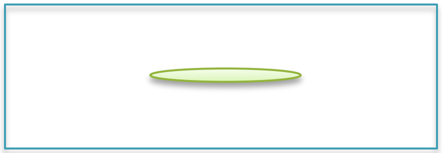
Round 4: 5 nodes (88% savings)

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

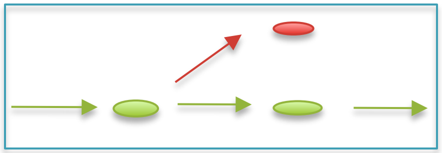
Node Types



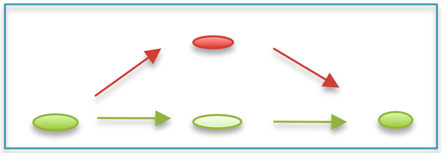
(Chaisson, 2009)



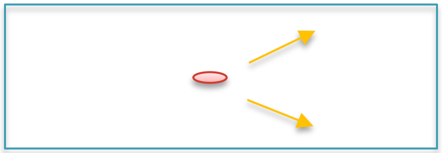
Isolated nodes (10%)



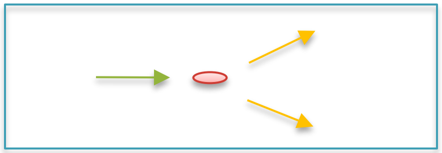
Tips (46%)



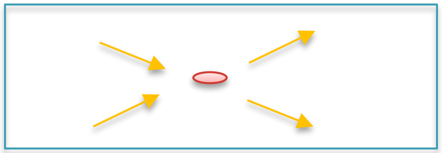
Bubbles/Non-branch (9%)



Dead Ends (.2%)



Half Branch (25%)



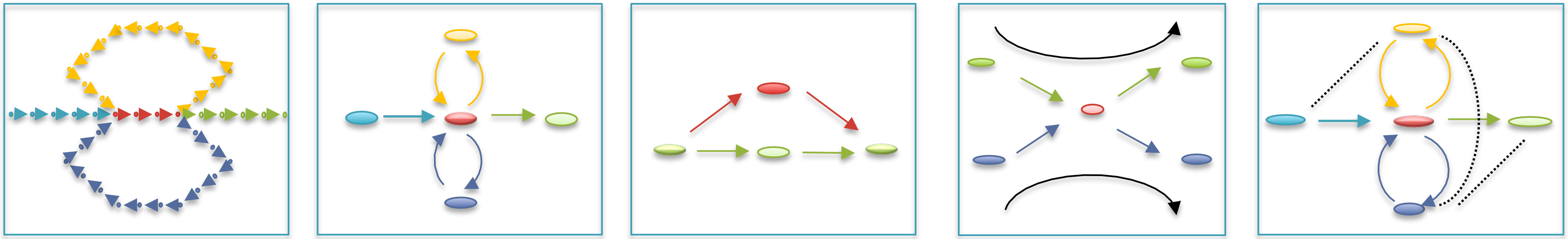
Full Branch (10%)

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



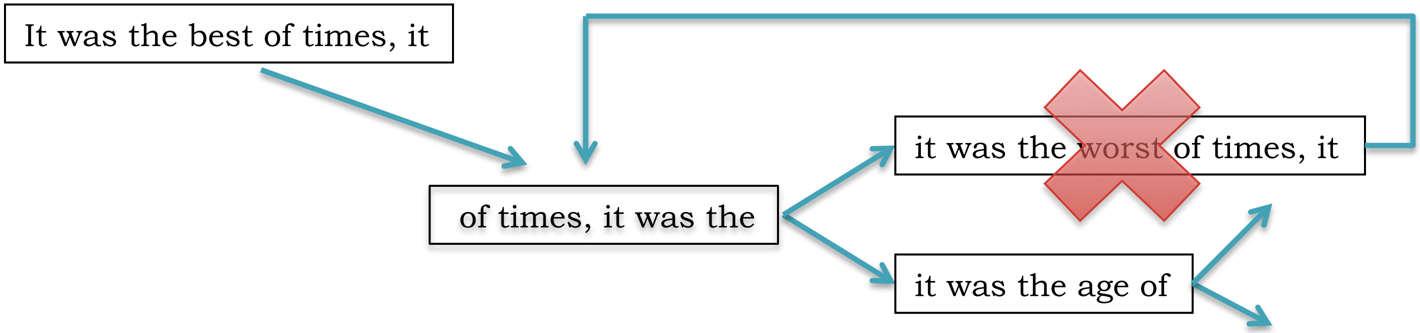
	Initial	Compressed	Error Correction	Resolve Repeats	Cloud Surfing
N	5.1 M	245,131	2,769	1,909	300
Max	27 bp	1,079 bp	70,725 bp	90,088 bp	149,006 bp
N50	27 bp	156 bp	15,023 bp	20,062 bp	54,807 bp

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



It was the best of times, it

of times, it was the

it was the worst of times, it

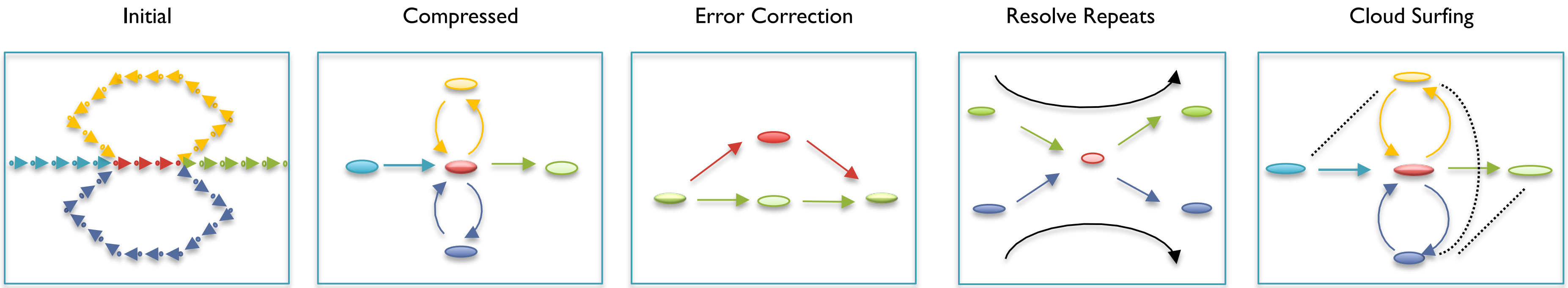
it was the age of

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)

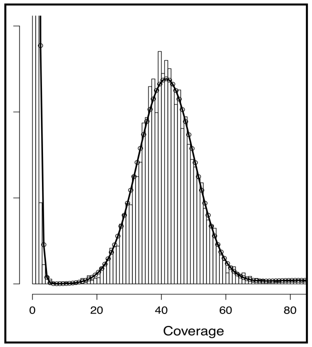


N	>7 B	>1 B	4.2 M	4.1 M	3.3 M
Max	27 bp	303 bp	20,594 bp	20,594 bp	20,594 bp
N50	27 bp	< 100 bp	995 bp	1,050 bp	1,427 bp*

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

Hadoop for NGS Analysis

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*)

CloudBurst

Highly Sensitive Short Read Mapping with MapReduce

100x speedup mapping on 96 cores @ Amazon

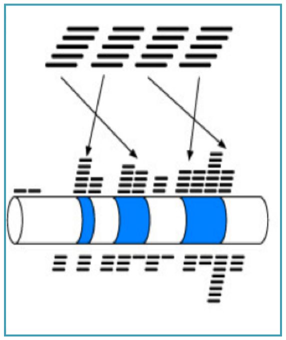
http://cloudburst-bio.sf.net

(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/

AMOS

Searching for SNPs in the Turkey Genome

Scan the de novo assembly to find 920k hetrozygous alleles

(Dalloul et al, 2010)

http://amos.sf.net

Summary

- Surviving the data deluge means computing in parallel
  - Cloud computing is an attractive platform for large scale sequence analysis and computation
- Use the right tool for the job
  - Embarassingly parallel = Condor/Hadoop
  - Loosely coupled = Hadoop/Dyrad
  - Tightly coupled = MPI/Hadoop
- Emerging technologies are a great start, but we need continued research
  - A word of caution: new technologies are new

Acknowledgements

Advisor

Steven Salzberg

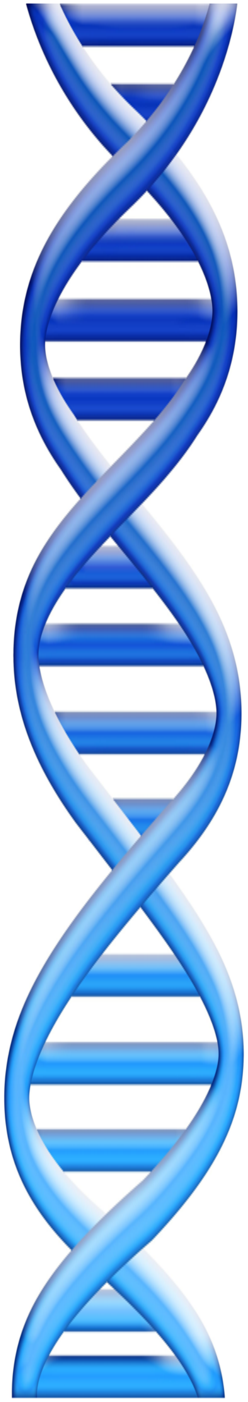
UMD Faculty

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

Break



Outline

Part 1: Theory

1. Genome Assembly by Analogy
2. DNA Sequencing and Genomics
3. Sequence Analysis in the Clouds
    1. Sequence Alignment
    2. Mapping & Genotyping
    3. Genome Assembly

Part 2: Practice

1. AWS Mini-Tutorial
2. Hadoop Mini-Tutorial

A Brief History of the Amazon Cloud

- Urban Legend
  - Additional capacity added every fall for the holiday shopping season, underutilized rest of the year…

- Official Story
  - Amazon is a technology company
    - Different divisions of Amazon share computation
  - Amazon Web Services is the 3rd Business Division
    - Retail & Seller Businesses

Amazon Web Services

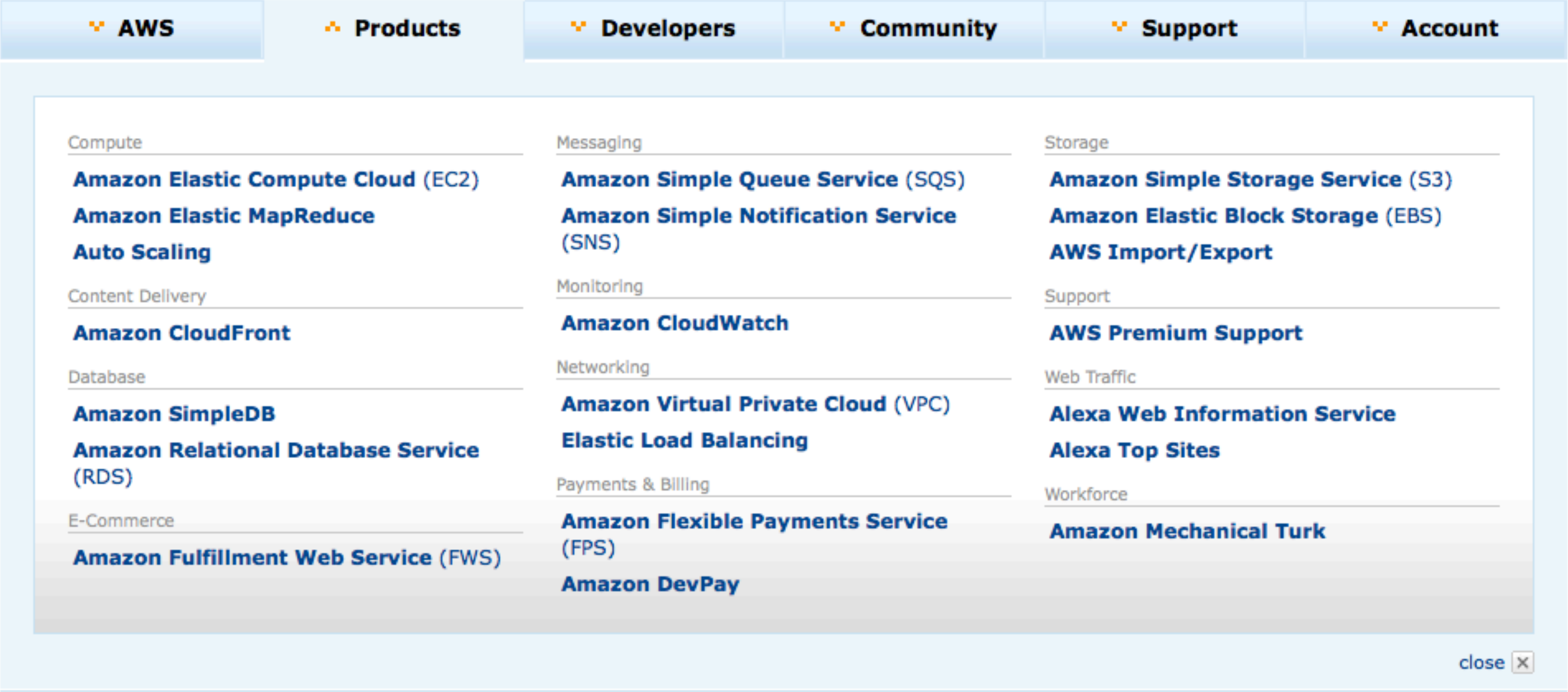
http://aws.amazon.com

- All you need is a credit card, and you can immediately start using one of the largest datacenters in the world

- Elastic Compute Cloud (EC2)
  - On demand computing power
    - Support for Windows, Linux, & OpenSolaris
    - Starting at 8.5¢ / core / hour

- Simple Storage Service (S3)
  - Scalable data storage
    - 10¢ / GB upload fee, 15¢ / GB monthly fee

Other Services



Compute

- **Amazon Elastic Compute Cloud** (EC2)
- **Amazon Elastic MapReduce**
- **Auto Scaling**

Content Delivery

- **Amazon CloudFront**

Database

- **Amazon SimpleDB**
- **Amazon Relational Database Service** (RDS)

E-Commerce

- **Amazon Fulfillment Web Service** (FWS)

Messaging

- **Amazon Simple Queue Service** (SQS)
- **Amazon Simple Notification Service** (SNS)

Monitoring

- **Amazon CloudWatch**

Networking

- **Amazon Virtual Private Cloud** (VPC)
- **Elastic Load Balancing**

Payments & Billing

- **Amazon Flexible Payments Service** (FPS)
- **Amazon DevPay**

Storage

- **Amazon Simple Storage Service** (S3)
- **Amazon Elastic Block Storage** (EBS)
- **AWS Import/Export**

Support

- **AWS Premium Support**

Web Traffic

- **Alexa Web Information Service**
- **Alexa Top Sites**

Workforce

- **Amazon Mechanical Turk**

Everything you need to run a large scale service & analysis suite in the clouds.

Cloud Solutions for DNA Sequence Analysis

- Rapid & elastic deployment of vast computation
  - Accessible, Reproducible, Reliable, Collaborative

- Why not?
  - Transfer: 200 GBs takes 1 hr – 2 weeks
  - Privacy & security: Excellent... with care
  - Expertise: Computing on 1000 cores is complex
  - Price: The meter is always running

  - Who will be on the line for making it all work?
    - Psychological and Institutional paradigm shift

EC2 Architecture

- Very large pool of machines
  - Effectively infinite resources
  - High-end servers with many cores and many GB RAM
- Machines run in a virtualized environment
  - Amazon can subdivide large nodes into smaller instances
  - You are 100% protected from other users on the machine
  - You get to pick the operating system, all installed software

Instance Types

Type	Price / hr	CPU	Resources
Micro *Web service*	2¢	1 core @ 1 ECU	.6 GB RAM 10 GB Disk
Standard *Light Tasks*	8.5¢	1 core @ 1 ECU	1.7 GB RAM 160 GB Disk
Extra Large *Mapping w/BWA*	68¢	4 core @ 2 ECU	15 GB RAM 1.7 TB Disk
HighCPU XL *Mapping w/Crossbow*	68¢	8 cores @ 2.5 ECU	7 GB RAM 1.7 TB Disk
HighMem Quad XL *Assembly*	$2.00	8 cores @ 3.25 ECU	68.4 GB RAM 1.7 TB Disk

ECU = EC2 Compute Unit. Approximately 1.0 – 1.2 GHz Intel Xeon from 2007

Reserved Instances make it cheaper for consistent use.
Pay for what you use, rounded UP to the next full hour

Amazon Machine Images

- A few Amazon sponsored images
  - – Suse Linux, Windows
- Many Community Images & Appliances
  - – Crossbow: Hadoop, Bowtie, SOAPsnp
  - – CloudBioLinux.com: Appliance for Genomics
- Build you own
  - – Completely customize your environment
  - – You results could be totally reproducible

Amazon S3

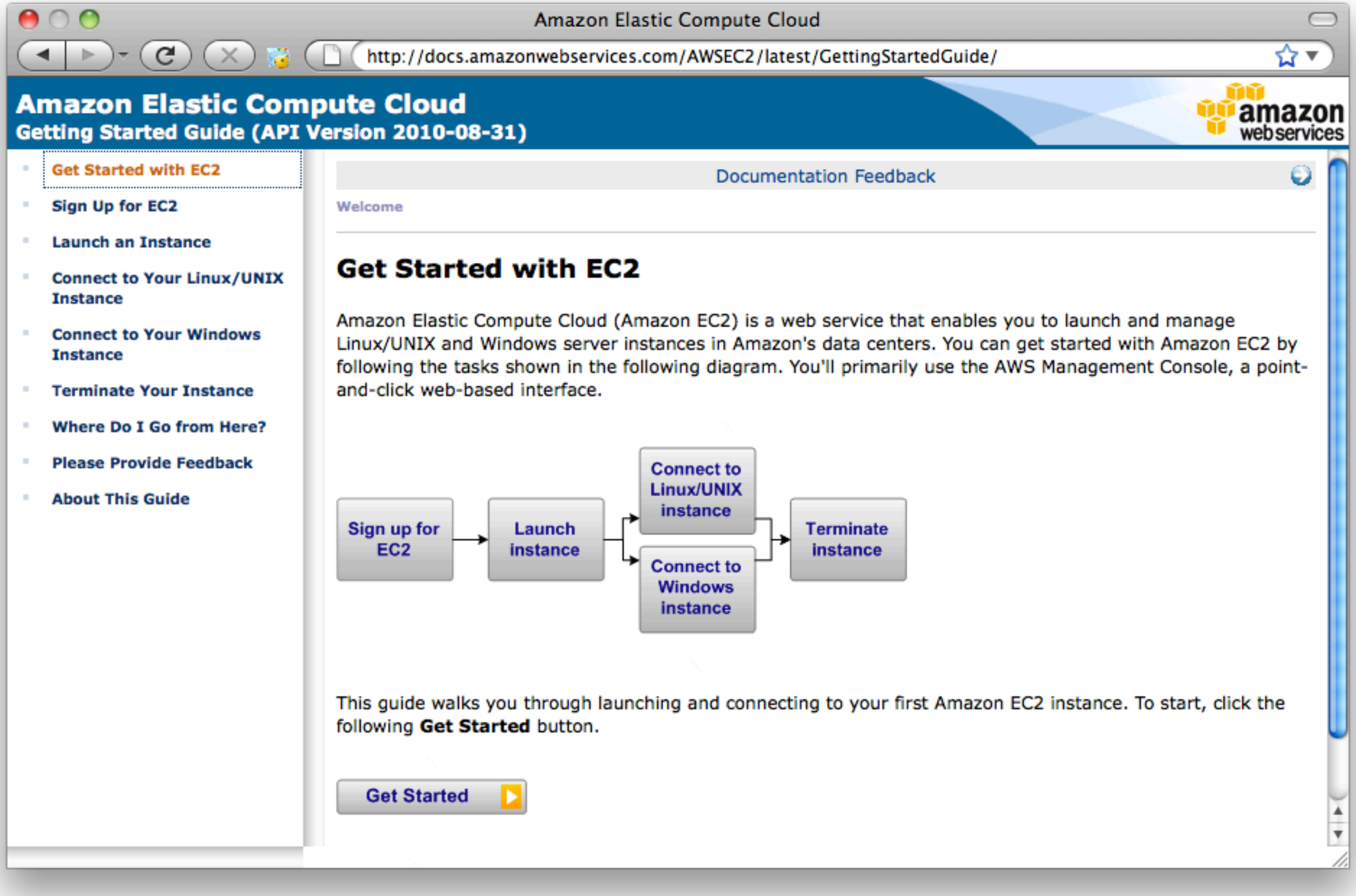
- S3 provides persistent storage for large volumes of data
  - Very high speed connection from S3 to EC2 compute nodes
  - Public data sets include s3://1000genomes

– Tiered pricing by volume
  - Pricing starts at 15¢ / GB / month
  - 5.5¢ / GB / month for over 5 PB
  - Pay for transfer in and out of Amazon

– Import/Export service for large volumes
  - FedEx your drives to Amazon

Getting Started

http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/



Amazon Elastic Compute Cloud
http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/
amazon web services

Amazon Elastic Compute Cloud

Getting Started Guide (API Version 2010-08-31)

- Get Started with EC2
- Sign Up for EC2
- Launch an Instance
- Connect to Your Linux/UNIX Instance
- Connect to Your Windows Instance
- Terminate Your Instance
- Where Do I Go from Here?
- Please Provide Feedback
- About This Guide

Documentation Feedback

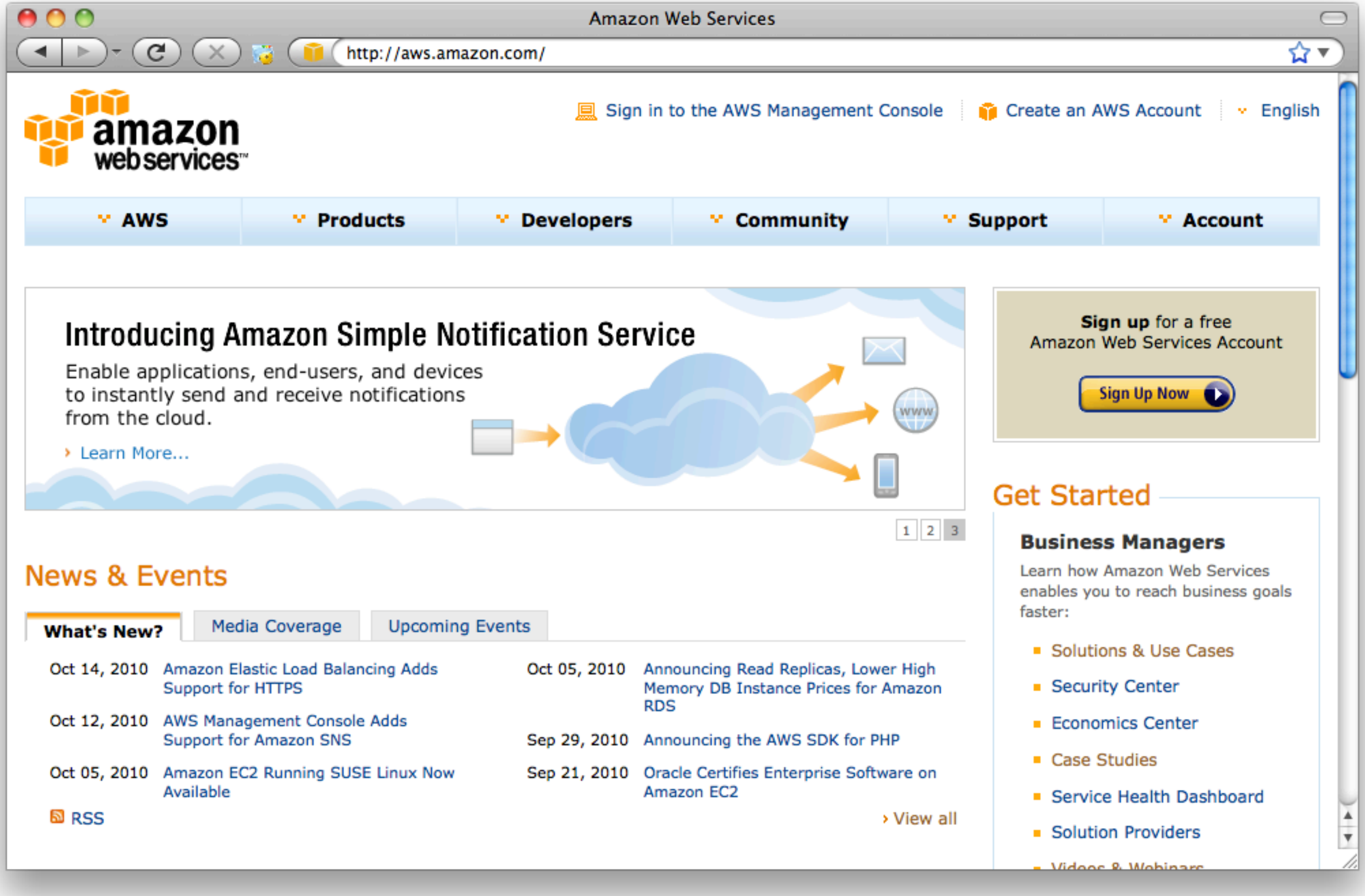
Welcome

Get Started with EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that enables you to launch and manage Linux/UNIX and Windows server instances in Amazon's data centers. You can get started with Amazon EC2 by following the tasks shown in the following diagram. You'll primarily use the AWS Management Console, a point-and-click web-based interface.

This guide walks you through launching and connecting to your first Amazon EC2 instance. To start, click the following **Get Started** button.

Signing Up



Introducing Amazon Simple Notification Service

Enable applications, end-users, and devices to instantly send and receive notifications from the cloud.

› Learn More...

Sign up for a free Amazon Web Services Account

News & Events

Oct 14, 2010	Amazon Elastic Load Balancing Adds Support for HTTPS
Oct 12, 2010	AWS Management Console Adds Support for Amazon SNS
Oct 05, 2010	Amazon EC2 Running SUSE Linux Now Available
Oct 05, 2010	Announcing Read Replicas, Lower High Memory DB Instance Prices for Amazon RDS
Sep 29, 2010	Announcing the AWS SDK for PHP
Sep 21, 2010	Oracle Certifies Enterprise Software on Amazon EC2

RSS

› View all

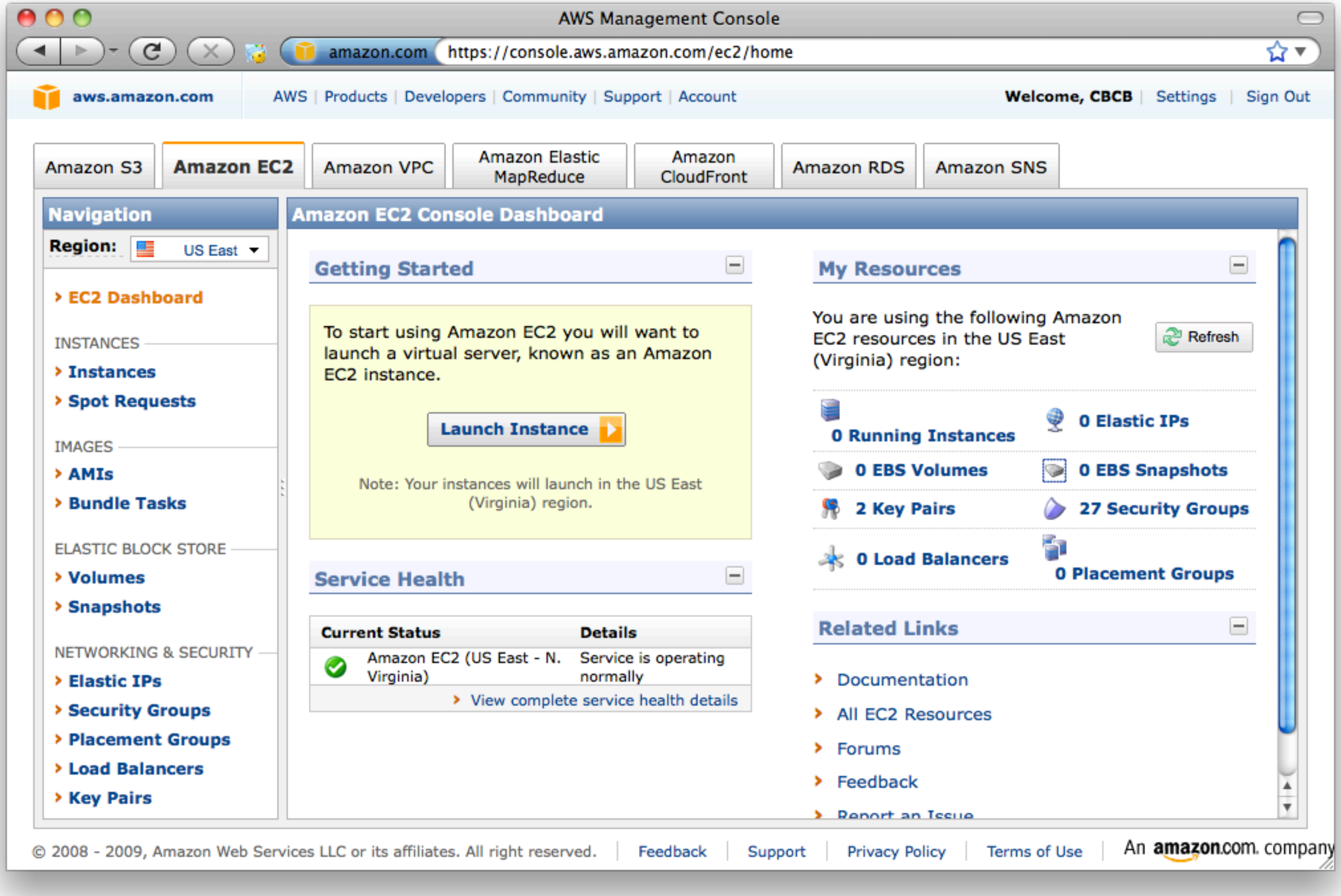
Get Started

Business Managers

Learn how Amazon Web Services enables you to reach business goals faster:

- Solutions & Use Cases
- Security Center
- Economics Center
- Case Studies
- Service Health Dashboard
- Solution Providers
- Videos & Webinars

AWS Management Console



Navigation

Region: US East

- EC2 Dashboard

INSTANCES

- Instances
- Spot Requests

IMAGES

- AMIs
- Bundle Tasks

ELASTIC BLOCK STORE

- Volumes
- Snapshots

NETWORKING & SECURITY

- Elastic IPs
- Security Groups
- Placement Groups
- Load Balancers
- Key Pairs

Amazon EC2 Console Dashboard

Getting Started

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Note: Your instances will launch in the US East (Virginia) region.

Service Health

Current Status	Details
✓ Amazon EC2 (US East - N. Virginia)	Service is operating normally

› View complete service health details

My Resources

You are using the following Amazon EC2 resources in the US East (Virginia) region:

Refresh

0 Running Instances	0 Elastic IPs
0 EBS Volumes	0 EBS Snapshots
2 Key Pairs	27 Security Groups
0 Load Balancers	0 Placement Groups

Related Links

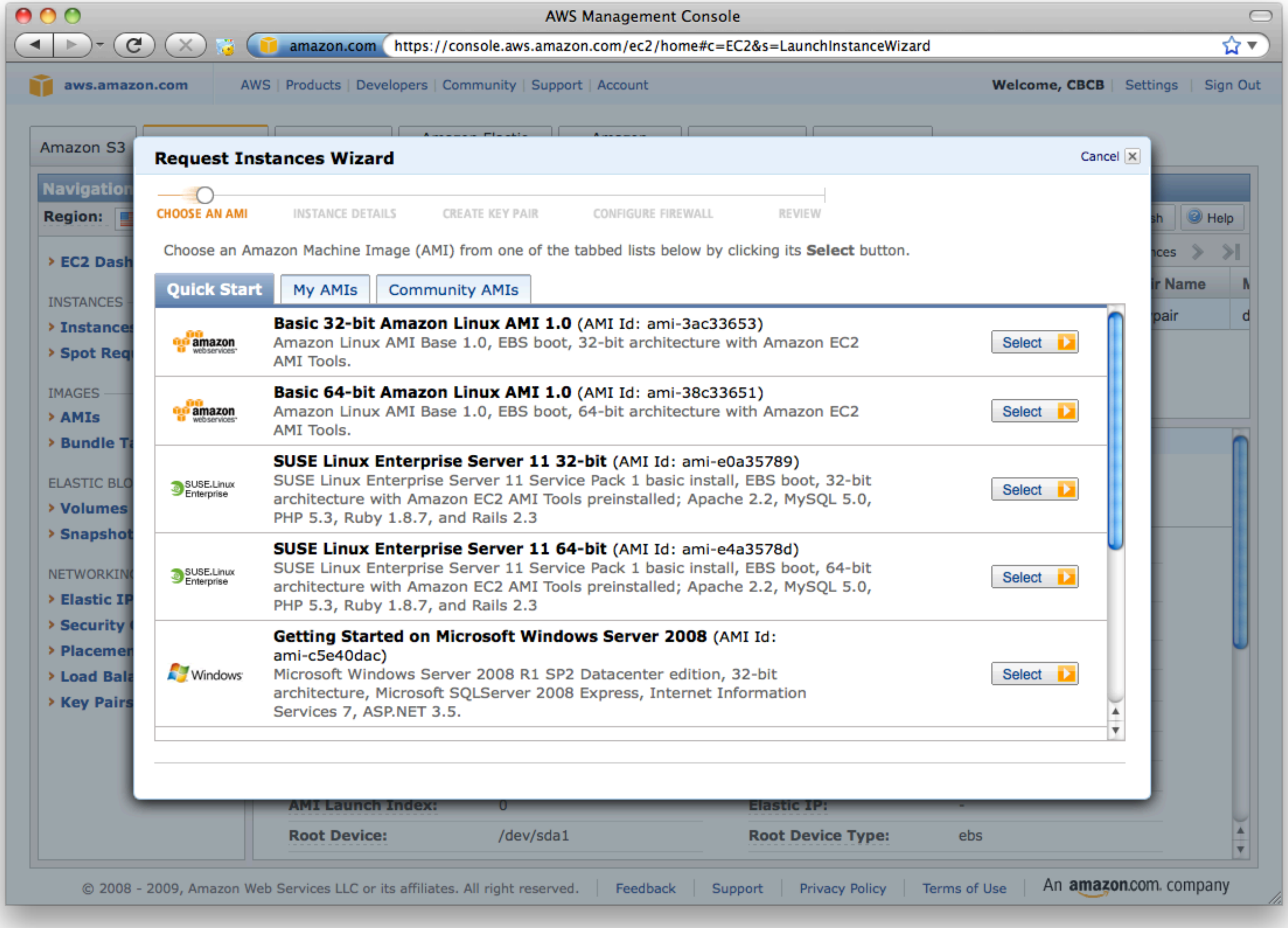
- Documentation
- All EC2 Resources
- Forums
- Feedback
- Report an Issue

© 2008 - 2009, Amazon Web Services LLC or its affiliates. All right reserved. | Feedback | Support | Privacy Policy | Terms of Use | An amazon.com. company

Running your First Cloud Analysis

1. Pick your AMI
   – Machine Image: Operating System & Tools
2. Pick your instance type & quantity
   – Micro - High-Memory Quadruple Extra Large
3. Pick your credentials
   – SSH Keys
4. Configure your Firewall
   – Protect your servers
5. Launch!

1. Pick your AMIs



Request Instances Wizard

Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Basic 32-bit Amazon Linux AMI 1.0 (AMI Id: ami-3ac33653)
Amazon Linux AMI Base 1.0, EBS boot, 32-bit architecture with Amazon EC2 AMI Tools.

Basic 64-bit Amazon Linux AMI 1.0 (AMI Id: ami-38c33651)
Amazon Linux AMI Base 1.0, EBS boot, 64-bit architecture with Amazon EC2 AMI Tools.

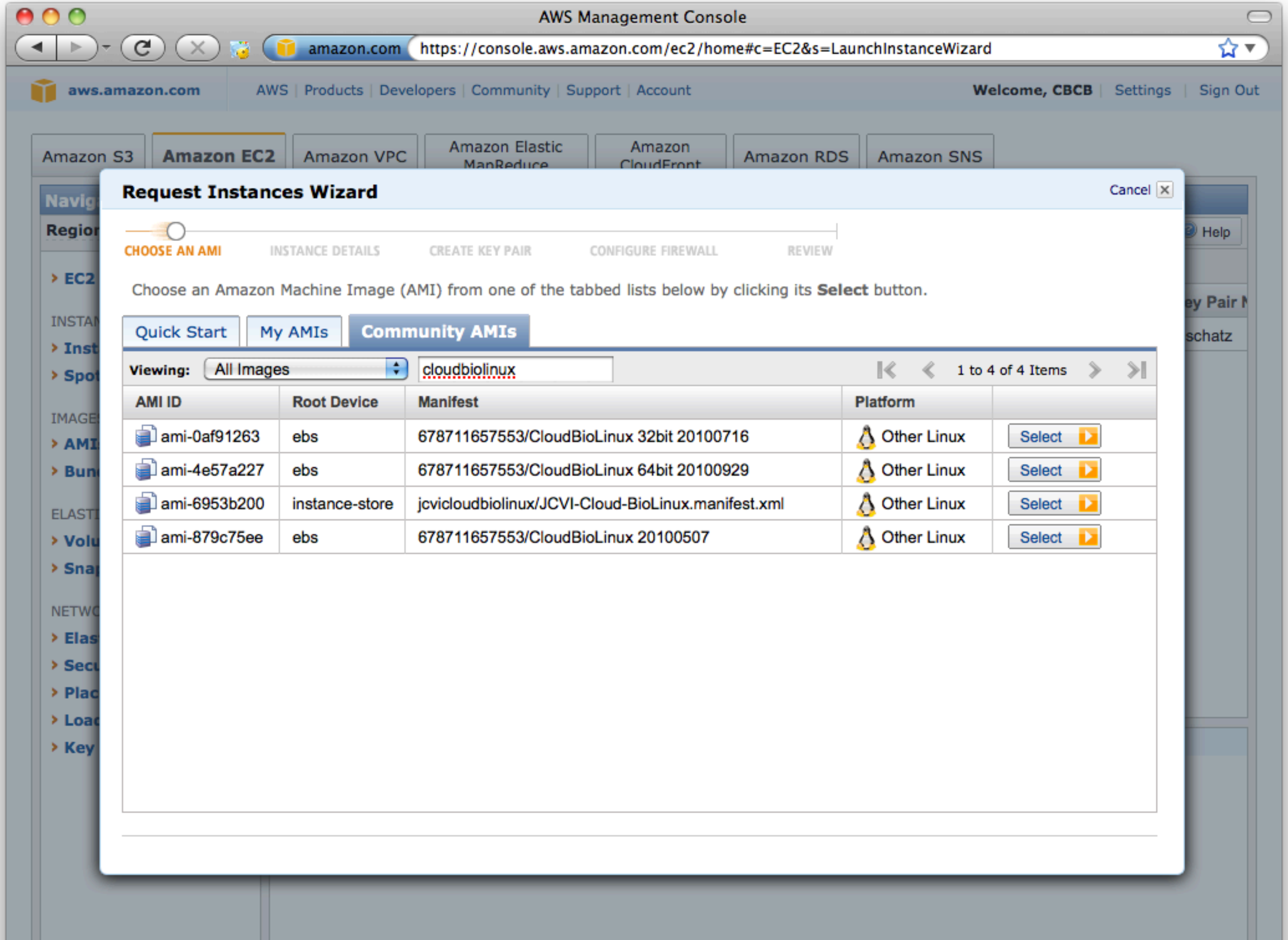
SUSE Linux Enterprise Server 11 32-bit (AMI Id: ami-e0a35789)
SUSE Linux Enterprise Server 11 Service Pack 1 basic install, EBS boot, 32-bit architecture with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, Ruby 1.8.7, and Rails 2.3

SUSE Linux Enterprise Server 11 64-bit (AMI Id: ami-e4a3578d)
SUSE Linux Enterprise Server 11 Service Pack 1 basic install, EBS boot, 64-bit architecture with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, Ruby 1.8.7, and Rails 2.3

Getting Started on Microsoft Windows Server 2008 (AMI Id: ami-c5e40dac)
Microsoft Windows Server 2008 R1 SP2 Datacenter edition, 32-bit architecture, Microsoft SQLServer 2008 Express, Internet Information Services 7, ASP.NET 3.5.

© 2008 - 2009, Amazon Web Services LLC or its affiliates. All right reserved.

CloudBioLinux



Request Instances Wizard

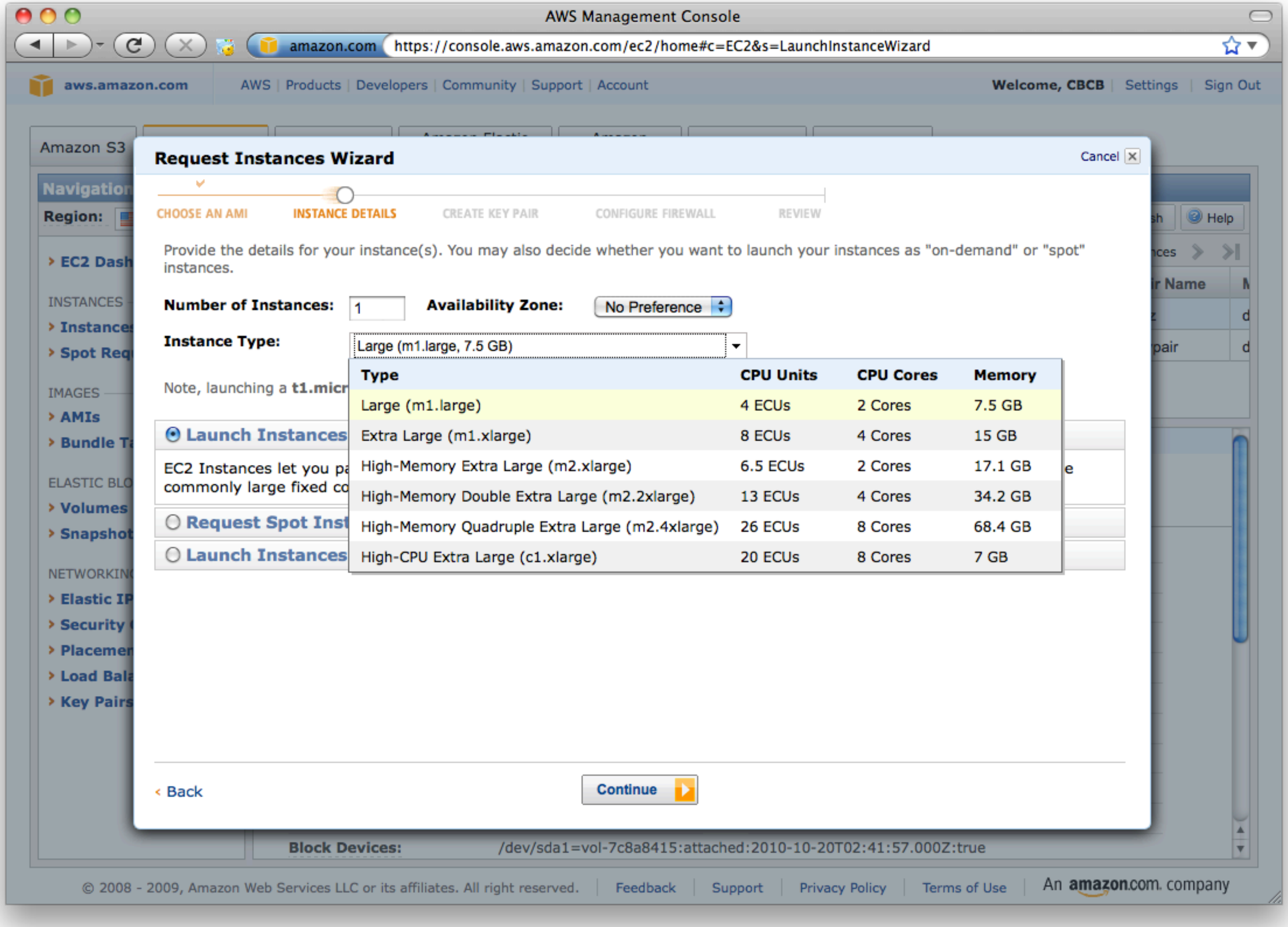
Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Viewing: All Images | cloudbiolinux

1 to 4 of 4 Items

AMI ID	Root Device	Manifest	Platform	
ami-0af91263	ebs	678711657553/CloudBioLinux 32bit 20100716	Other Linux	Select
ami-4e57a227	ebs	678711657553/CloudBioLinux 64bit 20100929	Other Linux	Select
ami-6953b200	instance-store	jcvicloudbiolinux/JCVI-Cloud-BioLinux.manifest.xml	Other Linux	Select
ami-879c75ee	ebs	678711657553/CloudBioLinux 20100507	Other Linux	Select

2. Pick your Instance Type



https://console.aws.amazon.com/ec2/home#c=EC2&s=LaunchInstanceWizard

Request Instances Wizard

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Availability Zone:** No Preference

Instance Type: Large (m1.large, 7.5 GB)

Type	CPU Units	CPU Cores	Memory
Large (m1.large)	4 ECUs	2 Cores	7.5 GB
Extra Large (m1.xlarge)	8 ECUs	4 Cores	15 GB
High-Memory Extra Large (m2.xlarge)	6.5 ECUs	2 Cores	17.1 GB
High-Memory Double Extra Large (m2.2xlarge)	13 ECUs	4 Cores	34.2 GB
High-Memory Quadruple Extra Large (m2.4xlarge)	26 ECUs	8 Cores	68.4 GB
High-CPU Extra Large (c1.xlarge)	20 ECUs	8 Cores	7 GB

Note, launching a **t1.micr**

● **Launch Instances**

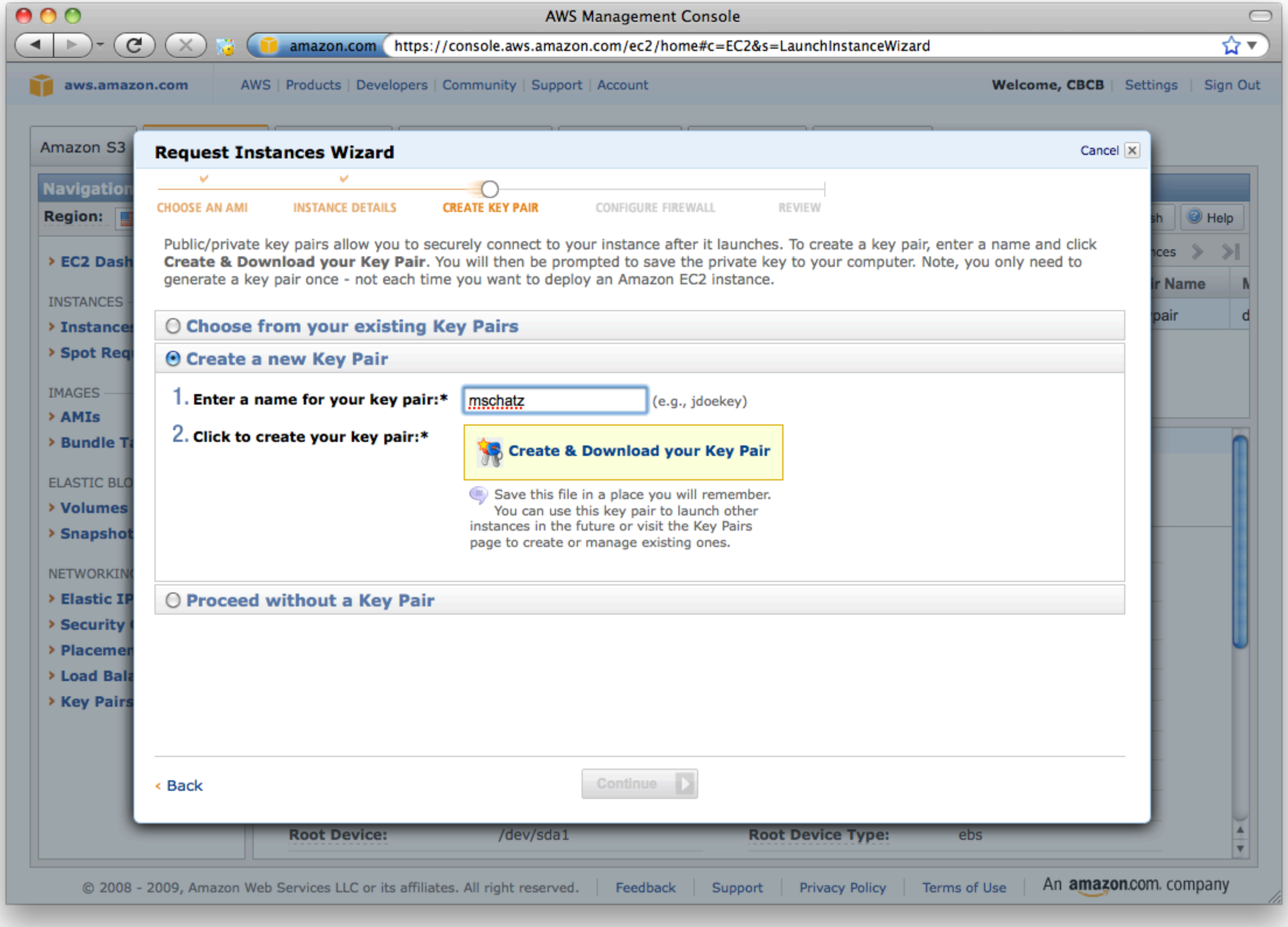
EC2 Instances let you pa … e
commonly large fixed co

○ **Request Spot Inst**

○ **Launch Instances**

Block Devices: /dev/sda1=vol-7c8a8415:attached:2010-10-20T02:41:57.000Z:true

3. Pick your Credentials



https://console.aws.amazon.com/ec2/home#c=EC2&s=LaunchInstanceWizard

Request Instances Wizard

Public/private key pairs allow you to securely connect to your instance after it launches. To create a key pair, enter a name and click **Create & Download your Key Pair**. You will then be prompted to save the private key to your computer. Note, you only need to generate a key pair once - not each time you want to deploy an Amazon EC2 instance.

○ Choose from your existing Key Pairs

● Create a new Key Pair

1. Enter a name for your key pair:* mschatz (e.g., jdoekey)
2. Click to create your key pair:* Create & Download your Key Pair

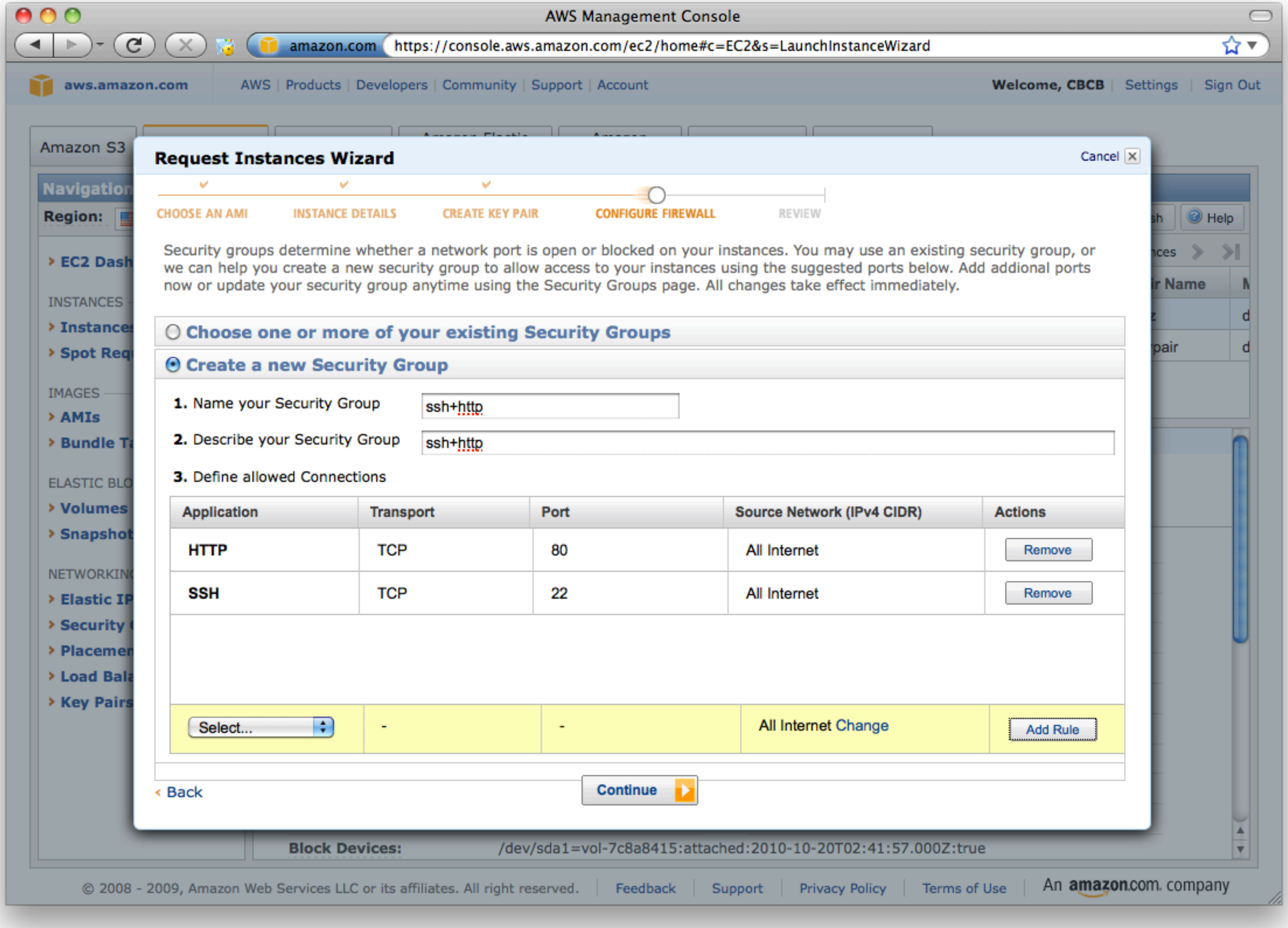
Save this file in a place you will remember. You can use this key pair to launch other instances in the future or visit the Key Pairs page to create or manage existing ones.

○ Proceed without a Key Pair

Root Device: /dev/sda1   Root Device Type: ebs

© 2008 - 2009, Amazon Web Services LLC or its affiliates. All right reserved.

4. Configure your Firewall



AWS Management Console
amazon.com https://console.aws.amazon.com/ec2/home#c=EC2&s=LaunchInstanceWizard
aws.amazon.com
AWS | Products | Developers | Community | Support | Account
Welcome, CBCB | Settings | Sign Out
Amazon S3
Navigation
Region:
EC2 Dash
INSTANCES
Instances
Spot Req
IMAGES
AMIs
Bundle T
ELASTIC BLO
Volumes
Snapshot
NETWORKIN
Elastic IP
Security
Placemen
Load Bala
Key Pairs
Help
Block Devices: /dev/sda1=vol-7c8a8415:attached:2010-10-20T02:41:57.000Z:true
© 2008 - 2009, Amazon Web Services LLC or its affiliates. All right reserved. | Feedback | Support | Privacy Policy | Terms of Use
An amazon.com. company

Request Instances Wizard

Cancel
CHOOSE AN AMI
INSTANCE DETAILS
CREATE KEY PAIR
CONFIGURE FIREWALL
REVIEW

Security groups determine whether a network port is open or blocked on your instances. You may use an existing security group, or we can help you create a new security group to allow access to your instances using the suggested ports below. Add addional ports now or update your security group anytime using the Security Groups page. All changes take effect immediately.

○ Choose one or more of your existing Security Groups

● Create a new Security Group

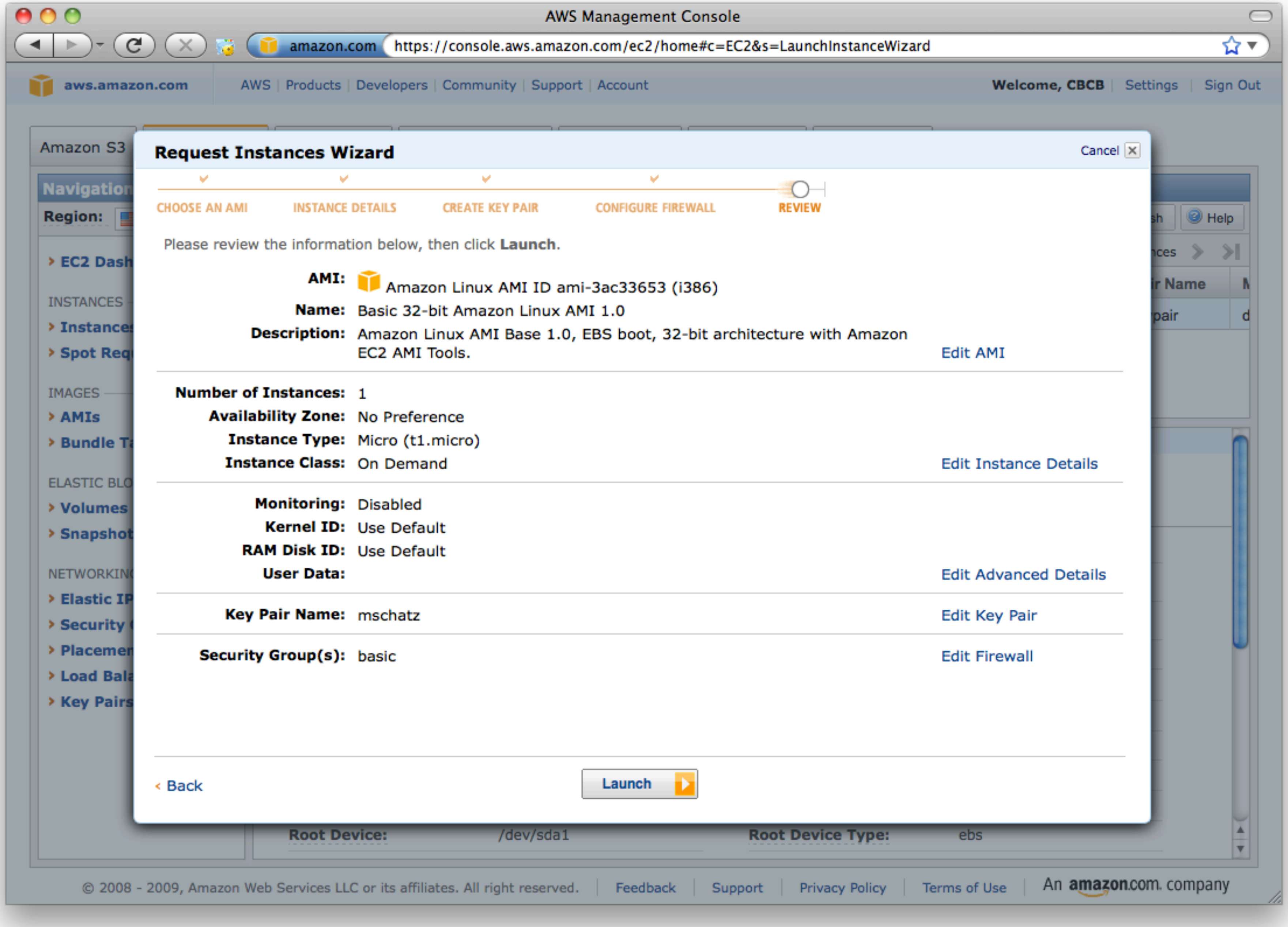
1. Name your Security Group: ssh+http

2. Describe your Security Group: ssh+http

3. Define allowed Connections

Application	Transport	Port	Source Network (IPv4 CIDR)	Actions
HTTP	TCP	80	All Internet	Remove
SSH	TCP	22	All Internet	Remove
Select...	-	-	All Internet Change	Add Rule

5. Launch!



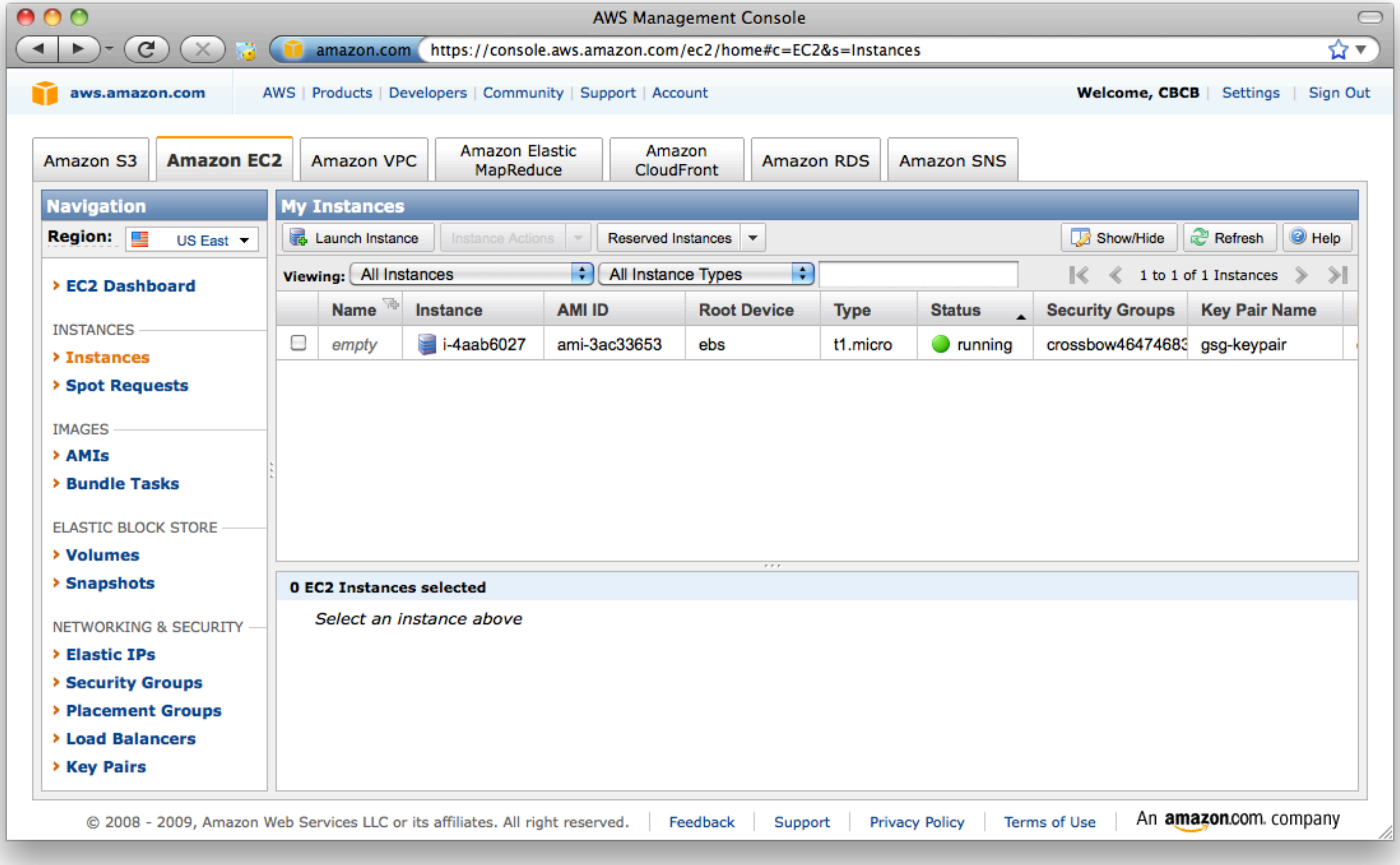
Request Instances Wizard

Please review the information below, then click **Launch**.

AMI:	Amazon Linux AMI ID ami-3ac33653 (i386)	
Name:	Basic 32-bit Amazon Linux AMI 1.0	
Description:	Amazon Linux AMI Base 1.0, EBS boot, 32-bit architecture with Amazon EC2 AMI Tools.	Edit AMI
Number of Instances:	1	
Availability Zone:	No Preference	
Instance Type:	Micro (t1.micro)	
Instance Class:	On Demand	Edit Instance Details
Monitoring:	Disabled	
Kernel ID:	Use Default	
RAM Disk ID:	Use Default	
User Data:		Edit Advanced Details
Key Pair Name:	mschatz	Edit Key Pair
Security Group(s):	basic	Edit Firewall

‹ Back
Launch
Root Device: /dev/sda1
Root Device Type: ebs
© 2008 - 2009, Amazon Web Services LLC or its affiliates. All right reserved.
Feedback | Support | Privacy Policy | Terms of Use
An amazon.com. company

Monitoring your Server



AWS Management Console

https://console.aws.amazon.com/ec2/home#c=EC2&s=Instances

Region: US East

Viewing: All Instances | All Instance Types

1 to 1 of 1 Instances

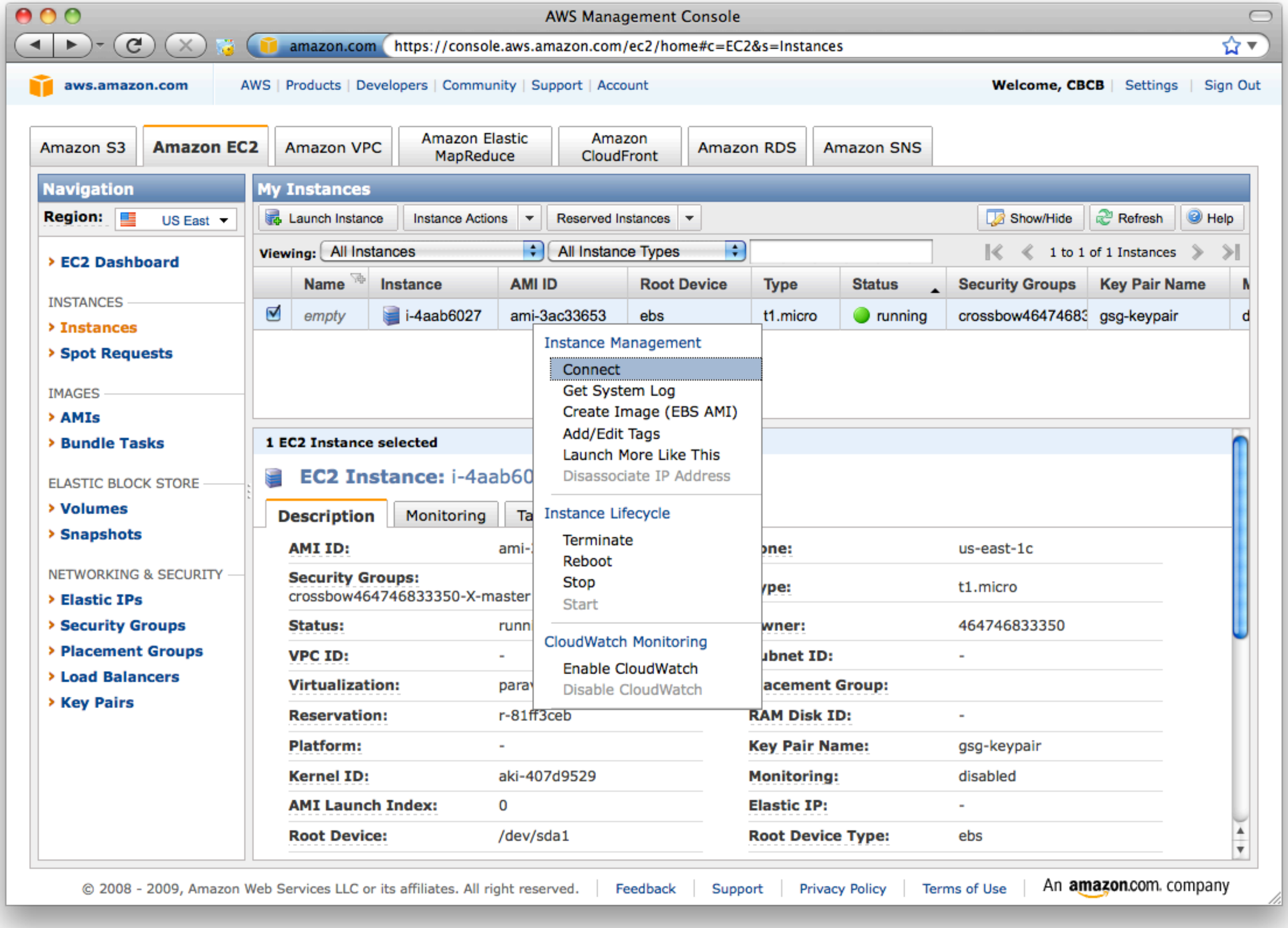
	Name	Instance	AMI ID	Root Device	Type	Status	Security Groups	Key Pair Name
□	*empty*	i-4aab6027	ami-3ac33653	ebs	t1.micro	● running	crossbow46474683	gsg-keypair

0 EC2 Instances selected

Select an instance above

© 2008 - 2009, Amazon Web Services LLC or its affiliates. All right reserved.

Connecting (1)



	Name	Instance	AMI ID	Root Device	Type	Status	Security Groups	Key Pair Name	M
☑	*empty*	i-4aab6027	ami-3ac33653	ebs	t1.micro	● running	crossbow46474683	gsg-keypair	d

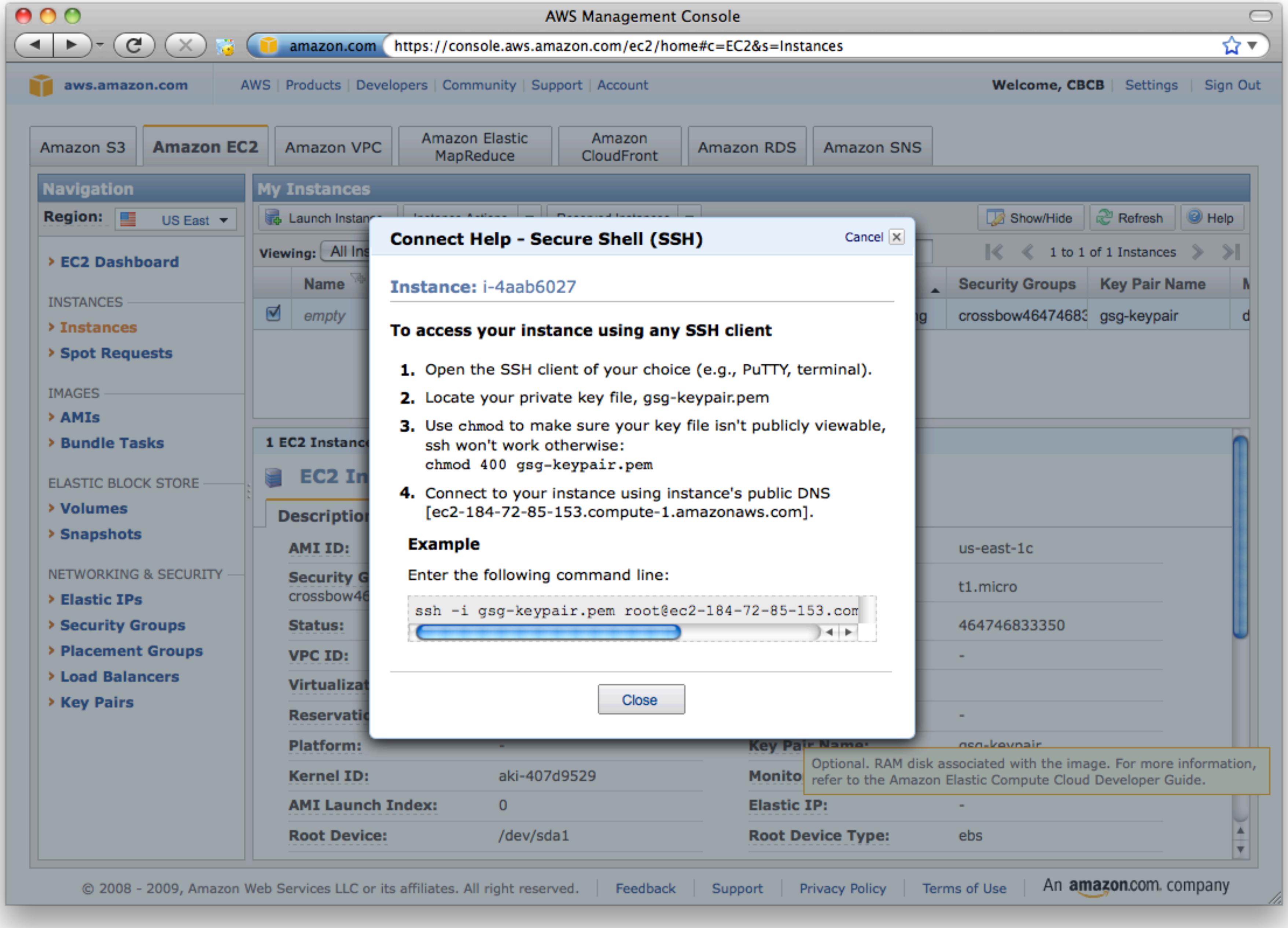
1 EC2 Instance selected

EC2 Instance: i-4aab60

AMI ID:	ami-	one:	us-east-1c
Security Groups: crossbow464746833350-X-master		ype:	t1.micro
Status:	runn	wner:	464746833350
VPC ID:	-	bnet ID:	-
Virtualization:	para	acement Group:	
Reservation:	r-81ff3ceb	RAM Disk ID:	-
Platform:	-	Key Pair Name:	gsg-keypair
Kernel ID:	aki-407d9529	Monitoring:	disabled
AMI Launch Index:	0	Elastic IP:	-
Root Device:	/dev/sda1	Root Device Type:	ebs

© 2008 - 2009, Amazon Web Services LLC or its affiliates. All right reserved. | Feedback | Support | Privacy Policy | Terms of Use | An amazon.com. company

Connecting (2)



https://console.aws.amazon.com/ec2/home#c=EC2&s=Instances

Name	Security Groups	Key Pair Name
empty	crossbow4647468	gsg-keypair

Connect Help - Secure Shell (SSH)

Instance: i-4aab6027

To access your instance using any SSH client

1. Open the SSH client of your choice (e.g., PuTTY, terminal).
2. Locate your private key file, gsg-keypair.pem
3. Use `chmod` to make sure your key file isn't publicly viewable, ssh won't work otherwise:
   `chmod 400 gsg-keypair.pem`
4. Connect to your instance using instance's public DNS [ec2-184-72-85-153.compute-1.amazonaws.com].

Example

Enter the following command line:

```
ssh -i gsg-keypair.pem root@ec2-184-72-85-153.com
```

AMI ID:			us-east-1c
Security G crossbow46			t1.micro
Status:			464746833350
VPC ID:			-
Virtualizat			
Reservati			-
Platform:	-	Key Pair Name:	gsg-keypair
Kernel ID:	aki-407d9529	Monito	
AMI Launch Index:	0	Elastic IP:	-
Root Device:	/dev/sda1	Root Device Type:	ebs

Optional. RAM disk associated with the image. For more information, refer to the Amazon Elastic Compute Cloud Developer Guide.

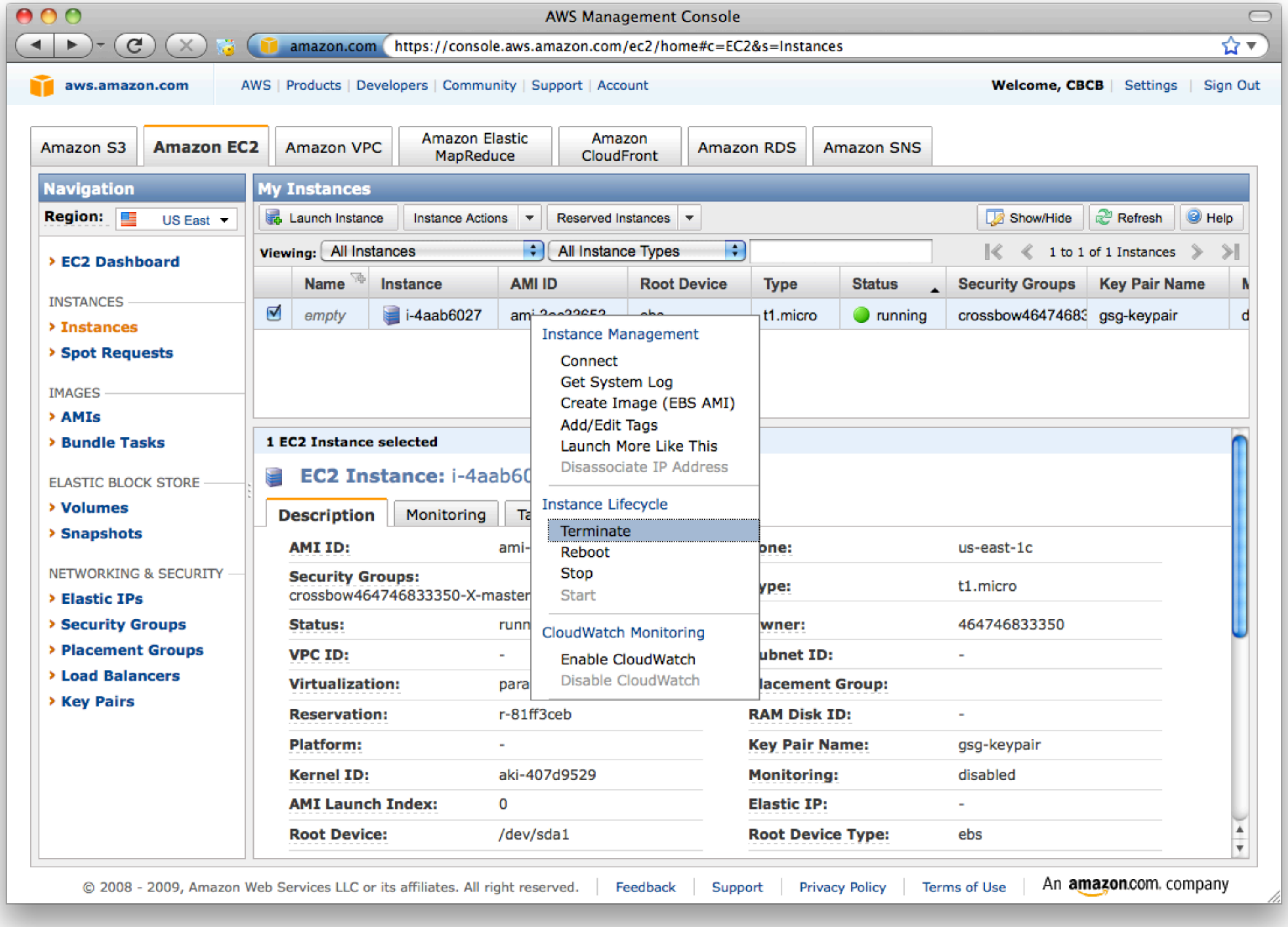
Calling SNPs in the Cloud ☺

```
chmod 400 mschatz.pem

scp -r -i mschatz.pem data.tgz ubuntu@ec2-174-129-123-73.compute-1.amazonaws.com:
ssh -i mschatz.pem ubuntu@ec2-174-129-123-73.compute-1.amazonaws.com

<remote>

ls

tar xzvf data.tgz
bowtie -S data/genomes/e_coli data/reads/e_coli_10000snp.fq ec_snp.sam
samtools view -bS -o ec_snp.bam ec_snp.sam
samtools sort ec_snp.bam ec_snp.sorted

samtools pileup -cv -f data/genomes/NC_008253.fna ec_snp.sorted.bam > snps

samtools index ec_snp.sorted.bam
samtools tview ec_snp.sorted.bam data/genomes/NC_008253.fna

exit

<local>

scp -i mschatz.pem ubuntu@ec2-174-129-123-73.compute-1.amazonaws.com:snps .
```

1000Genomes in the Cloud

```
s3cmd --configure

# cp data/.s3cfg .

s3cmd ls s3://1000genomes

s3cmd ls s3://1000genomes/Pilots_Bam/NA20828/

s3cmd get s3://1000genomes/Pilots_Bam/NA20828/*chr22* .

samtools view NA20828.SLX.maq.SRP000033.2009_09.chr22_1_49691432.bam
```

Terminating



	Name	Instance	AMI ID	Root Device	Type	Status	Security Groups	Key Pair Name	M
☑	*empty*	i-4aab6027	ami-3aa33653	ebs	t1.micro	● running	crossbow4647468	gsg-keypair	d

AMI ID:	ami-	**one:**	us-east-1c
Security Groups: crossbow464746833350-X-master		**ype:**	t1.micro
Status:	runn	**wner:**	464746833350
VPC ID:	-	**ubnet ID:**	-
Virtualization:	para	**lacement Group:**	
Reservation:	r-81ff3ceb	**RAM Disk ID:**	-
Platform:	-	**Key Pair Name:**	gsg-keypair
Kernel ID:	aki-407d9529	**Monitoring:**	disabled
AMI Launch Index:	0	**Elastic IP:**	-
Root Device:	/dev/sda1	**Root Device Type:**	ebs

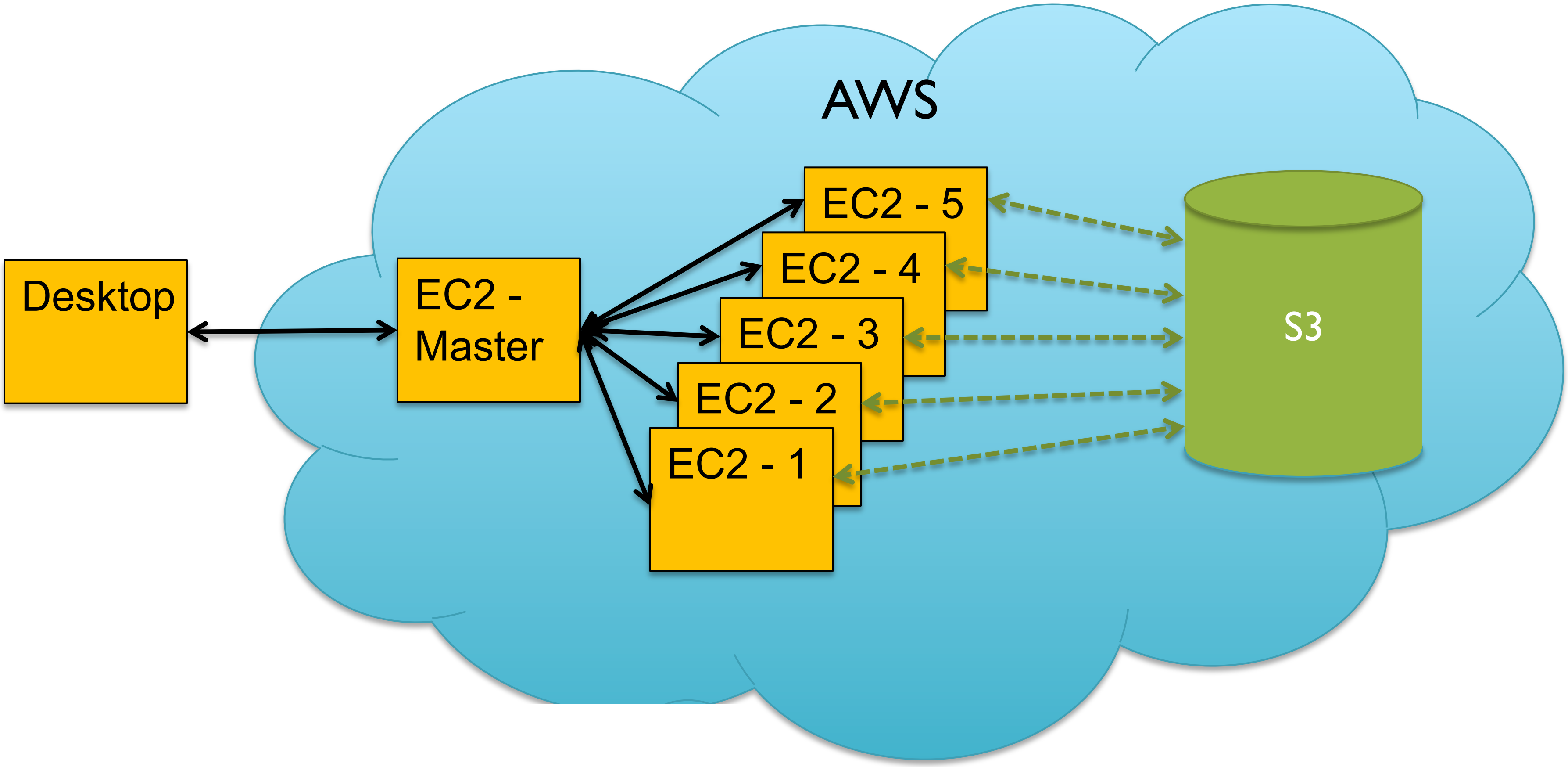
Total cost: 8.5¢

Reflections

- Launching and managing virtual clusters with the AWS Console is quick and easy
  - Entirely scriptable using ec2 tools
  - iPhone App also available

- Things get really interesting on 168 cores
  - 1 week CPU = 1 hour wall

Hadoop on AWS



Just 3 commands to bring up a 168 core (21 node) cluster & crunch terabytes:

```
$HADOOP/src/contrib/ec2/bin/hadoop-ec2 launch-cluster HADOOP 21
$HADOOP/src/contrib/ec2/bin/hadoop-ec2 <hadoop cmd> HADOOP
$HADOOP/src/contrib/ec2/bin/hadoop-ec2 terminate-cluster HADOOP
```

Kmer Code

kmer-map.pl

```
#!/usr/bin/perl

my $K = 15;

while (<>)
{
 for (my $i = 0;
      $i < length($_)-$K;
      $i++)
 {
  my $kmer = substr($_,$i,$K);
  print "$kmer\t1\n";
 }
}
```

kmer-reduce.pl

```
#!/usr/bin/perl

my $mer = "";
my $cnt = 0;

while (<>)
{
  chomp;

  my ($curmer, $curcnt) = split /\t/, $_;

  if ($curmer ne $mer)
  {
    print "$mer\t$cnt\n" if ($cnt > 0);
    $mer = $curmer; $cnt = 0;
  }

 $cnt += $curcnt;
}

print "$mer\t$cnt\n" if ($cnt > 0);
```

BashReduce

```
$ head -3 reads.txt
ATATTTTTTCTTGTTTTTTATATCCACAAACTCTTT
CCACAAAATCAATACCTTGTGGAATAAAATTGTCCA
TATTTTTTCTTGTTTTTTATATCCACAAACTCTTTT

$ cat reads.txt | ./kmer-map.pl | head -3
ATATTTTTTCTTGTT   1
TATTTTTTCTTGTTT   1
ATTTTTTCTTGTTTT   1

$ cat reads.txt | ./kmer-map.pl | sort \
  | ./kmer-reduce.pl | head -3
AAAAAAAAGTAGCTA   44
AAAAAAAGTAGCTAT   44
AAAAAAGCAAATGTG   17
```

Kmer Counting In Hadoop

```
#!/bin/sh

STREAMING=/usr/lib/hadoop-0.20/contrib/streaming/hadoop-
   streaming-0.20.2+320.jar

hadoop fs -mkdir /user/mschatz/kmertest/reads
hadoop fs -put reads.txt /user/mschatz/kmertest/reads

hadoop jar $STREAMING \
 -input  /user/mschatz/kmertest/reads \
 -output /user/mschatz/kmertest/kmers \
 -mapper ./kmer-map.pl \
 -reducer ./kmer-reduce.pl \
 -file ./kmer-map.pl \
 -file ./kmer-reduce.pl \
 -jobconf mapred.map.tasks=10 \
 -jobconf mapred.reduce.tasks=1

hadoop fs -cat /user/mschatz/kmertest/kmers/part-* | head -3
hadoop fs -rmr /user/mschatz/kmertest
```

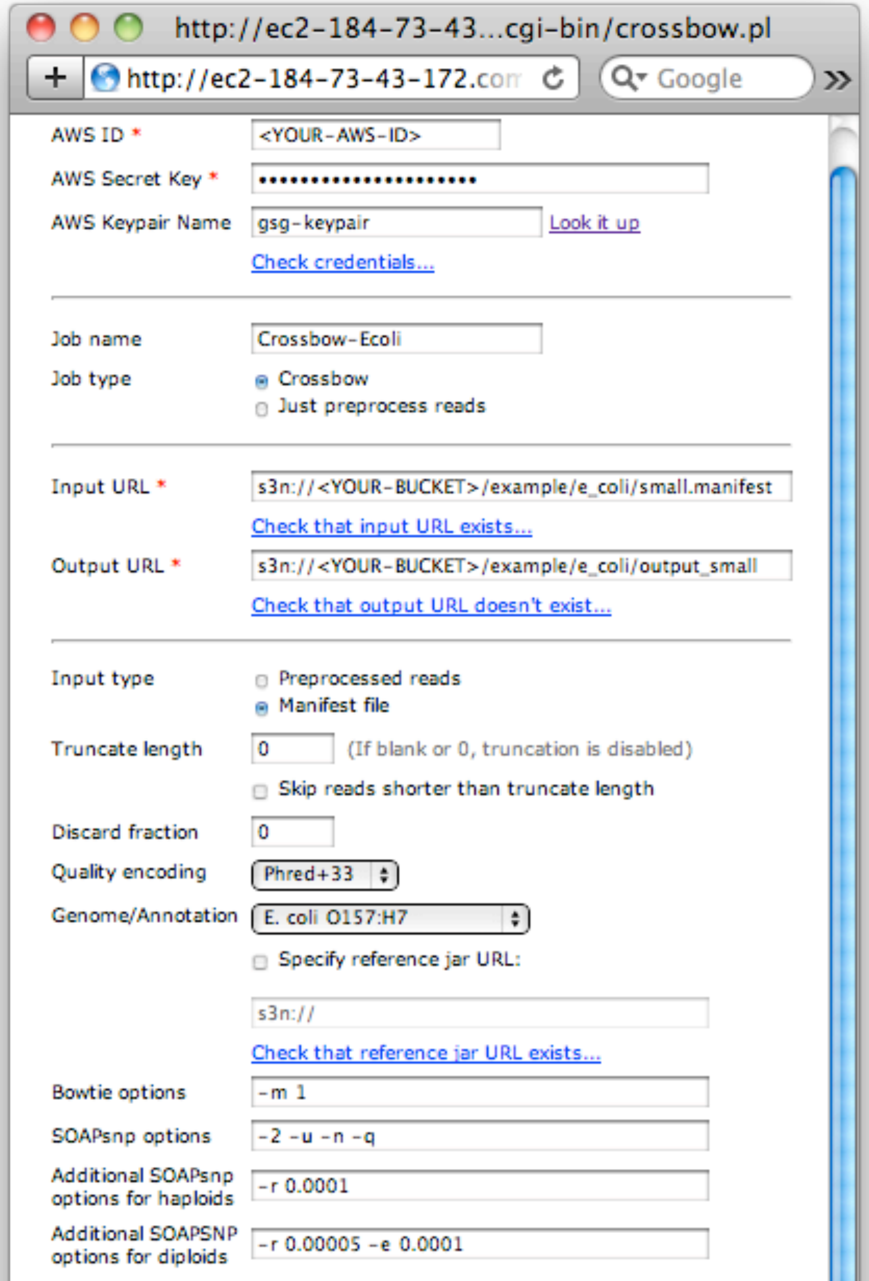
Hadoop Output

```
10/10/21 16:03:51 INFO mapred.FileInputFormat: Total input paths to process : 1
10/10/21 16:03:51 INFO streaming.StreamJob: getLocalDirs(): [/scratch0/hadoop/mapred/
    local]
10/10/21 16:03:51 INFO streaming.StreamJob: Running job: job_201009232028_2089
10/10/21 16:03:51 INFO streaming.StreamJob: To kill this job, run:
10/10/21 16:03:51 INFO streaming.StreamJob: /usr/lib/hadoop-0.20/bin/hadoop job  -
    Dmapred.job.tracker=szhdname01.umiacs.umd.edu:8021 -kill job_201009232028_2089
10/10/21 16:03:51 INFO streaming.StreamJob: Tracking URL: http://
    szhdname01.umiacs.umd.edu:50030/jobdetails.jsp?jobid=job_201009232028_2089
10/10/21 16:03:52 INFO streaming.StreamJob:  map 0%  reduce 0%
10/10/21 16:03:58 INFO streaming.StreamJob:  map 30%  reduce 0%
10/10/21 16:04:01 INFO streaming.StreamJob:  map 100%  reduce 0%
10/10/21 16:04:07 INFO streaming.StreamJob:  map 100%  reduce 20%
10/10/21 16:04:16 INFO streaming.StreamJob:  map 100%  reduce 100%
10/10/21 16:04:19 INFO streaming.StreamJob: Job complete: job_201009232028_2089
10/10/21 16:04:19 INFO streaming.StreamJob: Output: /user/mschatz/kmertest/kmers
```

```
AAAAAAAAGTAGCTA     44
AAAAAAAGTAGCTAT     44
AAAAAAGCAAATGTG     17
```

Crossbow Webform

http://bowtie-bio.sf.net/crossbow/ui.html


http://ec2-184-73-43...cgi-bin/crossbow.pl
+
http://ec2-184-73-43-172.com
Google

AWS ID *	<YOUR-AWS-ID>
AWS Secret Key *	•••••••••••••••••••••
AWS Keypair Name	gsg-keypair Look it up
	Check credentials...
Job name	Crossbow-Ecoli
Job type	● Crossbow ○ Just preprocess reads
Input URL *	s3n://<YOUR-BUCKET>/example/e_coli/small.manifest
	Check that input URL exists...
Output URL *	s3n://<YOUR-BUCKET>/example/e_coli/output_small
	Check that output URL doesn't exist...
Input type	○ Preprocessed reads ● Manifest file
Truncate length	0 (If blank or 0, truncation is disabled)
	□ Skip reads shorter than truncate length
Discard fraction	0
Quality encoding	Phred+33
Genome/Annotation	E. coli O157:H7
	□ Specify reference jar URL:
	s3n://
	Check that reference jar URL exists...
Bowtie options	-m 1
SOAPsnp options	-2 -u -n -q
Additional SOAPsnp options for haploids	-r 0.0001
Additional SOAPSNP options for diploids	-r 0.00005 -e 0.0001

- Enter your account info, manifest file, reference info, and pipeline settings
  - List of URLs to fastq files

- Crossbow
  - Parallel ftp
  - Parallel map
  - Parallel SNPs

More Information

- Amazon Web Services
  - http://aws.amazom.com
  - http://aws.amazon.com/free
- Hadoop
  - http://hadoop.apache.org
- Crossbow & Bowtie
  - http://bowtie-bio.sf.net

Thank You!

http://schatzlab.cshl.edu

@mike_schatz