Testing Consistent Reads from Standby Node

Konstantin Shvachko
09/03/2018

Overview

Reads from Standby is a new feature intended to readdress read requests from the Active NameNode to Observer Node(s). This allows the NameNode ensemble to handle more RPC calls. See more detail in the design document (HDFS-12943). This document sets the plan for testing and benchmarking of this feature.

Unit Tests

- Unit tests are based on MiniDFSCliuster
  - MiniDFSCliuster should start three (or more) NameNodes: Active, Standby, and Observer(s).
  - Configure HDFS clients to use ObserverReadProxyProvider.
  - Enable Fast Path (HDFS-13150) for tailing journal logs.
- Main functionality
  - For each read operation (annotated as @ReadOnly in ClientProtocol) ensure that the request is executed on the Observer.
  - For each write operation, ensure that it is executed on the Active.
- Consistency tests
  - Read after write. For read operations ensure that the read from Observer is not executed until the write is propagated to Observer via the journal.
  - One clients writes, another client reads. Ensure consistency.
- Failover tests
  - Standard HDFS HA failover tests should succeed when run with Active, Standby, and Observer and clients running ObserverReadProxyProvider.
  - Ensure that Active cannot be transitioned to Observer and vice versa.
  - Ensure that Observer can be transitioned to Standby and vice versa.
  - Introduce the single Observer failure. Reads should be automatically redirected to Active NameNode.
  - Run MiniDFSCliuster with multiple Observers. If one Observer fails, readers should be automatically redirected to the remaining Observers.
  - Emulate Observer node falling far behind the Active. Ensure readers switch over to another Observer instead of waiting for the lagging Observer to catch up. If there is only a single Observer, it should fall back to the Active.
Live Cluster Tests

Live cluster configuration includes four NameNodes: one Active, one Standby, and two Observers.

1. SLive test (HDFS-708).
   SLive allows to specify proportions of different HDFS operations on the cluster. Tune SLive according to the RPC % load on an existing cluster, and run it with these parameters. Run overnight. Check audit logs.

2. Reconfigure Balancer to retrieve block information from Observers. This should not affect the balancing process, but should reduce load on the NameNode.

3. Run Wordcount job as a basic test.
   Verify that audit logs on Active NN contain writes, and logs on Observers contain reads.

   GridMix runs in two modes:
   a. Submit a mix of synthetic jobs to the cluster
   b. Submit a mix of synthetic jobs based on traces generated from history files of real jobs modeling workload on an existing cluster.

5. Pig test.
   a. Run Pig integration tests
   b. Run PigMix

6. Hive test
   Run Hive integration tests

7. Presto testing on HDFS
   Run Presto queries

8. Spark Testing
   Run Spark Examples

9. Azkaban / Oozie / Airflow testing
   Run acceptance and integration tests

10. ObserverReadProxyProvider should support Configured- and IP- failover models.

Benchmarks

1. Dynamometer benchmark
   Setup a Dynamometer Active and Dynamometer Observer NameNodes. Run traces from existing logs at varying speeds.
   a. Maximize overall execution time
   b. Maximize latency of RPC requests
   c. Monitor RPC queue size

2. Run SLive with 5-10% writes and 90-95% reads. Fixed number of operations.
   a. Compare overall execution time (expect 2x performance gain)
   b. Monitor RPC queue size and latency of RPC requests
3. DFSIO with small files and / or tiny blocks
   a. Write benchmark.
      Performance should not degrade compared to single NameNode
   b. Read benchmark.
      Performance should scale proportionally to the number of Observers

4. Pig benchmark
   Run PigMix is a set of queries used to test Pig performance.

5. Hive benchmark
   Run Hive Testbench

6. Presto benchmarks
   Run TPC benchmark queries

7. Spark benchmark candidates
   a. Spark-Bench
   b. HiBench Suite

References

1. Consistent Reads from Standby Node design document
2. HDFS-12943. Consistent Reads from Standby Node
3. HDFS-13150. Fast Path to tail in-progress journal logs