

Hadoop Adventures At Spotify

Adam Kawa

Data Engineer



How many times
were you told
bad news?


Namenode is sad



Inbox x



Erik Bernhardsson <erikbern@spotify.com>

to hadoop, discovery 

You probably know this, but a ton of datanodes seem to be **down** :(

How many times
did your excellent weekend
change into a workday?

Hadoop down?



Inbox x



Vianney Brandicourt <vianney@spotify.com>

to Analytics

Hi Infra,

I hope you are having an excellent weekend

I was hoping to run a few things today, but I'm under the impression that **hadoop** is **down**?

How many times
did you change your passion
for glory?



Pablo Barrera <pablo@spotify.com>

to Analytics ▾

Everything is under control. Today Adam, Uldis, Tommie, Johanees (investigating and solving the issue) and Javier (offering food and acting as an umbrella) fixed a really tricky problem on the **cluster**.



Wouter de Bie <wouter@spotify.com>

to Pablo, Analytics ▾

Awesome work guys!! I'm really proud.

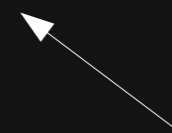
// Wouter

How many times
did you
surrender?

Many typos and grammar errors ;)



```
5:14 AM me: hi i did not fixed the cluster
5:15 AM NN is spending lots of time on being blocked
5:18 AM NN almost does not get (or process it really slowly) block reports
5:19 AM I tried a couple chanages in /etc/hadoop/conf/hdfs-site.xml and hadoop-env.sh
5:23 AM i must go sleep for a while
```



Hadoop nightmares guaranteed!

My answer to each of these questions:

At least once

Why Hadoop Adventures?

A rapidly **growing cluster!**

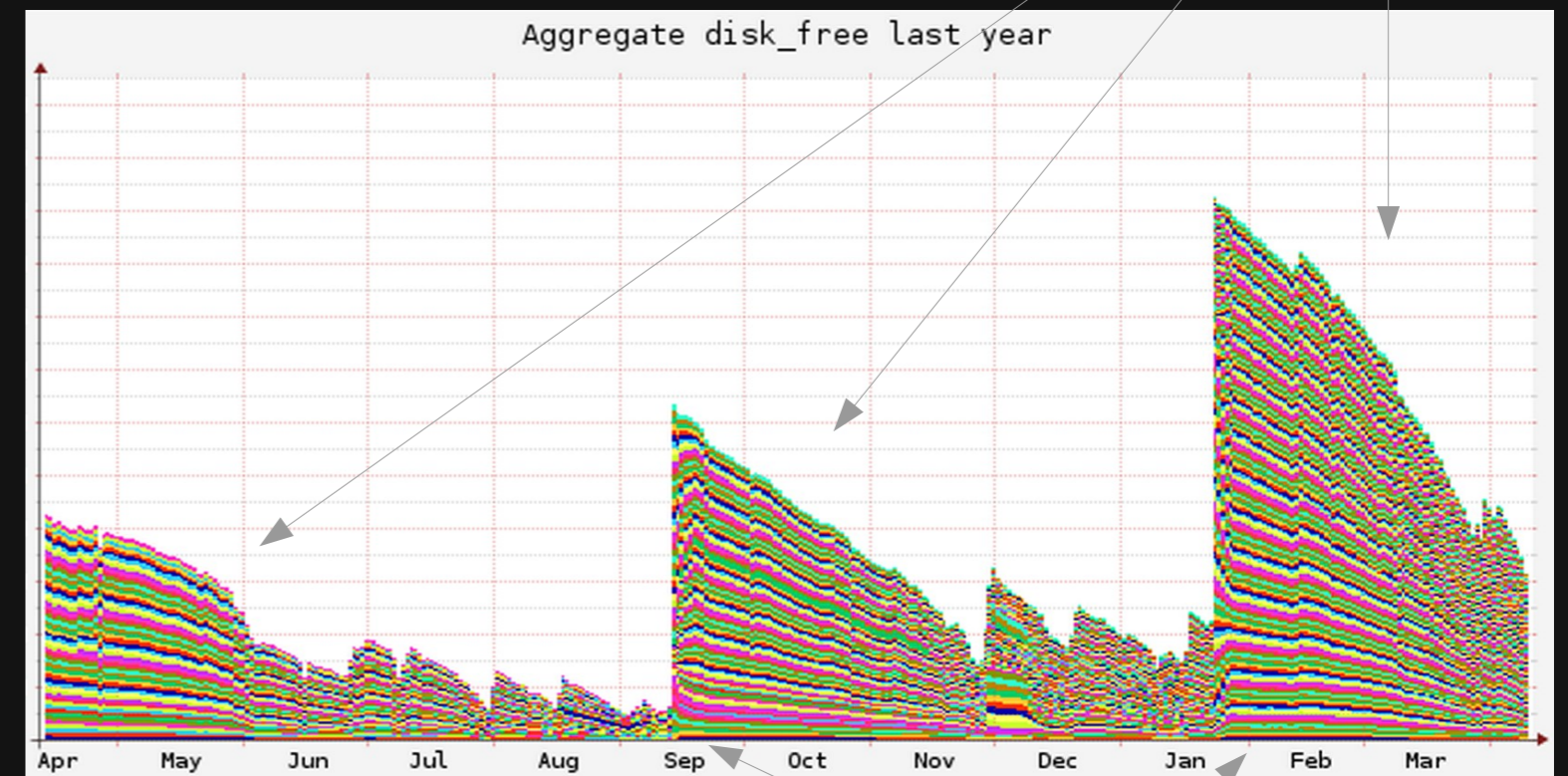
Configuration settings stop working

Masters have troubles coordinating slaves

More data + users + jobs

Own mistakes

The disk space is consumed quicker and quicker



Many nodes were added

What feelings did you get when it happened?

Responsibility, Critical, Expensive,
Interruptions, Frustration, Stress, Email,
Ashamed,
Cooperation, Learning, Surprise,
Fun, Exciting, Happy Pride

About Me

I needed to learn more about operating systems, hardware, networking...

A Java developer

who was developing MapReduce, Pig and Hive jobs

who started operating Hadoop and HBase on a 5-node cluster

who later joined Spotify

to meet a 138x larger Elephant



About The Elephant

I met this **Elephant**

He **grows rapidly** from 60 to 690 nodes

He stores 6.29PT of data

He receives 50TB of data each day

He seems to be **the largest in Europe**

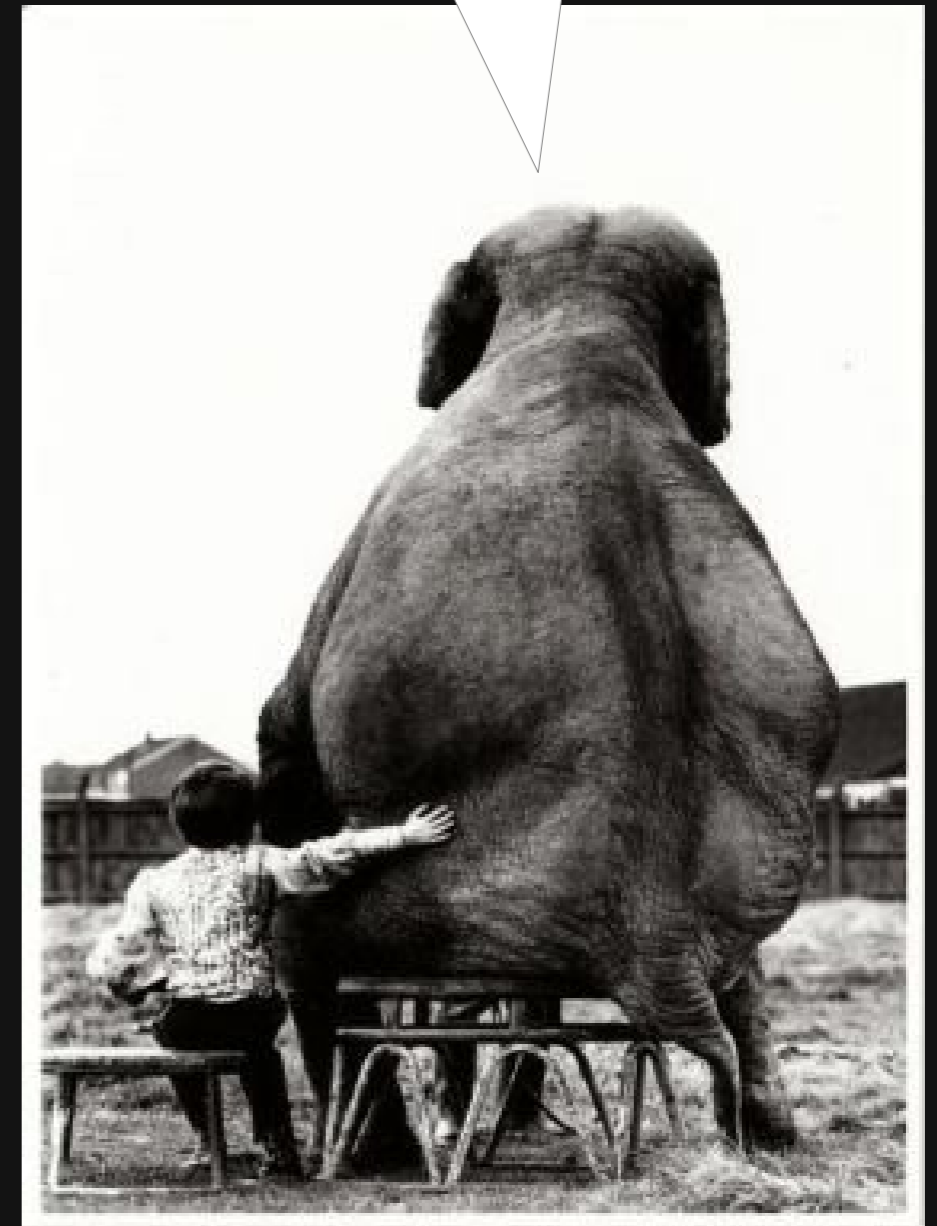
He **did not like me** at all

His friends did not like me too

We had **problems living together!**

I want to make them PUBLIC now!

Hands off
a little boy!



What Will I Talk About?

Five *real* (and my favourite) Hadoop incidents that
broke our cluster
or made it very unstable



Analytics Infrastructure / AI-1617

Cluster becomes unstable after around 1 hour of uptime

What Will I Share?

Real problems

Real mistakes

Real lessons learned

Real graphs

Real numbers

Real emails

Real excerpts from our conversations

Live Nodes	:	0 (Decommissioned: 0)
Dead Nodes	:	186 (Decommissioned: 0)

What Will Also I Share?

The *mistake* that I made and I *did not* like to talk about

Feeling: *Ashamed*



Adventure 1 (February 2013)

Troubles running more
resource-intensive Hive queries





Trouble running more resource-intensive Hive queries

Problem

A user can not run resource-intensive Hive queries

It happened immediately after significantly expanding the cluster





Trouble running more resource-intensive Hive queries

Description

The queries are valid

The queries run successfully on small datasets

But they fail on large datasets

Surprisingly they run successfully through other user accounts

The user has right permissions to HDFS directories and Hive tables



Trouble running more resource-intensive Hive queries

Observations

When this user runs a resource-intensive query

The cluster experiences stability problems

The NameNode becomes less responsive

It was losing the connection with DataNodes and mark them dead

But the DataNode daemons are running completely fine



Trouble running more resource-intensive Hive queries

The NameNode is throwing **thousands of warnings and exceptions** 14592 times only during 8 min (4768/min in a peak)

```
WARN org.apache.hadoop.security.ShellBasedUnixGroupsMapping: got exception trying to get groups for user <USERNAME>
org.apache.hadoop.util.Shell$ExitCodeException: id: <USERNAME>: No such user
    at org.apache.hadoop.util.Shell.runCommand(Shell.java:261)
    at org.apache.hadoop.util.Shell.run(Shell.java:188)
    at org.apache.hadoop.util.Shell$ShellCommandExecutor.execute(Shell.java:381)
    at org.apache.hadoop.util.Shell.execCommand(Shell.java:467)
    at org.apache.hadoop.util.Shell.execCommand(Shell.java:450)
    at org.apache.hadoop.security.ShellBasedUnixGroupsMapping.getUnixGroups(ShellBasedUnixGroupsMapping.java:86)
    at org.apache.hadoop.security.ShellBasedUnixGroupsMapping.getGroups(ShellBasedUnixGroupsMapping.java:55)
    at org.apache.hadoop.security.Groups.getGroups(Groups.java:89)
    at org.apache.hadoop.security.UserGroupInformation.getGroupNames(UserGroupInformation.java:1216)
    at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.<init>(FSPermissionChecker.java:51)
    at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkPermission(FSNamesystem.java:4541)
```



Trouble running more resource-intensive Hive queries

Normally

Hadoop is a very trusty elephant

The `username` comes from the client machine (and is not verified)

The `groupname` is resolved on the NameNode server

Using the shell command `"id -Gn <username>"`

If a user does `not have an account` on the NameNode server

The `ExitCodeException` `exception` is thrown



Possible Fixes

Create an user account on the NameNode server (dirty, insecure)

Use **AD/LDAP** for a user-group resolution

`hadoop.security.group.mapping.ldap.*` settings

If you also need the full-authentication, deploy Kerberos



Trouble running more resource-intensive Hive queries

Our Fix

We decided to use LDAP for a user-group resolution

However, LDAP settings in Hadoop did not work for us

Because `posixGroup` is not a supported filter group class

We found a workaround using `nsswitch.conf`



Lesson Learned

Know who is going to use your cluster

Know **who is abusing the cluster** (HDFS access and MapReduce jobs)

Parse the NameNode logs regularly

- Look for `FATAL`, `ERROR`, `Exception` messages

- Especially before and after expanding the cluster

Adventure 2 (March 2013)

DataNodes become
blocked sometimes (Part 1)





Problem

The DataNodes are marked dead by the NameNode

The DataNodes have not sent a heartbeat for a long time

Last Contact (seconds)

calc170	0	In Service	21.46	18.57	0.02	2.87	86.52	
calc171	976	In Service	21.46	18.69	0.03	2.75	87.08	
calc172	0	In Service	21.46	16.38	0.02	5.06	76.34	



When a DataNode Is Marked Dead

A costly block replication process is started

- Alive DataNodes consume resources to recreate missing replicas

- Map tasks are processing non-local blocks

Tasks and jobs experience problems

- They can fail or be not started at all (`BlockMissingException`)

WARNING : There are 50542 missing blocks. Please check the logs or run fsck in order to identify the missing blocks.

Live Nodes	:	194 (Decommissioned: 0)
Dead Nodes	:	19 (Decommissioned: 0)



Observations

\$ps shows the DataNode daemons exist on their servers

sudo jstack -F <datanode-pid> does not show anything interesting

Some of the DataNodes automatically re-join the cluster

The DataNodes seem to be blocked for a while



First (Relief) Idea!

Let's temporarily increase the datanode-liveness interval to get some breathing room...

Wouter: Maybe we should up the
`dfs.namenode.heartbeat.recheck-interval`?

Adam: What value could be OK? 15, maybe **20 minutes**?

Wouter: 20 maybe? It will give us a bit more breathing room.

Adam: Yes, the **replication will not start so quickly**..



A patch was quickly written, reviewed and deployed.

```
93 <property>
94   <name>dfs.namenode.heartbeat.recheck-interval</name>
95   <value>600</value>
96   <description>Determines datanode heartbeat interval in seconds.</description>
97 </property>
98
```

The formula is:

$2 * \text{dfs.namenode.heartbeat.recheck.interval} + 10 * \text{dfs.heartbeat.interval}$

and it should give us

$2 * 600 \text{ sec} + 10 * 3 \text{ sec} \rightarrow 20\text{min}:30\text{sec}$



Analytics Infrastructure / AI-1848

DataNodes becomes blocked sometimes (and thus marked dead by NameNode)

Crazy Idea!

Let's do other (important) tasks after deploying this (simple) change
... and do not measure its impact



Adventure 3 (March 2013)

The cluster becomes unstable
after around 1 hour of uptime





Cluster becomes unstable after around 1 hour of uptime

Problem

Each time, around 1 hour after restarting the NameNode

Majority (or all) of the DataNodes are marked dead by the NameNode

\$ps shows the the DataNode daemons

The DataNode servers are running fine

DataNodes usages	:	Min %	Median %	Max %	stdev %
		0 %	0 %	0 %	0 %
Live Nodes	:	0 (Decommissioned: 0)			
Dead Nodes	:	186 (Decommissioned: 0)			
Decommissioning Nodes	:	0			
Number of Under-Replicated Blocks	:	46389238			



Cluster becomes unstable after around 1 hour of uptime

Question?

Why does it happen each time,
exactly around 1 hour after restarting the NameNode?





Cluster becomes unstable after around 1 hour of uptime

First Idea!

"When a DataNode initially starts up,
as well as every hour thereafter,
it sends a block report to the NameNode"

You can read it in

- * a book,
- * blog posts,
- * even see it in the code!



Cluster becomes unstable after around 1 hour of uptime

Maybe?

A storm of block reports coming after the first hour
heavily overloads the NameNode?

starts a heavy garbage collection phase that freezes the NameNode?





Cluster becomes unstable after around 1 hour of uptime

Could it be a right fix?

Increase `dfs.blockreport.initialDelay` to something bigger than zero

Delay for first block report in seconds – a random value between 0 and `dfs.blockreport.initialDelay`.



Cluster becomes unstable after around 1 hour of uptime

Because

Changing `initialDelay` requires a restart of the NameNode

The restart will be longer (the block reports are sent with a delay)

... and it is a sunny Sunday afternoon

Let's also deploy memory+GC changes to the NN at the same time

We will solve more problems in fewer iterations!



Analytics Infrastructure / AI-1617

Cluster becomes unstable after around 1 hour of uptime

A couple of hours later...



Analytics Infrastructure / AI-1617

Cluster becomes unstable after around 1 hour of uptime

Wouter: Yeah, looks like **we're pretty good!**

Adam: Good timing, one evening before Monday...

Johannes: Yeah, maybe I'll get two or three hours of weekend!

▼  **Adam Kawa** added a comment - 2013-03-03 21:14

The cluster has been running fine for a couple hours. Looks that the issue is fixed.

We have:

1. increased `dfs.blockreport.initialDelay` to something bigger than 0, to avoid the situation when all DNs send block reports at the same time (the root cause of the issue)

Additionally (this had rather minor impact)

1. increased memory assigned to NN



Analytics Infrastructure / AI-1617

Cluster becomes unstable after around 1 hour of uptime

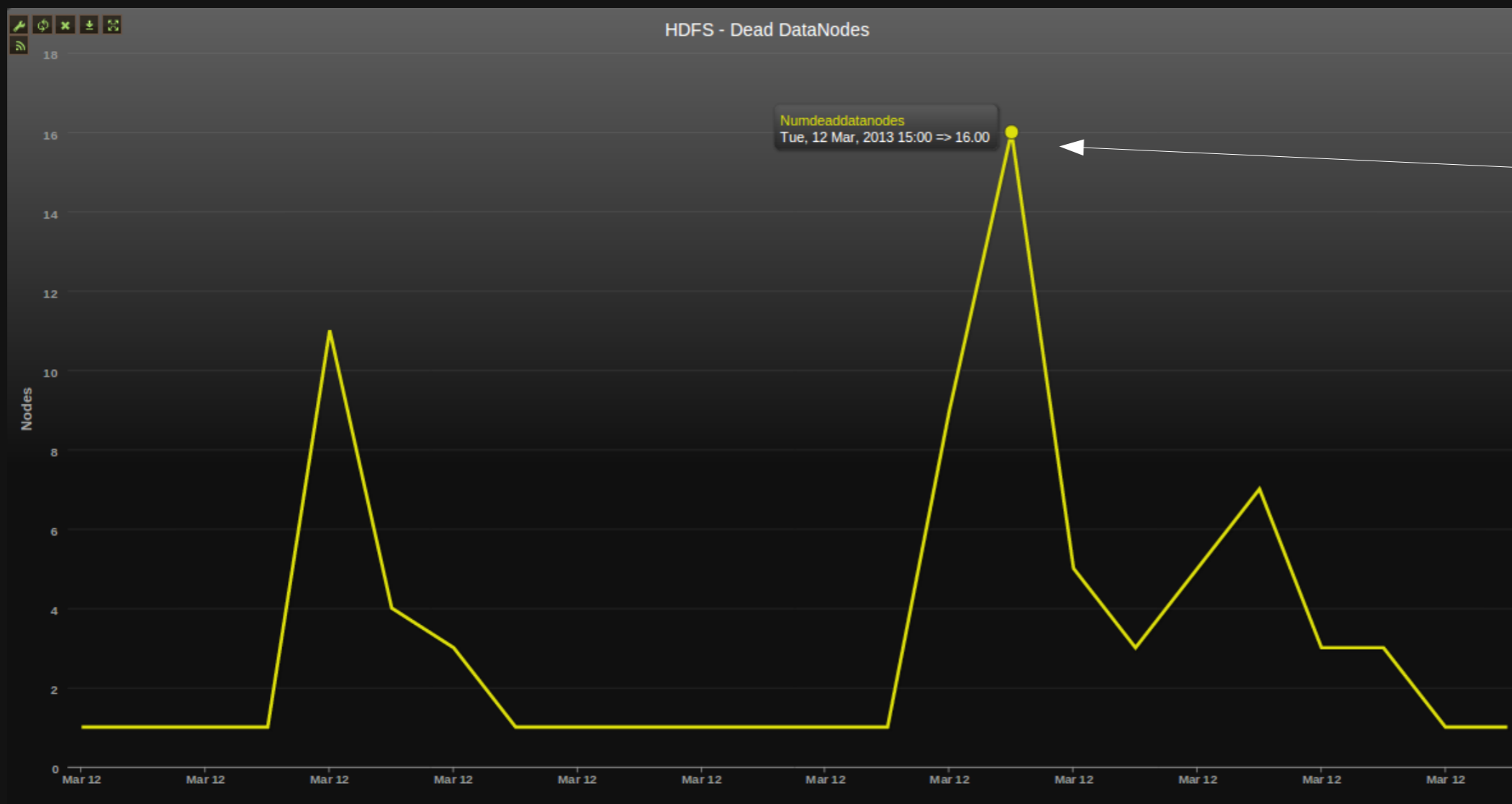
A couple of days later...



Cluster becomes unstable after around 1 hour of uptime

WARNING : There are 10779564 missing blocks.

Please check the logs or run fsck in order to identify the missing blocks.



Number
of
dead
DataNodes



Analytics Infrastructure / AI-1617

Cluster becomes unstable after around 1 hour of uptime

A Mail of Shame!

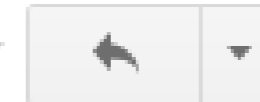
Buggy interval and lessons learned

Inbox x



Adam Kawa <kawaa@spotify.com>

Mar 11 ☆



to Wouter, Johannes ▾

Hi,

It turns out that we have run the cluster with shorted `datanode-liveness-interval` that expected. When I have changed this setting, I followed the information which says that this `interval` is configured in seconds. But after looking into the code, it turns out that should be specified in milliseconds.

Since timeout equals to $2 * \text{heartbeat.recheck.interval} + 10 * \text{heartbeat.interval}$ (by default is 3 seconds), our timeout was set to 31.2 seconds.

I do not feel good with this kind of bad-documented mistake

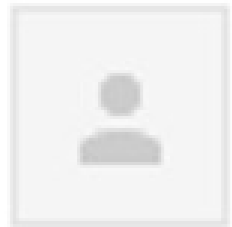
I had a dilemma whether to silently fix this interval or send this email...



Analytics Infrastructure / AI-1617

Cluster becomes unstable after around 1 hour of uptime

A Reply of support!



Johannes Fabian Rußek <jrussek@spotify.com>

to me, Wouter ▾

Nice find!

But we are still losing DNs every now and then, so the root of the issue has not been identified yet :(



Analytics Infrastructure / AI-1617

Cluster becomes unstable after around 1 hour of uptime

Next (Shocking) Finding!

What really happened that Sunday afternoon?



Adam Kawa <kawaa@spotify.com>



to Johannes, Wouter

Guys, do you remember this NN issue with random delay? (<https://jira.spotify.net/browse/AI-1617>)

It looks that ██████ changed the interval how often DNs sends block reports to NN and it is 6 hours instead of 1h (as it is commonly said).

I can not sleep due to this issue ;)

Kind regards,
Adam

NO storm of block report after the 1st hour means that ... tuning heap and GC helped



Cluster becomes unstable after around 1 hour of uptime

~~Too Many~~ Lessons Learned

Measure the impact of each single change

Never make bulk changes to the cluster

Double-check a description of a configuration parameter

Double-check the default values of configuration parameters

Question (almost) everything

Troubleshoot the cluster together interactively and non-interactively

Share the knowledge, even if you make a mistake

Give a support to your team-mates, even if they fail

Adventure 2 (March 2013)

DataNodes become blocked
sometimes (Part 2)





Observations

`sudo jstack -F <datanode-pid>` does not show anything interesting

`sudo -u hdfs jstack <datanode-pid>` shows something interesting

462 threads are blocked/waiting

19 threads are runnable

The threads are **waiting for the same lock**, to finish operations like:

```
org.apache.hadoop.hdfs.server.datanode.DataXceiver.readBlock
```

```
org.apache.hadoop.hdfs.server.datanode.fsdataset.impl.FsDatasetImpl.createRbw
```

```
org.apache.hadoop.hdfs.server.datanode.fsdataset.impl.FsVolumeList.getDfsUsed
```

```
org.apache.hadoop.hdfs.server.datanode.fsdataset.impl.FsDatasetImpl.initReplicaRecovery
```

```
org.apache.hadoop.hdfs.server.datanode.fsdataset.impl.FsDatasetImpl.finalizeBlock
```



DataNodes becomes blocked sometimes (and thus marked dead by NameNode)

The lock is held by other thread that is ... waiting for something else

```
1 "DataXceiver for client DFSCClient_NONMAPREDUCE_-561263399_1 at /10.255.10.18:39850 [Receiving block BP-133877431-10.
2   java.lang.Thread.State: WAITING (on object monitor)
3     at java.lang.Object.wait(Native Method)
4     at java.lang.Thread.join(Thread.java:1186)
5     - locked <0x00000000786fc8018> (a org.apache.hadoop.util.Daemon)
6     at java.lang.Thread.join(Thread.java:1239)
7     at org.apache.hadoop.hdfs.server.datanode.ReplicaInPipeline.stopWriter(ReplicaInPipeline.java:157)
8     at org.apache.hadoop.hdfs.server.datanode.fsdataset.impl.FsDatasetImpl.recoverRbw(FsDatasetImpl.java:706)
9     - locked <0x000000006890b10a0> (a org.apache.hadoop.hdfs.server.datanode.fsdataset.impl.FsDatasetImpl)
10    at org.apache.hadoop.hdfs.server.datanode.fsdataset.impl.FsDatasetImpl.recoverRbw(FsDatasetImpl.java:90)
11    at org.apache.hadoop.hdfs.server.datanode.BlockReceiver.<init>(BlockReceiver.java:163)
12    at org.apache.hadoop.hdfs.server.datanode.DataXceiver.writeBlock(DataXceiver.java:393)
13    at org.apache.hadoop.hdfs.protocol.datatransfer.Receiver.opWriteBlock(Receiver.java:98)
14    at org.apache.hadoop.hdfs.protocol.datatransfer.Receiver.processOp(Receiver.java:66)
15    at org.apache.hadoop.hdfs.server.datanode.DataXceiver.run(DataXceiver.java:219)
16    at java.lang.Thread.run(Thread.java:662)
```



recoverRbw tries to stop the other thread
it has the lock (the monitor object) that the other is waiting for

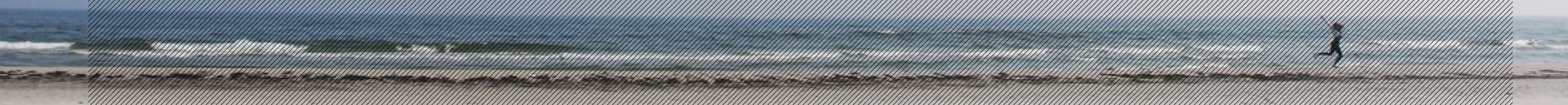
This **deadlock** is

Described in HDFS-3655

Duplicated by HDFS-5016

Related to HDFS-4851

Together
with many
other
threads





Analytics Infrastructure / AI-1848

DataNodes becomes blocked sometimes (and thus marked dead by NameNode)

Lesson Learned

Master the troubleshooting and monitoring tools

Contribute to Apache Hadoop more

- Send a post to Hadoop mailing list

- Raise a well-documented JIRA ticket

- Implement a patch



Adventure 4 (March 2013)

Tasks in Hive jobs are
constantly "KILLED UNCLEAN by
the user"





Task in hive jobs are constantly "KILLED_UNCLEAN by the user"

Problem

Surprisingly, ad-hoc Hive queries are running extremely long
Thousands of task are constantly being killed



Hadoop job_201303121738_7093 on jobtracker

User: hive

Job Name: DW7378:karinf:Hopps Hoppas Source start, referer

Job File: hdfs://namenode.c.lon.spotify.net:54310/user/hive/.staging/job_201303121738_7093/job.xml

Submit Host: aimee.lon.spotify.net

Submit Host Address: 78.31.10.2

Job-ACLs: All users are allowed

Job Setup: Successful

Status: Running



Started at: Wed Mar 13 08:45:22 UTC 2013

Running for: 12hrs, 20mins, 14sec

Job Cleanup: Pending

Only 1 task failed,
2x more task
were killed
than
were completed



Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	86.87% 	4808	443	247	4118	0	1 / 8707
reduce	0.00% 	999	999	0	0	0	0 / 0

Hadoop job_201303121738_7093 failures on jobtracker

Attempt	Task	Machine	State	Error	Logs
attempt_201303121738_7093_m_000000_0	task_201303121738_7093_m_000000	calc106.c.lon.spotify.net	KILLED	Request received to kill task 'attempt_201303121738_7093_m_000000_0' by user ----- Task has been KILLED_UNCLEAN by the user	Last 4KB Last 8KB All
attempt_201303121738_7093_m_000000_1	task_201303121738_7093_m_000000	calc74.c.lon.spotify.net	KILLED	Request received to kill task 'attempt_201303121738_7093_m_000000_1' by user ----- Task has been KILLED_UNCLEAN by the user	Last 4KB Last 8KB All
attempt_201303121738_7093_m_000002_0	task_201303121738_7093_m_000002	calc110.c.lon.spotify.net	KILLED	Request received to kill task 'attempt_201303121738_7093_m_000002_0' by user ----- Task has been KILLED_UNCLEAN by the user	Last 4KB Last 8KB All
attempt_201303121738_7093_m_000002_1	task_201303121738_7093_m_000002	calc74.c.lon.spotify.net	KILLED	Request received to kill task 'attempt_201303121738_7093_m_000002_1' by user ----- Task has been KILLED_UNCLEAN by the user ----- Request received to kill task 'attempt_201303121738_7093_m_000002_1' by user ----- Task has been KILLED by the user	Last 4KB Last 8KB All
attempt_201303121738_7093_m_000003_0	task_201303121738_7093_m_000003	calc161.c.lon.spotify.net	KILLED	Request received to kill task 'attempt_201303121738_7093_m_000003_0' by user ----- Task has been KILLED_UNCLEAN by the user	Last 4KB Last 8KB All
attempt_201303121738_7093_m_000003_1	task_201303121738_7093_m_000003	calc108.c.lon.spotify.net	KILLED	Request received to kill task 'attempt_201303121738_7093_m_000003_1' by user ----- Task has been KILLED_UNCLEAN by the user	Last 4KB Last 8KB All
attempt_201303121738_7093_m_000005_0	task_201303121738_7093_m_000005	calc75.c.lon.spotify.net	KILLED	Request received to kill task 'attempt_201303121738_7093_m_000005_0' by user ----- Task has been KILLED_UNCLEAN by the user	Last 4KB Last 8KB All
attempt_201303121738_7093_m_000005_1	task_201303121738_7093_m_000005	calc172.c.lon.spotify.net	KILLED	Request received to kill task 'attempt_201303121738_7093_m_000005_1' by user ----- Task has been KILLED_UNCLEAN by the user ----- Request received to kill task 'attempt_201303121738_7093_m_000005_1' by user ----- Task has been KILLED by the user	Last 4KB Last 8KB All



Task in hive jobs are constantly "KILLED_UNCLEAN by the user"

The logs show that the JobTracker gets a request to kill the tasks

Who can actually send a kill request?

User (using e.g. `mapred job -kill-task`)

JobTracker (a speculative duplicate, or when a whole job fails)

Fair Scheduler (if the preemption is enabled)





Task in hive jobs are constantly "KILLED_UNCLEAN by the user"

Two suspicious characters have a good alibi

Users

She/he is friendly, patient and respectful to others

JobTracker

The speculative execution is disabled

Jobs have not fail (yet)





Analytics Infrastructure / AI-1664

Task in hive jobs are constantly "KILLED_UNCLEAN by the user"

Key Observations

Tasks are usually killed quickly after the start

Surviving tasks are running fine for long time

Ad-hoc Hive queries are running in their own Fair Scheduler's pool



Task in hive jobs are constantly "KILLED_UNCLEAN by the user"

Eureka!

FairScheduler prefers to kill young!

Preempt the newest tasks
in an over-share pools
to forcibly give their slots
to starving pools





Analytics Infrastructure / AI-1664

Task in hive jobs are constantly "KILLED_UNCLEAN by the user"

Hive pool was running over its minimum and fair shares
Other pools were starving

So that

Fair Scheduler was (legally) killing Hive tasks from time to time

Fair Scheduler can kill to be KIND...



Possible Fixes

Tune minimum shares based on your workload

Tune preemption timeouts based on your workload

Disable the preemption

Limit the number of map/reduce tasks in a pool

Limit the number of jobs in a pool

Alternative

Capacity Scheduler





Analytics Infrastructure / AI-1664

Task in hive jobs are constantly "KILLED_UNCLEAN by the user"

Lessons Learned

A scheduler should NOT be considered as a "black-box"



Adventure 5 (April 2013)

JobTracker runs
super slowly



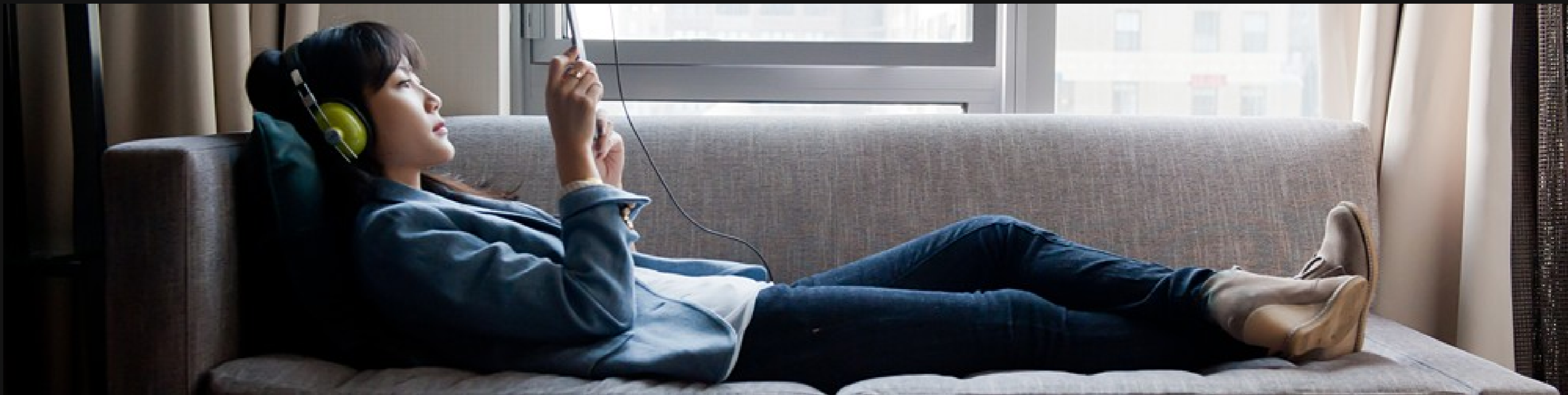


Analytics Infrastructure / AI-1870

JobTracker runs super slowly (big jobs slow it down?)

Problem

JobTracker becomes super slow... (but used to be super snappy)
Everybody is annoyed by unresponsive JobTracker web interface





JobTracker runs super slowly (big jobs slow it down?)

Observations

Many large jobs are running on the cluster

Only the largest one (at that time) run 58.538 tasks

It aims to process 21.82 TB of data

1.5 year of data from one of our datasets

It is an ad-hoc Hive job

It is still easy to run a large Hive job even if `hive.mapred.mode` is set to `strict...`



Analytics Infrastructure / AI-1870

JobTracker runs super slowly (big jobs slow it down?)

Possible solution

Limit the number of tasks per job using

`mapred.jobtracker.maxtasks.per.job`

← Unfortunately,
there are no separate properties
for map and reduce tasks





Real-Life Dialogue

Adam: What value for `mapred.jobtracker.maxtasks.per.job`?

Sven: Maybe 50K

Adam: Why not 30K?

Sven: We can run large production jobs and it is safer to start with a high value and maybe decrease it later.

Adam: Hmm? Sounds good...!
But maybe 40K, not 50K? ;)

Sven: OK. Let's meet in the middle...



JobTracker runs super slowly (big jobs slow it down?)

The previous approach is a great example of “guesstimation”
Negotiation skills are required!

Question?

Should a real data-driven company
make such a decision
based on a **guess or data?**



JobTracker runs super slowly (big jobs slow it down?)

The data and the answer

Based on 436K jobs from two months

The largest production job created 22.6K tasks (still to many!)

The jobs that create more than 23K tasks

Usually ad-hoc Hive queries or Python streaming jobs

These jobs fail, or are killed by impatient users in most cases

Hadoop can be used to ...
process data generated by Hadoop ;)



Lessons Learned

Make data-driven decisions

- Use Hadoop to analyze ... Hadoop

Administrators should cooperate with developers+analysts

- Developers+analysts will often adapt to the changes

Negotiation skills are still useful

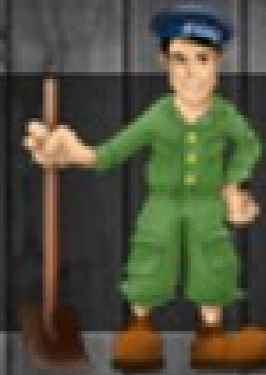
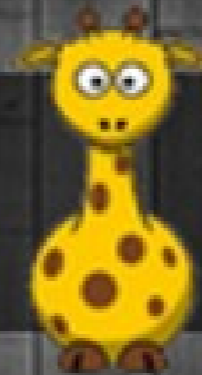
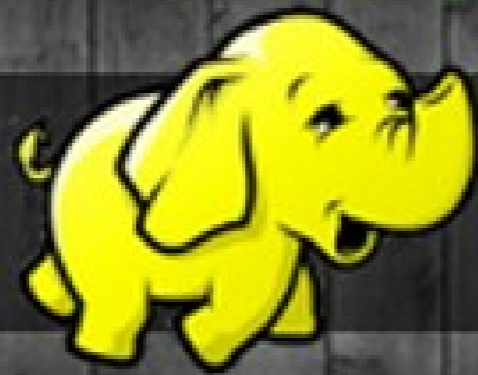
A technical blog
with some bytes of humour!

More Adventures at HakunaMapData.com

Nodes are marked dead due to a heavy swapping

The out-of-memory killer is killing TaskTrackers

The script for daily data cleaning runs more than one day!



Adventure

The nearest future will be
BIGGER

YARN + Tez + Fast SQL



Grand Lesson Learned

Hadoop is like a kid!

It needs love and care

It can make you proud, but it also will cause problems

It will grow and change quickly

It will bring new friends home

It will surprise you





Are there any ~~issues~~
questions? ;)

BONUS!



A person is sitting on the back of a car at night, looking at a smartphone. The scene is dimly lit with a blue tint, and there are blurred lights in the background. The person is wearing a dark jacket and has their hair tied back. The car's taillight is visible in the foreground.

One Question:

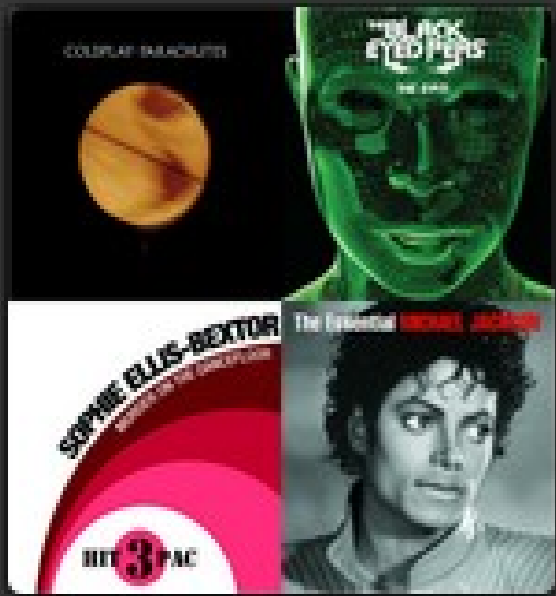
What can happen after some time of simultaneous development of MapReduce jobs maintenance of a large cluster and listening to perfect music for every moment?



A Possible Answer:

You may discover Hadoop in the lyrics of many popular songs!





Hadoop Playlist (more info: <http://bit.ly/1a0...>)

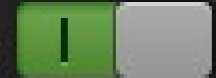
by you

Including artists: Coldplay, The Black Eyed Peas, Sophie Ellis-Bextor, Michael Jackson, The Cardigans, Natty King, Counting Crows, Ernie Sabella

Share...

Start Radio

Available Offline



12 tracks (46min)

	Track	Artist	Time	Album	Added	User
★	Yellow	Coldplay	4:30	Parachutes	Sun Sep ...	Adam Kawa
★	Meet Me Halfway	The Black Eyed Peas	4:45	THE E.N.D. (THE ENE...	Sun Sep ...	Adam Kawa
★	Murder On The Dancefloor	Sophie Ellis-Bextor	3:50	Murder On The Danc...	Sun Sep ...	Adam Kawa
🔊	Smooth Criminal (Radio Edit)	Michael Jackson	4:18	The Essential Michael...	Sun Sep ...	Adam Kawa
★	Lovefool - Radio Edit	The Cardigans	3:18	Best Of	Sun Sep ...	Adam Kawa
★	No Guns To Town	Natty King	3:53	Reggae Gold 2004	Fri Sep 27	Adam Kawa
★	Accidentally In Love	Counting Crows	3:09	Shrek 2	Sun Sep ...	Adam Kawa
★	Hakuna Matata - From Disney's "t...	Ernie Sabella, Josep...	3:34	Now That's What I Ca...	Sun Sep ...	Adam Kawa
★	Nine Million Bicycles	Katie Melua	3:19	Le Coeur Des Homm...	Wed Sep...	Adam Kawa
★	42	Coldplay	3:58	Viva La Vida Or Deat...	Fri Sep 27	Adam Kawa
★	I'm Your Puppet	James & Bobby Purify	3:00	Shake A Tail Feather! ...	Sun Sep ...	Adam Kawa
★	Dead And Gone - feat. Justin Tim...	T.I.	5:00	Paper Trail	Sun Sep ...	Adam Kawa

Want to join the band?

Check out spotify.com/jobs or @Spotifyjobs
for more information

kawaa@spotify.com

HakunaMapData.com



A crowd of people is silhouetted against a bright, golden sunset. Many hands are raised in the air, some pointing upwards, others in various expressive gestures. The scene conveys a sense of celebration and collective joy.

Thank you!