High Availability Name Node

Name Node Redundancy

David Liao| December 2011

eBay Inc. Confidential

Requirements

- Cluster will continue its services when any Name Nodes died (failed)
  - Master Name Node should failover to any Redundant Name Nodes.
  - Any Redundant Name Node failure should not affect Name Node Services at all.
- Redundant Name Nodes can join the cluster at any time.
- Minimum code change to existing hadoop code

Design

- Multiple Redundant Name Nodes run on same code:
  - Master Name Node (leader) serves the requests and generates WAL (Edit Logs).
  - Persisted Contents at all Name Nodes (FS Image + Edit Logs) are identical.
  - A few minutes of Live Logs are stored in Zookeeper.
  - Failover can be completed within few minutes.
  - Empty Redundant Name Node can joins cluster and catches up and become ready to swap in any time.
- Easy Leader election by using Zookeeper.
- No Source Code Change on client application.
  - It may require link to newer version of Hadoop Library. It is the case for current Hadoop and HBase anyway.

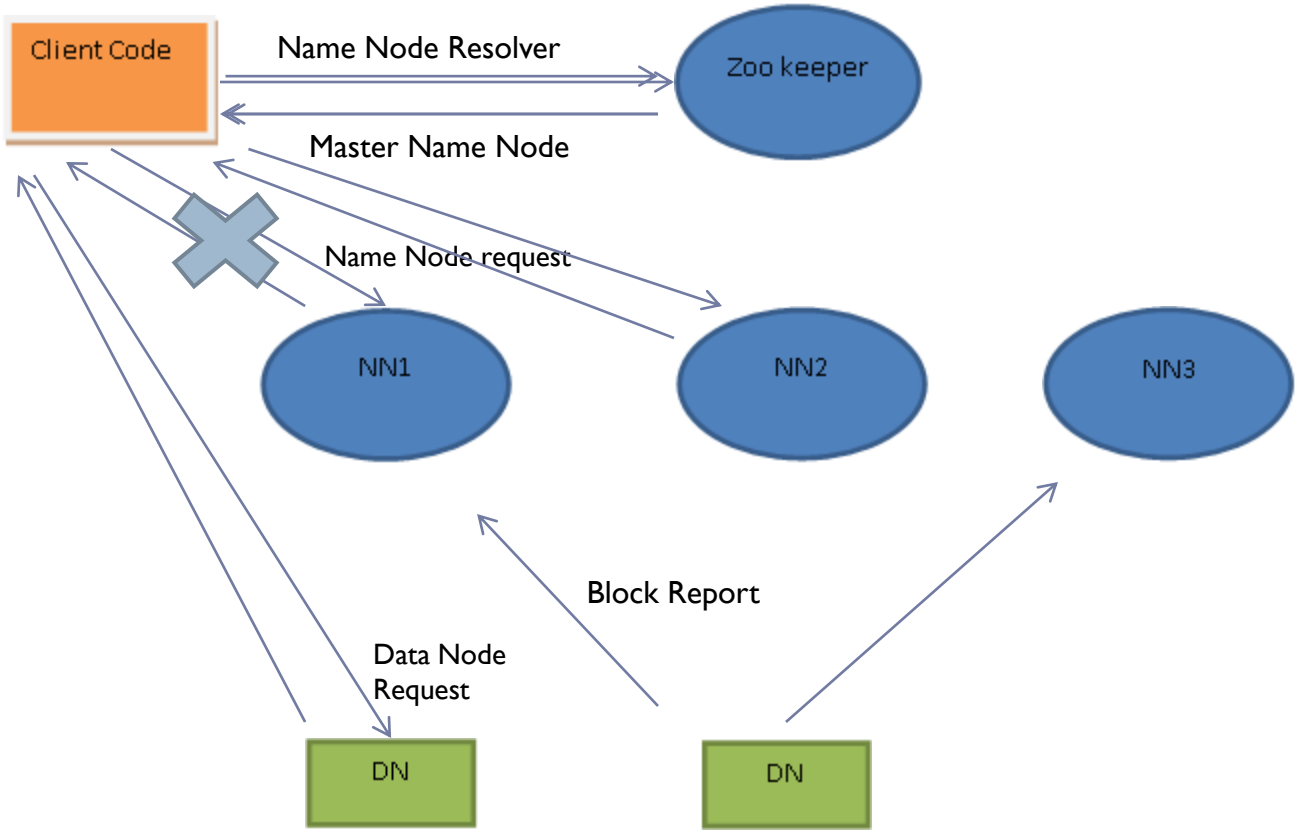
Persisted HDFS File System

- FS Image File
  - file of snapshot of HDFS File system
- Edit Logs Files
  - files of Edit Logs (WAL) since snapshot
- Live Edit Log
  - Live Edit Logs (WAL) are stored in Zookeeper and persisted to Edit Logs File later

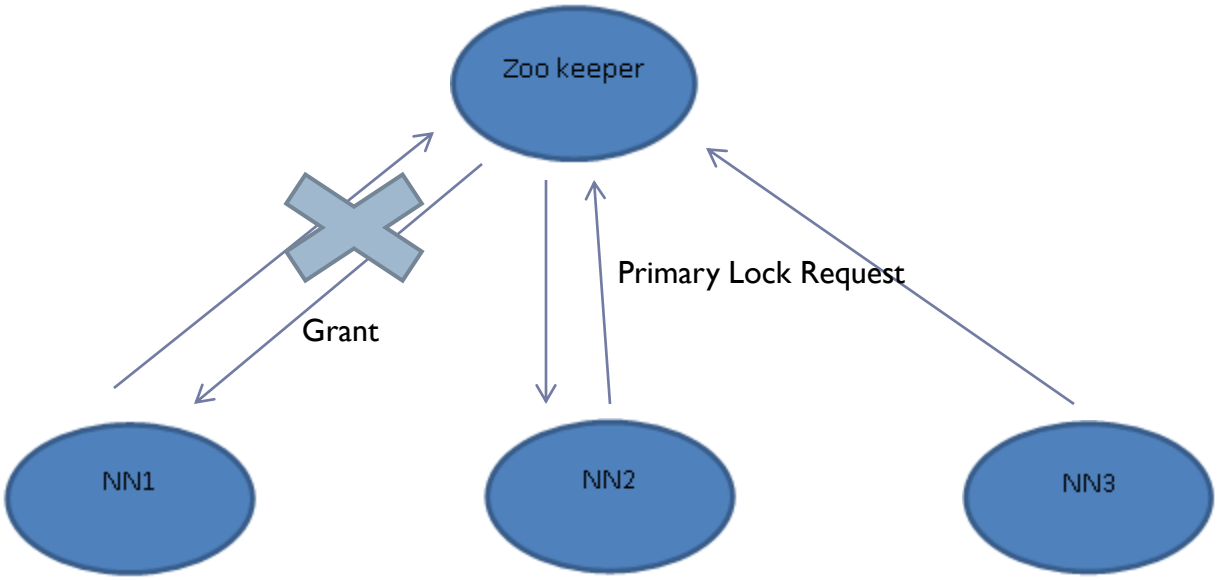
Using Zookeeper to elect leader and store a few minutes of Live Edit Logs

- Multiple Name Nodes run identical code, one of Name Nodes is selected by Zookeeper as Master role (leader) in random fashion (Same as Hbase Master election).
- Name Node can join the cluster at any time and copies persisted FS Image and Logs from Active Leader.
- A few minutes of Live Logs are stored to Zookeeper, which has higher availability and reliability.
- Background Live Log Publisher takes stored Live Logs bind them to one Edit Log File and distributes it to all live Name Nodes.

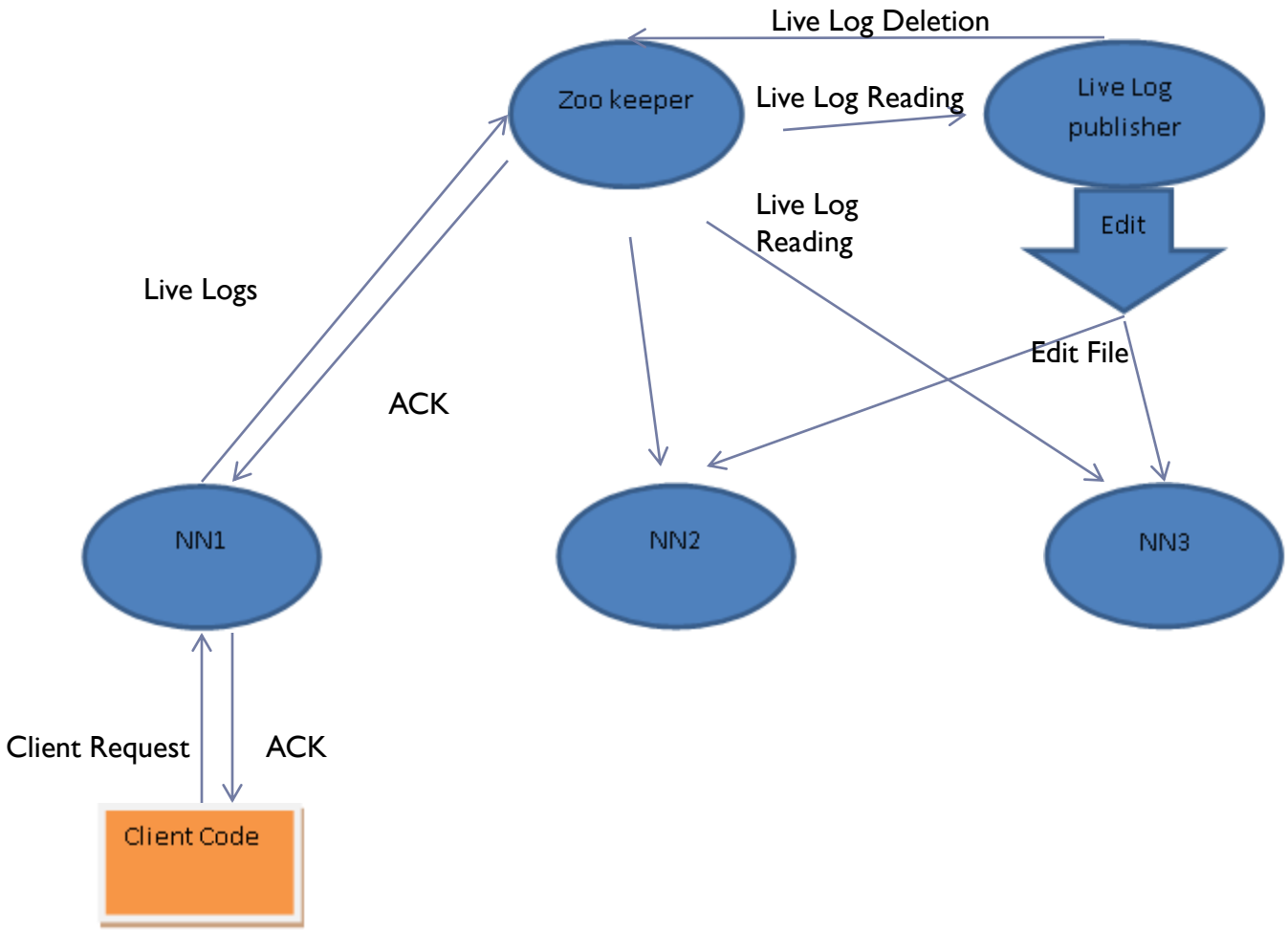
Client Library Code Name Node Resolver



NN Leader Election



Logs Flow


Live Log Deletion
Zoo keeper
Live Log Reading
Live Log publisher
Live Log Reading
Edit
Live Logs
ACK
Edit File
NN1
NN2
NN3
Client Request
ACK
Client Code

Persisted Image and Edits are tied to persisted Log ID. Last Persisted Log ID stored in Zookeeper.

- New properties of Log ID introduced to system.
- All Edit Log files and Image files can be identified by start and end Log ID.
- Image persisted Log ID always starts from 0, persisted Log Ending ID is stored in VERSION file. (Old Image File without this property will consider end with 0 as well.)
- Edits persisted ID is identified by its name and formatted as 'EDITS.start.end'

Any Redundant Name Node catches up with the leader

- Name Node joins cluster as Redundant Name Node.
- Before contesting for the leadership, it has to have latest persisted data (Image/Edit, last Persisted Log ID is stored in Zookeeper).
- Redundant Name Node will copy the persisted data from leader Name Node if it doesn't have latest Persisted data.

Live Log Publisher distributes newly generated Persisted Log to all Redundant Name Node.

- Log Distribute Task periodically packs Live Logs in Zookeeper and generates Persisted Edit Log file.
- Then it distributes packed Log file (Edits.start.end) to all Live Name Nodes. If distribution silently failed, it will try next time.
- It only updates Zookeeper Last Persisted Log ID and deletes persisted Live Log if distribution is successful.
- If Redundant Name Node Persisted Log is out of sync it has to catch up again by itself.

Image & Edits File on local machine

- |-dfs
- |---name
- |-----current
- |---permanentlog

- [david@d-sjn-00528874-CentOS-64-01 dfs]$ ls name/current
- Edits
- edits.0000000000000000000.00000000000000000003
- edits.0000000000000000003.00000000000000102199
- edits.0000000000000102199.00000000000000230206
- edits.0000000000000230206.00000000000000358613
- edits.0000000000000358613.00000000000000487334
- fsimage
- fstime
- VERSION

- [david@d-sjn-00528874-CentOS-64-01 dfs]$ cat name/current/VERSION
- #Mon Nov 07 10:01:50 PST 2011
- namespaceID=619115194
- imageMD5Digest=35ef82cf0d8aa8f1d660c33b131905ea
- nextPersistId=0
- cTime=0
- storageType=NAME_NODE
- layoutVersion=-33

Example of Other Name Nodes

- [david@d-sjn-00528874-CentOS-64-02 dfs]$ ls permanentlog/current/
- edits fsimage fstime VERSION

- [david@d-sjn-00528874-CentOS-64-02 dfs]$ cat permanentlog/current/VERSION
- #Mon Nov 07 10:02:49 PST 2011
- namespaceID=619115194
- imageMD5Digest=35ef82cf0d8aa8f1d660c33b131905ea
- nextPersistId=487334
- cTime=0
- storageType=NAME_NODE
- layoutVersion=-33

Data in Zookeeper

- Master lock: Whoever takes this lock become Master. Zookeeper guarantees that only one has this lock.
- Master Persisted data synchronization point: Redundant Name Node can download completed Persisted data from this point when it tries to catch up.
- Persisted Image/Log Meta data: Meta data describes the Persisted Image/Log FIFO properties (like Persisted Log ID and others).
- Live Log data: Master loads its Live Log to Zookeeper Log FIFO when serving the request.
- Beside the leader Name Node, all Redundant Live Name Nodes are also registered with Zookeeper

Zookeeper Meta Example

- [zk: localhost:2181(CONNECTED) 4] ls /hadoop
- [joinedsync, primary]
- [zk: localhost:2181(CONNECTED) 5] ls /hadoop/joinedsync
- [d-sjn-00528874-CentOS-64-01, d-sjn-00528874-CentOS-64-04, d-sjn-00528874-CentOS-64-02]

- [zk: localhost:2181(CONNECTED) 7] ls /hadoop/primary
- [lock, data, mastersync, meta]
- [zk: localhost:2181(CONNECTED) 10] ls /hadoop/primary/meta
- [lid, pid, nid]
- [zk: localhost:2181(CONNECTED) 11] ls /hadoop/primary/data
- [19, 35, 17, 36, 18, 33, 15, 34, 16, 13, 14, 37, 11, 12, 21, 20, 22, 23, 24, 25, 26, 27, 28, 29, 3, 2, 10, 1, 0, 30, 7, 6, 32, 5, 31, 4, 9, 8]

Client and Data Node get information from Zookeeper.

- Client and Data Node can resolve Leading Name Node from Zookeeper.
- Data Node can also get all Redundant Name Nodes and send Block Report information to all Redundant Name Nodes.

Q & A