Building an Impenetrable ZooKeeper

Kathleen Ting
Who Am I?

- Apache Sqoop Committer, PMC Member
- Author of *Apache Sqoop Cookbook*
- Technical Account Manager, Cloudera
- @kate_ting, kathleen@apache.org
What is ZooKeeper?

- Coordinator of distributed applications
- Small clusters reliably serve many coordination needs
- Canary in the Hadoop coal mine
3 ZooKeeper Ensemble

![Diagram of ZooKeeper Ensemble]

- ZK Leader
  - write to ZK Followers
  - read requests from Clients
- ZK Followers
  - write to ZK Leader
  - read requests from Clients
ZooKeeper’s Data Model

/  
  
  Boston
    
    Red Sox
    Patriots

  New York
    
    Yankees
    Giants
Why is ZooKeeper Important?

- **High Availability**
  - Replicate to withstand machine failures

- **Distributed Coordination**
  - One consistent framework to rule coordination across all systems
  - Observe every operation by every client in exactly the same order
Who Uses ZooKeeper?

- HBase
- MapReduce (YARN)
- HDFS (High Availability)
- Solr
- Kafka
- S4
- Storm
- Accumulo
- Search

- Numerous custom solutions: [https://cwiki.apache.org/confluence/display/ZOOKEEPER/poweredby](https://cwiki.apache.org/confluence/display/ZOOKEEPER/poweredby)
ZooKeeper Coordination Example

App --> HBase --> ZooKeeper --> HDFS --> JVM / Linux --> Disk/Network --> MR
What are Misconfigurations?

- Any diagnostic ticket requiring a change to ZooKeeper (HBase, Hadoop..) or to OS config files

- Comprise 44% of tickets

- e.g. resource-allocation: memory, file-handles, disk-space
Ticket Breakdown by Type

- Misconfig: 44%
- Bug: 34%
- App: 10%
- JVM/Linux: 8%
- Disk/NW: 4%
Ticket Breakdown by Component

- ZooKeeper: 43%
- HBase: 34%
- Pig: 10%
- Flume OG: 7%
- HDFS: 3%
- System: 3%
ZooKeeper Anti-Patterns

- Connection Mismanagement
- Time Mismanagement
- Disk Mismanagement
ZooKeeper Anti-Patterns

- Connection Mismanagement
- Time Mismanagement
- Disk Mismanagement
1. Too Many Connections

WARN [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:2181:NIOServerCnxn $Factory@247] - Too many connections from /xx.x.xx.xxx - max is 60

How can it be resolved?

- Running out of ZK connections?
  - Set maxClientCnxns=200 (per ZK node) in zoo.cfg
- HBase or Hive client leaking connections?
  - Manually close connections
  - Fixed in HBASE-3777, HBASE-4773, HBASE-5466, HIVE-3723
- Turn off IPv6 or java.net.preferIPv4Stack=true
2. Connection Closes Prematurely

ERROR: org.apache.hadoop.hbase.ZooKeeperConnectionException:
HBase is able to connect to ZooKeeper but the connection closes immediately.

How can it be resolved?

- If hbase.cluster.distributed = true in hbase-site, then in zoo.cfg, quorum can’t be set to localhost
- In hbase-site, set hbase.zookeeper.recoverable.waittime=30000ms
  - Provides enough time for HBase client to try another ZK server
  - Fixed in HBASE-3065
3. Pig Hangs Connecting to HBase

WARN org.apache.zookeeper.ClientCnxn: Session 0x0 for server null, unexpected error, closing socket connection and attempting reconnect java.net.ConnectionException: Connection refused

What causes this?
- Location of ZK quorum is not known to Pig (default 127.0.0.1:2181 fails)

How can it be resolved?
- Use Pig 10, which includes PIG-2115
- If there is overlap between TaskTrackers and ZK quorum nodes
  - Set hbase.zookeeper.quorum to final in hbase-site.xml
  - Otherwise, add "hbase.zookeeper.quorum=hadoophbasemaster.lan:2181" to "pig.properties" (fixed in PIG-2821)
ZooKeeper Anti-Patterns

- Connection Mismanagement
- Time Mismanagement
- Disk Mismanagement
4. Client Session Timed Out

INFO org.apache.zookeeper.server.ZooKeeperServer: Expiring session <id>, timeout of 40000ms exceeded

How can it be resolved?

- ZK and HBase need same session timeout values:
  - zoo.cfg: maxSessionTimeout=180000
  - hbase-site.xml: zookeeper.session.timeout=180000
- Don’t co-locate ZK with IO-intense DataNode or RegionServer
- Make sure your session timeout is sufficiently long
- Specify right amount of heap and tune GC flags
  - Turn on Parallel/CMS/Incremental GC
5. Clients Lose Connections

WARN org.apache.zookeeper.ClientCnxn - Session <id> for server <name>, unexpected error, closing socket connection and attempting reconnect

java.io.IOException: Broken pipe

Don’t use SSD drive for ZK transaction log

- ZK optimized for mechanical spindles and for sequential IO
- SSD provides little benefit and suffers from high latency spikes
  - SSD pre-allocates disk extents to avoid directory updates but that doubles the load on the SSD
  - SSD disk stops for 40 sec (which is greater than session timeout)
ZooKeeper Anti-Patterns

- Connection Mismanagement

- Time Mismanagement

- Disk Mismanagement
6. Unable to Load Database – Unable to Run Quorum Server

FATAL Unable to load database on disk
java.io.IOException: Failed to process transaction type: 2
error: KeeperErrorCode = NoNode for <file> at
org.apache.zookeeper.server.persistence.FileTxnSnapLog.restore
(FileTxnSnapLog.java:152)

How can it be resolved?
- Archive and wipe /var/zookeeper/version-2 if other two ZK servers are running
7. Unable to Load Database – Unreasonable Length Exception

FATAL Unable to load database on disk
java.io.IOException: Unreasonable length = 1048583

at
org.apache.jute.BinaryInputArchive.readBuffer(BinaryInputArchive.java:100)

How can it be resolved?

- Server allows a client to set data larger than the server can read from disk
- If a znode is not readable, increase jute.maxbuffer
  - Look for "Packet len <xx> is out of range" in the client log
  - Increase it by 20%
  - Set in JVMFLAGS=-Djute.maxbuffer=yy bin/zkCli.sh
- Fixed in ZOOKEEPER-1513
8. Failure to Follow Leader

WARN org.apache.zookeeper.server.quorum.Learner: Exception when following the leader java.net.SocketTimeoutException: Read timed out

What causes this?
- Disk IO contention, Network Issues
- ZK snapshot is too large (lots of ZK nodes)

How can it be resolved?
- Reduce IO contention by putting dataDir on dedicated spindle
- Increase initLimit on all ZK servers and restart, see ZOOKEEPER-1521
- Monitor network (e.g. ifconfig)
## Optimal Ensemble Size

<table>
<thead>
<tr>
<th># of ZK Servers</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Coordination</td>
</tr>
<tr>
<td>3</td>
<td>Reliability for production environment</td>
</tr>
<tr>
<td>5</td>
<td>Permits taking one server down for maintenance</td>
</tr>
</tbody>
</table>

Why not run 11 ZK servers?
Trust But Verify

- zk-smoketest
  - https://github.com/phunt/zk-smoketest
  - Verify new, updated, & existing installations
  - Identify latency issues

- zk-top
  - https://github.com/phunt/zktop
  - Unix “top” like utility for ZK

- 4 letter words/JMX (e.g. ruok, srvr)
  - http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_zkCommands
  - Use "stat" to get an idea what your request latency looks like
ZooKeeper Patterns

DOs

- Separate spindles for dataDir & dataLogDir
  - Avoids competition between logging and snapshots
  - Improves throughput and latency
- Allocate 3 or 5 ZK servers
- Tune Garbage Collection
- Run zkCleanup.sh script via cron

DON’Ts

- Don’t co-locate ZK with I/O intense DataNode or RegionServer
  - ZK is latency sensitive
- Don’t use SSD drive for ZK transaction log
Configure ZooKeeper Correctly..

..and it’ll be as impenetrable as a distributed system allows.

**ZK OFFICE HOURS**

Starts in 10 Min

@ Expo Hall (Table 1)

**SQOOP BOOK SIGNING**

First 25 Copies Are Free

Tomorrow Noon

@ O’Reilly Booth