cloudera
Ask Bigger Questions

Debugging Distributed Systems

Philip Zeyliger | philip@cloudera | @philz42 | Software Engineer

Strata, February 27, 2013

$whoami; whois cloudera.com

Purveyors of fine distributed software, including HDFS, MapReduce, HBase, Zookeeper, Impala, Hue, Crunch, Avro, Sqoop, Flume, ...

I work on Cloudera Manager (new version out yesterday!), helping our customers focus on their data problems, not their distributed system problems.

if only it were as easy as picking out the black sheep...

But it's usually more like this... Where's Walderbug?

Why so hard?

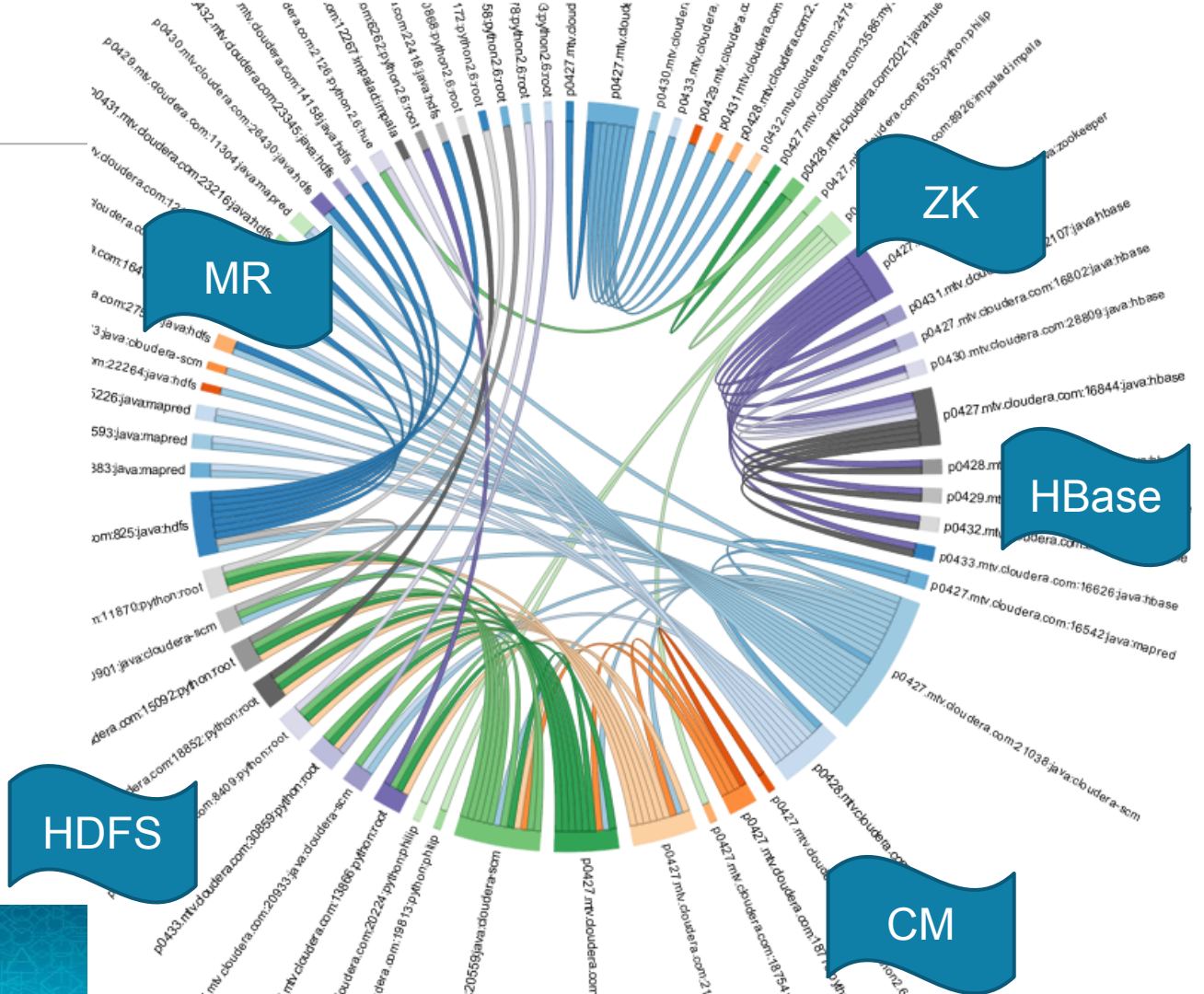
Layers

Networks

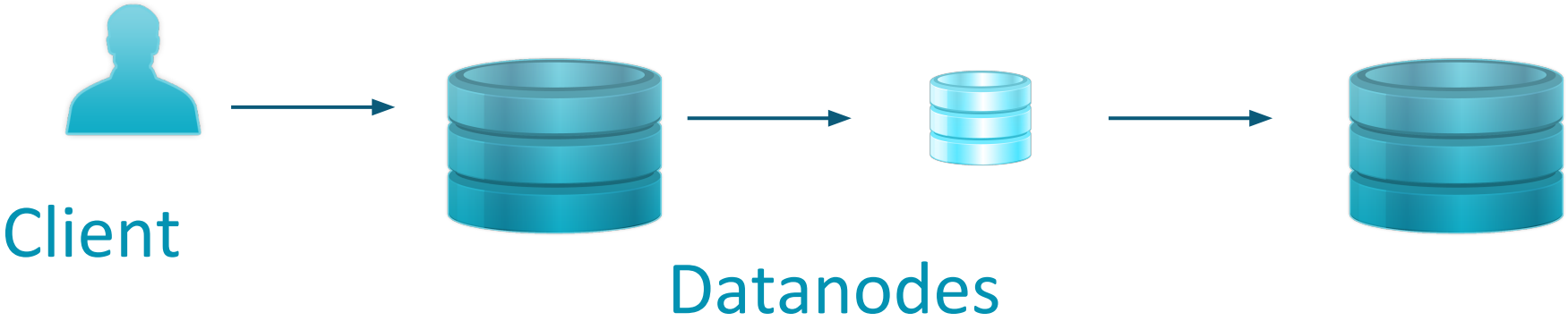
Partial Failure

Networks!

Excerpt of TCP Connections between components in a small cluster.



Writing to HDFS is a Relay Race



When one element in the relay race is slow, the entire team loses.

The Patented* Two Step Process

*not really

Step 1:
Figure out where the problem may be...
Find outliers in logs & metrics, tracing...

Zoom in

Rinse, Repeat after false starts

Step 2:
Dig in!
strace, Java, etc.

Preconditions

Versions the same?

DNS working? Really?

Clocks in sync?

Adding "host inspector" to detect these common issues helped significantly.

Inspector Results

Validations

- ✓ Inspector ran on all 7 hosts.
- ⚠ The following failures were observed in checking hostnames... ▼
  - Host p0430.mtv.cloudera.com expected to have name p0430.mtv.cloudera.com but resolved
  - Host p0432.mtv.cloudera.com expected to have name p0432.mtv.cloudera.com but res
- ✓ No errors were found while looking for conflicting init scripts.
- ✓ No errors were found while checking /etc/hosts.
- ✓ All hosts resolved localhost to 127.0.0.1.
- ✓ All hosts checked resolved each other's hostnames correctly.
- ✓ Host clocks are approximately in sync (within ten minutes).
- ✓ Host time zones are consistent across the cluster.
- ✓ No users or groups are missing.
- ✓ No kernel versions that are known to be bad are running.
- ✓ 0 hosts are running CDH3 and 7 hosts are running CDH4.
- ✓ All checked hosts are running the same version of components.
- ✓ All checked Cloudera Management Daemons versions are consistent with the server.
- ✓ All checked Cloudera Management Agents versions are consistent with the server.

Common (or, unpleasant) issues

Version Summary

Group 1 (CDH4)		
Hosts		
p0427.mtv.cloudera.com, p0428.mtv.cloudera.com, p0429.mtv.cloudera.com, p0430.mtv.cloudera.com		
Component	Version	
Flume NG	1.3.0-cdh4.2.0	
MapReduce 1 (CDH4 only)	2.0.0-mr1-cdh4.2.0	
HDFS (CDH4 only)	2.0.0-cdh4.2.0	
HttpFS (CDH4 only)	2.0.0-cdh4.2.0	
MapReduce 2 (CDH4 only)	2.0.0-cdh4.2.0	
Yarn (CDH4 only)	2.0.0-cdh4.2.0	
Hadoop	2.0.0-cdh4.2.0	
HBase	0.94.2-cdh4.2.0	
Hive	0.10.0-cdh4.2.0	
Mahout	0.7-cdh4.2.0	
Oozie	3.3.0-cdh4.2.0	
Pig	0.10.0-cdh4.2.0	
Sqoop	1.99.1-cdh4.2.0	
Whirr	0.8.0-cdh4.2.0	
Zookeeper	3.4.5-cdh4.2.0	CDH4
Impala	0.6-SNAPSHOT	Not applicable
Cloudera Manager Management Daemons	4.5.0-SNAPSHOT	Not applicable
Cloudera Manager Agent	4.5.0-SNAPSHOT	Not applicable

Versions across the cluster

Inspector Results

Validations

- ✓ Inspector ran on all 7 hosts.
- ⚠ The following failures were observed in checking hostnames... ▼
  - Host p0430.mtv.cloudera.com expected to have name p0430.mtv.cloudera.com but resolved (InetAddress.getLocalHost().getHostName()) itself to p0430.sf.cloudera
  - Host p0432.mtv.cloudera.com expected to have name p0432.mtv.cloudera.com but resolved (InetAddress.getLocalHost().getHostName()) itself to p0432.sf.cloudera
- ✓ No errors were found while looking for conflicting init scripts.
- ✓ No errors were found while checking /etc/hosts.
- ✓ All hosts resolved localhost to 127.0.0.1.
- ✓ All hosts checked resolved each other's hostnames correctly.
- ✓ Host clocks are approximately in sync (within ten minutes).
- ✓ Host time zones are consistent across the cluster.
- ✓ No users or groups are missing.
- ✓ No kernel versions that are known to be bad are running.
- ✓ 0 hosts are running CDH3 and 7 hosts are running CDH4.
- ✓ All checked hosts are running the same version of components.
- ✓ All checked Cloudera Management Daemons versions are consistent with the server.
- ✓ All checked Cloudera Management Agents versions are consistent with the server.

Version Summary

Group 1 (CDH4)

Easy: Health tests and monitoring

Life's better when a monitoring system tells you where to focus.

"Not able to communicate with the web server."

tasktracker (p0430)

Status | Processes | Commands

Summary

Host p0430.mtv.cloudera.com Good

Event Search: Alerts , Critical , All

Health Tests

Collapse All

- The Cloudera Manager agent was not able to communicate with this role's web server.
- 7 good.
  - This role's status was as expected. The role was started.
  - The health of this role's host was good.
  - Open file descriptors: 189. File descriptor limit: 32,768. Percentage in use: 0.58%.
  - This TaskTracker was not blacklisted.
  - This TaskTracker was connected to its JobTracker.
  - Average time spent in garbage collection was 5 ms per minute over the previous 5 minute(s).
  - This role's log directory (/var/log/hadoop-0.20-mapreduce) was on a filesystem with more than 10.0 GiB of its space free.

Health History

12:38 PM	Bad	Web Server Status	View
n 12 5:33 PM	Good	GC Duration	View
2 5:31 PM	Good	2 Good	View
2 5:31 PM	Good	4 Good	View
Jan 2 6:37 AM	Good	JobTracker Connectivity	View
Jan 2 6:36 AM	Bad	JobTracker Connectivity	View

[Cloudera Alert] The health of service mapreduce1 has become bad. Inbox x

noreply@localhost.mtv.cloudera.com
to me

The health test result for MAPREDUCE_TASK_TRACKERS_HEALTHY has become bad: Healthy TaskTrackers: 6. Concerning Tas
Percent healthy: 85.71%. Percent healthy or concerning: 85.71%. Critical threshold: 90.00%.

Time: Feb 20, 2013 12:38:31 PM

View Details on p0427.mtv.cloudera.com

Monitor Startup: false
Cluster: Cluster 1 - CDH4
Service: mapreduce1
Service Type: MapReduce
Health Test Results:

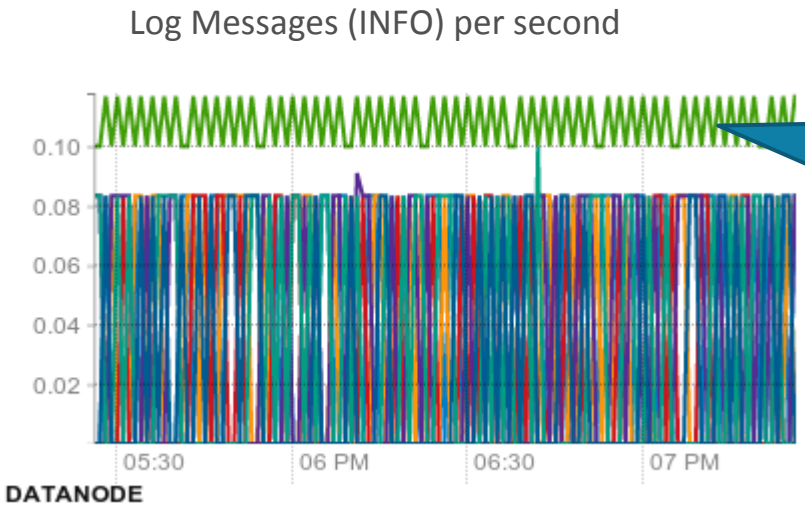
Health Test Name	Event Code	Severity	Co
MAPREDUCE_TASK_TRACKERS_HEALTHY	Service health check bad	Critical	The hea become ba TaskTrackers. 85.71%. Critical thres

E-mail, with link to the problem

Outliers: Logs

Don't just read logs; they're full of lies.

Instead, look at distribution of log sizes.



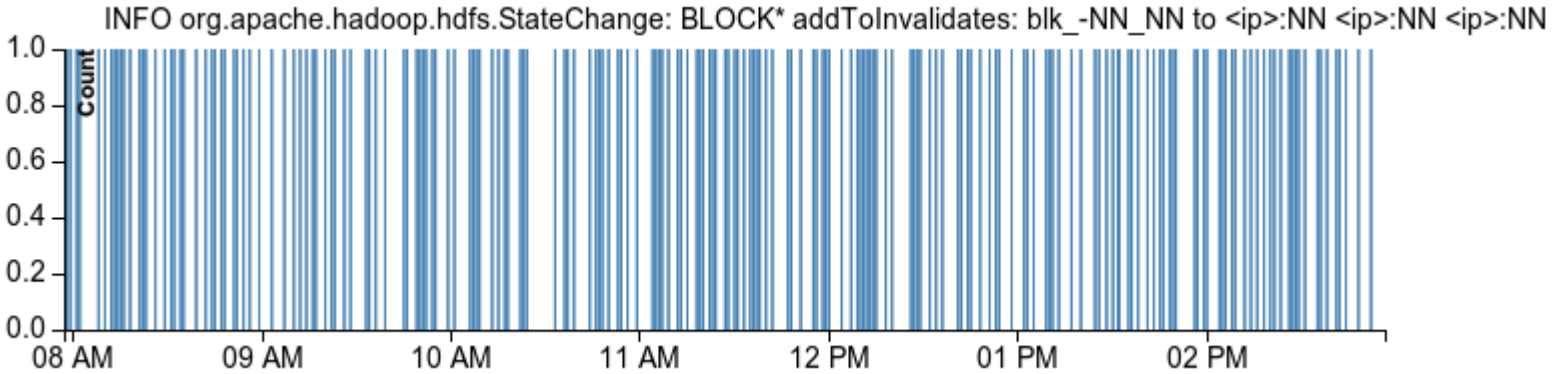
Why is this datanode different from the other datanodes?

When I look at logs, how I look at logs...

```
cat logs |
tr '[0-9]' N | # de-uniquify
sort |         # group...
uniq -c |      # count...
sort -n        # summarize
```

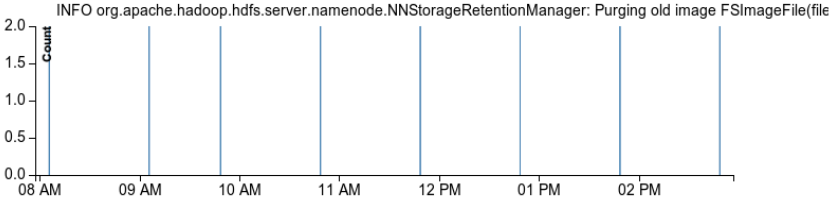
Leave "fancy clustering" for the data scientists.
Unix is good enough for us.

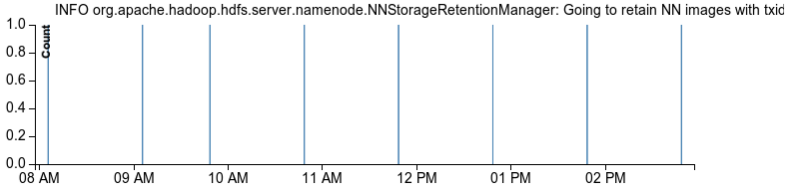
Is it boring?



This happens *all the time*. Ignore it. It's *log spam*.

Or maybe it's periodic? Does it explain any spikes?





git grep

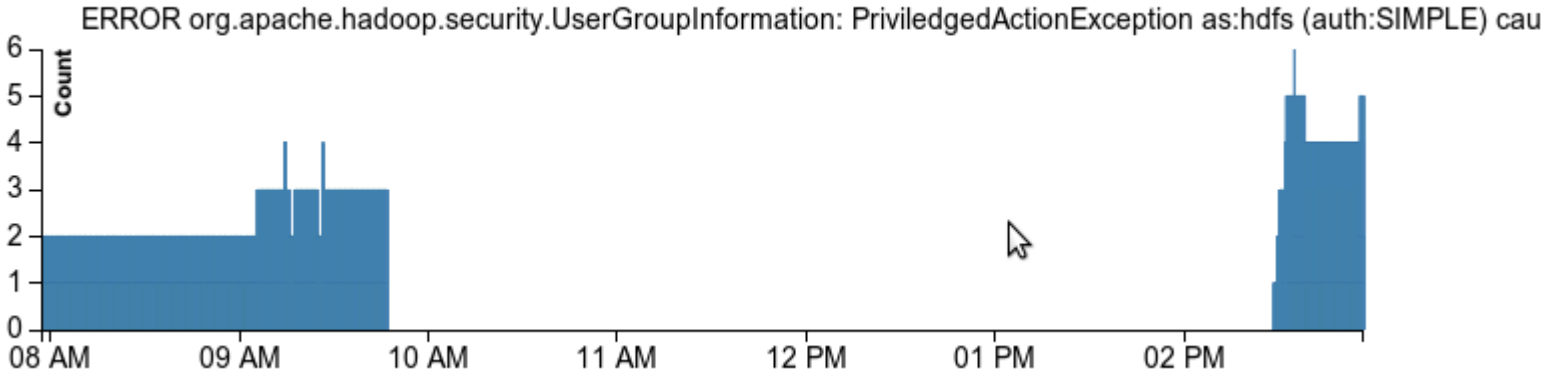
```
$find ~/src -maxdepth 2 -name
.git | wc -l
117
```

I have a ton of stuff checked out, ready for "git grep," from Hadoop to the JDK.

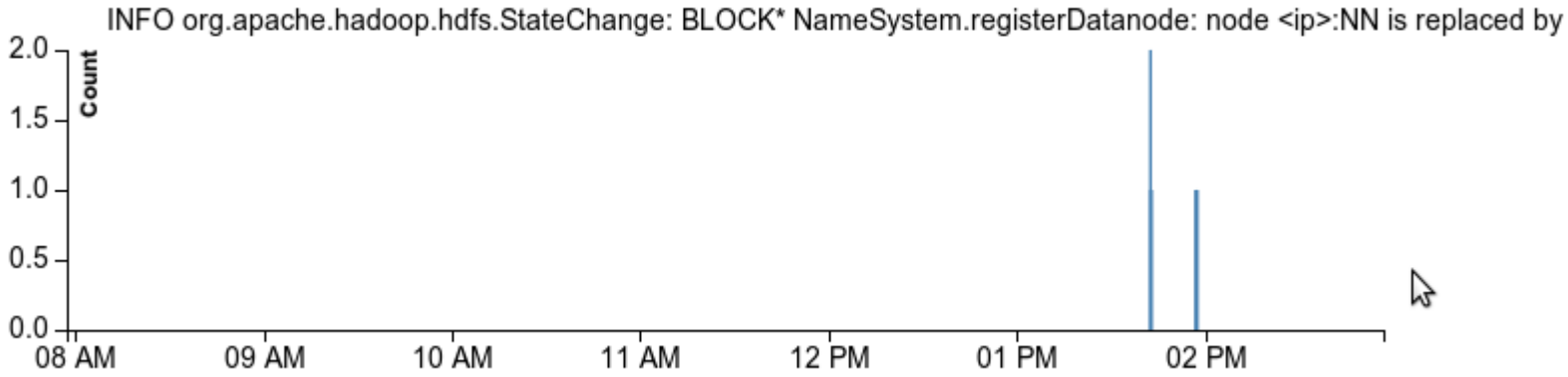
Great way to find those unclear log messages.

Did it come and go?

This is more interesting. What happened at 9:45 and then why did it re-start at 2:30?



Correlate with your problem period...



If you started experiencing a problem at 1:45, this might be interesting.

A few more tricks:

Focus on a specific time period. (Quick plug: Cloudera Manager lets you do this easily.)

If you see something you don't know about, see if it happens everywhere to see if it's boring.

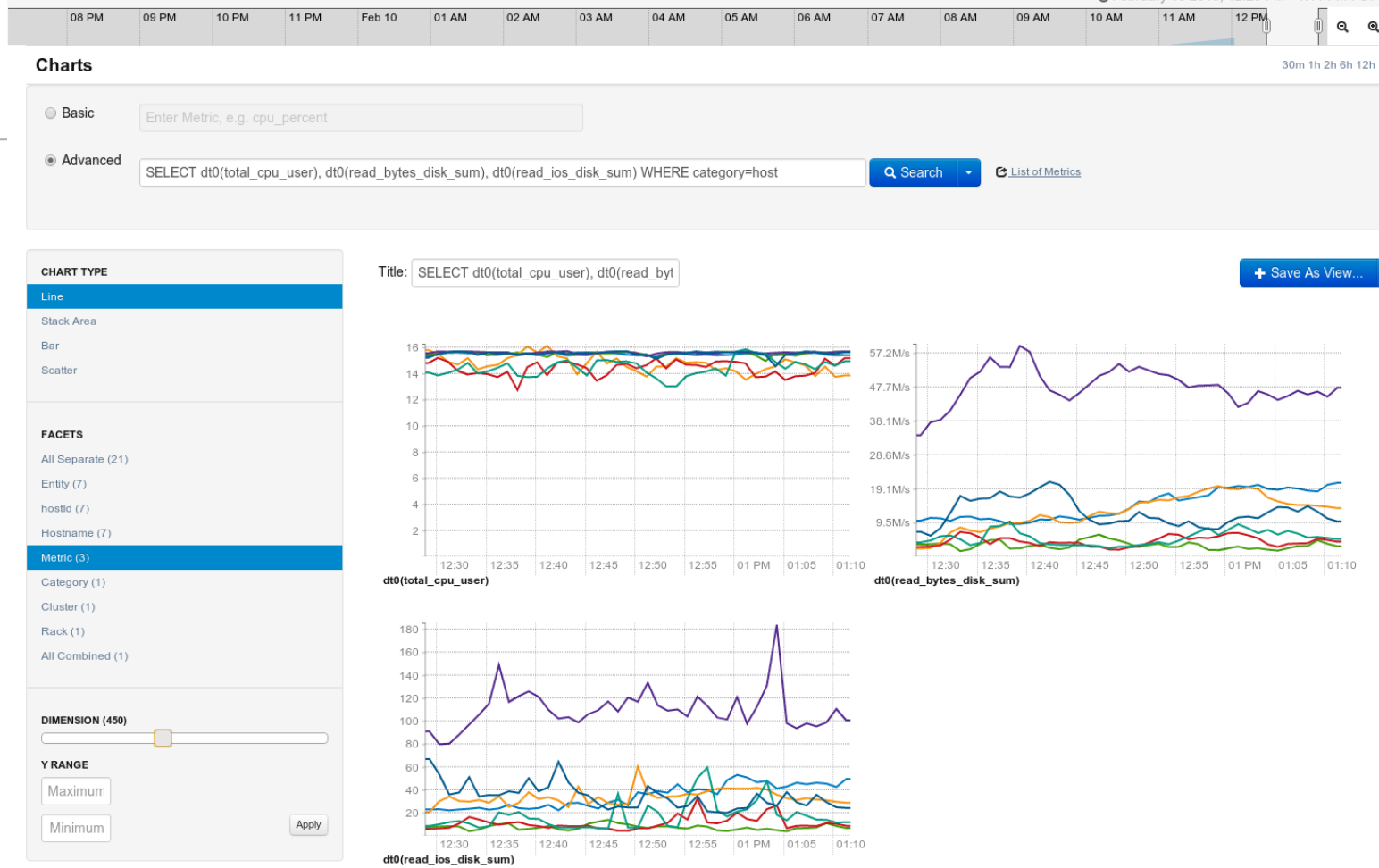
Think about it as a dataset. Logs are (host, process, time, message), organized in that order. Free yourself from that order, and access by message (histograms) or by time instead.

Colophon

http://philz.github.com/logvizjs/

The plots you saw today were produced with d3. See github repo for code.

Metrics


08 PM
09 PM
10 PM
11 PM
Feb 10
01 AM
02 AM
03 AM
04 AM
05 AM
06 AM
07 AM
08 AM
09 AM
10 AM
11 AM
12 PM
Charts
30m 1h 2h 6h 12h
Basic
Enter Metric, e.g. cpu_percent
Advanced
SELECT dt0(total_cpu_user), dt0(read_bytes_disk_sum), dt0(read_ios_disk_sum) WHERE category=host
Search
List of Metrics
CHART TYPE
Line
Stack Area
Bar
Scatter
FACETS
All Separate (21)
Entity (7)
hostId (7)
Hostname (7)
Metric (3)
Category (1)
Cluster (1)
Rack (1)
All Combined (1)
DIMENSION (450)
Y RANGE
Maximum
Minimum
Apply
Title:
SELECT dt0(total_cpu_user), dt0(read_byt
Save As View...
dt0(total_cpu_user)
dt0(read_bytes_disk_sum)
dt0(read_ios_disk_sum)

Queries are important

A distributed system with 100 hosts has ~50,000 individual time series.

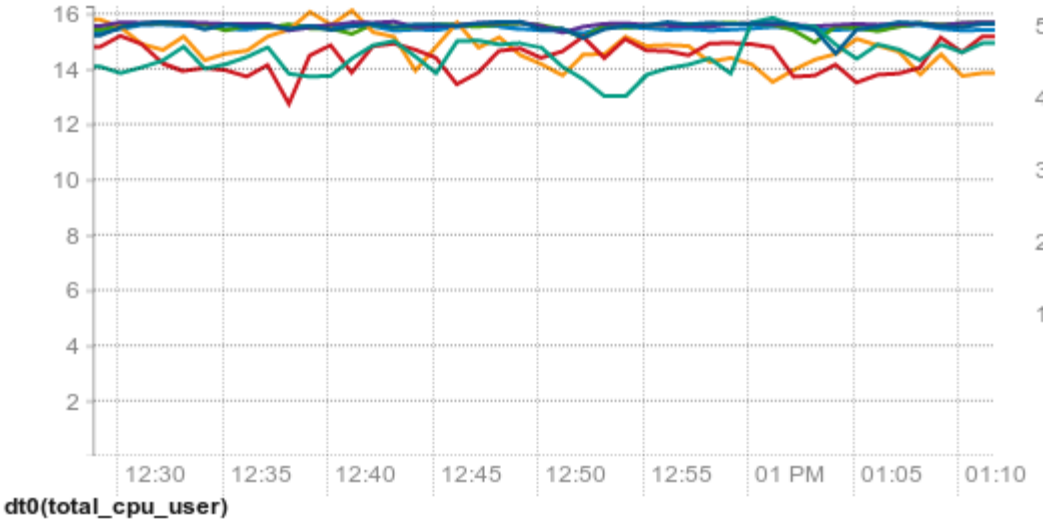
dvanced

SELECT dt0(total_cpu_user), dt0(read_bytes_disk_sum), dt0(read_ios_disk_sum) WHERE category=host

Search

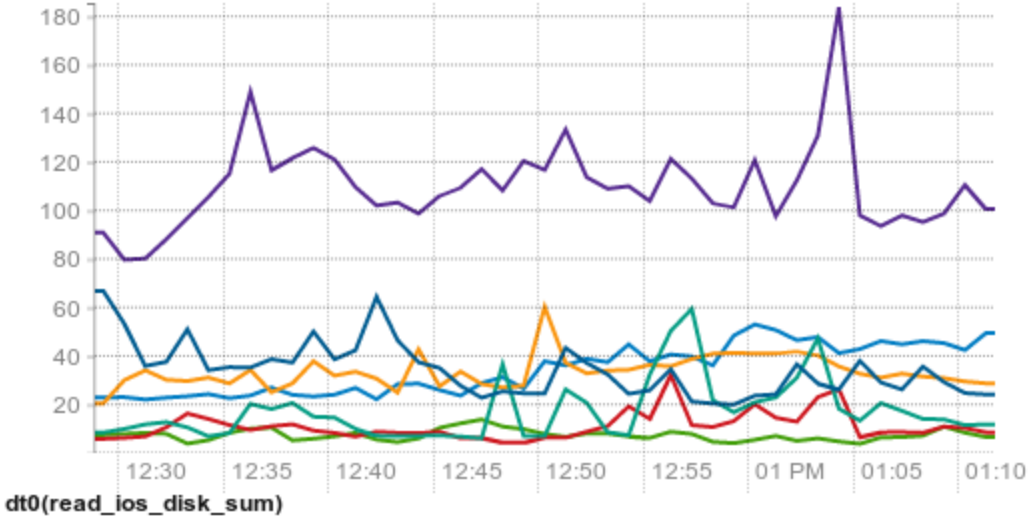
Look for outliers

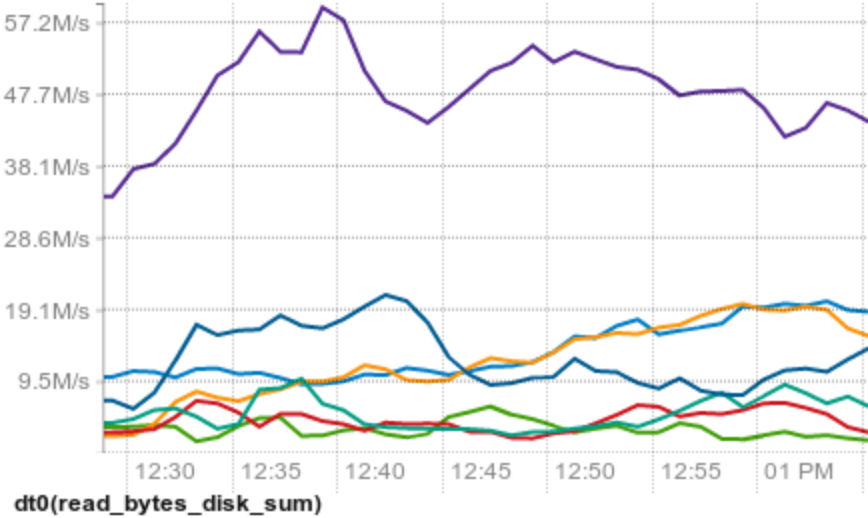
Nothing to see here;
all CPUs are pegged.



Outliers!

Why is one host pushing more IO than the other hosts?

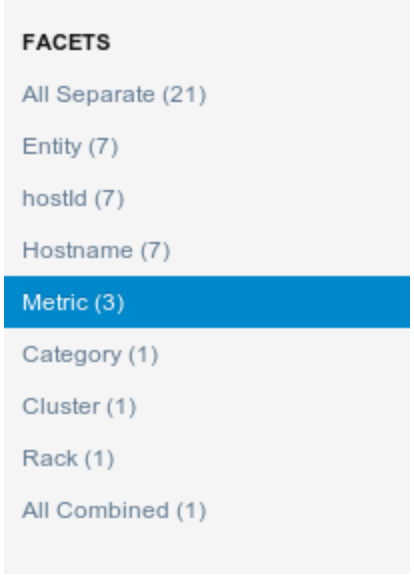




Faceting

Fancy name for "group by"

(See ggplot2, "grammar of graphics")



Tracing

- For systems that have this, it's amazing.
- Oddly harder to do in open source, because different layers are different projects, and there's a chicken-and-egg problem.

Google Dapper

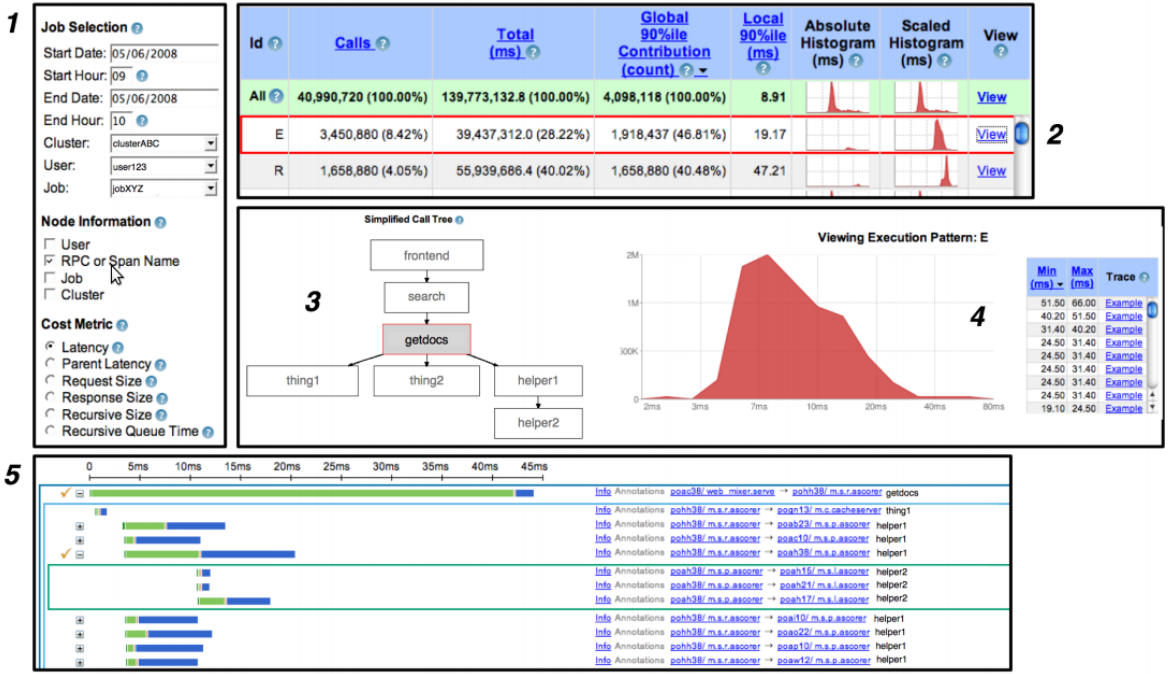


Figure 6: A typical user workflow in the general-purpose Dapper user interface.

Google AppEngine

Timeline | RPC Stats | CGI Environment | sys.path

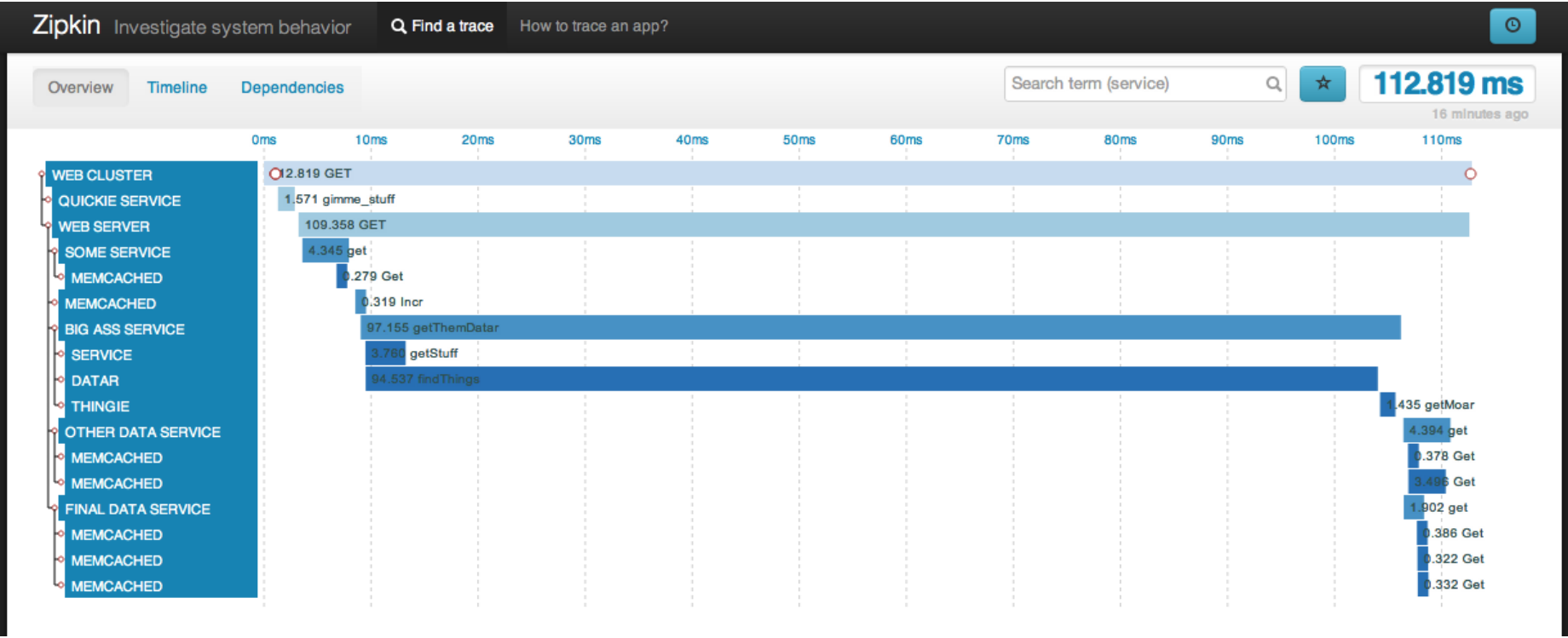
	0	500	1000	1500	2000	2500
urlfetch.Fetch	66ms					
datastore_v3.Put	217ms (245ms api)					
datastore_v3.Put	78ms (245ms api)					
datastore_v3.Put	60ms (245ms api)					
datastore_v3.Put	81ms (245ms api)					
datastore_v3.Put	74ms (245ms api)					
datastore_v3.Put	81ms (245ms api)					
datastore_v3.Put	73ms (245ms api)					
datastore_v3.Put	53ms (245ms api)					
datastore_v3.Put	81ms (245ms api)					
datastore_v3.Put	56ms (245ms api)					
mail.Send	45ms					
RPC Total	965ms (2450					
Grand Total	1361ms					

RPC Call Traces Expand All

RPC

- @0ms **urlfetch.Fetch** real=66ms api=0ms
- @70ms **datastore_v3.Put** real=217ms api=245ms
- @327ms **datastore_v3.Put** real=78ms api=245ms
- @443ms **datastore_v3.Put** real=60ms api=245ms
- @594ms **datastore_v3.Put** real=81ms api=245ms
- @701ms **datastore_v3.Put** real=74ms api=245ms
- @809ms **datastore_v3.Put** real=81ms api=245ms
- @928ms **datastore_v3.Put** real=73ms api=245ms
- @1020ms **datastore_v3.Put** real=53ms api=245ms
- @1098ms **datastore_v3.Put** real=81ms api=245ms
- @1239ms **datastore_v3.Put** real=56ms api=245ms
- @1315ms **mail.Send** real=45ms api=0ms

Twitter Zipkin



Slowly coming to Hadoop...

HTrace (https://github.com/cloudera/htrace)

HBASE-6449 introduces to HBase

Stay tuned!

Step 2: Digging in

Web UIs

Many distributed systems expose vital information over HTTP.

This is the Right Thing. Demand it!

Know what's available in your systems.

http://omel.ette.org/blog/2013/02/06/debug-servlets/

Configuration

Is it as expected?

Is it consistent across the cluster?

Where did that value come from anyway?

nightly-5.ent.cloudera.com:20101/conf

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<configuration>
  ▼<property>
     <name>mapreduce.job.ubertask.enable</name>
     <value>false</value>
     <source>mapred-default.xml</source>
   </property>
  ▼<property>
   ▼<name>
      yarn.resourcemanager.delayed.delegation-token.removal-interval-ms
     </name>
     <value>30000</value>
     <source>yarn-default.xml</source>
   </property>
  ▼<property>
     <name>yarn.resourcemanager.max-completed-applications</name>
     <value>10000</value>
     <source>yarn-default.xml</source>
   </property>
  ▼<property>
     <name>mapreduce.client.submit.file.replication</name>
     <value>10</value>
     <source>mapred-default.xml</source>
   </property>
  ▼<property>
     <name>yarn.nodemanager.container-manager.thread-count</name>
```

Stack Traces

Is it deadlocked?

Is it blocked on an external resource (e.g., a database)?

What's going on?

theseus02.sf.cloudera.com:50070/stacks

```
Process Thread Dump:
63 active threads
Thread 79 (359026458@qtp-1454691228-3):
  State: RUNNABLE
  Blocked count: 3
  Waited count: 3
  Stack:
    sun.management.ThreadImpl.getThreadInfo1(Native Method)
    sun.management.ThreadImpl.getThreadInfo(ThreadImpl.java:156)
    sun.management.ThreadImpl.getThreadInfo(ThreadImpl.java:121)
    org.apache.hadoop.util.ReflectionUtils.printThreadInfo(Reflec
    org.apache.hadoop.http.HttpServer$StackServlet.doGet(HttpServ
    javax.servlet.http.HttpServlet.service(HttpServlet.java:707)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
    org.mortbay.jetty.servlet.ServletHolder.handle(ServletHolder.
    org.mortbay.jetty.servlet.ServletHandler$CachedChain.doFilter
    org.apache.hadoop.http.lib.StaticUserWebFilter$StaticUserFilt
    org.mortbay.jetty.servlet.ServletHandler$CachedChain.doFilter
    org.apache.hadoop.http.HttpServer$QuotingInputFilter.doFilter
    org.mortbay.jetty.servlet.ServletHandler$CachedChain.doFilter
    org.mortbay.jetty.servlet.ServletHandler.handle(ServletHandle
    org.mortbay.jetty.security.SecurityHandler.handle(SecurityHan
    org.mortbay.jetty.servlet.SessionHandler.handle(SessionHandle
    org.mortbay.jetty.handler.ContextHandler.handle(ContextHandle
    org.mortbay.jetty.webapp.WebAppContext.handle(WebAppContext.j
    org.mortbay.jetty.handler.ContextHandlerCollection.handle(Con
    org.mortbay.jetty.handler.HandlerWrapper.handle(HandlerWrappe
Thread 77 (Trash Emptier):
  State: TIMED_WAITING
  Blocked count: 0
  Waited count: 1
  Stack:
    java.lang.Thread.sleep(Native Method)
```

Application Status Pages

Are nodes missing?

What's the version of the software?

Master: p0427.mtv.cloudera.com:60000

Local logs, Thread Dump, Log Level, Debug dump

Attributes

Attribute Name	Value	Description
HBase Version	0.92.1-cdh4.1.2, rUnknown	HBase version and revision
HBase Compiled	Thu Nov 1 18:01:09 PDT 2012, jenkins	When HBase version was compiled and by whom
Hadoop Version	2.0.0-cdh4.1.2, rf0b53c81cbf56f5955e403b49fcd27afd5f082dc	Hadoop version and revision
Hadoop Compiled	Thu Nov 1 17:33:23 PDT 2012, jenkins	When Hadoop version was compiled and by whom
HBase Root Directory	hdfs://p0428.mtv.cloudera.com:8020/hbase	Location of HBase home directory
HBase Cluster ID	3ddb2ed4-71a0-44a6-a39a-9fea811f69b9	Unique identifier generated for each HBase cluster
Load average	0.29	Average number of regions per regionserver. Naive computation.
Zookeeper Quorum	p0427.mtv.cloudera.com:2181	Addresses of all registered ZK servers. For more, see zk dump.
Coprocessors	[MasterAuditCoProcessor]	Coprocessors currently loaded loaded by the master
HMaster Start Time	Sat Jan 12 17:31:27 PST 2013	Date stamp of when this HMaster was started
HMaster Active Time	Sat Jan 12 17:31:27 PST 2013	Date stamp of when this HMaster became active

Tasks

Show All Monitored Tasks Show non-RPC Tasks Show All RPC Handler Tasks Show Active RPC Calls Show Client Operations View as JSON

Start Time	Description	State	Status
Sat Jan 12 17:31:27 PST 2013	*REPL IPC Server handler 2 on 60000*	*WAITING (since 4mins, 15sec ago)*	*Waiting for a call (since 4mins, 15sec ago)*
Sat Jan 12 17:31:27 PST 2013	*REPL IPC Server handler 1 on 60000*	*WAITING (since 4mins, 15sec ago)*	*Waiting for a call (since 4mins, 15sec ago)*
Sat Jan 12 17:31:27 PST 2013	*REPL IPC Server handler 0 on 60000*	*WAITING (since 4mins, 15sec ago)*	*Waiting for a call (since 4mins, 15sec ago)*
Sat Jan 12 17:31:27 PST 2013	*IPC Server handler 9 on 60000*	*WAITING (since 4sec ago)*	*Waiting for a call (since 4sec ago)*
Sat Jan 12 17:31:27 PST 2013	*IPC Server handler 8 on 60000*	*WAITING (since 0sec ago)*	*Waiting for a call (since 0sec ago)*
Sat Jan 12 17:31:27 PST 2013	*IPC Server handler 7 on 60000*	*WAITING (since 1sec ago)*	*Waiting for a call (since 1sec ago)*
Sat Jan 12 17:31:27 PST 2013	*IPC Server handler 6 on 60000*	*WAITING (since 4sec ago)*	*Waiting for a call (since 4sec ago)*
Sat Jan 12 17:31:27 PST 2013	*IPC Server handler 5 on 60000*	*WAITING (since 1sec ago)*	*Waiting for a call (since 1sec ago)*
Sat Jan 12 17:31:27 PST 2013	*IPC Server handler 4 on 60000*	*WAITING (since 1sec ago)*	*Waiting for a call (since 1sec ago)*
Sat Jan 12 17:31:27 PST 2013	*IPC Server handler 3 on 60000*	*WAITING (since 1sec ago)*	*Waiting for a call (since 1sec ago)*
Sat Jan 12 17:31:27 PST 2013	*IPC Server handler 2 on 60000*	*WAITING (since 1sec ago)*	*Waiting for a call (since 1sec ago)*
Sat Jan 12 17:31:27 PST 2013	*IPC Server handler 1 on 60000*	*WAITING (since 1sec ago)*	*Waiting for a call (since 1sec ago)*
Sat Jan 12 17:31:27 PST 2013	*IPC Server handler 0 on 60000*	*WAITING (since 1sec ago)*	*Waiting for a call (since 1sec ago)*

Tables

Catalog Table	Description

What else shows up?

Logs

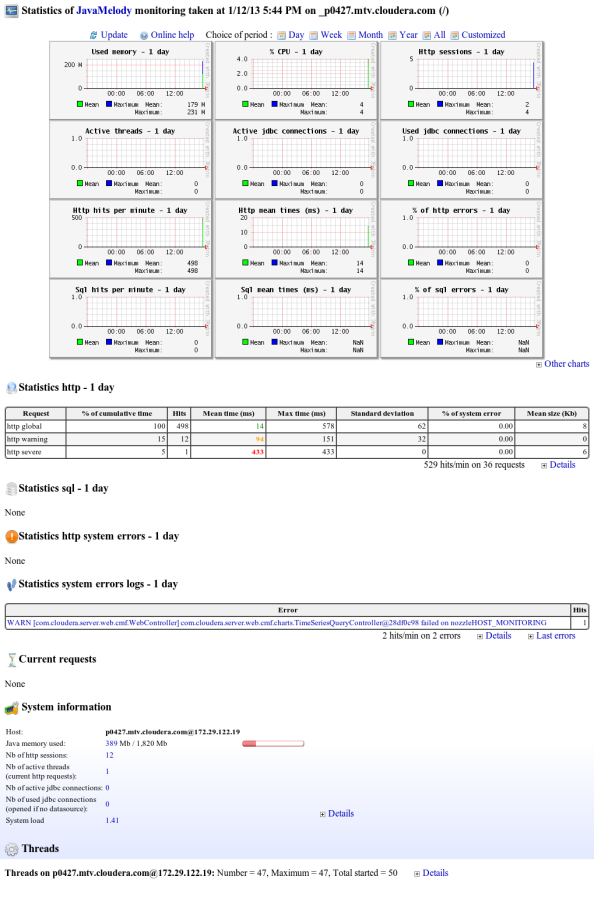
Log configurations

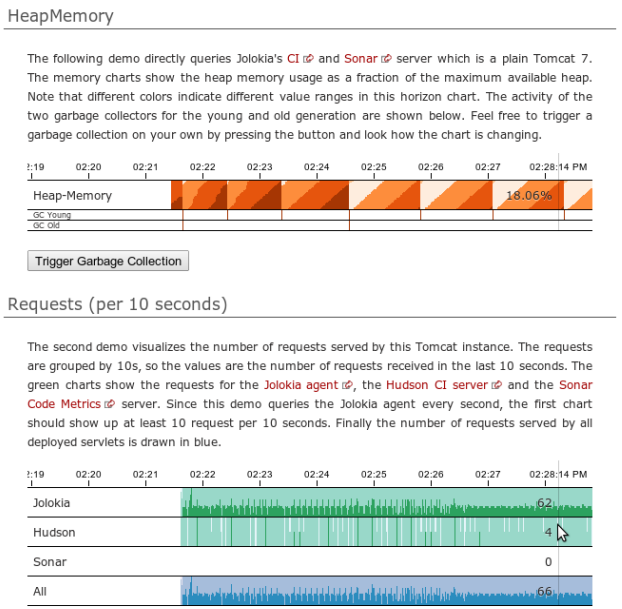
Metrics (and JMX)

Developers have no excuses!

Several pre-built solutions exist for these debug UIs. (JavaMelody, Jolokia)

Demand them!





Linux Toolbelt

- top: what's running; is it eating CPU?
- iotop: what's eating disk
- ps: what's running? with what options? in what dirs?
- lsof -P -n -p <pid>: what's reading what files? what has what ports open?
- /proc has lots of goodies

Linux Hammer: strace

All interesting things happen through system calls: reading and writing, RPCs, etc.

How else could I have known that /etc/resolv.conf had bad permissions? Ouch.

```
futex(0x7f572886b6d0, FUTEX_WAKE_PRIVATE, 2147483647) = 0
futex(0x7f572886b800, FUTEX_WAKE_PRIVATE, 2147483647) = 0
futex(0x7f5729025144, FUTEX_WAKE_PRIVATE, 2147483647) = 0
open("/usr/lib/ssl/openssl.cnf", O_RDONLY) = 6
fstat(6, {st_mode=S_IFREG|0644, st_size=10835, ...}) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f5729458000
read(6, "#\n# OpenSSL example configuratio"..., 4096) = 4096
read(6, "Netscape crash on BMPStrings or "..., 4096) = 4096
read(6, " this to avoid interpreting an e"..., 4096) = 2643
read(6, "", 4096)                          = 0
close(6)                                   = 0
munmap(0x7f5729458000, 4096)               = 0
futex(0x7f5726efc0d0, FUTEX_WAKE_PRIVATE, 2147483647) = 0
open("/usr/lib/x86_64-linux-gnu/openssl-1.0.0/engines/libgost.so", O_RDONLY|O_CLOEXEC) = 6
read(6, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\320V\0\0\0\0\0\0"..., 832) = 832
fstat(6, {st_mode=S_IFREG|0644, st_size=89080, ...}) = 0
mmap(NULL, 2184224, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 6, 0) = 0x7f5724ca9000
mprotect(0x7f5724cbd000, 2093056, PROT_NONE) = 0
mmap(0x7f5724ebc000, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 6, 0x13000) = 0x7f5724ebc000
close(6)                                   = 0
mprotect(0x7f5724ebc000, 4096, PROT_READ) = 0
open("/usr/share/locale/en_US.UTF-8/libdns.cat", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/usr/share/locale/en_US.UTF-8/LC_MESSAGES/libdns.cat", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/usr/share/locale/en/libdns.cat", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/usr/share/locale/en/LC_MESSAGES/libdns.cat", O_RDONLY) = -1 ENOENT (No such file or directory)
futex(0x7f57290247ac, FUTEX_WAKE_PRIVATE, 2147483647) = 0
futex(0x7f5729024864, FUTEX_WAKE_PRIVATE, 2147483647) = 0
open("/etc/resolv.conf", O_RDONLY)         = -1 EACCES (Permission denied)
```

Quick tour of the JRE

Listing running JVMs

```
$sudo /usr/java/jdk1.6.0_31/bin/jps -l
23675 org.apache.hadoop.hdfs.server.datanode.DataNode
23855 org.apache.hadoop.mapred.TaskTracker
32196 sun.tools.jps.Jps
24645
```

Looking at stack traces of a running JVM

```
$sudo -u hdfs /usr/java/jdk1.6.0_31/bin/jstack 23675 | head
2013-02-20 11:13:35
Full thread dump Java HotSpot(TM) 64-Bit Server                    de):

"Async disk worker #55 for volume /data/4/dfs2/dn/current" daemon prio=10
 tid=0x00007fa4c4bac800 nid=0x7f94 waiting on condition [0x00007fa4aae58000]
   java.lang.Thread.State: TIMED_WAITING (parking)
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for  <0x00000000c3103c90> (a j                    s.
 AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.parkNa                    )
    at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.
 awaitNanos(AbstractQueuedSynchronizer.java:2025)
    at java.util.concurrent.LinkedBlockingQueue.poll(LinkedBlockingQueue.java:424)
```

nid is the Linux thread id in hex; can be used with top H

locks taken and locks blocked are exposed; deadlocks are spottable this way.

Poor Man's Profiling

Being able to get a set of stack traces is enough to build a cheapo sampling profiler.

```
$cat bin/jpmp.sh
#!/bin/bash
# Original version: http://blog.tsunanet.net/2010/08/jpmp-javas-poor-mans-profiler.html
# Usage: ./jpmp.sh <pid> <num-samples> <sleep-time-between-samples>
pid=$1; nsamples=$2; sleeptime=$3
for x in $(seq 1 $nsamples)
  do
    jstack $pid
    sleep $sleeptime
  done | \
awk 'BEGIN { s = "" } \
/^"/ { if (s) print s; s = "" } \
/^   at / { sub(/\([^)]*\)?$/, "", $2); sub(/^java/, "j", $2); if (s) s = s "," $2; else s
= $2 } \
END { if(s) print s }' | \
sort | uniq -c | sort -rnk1
```

Memory Issues

Sometimes, you might have Garbage Collection issues. Look for high CPU. Fortunately, there is instrumentation, that you can turn on at runtime! Also, check out 'jstat'

```
$sudo -u mapred /usr/java/jdk1.6.0_31/bin/jinfo -flag +PrintGC  18311
$sudo -u mapred /usr/java/jdk1.6.0_31/bin/jinfo -flag +PrintGCTimeStamps  18311
$sudo -u mapred /usr/java/jdk1.6.0_31/bin/jinfo -flag +PrintGCDetails  18311
$sudo tail -f /proc/18311/cwd/logs/stdout.log
63237.523: [GC 63237.539: [ParNew: 18233K->350K(19136K), 0.0015310 secs] 54470K->36722K(83008K),
 0.0016710 secs] [Times: user=0.01 sys=0.00, real=0.01 secs]
63257.848: [GC 63257.848: [ParNew: 17374K->1710K(19136K), 0.0034400 secs] 53746K->38083K(83008K),
 0.0035460 secs] [Times: user=0.03 sys=0.00, real=0.00 secs]
63262.539: [GC 63262.539: [ParNew: 18360K->948K(19136K), 0.0033630 secs] 54733K->38542K(83008K),
 0.0034860 secs] [Times: user=0.02 sys=0.01, real=0.00 secs]
63273.979: [GC 63273.979: [ParNew: 17972K->809K(19136K), 0.0014940 secs] 55566K->38404K(83008K),
 0.0015880 secs] [Times: user=0.01 sys=0.00, real=0.00 secs]
```

More Unholy JVM Tricks

- Using 'jmap' to dump the heap; use Eclipse MAT to read the state to reason about it.
- Using the fact that JSP can be compiled at runtime to insert code into a running process.
- Using the 'instrumentation' API to inject code. BTrace is a system for doing so.
- Grabbing JMX metrics from a running process even if hasn't exposed them (Jolokia, https://github.com/philz/jvm-tools)

Quick Review: My Bag of Tricks

Zoomed Out

Logs: Outliers, Clustering, Visualizing

Metrics

Tracing

Zoomed In

HTTP-Based Debug Pages

Linux Introspection

JVM Introspection

Thanks!

Office Hours:
10:10am Thursday Expo Hall (Table B)
philip@cloudera.com @philz42

Cloudera Booth #701
HBaseCon: June 13, 2013


cloudera®
Ask Bigger Questions