

HDFS Router-based Federation

Íñigo Goiri
Microsoft

Chao Sun
Uber

High-level HDFS setup

Microsoft

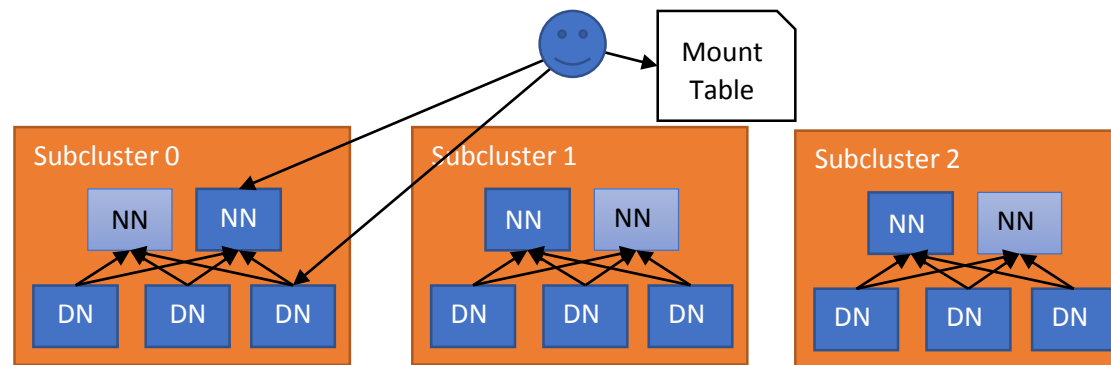
- 3 data centers
 - 2 more provisioning
- Harvested capacity [OSDI'16]
- Hadoop 2.7.1 (migrating to 2.9)
 - Many internal extensions
- Internal batch workloads

Uber

- 3 data centers
 - 4th is coming soon
- Thousands of servers
- 100+ PB of data
- Hadoop 2.8.2
 - Many custom and backported patches
- Serves applications
 - Fraud detection, ML/DL, ETA calculation...
- **Foundation for Uber's data lake**

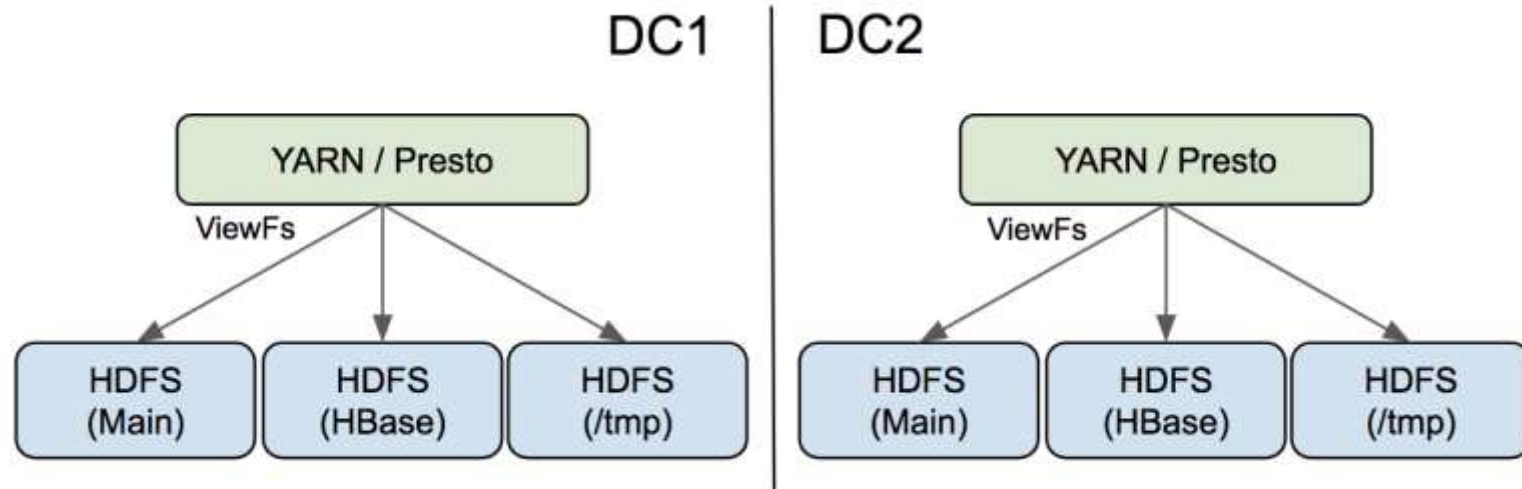
Current architecture in HDFS

- NameNode scalability/performance issues
- Split into independent namespaces (subclusters)
 - Fragmentation
 - Users in charge of choosing subcluster
- ViewFS to unify subclusters



Setup at Uber

- Use client-side ViewFS config
- Split into 3 clusters (each DC)
 - Main production HDFS cluster
 - Hbase cluster
 - Tmp cluster (Hive scratch directory, Yarn application logs, etc).

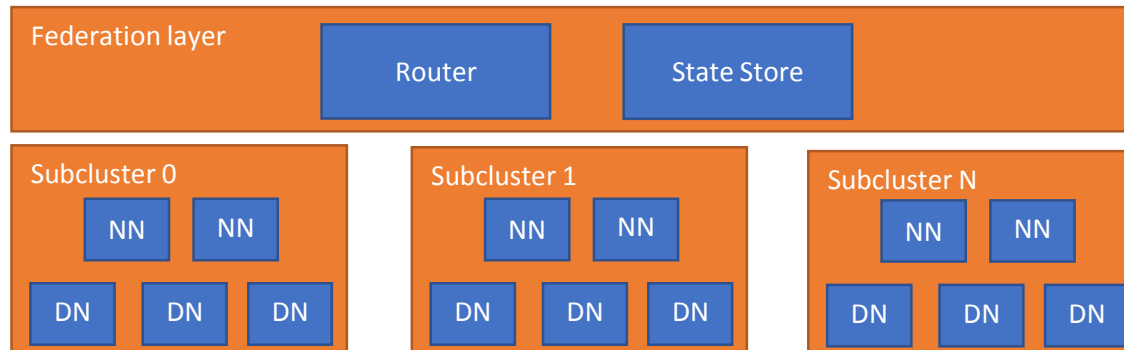


Problems with ViewFS

- Hard to maintain per-client mount table
 - Many HDFS clients, libraries, configurations,...
 - Each client may have its own mount table
 - Requires infrastructure to distribute mount table updates
 - Adding more datacenters
- Manual balancing of subclusters
- Solution
 - “Centralize” the mount table
 - Introduce a routing layer: Router-Based Federation (RBF)

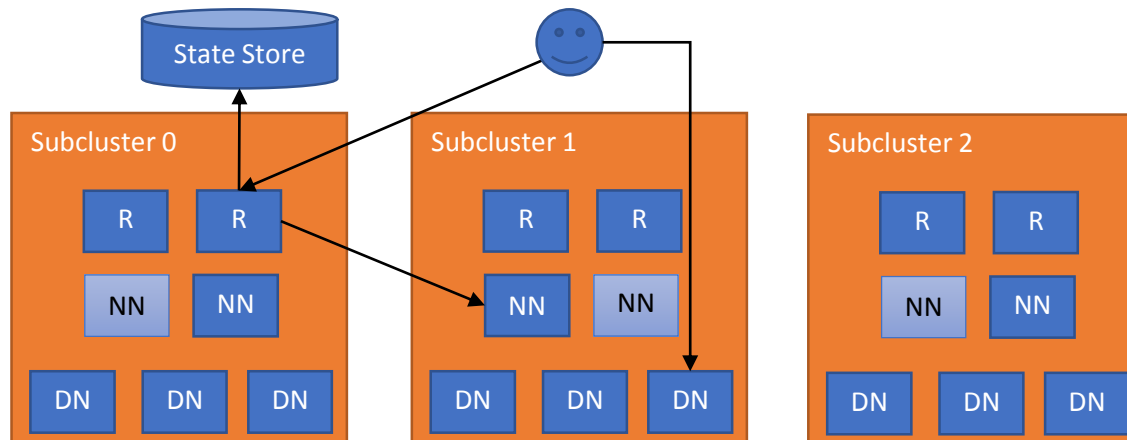
Our solution: RBF

- “Centralize” the mount table
- Introduce a routing layer: Router-Based Federation (RBF)
 - Router: Proxy requests from users to the right NameNode
 - State Store: Shared mount table and other state



Router

- Unified view of the federation
- Performance improvements
 - Caching State Store data
 - Includes modified client to contact NameNodes



State Store

- Shared information
 - Mount table: /DC0-2 → DC0-2, /
 - Federation membership: DC0, DC0-1, DC0-2, DC0-3
 - Router tracking: R1, R2, R11, R22,...
- Not critical for performance (cached in each Router)
- Implementations
 - ZooKeeper, HDFS, SQL server,...

HDFS RBF deployments

Microsoft

- 1 Router per Namenode
- All data centers
- ZooKeeper as the State Store
- Main workloads
 - Large deep learning
 - Application logs

Uber

- 2 routers
- 1 data center
- ZooKeeper as the State Store
- Workload
 - Hive traffic accessing scratch dir

Microsoft RBF deployment

- 23k servers
 - 8 subclusters
 - 28 NameNodes (4 per subcluster)
 - 28 Routers: load balancing
- Transparent access
 - RPC
 - WebHDFS
 - Web UI

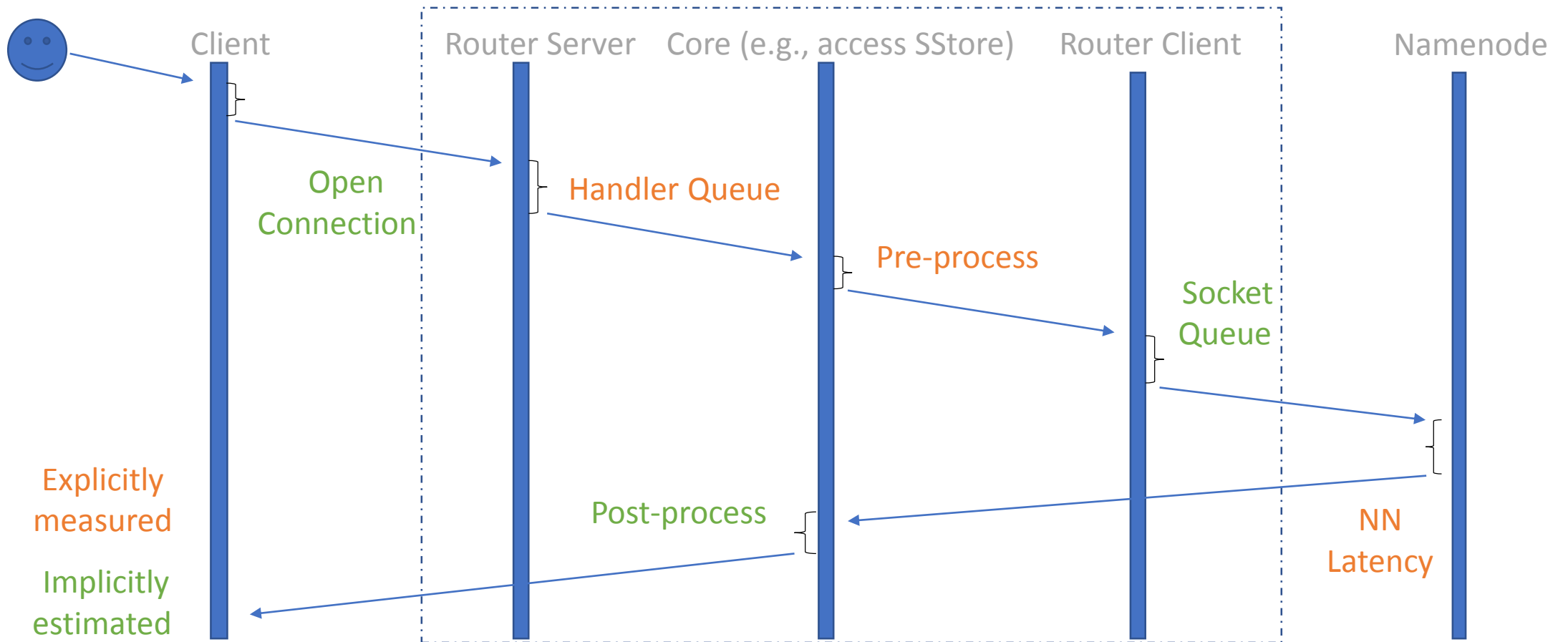
Mount Table

Global path	Target nameservice	Target path	Order	Read only	Owner	Group	Permission	Quota/Usage	Date Modified	Date Created
/	O4	/		✓	hadoop	supergroup	new-r-x	[NoQuota -] [NoQuota -]	2017/08/19 09:29:31	2017/08/19 09:29:31
/hadoop-logs	O4	/hadoop-logs		✓	hadoop	hdfs	new-r-x	[NoQuota -] [NoQuota -]	2018/04/16 09:48:50	2018/04/16 09:48:50
/usernames	O4-names	/		✓	hadoop	supergroup	new-r-x	[NoQuota -] [NoQuota -]	2017/12/13 18:23:46	2017/12/13 18:23:46
/tmp	O4-	/tmp	HASH_ALL	✓	hadoop	supergroup	new-r-x	[NoQuota -] [NoQuota -]	2017/11/27 14:36:55	2017/11/27 14:36:55
/user/ratf		/user/ratf	HASH	✓	hadoop	supergroup	new-r-x	[NoQuota -] [NoQuota -]	2017/08/09 14:38:44	2017/08/09 14:38:37
/user/jesaremi		/user/jesaremi	HASH	✓	hadoop	supergroup	new-r-x	[NoQuota -] [NoQuota -]	2017/08/10 15:29:26	2017/08/10 15:16:14

Namenode Information

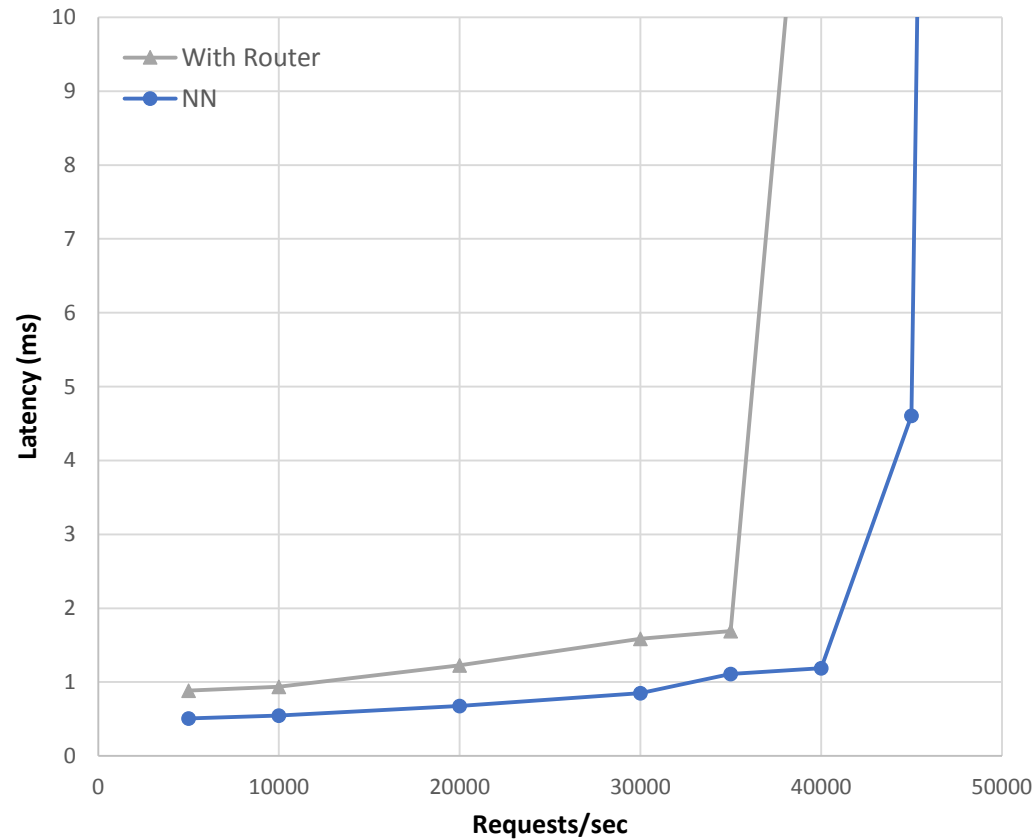
NameNode Information												Active	Standby	Safe mode	Unavailable
				Blocks						Nodes		Decom			
NameNode	Web address	Last Contact	Capacity	Files	Total	Missing	Under-Replicated	Live	Dead	Progress	Live	Dead			
O4-0	nn1	7	18.06 PB <div><div></div></div>	8150022	13802818	0	0	2955	2	5	2	0			
O4-0	nn2	8	18.06 PB <div><div></div></div>	8150024	13802819	0	0	2955	40	3	5	3			
O4-0	nn3	8	18.06 PB <div><div></div></div>	8150024	13802819	0	0	2955	169	0	7	21			
O4-0	nn4	8	18.06 PB <div><div></div></div>	8150022	13802818	0	0	2954	43	1	8	5			
O4-1	nn1	16	23.16 PB <div><div></div></div>	9929358	16126267	0	0	3859	159	0	4	25			
O4-1	nn2	8	23.17 PB <div><div></div></div>	9929834	16126748	0	0	3858	166	0	4	25			
O4-1	nn3	8	23.16 PB <div><div></div></div>	9929051	16128825	0	0	3858	347	0	4	41			
O4-1	nn4	6	23.17 PB <div><div></div></div>	9929294	16128933	0	597	3859	391	1	4	42			
O4-2	nn1	6	20.94 PB <div><div></div></div>	23910007	31351932	0	0	3561	107	1	4	9			
O4-2	nn2	16	20.94 PB <div><div></div></div>	23910007	31351932	0	0	3561	88	1	4	8			
O4-2	nn3	7	20.94 PB <div><div></div></div>	23910007	31351932	0	0	3561	41	1	4	2			
O4-2	nn4	7	20.94 PB <div><div></div></div>	23910010	31351935	0	181	3561	183	1	4	18			
O4-3	nn1	9	18.18 PB <div><div></div></div>	91479952	85912367	0	0	2993	30	10	1	1			
O4-3	nn2	8	18.18 PB <div><div></div></div>	91479952	85912367	1	1340841	2993	67	8	3	11			

Router adds latency

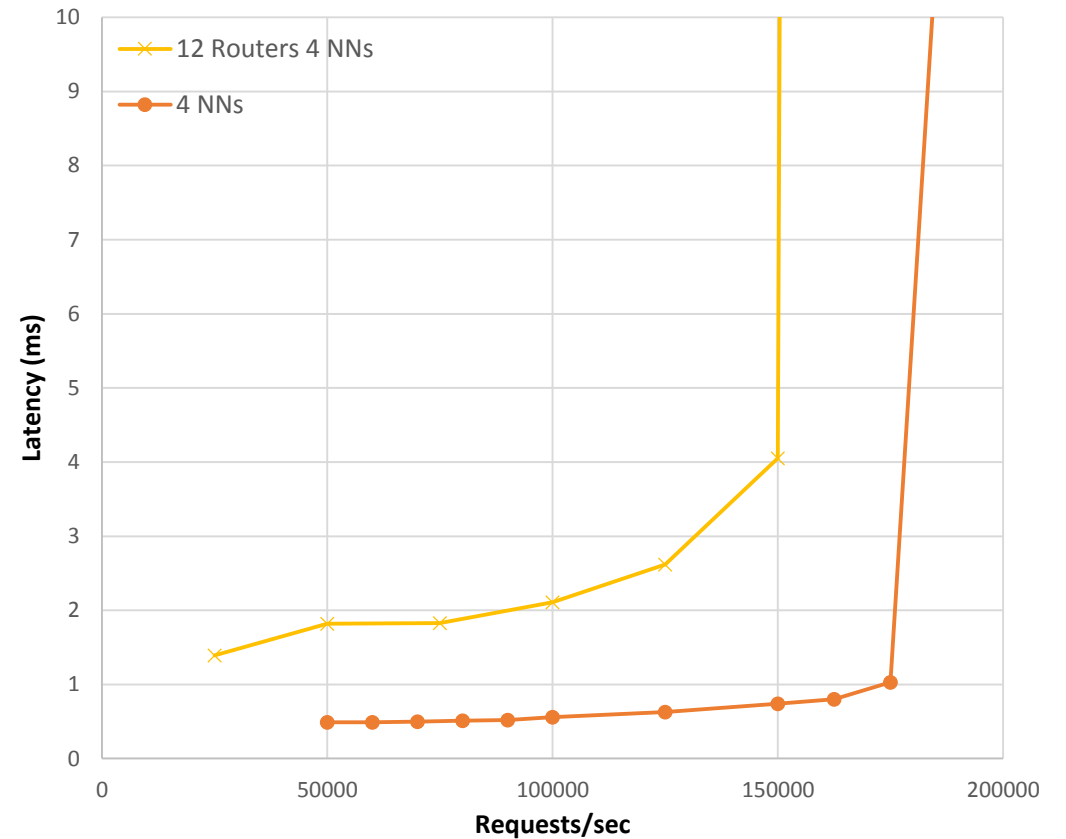


Performance and scalability

NN vs R→NN



4NN vs 12R → 4NN



Read metadata (cheapest access → worst case for Router)

Tuning Router RPC client

- Router uses a connection pool for each <user, namenode> pair
 - Connection creation/cleanup is done asynchronously
 - Hard to configure: Connection pool size, creator queue size,...
- Default IPC client is a bottleneck
 - Router forwarding many requests to NameNodes
 - [HADOOP-13144](#) allow multiple concurrent client/connections
 - [HDFS-13274](#) leverage concurrent connections

Active development

- Many contributors
 - VipShop, Uber, Huawei,...
- Bug fixes
- New features:
 - WebHDFS
 - Better subcluster isolation
 - Federated quotas
 - Tracking the state of the Routers
 - Spreading mount points across subclusters

Spreading mount points across subclusters

- Large jobs processing data spanning multiple subclusters
- Similar to merge mount points (HADOOP-8298)
- Policies to decide the subcluster to write a file
 - Consistent hashing, available space, local subcluster,...
- Limitations
 - Needs to find files in multiple subclusters
 - Renaming has some inefficiencies
 - Subcluster unavailability is hard to manage

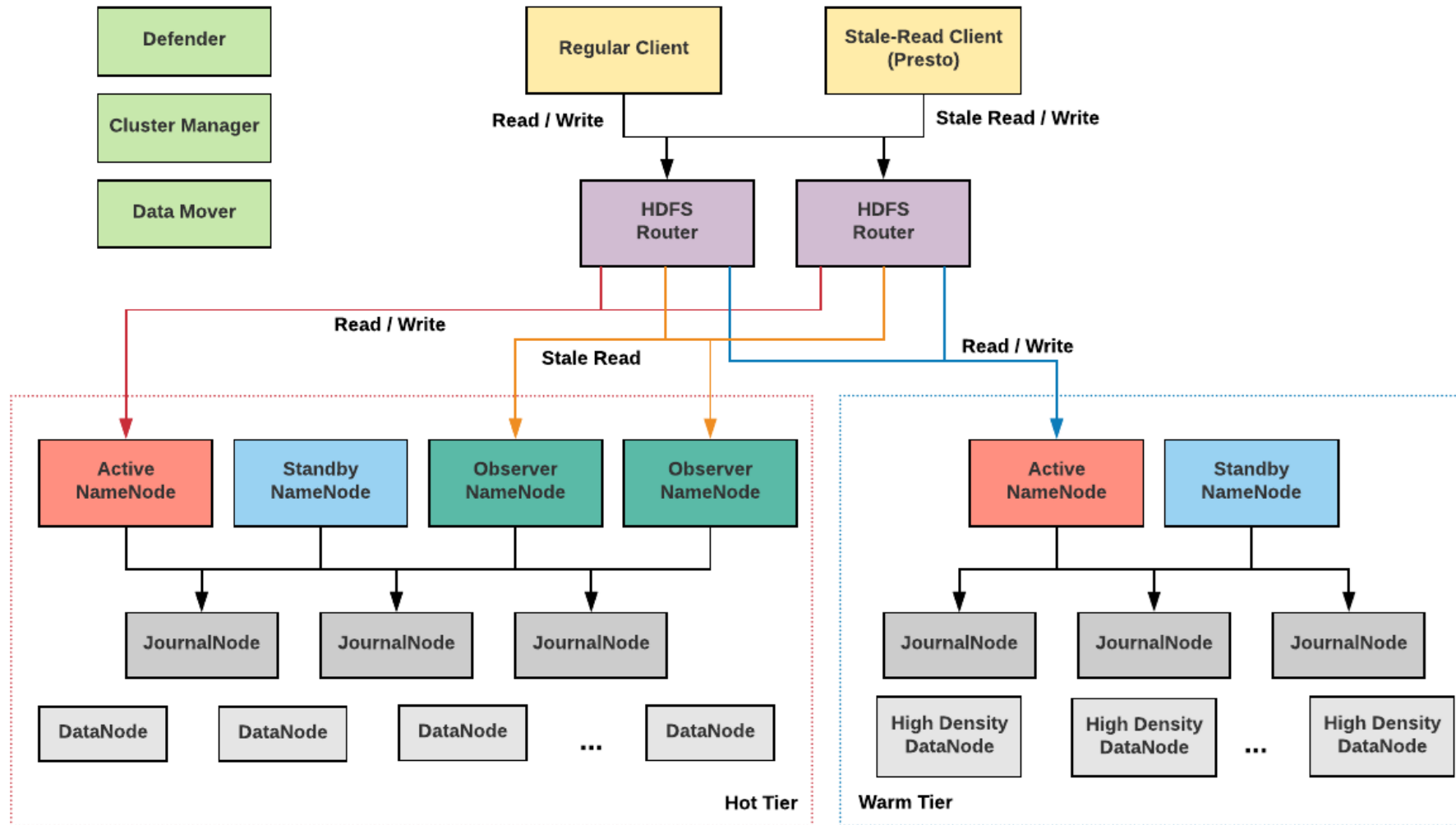
Open issues

- Router → Namenode connections ([HDFS-13274](#))
- Security (HDFS-13532)
 - Manage delegation tokens
 - LinkedIn driving this effort
- Rebalancer (HDFS-13123)
- inodes (HDFS-13469)
 - We need to identify the nameservice
- Other issues tracked in HDFS-12615

Rebalancer

- Subclusters can get unbalanced
 - Load, space,...
- Manually move files/folders between subclusters
 - Hard to provide strong consistency
- Transparent data movement between subclusters
 - Service to monitor if subclusters balanced
 - Policies to decide what to move and trigger rebalancing [ATC'17]
- Rebalancer
 - Move data: DistCp, Tiered storage, DN block linking,...
 - Change namespace (mount table) consistently

Uber future plan: overview



Uber future plan: detailed

- Routers on top of all HDFS clusters
- Service to initiate data movement between hot and warm clusters
 - Similar to Rebalancer, implemented in Golang
- Service to track dataset temperature and initiate data movement
- Observer (ReadOnly) NameNodes to serve stale read requests
 - [HDFS-12943](#)
 - > 90% traffic are read RPC requests
 - Mostly for Presto queries

New ideas/brainstorming

- Federating federations
 1. Creating a hierarchy: putting Routers in front of other Routers
 2. Large federation: sharing State Store across deployments
- Connecting DataNodes to RBF
 - DNs need to be configured to join subclusters
 - Can DNs use the Routers to join subclusters?
 1. Heartbeat through the Router
 2. Find the Datanodes through the Router