



Partial Failure

Why is it? []

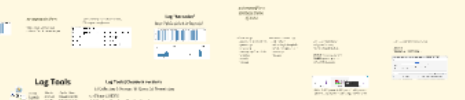
Why is it? []

If you are debugging, not monitoring. Monitor, check to see if you have some key across all of your clusters.

Breadth



Logs



Log Tools



Metrics Plot and Compare



Tracing

Outlier Grand Finale: Query all the things!

ask questions of several hundred nodes



Something I'm working on...

Scylla: All "Information is Power" for systems



Depth

you've found a problematic machine/process, now what?

status pages
Linux
Java

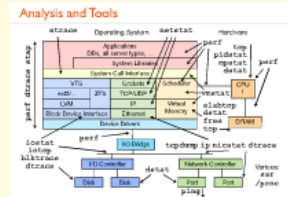


status pages for Hadoop: /jmx, /conf, /stacks no excuses for developers huzzah for HTTP

Don't leave the house without...

strace did I mention strace?
jstack
pstack
ps/top/htop

@brendangregg's sweet slide



open source is... open source!
git grep (et al.) are undervalued
I have the source of pretty much every bit of the stack locally.

Java stuff...
jps
jstack you can also do unholy things with "agents"; instrument running code, etc.
jmap
jinfo
-Xoblog -XX:+HeapDumpOnOutOfMemoryError -Djava.net.preferIPv4Stack=true
attaching debuggers remotely

[the end]
philip@cloudera.com @phil42

review start broadly to find outliers
invest in monitoring/canaries: find simpler cases that fail
tool up
dig in with "traditional" tools
focus on communication paths

Debugging Distributed Systems

Philip Zeyliger

@philz42

philip@cloudera.com



@pmitz42

philip@cloudera.com


Why is it hard?

To mix metaphors,
picking out the black sheep
is more like "Find Waldo"

Layers

Networks

Partial Failure



Partial
Failure

inversion

from many services on a host
to many hosts for a service

The Generic Approach

Monitoring

Preconditions

Canaries

Metrics

Breadth

Findings outliers

Logs

Metrics

Tracing

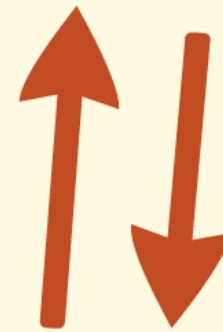
Depth

Code Inspection

JVM Inspection

OS Debugging

Find Waldo (=outlier)



If you see something,
say something.

Rather, check to see if
it's the same way across all of your
cluster.

Monitoring

Preconditions

Inspector Results

Validations

- ✓ Inspector ran on all 33 nodes
- ✓ Individual nodes received their own HDFS names copy
- ✓ No errors were had or files looking for conflicting file exists
- ✓ No errors were had or files in pending state
- ✓ All nodes received lockdowns (FSLock)
- ✓ All HDFS checks received back others' HDFS names correctly
- ✓ Host checks are approximately 1/2 to 3/4 in progress
- ✓ Host file system are consistent across the cluster
- ✓ All namenode storage are running
- ✓ No bad nodes that are known to be bad are running
- ✓ No performance concerns with the current HDFS NameNode

1 error are reported with 100% OK success

1 error are reported with 100% OK success

All managed nodes have consistent network name

All checked Cluster Management Operations are consistent with the server

All checked Cluster Management Operations are consistent with the server

Health Tests

HDFS Summary October 23 2013, 1:52:31 PM PDT

Configured Capacity 20.4 TB (4.1 TB)

NameNode Web UI [at 1001.hadoop.cloudera.com \(Active\)](#)

Event Search [Alerts](#), [Critical](#), [All](#)

Reports [View](#), [Browse Filesystem](#)

Health Tests [Collapse All](#)

4,081 missing blocks in the cluster. 1,061,277 total blocks in the cluster. Percentage missing blocks: 0.38%. Critical threshold: any.

The active NameNode's health is concerning.

5 good.

Space free in the cluster: 1.1 PB. Capacity of the cluster: 1.4 PB. Percentage of capacity free: 83.04%.

0 blocks with corrupt replicas in the cluster. 1,061,577 total blocks in the cluster. Percentage blocks with corrupt replicas: 0.00%.

4,081 under replicated blocks in the cluster. 1,061,577 total blocks in the cluster. Percentage under replicated blocks: 0.38%.

Canary test of file create, write, read and delete operations succeeded.

Healthy DataNode: 71. Concerning DataNode: 1. Total DataNode: 72. Percent healthy: 98.61%. Percent healthy or concerning: 100.00%.

Test disabled because HDFS is not configured with a second NameNode: Test of whether there is a running, healthy, standby NameNode.

Canaries

while true:

open()

write()

read()

sleep()



Preconditions

Inspector Results

Validations

- ✓ Inspector ran on all 80 hosts.
- ✓ Individual hosts resolved their own hostnames correctly.
- ✓ 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.
- ✓ No performance concerns with Transparent Huge Pages settings.
- ⚠ 1 hosts are reporting with NONE CDH version
- ⚠ There are mismatched versions across the system. See details below for details on which hosts are running what versions of compor
- ✓ All managed hosts have consistent versions of Java.
- ✓ All checked Cloudera Management Daemons versions are consistent with the server.
- ✓ All checked Cloudera Management Agents versions are consistent with the server.

Health Tests

HDFS Summary October 23 2013, 1:53:31 PM PDT

Configured Capacity	<div style="width: 20%; background-color: green; border: 1px solid black;"></div> 285.4 TiB/1.4 PiB
NameNode Web UI	e1101.halxg.cloudera.com (Active)
Event Search	Alerts , Critical , All
Reports	View , Browse Filesystem

Health Tests [Collapse All](#)

4,081 missing blocks in the cluster. 1,061,577 total blocks in the cluster. Percentage missing blocks: 0.38%. Critical threshold: any.

The active NameNode's health is concerning.

▼ 5 good.

Space free in the cluster: 1.1 PiB. Capacity of the cluster: 1.4 PiB. Percentage of capacity free: 80.04% ⌵

0 blocks with corrupt replicas in the cluster. 1,061,577 total blocks in the cluster. Percentage blocks with corrupt replicas: 0.00%.

4,081 under replicated blocks in the cluster. 1,061,577 total blocks in the cluster. Percentage under replicated blocks: 0.38%.

Canary test of file create, write, read and delete operations succeeded.

Healthy DataNode: 71. Concerning DataNode: 1. Total DataNode: 72. Percent healthy: 98.61%. Percent healthy or concerning: 100.00%.

Test disabled because HDFS is not configured with a second NameNode: Test of whether there is a running, healthy, standby NameNode.

Canaries

while true:

open()

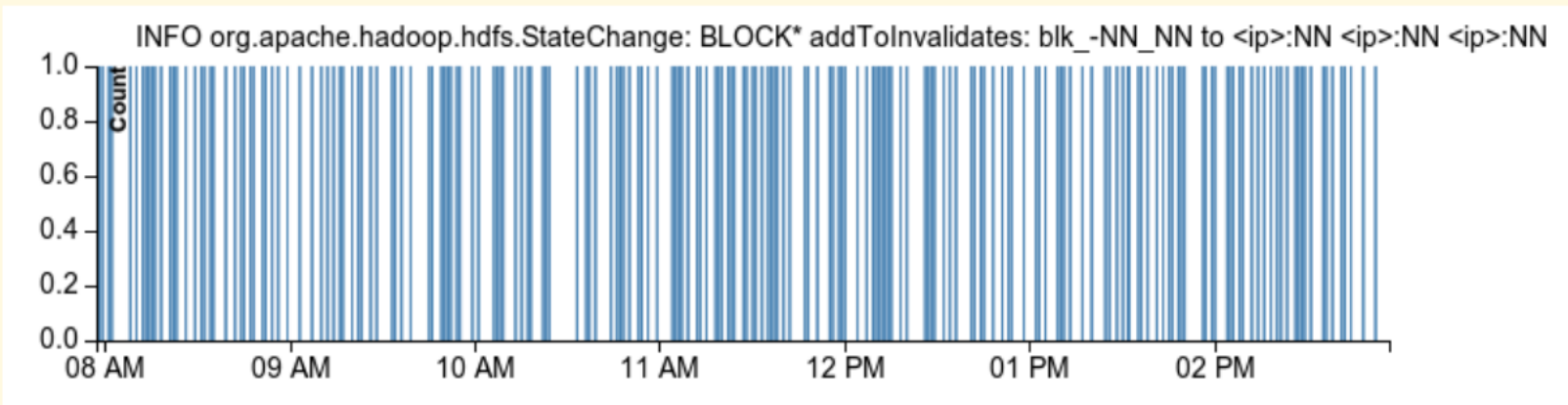
write()

read()

sleep()



Logs



... are full of lies

so aggregate them

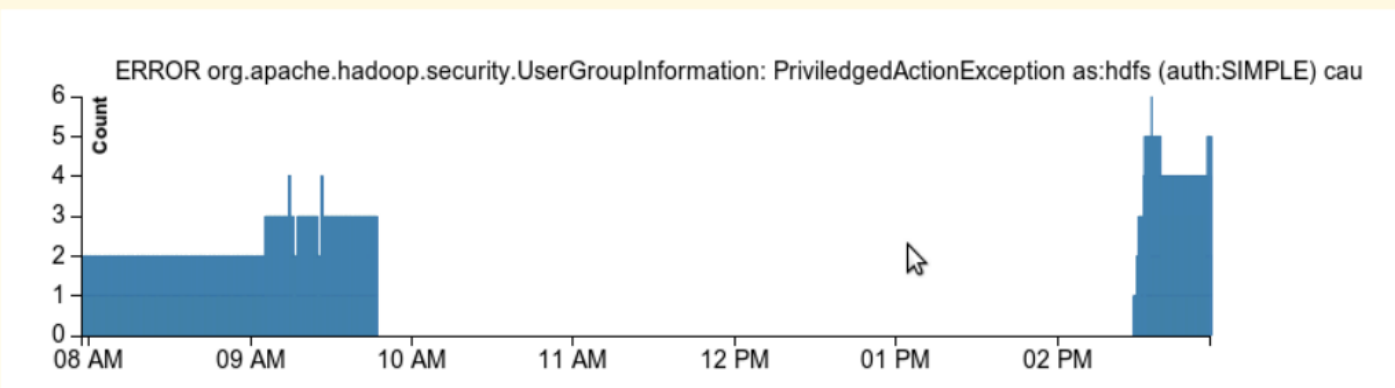
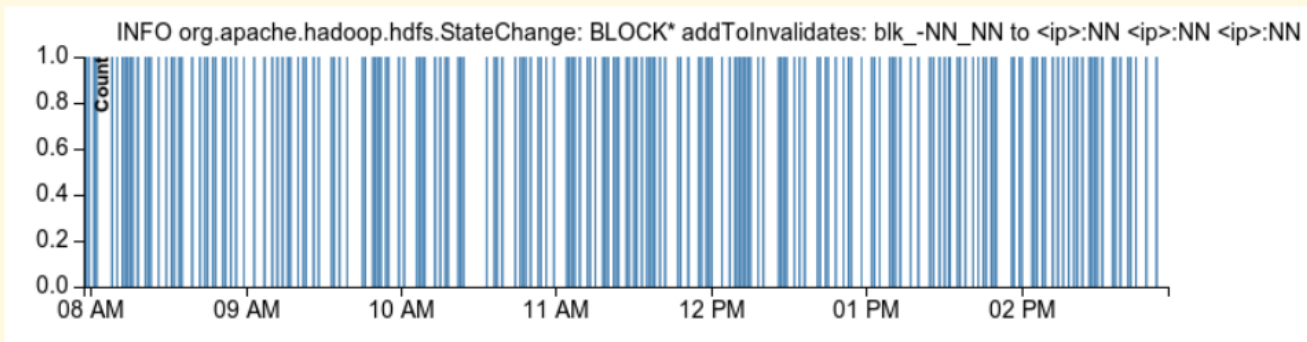
Think of logs less like a story,
and more like a dataset for analysis.

Leave "clustering" for the data scientists,
UNIX is good enough for us!

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

Log "Barcodes"

<http://philz.github.io/logvizjs/>



*put yourself in a
database frame
of mind*

Schema of Logs

- process instance (machine)
- process type
- timestamp
- message type (line of code)
- grouping
- severity
- message

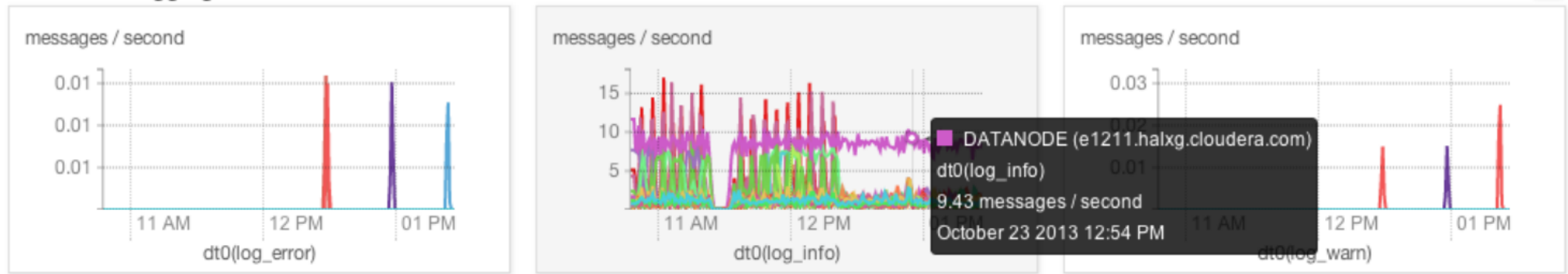
Traditional Access to Logs

- ssh machine
- `cd /var/log/hadoop-hdfs`
- `vi $(ls -rt *.log | tail -n 1)`
- timestamp
- message

e.g.: Look at distribution
of log sizes (or rates),
instead of logs themselves.

```
SELECT hostname, count(*)  
WHERE process_type = ...  
GROUP BY hostname
```

Datanode Logging Rates



```
SELECT dt0(log_warn), dt0(log_info), dt0(log_error)
WHERE category=ROLE and roletype=DATANODE
```

e.g.: Look during a problem period

SELECT *

WHERE date BETWEEN ...

October 23 2013, 10:47:45 AM - 1:26:30 PM PDT

04 PM 05 PM 06 PM 07 PM 08 PM 09 PM 10 PM 11 PM Wed 23 01 AM 02 AM 03 AM 04 AM 05 AM 06 AM 07 AM 08 AM 09 AM 10 AM 11 AM 12 PM 01 PM 02 PM

Logs

30m 1h 2h 6h 12h 1d

Enter Search Phrase or Regular Expression [Additional Settings](#)

Select Sources Hosts Minimum Log Level

All Sources

Cluster 1 - CDH 5

- hbase1
- hdfs1
- hive1
- hue1
- impala1
- ks_indexer1
- mapreduce1
- oozie1
- solr1
- sqoop1
- yarn
- zookeeper1

Cloudera Management Services

- mgmt1

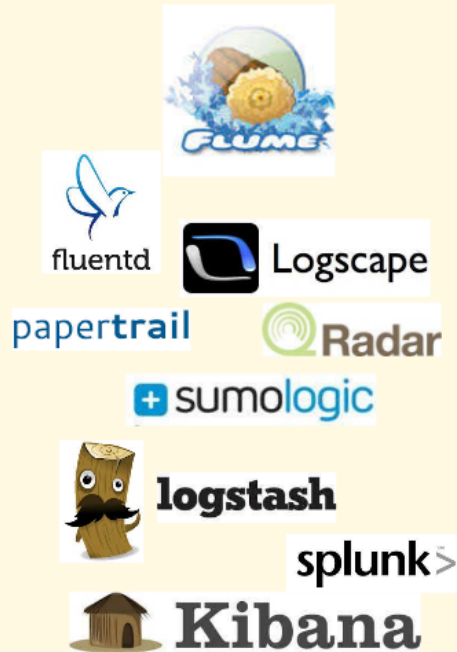
Cloudera Manager

- Cloudera Manager

Previous Next Machines Searched: 1, Errors: 0, Search Time: 110 ms, [More Statistics](#)

Host	Log Level	Time	Source	Message
e1211.halxg.cloudera.com	INFO	October 23 2013 10:47 AM	DataNode	Receiving BP-721828588-10.20.126.101-1378407836291:blk_1073780569_1099514496592 src: /10.20.128.111:52232 dest: /10.20.128.111:50010 View Log File

Log Tools



Splunk

Fluentd

Apache Flume

Logstash

Graylog2

Cloudera Search

Kibana

Loggly

Cloudera Impala

Logster

Scalyr

PaperTrail

LogScape

SumoLogic

QRadar

Arcsight

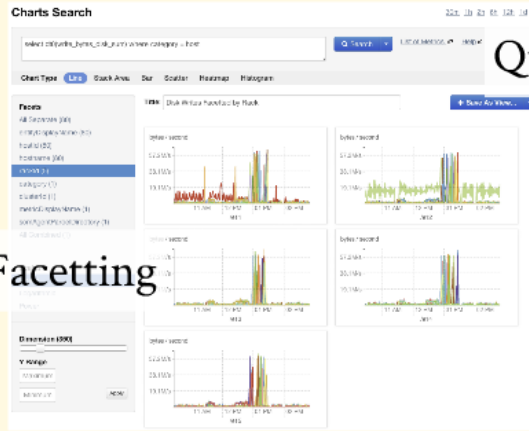
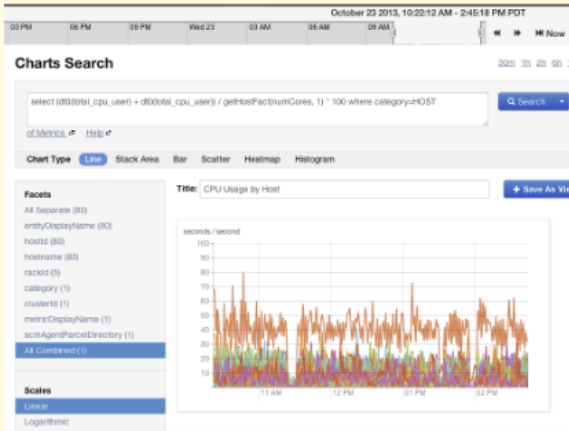
Log Tools (Cloudera version)

(a) Collection & Storage, (b) Query, (c) Presentation

(a) Flume & HDFS

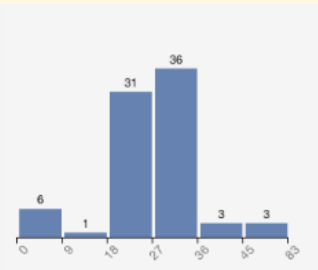
(b) Cloudera Search or Cloudera Impala

(c) Hue

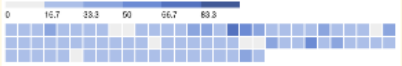


Queryability!

Facetting



a good read: "grammar of gra



Metrics

Plot and Compare

Charts Search

[30m](#) [1h](#) [2h](#) [6h](#) [1d](#)

```
select (dt0(total_cpu_user) + dt0(total_cpu_user)) / getHostFact(numCores, 1) * 100 where category=HOST
```

Search

[of Metrics](#) [Help](#)

Chart Type **Line** Stack Area Bar Scatter Heatmap Histogram

Facets

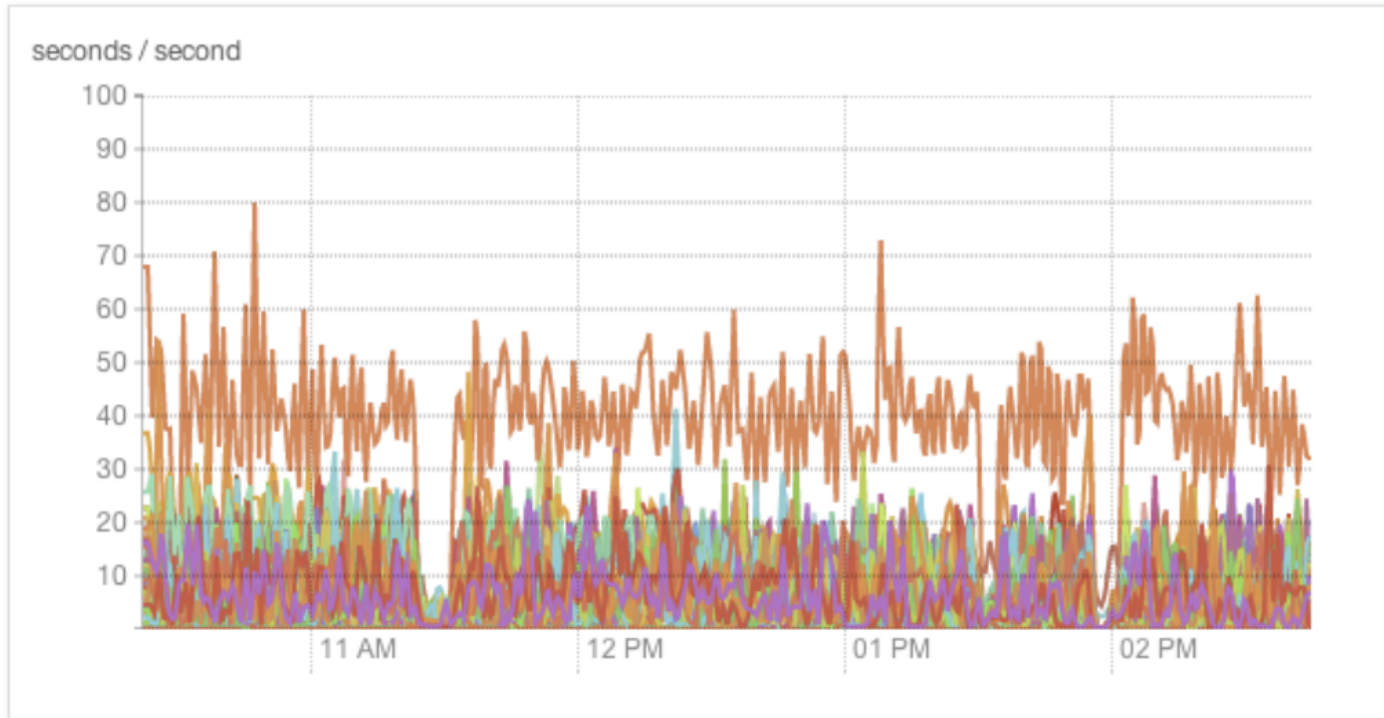
- All Separate (80)
- entityDisplayName (80)
- hostId (80)
- hostname (80)
- rackId (5)
- category (1)
- clusterId (1)
- metricDisplayName (1)
- scmAgentParcelDirectory (1)
- All Combined (1)**

Scales

- Linear**
- Logarithmic

Title: CPU Usage by Host

Save As View



Charts Search

30m 1h 2h 6h 12h 1d

select dt0(write_bytes_disk_sum) where category = host

Search

List of Metrics Help

Chart Type **Line** Stack Area Bar Scatter Heatmap Histogram

Facets

- All Separate (80)
- entityDisplayName (80)
- hostId (80)
- hostname (80)
- rackId (5)**
- category (1)
- clusterId (1)
- metricDisplayName (1)
- scmAgentParcelDirectory (1)
- All Combined (1)

Facetting

Dimension (350)

Y Range

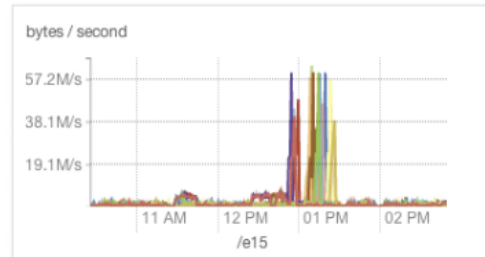
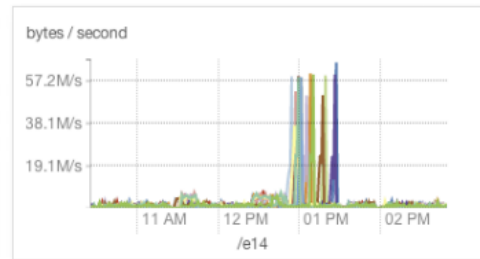
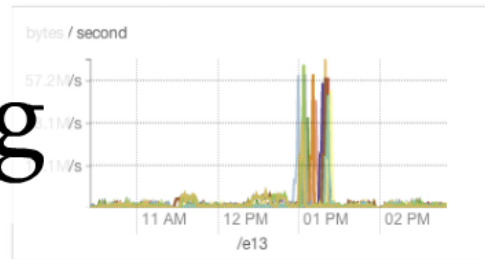
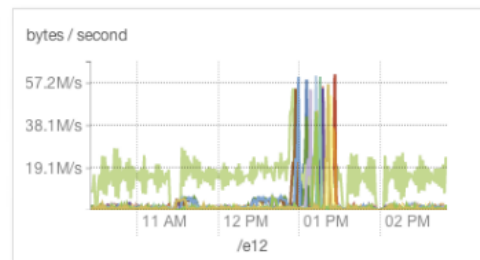
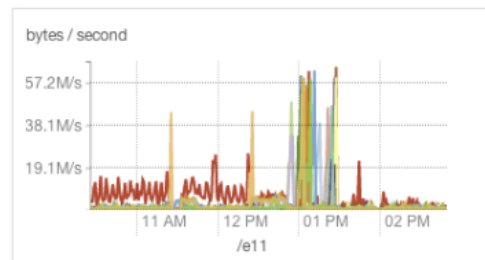
Maximum

Minimum

Apply

Title: Disk Writes Facetted by Rack

+ Save As View...

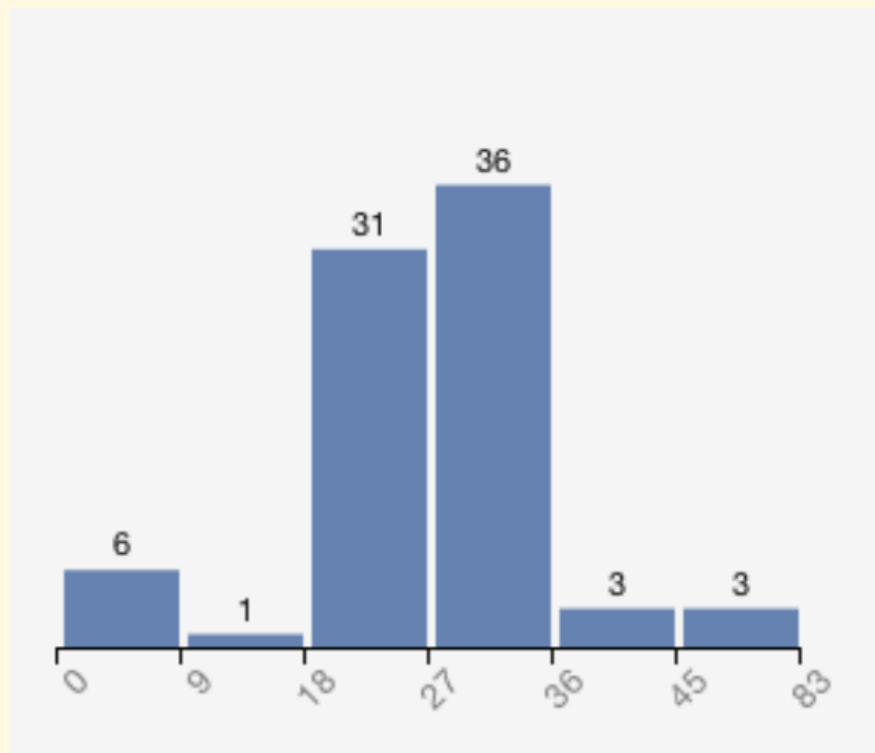


Queryability!

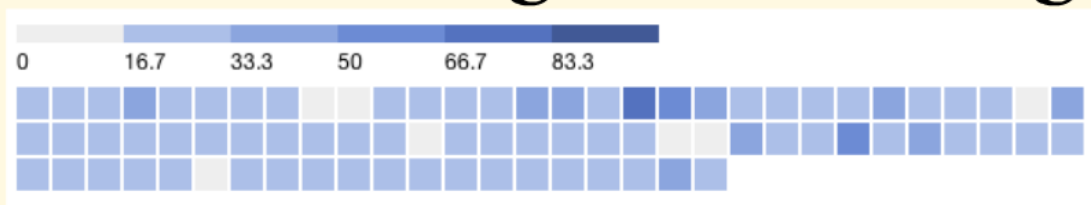
0 16.7



ity!



a good read: "grammar of graphics"



Outlier Grand Finale: Query all the things!

ask questions
of many hosts at once

Something I'm working on...

SysSQL

*An "information schema"
for systems*

```
# (for x in nightly47-{1..8}.ent.cloudera.com; do  
> ssh $x ns -A --no-headers -o user: done) |
```

SysSQL



```
SELECT username, count(*),  
ROUND(sum(mem_rss)/1024.0/1024.0,2) AS "Total RSS (MB)"  
FROM processes  
GROUP BY username  
ORDER BY count(*) DESC
```

Fetch Timeout

Query Timeout

Execute »

Help

[Overview](#) 
[Query Syntax](#) 

Schemas

[processes](#)
[threads](#)
[connections](#)
[hosts](#)
[mounts](#)
[network_interfaces](#)

Examples

[Top CPU Consumers](#)
[Connections by State](#)
[Processes by User](#)
[Cumulative CPU](#)
[Free Space](#)
[Network Connections](#)
[Listening Daemons](#)
[Pairwise Process Connections](#)

Logs(0) Counters

Results

28 rows returned, 0.023 total query time, 2.8105720000057772 CPU time on agents, 1.3 MiB transferred. 7/7 fetches succesful.

Showing 1 to 28 of 28 entries [First](#) [Previous](#) **1** [Next](#) [Last](#) Show entries

username	count(*)	Total RSS (MB)
root	3282	1386.32
cloudera-scm	171	14520.66
philip	40	8760.35
avahi	14	9.77


SysSQL

```
SELECT username, count(*),  
       ROUND(sum(mem_rss)/1024.0/1024.0,2) AS "Total RSS (MB)"  
FROM processes  
GROUP BY username  
ORDER BY count(*) DESC
```

Fetch Timeout 

10



Query Timeout 


10



Execute »

Help

[Overview](#) 

[Query Syntax](#) 

Schemas

[processes](#)

[threads](#)

[connections](#)

[hosts](#)

[mounts](#)

[network interface](#)

Examples

[Top CPU Consuming](#)

[Connections by](#)


```
# (for x in nightly47- $\{1..8\}$ .ent.cloudera.com; do
>   ssh $x ps -A --no-headers -o user; done) |
> sort |
> uniq -c |
> sort -rn
675 root
 14 hbase
 12 solr
 12 hdfs
 11 mapred
  8 qpidd
  8 ntp
  8 nscd
  8 mysql
  8 dbus
  8 cloudera-scm
  7 hue
  7 flume
  5 yarn
  5 impala
  3 hive
  2 zookeeper
```

Ok, fine, you can write queries in UNIX too.

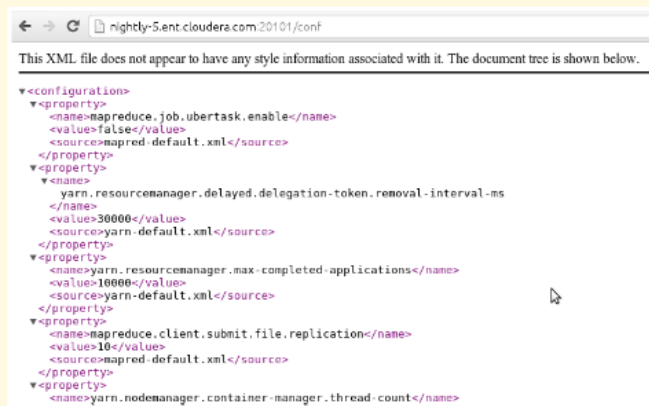
Depth

*you've found
a problematic
machine/process,
now what?*

status pages

Linux

Java



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>
```

/conf

status pages

for Hadoop:

/jmx, /conf, /stacks

no excuses for developers

huzzah for HTTP

← → ↻ 📄 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>
```



Don't leave the house without...

strace

jstack

pstack

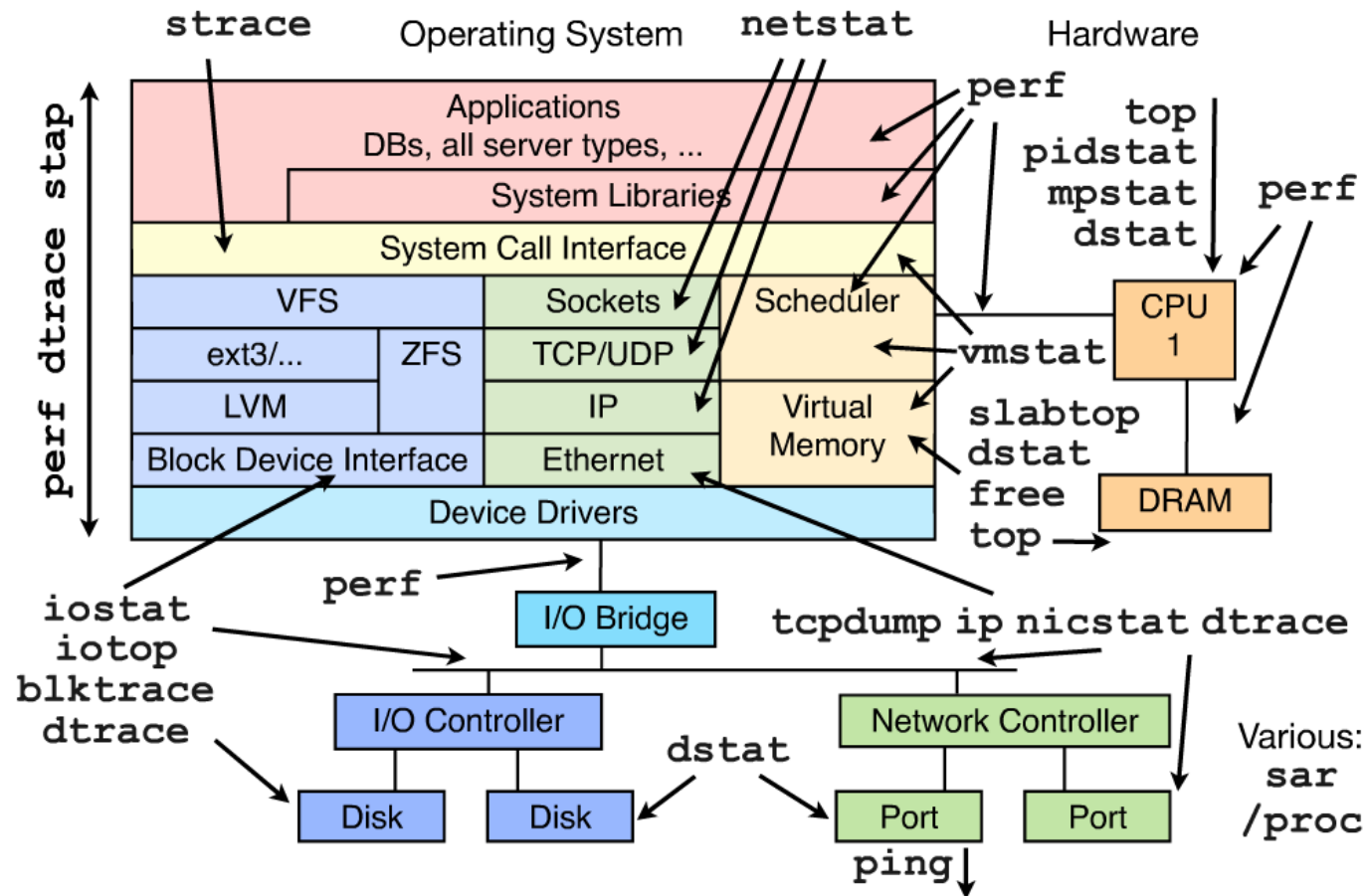
ps/top/htop

did I mention

strace?

@brendangregg's sweet slide

Analysis and Tools



open source is...
open source!

git grep (et al.) are undervalued

*I have the source of pretty
much every bit of the stack locally.*

Java stuff...

jps

jstack *you can also do unholy*

jmap *things with "agents"; instrument*

jinfo *running code, etc.*

`-Xdebug -Xrunjdwp:server=y,transport=dt_socket,address=4000,suspend=n`

attaching debuggers remotely



review

start broadly to find outliers

invest in monitoring/canaries: find simpler cases that fail

tool up

dig in with "traditional" tools

focus on communication paths



[**the end**]

philip@cloudera.com @philz42

roadly to find outliers