



Java is a trademark of Sun Microsystems, Inc.



JavaOneSM

Matchmaking in the Cloud: Amazon EC2 and Apache Hadoop at eHarmony

Steve Kuo, Software Architect
Joshua Tuberville, Software Architect

Goal

- > Leverage EC2 and Hadoop to scale data intensive problem that are exceeding the limits of our data center and data warehouse environment.

Agenda

- > Background and Motivation
- > Hadoop Overview
- > Hadoop at eHarmony
- > Architecture
- > Tools and Performance
- > Roadblocks
- > Future Directions
- > Summary
- > Questions

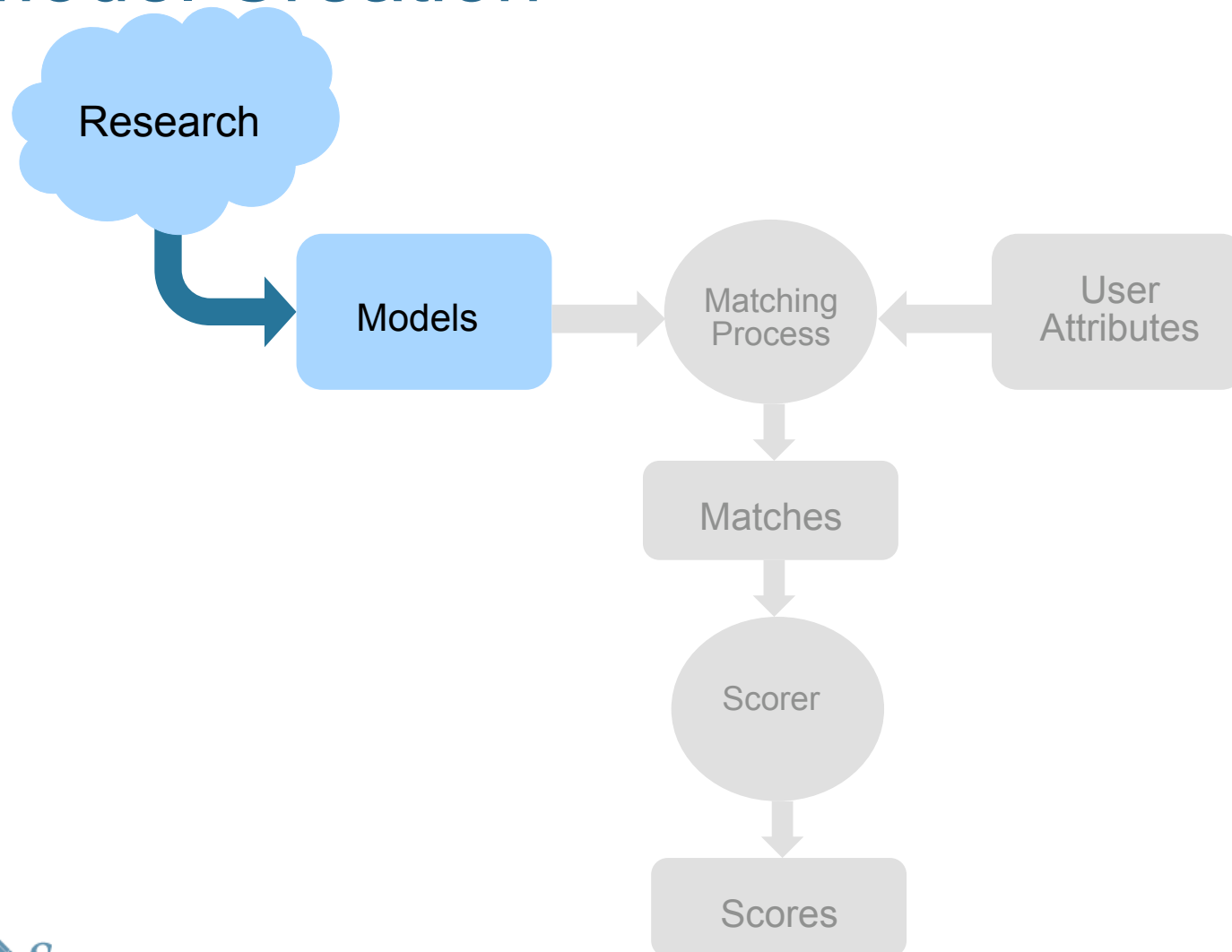
About eHarmony

- > Online subscription-based matchmaking service
- > Launched in 2000 using compatibility models
- > Available in United States, Canada, Australia and United Kingdom.
- > On average, 236 members in US marry every day.
- > More than 20 million registered users.

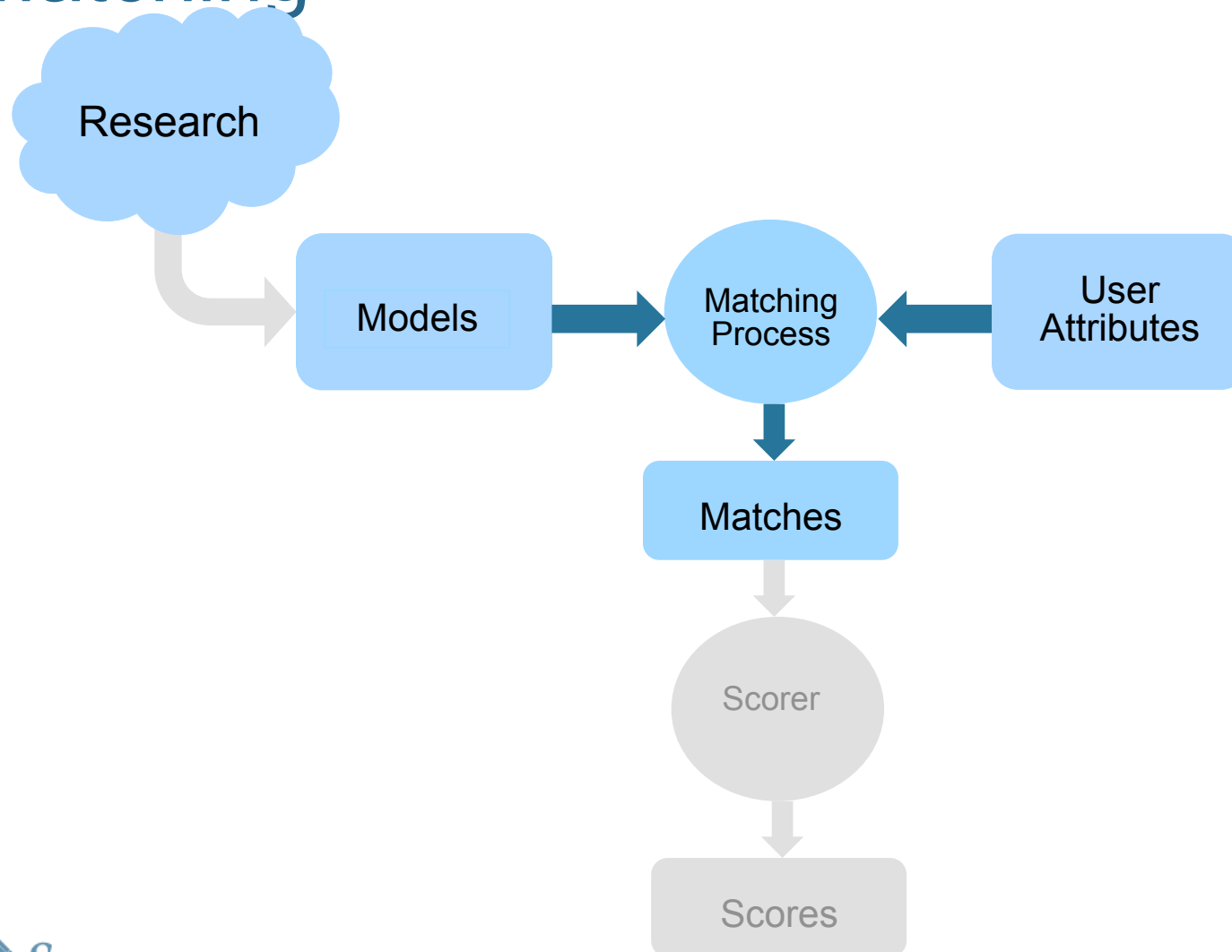
About eHarmony

- > Models are based on decades of research and clinical experience in psychology
- > Variety of user attributes
 - Demographic
 - Psychographic
 - Behavioral

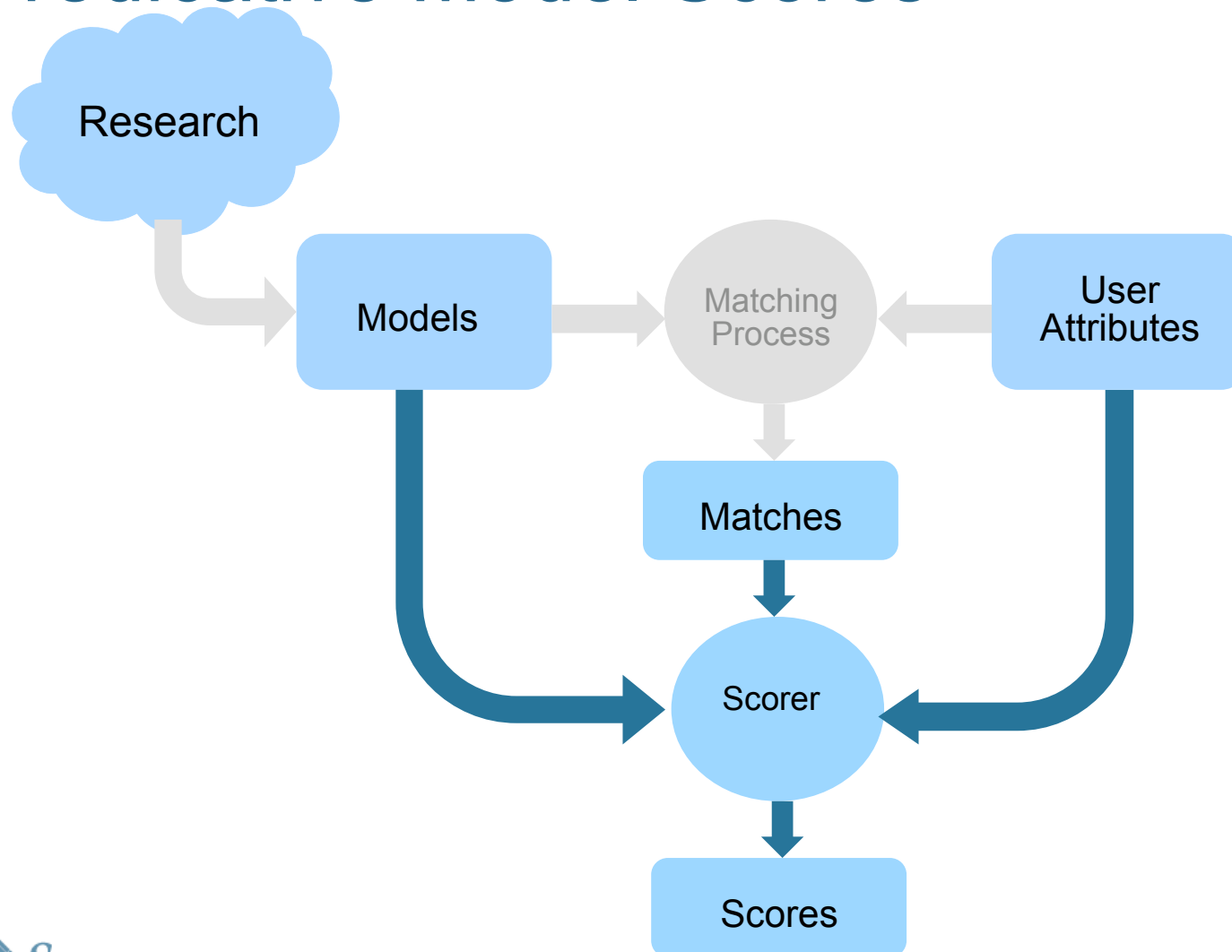
Model Creation



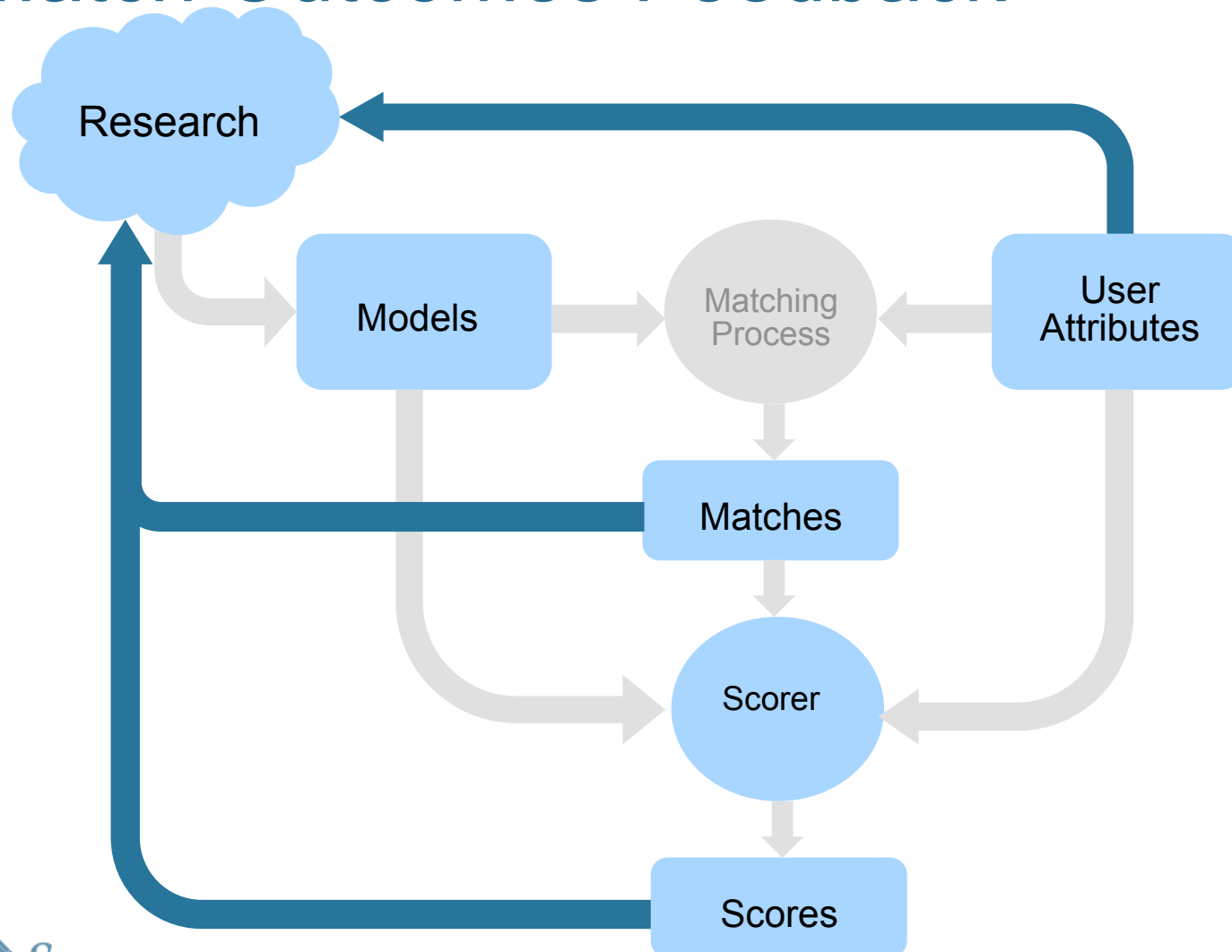
Matching



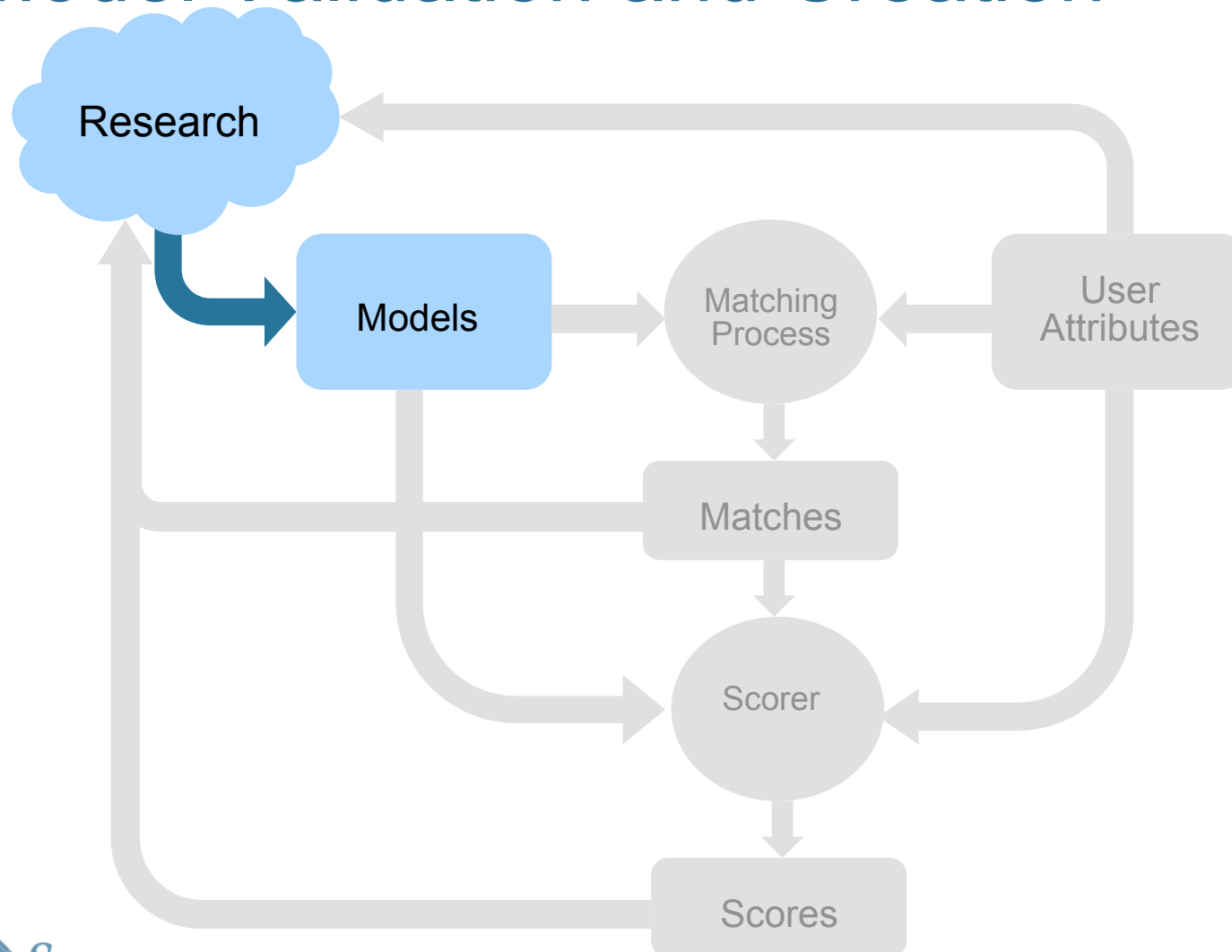
Predicative Model Scores



Match Outcomes Feedback



Model Validation and Creation



Scorer Requirements

- > Tens of GB of matches, scores and constantly changing user features are archived daily
- > TB of data currently archived and growing
- > 10x our current user base
- > All possible matches = $O(n^2)$ problem
- > Support a growing set of models that may be
 - arbitrarily complex
 - computationally and I/O expensive.

Scaling Challenges

- > Current architecture is multi-tiered with a relational back-end
- > Scoring is DB join intensive
- > Data need constant archiving
 - Matches, match scores, user attributes at time of match creation
 - Model validation is done at a later time across many days
- > Need a non-DB solution



- > Open Source Java implementation of Google's MapReduce paper
 - Created by Doug Cutting
- > Top-level Apache project
 - Yahoo is a major sponsor



- > Distributes work across vast amounts of data



- > Distributes work across vast amounts of data
- > Hadoop Distributed File System (HDFS) provides reliability through replication



- > Distributes work across vast amounts of data
- > Hadoop Distributed File System (HDFS) provides reliability through replication
- > Automatic re-execution on failure/distribution



- > Distributes work across vast amounts of data
- > Hadoop Distributed File System (HDFS) provides reliability through replication
- > Automatic re-execution on failure/distribution
- > Scale horizontally on commodity hardware



- > Simple Storage Service (S3) provides cheap unlimited storage

Storage	\$0.15/GB	First 50 TB/Month
Transfer	\$0.17/GB	First 10 TB/Month



- > Simple Storage Service (S3) provides cheap unlimited storage.

Storage	\$0.15/GB	First 50 TB/Month
Transfer	\$0.17/GB	First 10 TB/Month

- > Elastic Cloud Computing (EC2) enables horizontal scaling by adding servers on demand.

C1-medium	\$0.2/hour	2 cores	1.7 GB Memory	32-bit OS
C1-xlarge	\$0.8/hour	8 cores	7 GB Memory	64-bit OS



- > Elastic MapReduce is a hosted Hadoop framework on top EC2 and S3.
- > It's in beta and US only.
- > Pricing is in addition to EC2 and S3.

Instance Type	EC2	Elastic MapReduce
C1-medium	\$0.2/hour	\$0.03/hour
C1-xlarge	\$0.8/hour	\$0.12/hour

Agenda

- > Background and Motivation
- > **Hadoop Overview**
- > Hadoop at eHarmony
- > Architecture
- > Tools and Performance
- > Roadblocks
- > Future Directions
- > Summary
- > Questions

How MapReduce Works

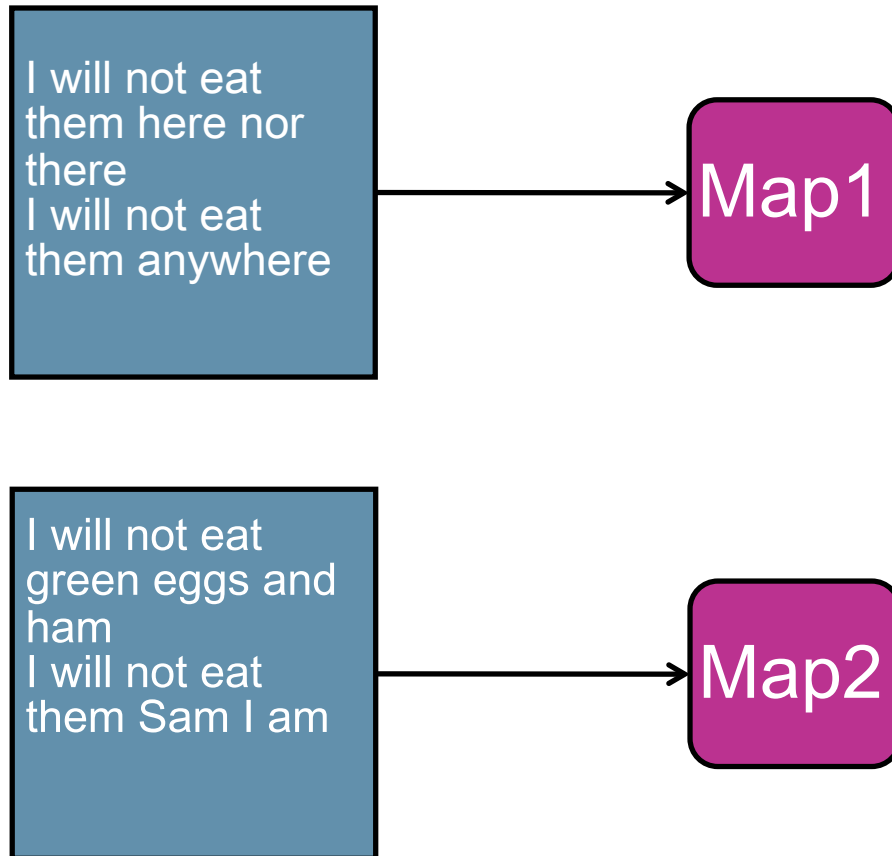
- > Applications are modeled as a series of maps and reductions
- > Example - Word Count
 - Counts the frequency of words
 - Modeled as one Map and one Reduce
 - Data as key -> values

Word Count - Input

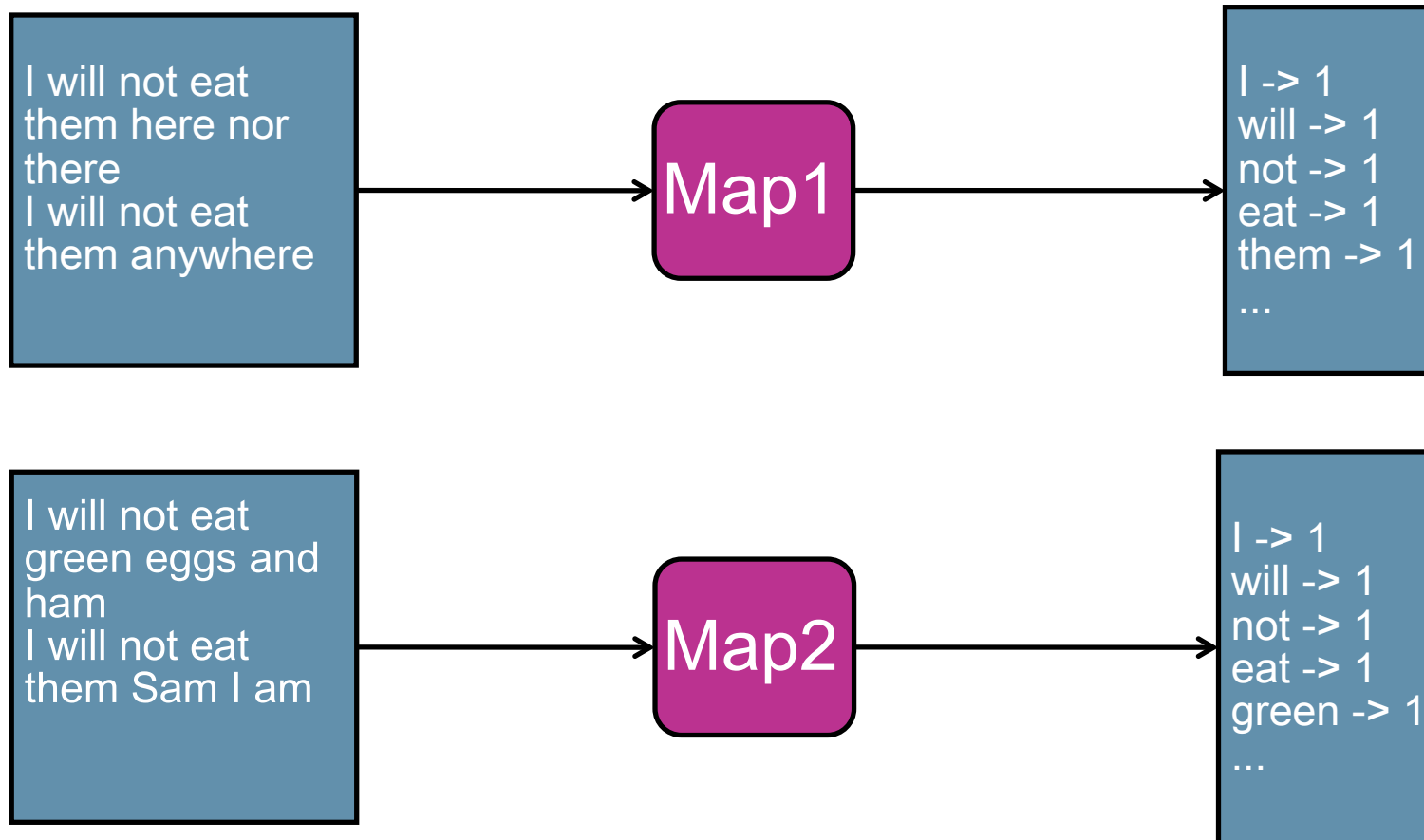
I will not eat
them here nor
there
I will not eat
them anywhere

I will not eat
green eggs and
ham
I will not eat
them Sam I am

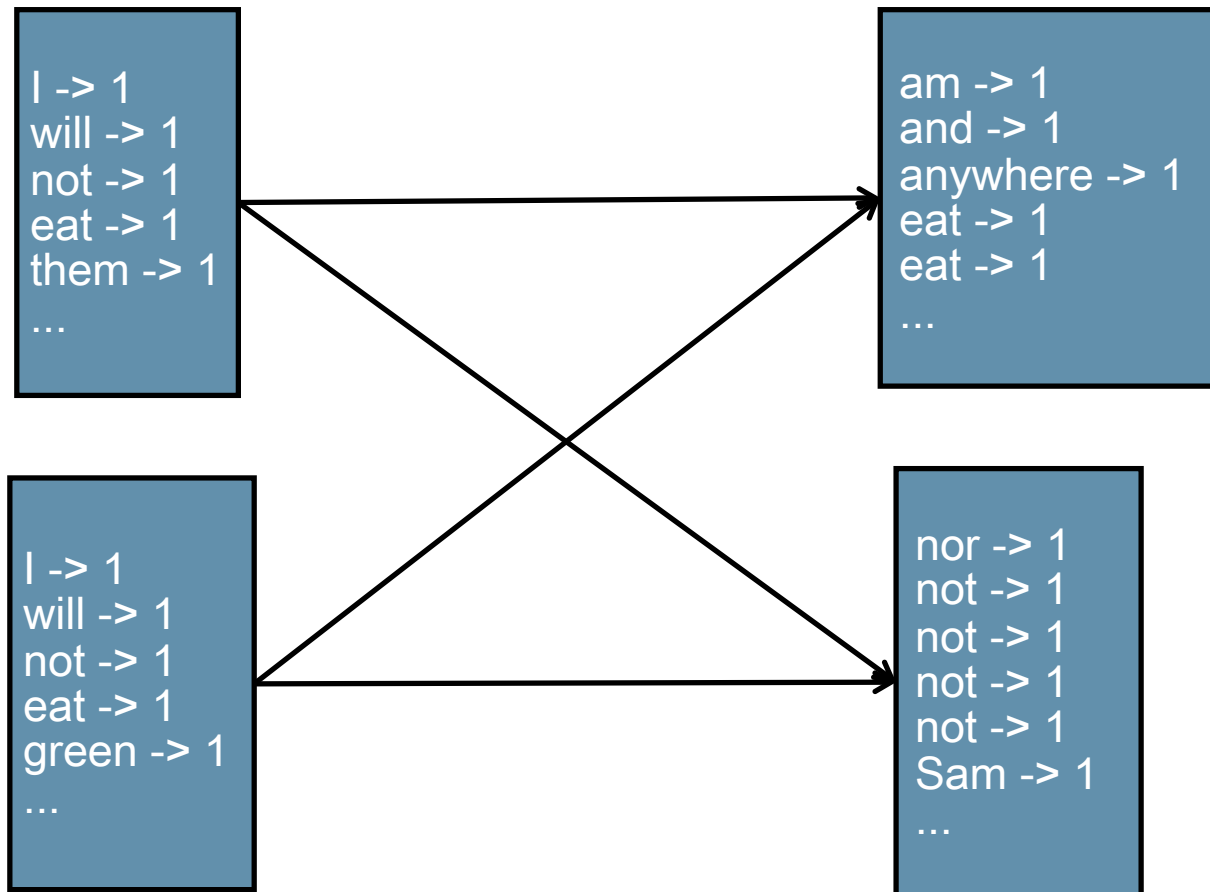
Word Count – Pass to Mappers



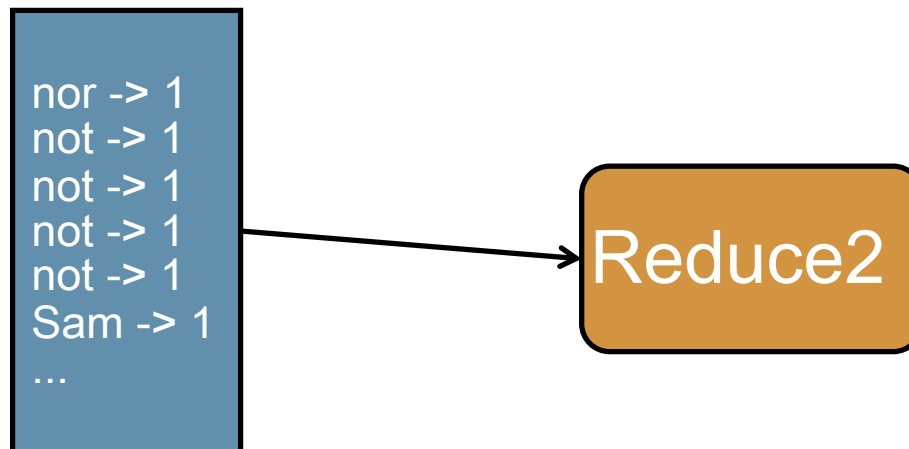
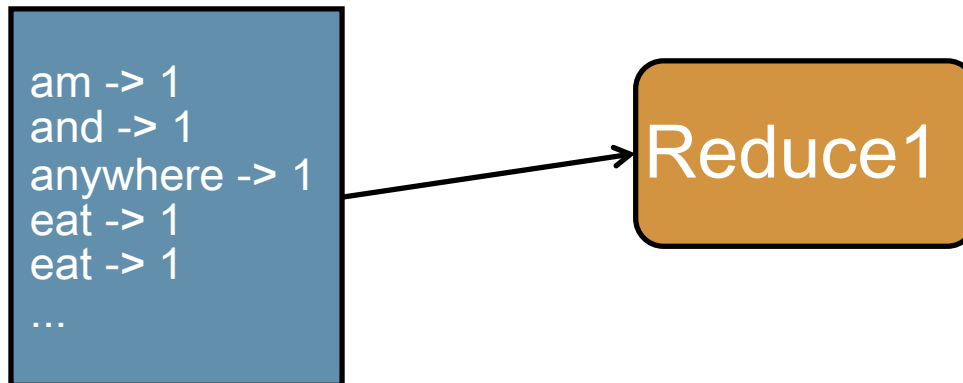
Word Count – Perform Map



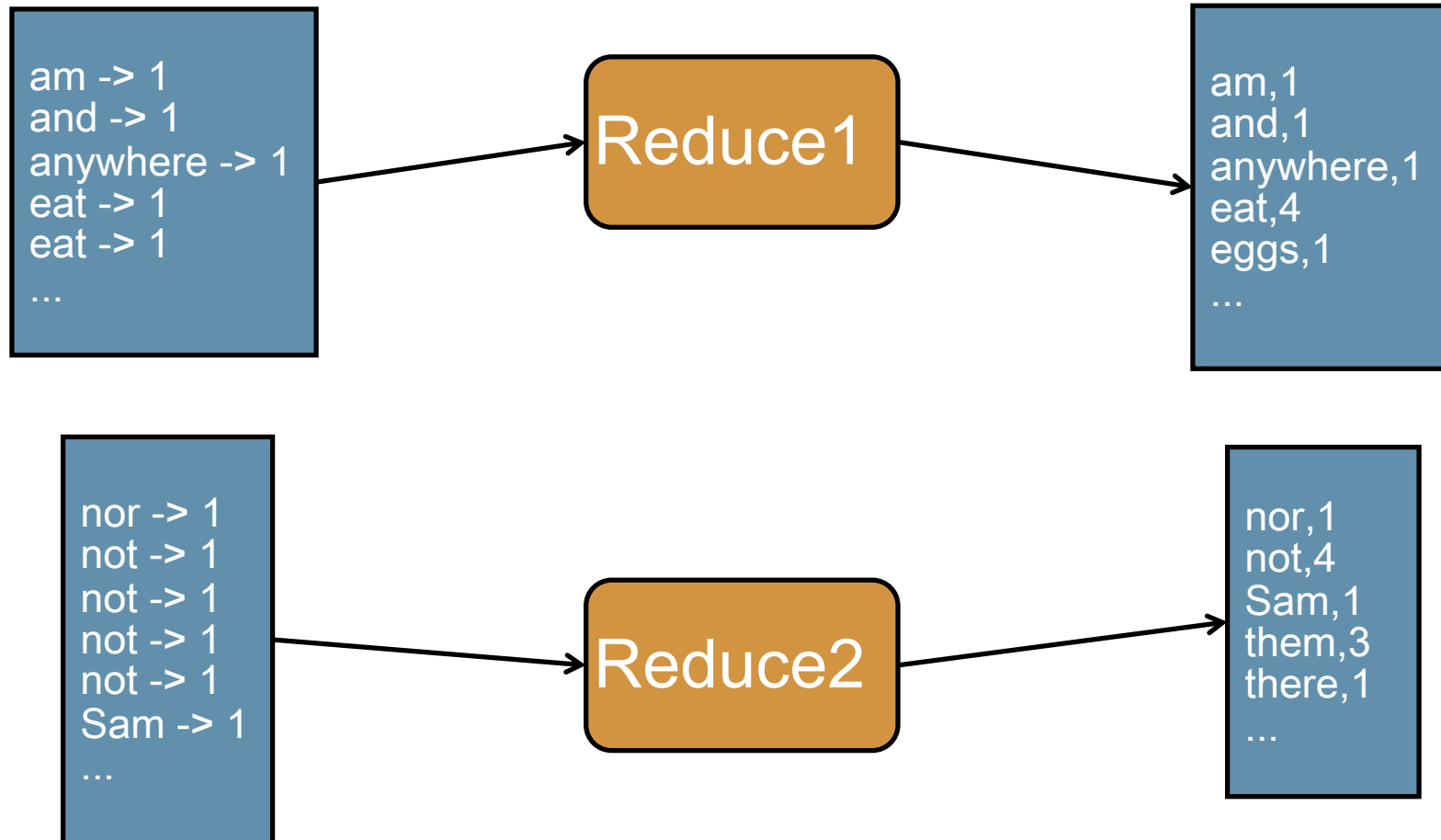
Word Count – Shuffle and Sort



Word Count – Pass to Reducers



Word Count – Perform Reduce



Hadoop Benefits

- > Mapper and Reducer are written by you
- > Hadoop provides
 - Parallelization
 - Shuffle and sort

Agenda

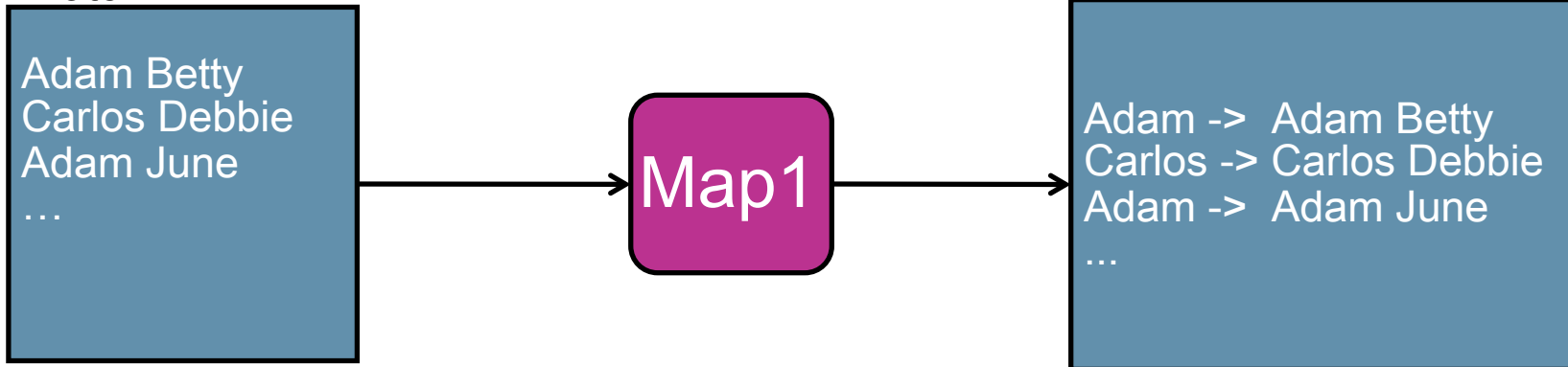
- > Background and Motivation
- > Hadoop Overview
- > [Hadoop at eHarmony](#)
- > Architecture
- > Tools and Performance
- > Roadblocks
- > Future Directions
- > Summary
- > Questions

Real World MapReduce

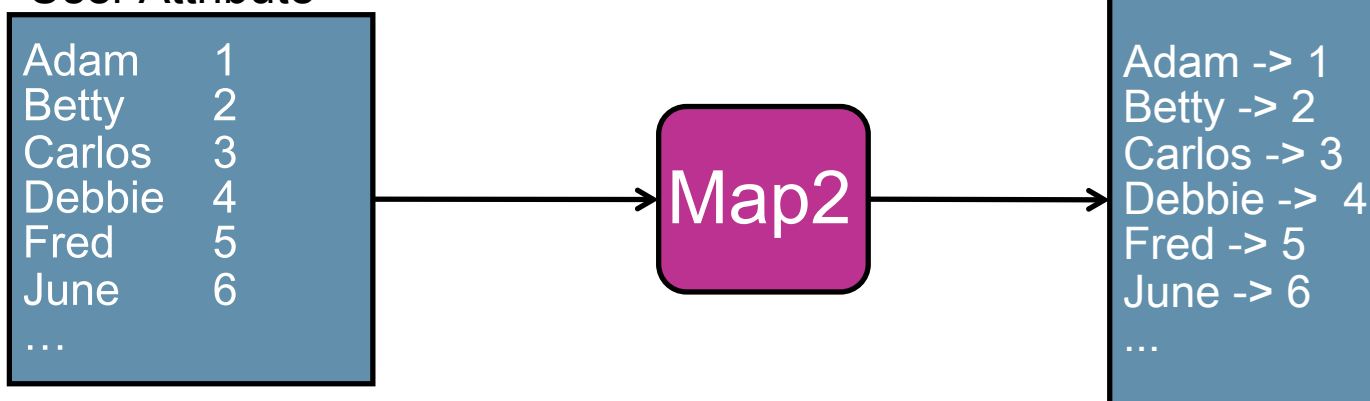
- > Complex applications use a series of MapReduces
- > Values from one step can become keys for another
- > Match Scoring Example
 - Join match data and user A attributes into one line
 - Join above with user B attributes and calculate the match score
 - Group by match scores by user

Join: Map

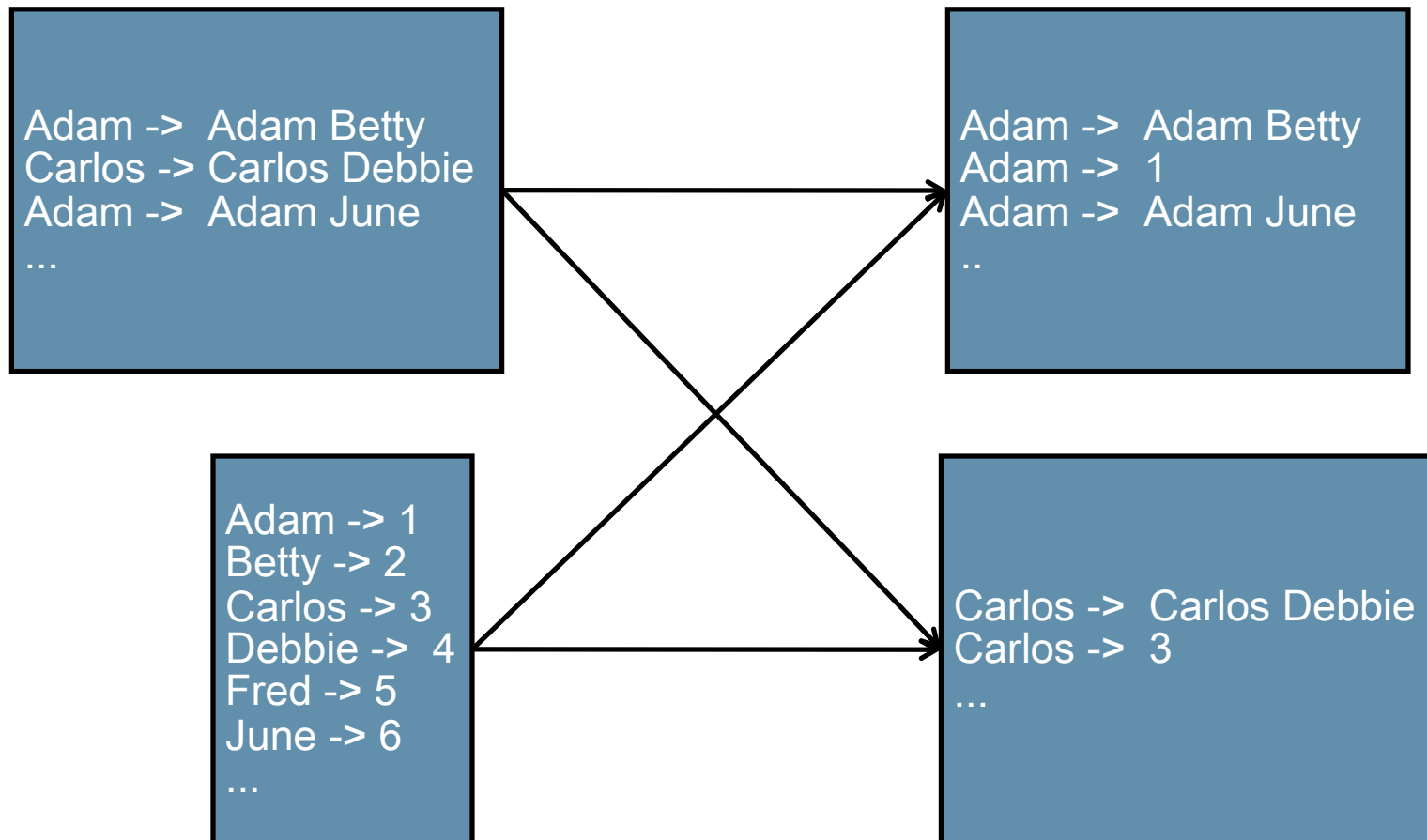
Match



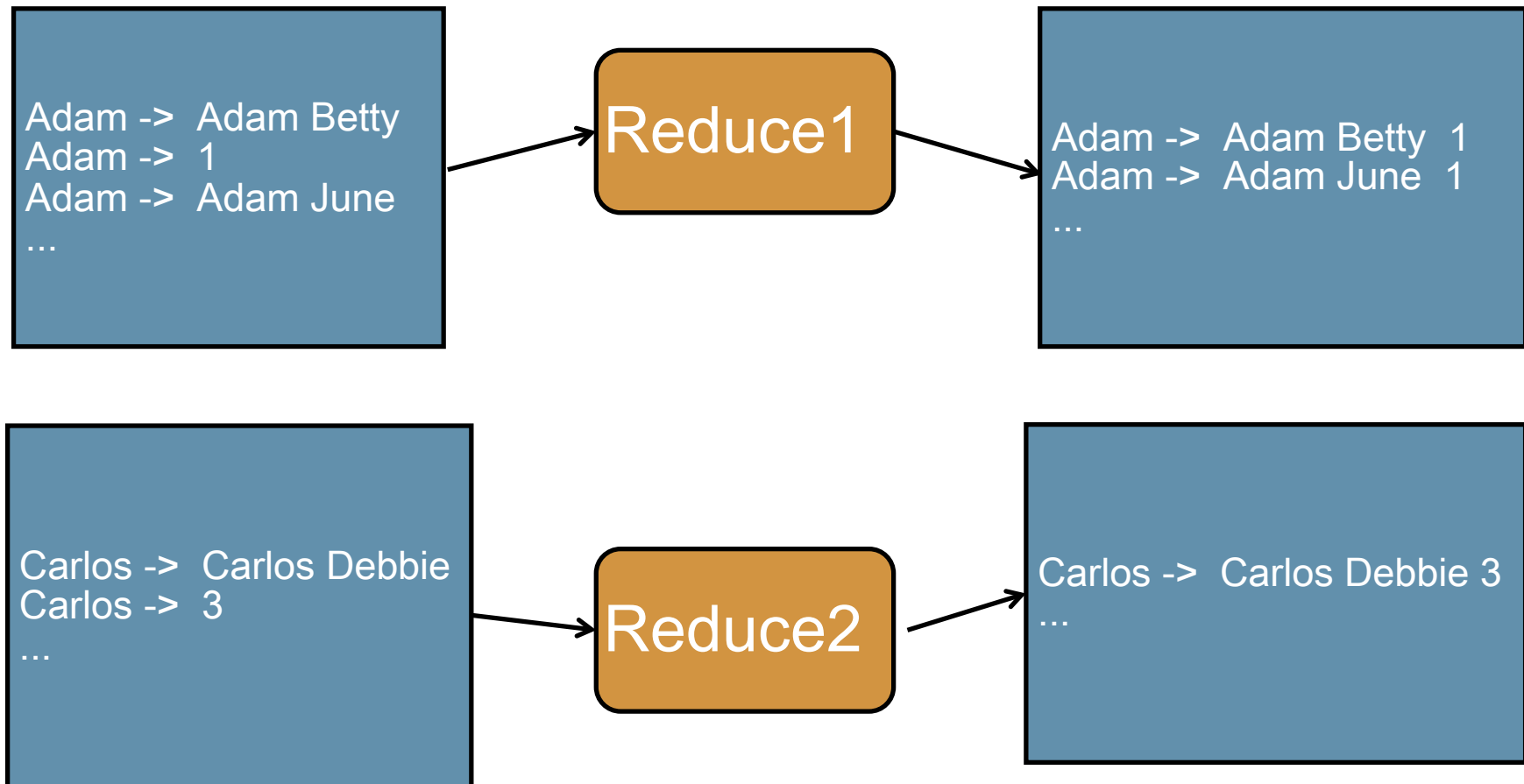
User Attribute



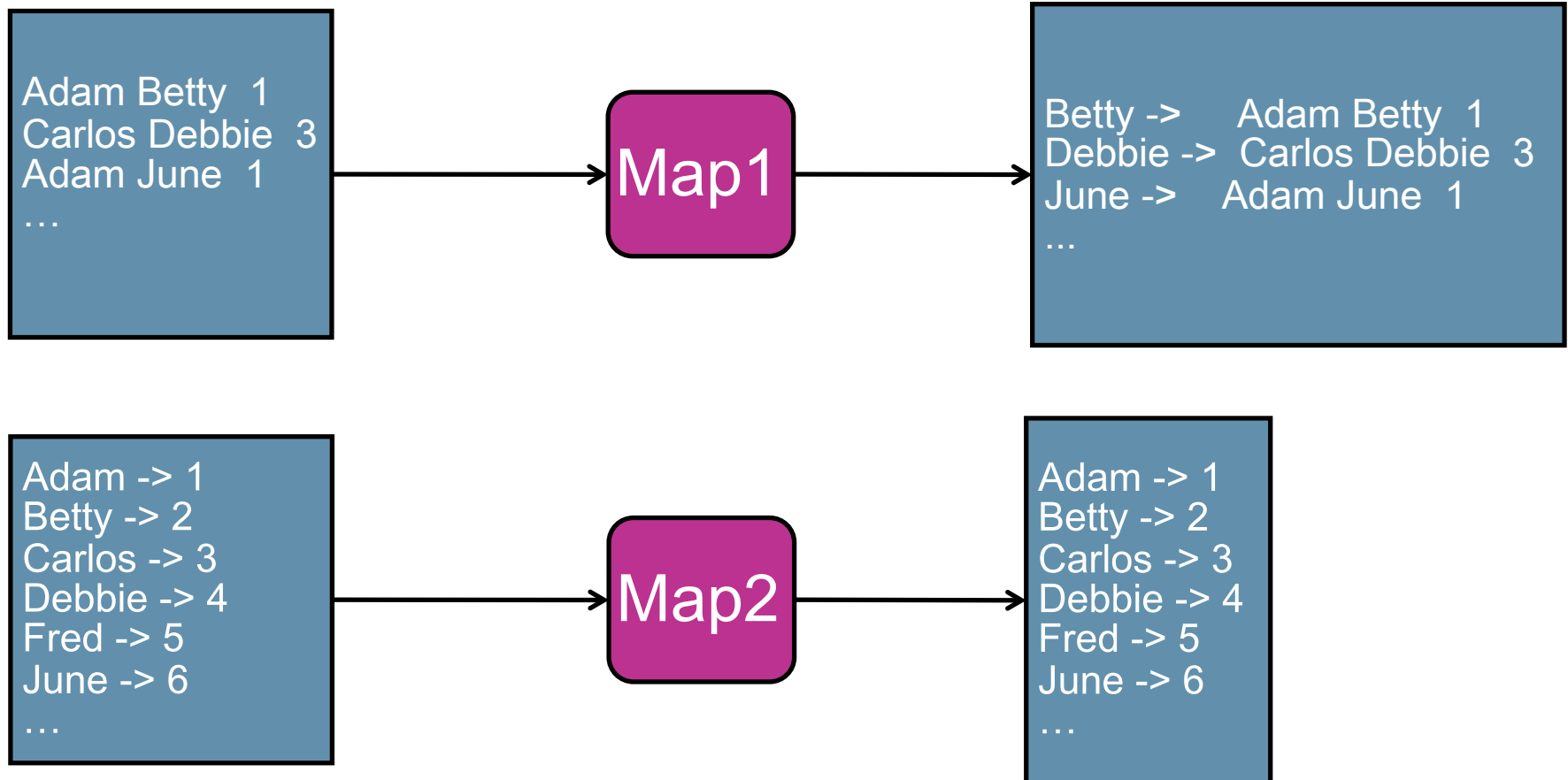
Join: Shuffle and Sort



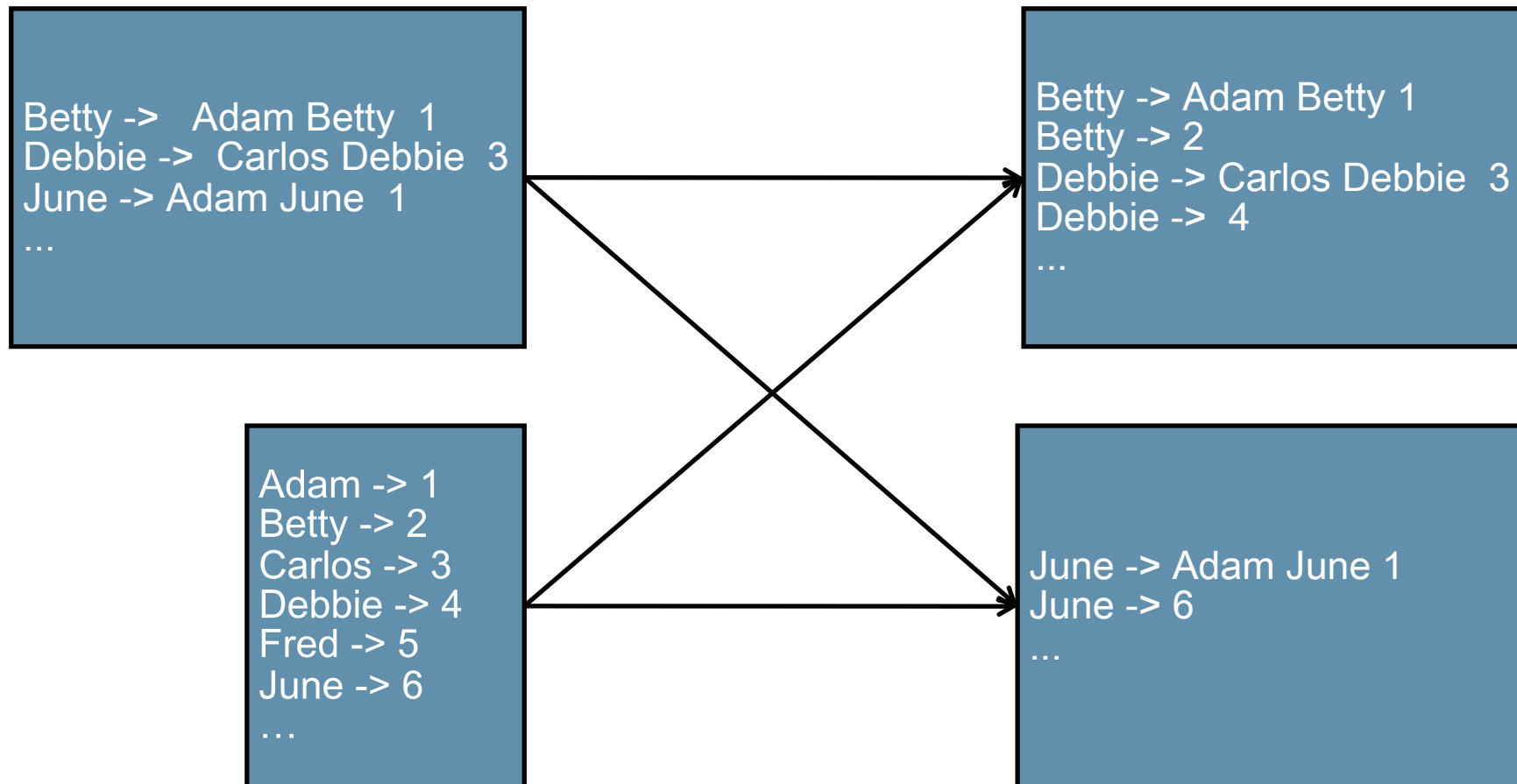
Join: Reduce



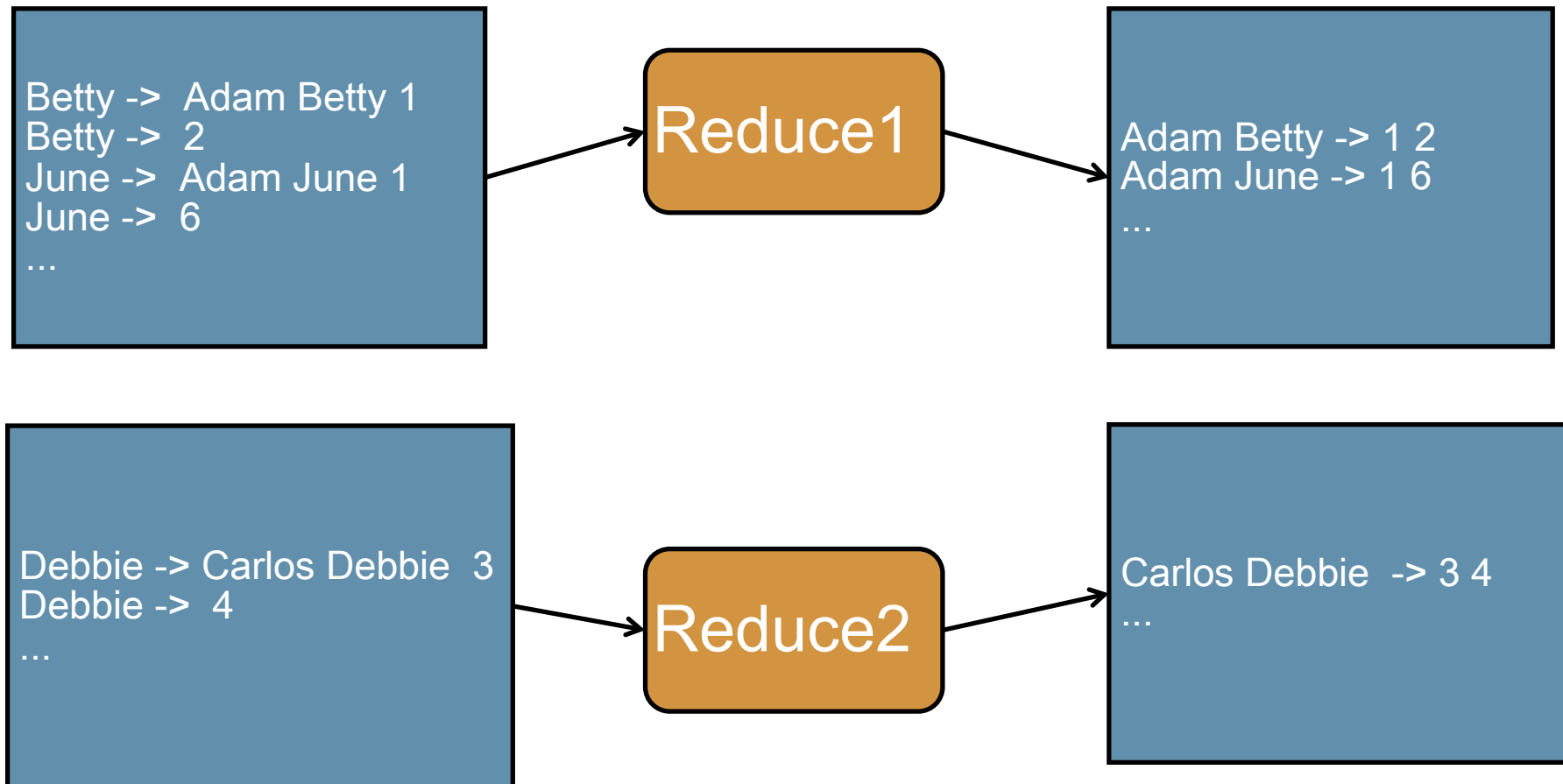
Join and Score: Map



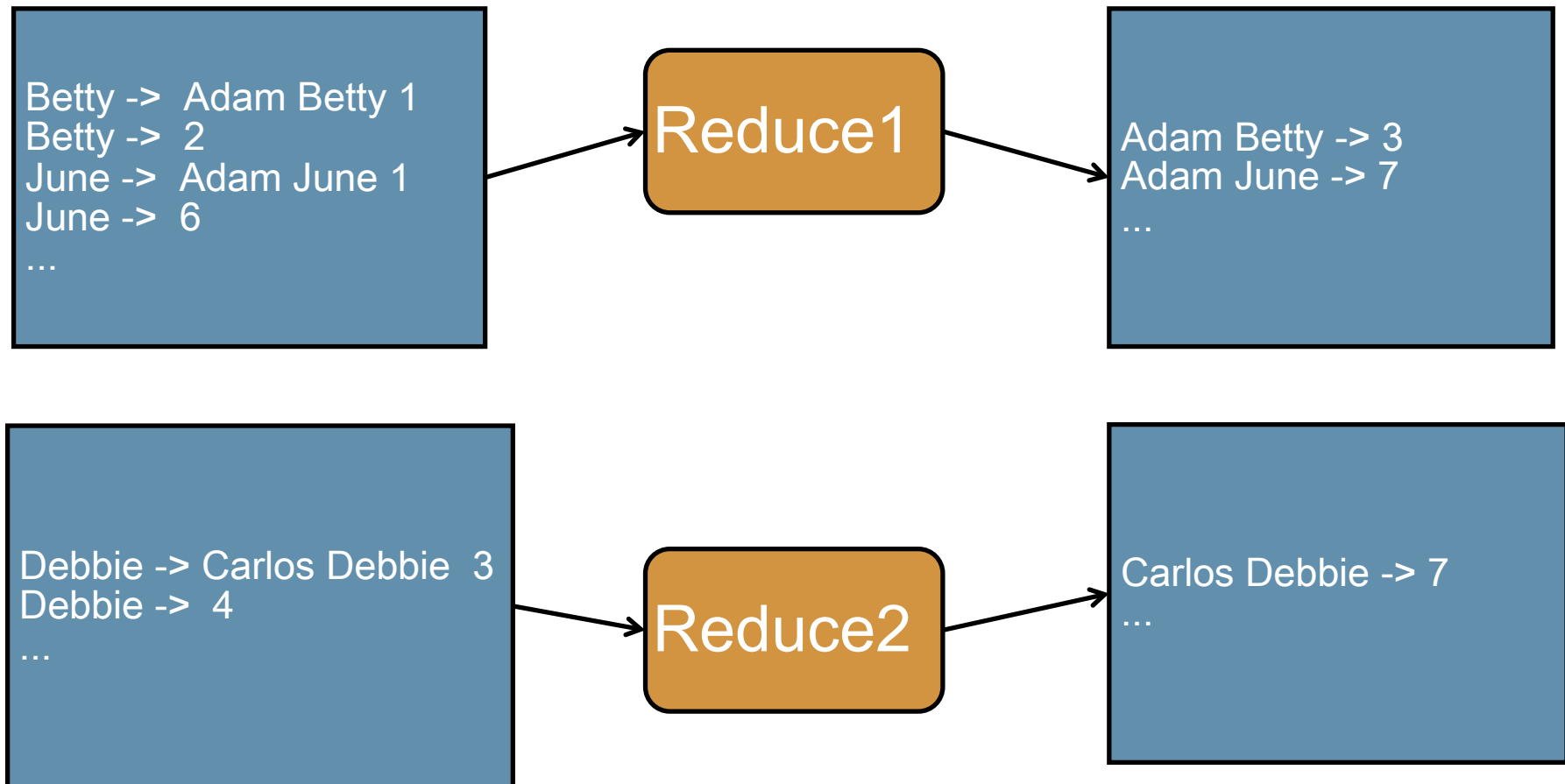
Join and Score: Shuffle and Sort



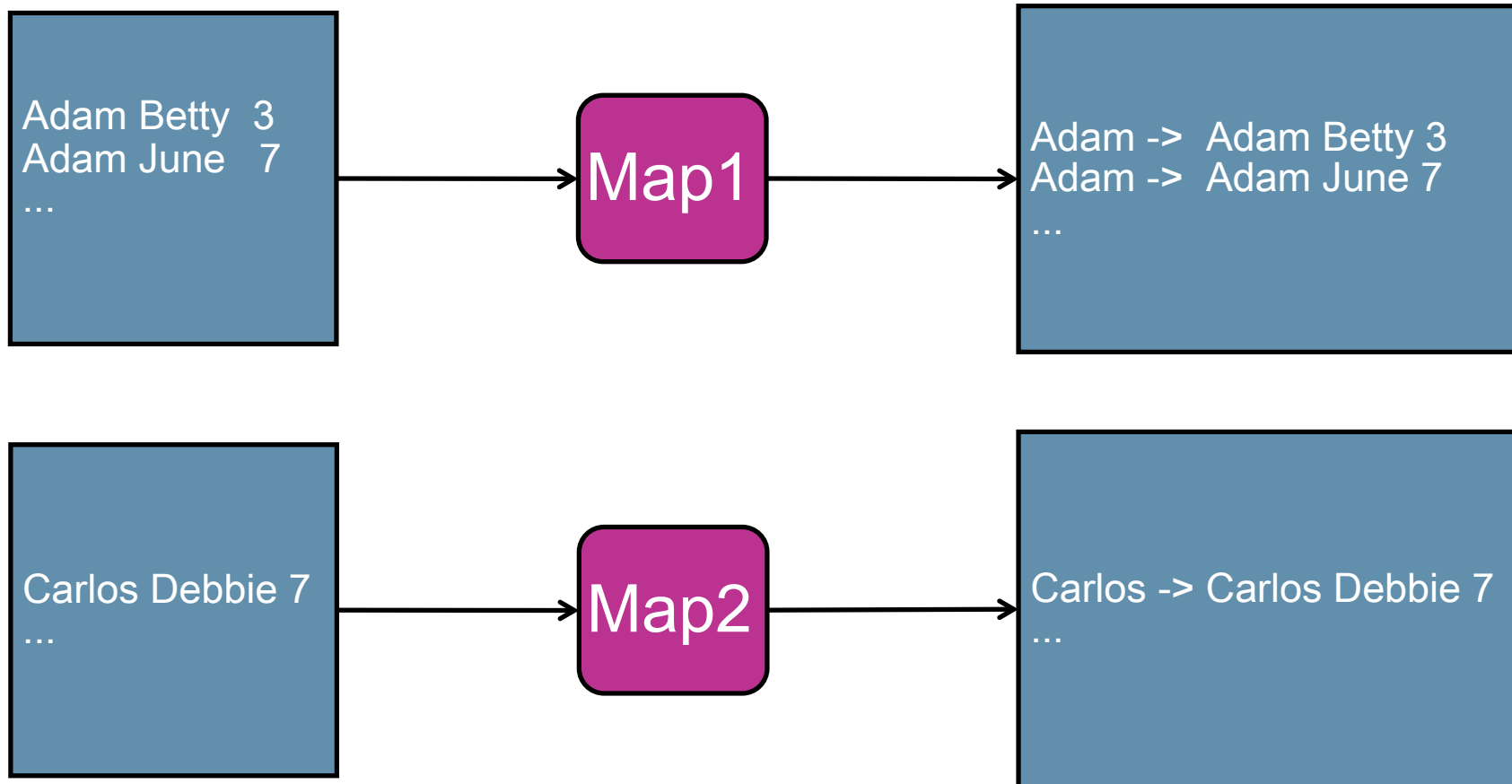
Join and Score: Reduce



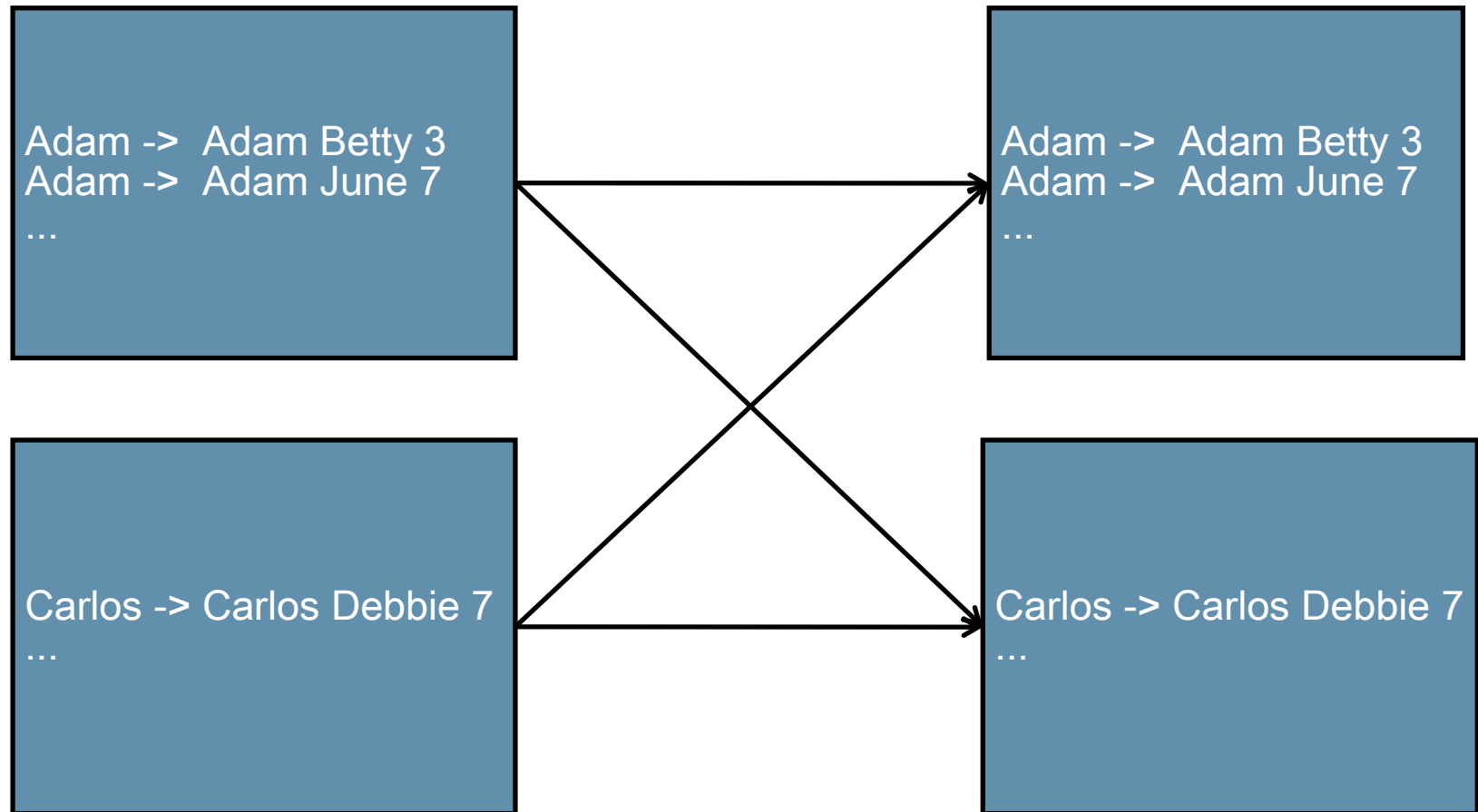
Join and Score: Reduce (Optimized)



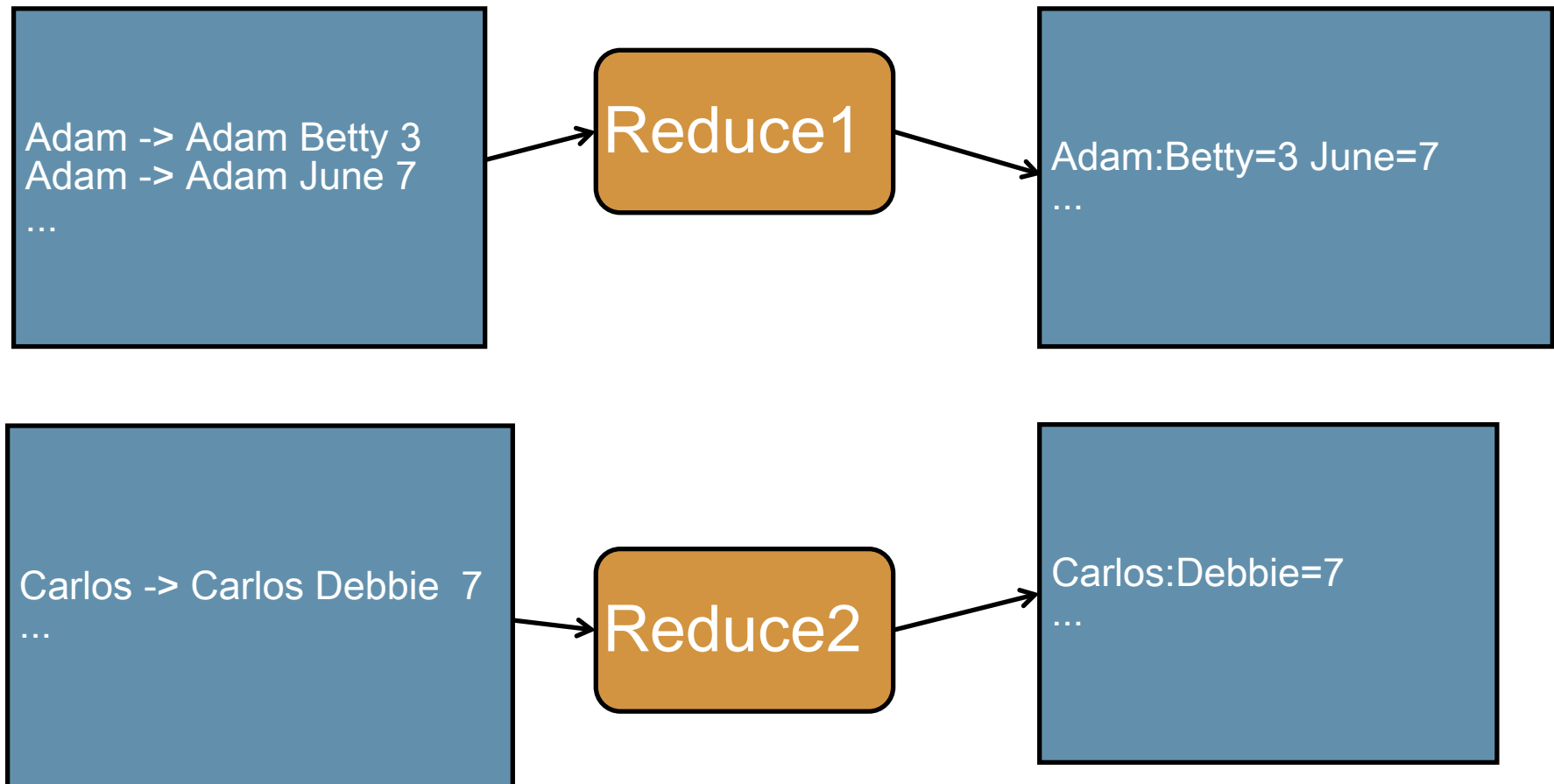
Combine: Map



Combine: Shuffle and Sort



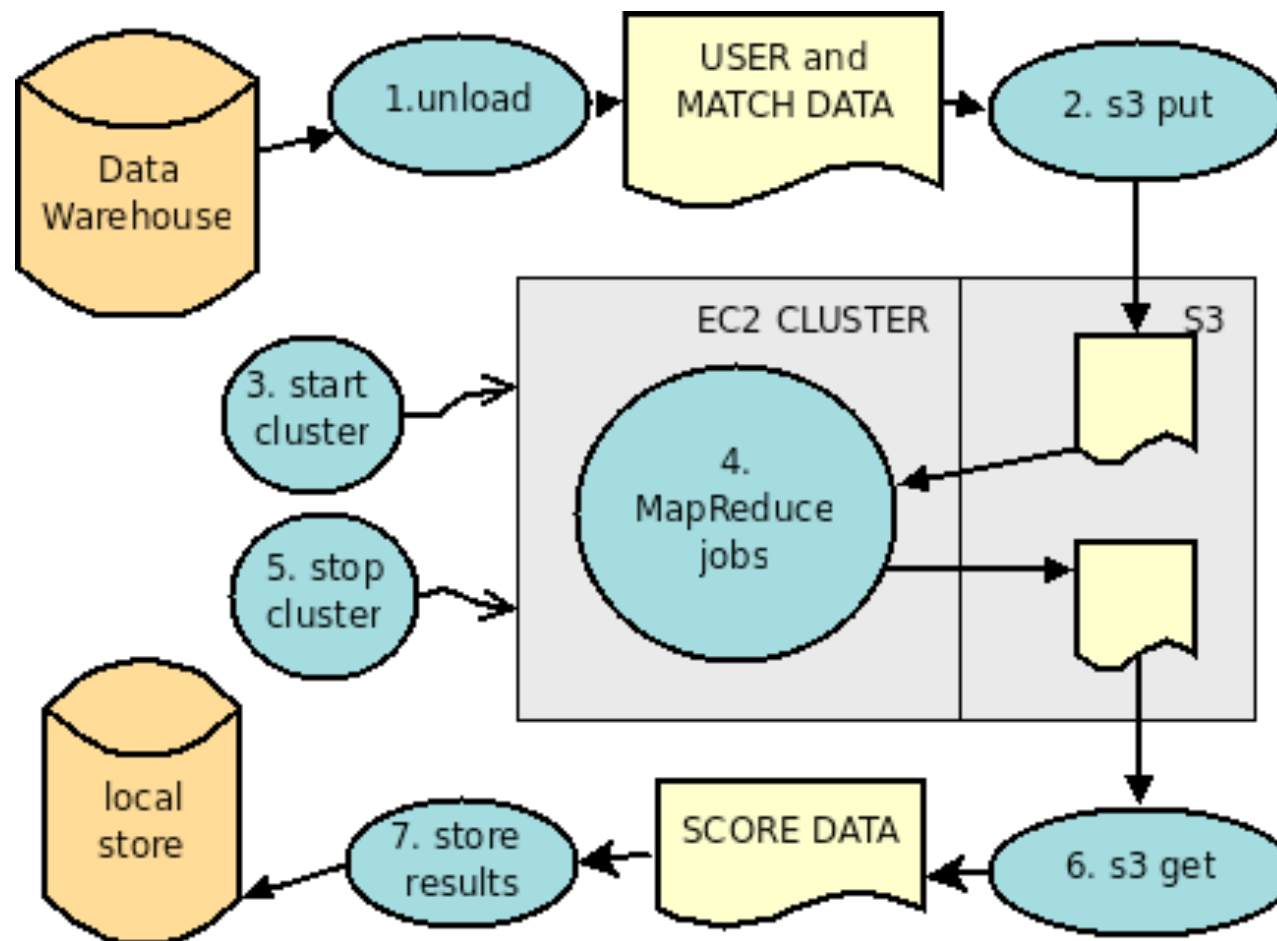
Combine: Reduce



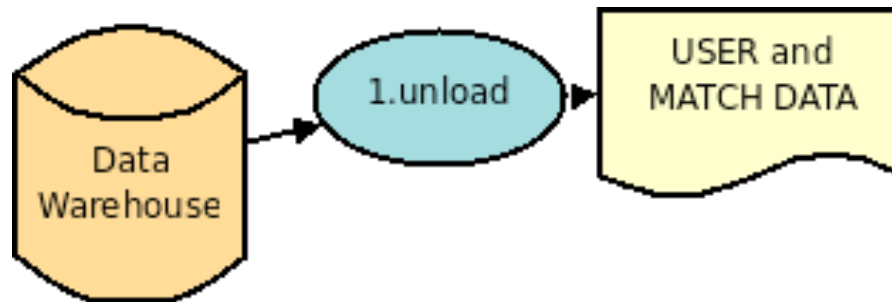
Agenda

- > Background and Motivation
- > Hadoop Overview
- > Hadoop at eHarmony
- > **Architecture**
- > Tools and Performance
- > Roadblocks
- > Future Directions
- > Summary
- > Questions

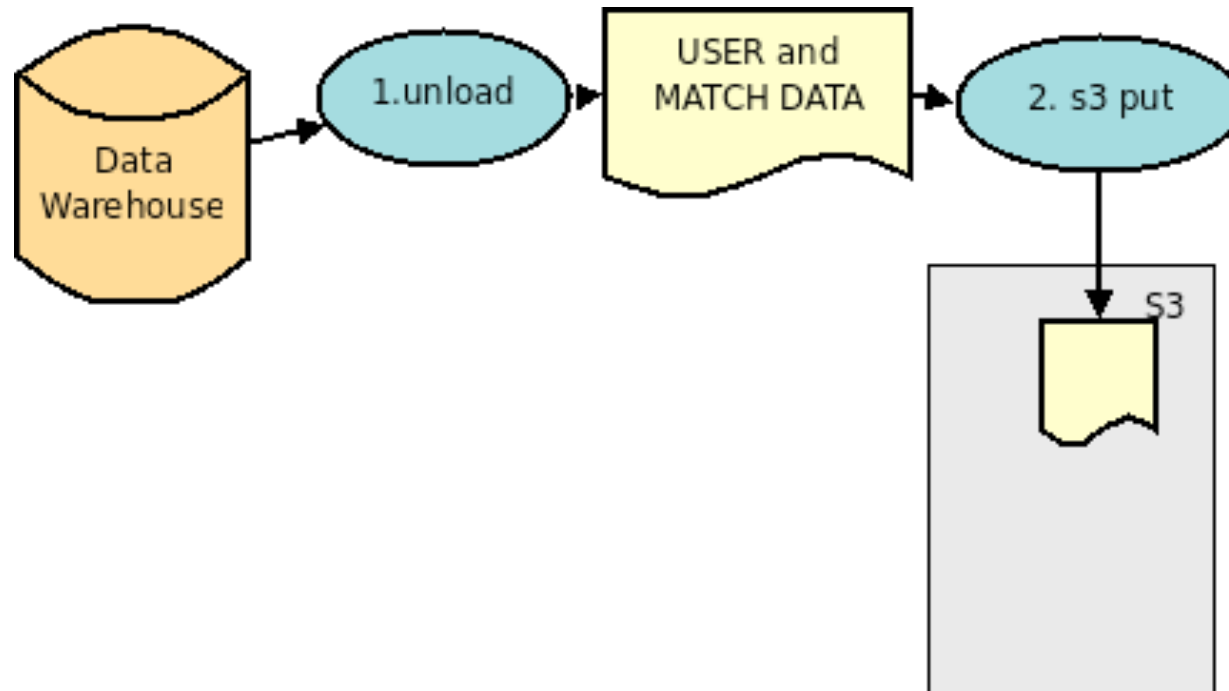
Overview



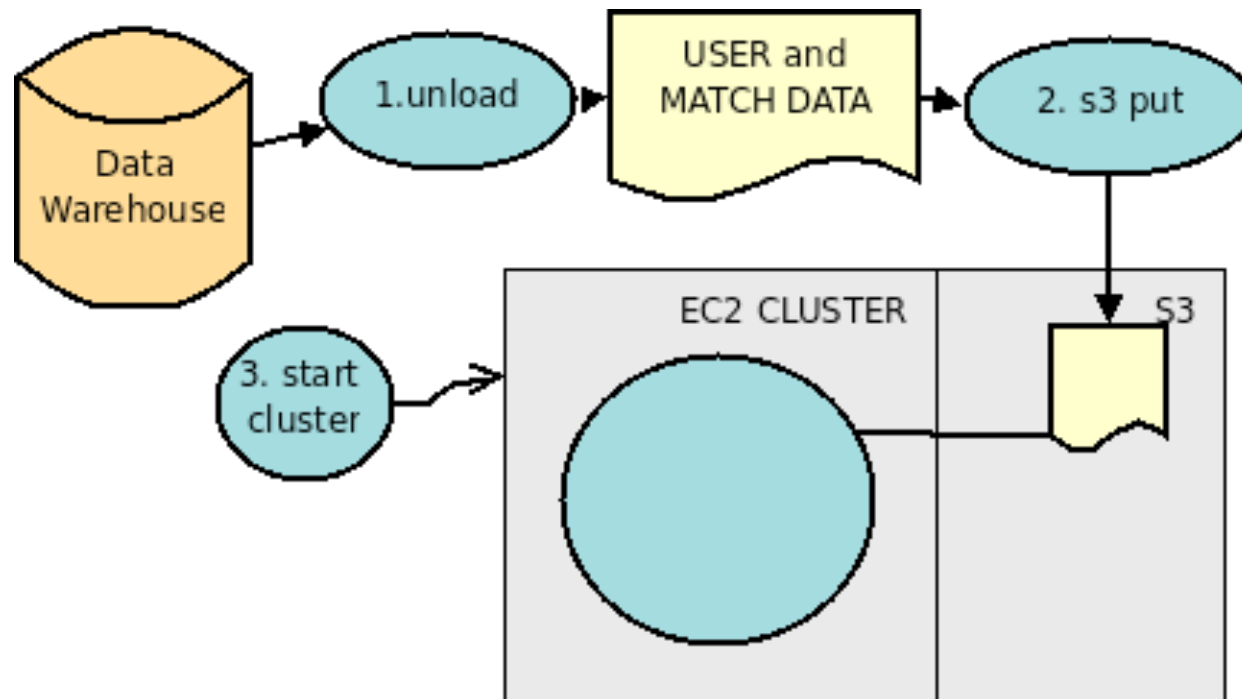
Hadoop Run – Data Warehouse Unload



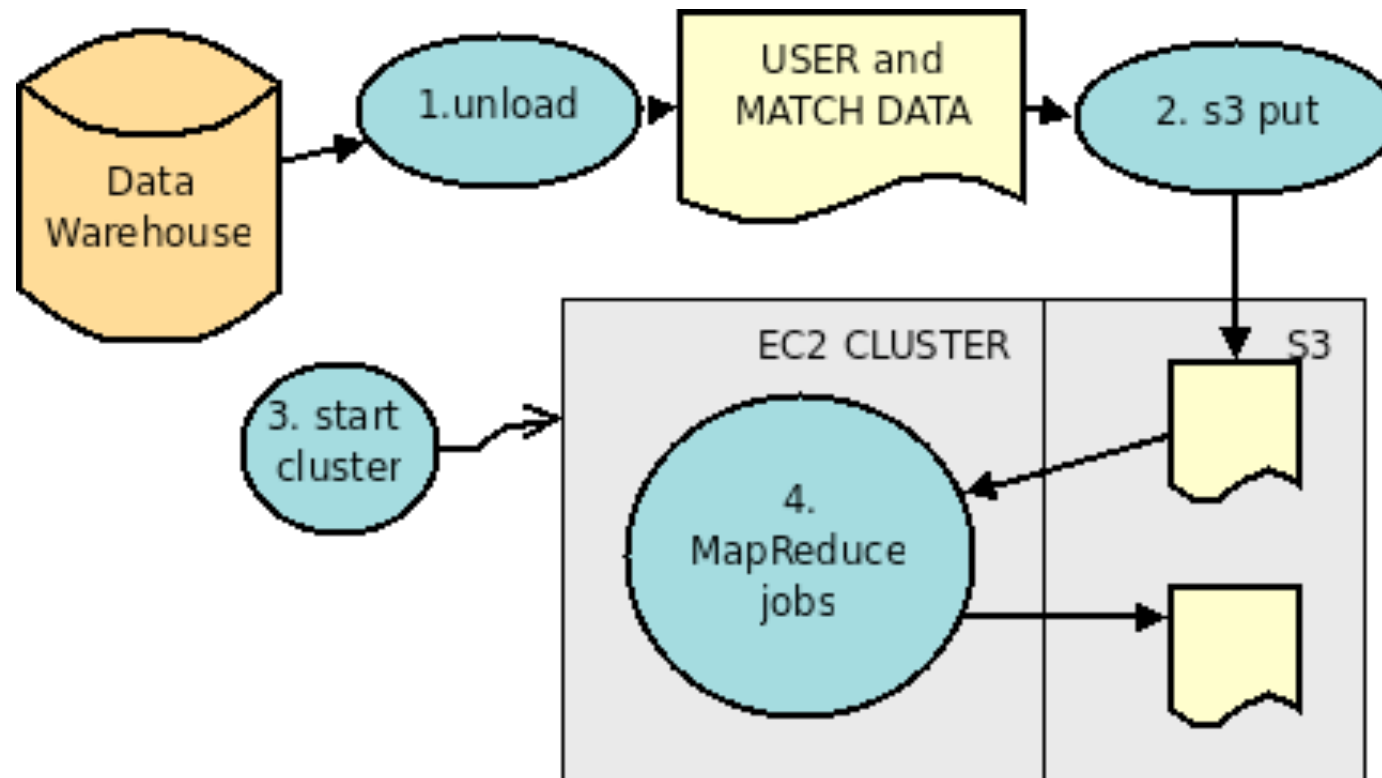
Hadoop Run – Upload to S3



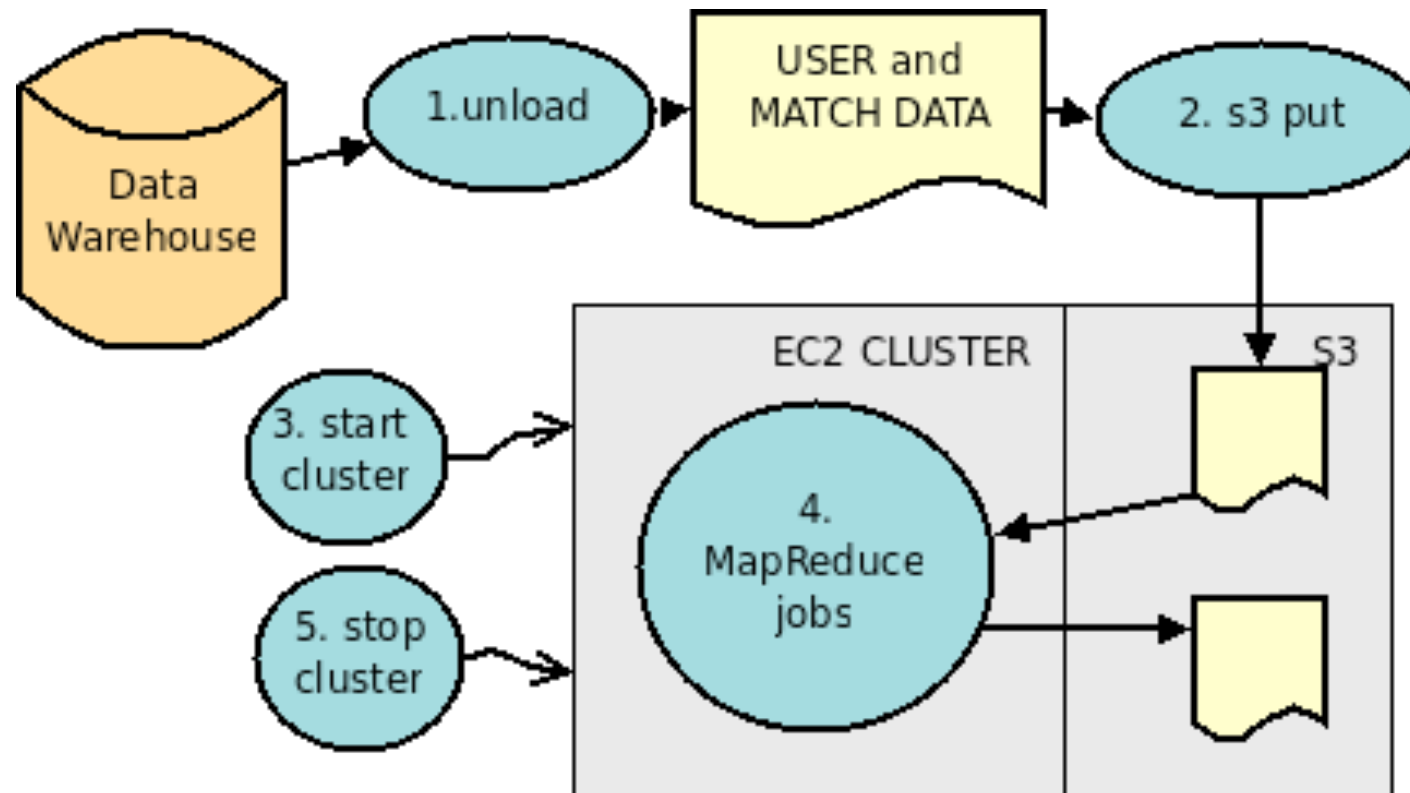
Hadoop Run – Start EC2 Cluster



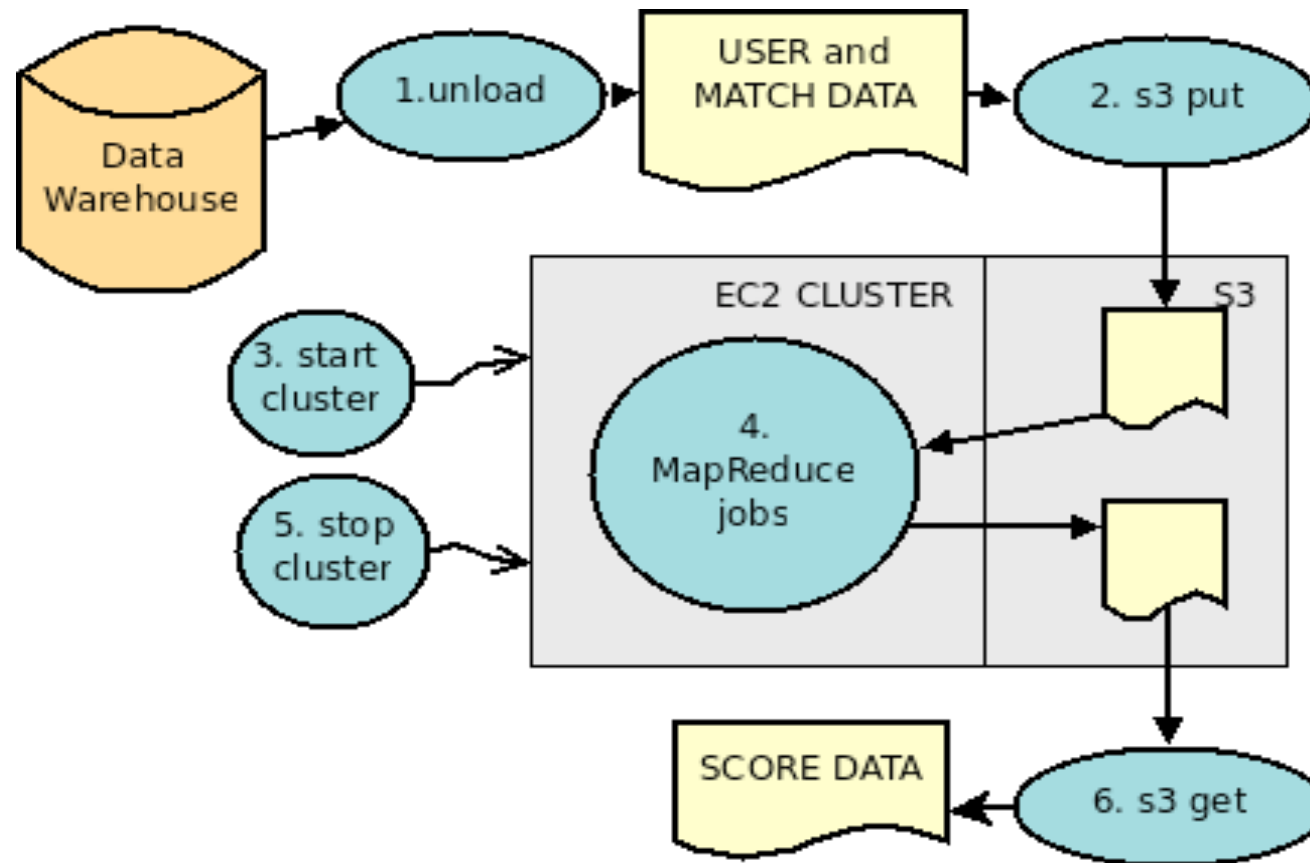
Hadoop Run – MapReduce Jobs



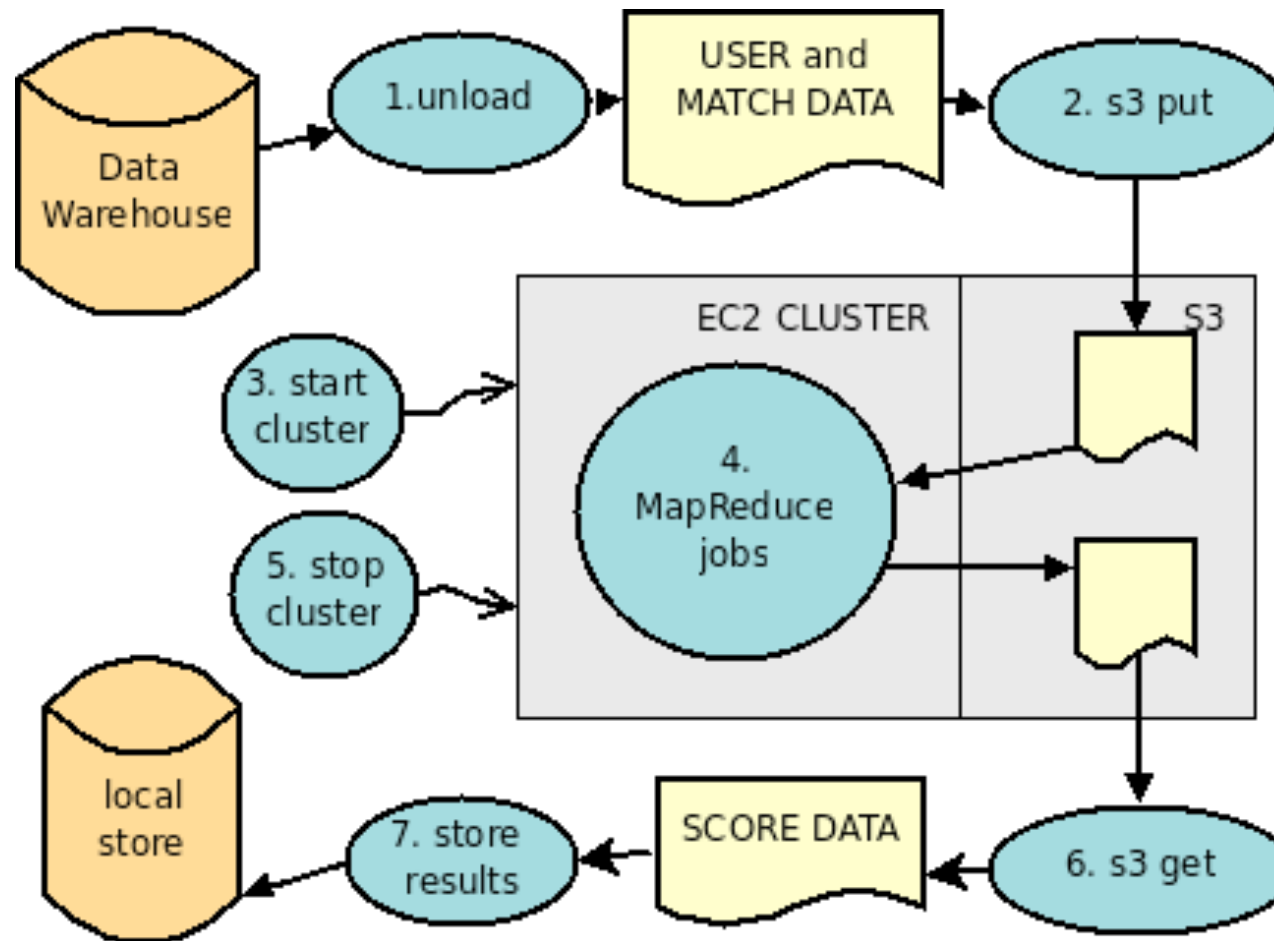
Hadoop Run – Stop EC2 Cluster



Hadoop Run – Download from S3



Hadoop Run – Store Result



AWS Elastic MapReduce

- > Only need to think in terms of Elastic MapReduce job flow
- > EC2 cluster is managed for you behind the scenes
- > Each job flow has one or more steps
- > Each step is a Hadoop MapReduce process
- > Each step can read and write data directly from and to S3
- > Based on Hadoop 0.18.3

Elastic MapReduce for eHarmony

- > Vastly simplify our scripts
 - No need to explicitly allocate, start and shutdown EC2 instances
 - Individual jobs were managed by a remote script running on master node (no longer required)
 - Jobs are arranged into a job flow, created with a single command
 - Status of a job flow and all its steps are accessible by a REST service

Simplified Job Control

- > Before Elastic Map Reduce, we had to explicitly:
 - Allocate cluster
 - Verify cluster
 - Push application to cluster
 - Run a control script on the master
 - Kick off each job step on the master
 - Create and detect a job completion token
 - Shut the cluster down
- > Over 150 lines of script just for this

Simplified Job Control

- > With Elastic MapReduce we can do all this with a single local command
- > Uses jar and conf files stored on S3

```
#{ELASTIC_MR_UTIL} --create --name #{JOB_NAME} --num-instances #{NUM_INSTANCES}  
--instance-type #{INSTANCE_TYPE} --key_pair #{KEY_PAIR_NAME}  
--log-uri #{SCORER_LOG_BUCKET_URL}  
--jar #{SCORER_JAR_S3_PATH} --main-class #{MR_JAVA_PACKAGE}.join.JoinJob --arg -  
xconf --arg #{MASTER_CONF_DIR}/join-config.xml  
--jar #{SCORER_JAR_S3_PATH} --main-class #{MR_JAVA_PACKAGE}.scorer.ScorerJob --  
arg -xconf --arg #{MASTER_CONF_DIR}/scorer-config.xml  
--jar #{SCORER_JAR_S3_PATH} --main-class  
#{MR_JAVA_PACKAGE}.combiner.CombinerJob --arg -xconf --arg #{MASTER_CONF_DIR}/  
combiner-config-#{TARGET_ENV}.xml
```

Agenda

- > Background and Motivation
- > Hadoop Overview
- > Hadoop at eHarmony
- > Architecture
- > Tools and Performance
- > Roadblocks
- > Future Directions
- > Summary
- > Questions

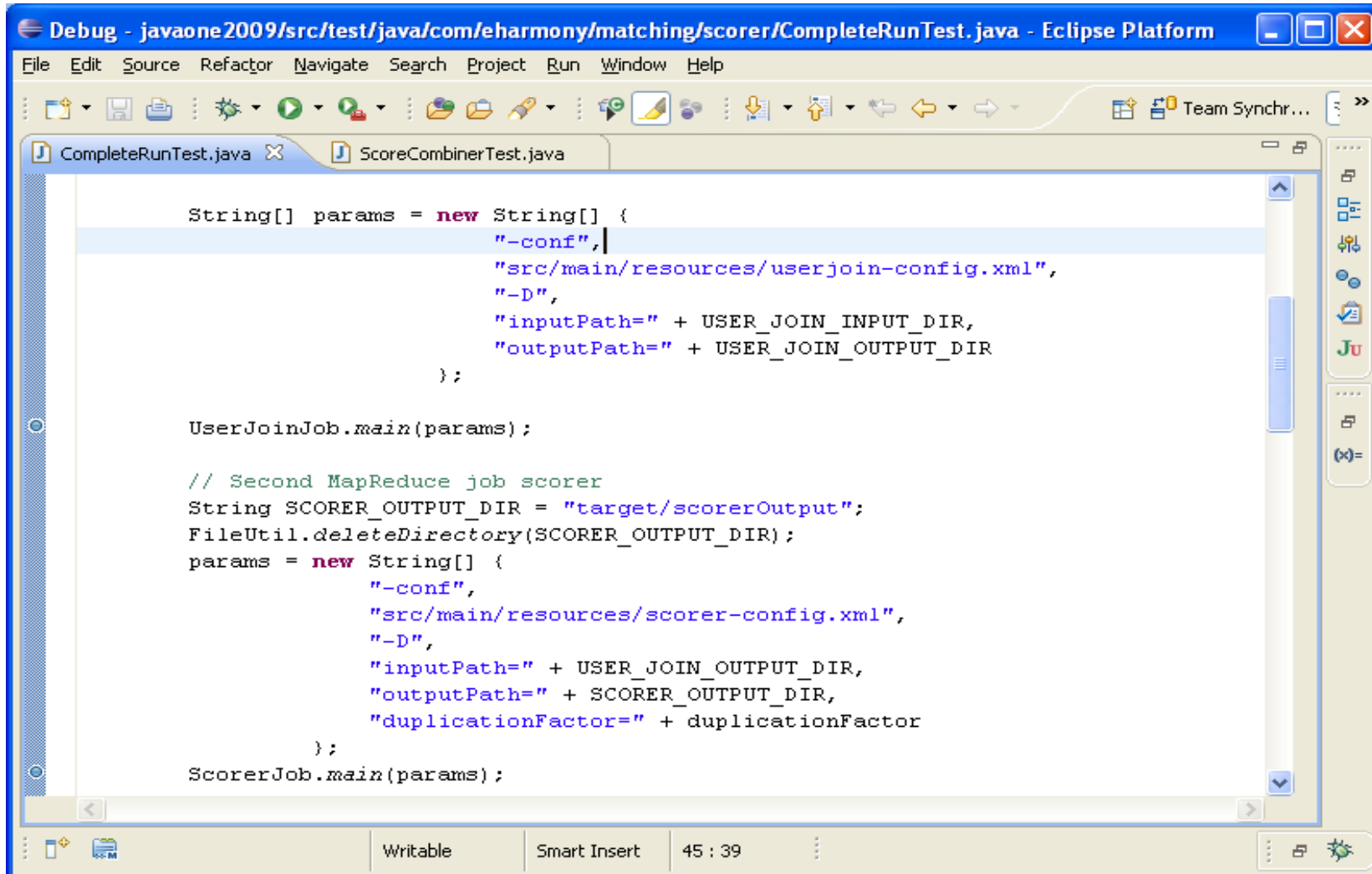
Development Environment

- > Cheap to set up on Amazon
- > Quick setup
 - Number of servers is controlled by a config variable
- > Identical to production
- > Separate development account recommended

Testing Demo

- > Test Hadoop jobs in IDE
- > A Hadoop cluster is not required

Unit Test Demo



```
Debug - javaone2009/src/test/java/com/eharmony/matching/scorer/CompleteRunTest.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
CompleteRunTest.java ScoreCombinerTest.java

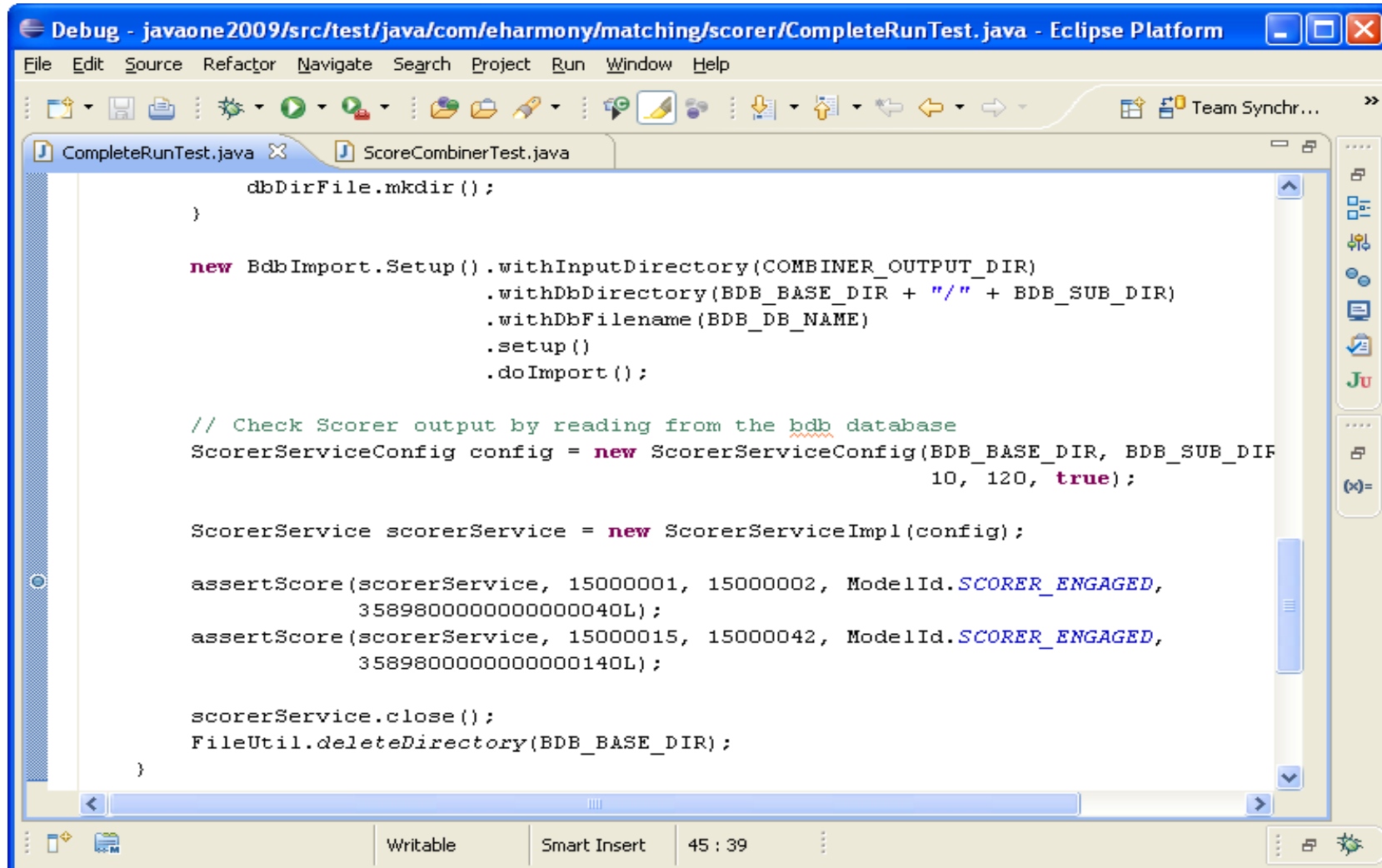
String[] params = new String[] {
    "-conf",
    "src/main/resources/userjoin-config.xml",
    "-D",
    "inputPath=" + USER_JOIN_INPUT_DIR,
    "outputPath=" + USER_JOIN_OUTPUT_DIR
};

UserJoinJob.main(params);

// Second MapReduce job scorer
String SCORER_OUTPUT_DIR = "target/scorerOutput";
FileUtil.deleteDirectory(SCORER_OUTPUT_DIR);
params = new String[] {
    "-conf",
    "src/main/resources/scorer-config.xml",
    "-D",
    "inputPath=" + USER_JOIN_OUTPUT_DIR,
    "outputPath=" + SCORER_OUTPUT_DIR,
    "duplicationFactor=" + duplicationFactor
};

ScorerJob.main(params);
```

Unit Test Demo



```
Debug - javaone2009/src/test/java/com/eharmony/matching/scorer/CompleteRunTest.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help

CompleteRunTest.java x ScoreCombinerTest.java

    dbDirFile.mkdir();
}

new BdbImport.Setup().withInputDirectory(COMBINER_OUTPUT_DIR)
    .withDbDirectory(BDB_BASE_DIR + "/" + BDB_SUB_DIR)
    .withDbFilename(BDB_DB_NAME)
    .setup()
    .doImport();

// Check Scorer output by reading from the bdb database
ScorerServiceConfig config = new ScorerServiceConfig(BDB_BASE_DIR, BDB_SUB_DIR,
    10, 120, true);

ScorerService scorerService = new ScorerServiceImpl(config);

assertScore(scorerService, 15000001, 15000002, ModelId.SCORER_ENGAGED,
    3589800000000000040L);
assertScore(scorerService, 15000015, 15000042, ModelId.SCORER_ENGAGED,
    35898000000000000140L);

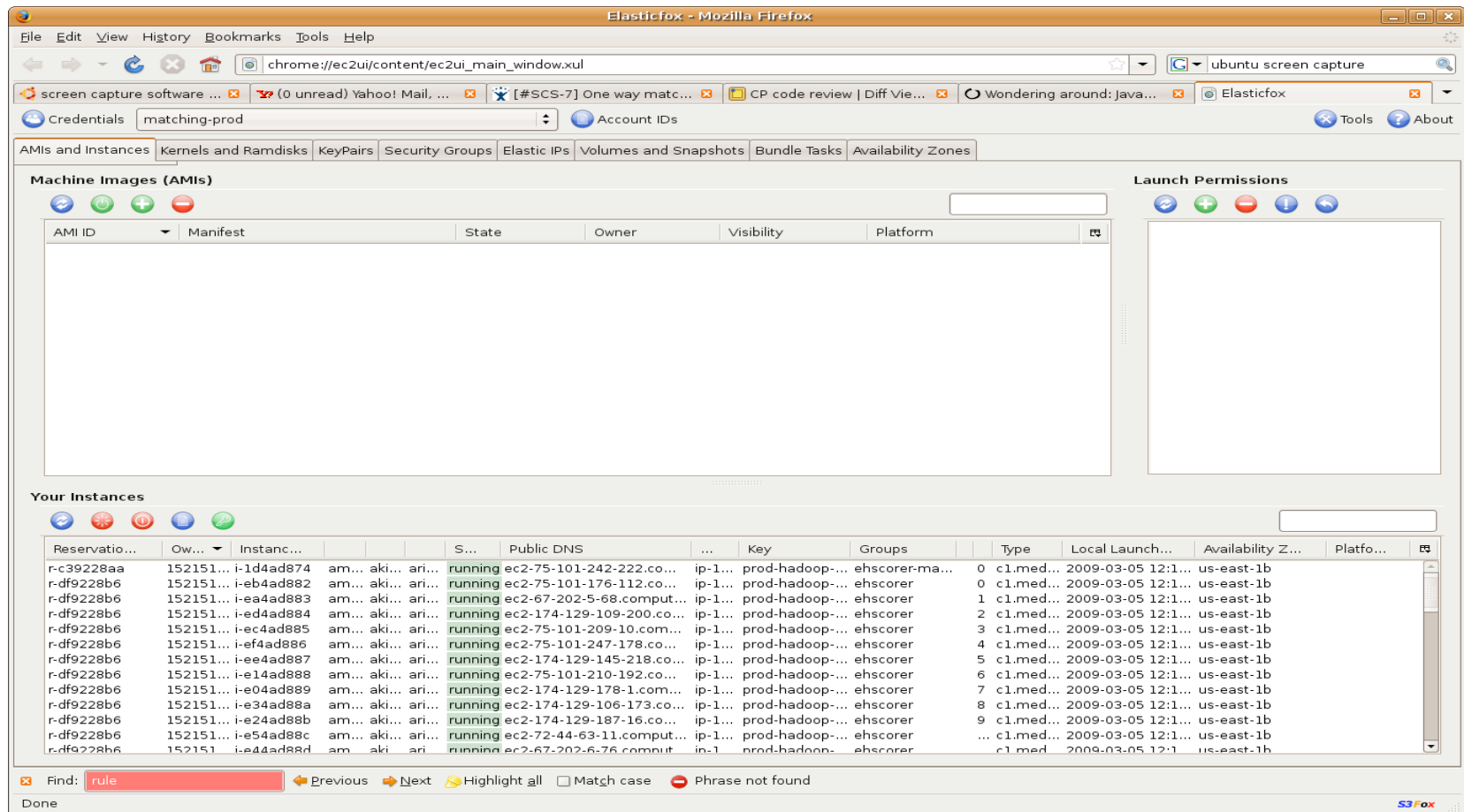
scorerService.close();
FileUtil.deleteDirectory(BDB_BASE_DIR);
}
```

Writable Smart Insert 45:39

EC2, S3 and Hadoop Monitoring Tools

- > ElasticFox for EC2
- > Hadoop provides web pages for job and disk monitoring.
- > Tim Kay's AWS command line tool for S3
- > Plus many more

ElasticFox – Manage EC2 Cluster



The screenshot shows the ElasticFox web interface in Mozilla Firefox. The browser address bar shows `chrome://ec2ui/content/ec2ui_main_window.xul`. The interface has several tabs: "Machine Images (AMIs)", "Launch Permissions", and "Your Instances". The "Your Instances" tab is active, displaying a table of EC2 instances.

Reservatio...	Own...	Instanc...		S...	Public DNS	...	Key	Groups		Type	Local Launch...	Availability Z...	Platfo...	
r-c39228aa	152151...	i-1d4ad874	am...	aki...	ari...	running	ec2-75-101-242-222.co...	ip-1...	prod-hadoop...	ehscorer-ma...	0	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-e4ad882	am...	aki...	ari...	running	ec2-75-101-176-112.co...	ip-1...	prod-hadoop...	ehscorer	0	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-ea4ad883	am...	aki...	ari...	running	ec2-67-202-5-68.comput...	ip-1...	prod-hadoop...	ehscorer	1	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-ed4ad884	am...	aki...	ari...	running	ec2-174-129-109-200.co...	ip-1...	prod-hadoop...	ehscorer	2	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-ec4ad885	am...	aki...	ari...	running	ec2-75-101-209-10.com...	ip-1...	prod-hadoop...	ehscorer	3	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-ef4ad886	am...	aki...	ari...	running	ec2-75-101-247-178.co...	ip-1...	prod-hadoop...	ehscorer	4	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-ee4ad887	am...	aki...	ari...	running	ec2-174-129-145-218.co...	ip-1...	prod-hadoop...	ehscorer	5	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-e14ad888	am...	aki...	ari...	running	ec2-75-101-210-192.co...	ip-1...	prod-hadoop...	ehscorer	6	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-e04ad889	am...	aki...	ari...	running	ec2-174-129-178-1.com...	ip-1...	prod-hadoop...	ehscorer	7	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-e34ad88a	am...	aki...	ari...	running	ec2-174-129-106-173.co...	ip-1...	prod-hadoop...	ehscorer	8	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-e24ad88b	am...	aki...	ari...	running	ec2-174-129-187-16.co...	ip-1...	prod-hadoop...	ehscorer	9	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-e54ad88c	am...	aki...	ari...	running	ec2-72-44-63-11.comput...	ip-1...	prod-hadoop...	ehscorer	...	cl.med...	2009-03-05 12:1...	us-east-1b
r-df9228b6	152151...	i-e4ad88d	am...	aki...	ari...	running	ec2-67-202-6-76.comput...	ip-1...	prod-hadoop...	ehscorer	...	cl.med...	2009-03-05 12:1...	us-east-1b

At the bottom of the interface, there is a search bar with the text "Find: rule" and buttons for "Previous", "Next", "Highlight all", "Match case", and "Phrase not found". The status bar at the bottom left says "Done" and the bottom right says "S3Fox".

Hadoop JobTracker – Monitor Jobs

ip-10-250-150-145 Hadoop Map/Reduce Administration - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://compute-1.amazonaws.com:50030/jobtracker.jsp

ip-10-250-150-1... Yahoo! Los Angeles Ti... (0 unread) Yaho... [#SCS-7] One ... CP code review ... Wondering arou... Elasticfox

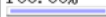
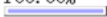

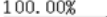




Cluster Summary

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node
0	0	4	49	98	98	4.00

Running Jobs

Running Jobs
none

Completed Jobs

Jobid	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed
job_200903051514_0001	root	distcp	100.00% 	19	19	100.00% 	0	0
job_200903051514_0002	root	Join	100.00% 	32	32	100.00% 	98	98
job_200903051514_0003	root	Scorer	100.00% 	5240	5240	100.00% 	98	98
job_200903051514_0004	root	Combiner	100.00% 	490	490	100.00% 	98	98

Failed Jobs

Failed Jobs
none

Find: rule Previous Next Highlight all Match case Phrase not found

Done

S3fox

Hadoop DFS – Monitor Disk Usage

Hadoop NameNode ip-10-250-150-145.ec2.internal:50001 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://compute-1.amazonaws.com:50070/dfshealth.jsp

ubuntu screen capture

Hadoop NameN... Yahoo! Los Angeles Ti... (0 unread) Yaho... [#SCS-7] One ... CP code review ... Wondering arou... Elasticfox


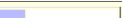
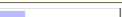


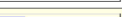










[Browse the filesystem](#)

Cluster Summary

373 files and directories, 5966 blocks = 6339 total. Heap Size is 11.45 MB / 988.94 MB (1%)

Capacity : 16.01 TB
 DFS Remaining : 13.83 TB
 DFS Used : 1.08 TB
 DFS Used% : 6.71 %
[Live Nodes](#) : 49
[Dead Nodes](#) : 0

Live Datanodes : 49

Node	Last Contact	Admin State	Size (GB)	Used (%)	Used (%)	Remaining (GB)	Blocks
ip-10-250-106-208	0	In Service	334.64	6.65		289.22	365
ip-10-250-107-242	0	In Service	334.64	5.85		291.85	317
ip-10-250-111-224	2	In Service	334.64	6.47		289.84	350
ip-10-250-114-48	0	In Service	334.64	6.69		289.12	364
ip-10-250-119-176	0	In Service	334.64	6.59		289.42	356
ip-10-250-123-175	0	In Service	334.64	6.79		288.78	375
ip-10-250-126-50	0	In Service	334.64	6.37		290.13	346
ip-10-250-147-32	0	In Service	334.64	7.1		287.75	387
ip-10-250-151-162	0	In Service	334.64	6.63		289.28	363
ip-10-250-155-240	2	In Service	334.64	7.09		287.75	384
ip-10-250-155-97	0	In Service	334.64	7.16		287.55	389
ip-10-250-158-241	2	In Service	334.64	6.68		289.14	363
ip-10-250-159-48	2	In Service	334.64	7.09		287.92	386
ip-10-250-163-191	2	In Service	334.64	6.51		289.67	352
ip-10-250-186-225	0	In Service	334.64	6.55		289.56	358
ip-10-250-191-81	0	In Service	334.64	6.7		289.15	362

Find: rule Previous Next Highlight all Match case Phrase not found

Done

S3Fox

AWS Management Console

- > Useful for Elastic MapReduce
 - Terminate job flow
 - Track execution
- > Also monitor EC2

AWS Management Console

Elastic MapReduce^{BETA}

AWS Management Console - Mozilla Firefox

https://console.aws.amazon.com/elasticmapreduce/home

Most Visited Getting Started Latest Headlines Share on Facebook Shorten with bit.ly http://maven2.ehar...

Amazon pass... Matching Am... Bike to Work ... [#MAT-127] ... Issue Navigat... [#MKR-10377] ... Advanced Ba... Revision 2990... Elasticfox AWS Manage...

Home > Your Account > AWS Management Console BETA

Welcome, [User Name] - [User Email] | Sign Out

Overview Amazon EC2 Amazon Elastic MapReduce

Your Elastic MapReduce Job Flows

Create New Job Flow Terminate Show/Hide Refresh Help

Viewing: All

Name	State	Creation Date	Elapsed Time	Normalized Instance Hours
scorer-prod	RUNNING	2009-05-07 14:41 PDT	2 hours 6 minutes	300
scorer-prod	TERMINATED	2009-05-07 11:39 PDT	0 hours 0 minutes	0

1 Job Flow selected

Id: j-29GIY8YBGJIN

Name: scorer-prod

State: RUNNING

Last State Change Reason: Running step

Availability Zone: us-east-1c

Master Type: c1.medium

Key Name: prod-elastic-mapreduce-key

Master Public DNS Name: ec2-174-129-186-10.compute-1.amazonaws.com

Creation Date: 2009-05-07 14:41 PDT

Start Date: 2009-05-07 14:46 PDT

End Date: -

Instance Count: 50

Slave Type: c1.medium

Log URI: s3n://eharmony-matching-prod-scanner-logs/

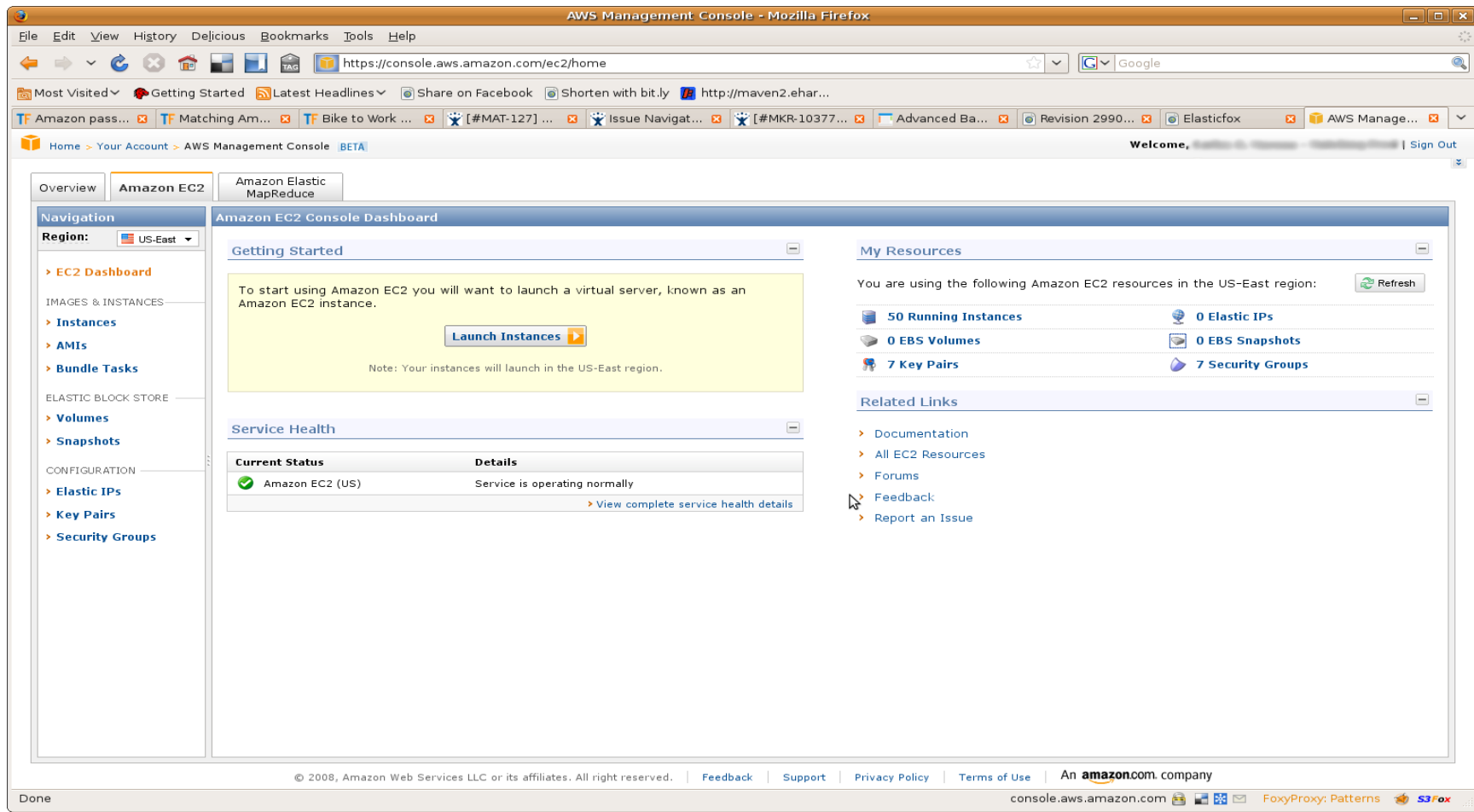
Step Name	State	Start Date	End Date	Jar	Main Class	Args
Example Jar Step	COMPLETED	2009-05-07 14:46 PDT	2009-05-07 15:34 PDT	s3n://eharmony-matching-prod-scanner-logs/scorer-1.5.jar	com.eharmony.matching.scorer.mapreduce.join.JoinJob	-xconf s3n://eharmony-matching-prod-scanner-logs/elastic-mapreduce-femalejoin-config-prod.xml
Example Jar Step	RUNNING	2009-05-07 15:34 PDT	-	s3n://eharmony-matching-prod-scanner-logs/scorer-1.5.jar	com.eharmony.matching.scorer.mapreduce.scorer.ScorerJob	-xconf s3n://eharmony-matching-prod-scanner-logs/elastic-mapreduce-scorer-config.xml
Example Jar Step	PENDING	-	-	s3n://eharmony-matching-prod-scanner-logs/scorer-1.5.jar	com.eharmony.matching.scorer.mapreduce.combiner.CombinerJob	-xconf s3n://eharmony-matching-prod-scanner-logs/elastic-mapreduce-combiner-config-prod.xml

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

Done console.aws.amazon.com FoxyProxy: Patterns S3Fox

AWS Management Console

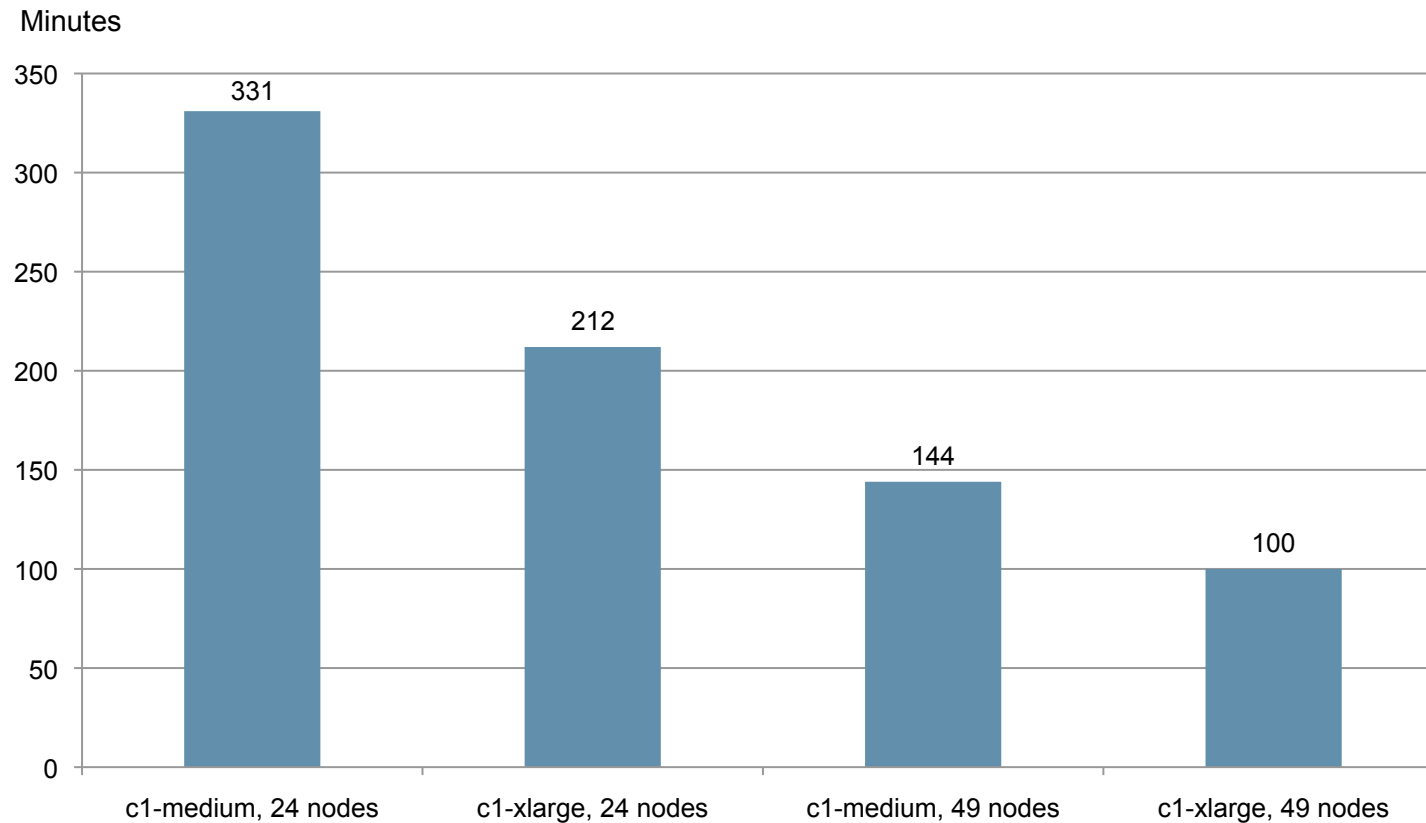
EC2



The screenshot shows the AWS Management Console for Amazon EC2 in the US-East region. The interface includes a navigation sidebar on the left with links to EC2 Dashboard, Instances, AMIs, Bundle Tasks, Volumes, Snapshots, Elastic IPs, Key Pairs, and Security Groups. The main content area is titled "Amazon EC2 Console Dashboard" and features a "Getting Started" section with a "Launch Instances" button. Below this is a "Service Health" section showing the current status of Amazon EC2 (US) as "Service is operating normally". On the right, the "My Resources" section displays a summary of resources: 50 Running Instances, 0 Elastic IPs, 0 EBS Volumes, 0 EBS Snapshots, 7 Key Pairs, and 7 Security Groups. A "Related Links" section at the bottom right provides links to Documentation, All EC2 Resources, Forums, Feedback, and Report an Issue. The footer of the console displays copyright information and links to Feedback, Support, Privacy Policy, and Terms of Use.

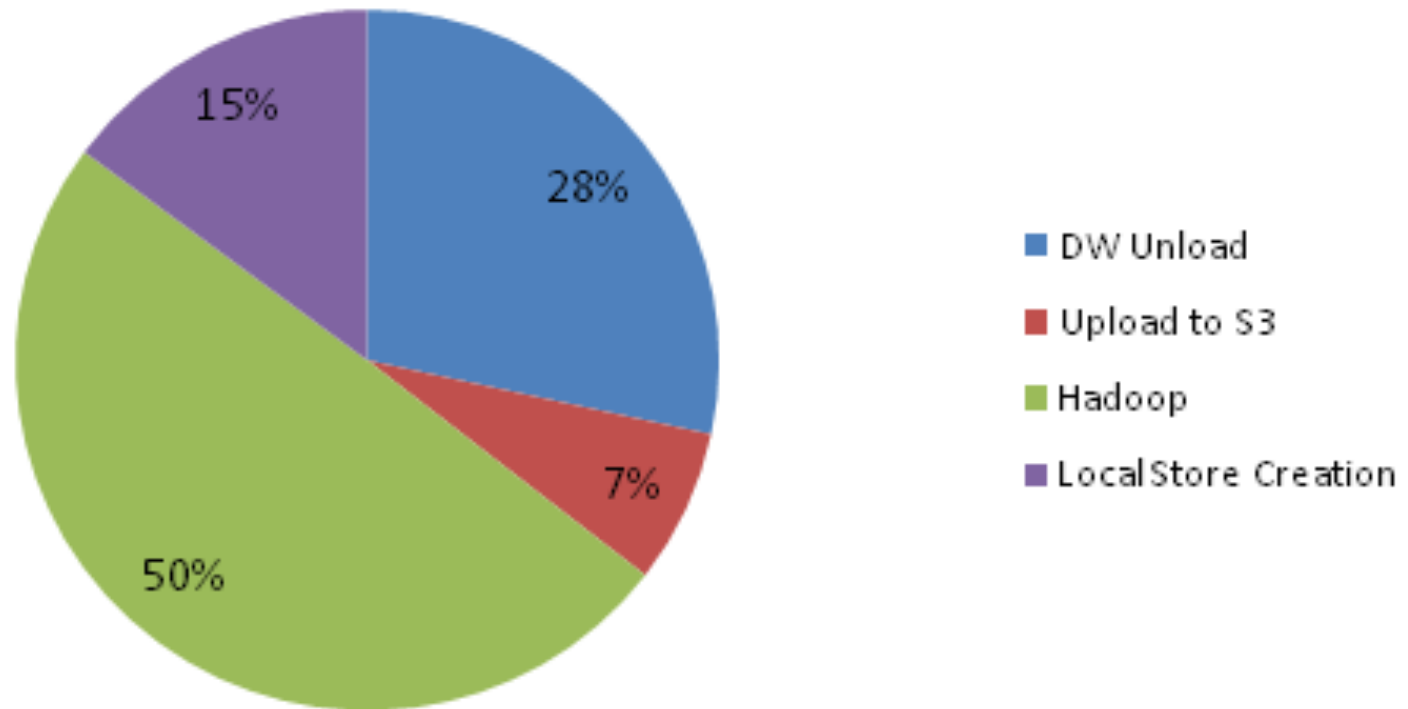
© 2008, Amazon Web Services LLC or its affiliates. All right reserved. | [Feedback](#) | [Support](#) | [Privacy Policy](#) | [Terms of Use](#) | An [amazon.com](#) company

Hadoop on EC2 Performance



Hadoop 0.18.0

Total Execution Time



Our Cost

- > Average EC2 and S3 Cost
 - Each run is 2 to 3 hours
 - \$1200/month for EC2
 - \$100/month for S3

Our Cost

- > Average EC2 and S3 Cost
 - Each run is 2 to 3 hours
 - \$1200/month for EC2
 - \$100/month for S3
- > Projected in-house cost
 - \$5000/month for a local cluster of 50 nodes running 24/7
 - A new company needs to add data center and operation personnel expense

Agenda

- > Background and Motivation
- > Hadoop Overview
- > Hadoop at eHarmony
- > Architecture
- > Tools and Performance
- > Roadblocks
- > Future Directions
- > Summary
- > Questions

Process Control is Non-Trivial

- > Scripts galore
- > One bash script for each stage
- > Moving to Ruby
 - Good exception handling and control structures
 - More productive

Design for Failure

- > The overall process depends on the success of each stage
- > Every stage is unreliable
- > Hadoop master node is a single point of failure; slave nodes are fault tolerant
- > Need to build retry logic to handle failures

Design for Failure

S3 Web Service

- > S3 web service can time out
- > Add logic to validate file is correctly uploaded/downloaded from S3
- > We retry once on failure

Design for Failure

EC2 Web Service

- > On very rare occasions, EC2 would allocate less slave nodes than requested – usually one less.
- > On even rarer occasions, no node is allocated.
- > Right now script fails and sends an alert email.
- > Consider changing the script to have it continue if the number of slave nodes missed is small
- > Or reallocate the missing nodes

Design for Failure

Elastic MapReduce^{BETA}

- > Provisioning of the servers is not yet stable
 - Frequently failed in the first two weeks of the beta program
 - It blocks until provisioning is complete
 - Only billed for job execution time
- > It's handy to terminate a hanged job with Amazon Web Services Management Console
- > Amazon recognizes this shortcoming and is working on a solution

Data Shuffle

- > Spend same amount of time getting data out of DW, upload to and download from S3, creating local store as running Hadoop
- > Hadoop and EC2 scale nicely
- > New scaling challenge is to reduce the data shuffle time and error recovery.

Agenda

- > Background and Motivation
- > Hadoop Overview
- > Hadoop at eHarmony
- > Architecture
- > Tools and Performance
- > Roadblocks
- > **Future Directions**
- > Summary
- > Questions

Handle More Modeling Data

- > Data warehouse unloads additional dimensions of user data into separate files for performance reason.
- > Currently user and match data is joined into one row which is not scalable.
- > Each model does not use all data.
 - Partial joins
 - Partial scores
 - Combine partial scores to final score
 - Load the delta of user data between runs

Data Analysis on the Cloud

- > Daily reporting: use Hadoop instead of depending on data warehouse.
 - Median/Mean score per user
 - Median/Mean number of matches per user, country, zipcode

Data Analysis on the Cloud with Pig

- > Our success opens Hadoop to other teams
- > Provides expertise on Hadoop
- > Data analysis with Pig
 - High-level language on top of Hadoop: think SQL for RDBMS
 - Hadoop subproject
 - Still not non-engineer friendly for troubleshooting
 - Slower than hand-rolled MapReduce
- > Yet to evaluate Hive, Cascade, others

Data Analysis with R

> Data analysis with R

- Excellent environment for statistical computing and graphic presentation
- Open source implementation of S
- Supports a plugin framework
- New packages are continued being developed by an active community
- Limitation is that all data must be in memory
- Start to investigate RHIPE which aims to integrate R and Hadoop

Agenda

- > Background and Motivation
- > Hadoop Overview
- > Hadoop at eHarmony
- > Architecture
- > Tools and Performance
- > Roadblocks
- > Future Directions
- > **Summary**
- > Questions

Lesson Learned

- > Reliability is the biggest challenge
- > Getting data out of DW is difficult and time consuming
- > Controlling process with shell scripts is a hassle
- > Dev tools really easy to work with and just work right out of the box
- > Standard Hadoop AML worked great
- > Easy to write unit tests for MapReduce
- > Hadoop community support is great.
- > EC2/S3/EMR are cost effective

Acknowledgement

- > 236 members marry a day is based on survey conducted by Harris Interactive Research.
- > Hadoop is an Apache project
- > S3 and EC2 are part of Amazon Web Service.
- > R's URL is <http://www.r-project.org/>
- > RHIPE's URL is <http://ml.stat.purdue.edu/rhipe>

Agenda

- > Background and Motivation
- > Hadoop Overview
- > Hadoop at eHarmony
- > Architecture
- > Tools and Performance
- > Roadblocks
- > Future Directions
- > Summary
- > Questions



JavaOneSM

Thank You

Steve Kuo, Software Architect
Joshua Tuberville, Software Architect
eHarmony

