Self introduction

- Akira Ajisaka (NTT DATA)
  - Apache Hadoop Committer & PMC member
    - 130+ commits in 2015
    - Working on usability and supportability
  - "Open-Source Professional Services" team
    - Has deployed and supported 10k+ nodes of Hadoop clusters overall for 7 years
    - Contributing to Apache Hadoop 6th in the world with NTT [1]
    - ASF Committers: Spark, Yetus, HTrace(incubating)

http://ajisakaa.blogspot.com/2015/02/the-activities-of-apache-hadoop.html
Agenda

- What's Apache Hadoop 3?
- Differences between Hadoop 3 and 2
  - New Features
  - Incompatible Changes
- Current Status
- Summary
What's Apache Hadoop 3?
Disclaimer

- Apache Hadoop 3 is now undefined
  - Not released
  - Not decided what feature is in or not

- This presentation is not always correct

- I'd like to introduce as far as I know
What's Apache Hadoop 3?

Hadoop 3 and 2 were diverged in 2011 (5 years ago!)
Discussions for releasing Apache Hadoop 3

2014/6
- Discussed releasing Hadoop 3 with upgrading JDK version
- Decided to make 2.6 the last release that supports JDK6

2015/3
- Oracle JDK7 is EoL, so we need to upgrade
- After all we put off the event for another 12 months

2016/2
- It's time to revisit Hadoop 3 release plans
- Developers are agreed with releasing Hadoop 3
When will it be released?

- Developer mailing list:
  - Releasing alphas through the summer
  - Freeze and stabilize for GA in Nov/Dec.

- I suppose Hadoop 3 GA will be released in the end of 2016

- There are many tasks to do
  - I'll introduce them in this slide later
Difference between Hadoop 3 and 2
YARN is unchanged

- YARN: the biggest difference between Hadoop 1 & 2
- Hadoop 3 still uses YARN
New features

- 440 fixed issues only in Hadoop 3 (as of 5/8/2016)
  - [https://s.apache.org/GpUq](https://s.apache.org/GpUq)

- HDFS erasure coding
- Shell script rewrite
- Task level native optimization
- Derive heap size or mapreduce.*.memory.mb automatically
- Support more than 2 NameNodes
- and more
Break compatibility

- Major version up is to clean up the code
  - Deprecated APIs can be removed only in changing major version
    - @Public and @Stable Java API
    - REST API
    - Metrics/JMX
    - CLI
    - Environment variables
  - Wire-compatibility can be broken
    - 2.X client cannot talk to 3.X server and vice versa
- Compatibility Guide:
Erasure Coding (HDFS-7285)

Problem
- Reduce costs of storage
- Blocks are replicated to 3 DNs
  - 3x storage overhead is costly

Solution
- Use Erasure Code

<table>
<thead>
<tr>
<th></th>
<th>3-replication</th>
<th>(6,3)-Reed-Solomon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tolerates</td>
<td>2 failures</td>
<td>3 failures</td>
</tr>
<tr>
<td>Disk Usage</td>
<td>3x</td>
<td>1.5x</td>
</tr>
</tbody>
</table>
Erasure Coding: Write files using (6,3)-Reed-Solomon

- Write data to 9 DNs in parallel

![Diagram showing file write process using Erasure Coding]

- ECClient
- 6 Data Blocks
- DN1
- DN6
- DN7
- DN9
- 3 Parity Blocks
- Incoming Data
Erasure Coding: Read files

- Read data from 6 DNs in parallel
Erasure Coding: Read files when DN fails

- Read data from **arbitrary** 6 DNs in parallel
Erasure Coding: Current Status

Phase 1: striping layout
- C = 64KB (default)
- Work for small files
- No data locality
- Available on trunk

Phase 2: contiguous layout
- C = 128MB (= HDFS Block size)
- Not work for small files
- Data locality
- Now in progress (HDFS-8030)
Shell Script Rewrite (HADOOP-9902)

- Hadoop and Shell Script
  - Launching daemons
  - Hadoop CLI
- Difficult to understand
  - What is the correct env var to set a option
    - java classpath?
    - java.library.path?
    - GC options?
  - How to add the option to the env var
  - We have to read **almost all** the shell scripts!
After rewriting the scripts ...

■ Easy to understand
■ Because shell API doc is available
  ● Shelldoc maker generates docs from the scripts
  ● Similar to JavaDoc

Public API

My documentation build (trunk):
Very similar to .bashrc

- Read the API doc
- Create your own ~/.hadoopXX
  - .hadoop-env : hadoop-env.sh for each user
  - .hadooprc : called after shell env vars are configured
- And that's all :)

ex.) Set additional classpath (.hadooprc)

```
hadoop_add_classpath /path/to/my/jar
```
--debug option is available

```
$ hadoop --debug version
DEBUG: hadoop_parse_args: processing version
DEBUG: hadoop_parse: asking caller to skip 1
DEBUG: HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
(snip)
DEBUG: Applying the user's .hadooprc
DEBUG: Initial CLASSPATH=/path/to/my/jar
DEBUG: Initialize CLASSPATH
(snip)
DEBUG: Initial CLASSPATH=/usr/local/hadoop/share/hadoop/common/lib/*
```

CLASSPATH was overwritten!!
(before HADOOP-13045)

- Useful for troubleshooting
Many new features, bug fixes, improvements

- 'hadoop distch' to change the ownership and permissions on many files via MapReduce job
- 'hadoop jnopath' to print java.library.path
- 'hadoop --daemon' instead of hadoop-daemon.sh
  - ex.) hdfs --daemon status namenode
  - The return code for status is LSB-compatible
  - hadoop-daemon(s).sh are now deprecated
- .out files are now appended (not overwritten)
  - Allows external log rotation
- and many more
  - see https://issues.apache.org/jira/browse/HADOOP-9902
Task level native optimization (MAPREDUE-2841)

- Add a native implementation of the map output collector
  - Sort, Spill and IFile serialization

- Prequisites
  - Built with -Pnative option
  - Custom writable types and comparators are not supported

- Setting

```xml
<property name="mapreduce.job.map.output.collector.class" value="org.apache.hadoop.mapred.nativetask.NativeMapOutputCollectorDelegator"/>
```

Tips: compact xml form will be supported in Hadoop 3 (HADOOP-6964)
Release Note in the issue:

- "For **shuffle-intensive** jobs this may provide speed-ups of 30% or more."

Benchmarked with 3 slaves (m3.xlarge)
- CentOS 7.2
- 3.0.0-SNAPSHOT (revision 5865fe2b)

A very **shuffle-intensive** wordcount job
- Input: 2.6GB (compressed)
- Shuffle: 14GB
- Output: 10GB
Benchmark result

- **Result:** 506sec -> 383sec *(25% speedup)*
  - Average Map Time: 182sec -> 89sec

- Map task log (native optimization enabled)

```
INFO [main] org.apache.hadoop.mapred.nativetask.NativeMapOutputCollectorDelegator: Native output collector can be successfully enabled!
```

- **Tips:** Backported to CDH 5.2.0 or later
In Hadoop 2, **two similar properties must be set**:

- `mapreduce.\{map,reduce\}.memory.mb`
  - The amount of memory to request from the scheduler for each task (ex. 2048)
- `mapreduce.\{map,reduce\}.java.opts`
  - Java options for YARN containers (ex. `-Xmx2G`)

In Hadoop 3, **either** is enough

- `.java.opts` is derived from `.memory.mb` and vice versa
  
  - `.java.opts = .memory.mb * mapreduce.job.heap.memory-mb.ratio`
  - `.memory.mb = .java.opts / mapreduce.job.heap.memory-mb.ratio`
- Hadoop 2 now supports only 2 NameNodes
  - 1 active and 1 standby

- Hadoop 3 supports 2 or more standby NameNodes
  - provides additional fault-tolerance
  - avoids multiple standby NNs to checkpoint at the same time
  - # of standby should be small due to block report

- FYI: ResourceManager already supports multi-standby in branch-2
Other new features

- metrics2 sink plugin for Apache Kafka (HADOOP-10949)
- .jhist default format is changed from json to binary (MAPREDUCE-6613)
- Use FileOutputCommitter v2 by default (MAPREDUCE-6336)
- Allow/disallow snapshots via WebHDFS (HDFS-9057)
- Check and make checkpoint before stopping NameNode (HDFS-6353)
- YARN TimelineServer v2 (YARN-2928)
- YARN WebUI v2 (YARN-3368)
- Dynamic subcommands (HADOOP-12930)
- and many other changes
Incompatible Changes
Incompatible changes

- Many deprecated APIs will be removed
  - `hftp/hsftp/s3` -> `webhdfs/s3{a,n}`
  - Metrics v1
  - `org.apache.hadoop.Records`
  - and more

- Improved CLI output
  - `'mapred job -list' shows the job name as well`
  - `'hadoop fs -du' shows the raw disk usage, and aligned more unix-like`
  - and more

- Search 'Incompatible change' flag
  - [https://s.apache.org/sMO4](https://s.apache.org/sMO4)
Bump up the versions of the libraries

- Drop JDK7 support (HADOOP-11858)
- Hello Lambda!
- Dependency Hell
  - Tomcat
  - Jetty
  - Jersey
  - Guava
  - Log4J
  - Jackson
  - and many many, many more...
Classpath isolation (HADOOP-11656)

- Relaxing the "dependency hell"
  - Separate client and server jars
  - Client jar does not pull any third party dependencies

- If the isolation is done ...
  - We can safely upgrade the libraries in server code
  - In branch-2, the upgrade is incompatible :(
Current Status
Current Status

- Most of the new features are already available
  - Erasure Coding
  - Shell Scripts Rewrite
  - Task level native optimization
- Need more contribution
  - Bumping the library versions
  - Classpath isolation
  - Remove deprecated XXX
- Discussion
  - Release from trunk or cut new branch-3
  - How should we make the releases alpha, beta, or GA?
Summary
Summary

- Apache Hadoop 3 has
  - Many new features and code cleanups
  - Many remaining tasks

- Need your help to release Hadoop 3 earlier!
  - Not only creating patches but also testing are welcome!

- (Probably) Hadoop 3 GA will be released in 2016
Suppose there are many Apache Hadoop developers/users in this room :)  

Tell me
- if there are any required tasks not in this talk
- if I am misunderstanding something
- everything you want to tell

Discussion
- Release from trunk or cut branch-3
- When we should support JDK9?
- What is alpha/beta/GA?
- ...

I'd like to feedback to the community
Appendix

- Native erasure coding support inside HDFS (Strata + Hadoop World New York 2015)