cloudera
Ask Bigger Questions

Yarns about YARN: Migrating to MapReduce v2

Kathleen Ting | @kate_ting

Technical Account Manager, Cloudera | Sqoop PMC Member

Big Data Camp LA

June 14, 2014

Who Am I?

- Started 3 yr ago as 1st Cloudera Support Eng
- Now manages Cloudera's 2 largest customers

- Sqoop Committer, PMC Member
- Co-Author of the Apache Sqoop Cookbook

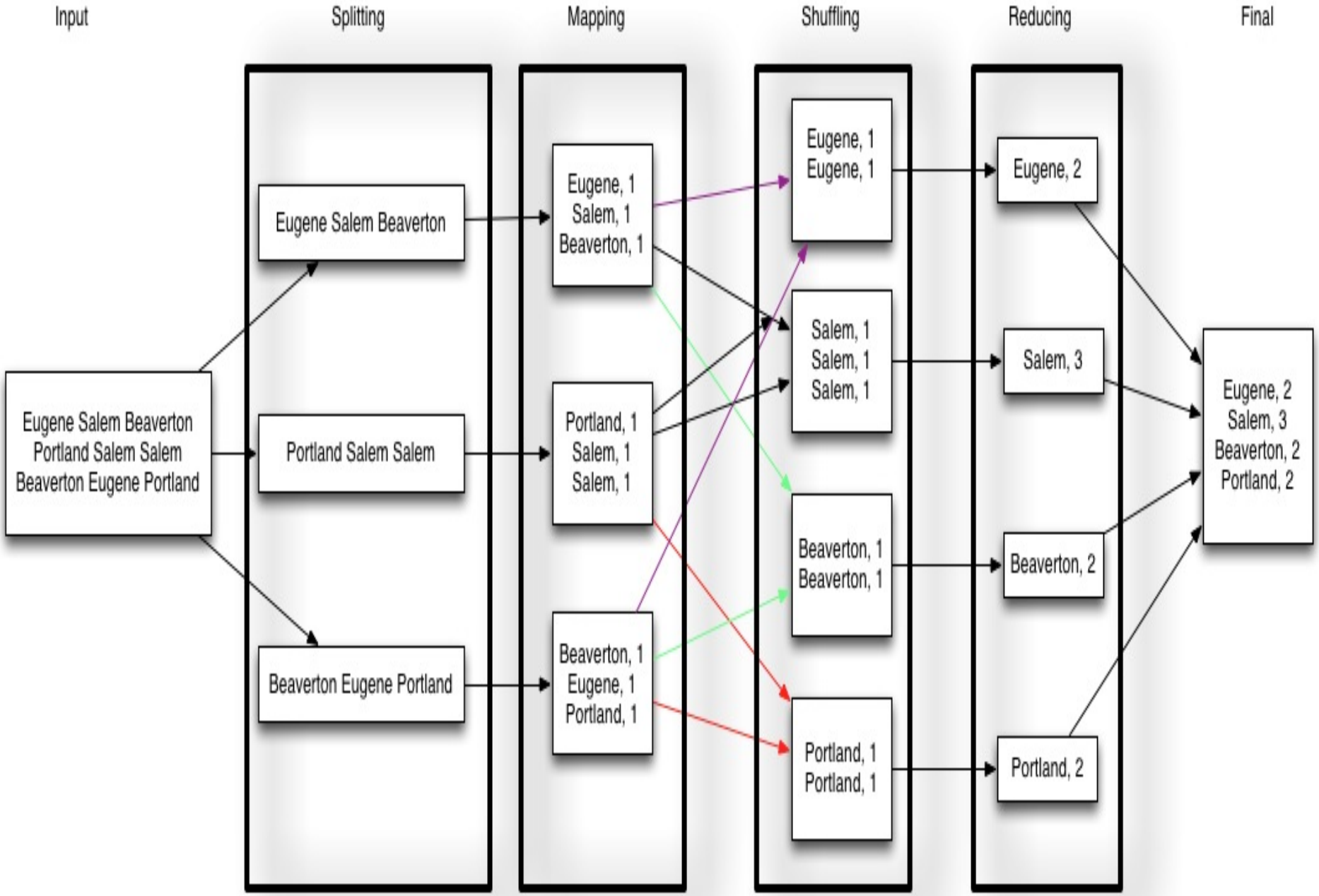
- MRv1 misconfig talk viewed 20k on slideshare

Agenda

- MapReduce Example
- MR2 Motivation
- Support Ticket Categorization
- What are Misconfigurations?
- Memory Misconfigurations
- Thread Misconfigurations
- Federation Misconfigurations
- YARN Memory Misconfigurations

Agenda

- MapReduce Example
- MR2 Motivation
- Support Ticket Categorization
- What are Misconfigurations?
- Memory Misconfigurations
- Thread Misconfigurations
- Federation Misconfigurations
- YARN Memory Misconfigurations


Input
Splitting
Mapping
Shuffling
Reducing
Final
Eugene Salem Beaverton Portland Salem Salem Beaverton Eugene Portland
Eugene Salem Beaverton
Portland Salem Salem
Beaverton Eugene Portland
Eugene, 1 Salem, 1 Beaverton, 1
Portland, 1 Salem, 1 Salem, 1
Beaverton, 1 Eugene, 1 Portland, 1
Eugene, 1 Eugene, 1
Salem, 1 Salem, 1 Salem, 1
Beaverton, 1 Beaverton, 1
Portland, 1 Portland, 1
Eugene, 2
Salem, 3
Beaverton, 2
Portland, 2
Eugene, 2 Salem, 3 Beaverton, 2 Portland, 2

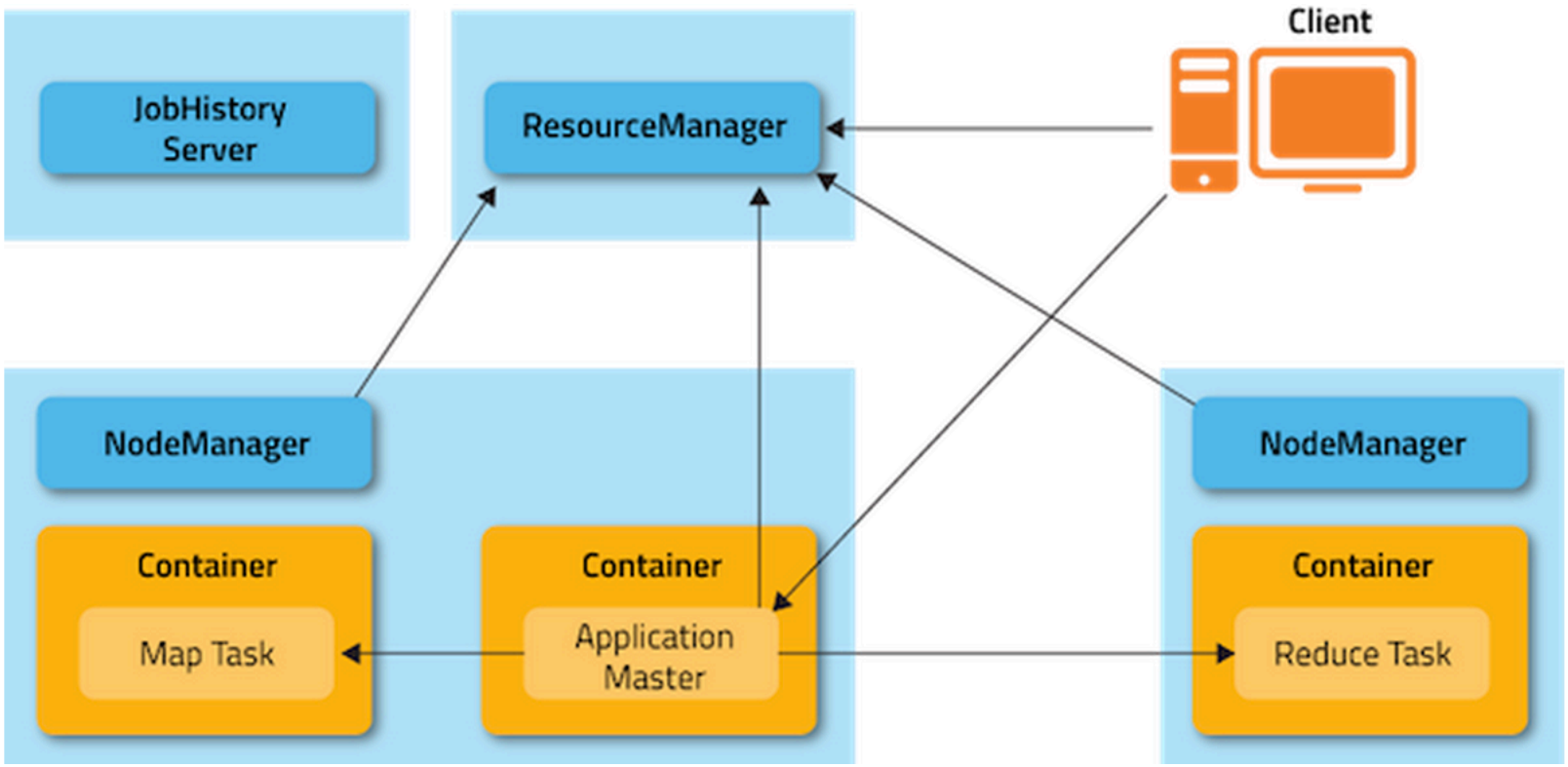
Agenda

- MapReduce Example
- MR2 Motivation
- Support Ticket Categorization
- What are Misconfigurations?
- Memory Misconfigurations
- Thread Misconfigurations
- Federation Misconfigurations
- YARN Memory Misconfigurations

MR2 Motivation

- Higher cluster utilization
  - Recommended MRv1 run only at 70% cap
  - Resources not used can be consumed by another
- Lower operational costs
  - One cluster running MR, Spark, Impala, etc
  - Don’t need to transfer data between clusters
  - Not restricted to < 5k cluster

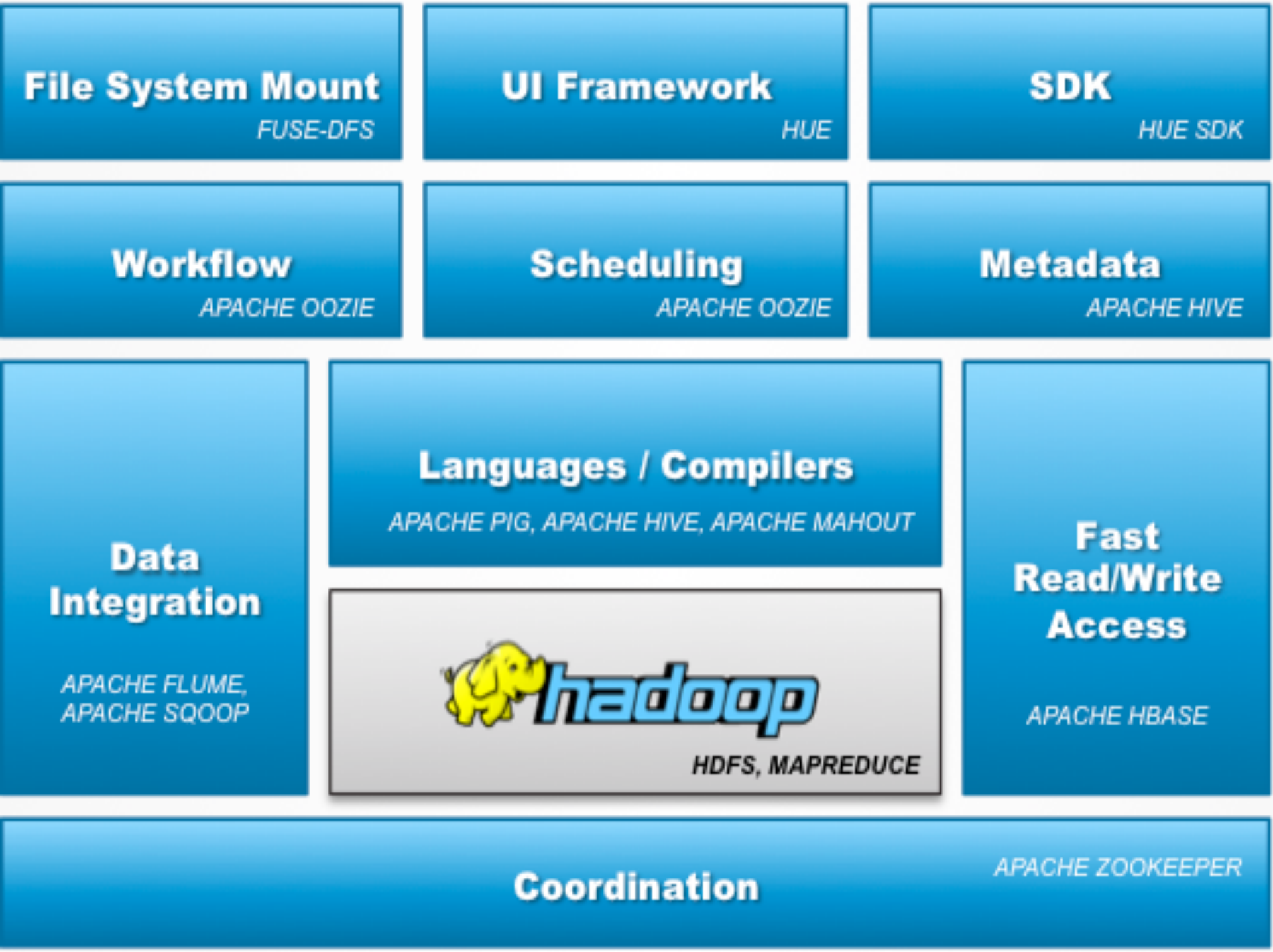
MRv2 Architecture



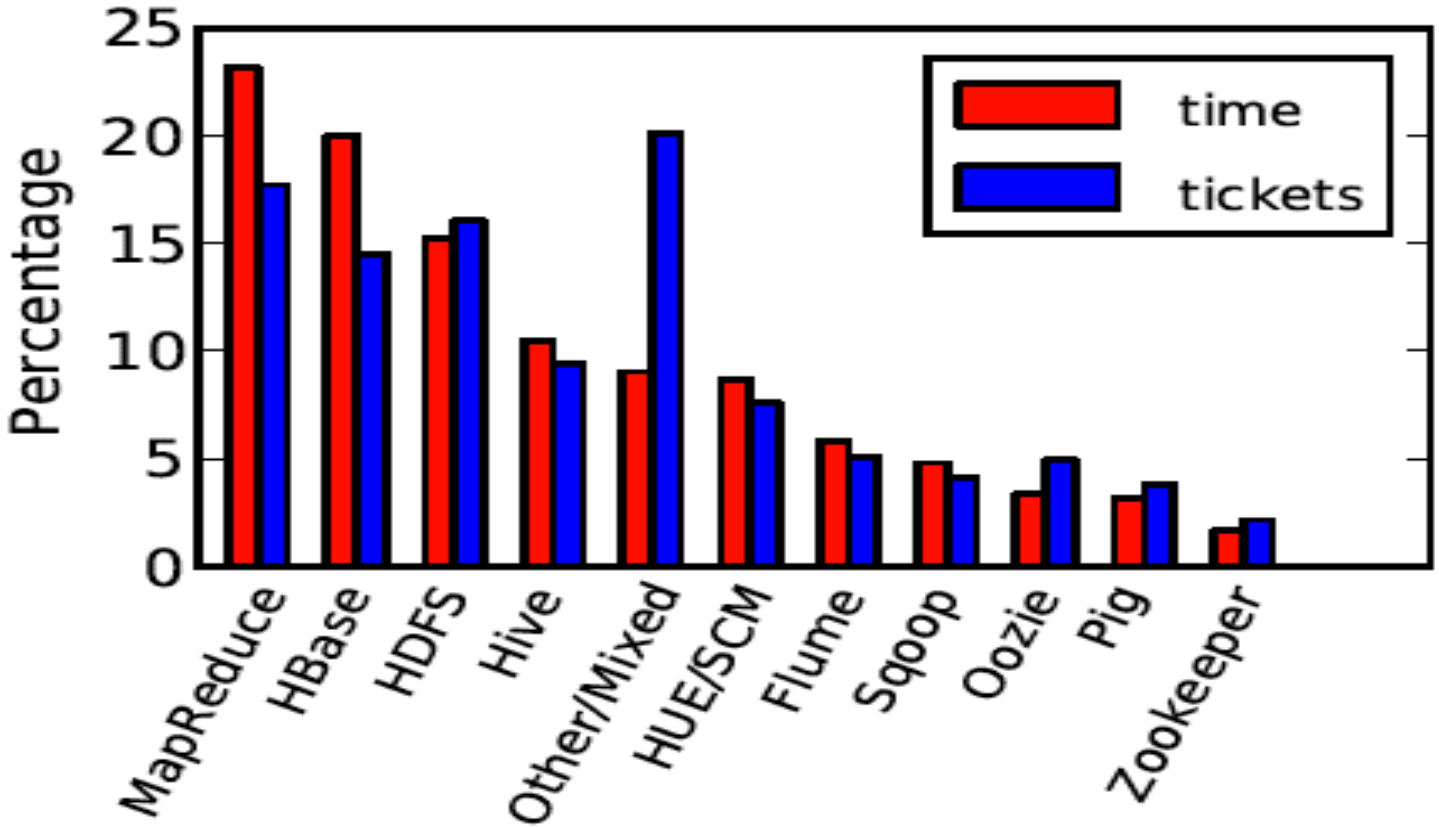
http://blog.cloudera.com/blog/2013/11/migrating-to-mapreduce-2-on-yarn-for-operators/

Agenda

- MapReduce Example
- MR2 Motivation
- Support Ticket Categorization
- What are Misconfigurations?
- Memory Misconfigurations
- Thread Misconfigurations
- Federation Misconfigurations
- YARN Memory Misconfigurations


File System Mount
FUSE-DFS
UI Framework
HUE
SDK
HUE SDK
Workflow
APACHE OOZIE
Scheduling
APACHE OOZIE
Metadata
APACHE HIVE
Data Integration
APACHE FLUME, APACHE SQOOP
Languages / Compilers
APACHE PIG, APACHE HIVE, APACHE MAHOUT
hadoop
HDFS, MAPREDUCE
Fast Read/Write Access
APACHE HBASE
Coordination
APACHE ZOOKEEPER

MapReduce is Central to Hadoop

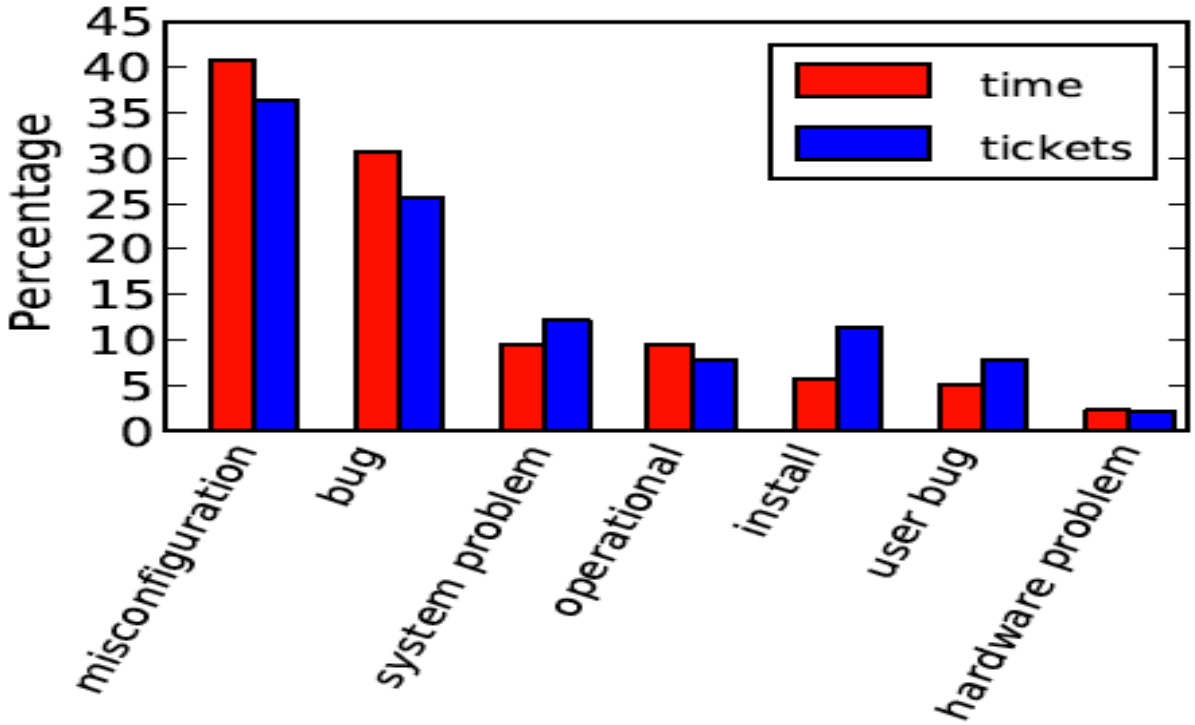


Agenda

- MapReduce Example
- MR2 Motivation
- Support Ticket Categorization
- What are Misconfigurations?
- Memory Misconfigurations
- Thread Misconfigurations
- Federation Misconfigurations
- YARN Memory Misconfigurations

What are Misconfigurations?

- Issues requiring change to Hadoop or to OS config files
- Comprises 35% of Cloudera Support Tickets
- e.g. resource-allocation: memory, file-handles, disk-space



Agenda

- MapReduce Example
- MR2 Motivation
- Support Ticket Categorization
- What are Misconfigurations?
- Memory Misconfigurations
- Thread Misconfigurations
- Federation Misconfigurations
- YARN Memory Misconfigurations

1. Task Out Of Memory Error (MRv1)

```
FATAL org.apache.hadoop.mapred.TaskTracker:
Error running child : java.lang.OutOfMemoryError:
Java heap space
       at org.apache.hadoop.mapred.MapTask
$MapOutputBuffer.<init>
```

- What does it mean?
  - Memory leak in task code
- What causes this?
  - MR task heap sizes will not fit

1. Task Out Of Memory Error (MRv1)

- MRv1 TaskTracker:
  - mapred.child.ulimit > 2*mapred.child.java.opts
  - 0.25*mapred.child.java.opts < io.sort.mb < 0.5*mapred.child.java.opts
- MRv1 DataNode:
  - Use short pathnames for dfs.data.dir names
    - e.g. /data/1, /data/2, /data/3
  - Increase DN heap
- MRv2:
  - Manual tuning of io.sort.record.percent not needed
  - Tune mapreduce.map|reduce.memory.mb
  - mapred.child.ulimit = yarn.nodemanager.vmem-pmem-ratio
    - Moot if yarn.nodemanager.vmem-check-enabled is disabled

Todd Lipcon
@tlipcon

Following

if sum(max heap size) > physical RAM - 3GB, go directly to jail. do not pass go. do not collect $200.

2. JobTracker Out of Memory Error

```
ERROR org.apache.hadoop.mapred.JobTracker: Job
initialization failed:
java.lang.OutOfMemoryError: Java heap space
at
org.apache.hadoop.mapred.TaskInProgress.<init>(TaskInProg
ress.java:122)
```

- What does it mean?
  - Total JT memory usage > allocated RAM
- What causes this?
  - Tasks too small
  - Too much job history

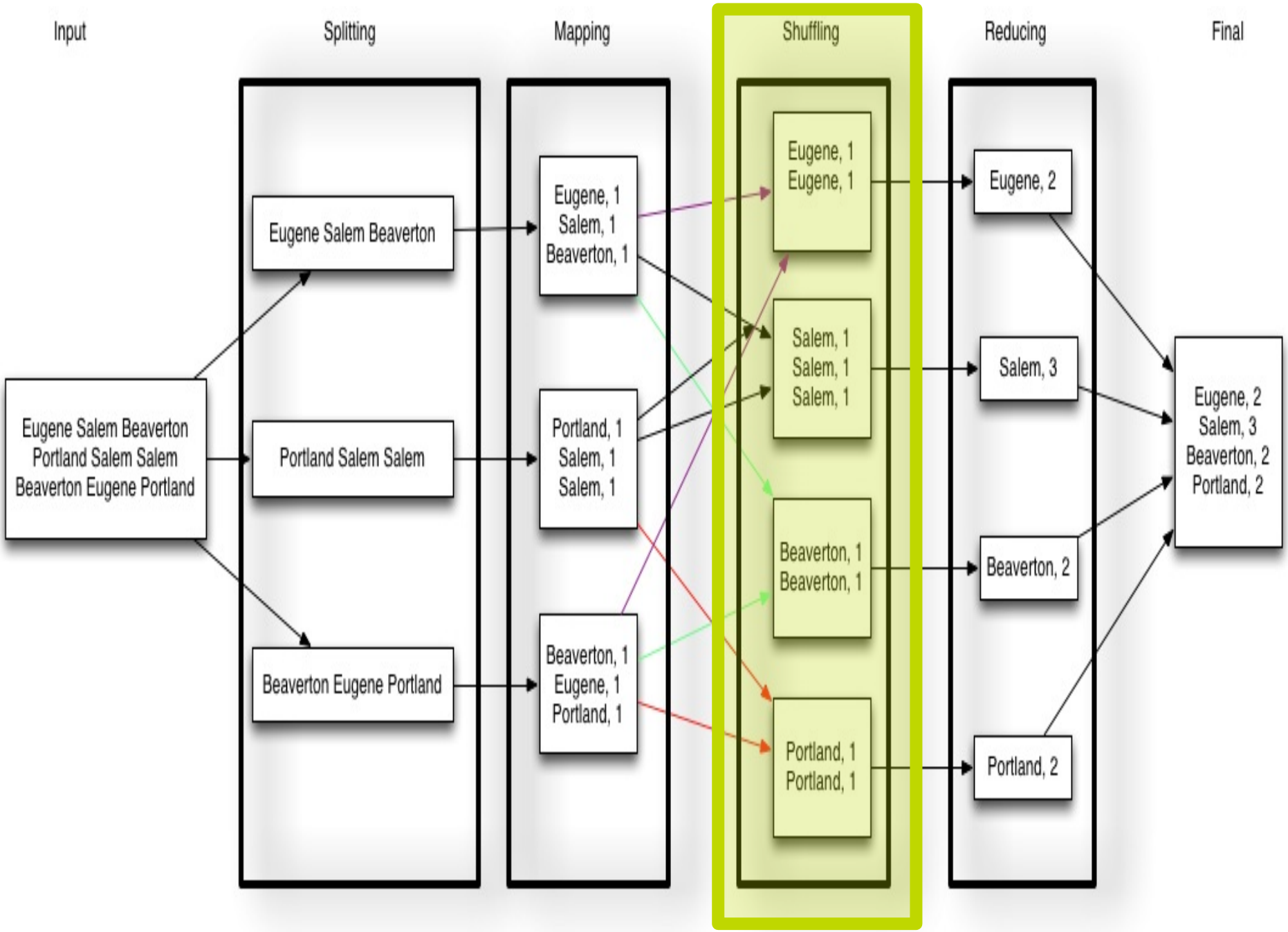
2. JobTracker Out of Memory Error

- How can it be resolved?
  - sudo -u mapreduce jmap -histo:live <pid>
    - histogram of what objects the JVM has allocated
  - Increase JT heap
  - Don't co-locate JT and NN
  - mapred.job.tracker.handler.count = ln(#TT)*20
  - mapred.jobtracker.completeuserjobs.maximum = 5
  - mapred.job.tracker.retiredjobs.cache.size = 100
  - mapred.jobtracker.retirejob.interval = 3600000
  - YARN has Uber AMs (run in single JVM)
    - One AM per MR job
      - Not restricted to keeping 5 jobs in memory

Agenda

- MapReduce Example
- MR2 Motivation
- Support Ticket Categorization
- What are Misconfigurations?
- Memory Misconfigurations
- Thread Misconfigurations
- Federation Misconfigurations
- YARN Memory Misconfigurations

Fetch Failures


Input
Splitting
Mapping
Shuffling
Reducing
Final
Eugene Salem Beaverton Portland Salem Salem Beaverton Eugene Portland
Eugene Salem Beaverton
Portland Salem Salem
Beaverton Eugene Portland
Eugene, 1 Salem, 1 Beaverton, 1
Portland, 1 Salem, 1 Salem, 1
Beaverton, 1 Eugene, 1 Portland, 1
Eugene, 1 Eugene, 1
Salem, 1 Salem, 1 Salem, 1
Beaverton, 1 Beaverton, 1
Portland, 1 Portland, 1
Eugene, 2
Salem, 3
Beaverton, 2
Portland, 2
Eugene, 2 Salem, 3 Beaverton, 2 Portland, 2

3. Too Many Fetch-Failures

MR1: `INFO org.apache.hadoop.mapred.JobInProgress: Too many fetch-failures for output of task`

MR2: `ERROR org.apache.hadoop.mapred.ShuffleHandler: Shuffle error: java.io.IOException: Broken pipe`

- What does it mean?
  - Reducer fetch operations fail to retrieve mapper outputs
  - Too many could blacklist the TT
- What causes this?
  - DNS issues
  - Not enough http threads on the mapper side
  - Not enough connections

3. Too Many Fetch-Failures

MR1:

- mapred.reduce.slowstart.completed.maps = 0.80
  - Unblocks other reducers to run while a big job waits on mappers
- tasktracker.http.threads = 80
  - Increases threads used to serve map output to reducers
- mapred.reduce.parallel.copies = SQRT(Nodes), floor of 10
  - Allows reducers to fetch map output in parallel

MR2:

- Set ShuffleHandler configs:
  - yarn.nodemanager.aux-services = mapreduce_shuffle
  - yarn.nodemanager.aux-services.mapreduce_shuffle.class = org.apache.hadoop.mapred.ShuffleHandler
- tasktracker.http.threads N/A
  - max # of threads is based on # of processors on machine
  - Uses Netty, allowing up to twice as many threads as there are processors

Agenda

- MapReduce Example
- MR2 Motivation
- Support Ticket Categorization
- What are Misconfigurations?
- Memory Misconfigurations
- Thread Misconfigurations
- Federation Misconfigurations
- YARN Memory Misconfigurations

4. Federation: Just (Don’t) Do It

= spreads FS metadata across NNs

= is stable (but ViewFS isn’t)

= is meant for 1k+ nodes

≠ multi-tenancy

≠ horizontally scale namespaces

➔NN HA + YARN

➔RPC QoS

Agenda

- MapReduce Example
- MR2 Motivation
- Support Ticket Categorization
- What are Misconfigurations?
- Memory Misconfigurations
- Thread Misconfigurations
- Federation Misconfigurations
- YARN Memory Misconfigurations

5. Optimizing YARN Virtual Memory Usage

Problem:

```
Current usage: 337.6 MB of 1 GB physical
memory used; 2.2 GB of 2.1 GB virtual memory
used. Killing container.
```

Solution:

- Set yarn.nodemanager.vmem-check-enabled = false
- Determine AM container size:
  yarn.app.mapreduce.am.resource.cpu-vcores
  yarn.app.mapreduce.am.resource.mb
- Sizing the AM: 1024mb (-Xmx768m)
  - Can be smaller because only storing one job unless using Uber

6. CPU Isolation in YARN Containers

- `mapreduce.map.cpu.vcores`
  `mapreduce.reduce.cpu.vcores`
  (per-job config)
- `yarn.nodemanager.resource.cpu-vcores`
  (slave service side resource config)
- `yarn.scheduler.minimum-allocation-vcores`
  `yarn.scheduler.maximum-allocation-vcores`
  (scheduler allocation control configs)
- `yarn.nodemanager.linux-container-executor.resources-handler.class`
  (turn on cgroups in NM)

7. Understanding YARN Virtual Memory

Situation:

`yarn.nodemanager.resource.cpu-vcores`

> actual cores

`yarn.nodemanager.resource.memory-mb`

> RAM

Effect:

- Exceeding cores = sharing existing cores, slower
- Exceeding RAM = swapping, OOM

Bonus: Fair Scheduler Errors

```
ERROR
org.apache.hadoop.yarn.server.resourc
emanager.scheduler.fair.FairScheduler
: Request for appInfo of unknown
attemptappattempt_1395214170909_0059_
000001
```

Harmless message fixed in YARN-1785

YARN Resources

- Migrating to MR2 on YARN:
  - For Operators: http://blog.cloudera.com/blog/2013/11/migrating-to-mapreduce-2-on-yarn-for-operators/
  - For Users: http://blog.cloudera.com/blog/2013/11/migrating-to-mapreduce-2-on-yarn-for-users/
  - http://blog.cloudera.com/blog/2014/04/apache-hadoop-yarn-avoiding-6-time-consuming-gotchas/
- Getting MR2 Up to Speed:
  - http://blog.cloudera.com/blog/2014/02/getting-mapreduce-2-up-to-speed/

Takeaways

- **Want to DIY?**
  - Take Cloudera's Admin Training - now with 4x the labs
- **Get it right the first time with monitoring tools.**
  - "Yep - we were able to download/install/configure/ setup a Cloudera Manager cluster from scratch in minutes :)"
- **Want misconfig updates?**
  - Follow @kate_ting