Taming Operations in the Apache Hadoop Ecosystem

Jon Hsieh, jon@cloudera.com
Kate Ting, kate@cloudera.com
USENIX LISA ’14
Nov 14, 2014
$ whoami

• Jon Hsieh, Cloudera
  – Software engineer
  – HBase Tech Lead
  – Apache HBase committer/PMC

• Kate Ting, Cloudera
  – Technical account manager
  – *Apache Sqoop Cookbook* co-author
  – Former customer operations engineer
Cloudera and Apache Hadoop

- Apache Hadoop is an open source software framework for distributed storage and distributed processing of Big Data on clusters of commodity hardware.
- Cloudera is revolutionizing enterprise data management by offering the first unified Platform for Big Data, an enterprise data hub built on Apache Hadoop.
  - Distributes CDH, a distribution Hadoop Distribution.
  - Teaches, consults, and supports customers building applications on the Hadoop stack.
  - The world-wide Cloudera Customer Operations Engineering team has closed tens of thousands of support incidents over six years.
Outline

• Anatomy of a Hadoop System

• Managing Hadoop Clusters

• Troubleshooting Hadoop Applications
Anatomy of a Hadoop System
Distributed systems

• A Distributed system is a set of many processes trying to acting like one a single process.

• Ideally you only deal with the system as a whole

• But in practice you deal with the parts

• And assemble a set of tools to bring it together
A Hadoop Machine

Machine
- Hadoop Daemons
- JVM
- Linux
- Disk, CPU, Mem
A Hadoop Rack

Rack

Top of Rack Switch

Machine
Hadoop Daemons
JVM
Linux
Disk, CPU, Mem

Machine
Hadoop Daemons
JVM
Linux
Disk, CPU, Mem

Machine
Hadoop Daemons
JVM
Linux
Disk, CPU, Mem

©2014 Cloudera, Inc. All rights reserved.
Hadoop Stack
Hadoop Stack - Storage

- Random Access Storage: HBase
- File Storage: HDFS
- Coordination: zookeeper; Security: Sentry
- JVM
- Linux
- Disk, CPU, Mem
Hadoop Stack - Processing

Batch processing: MapReduce

In Mem processing: Spark

Interactive SQL: Impala

Search: Solr

Resource Management: YARN

Random Access Storage: HBase

File Storage: HDFS

Coordination: zookeeper;
Security: Sentry

JVM
Linux
Disk, CPU, Mem

JVM
Linux
Disk, CPU, Mem

JVM
Linux
Disk, CPU, Mem

JVM
Linux
Disk, CPU, Mem

©2014 Cloudera, Inc. All rights reserved.
Hadoop Stack - Integration

Batch processing: MapReduce
In Mem processing: Spark
Interactive SQL: Impala
Search: Solr

Resource Management: YARN

Random Access Storage: HBase
File Storage: HDFS

Coordination: zookeeper;
Security: Sentry

Event ingest: Flume, Kafka
DB Import Export: Sqoop

Other systems: httpd, sas, custom apps etc

©2014 Cloudera, Inc. All rights reserved.
Hadoop Stack – User Interface

User interface: HUE

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Batch processing: MapReduce

In Mem processing: Spark

Interactive SQL: Impala

Search: Solr

Resource Management: YARN

Random Access Storage: HBase

File Storage: HDFS

JVM

Linux

Disk, CPU, Mem

Coordination: zookeeper;
Security: Sentry

©2014 Cloudera, Inc. All rights reserved.
Hadoop Stack

User interface: HUE

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Batch processing: MapReduce

In Mem processing: Spark

Interactive SQL: Impala

Search: Solr

Resource Management: YARN

Random Access Storage: HBase

File Storage: HDFS

Coordination: zookeeper;
Security: Sentry

JVM
Linux
Disk, CPU, Mem

JVM
Linux
Disk, CPU, Mem

JVM
Linux
Disk, CPU, Mem

JVM
Linux
Disk, CPU, Mem

Event ingest: Flume, Kafka

DB Import Export: Sqoop

Other systems: httpd, sas, custom apps etc

©2014 Cloudera, Inc. All rights reserved.
Managing a Hadoop Cluster
Understanding Normal

• Establish normal
  – Boring logs are good

• So that you can detect abnormal
  – Find anomalies and outliers
  – Compare performance

• And then isolate root causes
  – Who are the suspects
  – Pull the thread by interrogating
  – Correlate across different subsystems
Basic tools

• What happened?
  – Logs

• What state are we in now?
  – Metrics
  – Thread stacks

• What happens when I do this?
  – Tracing
  – Dumps

• Is it alive?
  – listings
  – Canaries

• Is it OK?
  – fsck
Diagnostics for single machines

Analysis and Tools

©2014 Cloudera, Inc. All rights reserved.
## Diagnosis Tools

<table>
<thead>
<tr>
<th></th>
<th>HW/Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logs</strong></td>
<td>/log, /var/log, dmesg</td>
</tr>
<tr>
<td><strong>Metrics</strong></td>
<td>/proc, top, iotop, sar, vmstat, netstat</td>
</tr>
<tr>
<td><strong>Tracing</strong></td>
<td>Strace, tcpdump</td>
</tr>
<tr>
<td><strong>Dumps</strong></td>
<td>Core dumps</td>
</tr>
<tr>
<td><strong>Liveness</strong></td>
<td>ping</td>
</tr>
<tr>
<td><strong>Corrupt</strong></td>
<td>fsck</td>
</tr>
</tbody>
</table>
Diagnostics for the JVM

- Most Hadoop services run in the Java VM, intros new java subsystems
  - Just in time compiler
  - Threads
  - Garbage collection

- Dumping Current threads
  - jstacks (any threads stuck or blocked?)

- JVM Settings:
  - Enabling GC Logs: `-XX:+PrintGCDateStamps` `-XX:+PrintGCDetails`
  - Enabling OOME Heap dumps: `-XX:+HeapDumpOnOutOfMemoryError`

- Metrics on GC
  - Get gc counts and info with `jstat`

- Memory Usage dumps
  - Create heap dump: `jmap -dump:live,format=b,file=heap.bin <pid>`
  - View heap dump from crash or jmap with `jhat`
## Diagnosis Tools

<table>
<thead>
<tr>
<th></th>
<th>HW/Kernel</th>
<th>JVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logs</td>
<td>/log, /var/log dmesg</td>
<td>Enable gc logging</td>
</tr>
<tr>
<td>Metrics</td>
<td>/proc, top, iotop, sar, vmstat, netstat</td>
<td>jstat</td>
</tr>
<tr>
<td>Tracing</td>
<td>Strace, tcpdump</td>
<td>Debugger, jdb, jstack</td>
</tr>
<tr>
<td>Dumps</td>
<td>Core dumps</td>
<td>jmap</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enable OOME dump</td>
</tr>
<tr>
<td>Liveness</td>
<td>ping</td>
<td>Jps</td>
</tr>
<tr>
<td>Corrupt</td>
<td>fsck</td>
<td></td>
</tr>
</tbody>
</table>

©2014 Cloudera, Inc. All rights reserved.
Tools for lots of machines

Other systems: httpd, sas, custom apps etc
Tools for lots of machines

Other systems: httpd, sas, custom apps etc

©2014 Cloudera, Inc. All rights reserved.
Ganglia

- Collects and aggregates metrics from machines
- Good for what’s going on right now and describing normal perf
- Organized by physical resource (CPU, Mem, Disk)
  - Good defaults
  - Good for pin pointing machines
  - Good for seeing overall utilization
  - Uses RRDTool under the covers

- Some data scalability limitations, lossy over time.
- Dynamic for new machines, requires config for new metrics
Graphite

- Popular alternative to ganglia
- Can handle the scale of metrics coming in
- Similar to ganglia, but uses its own RRD database.
- More aimed at dynamic metrics (as opposed to statically defined metrics)
Nagios

- Provides alerts from canaries and basic health checks for services on machines
- Organized by Service (httpd, dns, etc)
- Defacto standard for service monitoring
- Lacks distributed system know how
  - Requires bespoke setup for service slaves and masters
  - Lacks details with multitenant services or short-lived jobs
Tools for the Hadoop Stack
Tools for the Hadoop Stack

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Batch processing: MapReduce

In Mem processing: Spark

Interactive SQL: Impala

Search: Solr

Resource Management: YARN

Random Access Storage: HBase

File Storage: HDFS

Coordination: Zookeeper; Security: Sentry

Event ingest: Flume, Kafka

DB Import Export: Sqoop

Other systems: httpd, sas, custom apps etc

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Ganglia*, Nagios*

Ganglia, Nagios

User interface: HUE
Tools for the Hadoop Stack

User interface: HUE

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Batch processing: MapReduce

In Mem processing: Spark

Interactive SQL: Impala

Search: Solr

Random Access Storage: HBase

File Storage: HDFS

Resource Management: YARN

Coordination: zookeeper; Security: Sentry

Event ingest: Flume, Kafka

DB Import Export: Sqoop

Other systems: httpd, sas, custom apps etc

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Batch processing: MapReduce

In Mem processing: Spark

Interactive SQL: Impala

Search: Solr

Random Access Storage: HBase

File Storage: HDFS

Resource Management: YARN

Coordination: zookeeper; Security: Sentry

Event ingest: Flume, Kafka

DB Import Export: Sqoop

Other systems: httpd, sas, custom apps etc

©2014 Cloudera, Inc. All rights reserved.
Diagnostics for the Hadoop Stack

• Single client call can trigger many RPCs spanning many machines
• Systems are evolving quickly
• A failure on one daemon, by design, does not cause failure of the entire service

• Logs:
  – Each service’s master and slaves have their own logs: /var/log/
  – There are lot of logs and they change frequently

• Metrics:
  – Each daemon offers metrics, often aggregated at masters

• Tracing:
  – Htrace (integrated into Hadoop, HBase, recently in Apache Incubator)

• Liveness:
  – Canaries, service/daemon web uis
## Diagnosis Tools

<table>
<thead>
<tr>
<th></th>
<th>HW/Kernel</th>
<th>JVM</th>
<th>Hadoop Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logs</td>
<td>/log, /var/log dmesg</td>
<td>Enable gc logging</td>
<td>*/var/log/hadoop</td>
</tr>
<tr>
<td>Metrics</td>
<td>/proc, top, iotop, sar, vmstat, netstat</td>
<td>jstat</td>
<td>*/stacks, */jmx,</td>
</tr>
<tr>
<td>Tracing</td>
<td>Strace, tcpdump</td>
<td>Debugger, jdb, jstack</td>
<td>htrace</td>
</tr>
<tr>
<td>Dumps</td>
<td>Core dumps</td>
<td>jmap</td>
<td>*/dump</td>
</tr>
<tr>
<td>Liveness</td>
<td>ping</td>
<td>Jps</td>
<td>Web UI</td>
</tr>
<tr>
<td>Corrupt</td>
<td>fsck</td>
<td></td>
<td>HDFS fsck, HBase hbck</td>
</tr>
</tbody>
</table>

©2014 Cloudera, Inc. All rights reserved.
Tools for the Hadoop Stack

- **Batch processing:** MapReduce
- **In Mem processing:** Spark
- **Interactive SQL:** Impala
- **Search:** Solr
- **Resource Management:** YARN
- **Random Access Storage:** HBase
- **File Storage:** HDFS
- **Coordination:** zookeeper; Security: Sentry
- **Coordination:** Zookeeper
- **Security:** Sentry
- **Event ingest:** Flume, Kafka
- **DB Import Export:** Sqoop
- **Other systems:** httpd, sas, custom apps etc

**Languages/APIs:** Hive, Pig, Crunch, Kite, Mahout
**Interactive SQL:** Impala
**Search:** Solr
**Resource Management:** YARN
**Random Access Storage:** HBase
**File Storage:** HDFS
**Coordination:** Zookeeper
**Security:** Sentry
**Event ingest:** Flume, Kafka
**DB Import Export:** Sqoop
**Other systems:** httpd, sas, custom apps etc

©2014 Cloudera, Inc. All rights reserved.
Tools for the Hadoop Stack

User interface: HUE

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Batch processing: MapReduce

In Mem processing: Spark

Interactive SQL: Impala

Search: Solr

Resource Management: YARN

Ganglia*, Nagios*

Random Access Storage: HBase

File Storage: HDFS

Coordination: zookeeper

Security: Sentry

Event ingest: Flume, Kafka

DB Import Export: Sqoop

Other systems: httpd, sas, custom apps etc

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Ganglia*, Nagios*

Ganglia, Nagios

©2014 Cloudera, Inc. All rights reserved.
Ganglia and Nagios are not enough

- Fault tolerant distributed system masks many problems (by design!)
  - Some failures are not critical – failure condition more complicated
- Lacks distributed system know how
  - Requires bespoke setup for service slaves and masters
  - Lacks details with multitenant services or short-lived jobs
- Hadoop services are logically dependent on each other
  - Need to correlate metrics across different service and machines
  - Need to correlate logs from different services and machines.
  - Young systems where Logs are changing frequently
  - What about all the logs?
  - What of all these metrics do we really need?
- Some data scalability limitations, lossy over time
  - What about full fidelity?
OpenTSDB (Time Series Database)
- Efficiently stores metric data into HBase
- Keeps data at full fidelity
- Keep as much data as your HBase instance can handle.

Free and Open Source
Cloudera Manager

- Extracts hardware, OS, and Hadoop service metrics specifically relevant to Hadoop and its related services.
  - Operation Latencies
  - # of disk seeks
  - HDFS data written
  - Java GC time
  - Network IO

- Provides
  - Cluster preflight checks
  - Basic host checks
  - Regular health checks

- Uses RRD

- Provides distributed log search

- Monitors logs for known issues

- Point and click for useful utils (lsOf, jstack, jmap)

- Free
Troubleshooting a Hadoop System
The Law of Cluster Inertia

A cluster in a **good state** stays in a **good state**, and a cluster in a **bad state** stays in a **bad state**, unless acted upon by an **external force**.
External Forces

- Failures
- Acts of God
- Users
- Admins
Failures as an External Force

User interface: HUE

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Interactive SQL: Impala

Search: Solr

Resource Management: YARN

Random Access Storage: HBase

File Storage: HDFS

Coordination: zookeeper; Security: Sentry

Event ingest: Flume, Kafka

DB Import Export: Sqoop

Other systems: httpd, sas, custom apps etc

©2014 Cloudera, Inc. All rights reserved.
Acts of God as an External Force

- Cluster
- Backbone Switch
- Rack
- Top of Rack Switch
- Machine
  - Hadoop Daemons
  - JVM
  - Linux
  - Disk, CPU, Mem
- Rack
- Top of Rack Switch
- Machine
  - Hadoop Daemons
  - JVM
  - Linux
  - Disk, CPU, Mem

WAN

Power Failure

©2014 Cloudera, Inc. All rights reserved.
Users as an external force

• Use Security to protect systems from users
  – Prevent and track

• Authentication – proving who you are
  – LDAP, Kerberos

• Authorization – deciding what you are allowed to do
  – Apache Sentry (incubating), Hadoop security, HBase security

• Audit – who and when was something done?
  – Cloudera Navigator
Admns as an external force

Upgrades
• Linux
• Hadoop
• Java

Misconfiguration
• Memory Mismanagement
  – TT OOME
  – JT OOME
  – Native Threads
• Thread Mismanagement
  – Fetch Failures
  – Replicas
• Disk Mismanagement
  – No File
  – Too Many Files
Example Application Pipeline

ingest process export

ingest process export

ingest process export

©2014 Cloudera, Inc. All rights reserved.
Example Application Pipeline - Ingest

Random Access Storage: HBase

File Storage, HDFS

Coordination: zookeeper;
Security: Sentry

Custom app feeds event data in to HBase

Other systems: httpd, sas, custom apps etc

©2014 Cloudera, Inc. All rights reserved.
Example Application Pipeline - processing

Languages/APIs: Hive, Pig, Crunch, Kite, Mahout

Batch Processing: MapReduce

Resource Management: YARN

Random Access Storage: HBase

File Storage: HDFS

Coordination: zookeeper; Security: Sentry

Event ingest: Flume, Kafka

Other systems: httpd, sas, custom apps etc

Map Reduce Job generates new artifact from HBase data and writes to hdfs

Map Reduce Job generates new artifact from HBase data and writes to hdfs
Case study 1: slow jobs after Hadoop upgrade

Symptom:
After an upgrade, activity on the cluster eventually began to slow down and the job queue overflowed.
Finding the right part of the stack

E-SPORE (from Eric Sammer’s *Hadoop Operations*)

- **Environment**
  - What is different about the environment now from the last time everything worked?

- **Stack**
  - The entire cluster also has shared dependency on data center infrastructure such as the network, DNS, and other services.

- **Patterns**
  - Are the tasks from the same job? Are they all assigned to the same tasktracker? Do they all use a shared library that was changed recently?

- **Output**
  - Always check log output for exceptions but don’t assume the symptom correlates to the root cause.

- **Resources**
  - Do local disks have enough? Is the machine swapping? Does the network utilization look normal? Does the CPU utilization look normal?

- **Event correlation**
  - It’s important to know the order in which the events led to the failure.
Example Application Pipeline - processing

- Batch processing: MapReduce
- Resource Management: YARN
- Random Access Storage: HBase
- File Storage: HDFS
- Languages/APIs: Hive, Pig, Crunch, Kite, Mahout
- Map Reduce Job generates new artifact from HBase data and writes to hdfs
- Coordination: zookeeper; Security: Sentry
- Event ingest: Flume, Kafka
- Other systems: httpd, sas, custom apps etc
- Disk, CPU, Mem

©2014 Cloudera, Inc. All rights reserved.
Case study 1: slow jobs after Hadoop upgrade

Evidence:

Isolated to Processing phase (MR).

In TT Logs, found an innocuous but anomalous log entry about “fetch failures.”

Many users had run into this MR problem using different versions of MR.

Workaround provided: remove the problem node from the cluster.
Case study 1: slow jobs after Hadoop upgrade

Root cause:
All MR versions had a common dependency on a particular version of Jetty (Jetty 6.1.26).

Dev was able to reproduce and fix the bug in Jetty.
Case study 2: slow jobs after Linux upgrade

Symptom:
After an upgrade, system CPU usage peaked at 30% or more of the total CPU usage.
Case study 2: slow jobs after Linux upgrade

Evidence:

Used tracing tools to isolate a majority of time was inexplicably spent in virtual memory calls.

http://structureddata.org/2012/06/18/linux-6-transparent-huge-pages-and-hadoop-workloads/
Machine

Hadoop Daemons

JVM

Linux

Disk, CPU, Mem

©2014 Cloudera, Inc. All rights reserved.
Case study 2: slow jobs after Linux upgrade

Root cause:
Running RHEL or CentOS versions 6.2, 6.3, and 6.4 or SLES 11 SP2 has a feature called "Transparent Huge Page (THP)" compaction which interacts poorly with Hadoop workloads.
Case study 3: slow jobs at a precise moment

Symptom:
High CPU usage and responsive but sluggish cluster.

30 customers all hit this at the exact same time: 6/30/12 at 5pm PDT.
Case study 3: slow jobs at a precise moment

Evidence:
Checked the kernel message buffer (run `dmesg`) and look for output confirming the leap second injection.

Other systems had same problem.
Case study 3: slow jobs at a precise moment

Root cause:
Linux OS kernel mishandled a leap second added.
Similar symptoms, different problem
Case study 1: slow jobs after Hadoop upgrade

After an upgrade, activity on the cluster eventually began to slow down and the job queue overflowed.

In TT Logs, found an innocuous but anomalous log entry about “fetch failures.” Many users had run into this MR problem using different versions of MR. Workaround provided: remove the problem node from the cluster.

All MR versions had a common dependency on a particular version of Jetty (Jetty 6.1.26). Dev was able to reproduce and fix the bug in Jetty.
Case study 2: slow jobs after Linux upgrade

Symptom
After an upgrade, system CPU usage peaked at 30% or more of the total CPU usage.

Evidence
HT to Greg Rahn, Brendan Gregg’s flame graph tool, and perf tool which proved that majority of time was inexplicably spent in virtual memory calls.

Root Cause
Running RHEL or CentOS versions 6.2, 6.3, and 6.4 or SLES 11 SP2 has a feature called “Transparent Huge Page (THP)” compaction which interacts poorly with Hadoop workloads.
## Case study 3: slow jobs at a precise moment

### Symptom
High CPU usage and responsive but sluggish cluster.
30 customers all hit this at the exact same time.

### Evidence
Checked the kernel message buffer (run `dmesg`) and look for output confirming the leap second injection.

### Root Cause
Linux OS kernel mishandled a leap second added on 6/30/12 at 5pm PDT.
Lessons learned

Methodology
More crucial than the specific troubleshooting methodology used is to use one.

Tools
More crucial than the specific tool used is the type of data analyzed and how it’s analyzed.

Learn from failure
Capture for posterity in a knowledge base article, blog post, or conference presentation.

©2014 Cloudera, Inc. All rights reserved.
Conclusions
Takeaway

Anatomy of a Hadoop System
A cluster in a good state stays in a good state, and a cluster in a bad state stays in a bad state, unless acted upon by an external force.

Managing Hadoop Clusters
Cloudera has seen a lot of diverse clusters and used that experience to build tools to help diagnose and understand how Hadoop operates.

Troubleshooting Hadoop Applications
Similar symptoms can lead to different root causes. Use tools to assist with event correlation and pattern determination.
Shameless plug

Cloudera Live
A full demo cluster + tutorial and sample data/queries in the cloud – free access for two weeks (plus, clusters in Tableau and Zoomdata flavors!)
cloudera.com/live

Cloudera Community
Ask technical questions/get technical answers from Cloudera Software Engineers and other users
cloudera.com/community

In Europe!
Account Executives, Solutions Architects, Training
cloudera.com/careers
Thank you!

Office hours @2pm in Willow A

@jmhsieh
@kate_ting