New HBase Features

Ted Yu

yuzhihong@gmail.com

About myself

- **Graduated from Tsinghua University in 1992**
- **Hbase PMC member since June 2011**
- **Senior Member of Technical Staff @ Hortonworks**
- **Hbase 0.92.2 Release Manager**

Snapshot Types

- **Offline snapshot: table disabled**
- **Globally consistent snapshot**

① **consistent across all servers**

② **two-phase commit was planned**

③ **Barrier based Procedure implemented**

④ **Unavailability SLA maintained per region**

- **Timestamp consistent snapshot: point-in-time according to each server**

DISTRIBUTED BARRIER PROCEDURE

- **HBASE-7212, simplified version of HBASE-6573**

- **Globally consistent snapshot requires the ability to quiesce a set of regionservers before allowing progress.**

- **A failure in one regionserver results in cancelation on all others**

- **Need to be able to force failure after a specified timeout elapses**

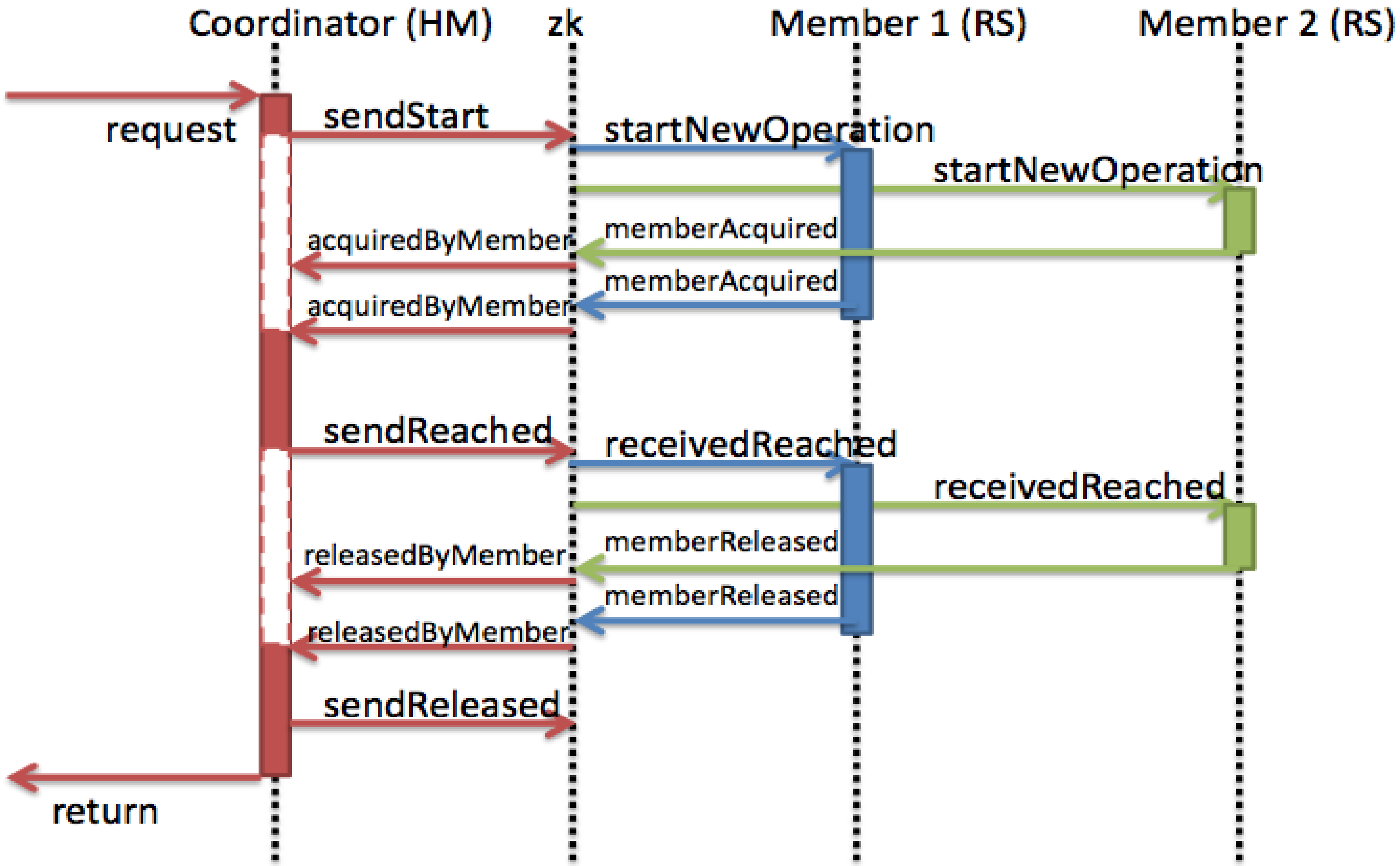
- **Solution: Users need only implement methods**

① **to acquireBarrier,**

② **to act when insideBarrier,**

③ **and to releaseBarrier**

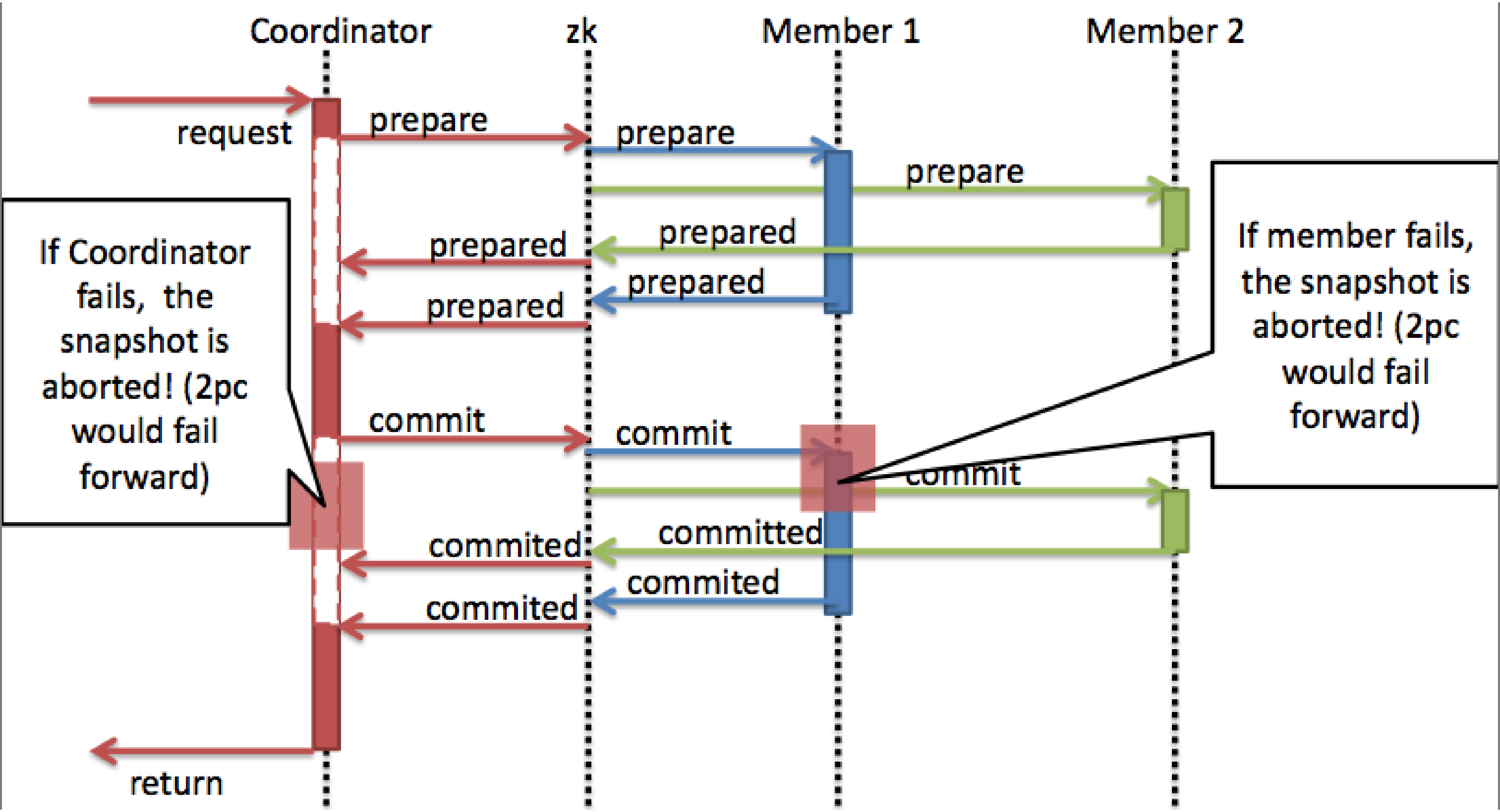
Barrier Procedure coordination



Procedure vs 2 Phase Commit

- 2PC is a distributed transaction protocol that supports ACID semantics
- After Commit is decided at the Coordinator the commit must recover and fail forward.
- Procedure has similar communications phases but does not support ACID semantics
- Does not recover on failures
- If we fail anywhere the snapshot fails, and another can be taken without adversely affecting original table

Procedure coordination



If Coordinator fails, the snapshot is aborted! (2pc would fail forward)

If member fails, the snapshot is aborted! (2pc would fail forward)

Procedure / Subprocedure

- **ProcedureCoordinator**

*_**ProcedureCoordinatorRpcs**_

ZKProcedureCoordinatorRpcs

    ZKProcedureUtil

- **Procedure**

public class Procedure implements Callable<Void>, ForeignExceptionListener {

- **ProcedureMember**

*_**ProcedureMemberRpcs**_

ZKProcedureMemberRpcs

    ZKProcedureUtil

- **Subprocedure**

SubprocedureFactory

ZK interactions: Acquire Barrier

- **Coordinator starts by wiping out, then creating and watching these znodes**

① **.../online-snapshot/acquired**

② **.../online-snapshot/reached**

③ **.../online-snapshot/abort**

- **Coordinator drops a new id (snapshot id) in acquired**

① **.../online-snapshot/acquired/snapshot121127**

- **Members see this and do their local acquire and complete by inserting an acquired node**

① **.../online-snapshot/acquired/snapshot121127/server1**

② **.../online-snapshot/acquired/snapshot121127/server2**

ZK interactions: Reached barrier

- **If all members successfully drop nodes in, the coordinator notifies that the global barrier has been reached by dropping a new znode**

① **.../online-snapshot/reached/snapshot121127**

- **Members see this and then start their reached in-barrier operation. When complete they insert znodes**

① **.../online-snapshot/reached/snapshot121127/server1**

② **.../online-snapshot/reached/snapshot121127/server2**

- **When Coordinator sees all are completed, it deletes all these znodes**

ZK interactions: Aborting

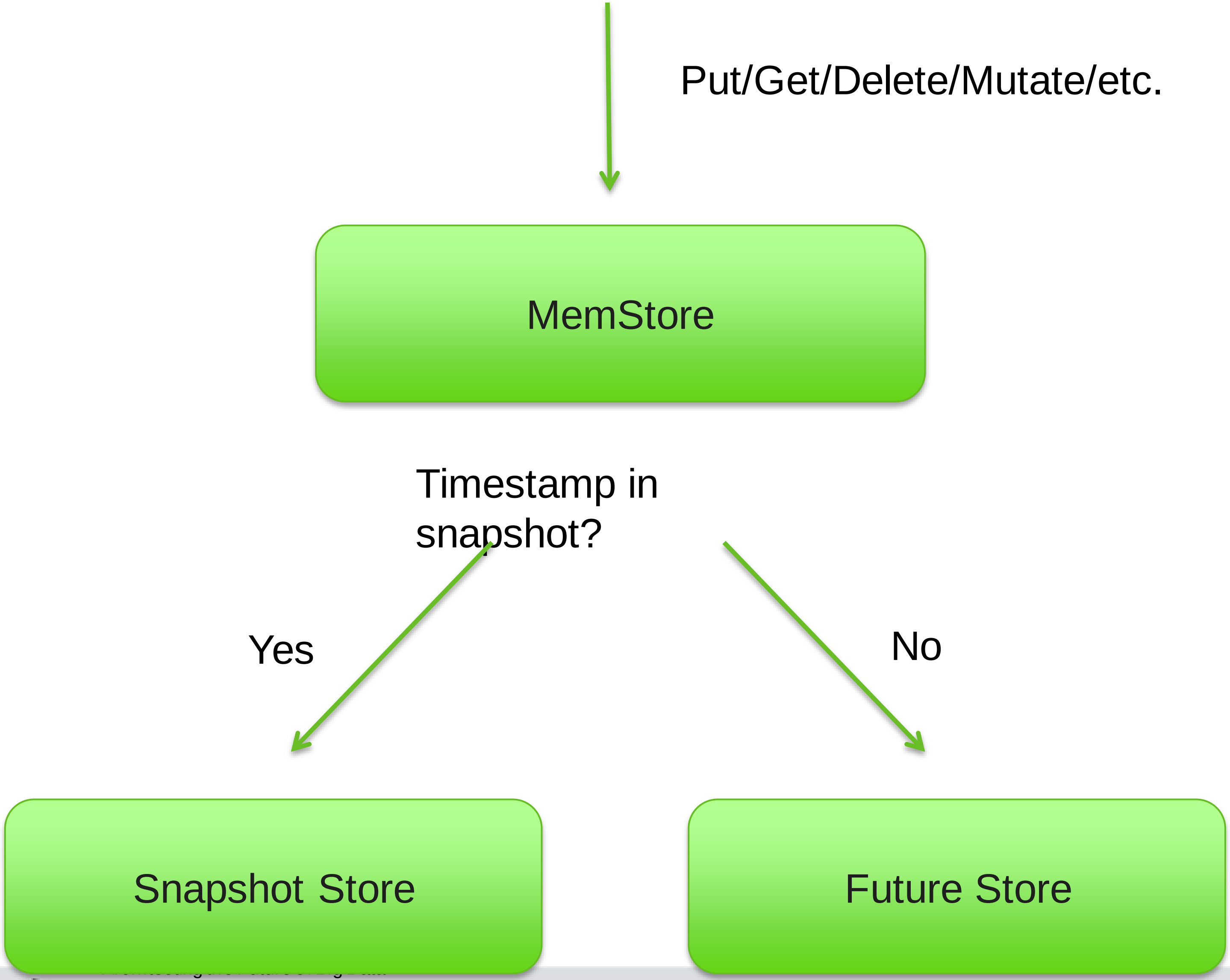
- **If anybody encounters an error and is unable to complete due to timeout, it will drop a node in the aborted znode dir. (Note that the source member is not part of the name!)**

① **.../online-snapshot/abort/snapshot121127**

- **It contains a protobuf serialized ExternalException. This contains the source name. It is deserialized by all others and everyone gets receiveError calls with the exception. Everyone bails out.**

- **Eventually the coordinator will delete all znodes related to this Procedure**

Timestamp Consistent Snapshots



Put/Get/Delete/Mutate/etc.

MemStore

Timestamp in snapshot?

Yes

No

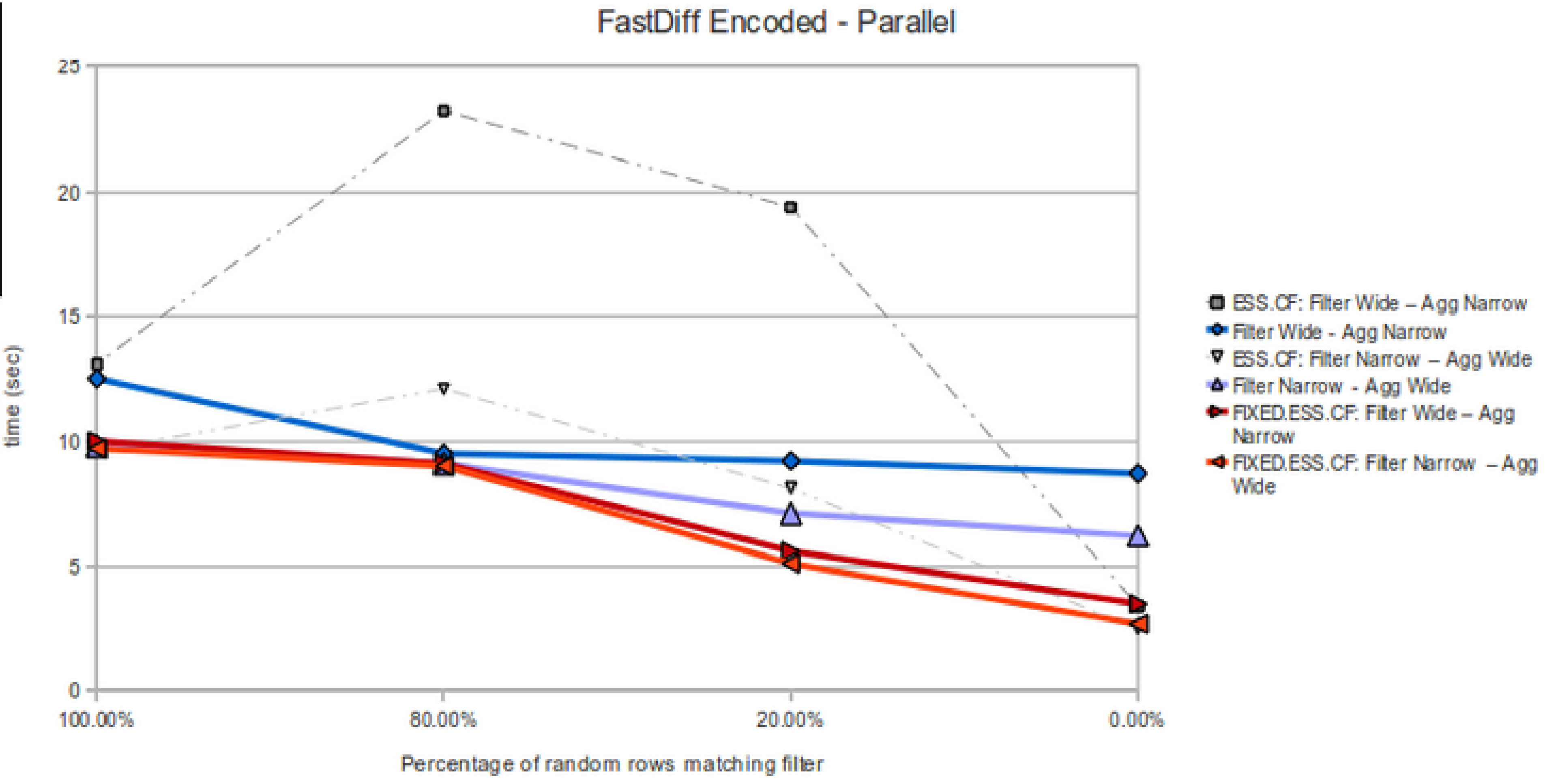
Snapshot Store

Future Store

Snapshot Recovery Options

- **Export snapshot**
  - Send snapshot to another cluster
  - All required HFiles/Hlogs
- **Clone snapshot**
  - Create new table from snapshot
- **Restore table**
  - Rollback table to specific state
  - Handles region creation / deletion
  - Fixes META for you
- **MapReduce over snapshot files (HBASE-8369)**

Essential Column Family – HBASE-5416



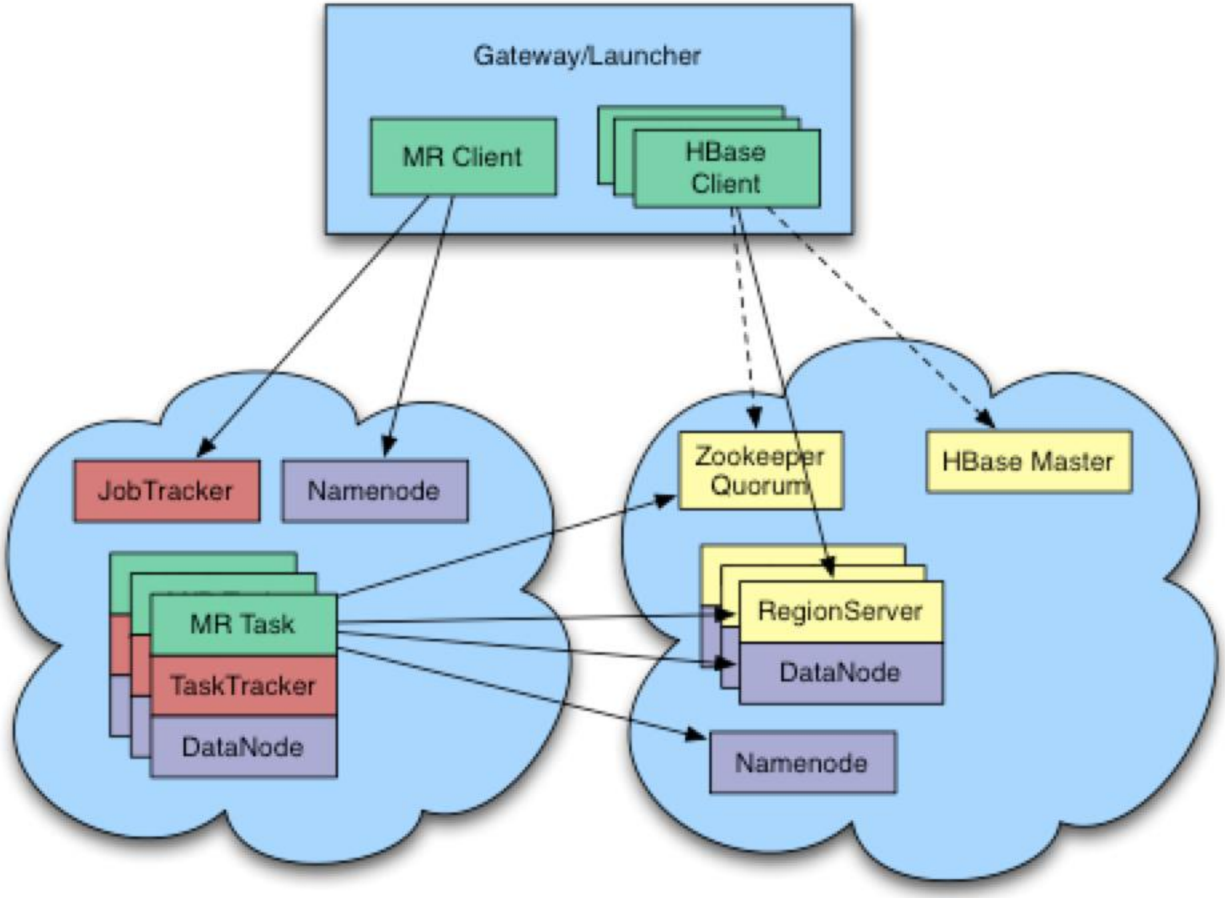
Hosted Multi-tenant Service

- Isolated Deployment
- Security

1. Authentication
2. Authorization

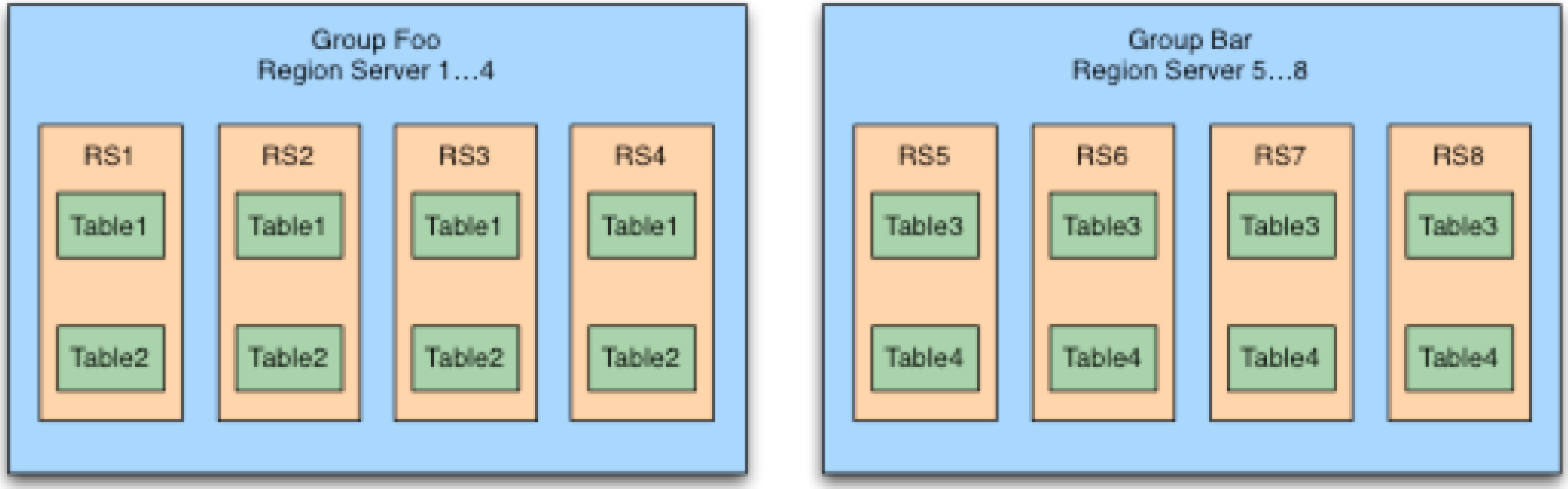
- Region Server Group (HBASE-6721)
- Namespace (HBASE-8015)

Isolated Deployment

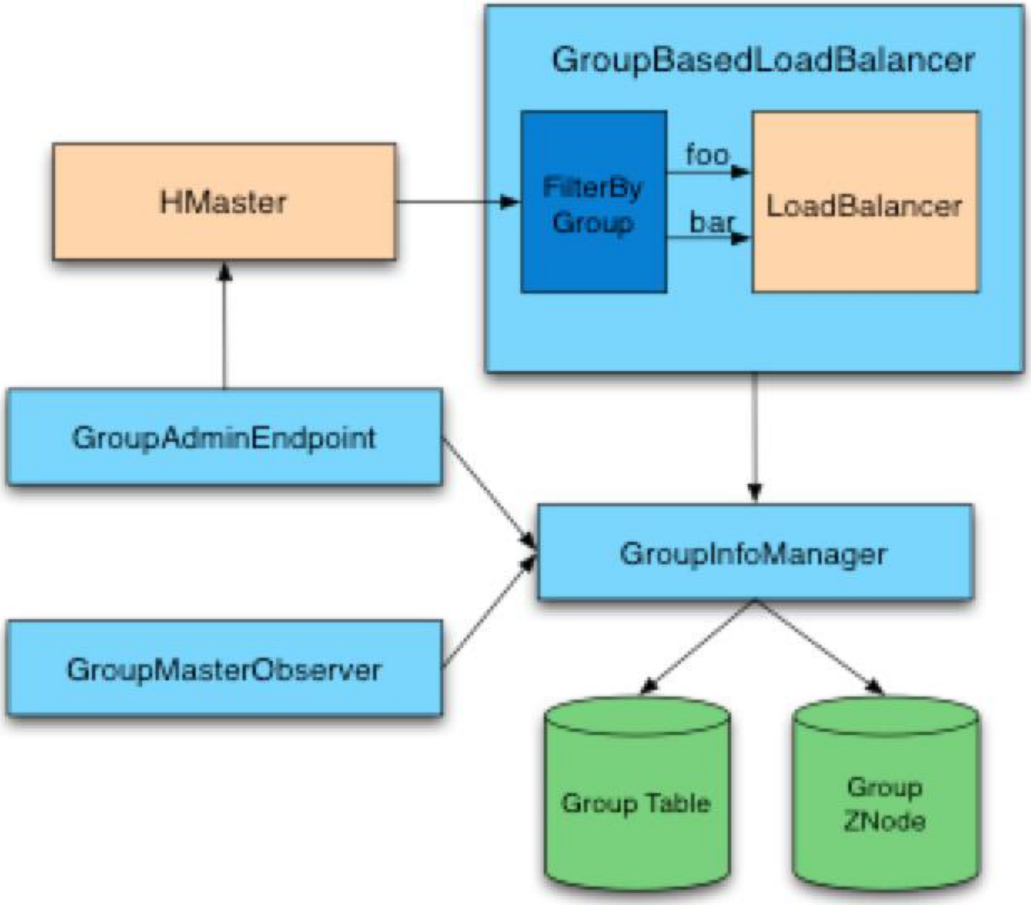


Region Server Groups

- **Region Servers are partitioned into groups**
- **Tables are partitioned respecting group boundaries**
- **Resource Isolation**
- **Flexibility with configuration (**group_add, group_move_tables**)**



Region Server Groups, Cont'd

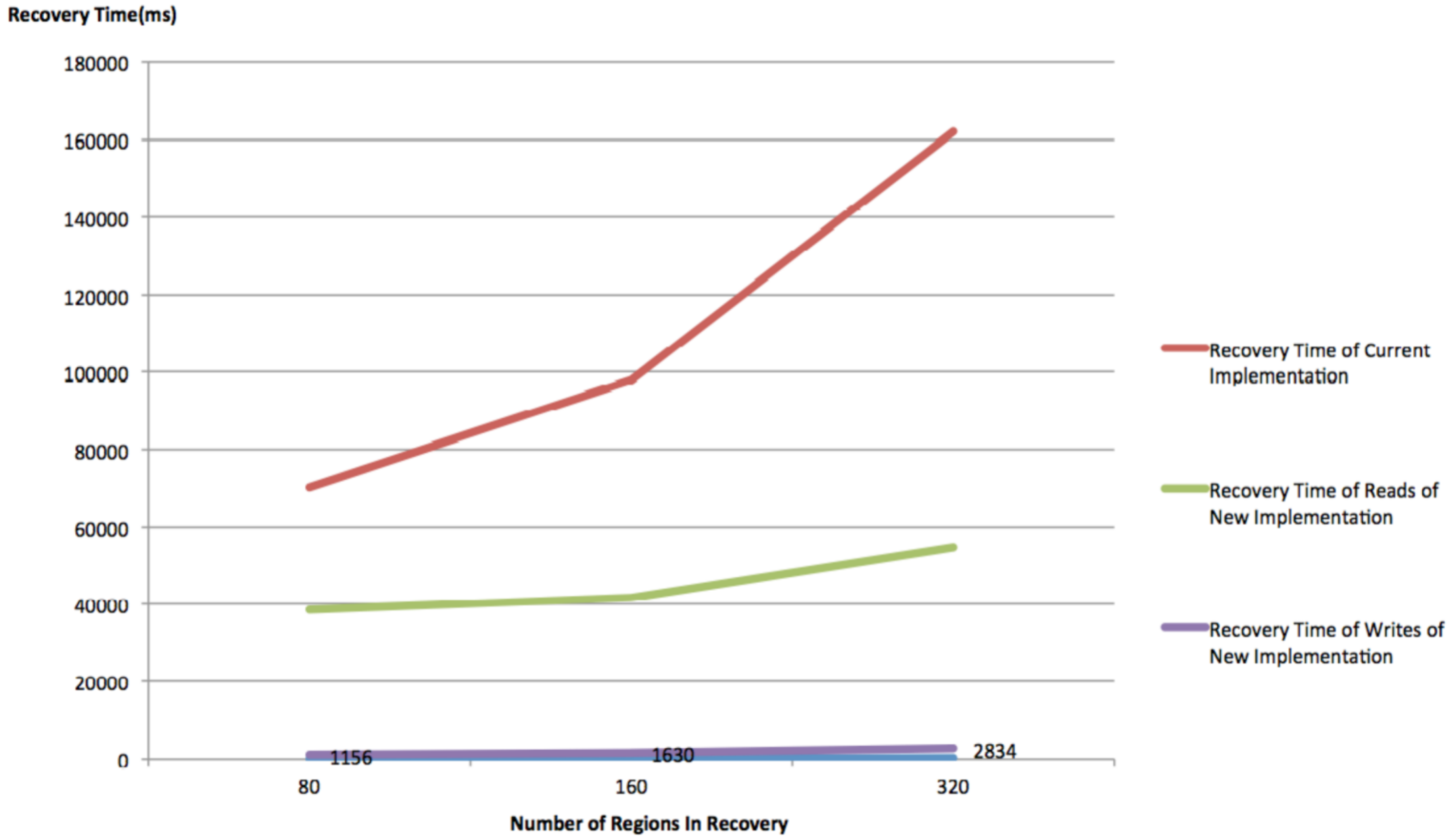


Namespace

- **Table Name: <table namespace>.<table qualifier>**
  - i.e. my_ns.my_table
- **Reserved namespaces**
  - Default – tables with no explicit namespace
  - System – tables are guaranteed to be assigned prior to user tables
- **Namespace Admin can create/drop member tables**
- **Table Path: /<hbaseRoot>/data/<namespace>/<tableName>**
  - /hbase/data/my_ns/my_ns.my_table

Distributed Log Replay (HBASE-7006)

- **Distributed log splitting suffers from creation of many small files**
- **Distributed log replay scales linearly with number of WAL files / regions**



Q & A