HBase在阿里搜索领域的应用与扩展

DTCC

2015中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2015

大数据技术探索和价值发现

IT168.com ChinaUnix ITPUB

Self-introduction

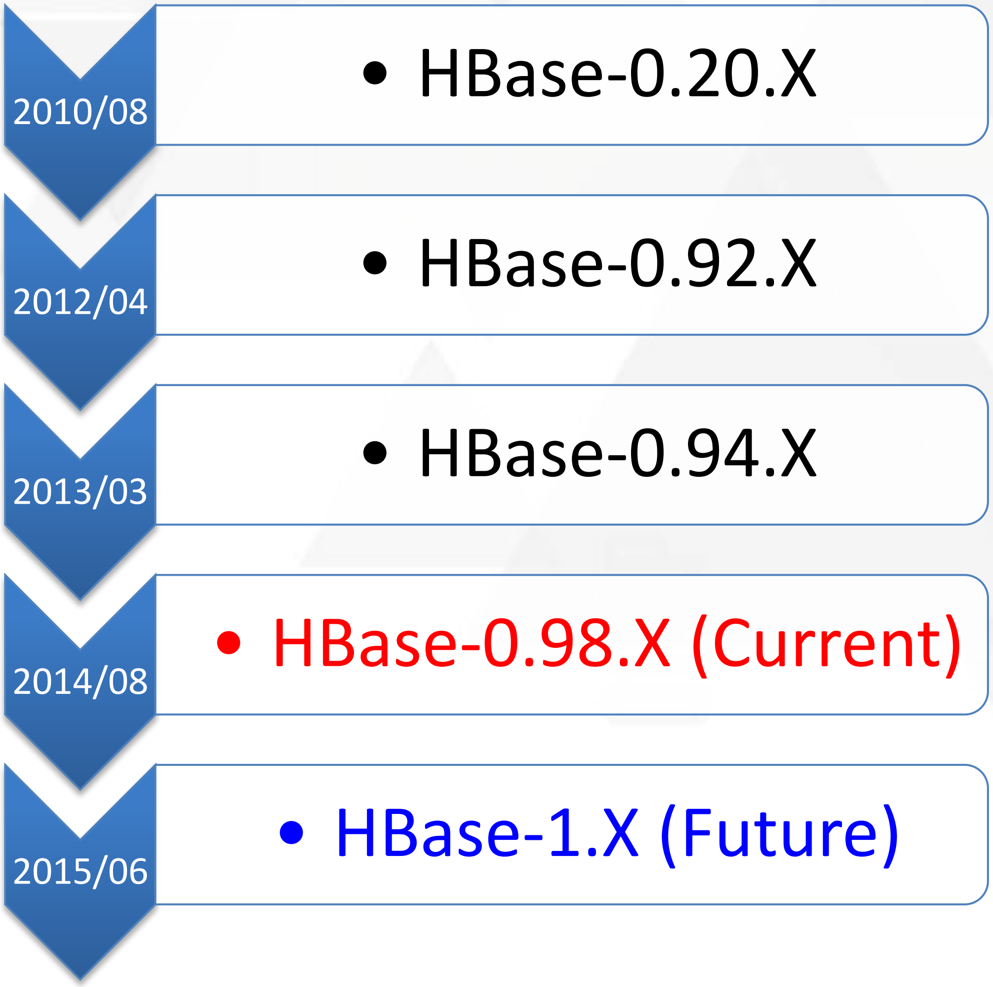
- 徐斌，花名“雨田”，阿里巴巴搜索事业部，搜索研发专家
- 2009年，本科毕业于华中科技大学，软件工程专业
- 主要工作领域：搜索抓取系统，搜索存储平台
- 微博：淘宝雨田
- 邮箱：yutian.xb@taobao.com

Agenda

1. HBase in Ali-Search
2. Improvements & Maintenance
3. Extensional Projects
4. Future
5. Q & A

HBase in Ali-Search

Upgrade History



2010/08
- HBase-0.20.X

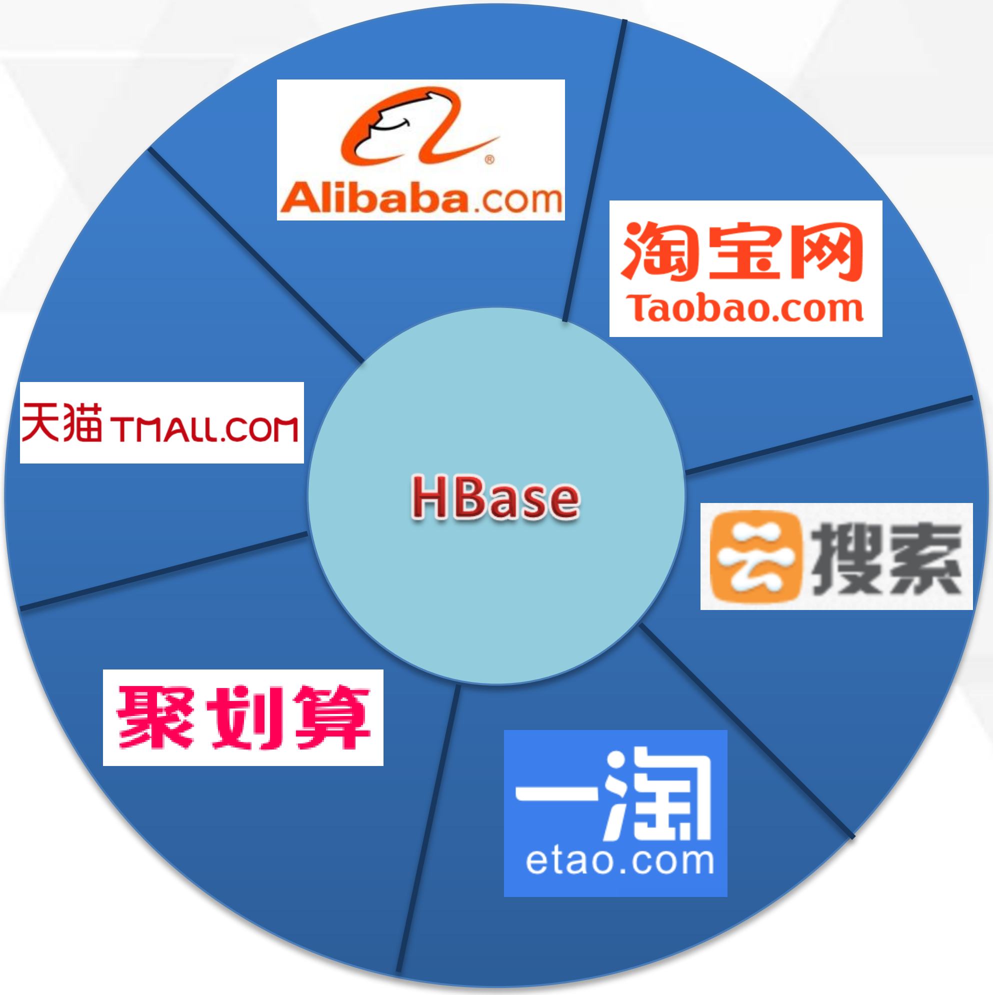
2012/04
- HBase-0.92.X

2013/03
- HBase-0.94.X

2014/08
- HBase-0.98.X (Current)

2015/06
- HBase-1.X (Future)

Who is using HBase?

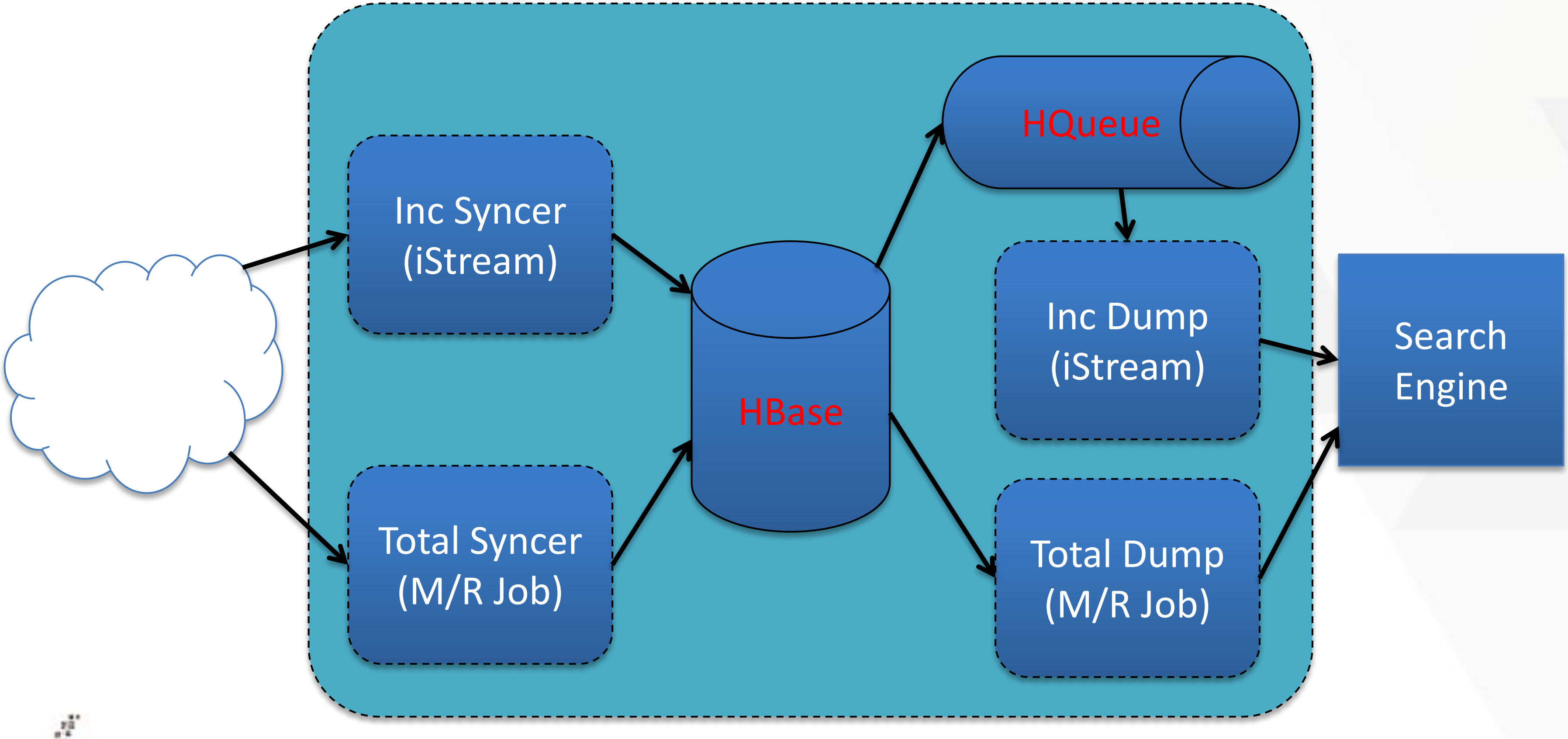


HBase

Scenario 1 – Taobao Search

Taobao data: billions of items

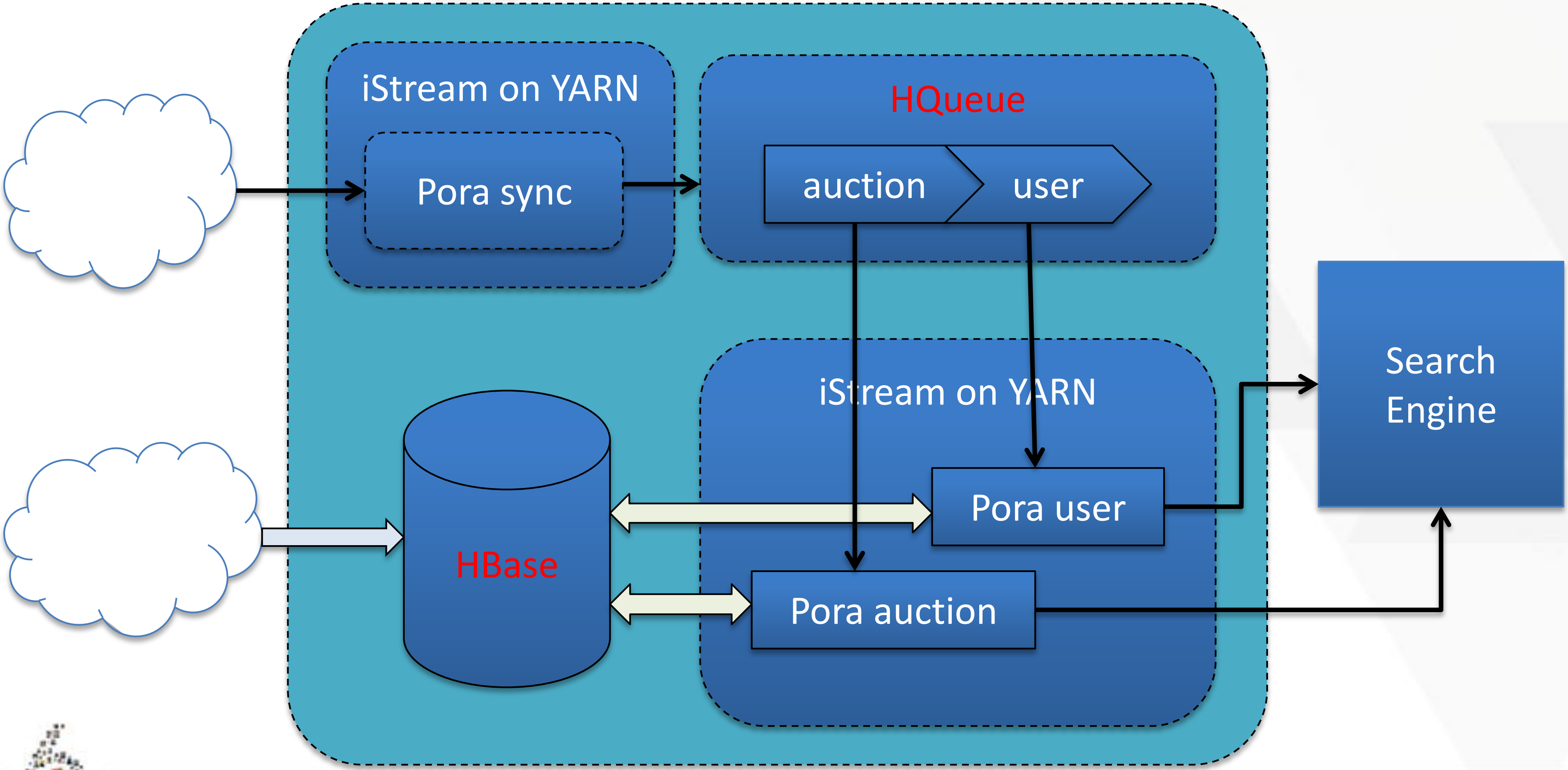
Tmall data: hundreds of millions of items



Scenario 2 – PORA

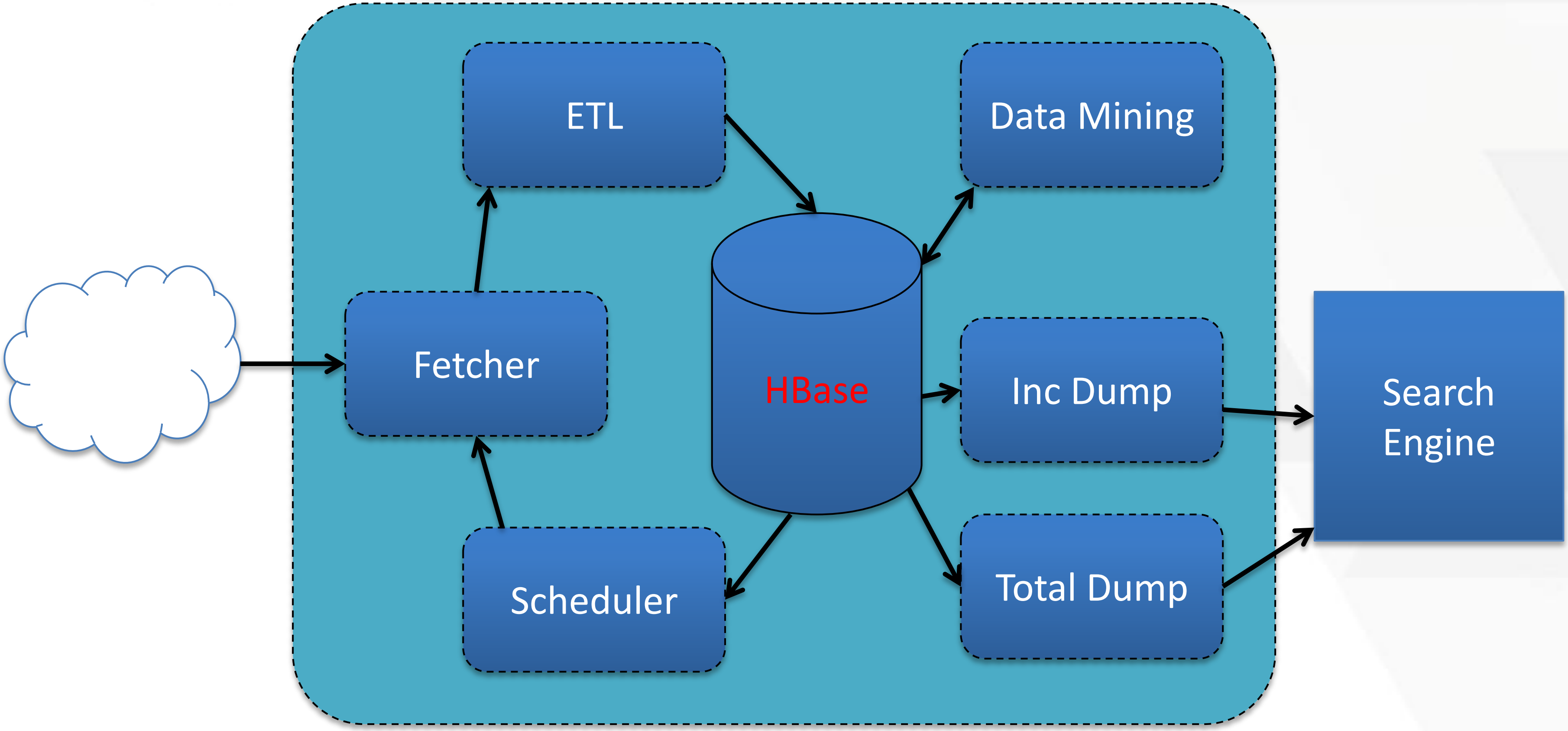
PORA: Personal Offline Real-time Analyze

User log data: tens of billions of records per day



Scenario 3 – Web Crawling

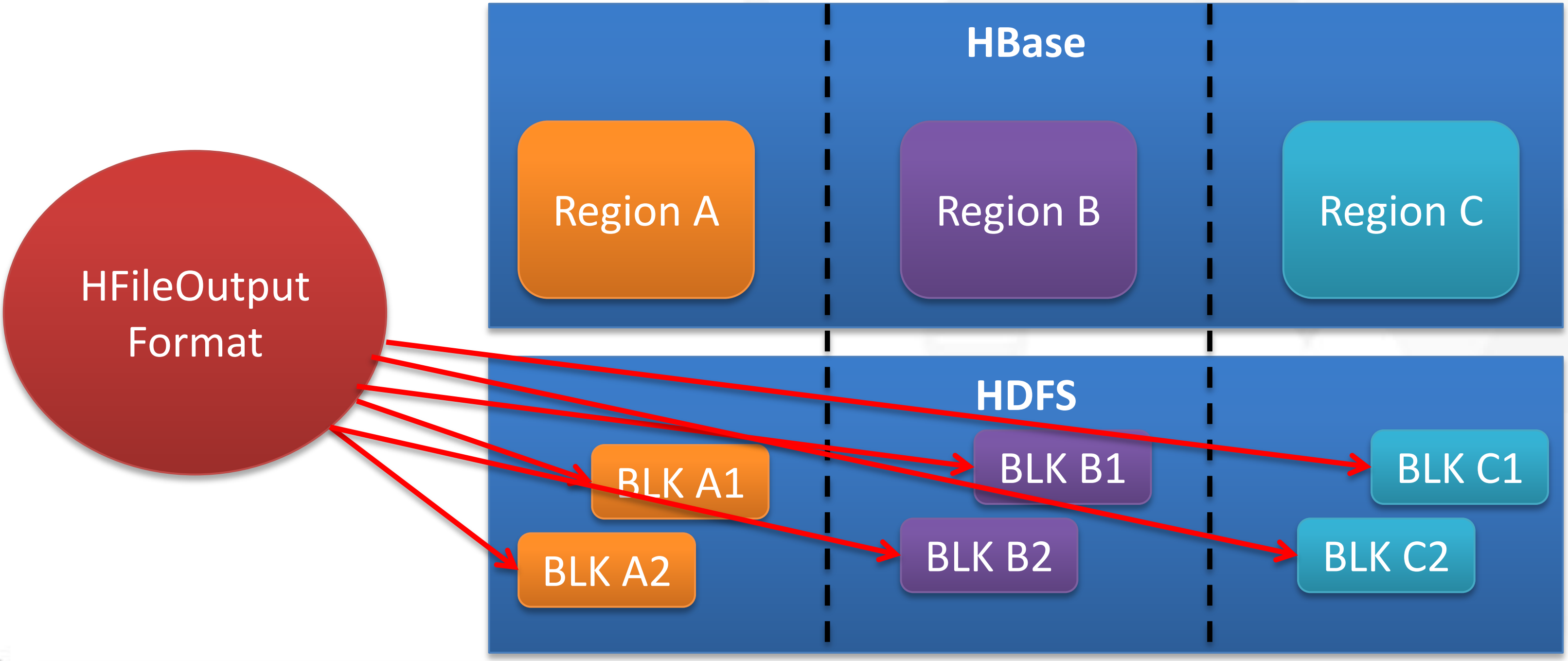
Web data: tens of billions of pages



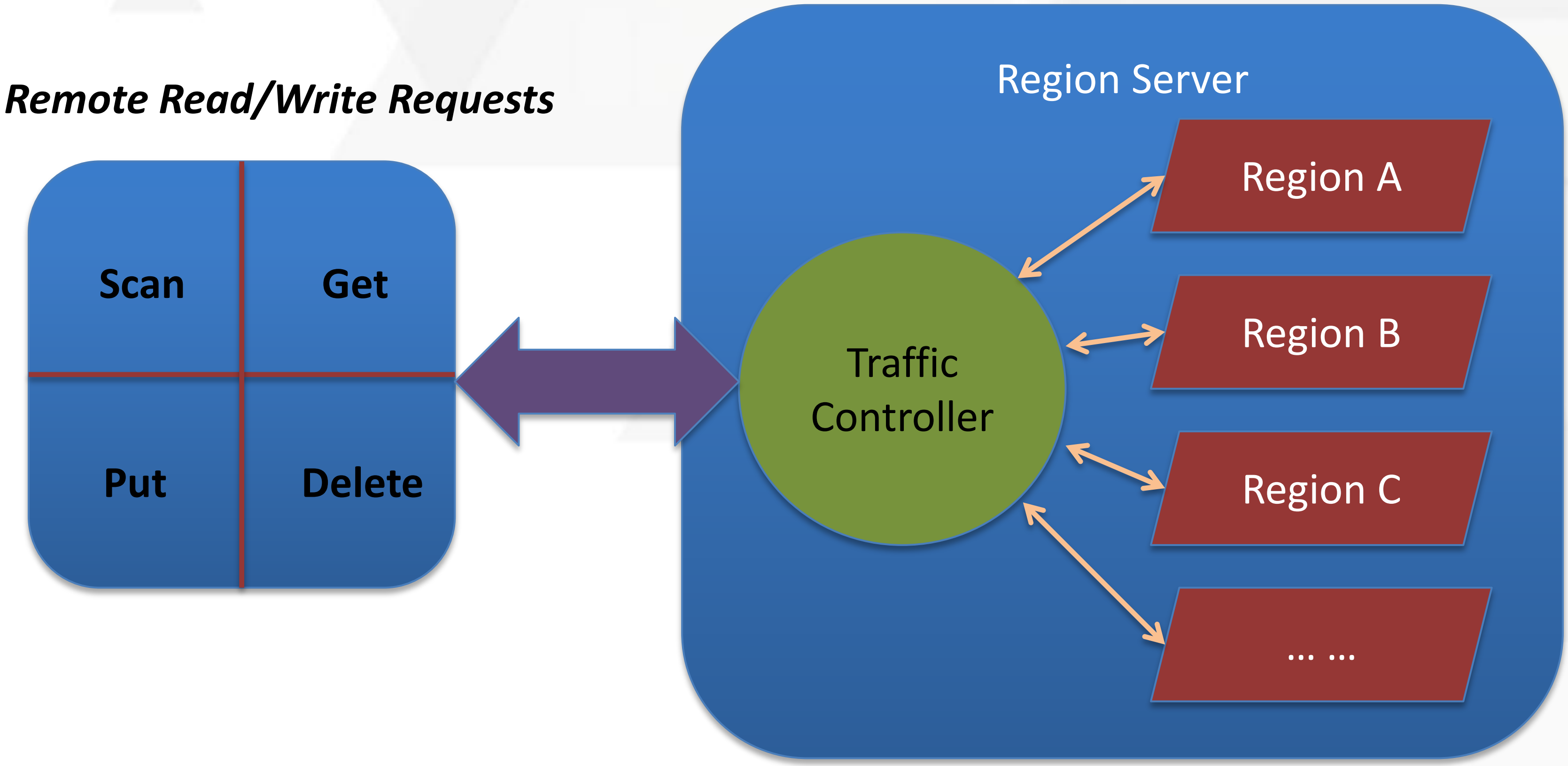
Improvements & Maintenance

Lower Disk I/O

Generate HFile directly onto the node which holds the HDFS replica of target region, then bulkload it with high locality, saving the I/O of major compaction. ([HBase-12596]())

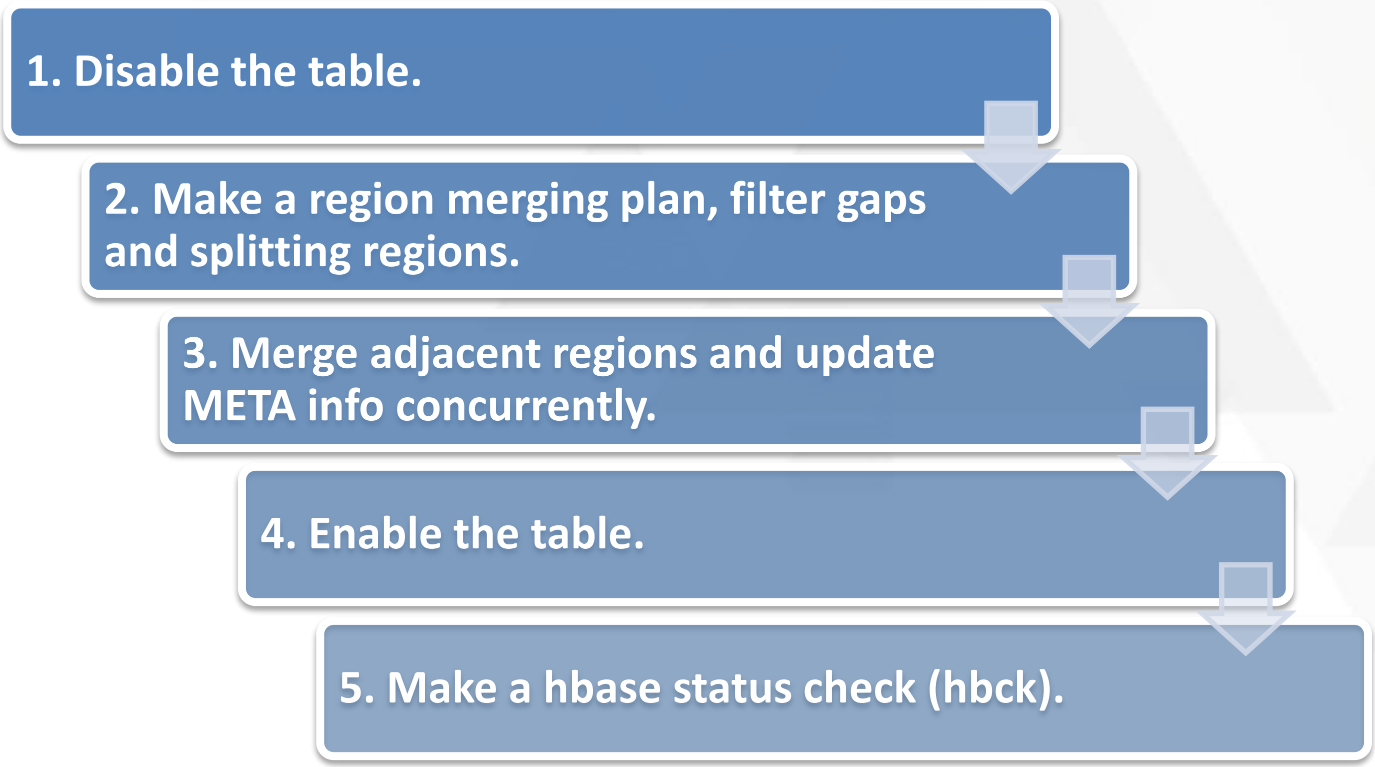


Limit Bandwidth



Offline Region Merge

Online region merge mechanism is so slow that we need to find another way to merge thousands (maybe tens of thousands) of regions at a time.



1. Disable the table.

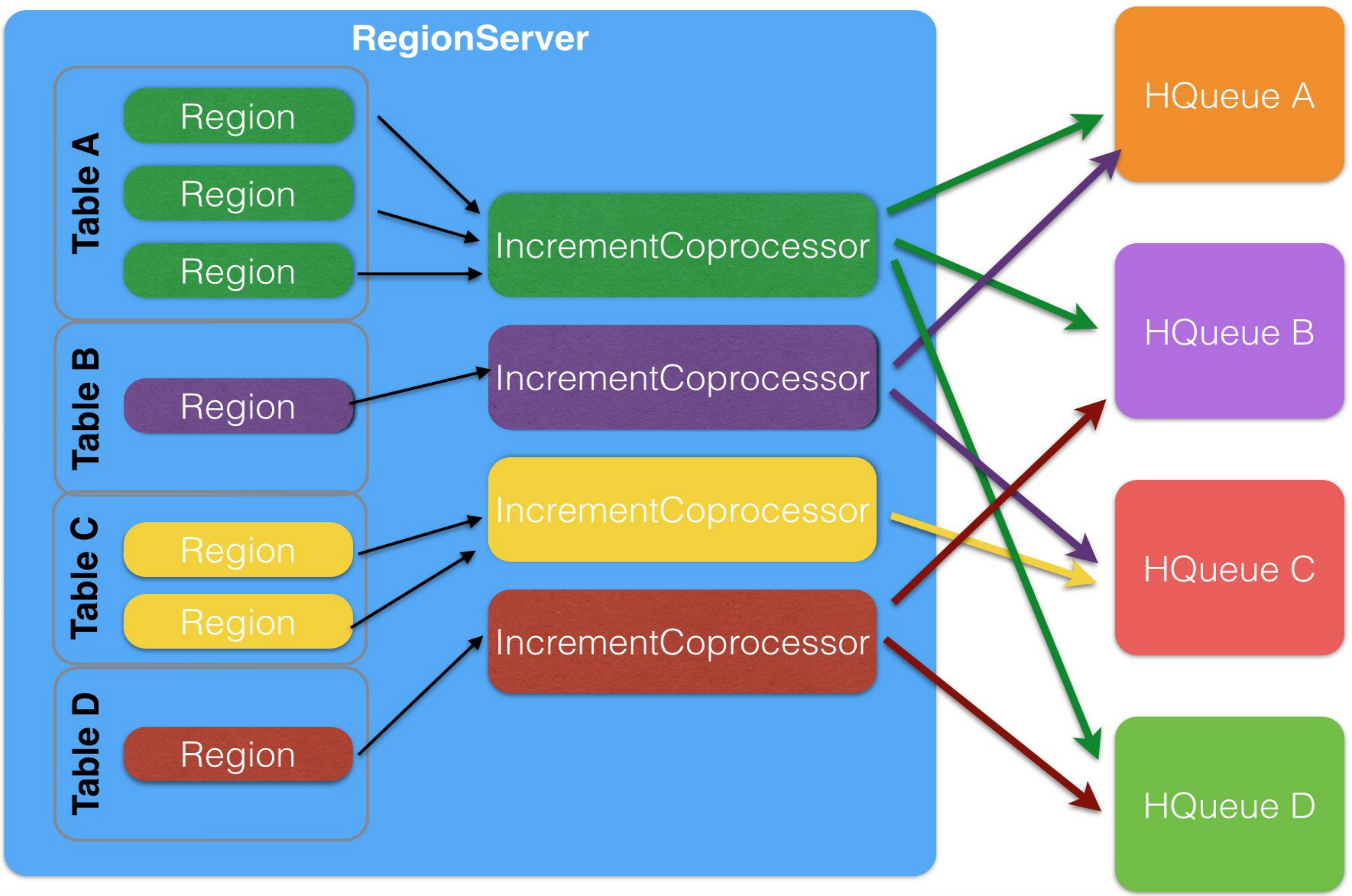
2. Make a region merging plan, filter gaps and splitting regions.

3. Merge adjacent regions and update META info concurrently.

4. Enable the table.

5. Make a hbase status check (hbck).

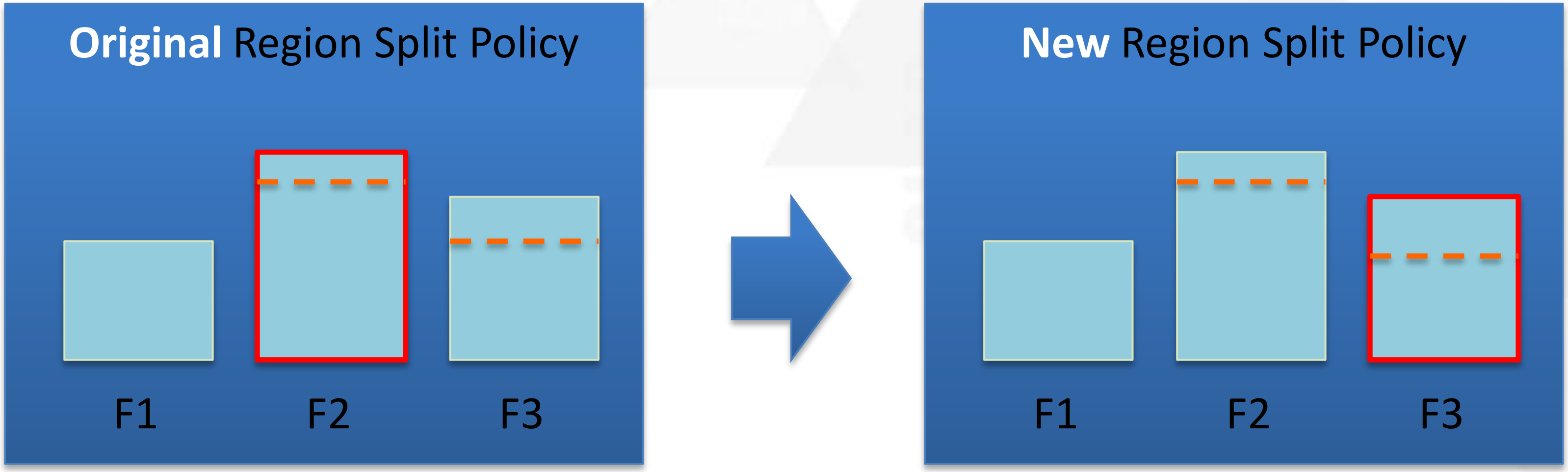
Incremental Trigger



Region Split Policy

- Set a constant limit for each family, if not, use the region max size limit instead. Region split will be triggered if any family reaches its size limit.
- The split point is determined by the family who exceeds the most proportion of its size limit.

For example:
SizeOf(F1) = 5M, SizeOf(F2) = 15M, SizeOf(F3) = 10M
LimitOf(F1)= 10M, LimitOf(F2) = 14M, LimitOf(F3) = 8M



Cluster Availability

The strategy we find and deal with sick Region Server.

- Phase 1: Region probe (< 3 min)
- Phase 2: Save logs (< 30 sec)
- Phase 3: RS shutdown (< 3 min)
- Phase 4: Force kill RS (< 30 sec)
- Phase 5: Phone alarm(< 30 sec)

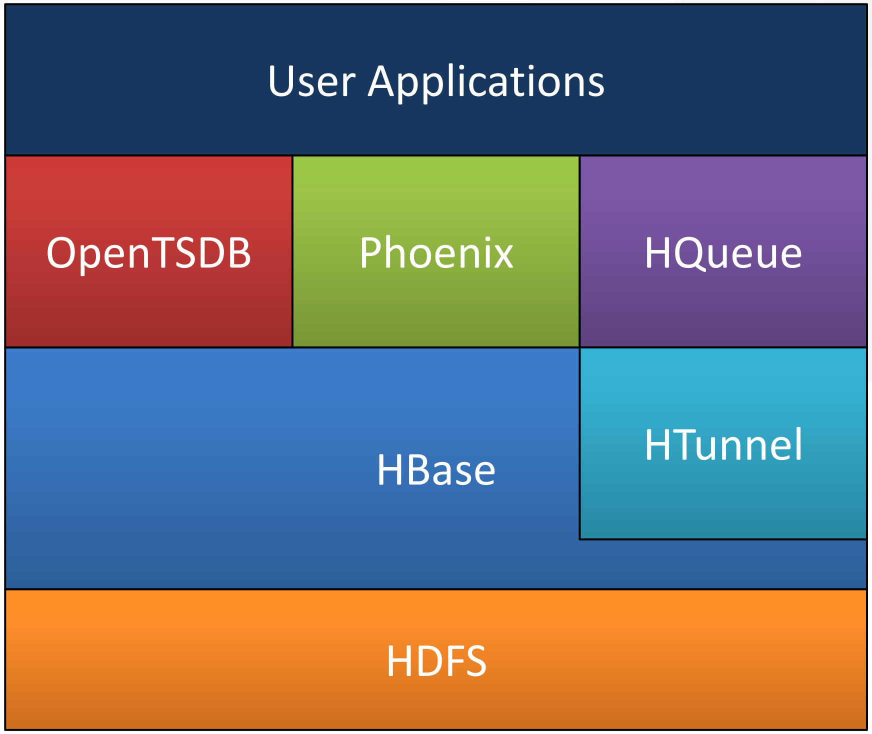
Other Optimizations

- Enhanced simple balance strategy
- Enhanced rolling upgrade
- Customized tableInputFormat
- More ganglia metrics for client requests

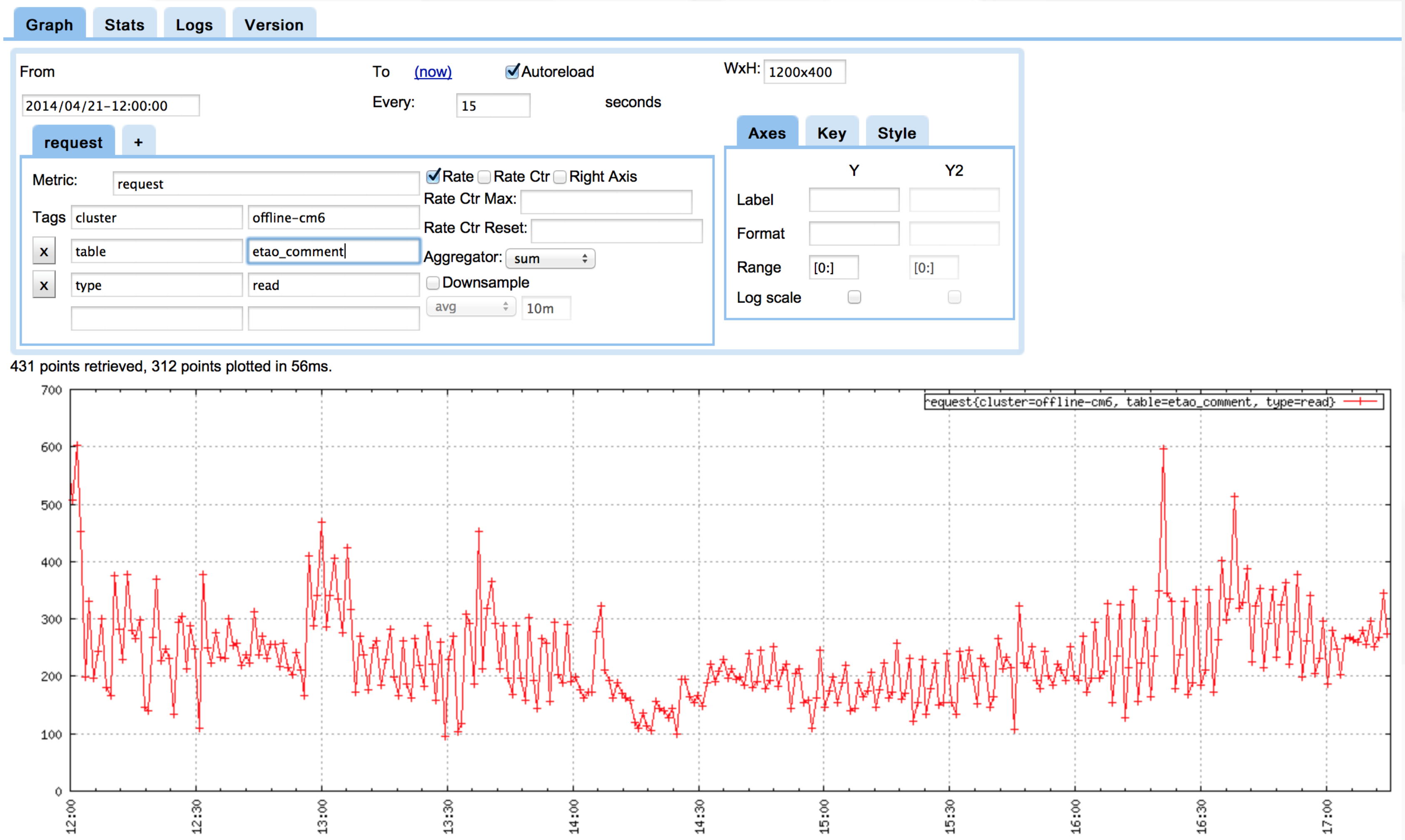
Extensional Projects

Overview

OpenTSDB **- an open-source, distributed time series database**
Phoenix **- a SQL skin over HBase**
HQueue **- a distributed and persistent message-oriented middleware**
HTunnel **- a WAL tracker and deliverer for HBase**



OpenTSDB



Graph | Stats | Logs | Version

From: 2014/04/21-12:00:00

To (now) ☑ Autoreload

Every: 15 seconds

WxH: 1200x400

request | +

Metric: request

Tags: cluster offline-cm6

x table etao_comment

x type read

☑ Rate ☐ Rate Ctr ☐ Right Axis

Rate Ctr Max:

Rate Ctr Reset:

Aggregator: sum

☐ Downsample

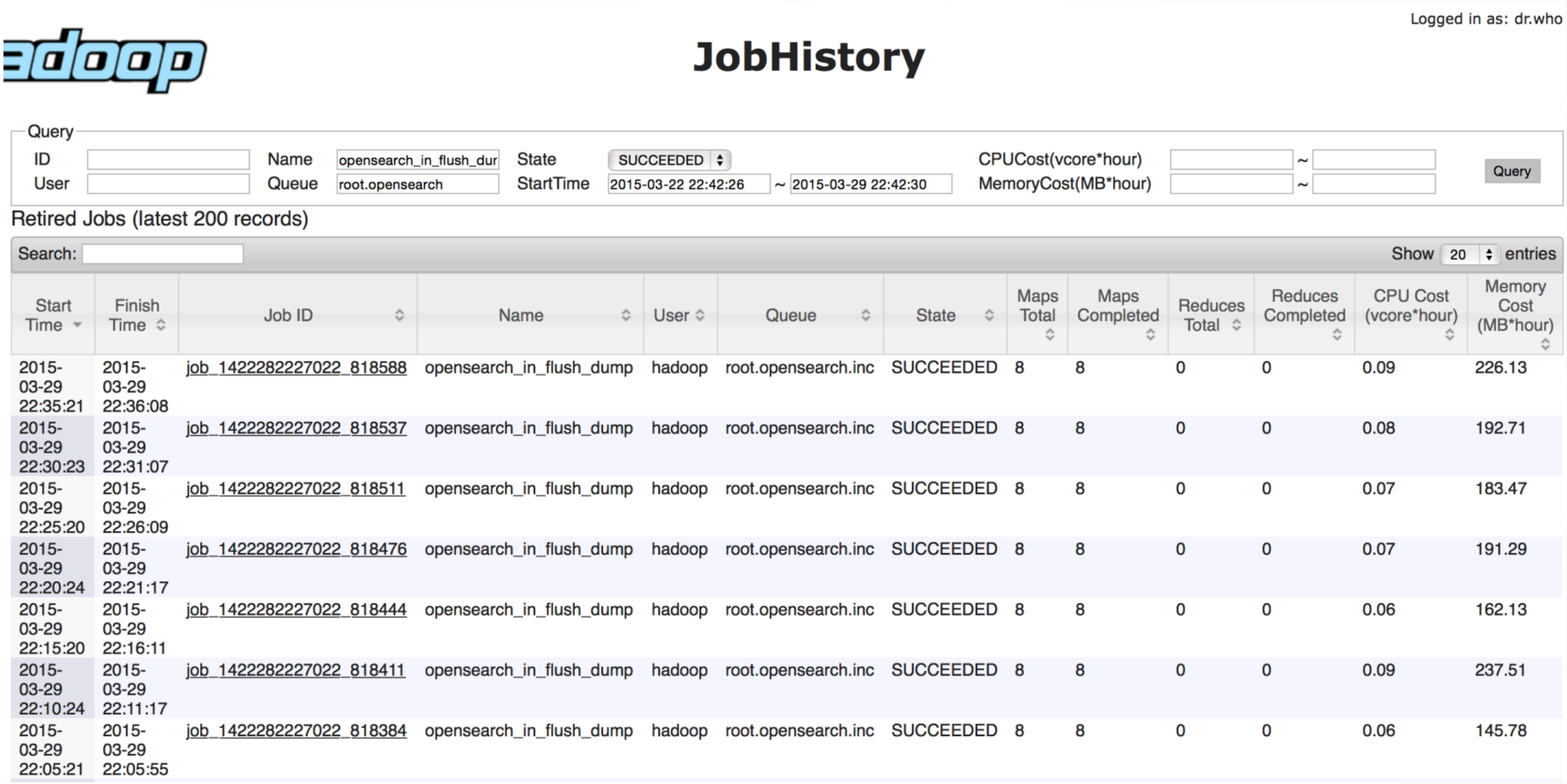
avg 10m

Axes | Key | Style

	Y	Y2
Label		
Format		
Range	[0:]	[0:]
Log scale	☐	☐

431 points retrieved, 312 points plotted in 56ms.

Phoenix



Logged in as: dr.who

JobHistory

Query

ID		Name	opensearch_in_flush_dur	State	SUCCEEDED	CPUCost(vcore*hour)	~
User		Queue	root.opensearch	StartTime	2015-03-22 22:42:26 ~ 2015-03-29 22:42:30	MemoryCost(MB*hour)	~

Query

Retired Jobs (latest 200 records)

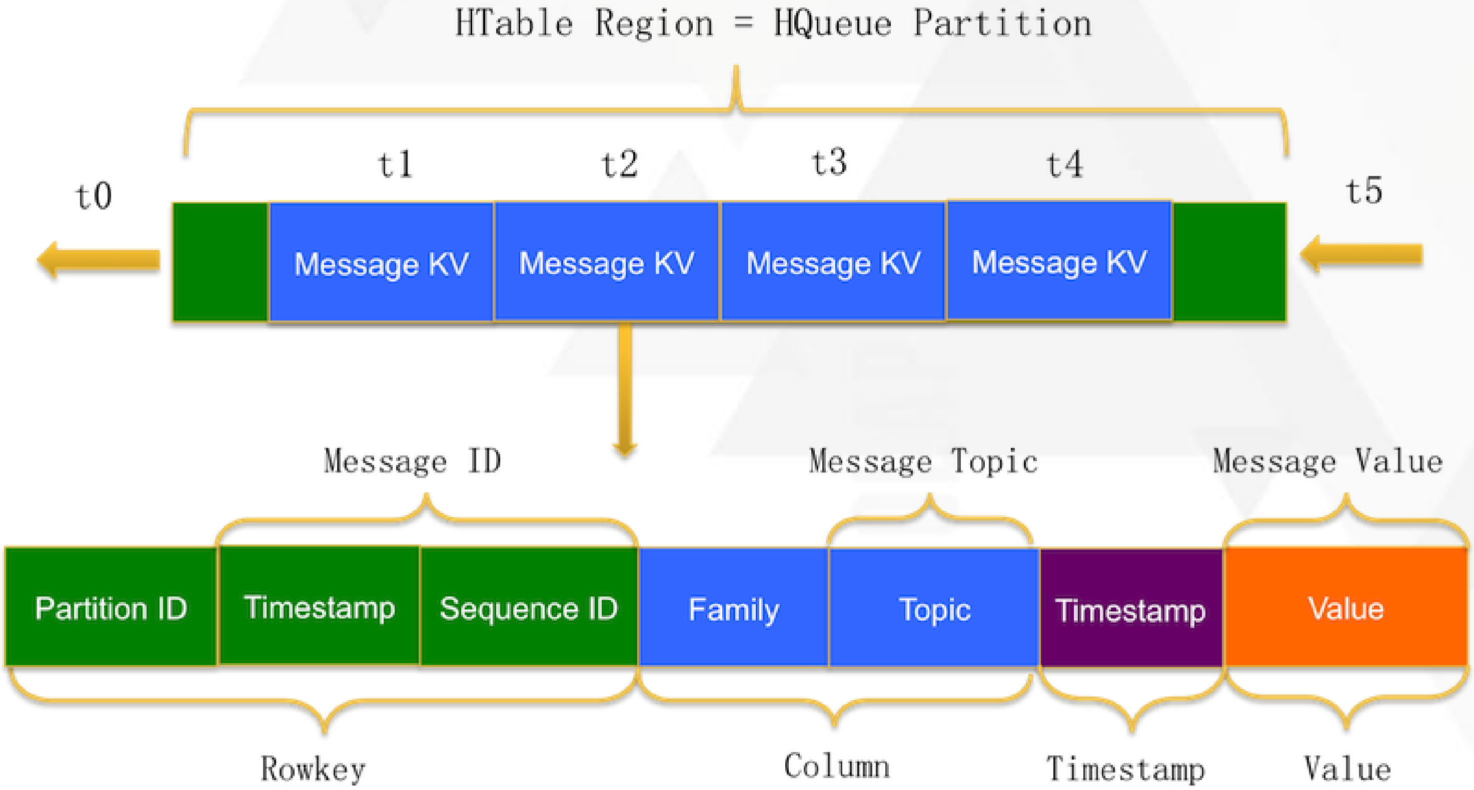
Search:

Show 20 entries

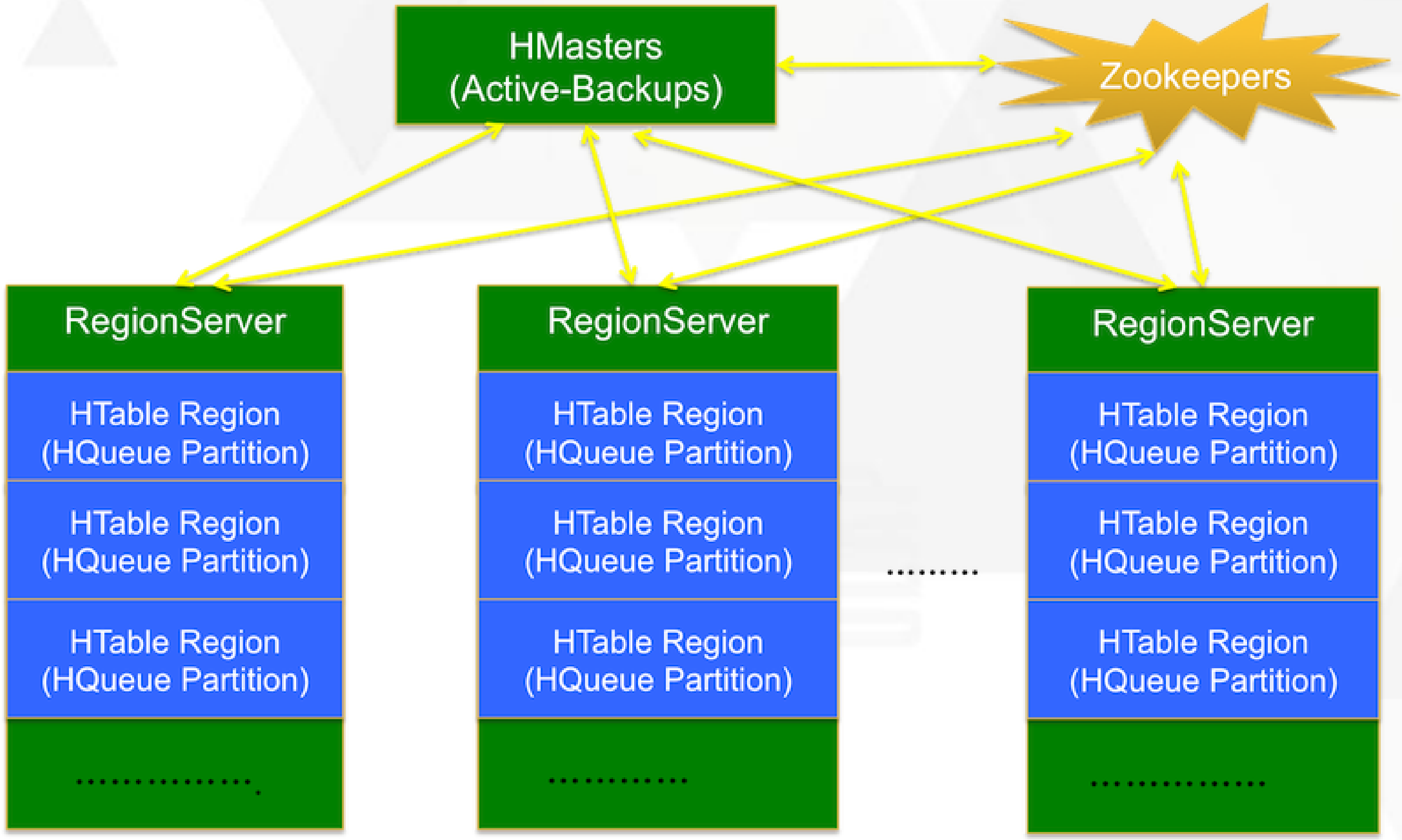
Start Time	Finish Time	Job ID	Name	User	Queue	State	Maps Total	Maps Completed	Reduces Total	Reduces Completed	CPU Cost (vcore*hour)	Memory Cost (MB*hour)
2015-03-29 22:35:21	2015-03-29 22:36:08	job_1422282227022_818588	opensearch_in_flush_dump	hadoop	root.opensearch.inc	SUCCEEDED	8	8	0	0	0.09	226.13
2015-03-29 22:30:23	2015-03-29 22:31:07	job_1422282227022_818537	opensearch_in_flush_dump	hadoop	root.opensearch.inc	SUCCEEDED	8	8	0	0	0.08	192.71
2015-03-29 22:25:20	2015-03-29 22:26:09	job_1422282227022_818511	opensearch_in_flush_dump	hadoop	root.opensearch.inc	SUCCEEDED	8	8	0	0	0.07	183.47
2015-03-29 22:20:24	2015-03-29 22:21:17	job_1422282227022_818476	opensearch_in_flush_dump	hadoop	root.opensearch.inc	SUCCEEDED	8	8	0	0	0.07	191.29
2015-03-29 22:15:20	2015-03-29 22:16:11	job_1422282227022_818444	opensearch_in_flush_dump	hadoop	root.opensearch.inc	SUCCEEDED	8	8	0	0	0.06	162.13
2015-03-29 22:10:24	2015-03-29 22:11:17	job_1422282227022_818411	opensearch_in_flush_dump	hadoop	root.opensearch.inc	SUCCEEDED	8	8	0	0	0.09	237.51
2015-03-29 22:05:21	2015-03-29 22:05:55	job_1422282227022_818384	opensearch_in_flush_dump	hadoop	root.opensearch.inc	SUCCEEDED	8	8	0	0	0.06	145.78

What’s HQueue?

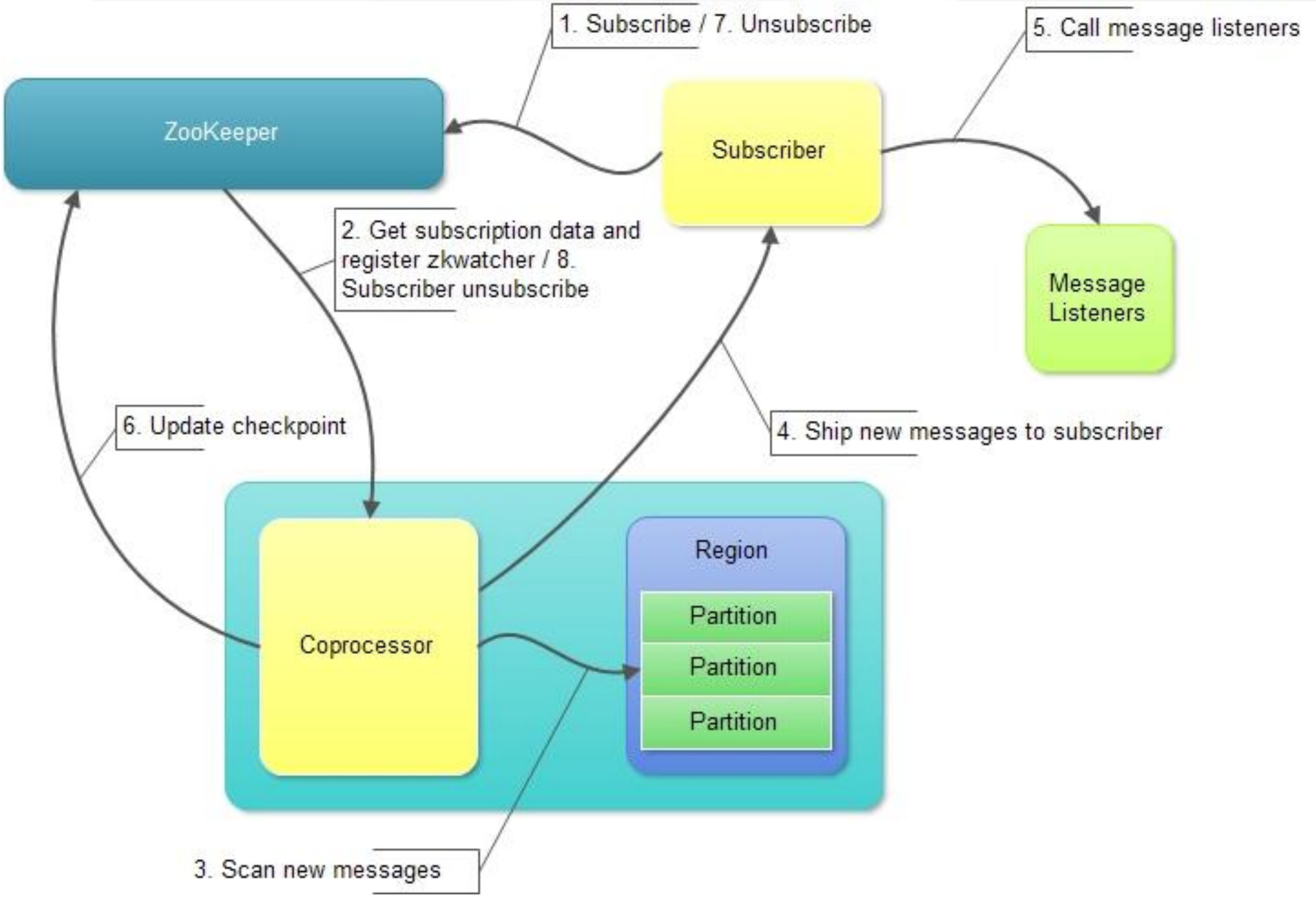
HQueue is a distributed and persistent message-oriented middleware based on HBase.



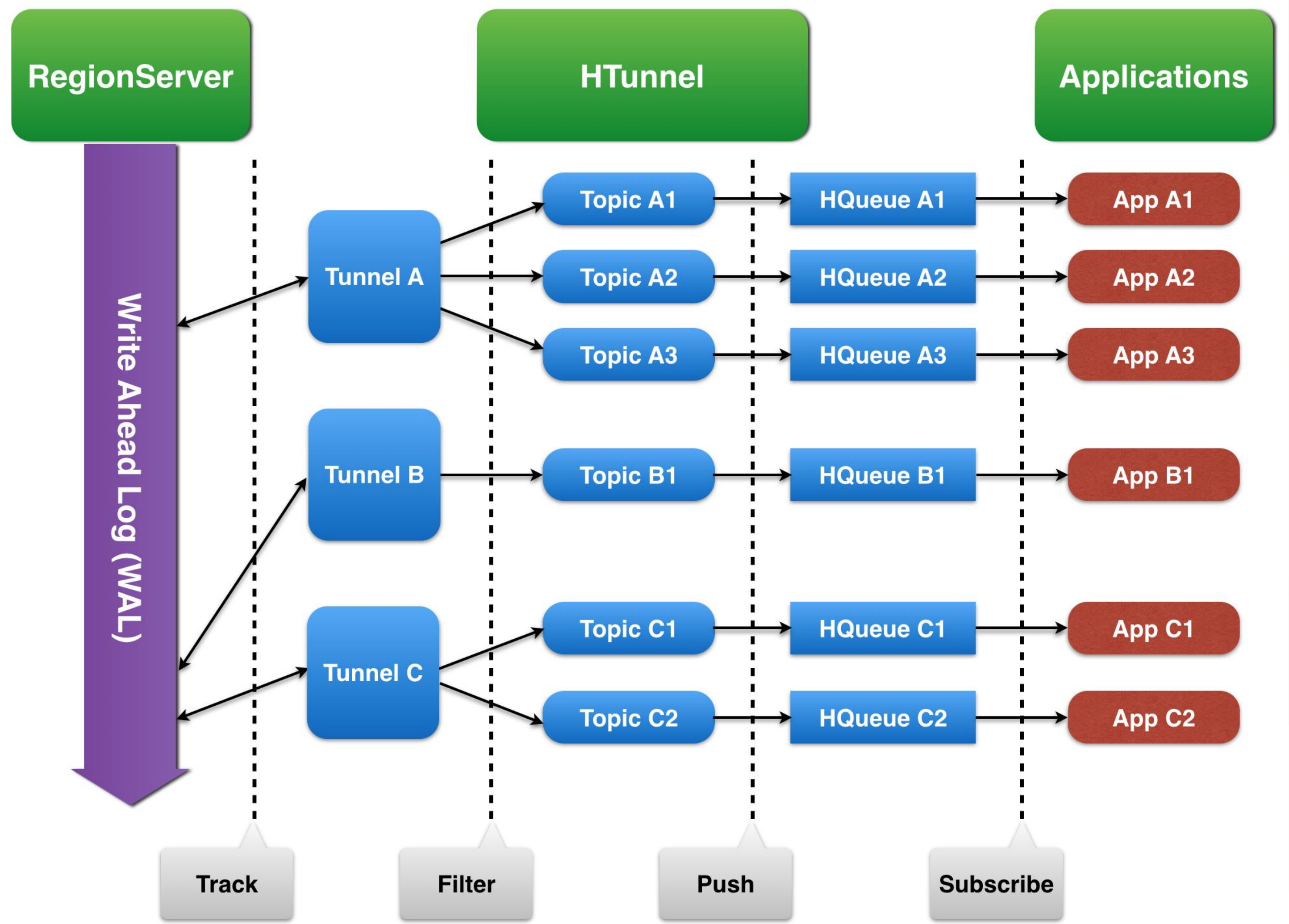
HQueue in HBase



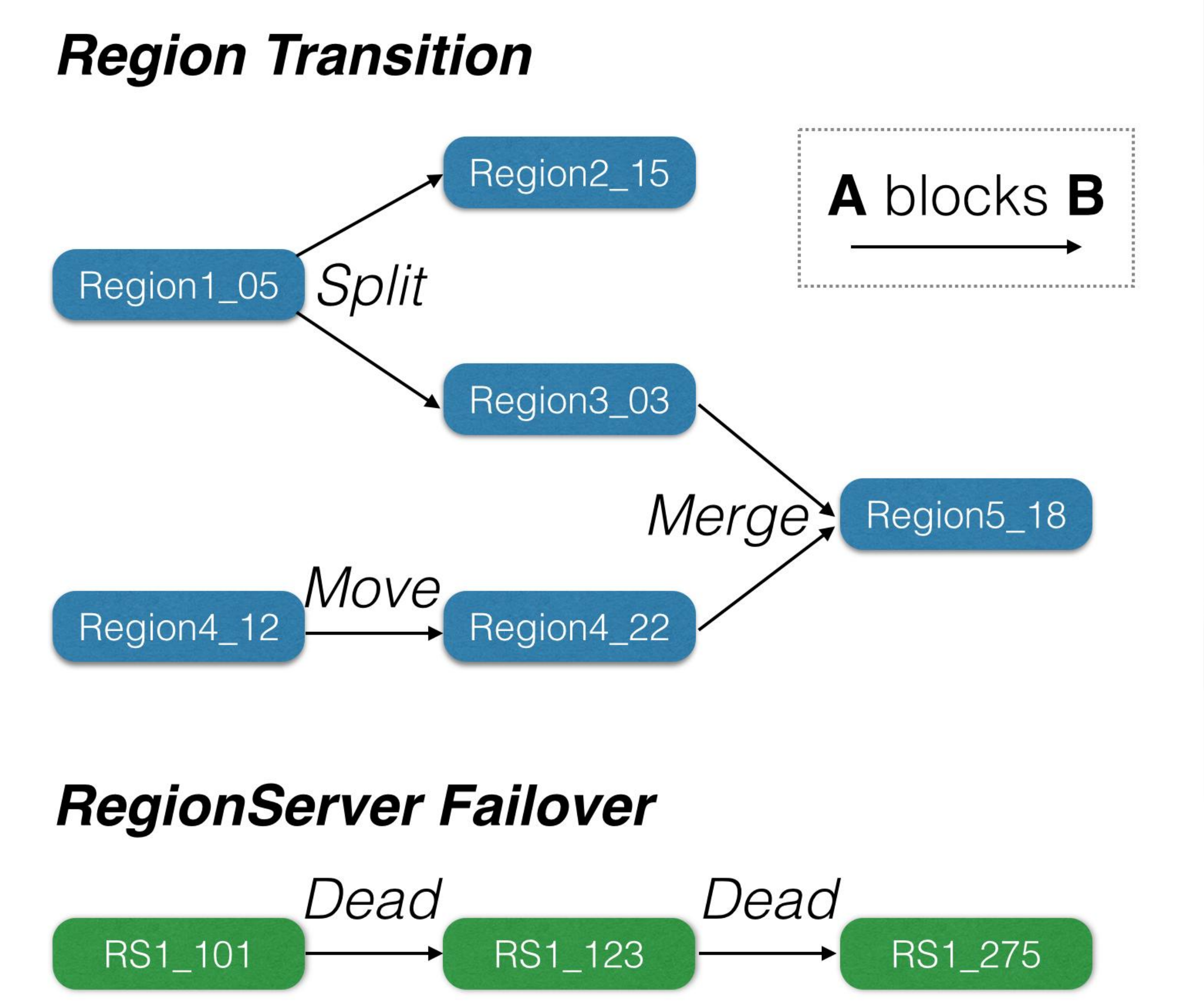
HQueue Subscription


1. Subscribe / 7. Unsubscribe
5. Call message listeners
ZooKeeper
Subscriber
2. Get subscription data and register zkwatcher / 8. Subscriber unsubscribe
Message Listeners
6. Update checkpoint
4. Ship new messages to subscriber
Region
Partition
Partition
Partition
Coprocessor
3. Scan new messages

Why HTunnel?


RegionServer
HTunnel
Applications
Write Ahead Log (WAL)
Tunnel A
Tunnel B
Tunnel C
Topic A1
Topic A2
Topic A3
Topic B1
Topic C1
Topic C2
HQueue A1
HQueue A2
HQueue A3
HQueue B1
HQueue C1
HQueue C2
App A1
App A2
App A3
App B1
App C1
App C2
Track
Filter
Push
Subscribe

HTunnel DAG


Region Transition
A blocks B
Region1_05
Split
Region2_15
Region3_03
Merge
Region5_18
Region4_12
Move
Region4_22
RegionServer Failover
RS1_101
Dead
RS1_123
Dead
RS1_275

Future

- HBase-1.X (Multi-WAL HBASE-5699)
- HBase-2.X (HBase Read HA HBASE-10070)
- Tiered Storage Support in HDFS (HDFS-2832)
- Phoenix with Merged Index (PHOENIX-1801)

Q
&
A

IT168.com ChinaUnix ITPUB

THANKS