State of the Database

@HBase
http://hbase.apache.org

Nick Dimiduk (@xefyr)
http://n10k.com

2015-09-28

#apachebigdata

Agenda

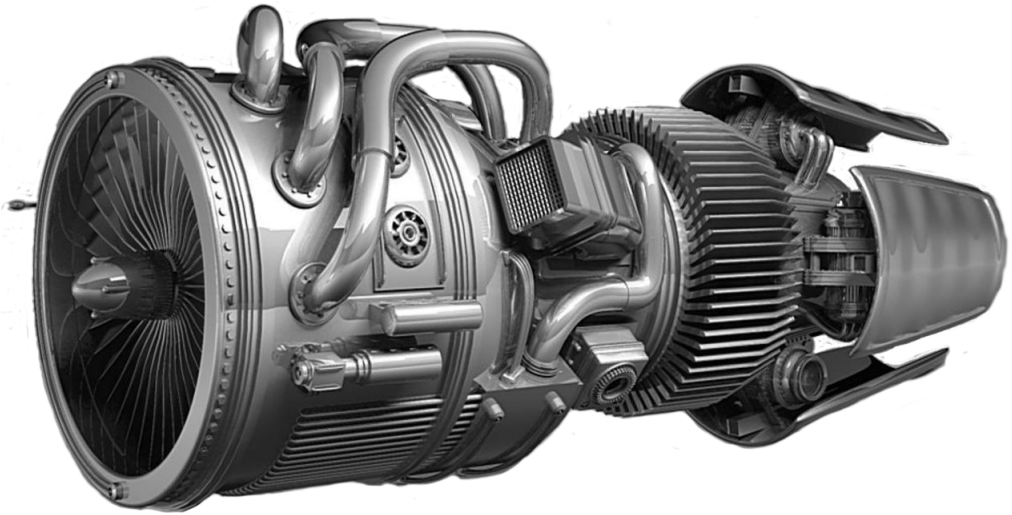
- State of the Project
- State of the Software
- State of the Ecosystem
- Latest Releases
- Bonus Content!
- Q & A

Who we are, what we do, why we do it

STATE OF THE PROJECT

Project: Vision

Simple, steady, and powerful: “A first class high performance horizontally scalable data storage *engine* for Big Data, suitable as the store of record for mission critical data.”



Hortonworks®

APACHE HBASE

Project: Usage

- Data access for medium- and high-scale services
  - Hundreds of enterprises and startups
  - Some of the largest Internet companies in the world
- Running major production workloads since 2011
- Use-cases
  - messaging, security, measurement/“IoT”, collaboration, digital media, digital advertising, telecommunications, computational biology, clinical informatics/healthcare, insurance


salesforce
SOFTWARE
Pacific Northwest
NATIONAL LABORATORY
OCLC
AOL
StumbleUpon
Dropbox
explorys
ebay
facebook
TREND MICRO
rocket fuel
GROUPON
NOKIA
mozilla FOUNDATION
YAHOO!
runa
APACHE HBASE
Adobe
jive
OPOWER
FINRA
Bloomberg
HUAWEI
FLURRY
photobucket
NEXTBIO
intuit
KLOUT
GAP
Urban Airship
NG DATA
box
Cerner
tumblr.

Project: Goals

- Availability: Always more, always faster
- Stability and operability
- Scaling up, scaling down
- Up-to-date with “commodity” hardware
- Multi-tenancy
- Diversity of ecosystem

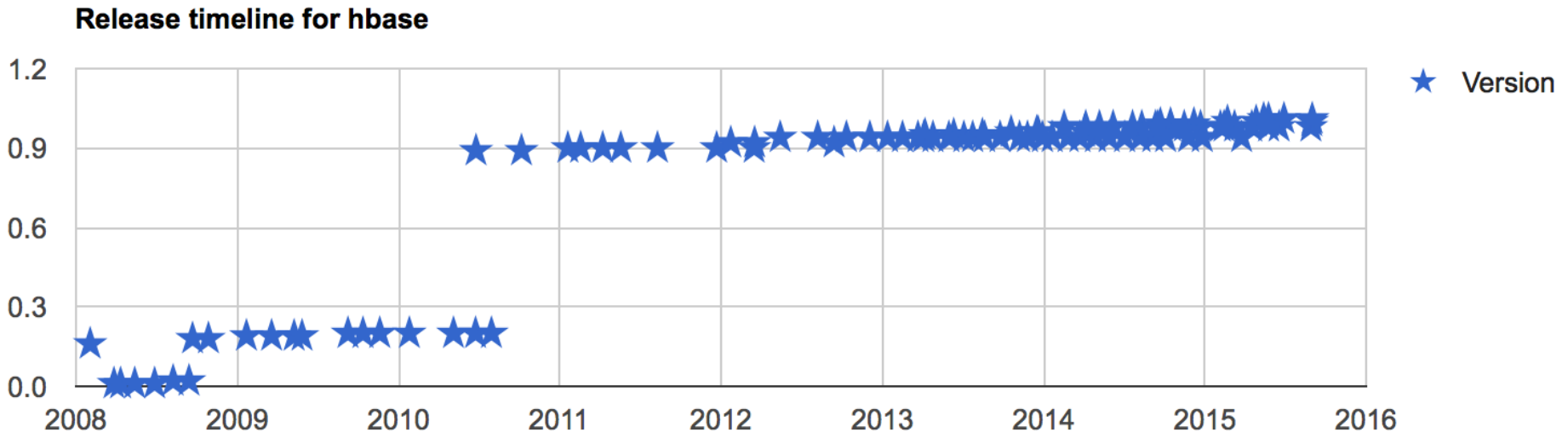
Regarding the codebase

STATE OF THE SOFTWARE

State of the Software

- Mature codebase
  - 100+ contributors (40+ committers)
  - 1.1M lines of code (each active branch)
  - est. 1200+ human-years' effort
- Clusters sizes from 10 to 1000+ machines
  - that we know of!
- Runs on HDFS, MapR, Gluster, GPFS, Lustre
- HBase as a Service
  - AWS/EMR, HDInsight, Qubole, Google (sort-of)

Software: Releases



Software: Semantic Versioning

MAJOR-MINOR-PATCH[-identifier]

- Client/Server wire compatibility
- Server/Server feature compatibility
- API compliance guarantees
- ABI compliance guarantees

http://hbase.apache.org/book.html#hbase.versioning

Software: Active Development

- Smaller regions, more regions
  - Less write amplification
  - 1M+ region clusters
- Stability
  - ProcedureV2
  - Assignment improvements/stability
- Backup, restore tools
  - Built on snapshots, easier operations

Software: Active Development

- Adaption: Workloads
  - HBase as Medium Object Store (MOB)
- Tunable Availability
  - Region replicas
  - TIMELINE consistency
- Coprocessor API stability
- Less GC, more RAM (off-heap)

Software: Active Development

- Multi-tenancy
  - Table groups
  - Quotas
  - Priorities
- Improved machine utilization
  - More RAM (100's of GB)
  - IOPS
  - Better concurrency

The whole enchilada

STATE OF THE ECOSYSTEM

State of the Ecosystem

- OpenTSDB
- Transaction Managers
  - Themis, Tephera, Omid2, LeanXcale
- Graph engines
  - Titan, Giraph, Zen, S2Graph
- **Myriad SQL's**
- **Other Hadoop components**
- Google Cloud Bigtable

Ecosystem: SQL

Ecosystem: Hadoop Components

- YARN-2928 Application Timeline Service
- HIVE-9452 HBase to store Hive metadata
- AMBARI-5707 Ambari Metrics System

Come and get it!

LATEST RELEASES

Release: 0.94

- Last (final?) release: 0.94.27, 2015-03-26
- “ancient history”
  - No new deployments
  - Existing users highly encouraged to upgrade
- Requires downtime to upgrade

😫 😡 (ノ °□°)ノ ⌒ ┻━┻

Release: 0.98

- Last release: 0.98.14, 2015-08-31
- “legacy”
  - Most production deploys (probably)
  - Largest production clusters (probably)
  - New features back-ported when possible

Release 1.x

- Last release: 1.1.2, 2015-09-01
- “stable”
  - Production deploys moving here
  - Active development
- Rolling upgrade from 0.98.x

😄 😍 ヽ(´ー`)ノ

Release 1.0

- Released 1.0.0, 2015-02-24
- Adopting semantic versioning
  - Patch releases don't quite follow spec yet
- Client / Server API cleanup
  - Interfaces, builder pattern, @InterfaceAudience
- **Region Replicas**
  - Trade Consistency, resources for Availability

github.com/ndimiduk/hbase-1.0-api-examples

Region Replicas

- Multiple Region Servers host each region
  - Primary + N read replicas (usually N=2)
  - Primary is authority on reads and writes
  - Replicas tail replicate edits, offer TIMELINE view
- Client's choice
  - Read primary only for "classic" strong consistency
  - Fan-out reads for faster, potentially TIMELINE results

Release 1.1

- Release 1.1.0, 2015-05-15
- Async RPC client
- Scanner improvements
  - RPC chunking, heartbeat messages, API
- RPC throttling
  - quotas for per user, table, namespace
- Compaction throttling, monitoring
- **ProcedureV2**
  - Improved operational reliability

ProcedureV2

- Distributed, fault-tolerant operations
  - Multiple steps on multiple machines
  - Roll-back in case of failure
- Coordination of long-running procedures
  - Compactions, splits, &c.
- Progress tracking
  - Notifications across multiple machines
  - Current status inquiries

Branch-1.2

- Next up in 1.x line
- Java 8 support
- Native checksums
- SyncTable
- Flush-per-store
- ProcV2 all the things!
- (More) Compaction improvements
- **Region normalizer**

Region Normalizer

- Anti-entropy for region size
  - Converge towards uniform size
  - Compliments balancer working toward uniform distribution
- Managed by Master, runs in the background (like balancer)
- Pluggable normalization strategies ("simple" default)
- Use-cases
  - Merge away regions from expired timeseries data
  - Smooth uneven bulk loads
  - Correct operator initial split guesses
  - Ease upgrades from ancient versions (0.92/1g vs. today/20g)

Thanks!

@HBase

http://hbase.apache.org

2015-09-28

Nick Dimiduk (@xefyr)

http://n10k.com

#apachebigdata

Ask and you shall receive

BONUS CONTENT!

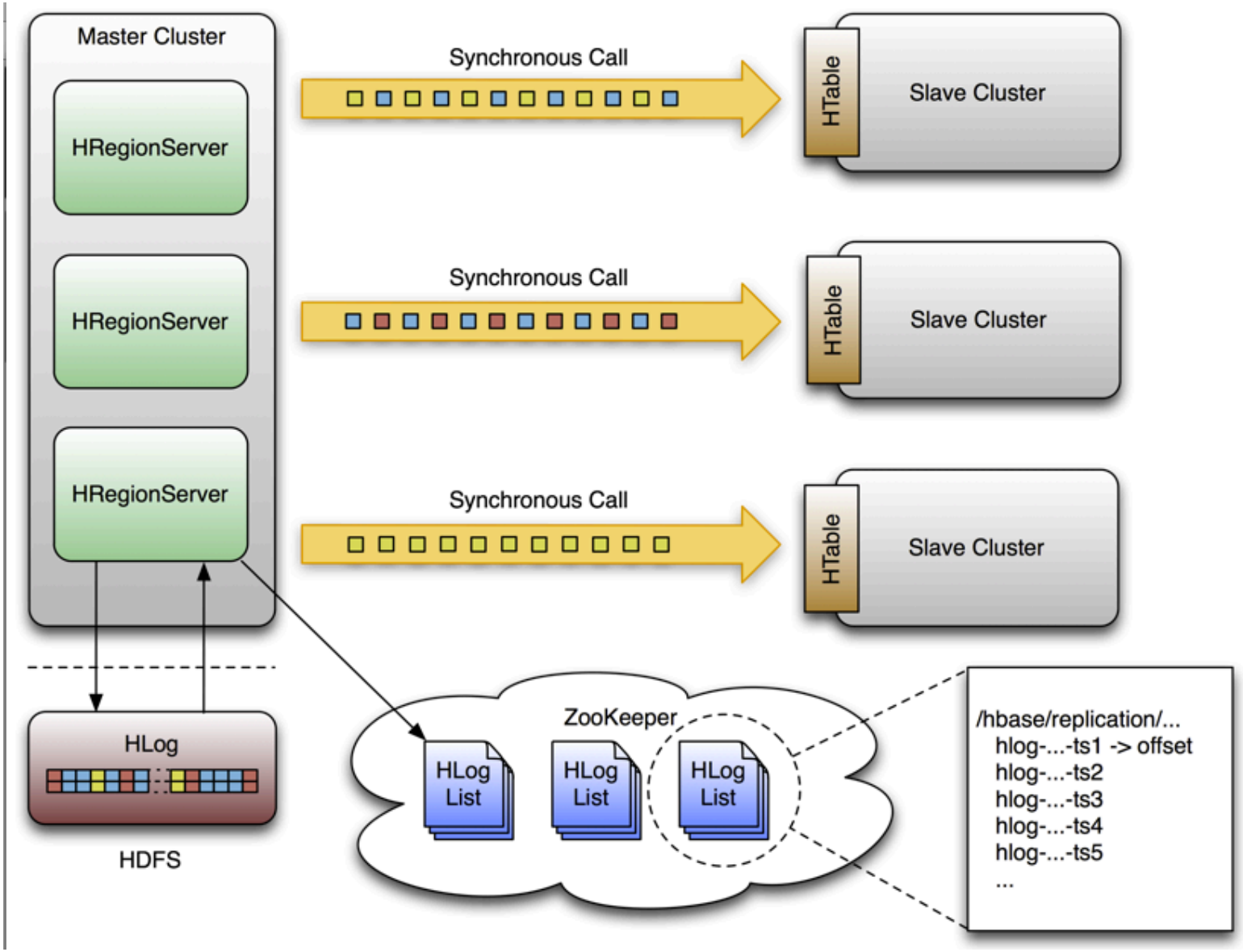
Agenda

- Replication
- Filters
- Coprocessors

Replication

- Keep data synchronized between clusters
  - Supports **multiple destinations**
  - **Cyclical graphs** supported
  - Configurable at **Column Family** granularity
- Uses WAL shipping to propagate data
- Replication state, status stored in ZooKeeper
- General purpose interface for asynchronously shipping edits from a cluster
  - Other HBase clusters, Region Replicas, SOLR/ElasticSearch

hbase.apache.org/book.html#_cluster_replication


Master Cluster
HRegionServer
HRegionServer
HRegionServer
Synchronous Call
Synchronous Call
Synchronous Call
HTable
Slave Cluster
HTable
Slave Cluster
HTable
Slave Cluster
HLog
HDFS
ZooKeeper
HLog List
HLog List
HLog List
/hbase/replication/...
hlog-...-ts1 -> offset
hlog-...-ts2
hlog-...-ts3
hlog-...-ts4
hlog-...-ts5
...

Filters

- Additional applied to reads
  - Use in conjunction with specifying start, end rows, &c.
- Run on the Region Servers
  - Included in GET, SCAN request
- Explicitly exclude data based on criteria
  - I.E., value >= 10
- Implicitly exclude data by hinting seeks
  - INCLUDE_AND_NEXT_COL, NEXT_ROW, SEEK_NEXT_USING_HINT
- Operate on data read from BlockCache

Filters

- 30+ Filters included in distribution
- Mini-language for use in Thrift, REST
  - `"(PrefixFilter ('row2') AND (QualifierFilter (>=, 'binary:xyz'))) AND (TimestampsFilter ( 123, 456))"`
  - hbase.apache.org/book.html#thrift.filter_language
- Simple interface, Implement your own!

```
public class PageFilter extends FilterBase {
  public PageFilter(long pageSize) {…}

  public boolean filterRowKey(Cell c) {
    return false;
  }

  public ReturnCode filterKeyValue(Cell c) {
    return ReturnCode.INCLUDE;
  }

  public boolean filterAllRemaining() {
    return this.rowsAccepted >= this.pageSize;
  }

  public filterRow() {
    this.rowsAccepted++;
    return this.rowsAccepted > this.pageSize;
  }
}
```

Coprocessors

- Extension points for HBase
  - Think Linux Kernel Module, not Stored Procedure
  - I.E., customize compactions, Table constraints
- Observers
  - pre- and post-execution logic
  - I.E., MasterObserver#preTruncateTable, RegionObserver#postScannerNext
- Endpoints
  - Cluster RPC extensions
  - I.E., RowCountEndpoint, BulkDeleteEndpoint

```
public class RowCountEndpoint implements
    ExampleProtos.RowCountService {
  public void getRowCount(…) {
    Scan = new Scan();
    InternalScanner scanner =
      env.getRegion().getScanner(scan);
    …
    do {
      count++;
    } while (scanner.next());
    // return count
  }
}
```

Thanks!

@HBase

http://hbase.apache.org

2015-09-28

Nick Dimiduk (@xefyr)

http://n10k.com

#apachebigdata