APACHE
HBASE

# Project Status

**HBaseConAsia2018, Beijing**

Michael Stack <stack@apache.org>
Yu Li <liyu@apache.org>

HBASE
HUAWEI
MI 小米
Alibaba Group 阿里巴巴集团
FiberHome
Apache HBaseCon Asia
— 2017.8 中国 . 深圳 —

Pervasive...

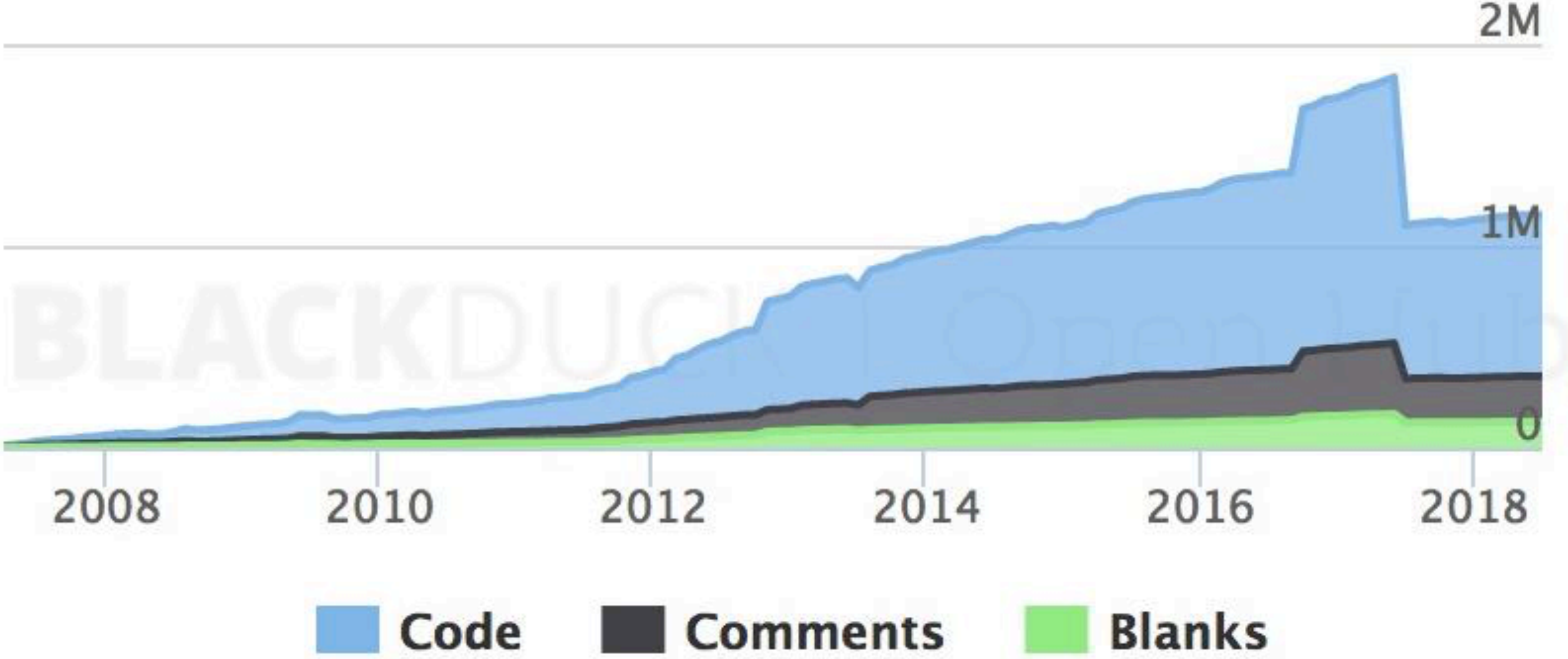
...distributed, scalable, big data store

# In a nutshell...

- ...15,409 commits made by 311 contributors
- ...representing 800,490 lines of code
- ...mostly written in Java
- ...has a well established, mature codebase
- ...maintained by a very large development team
- ...with stable Y-O-Y commits
- ...took an estimated 222 years of effort (COCOMO model)
- ...starting with its first commit in April, 2007 (>10 years old!)

Source https://www.openhub.net/p/hbase

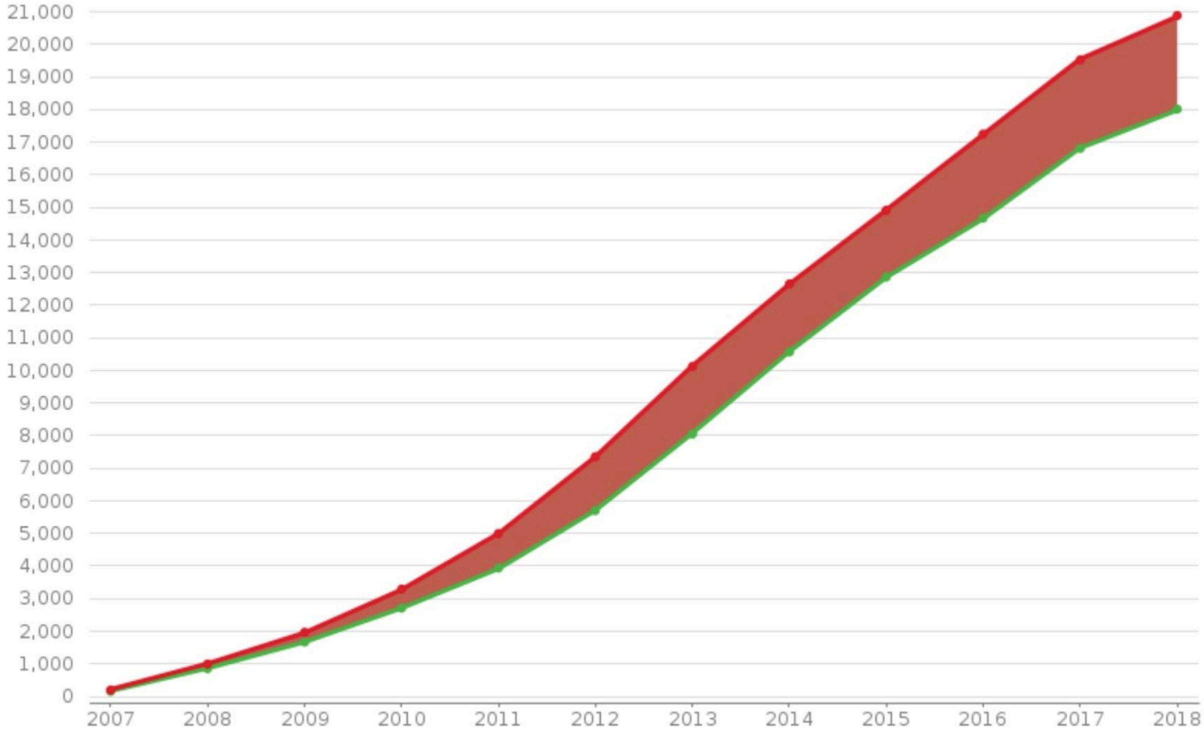
# LOC



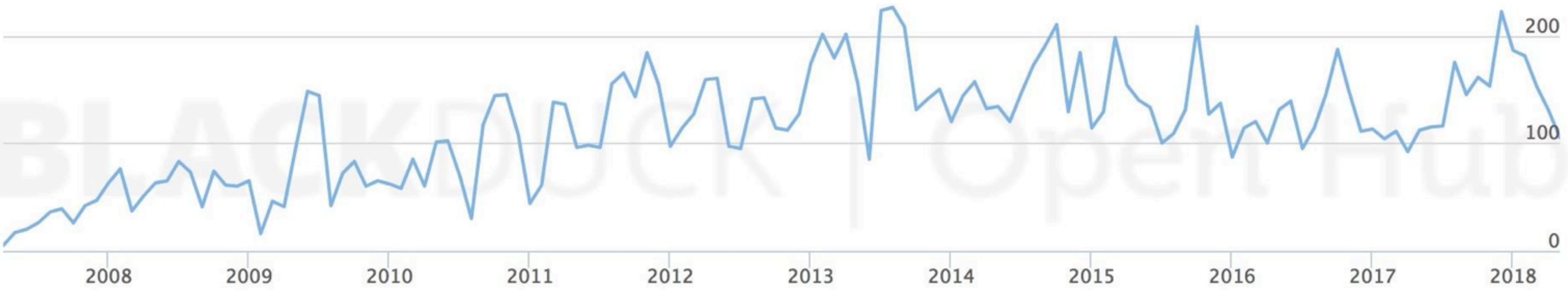
Source https://www.openhub.net/p/hbase

# Issues

This chart shows the number of issues **created** vs the number of issues **resolved** in the last **4000** days.



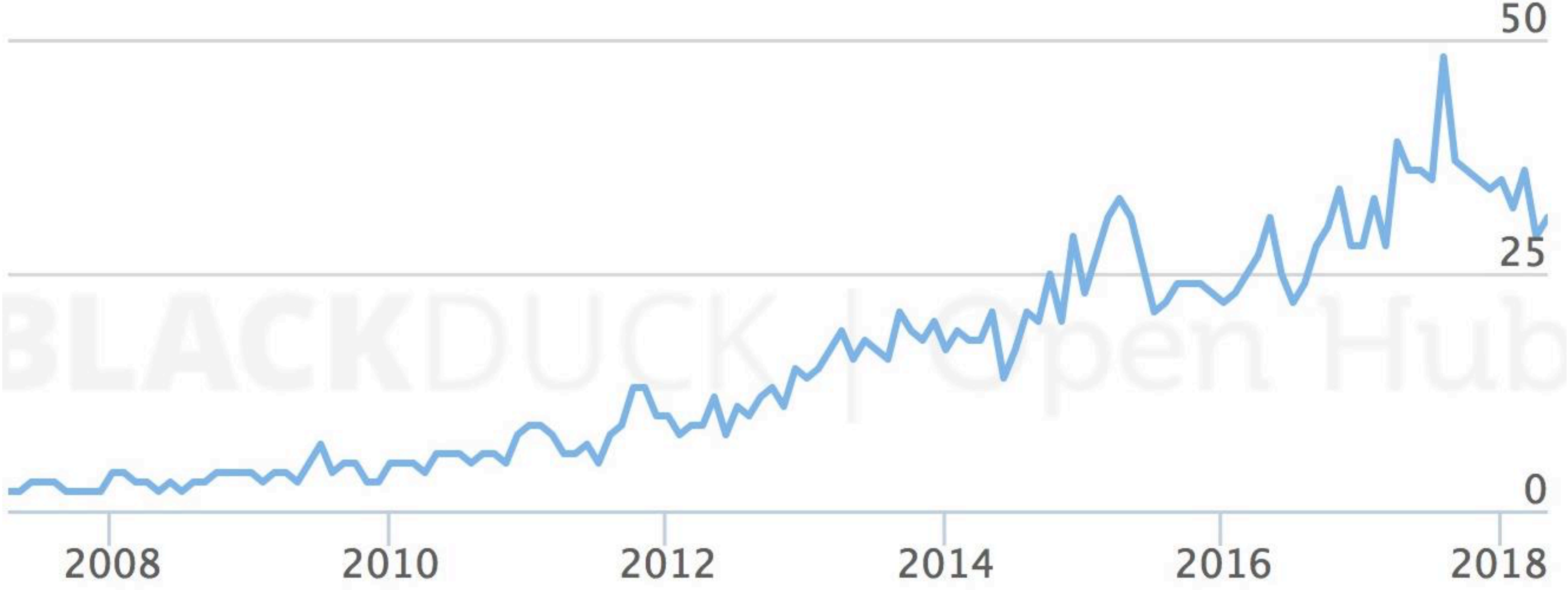
# Commits per month



BLACK DUCK | Open Hub

Source https://www.openhub.net/p/hbase

# Contributors



Source https://www.openhub.net/p/hbase

# New PMC Chairperson!

*“The HBase project represents solid, useful computer science that solves problems and runs businesses every day. To keep that going, we need to keep bringing new ideas and approaches into the project...we need to continue to attract people from all backgrounds and parts of the world. I'd love to see more women, more people of color, and even more worldwide diversity. I'd like to see more contributions from people not employed by big data platform companies.*

*“If we continue to strive for a diversity of ideas and experiences, we'll keep innovating so that HBase remains relevant for years to come.*”

Misty Linville, Vice-President of the Apache HBase Project

# Our project

- Apache HBase is an Open Source Apache project.
- It’s what **we** want to make of it.
- No owners!
- Anyone can help!
- All welcome!
- The more, the merrier!

# Active branches

| ***Active Branches*** | ***Latest Branche Release*** | ***Release Manager*** |
|---|---|---|
| branch-1.2 | 1.2.6.1 (EOL’d) | Sean Busbey |
| branch-1.3 | 1.3.2.1 (Yahoo) | Francis Liu |
| branch-1.4 | 1.4.6 (Current Stable) | Andrew Purtell |
| branch-1.5 | Coming... | Andrew Purtell |
| branch-2.0 | 2.0.1 | Michael Stack |
| branch-2.1 | 2.1.0 | Duo Zhang |
| branch-2.2 | <null> | <null> |
| branch-3 | <null> | <null> |

# hbase-2.0.0

# 2.0.0: Long-time coming

- Branched **four years** ago
- Released end-of-April, 2018
- Took > 1 year to stabilize
  - hbase-2.0.0 released, April 29th, 2018
  - hbase-2.0.0-beta2 released, March 22nd, 2018
  - hbase-2.0.0-beta1 released January, 16th, 2018
  - hbase-2.0.0-alpha4 released November 4th, 2017
  - hbase-2.0.0-alpha3 released September 17th, 2017
  - hbase-2.0.0-alpha2 released August 21st, 2017
  - hbase-2.0.0-alpha1 released June 22nd, 2017
- Multiple Release Managers
  - Matteo Bertozzi, Stephen Yuan Jiang, yours truly...

# Lets not do this again!

2.0.0

- Backed-up mountains of “Tech Debt”
  - Rotted Unit Tests
    - *“...99/100 it was the test, not Apache Infra”*
  - Performance regressions
    - Not out of the woods yet…

2.0.0
TM
GOAL
s

# Goals: Compatibility

- Double-down on Semantic Versioning, semver
  - Adopted in hbase-1.0.0
  - MAJOR.MINOR.PATCH[-IDENTIFIER]
    - E.g. 2.0.0-alpha1

2.0.0

# Goals: Compatibility

- But…
  - Semantic Versioning is about API only
    - ***What about….***
      - Internal/External Interfaces
        - Where is Client Interface when Spark/MapReduce
- It’s complicated...

# Goals: Compatibility

- From ~~Hadoop…~~ Yetus, annotations
  - `InterfaceAudience.Public`
    - Get/Put/Scan/Connection
  - `InterfaceAudience.LimitedPrivate`
    - Coprocessors, Replication, etc.
  - `InterfaceAudience.Private`
    - Internal only
- What about…
  - Source/Binary compatibility
  - Serializations
    - Wire
    - Formats in HDFS/Zookeeper
  - Dependencies
- See refguide semver section
  - http://hbase.apache.org/book.html#hbase.versioning

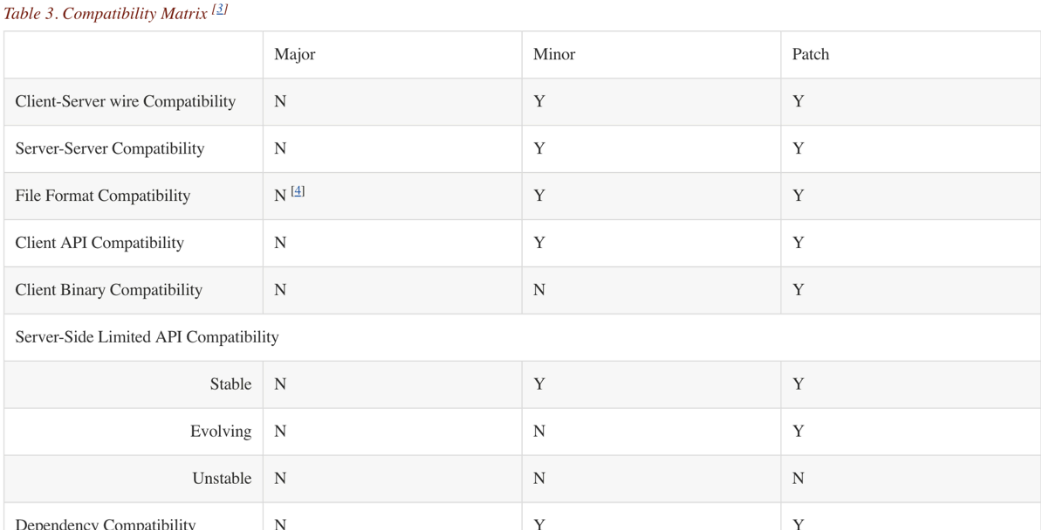


*Table 3. Compatibility Matrix* [3]

| | Major | Minor | Patch |
|---|---|---|---|
| Client-Server wire Compatibility | N | Y | Y |
| Server-Server Compatibility | N | Y | Y |
| File Format Compatibility | N [4] | Y | Y |
| Client API Compatibility | N | Y | Y |
| Client Binary Compatibility | N | N | Y |
| Server-Side Limited API Compatibility | | | |
| Stable | N | Y | Y |
| Evolving | N | N | Y |
| Unstable | N | N | N |
| Dependency Compatibility | N | Y | Y |

# Goals: Compatibility

- Grey areas…
  - Coprocessors
    - Free access HBase core
    - Change to hbase internals => broken Coprocessors
    - `InterfaceAudience.LimitedPrivate`
  - Published metrics/jmx
  - Protobufs
    - `hbase-protocol/hbase-protocol-shaded`

# Goals: Compatibility in 2.0.0

- **We adhere to SemVer for DML in 2.x**
  - Not for DDL
- hbase-1.x client can work against hbase-2.x cluster
  - Even 1.x Coprocessor Endpoints work on an hbase-2.x cluster
  - Read-only DDL/Admin of hbase-2.x from hbase-1.x client
  - Replication 1 ⇔ 2 works
- Extensive curation of what is public/private
- Purged Guava/Protobuf from API
- Coprocessors
  - Revamped
- No Singularity! No downtime! Rolling upgrade from hbase1!
  - Experimental! From 1.4.x to 2.1.x has been tested.

# Goals: Compatibility

- Still plenty to do
  - Ongoing effort...
  - 3.0.0!

# Goals: Others

- Scale
    - More Regions, bigger clusters
- Performance
    - Inline read/write but also macro-aspect: restart, assign, etc.
    - Better resource utilization
        - I/O, RAM
- Fix primary root of operational woes/bugs
    - Master Region Assignment

2.0.0

# Insides

- Currently >4500 issues resolved
  - ~3k exclusive to 2.0.0+

# Insides: Prerequisites

- JDK8 only
- Hadoop-2.7.7 minimum*
  - Works against the coming Hadoop-3.x

*Be wary of “*...not stable / production ready*” Hadoops

# Insides: Features

- New Master Core (A.K.A *AMv2*)
- Off-heap Read/Write path
- In-memory Compaction (“Accordion”)
- And more...

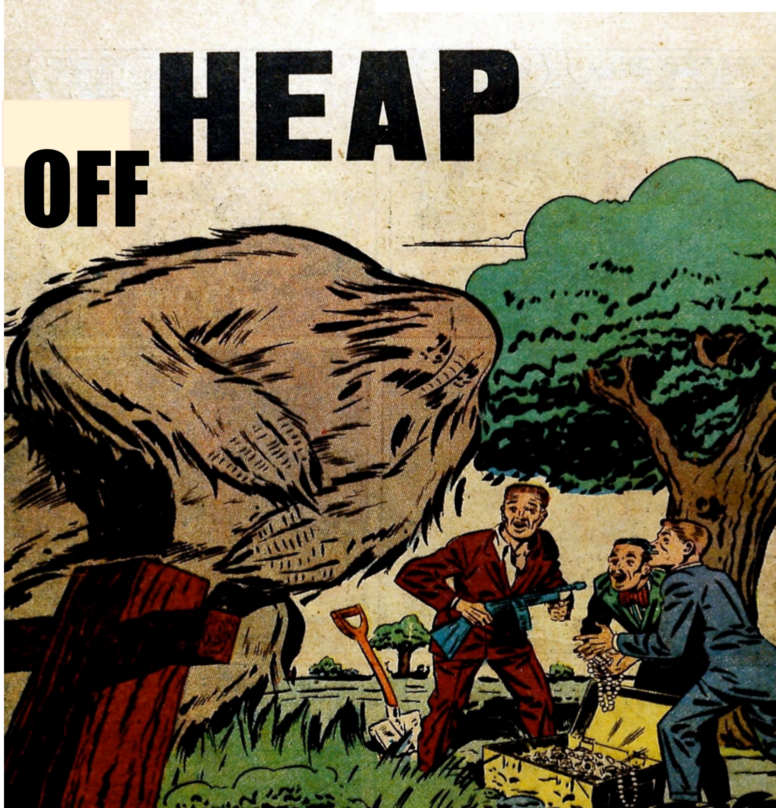
# Insides: Assignment Manager

- New Master Core (A.K.A AMv2)
  - Assignment Manager v1 (AMv1) root of many operational headaches
- Prompt assign of millions of Regions, faster startup, larger scale
- Scrutable/Standalone Testable
- One *hbase:meta* writer only, the Master
- No more intermediate state in ZK
  - At other end of an RPC...
  - Only final state published to *hbase:meta*
  - No more distributed state: some in Master memory, some in ZK, some in HDFS.
- New degree of ***Resilience***

# Insides: Off-heap

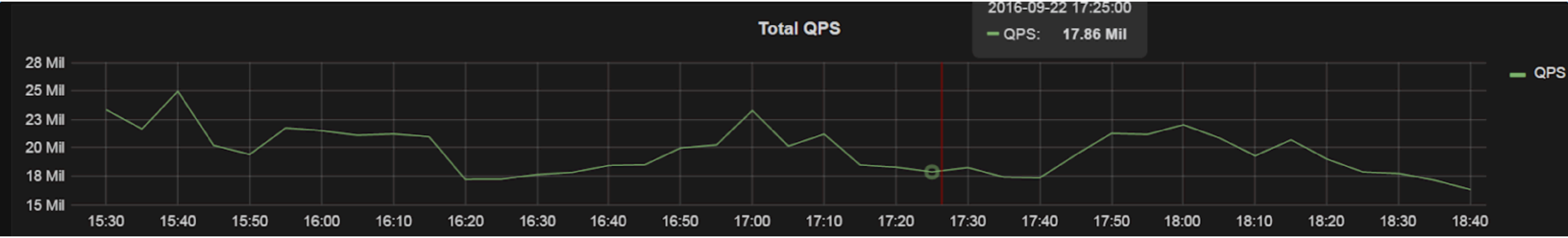
2.0.0

- Smaller JVM heaps, less copying
    - But more accounting!
- Off-heap Read Path
    - HDFS=>BucketCache=>Outbound Socket
    - ~latency
        - Cache more
        - Less GC, less erratic
- Off-heap Write Path
    - RPC=>HDFS data kept off-heap
        - Async DFS WAL Client
- Off-heap
    - Socket ⇔ Socket
    - Off-heap fragmentation anyone?
    - On by default?

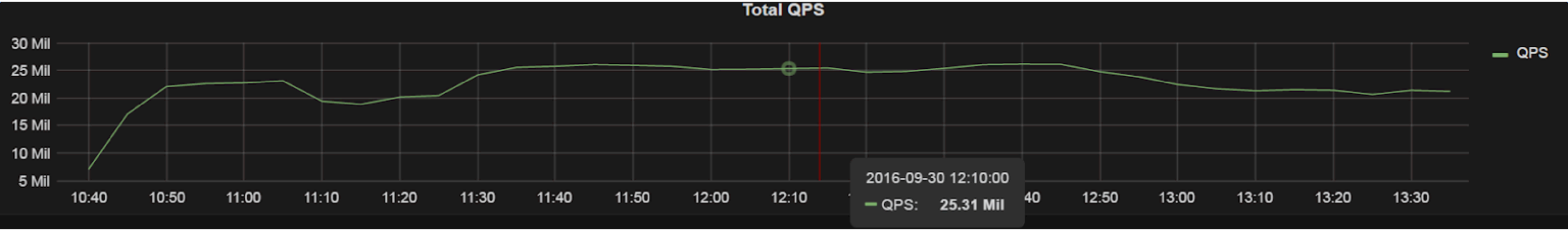


# Insides: Offheap

Before:



After:



# Insides: Accordion

2.0.0

- In-memory LSM
  - In-memory flush from ConcurrentSkipListMap to read-only pipeline of ‘Segments’
    - Optional in-memory compaction of Segments
      - Prune deletes/versions early
      - Benefit depends on # of versions
    - Memory ‘breathes’ like an Accordion’s bellows…
    - Less flushing, write amplification
- Flavors:
  - NONE, BASIC, ENHANCED, ADAPTIVE
- Symbiosis w/ Offheaping
  - MemStore SLABs off-heap
  - Compact in-memory layout
    - CellChunkMap, NOT CSLM

# Insides: Miscellaneous

2.0.0

- New Netty Server/Client chassis
- Filesystem Quotas on Table/Namespace sizes
  - Per Table, per Namespace -- not per User
  - Violation Policy
  - Via Shell
- Medium Object Blobs (MOB)
  - Documents, Images, etc.
  - 100k => 10MB
  - Consistency, security, snapshot and HBase replication between clusters

# Insides: Miscellaneous

- Async DFSClient
  - Fanout, fail-fast
  - Less resources
- New Async Client
  - NOT asynchbase
  - Complete, Admin ops too.
- Update+Relocation of core dependencies
  - `hbase-thirdparty`
    - Guava 0.12 => 0.22
    - Protobuf 2.5 => 3.3
    - Netty

# Insides: Miscellaneous

- RegionServer Groups (rsgroups)
  - Coarse Isolation
  - Tables x RegionServers
- WALs and HFiles in different Filesystems
  - AWS EMR HBase on S3 offering  (Amazon S3 Storage Mode)

# Insides: Miscellaneous

2.0.0

- HBASE-14061 Support CF-level Storage Policy
- HBASE-17408 Introduce per request limit by number of mutations (ChiaPing Tsai)
- HBASE-17320 Add inclusive/exclusive support for startRow and endRow of scan
- HBASE-17174 Refactor the AsyncProcess, BufferedMutatorImpl, and HTable
- HBASE-16010 Put draining function through Admin API (Matt Warhaftig)
- HBASE-17262 Refactor RpcServer so as to make it extendable and/or pluggable
- HBASE-17282 Reduce the redundant requests to meta table
- HBASE-15432 TableInputFormat - support multi column family scan (Xuesen Liang)
- HBASE-17313 Add BufferedMutatorParams#clone method (Joep Rottinghuis)
- HBASE-17277 Allow alternate BufferedMutator implemenation Specify the name of an alternate BufferedMutator implementation by either:
- HBASE-16700 Allow for coprocessor whitelisting
- HBASE-17194 Assign the new region to the idle server after splitting (ChiaPing Tsai)
- HBASE-17110 Improve SimpleLoadBalancer to always take server-level balance into account
- Revert "HBASE-15513 hbase.hregion.memstore.chunkpool.maxsize is 0.0 by default (Vladimir Rodionov)"
- HBASE-16733 add hadoop 3.0.0-alpha1 to precommit checks
- HBASE-13259 mmap() based BucketCache IOEngine
- HBASE-16447 Replication by namespaces config in peer
- ***HBASE-15968 New behavior of versions considering mvcc and ts rather than ts only***
- ***HBASE-15816 Provide client with ability to set priority on Operations***
- HBASE-16213 A new HFileBlock structure for fast random get
- HBASE-14004 [Replication] Inconsistency between Memstore and WAL may result in data in remote cluster that is not in the origin
- HBASE-10367 RegionServer graceful stop / decommissioning
- ***HBASE-16290 Dump summary of callQueue content; can help debugging***
- HBASE-14790 Implement a new DFSOutputStream for logging WAL only
- HBASE-18410 FilterList Improvement
- ***HBASE-8329 Limit compaction speed -- throttling is enabled by default.***
- HBASE-14969 Add throughput controller for flush
- HBASE-19358 Improve the stability of splitting log when do fail over
- HBASE-18882 [TEST] Run MR branch-1 jobs against hbase2 cluster
- ...

# How Coprocessors Changed

- Pass Interfaces instead of Implementations
  - TableDescriptor instead of HTableDescriptor and Region instead of HRegion
  - Implementations can change w/o breaking CPs
- Refactor so less boilerplate, more compile-time checking.
- Some operations changed locus
  - E.g. splits are done by Master in hbase2, RegionServer in hbase1.
- Purge of PB from CP API
  - Passed raw Protobufs to Coprocessors. Pass POJOs/Interfaces instead.
- CP API Pruning
  - Removed hooks on internals that were too private to expose.
  - Also hooks that allowed Coprocessor do “crazy” stuff: e.g. create own StoreFiles
- Deprecated methods have been removed

# Upgrading to 2.0.x

- Read Upgrading Chapter in refguide
  - Includes list of notable changes…
    - Changed defaults, namings, etc., based off user feedback
    - Configs/metrics misspellings fixed, some renames
  - Removals
    - DLR
    - prefix-tree encoding
    - Deprecated pre-1.0.0 features and classes
  - Coprocessors Upgrade Recipe
    - Rebuild against new API otherwise they will fail to load and HBase processes will die
- New CHANGES.md and RELEASENOTES.md (courtesy of Apache Yetus).
- Before upgrade, run Pre-Upgrade Validator
  - `$ bin/hbase pre-upgrade …`
- ~~HBCK1~~

# Goals: Others, revisited

- Scale
  - More Regions, bigger clusters
- Performance
  - Inline read/write but also macro aspect: restart, assign, etc.
  - Better resource utilization
    - I/O, RAM
- Fix primary root of operational woes/bugs
  - Master Region Assignment

- 2.0.x
  - Bug-Fixing
    - AMv2 Corner cases, teething...
  - Perf
    - Ongoing
  - Time-based releases
    - Every 6-weeks or so…
    - Just stabilizing…
- HBCK2
  - New standalone project

# hbase-2.1.x

# 2.1.x

- Duo Zhang, Release Manager
- Pv2 based replication peer modification
  - Operation will be synchronous
- Serial Replication
  - Preserve edit order crossing the chasm
- Sensible-looking CLASSPATH for your client
- Lots of AMv2 fixes...

# Future (3.0.0+): Ease of Use

- Native SQL support
  - Internal
    - Lightweight SQL support
      - Something like CQL (Apache Calcite, Presto, Apache Derby?)
    - Official Secondary-Index
      - Better implementation with Procedure v2
  - External
    - Better integration with Spark SQL
      - An official spark connector in the hbase project
    - Phoenix
      - SQL-tier over HBase
- Configuration
  - Complete dynamic online configuration change; no need of a rolling restart.
  - Templates and Auto-tuning; ergonomic adjustment to suit current load.

# Future (3.0.0+): Performance

- Aim at making the best Java LSM storage engine
- E2E asynchronous
    - Since 2.0.0, we have async client, async WAL
        - Async HDFS API underway
    - We will have: SEDA request handling on server side
        - Better write pipeline
        - Complete at Alibaba, upstream coming soon
    - E2E asynchronous is coming...
- New Cell format
    - Plumbed end-to-end
    - Better-integrated `sequenceid` & Cell-*Tags*
    - More CPU-cache friendly

# Future (3.0.0+): Scale and Stability

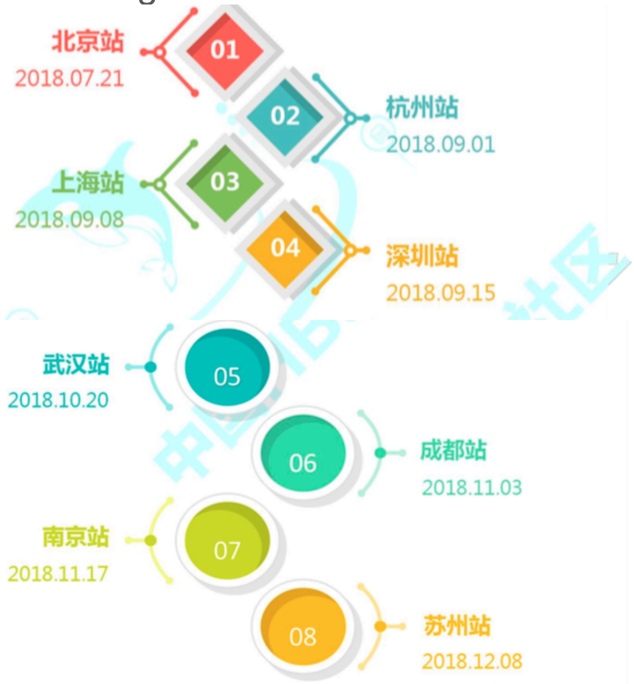
- Split meta table
  - Distribute i/o and so we can store more metadata
- New Distributed Log Replay redo
- Clusters with 1M/10M regions
  - Smaller regions, less write amplification

# Future (3.0.0+): Internal

- Framework Refactor
  - Distribution Management Framework
    - Procedure V2 polish
  - SEDA framework
  - Core abstraction
    - Support “embedded” mode
    - Support other storage engine like rocksdb
  - FileSystem Interface Redo
    - Bring into hbase atomic operations like rename rather than ask HDFS do it for us
      - Eliminate trips to NameNode
    - Remove dependency on HDFS, better support Cloud/S3
- Hybrid Logical Clocks
- Externalized Compaction
  - Spark

# Future (3.0.0+): Join Us!

- More help needed
  - We need more contributors/committers to make the bright future!
- Don’t let language block you
  - PMC would like to see world diversity
  - A Chinese HBase group already setup
  - More HBase China meetups coming
    - Hangzhou/Shanghai/Shenzhen...
- Don’t be shy
  - Why not upstream your great work?
  - What’s in your way?
  - Just let us know
- Don’t hesitate
  - Join us!



Thank you

Images:

- Insides: http://alienanthology.wikia.com/wiki/File:Ash%27s_insides.png
- Goal: https://commons.wikimedia.org/wiki/File:Goal-com-logo-eps-vector-image.png
- Future: https://thenextweb.com/insider/2015/12/30/artificial-intelligence-free-money-and-the-end-of-capitalism/
- Sea: https://playgroundforlife.files.wordpress.com/2010/08/gerhard-richter-seascape-1998-sfmoma.jpg