Security Level:

ZooKeeper In The Wild

ApacheCon Budapest 2014

www.huawei.com

Rakesh Radhakrishnan
rakeshr@apache.org
HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

$whoami

- Technical Lead Engineer, Huawei India R & D
- Apache ZooKeeper committer
- Apache BookKeeper committer
- Hadoop user

rakeshr@apache.org

https://www.linkedin.com/in/rakeshadr

https://twitter.com/rakeshadr

Agenda

- Introduction
- ZooKeeper in Hadoop
- Pitfalls
  - Deploy ZooKeeper cluster
  - Manage ZooKeeper cluster – Maintain & Monitor
  - Use ZooKeeper Client

What is ZooKeeper ?

- ✓ A high performance centralized coordination service for distributed applications.
- ✓ Simplifies the implementation of many advanced patterns in distributed systems like,

  configuration store,

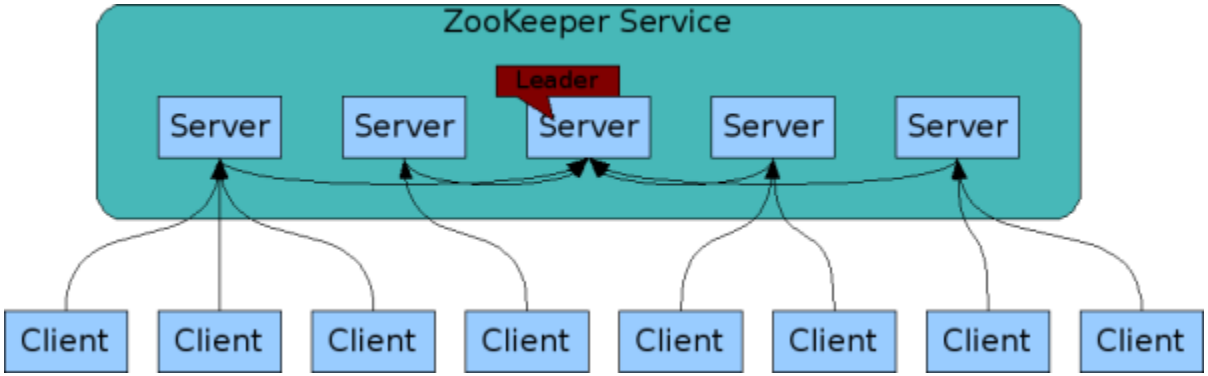
  distributed lock,

  queueing,

  leader election,

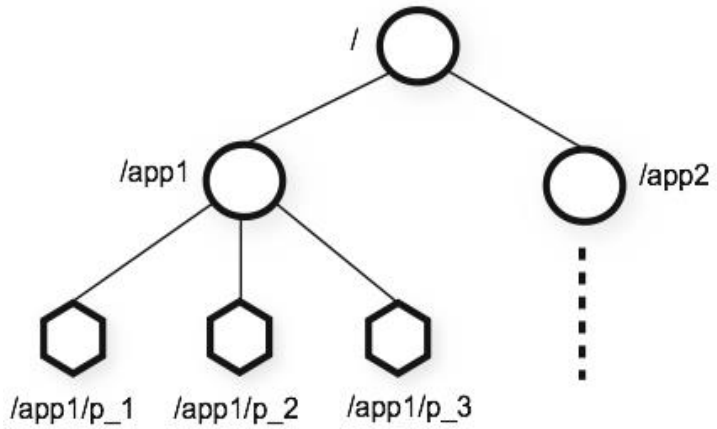
  coordination and many more..

ZooKeeper Architecture



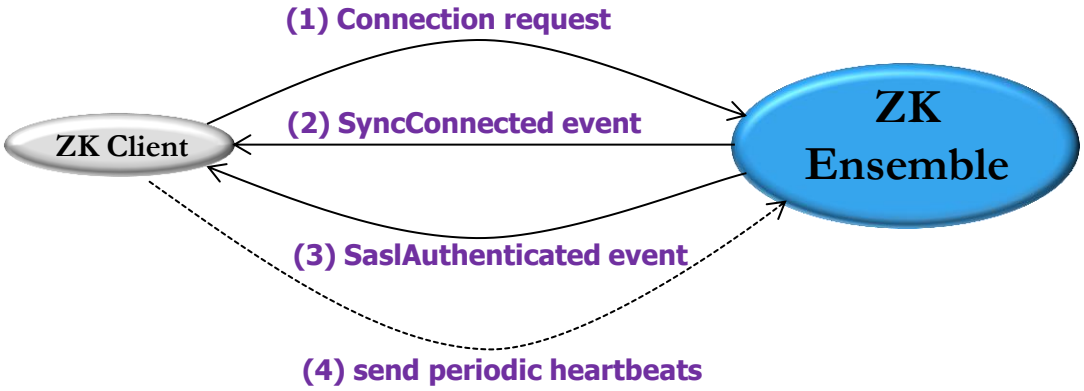
- ✓ **distributed** over a set of machines and **replicated**.
- ✓ all servers store a **copy of the data** (in memory as well as local file system)
- ✓ a **LEADER** is elected at the startup
- ✓ LEADER will do **atomic broadcast** to all other servers (**Z**ooKeeper**A**tomic**B**roadcast)
- ✓ strong **ordering guarantees**
- ✓ **no partial** read/writes

ZooKeeper Data Model



- ✓ **hierarchical** namespace
- ✓ each node in the namespace is called as a **'zNode'**
- ✓ every zNode in the name space is **identified by a path** (for example, /app1).
- ✓ zNode types – **persistent** and **ephemeral**
- ✓ each zNode has **data** and **optionally can have children**
- ✓ **no rename** of zNode
- ✓ **add/remove watchers** to the zNode

ZooKeeper Session

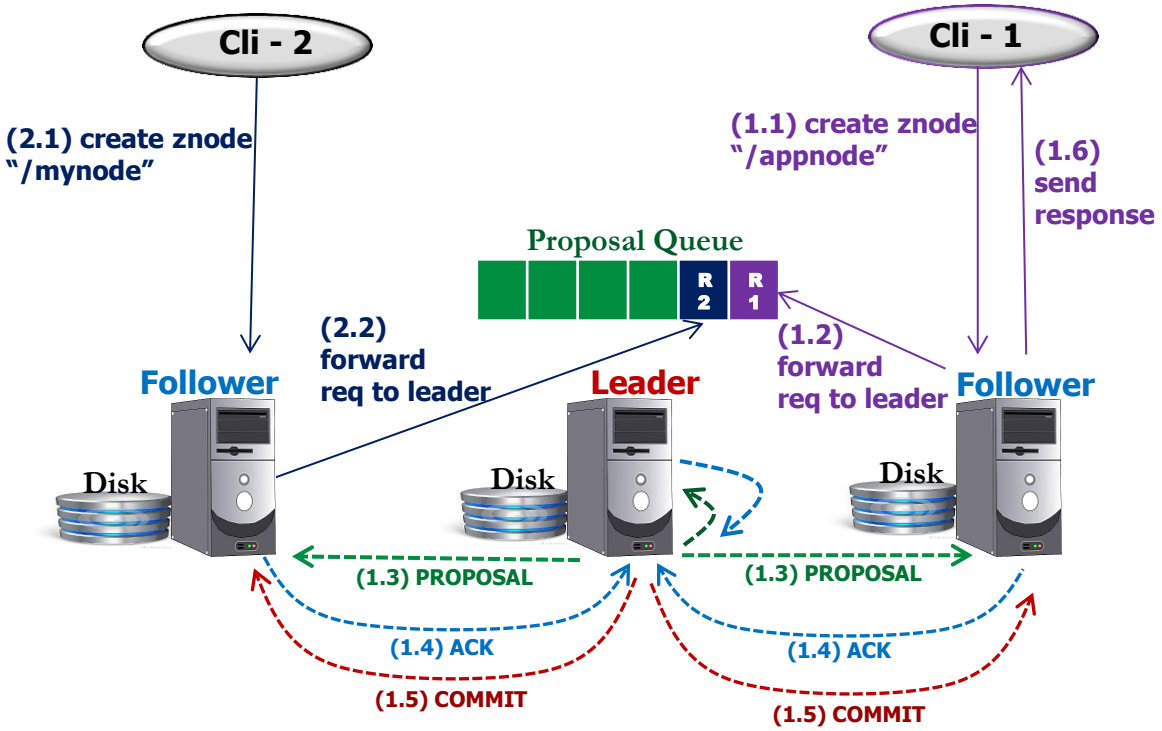


- ✓ client establishes a **session** with a single server
- ✓ session has **uniqueid** and **timeout**
- ✓ auto keep-alive **heartbeats**
- ✓ automatic **failover**
- ✓ sends request to server in **FIFO order**
- ✓ **no operation retries** in case of connection loss

Connection Events are:

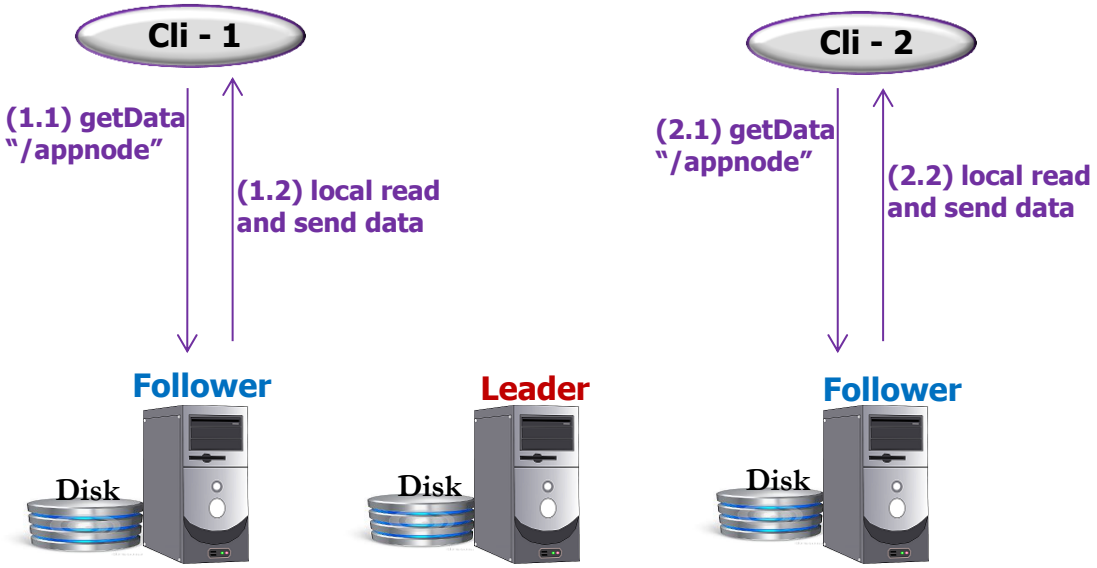
- SyncConnected
- SaslAuthenticated
- Disconnected
- AuthFailed
- Expired

Writes



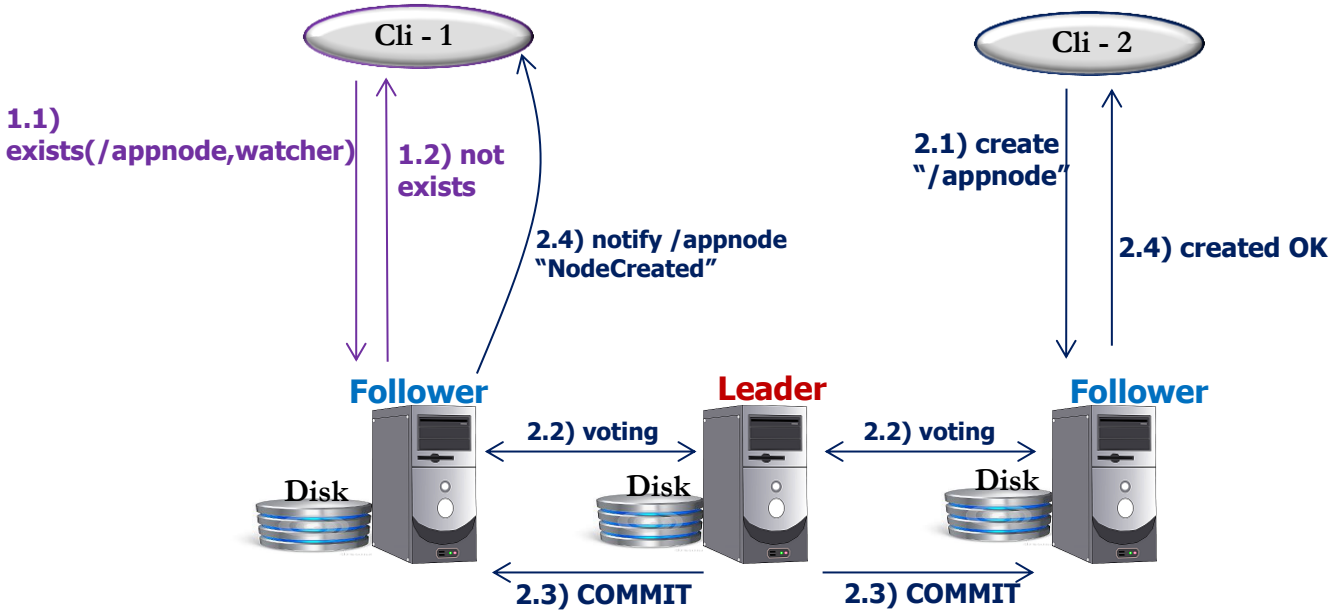
- ✓ all updates **routed through leader**
- ✓ every update has **a unique transaction id (zxid)**
- ✓ needs **majority consensus**
- ✓ **persists** the change on disk before sending response to client

Reads



- ✓ read from the **connected server memory**

Watches



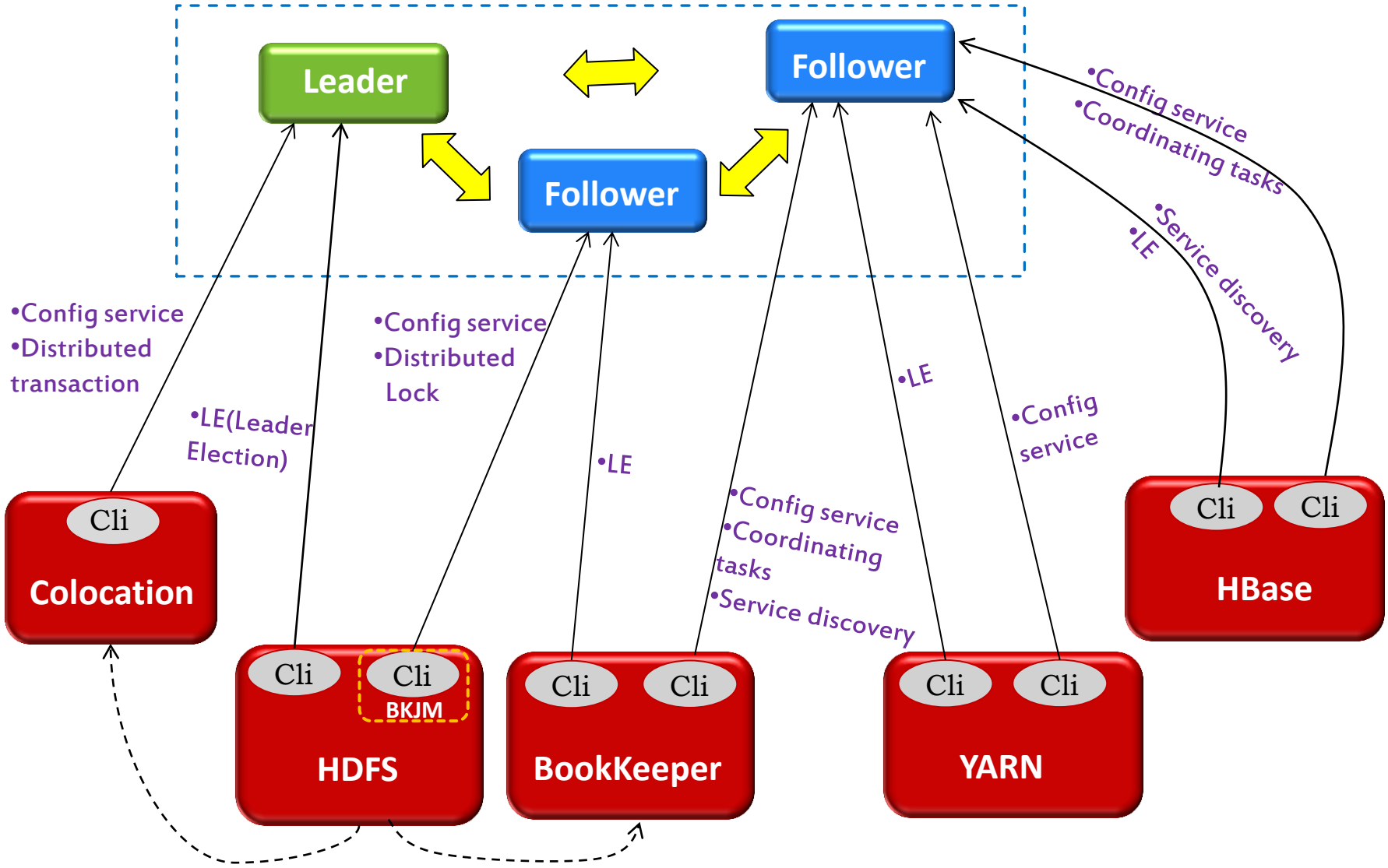
- ✓ **one time** trigger
- ✓ will sent notifications **asynchronously** to the watchers(callback object)
- ✓ watches will be triggered **after committing** the changes to disk
- ✓ two types of watches – **data & child watches**
- ✓ client can **remove watches**

ZooKeeper APIs

- ✓ simple APIs
- ✓ sync & async versions

Operation	Type
create	Write
getData	Read
setData	Write
exists	Read
getChildren	Read
multi	Read/Write
reconfig	Write
setACL	Write
sync	Write
setWatches	Read
removeWatches	Read
delete	Write

ZooKeeper in hadoop



How to…

Deploy ZK cluster

Manage ZK cluster

Use ZK client

Pitfalls…

Deployment

1) Ideal number of servers

2) Design topology

3) Choose right server role

4) Configuration is the key

Management

1) Disk full

2) JMX MBeans are missing

3) How to reconfigure ZK cluster

4) Expecting things to fail - A lot

Use Client

1) Operation after Connection loss

2) Herd Effect

3) Number of direct children

4) High latency with many writes

5) Multi-tenancy at session level

6) Careful about Watchers

Deployment

Cases:

1) Ideal number of servers

2) Design topology

3) Choose right server role

4) Configuration is the key

Case#1: Ideal number of servers ?

Scenario

Solution

Quorum = Leader + Followers,

(2n+1) nodes can tolerate failure of 'n' nodes.

Case#2: Design topology

Scenario

Hadoop & other processes in a single machine:

- Datanode
- RegionServer
- NodeManager
- **ZooKeeper** ✗
- Other user application(s)

Solution

✓ Don't **colocate** ZooKeeper server with I/O intense processes.

✓ Dedicated disk for **dataLogDir** to reduce disk contention

✓ Datatree resides in memory, **should have enough JVM heap size.**

For example, if you have 4G of RAM, do not set the Java max heap size to 6G or even 4G. Avoid situation in which ZooKeeper swaps to disk.

Case#3: Choose right server role

Scenario

Adding new Servers will affect –

- ✓ write performance
- ✓ the availability

Solution

ZK Server can be either **Participant** (a voting member) or **Observer** (a non-voting member)

Voting member:

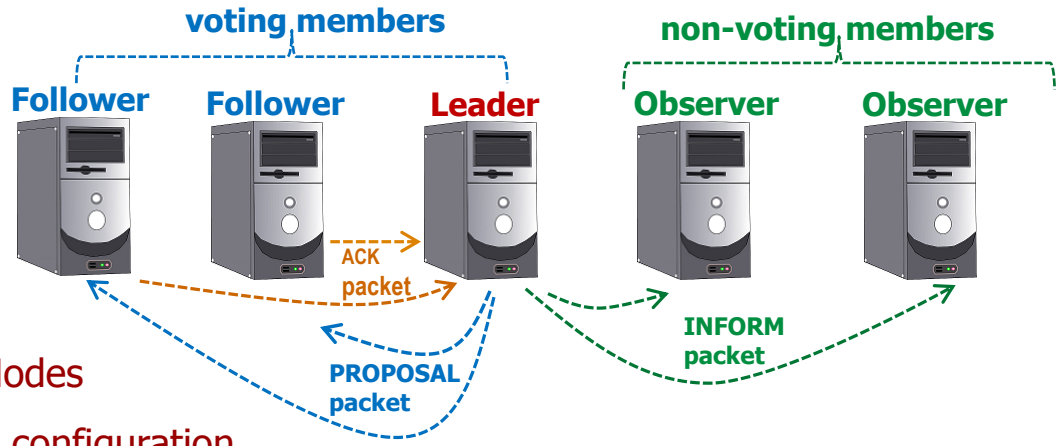
member of an ensemble which will **vote on proposals** and can become **Leader**

Non-voting member:

member of an ensemble which only **hear the results of votes**

✓ **Choose server role as OBSERVER**

Case#3: Choose right server role



ZK Server as Observer:

- ✓ leader will inform the changes of zNodes
- ✓ can easily add by simple changes in configuration

zoo.cfg file

```
51  # Set the server mode to observer
52  peerType=observer
53
54  # adding :observer to the server definition line of each Observer
55  server.1=HOST-10-151-40-19:2888:3888:observer
```

Benefits:

- ✓ big-performance improvement for read-heavy workloads
- ✓ allows to scale without hurting write performance
- ✓ failure of Observer server will not affect the availability

Example usecase:

- ✓ As a datacenter bridge - ensemble runs entirely in one datacenter, and the second datacenter runs only Observers. Therefore voting protocol doesn't takesplace across a high-latency intra-datacenter link, and improves performance.

Case#4: Configuration is the Key

Scenario

Changing Host configurations like **vm.swappiness**

High value of swappiness will make the kernel page out, can cause **ZooKeeper times out**

- ✓ It can be set to a value between **0-100**; the higher the value, the more aggressive the kernel is in seeking out inactive memory pages and swapping them to disk.
- ✓ Swapping to disk can cause **operations to timeout** and potentially fail if the disk is performing other I/O operations.

Solution

- ✓ High value may cause problems such as **lengthy garbage collection pauses**
- ✓ It is important to avoid swapping, which will seriously **degrade ZK performance**
- ✓ Recommends **to set the value to 0**

Case#4: Configuration is the Key

Scenario

ZooKeeper cluster is unstable and not forming quorum

Ensemble detail **is not consistent in all the machines**

zoo.cfg file

```
29 # ZooKeeper server and its port no.
30 # ZooKeeper ensemble should know about every other machine in the ensemble
31 # specify server id by creating 'myid' file in the dataDir
32 # use hostname instead of IP address for convenient maintenance
33 server.1=HOST-10-151-40-19:2888:3888
34 server.2=HOST-10-151-40-20:2888:3888
35 server.3=HOST-10-151-40-21:2888:3888
36 server.4=HOST-10-151-40-22:2888:3888
37 server.5=HOST-10-151-40-23:2888:3888
```

Solution

✓ **ZooKeeper configurations**

- **Ensemble details** should be same in every machine

Management: Maintain & Monitor

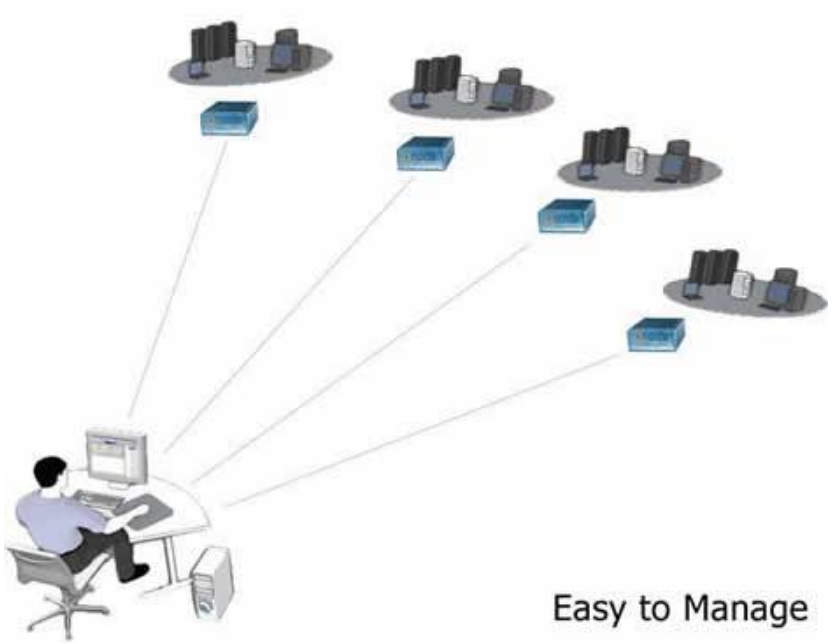
Cases:

1) Disk full

2) JMX MBeans are missing

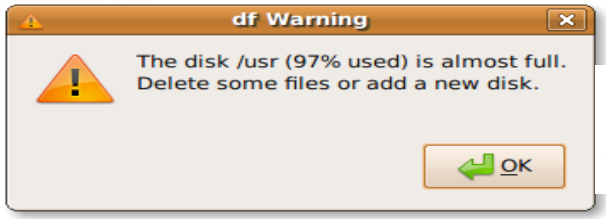
3) How to reconfigure ZK cluster

4) Expecting things to fail - A lot



Case#1: Disk full

Scenario



Due to **old snapshots & transaction logs**

zoo.cfg file

```
39  #
40  # Be sure to read the maintenance section of the
41  # administrator guide before turning on autopurge.
42  #
43  # http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
44  #
45  # The number of snapshots to retain in dataDir
46  autopurge.snapRetainCount=3
47  # Purge task interval in hours
48  # Set to "0" to disable auto purge feature
49  autopurge.purgeInterval=1
```

Solution

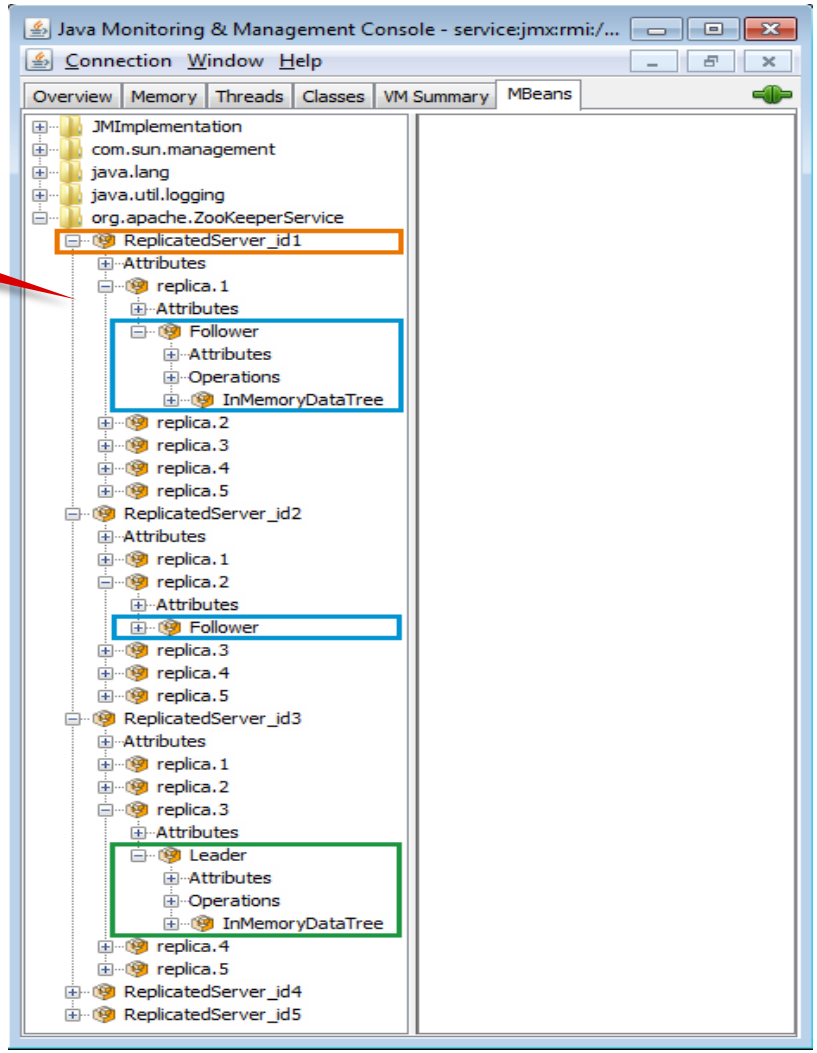
✓ cleanup dataDir & dataLogDir by scheduling **purge task**

Reference:
http://zookeeper.apache.org/doc/trunk/zookeeperAdmin.html#Ongoing+Data+Directory+Cleanup

Case#2: JMX MBeans are missing ?

Scenario

LeaderElection **MBean is unregistered and not available**



✓ has **hierarchy** of MBeans

✓ need to **enable** remote JMX

✓ MBeans will be **registered** and **de-registered** dynamically based on the server state.

Server states are:

- **LOOKING** (forming quorum)
- **FOLLOWER**
- **LEADER**
- **OBSERVER**

Solution

✓ Expect MBeans based on server state.

Case#3: How to reconfigure ZK cluster ?

Scenario

- machine is **slow**
- machine is **dead**
- wants to modify **address, port, server role**
- **add** a new server

Solution

✓ dynamic reconfiguration of ZK cluster feature is available since **3.5.0-alpha release**.

✓ use **reconfig()** client APIs or shell admin commands to,

- **add** new member
- **remove** an existing member
- **update** address & ports, server roles

✓ **updateServerList()** client API

-a probabilistic client re-balancing algorithm (move the client to a new server)

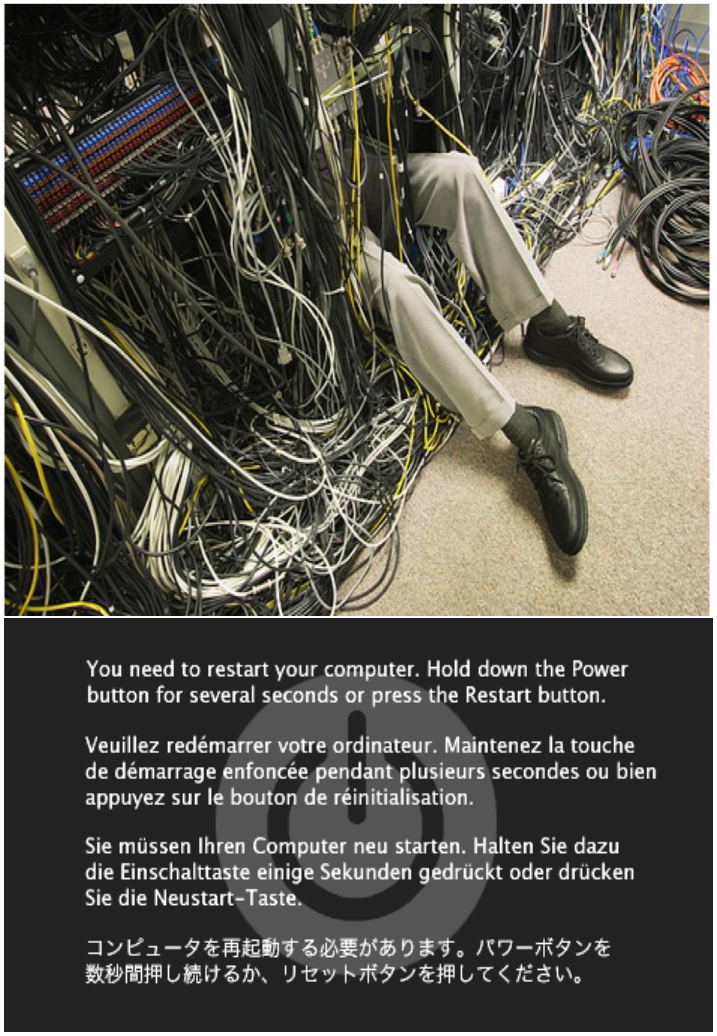
Reference: https://issues.apache.org/jira/browse/ZOOKEEPER-107

Case#4: Expecting things to fail – A lot

Remember,

- ✓ **Hardware -**
  power supplies, hard drives, network outages etc.

- ✓ **Operating System -**
  Kernel panics, zombie processes, dropped packets etc.



Use ZooKeeper Client

Cases:

1) Operation after Connection loss

2) Herd Effect

3) Number of direct children

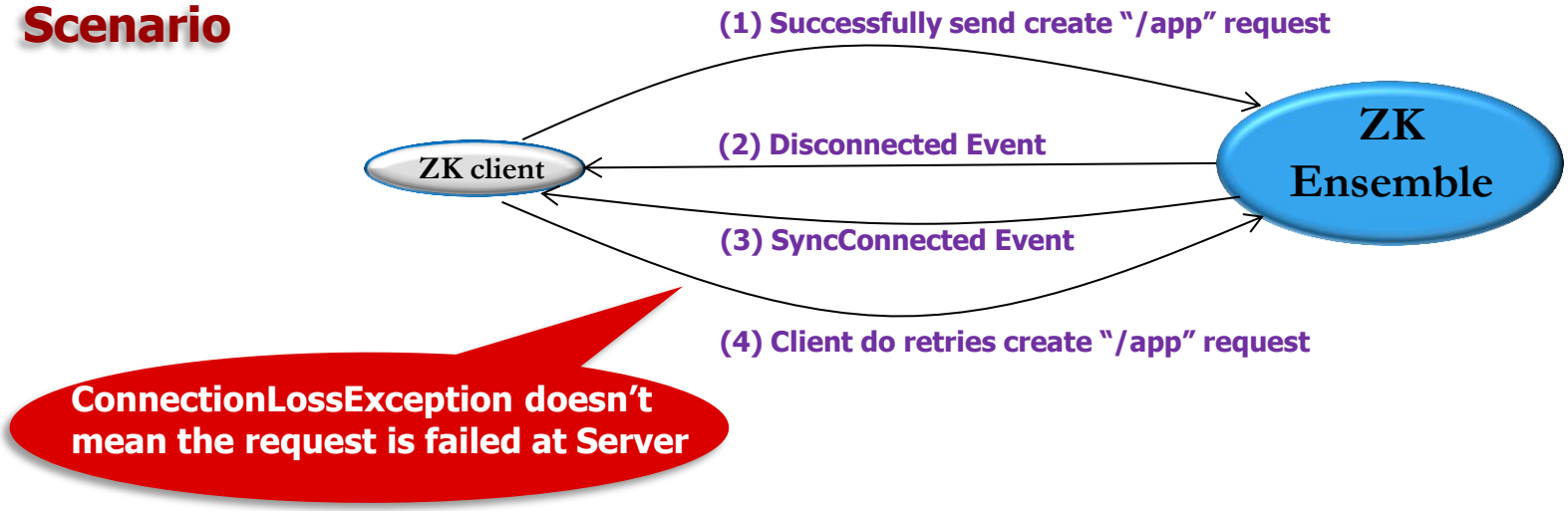
4) High latency with many writes

5) Multi-tenancy at session level

6) Careful about Watchers

Case#1: Manage operation after connection loss

Scenario



Solution

Since the request successfully reaches quorum, quorum will continue with voting and update the changes based on the ACK responses.

Retrying ZooKeeper Client can either **read the update before next retries** or **handle KeeperExceptions** properly

Case#2: Herd Effect

Scenario **too many watches on a single zNode…**

- ✓ cause large spike on n/w traffic.
- ✓ one such example is **distributed lock impl**

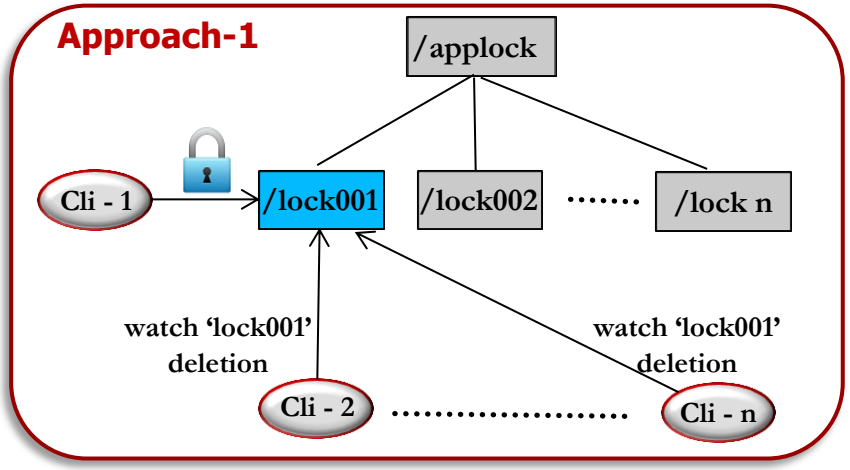


Figure-1: Watch on "lock" zNode deletion

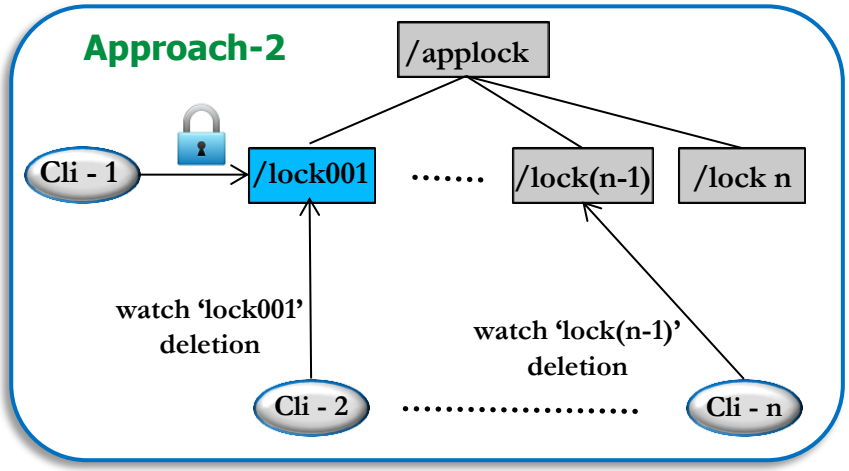


Figure-2: Watch on "predecessor" zNode deletion

Solution

Distributed Lock Recipe:

step-1: client who created the zNode with least sequence number will always get the lock

step-2: others will watch for zNode deletion and not interested on every master lock changes

Approach-2 : has significant benefits of **less number of watch notifications**.

Case#3: Number of children on a single znode

Scenario

For example,

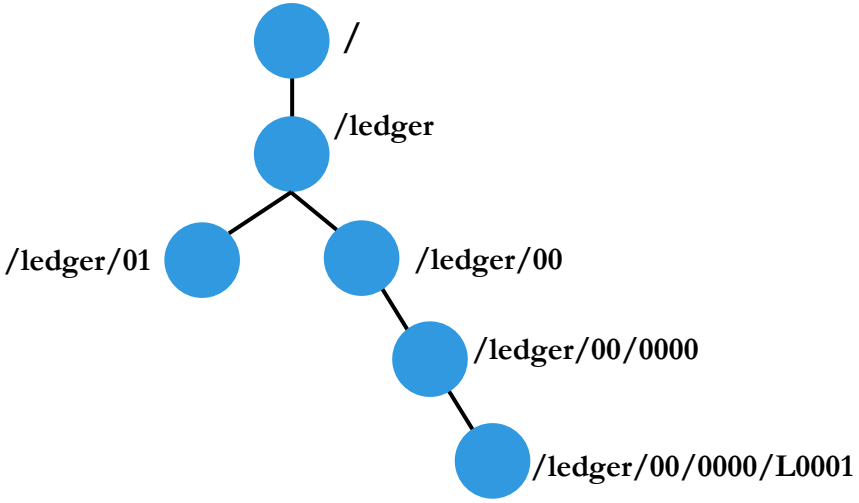
Assume 'zNode name' is less than 100 bytes, you probably want to limit the maximum number of children to **10,000**.

can't grow the list of children to more than 1 MB 'jute.maxbuffer' (the sum of the names of all of the children)

Solution

Recommended hierarchical structure.

Reference: **Apache BookKeeper** stores ledger metadata hierarchically.

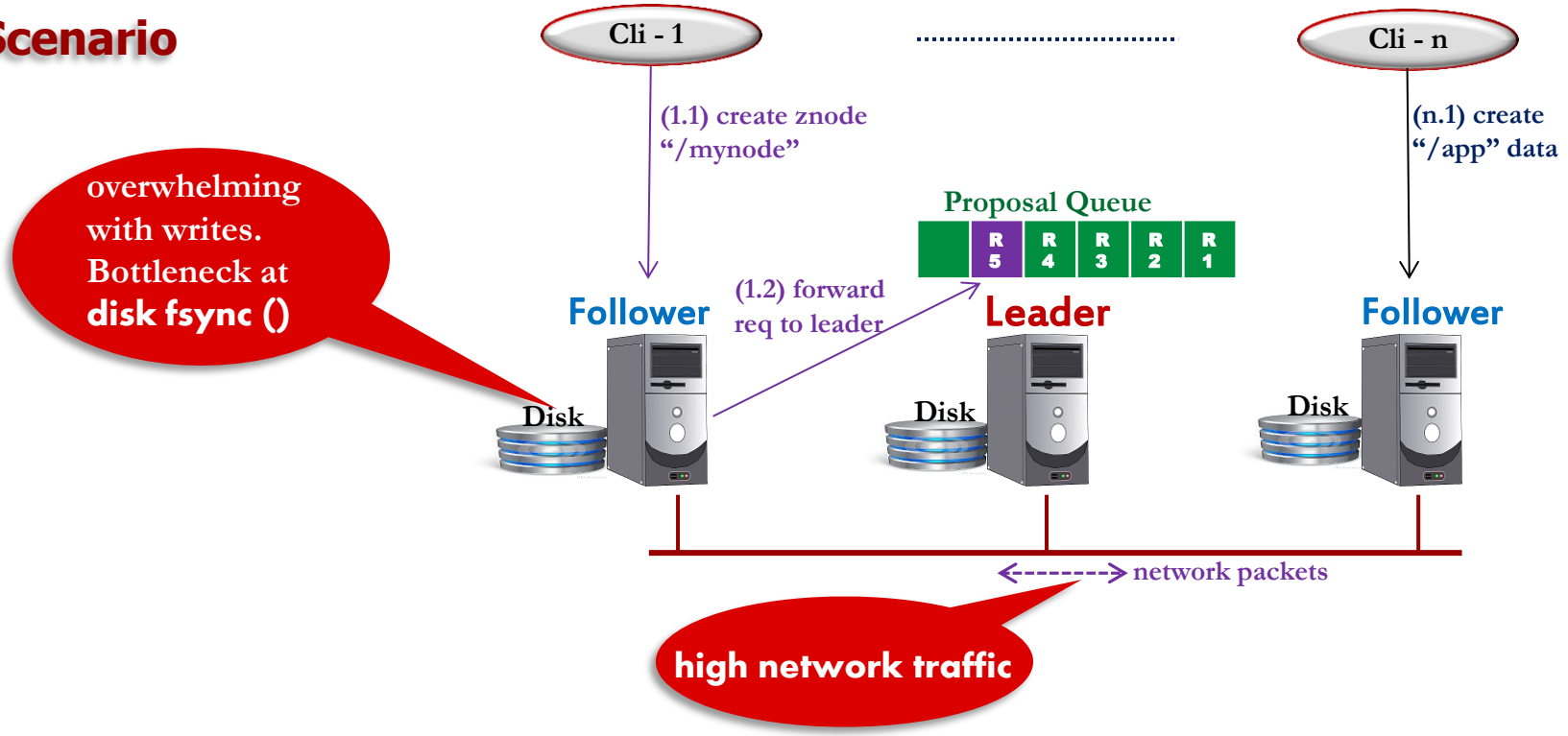


Example, L0000000001 is split into 3 parts (level1/level2/L(level3), which is stored in ledger/00/0000/L0001.

Here BookKeeper can store upto **9999999999** number of ledgers.

Case#4: High latency with many writes

Scenario



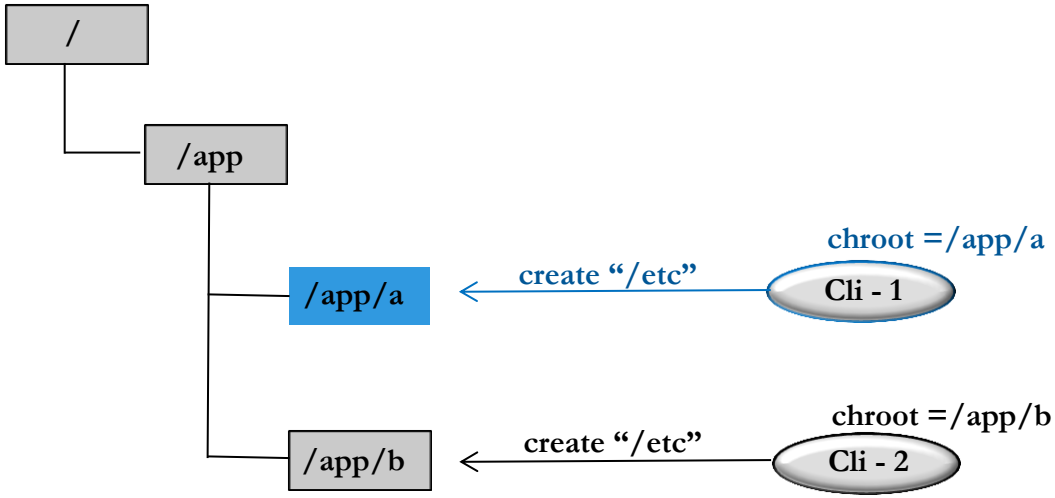
Solution

- ✓ meant for **80% read** and **20% write**,
- ✓ should be careful on the number of simultaneous write operations

Case#5: Multi-tenancy at session level

Scenario

- ✓ Multiple applications share the same Zookeeper ensemble.
- ✓ User application need to make sure that znodes are unique.



Solution

- ✓ An optional **"chroot" suffix** also be appended to the connection string.
  For example, "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002/app/a"
- ✓ client can code his/her application as if it were rooted at "/", while actual location (say /app/a) and **all paths would be relative to this**

Case#6: Careful about Watchers

Scenario

Sometimes watch notification is missing

Solution

- ✓ If a **znode changes multiple times** between getting the event and setting the watch again, it will miss the changes. Carefully handle this.
- ✓ Recommends to process the watch notifications **in separate application thread**

A few points to remember

Deployment

- Highly available – decide tolerated failures
- Choose right server role
- Don’t colocate with IO intense processes

Manage

- Cleanup datadir & transaction logs periodically
- Dynamic reconfiguration APIs to change topology

Use Client

- Connection loss doesn’t mean operation failed at Server
- ZK meant for 80% read and 20% write
- Careful when adding many children under single zNode
- Careful about Watchers

Questions ?

Who uses ?

Yahoo

Twitter

Facebook

Huawei

Netflix

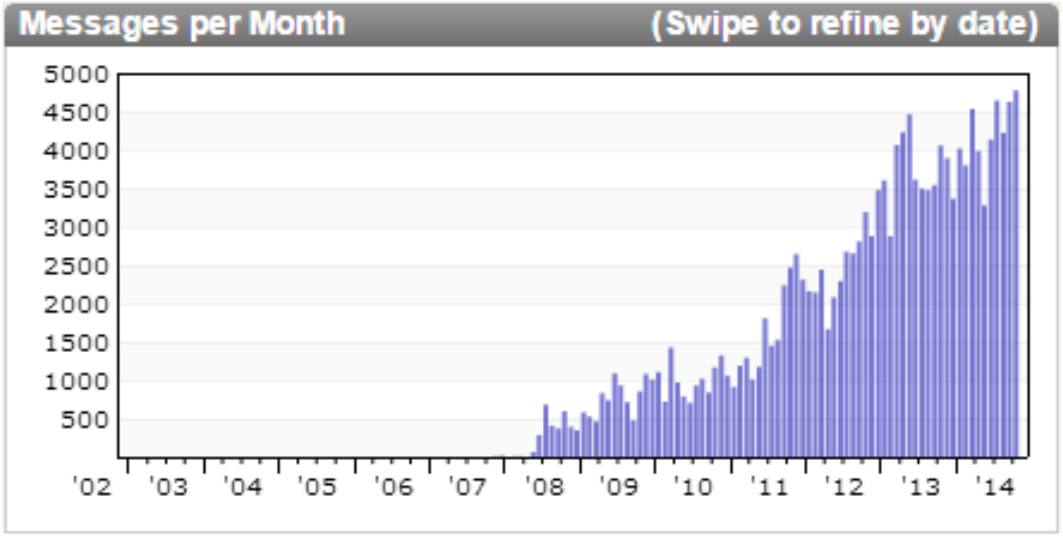
Rackspace

Apache Projects like, Hadooop, BookKeeper, Solr, Kafka etc.

and many more…

https://cwiki.apache.org/confluence/display/ZOOKEEPER/PoweredBy

Community



http://apache.markmail.org/search/?q=zookeeper

Client libraries

- ✓ **Java** and **C** clients available with ZooKeeper release

curator.apache.org

Where are we ?

Released

- ✓ **3.5.0-alpha** on 6 August, 2014
- ✓ **3.4.6 (stable version)** on 10 March, 2014

Future Directions

- ✓ **Wire encryption** - https://issues.apache.org/jira/browse/ZOOKEEPER-2063
- ✓ **Stabilize dynamic reconfiguration feature**
- ✓ **Improve usability, reliability**

References

- ✓ http://zookeeper.apache.org/
- ✓ https://wiki.apache.org/hadoop/ZooKeeper
- ✓ http://hadoop.apache.org/
- ✓ http://zookeeper.apache.org/bookkeeper/
- ✓ http://web.stanford.edu/class/cs347/reading/zab.pdf

Photo Attributions

https://zookeeper.apache.org/doc/r3.4.6/zookeeperOver.html

https://peterskastner.files.wordpress.com/2011/02/it-guy-0013.png

http://www.learnhowtoorap.com/images/how-to-write-pic.jpg

http://en.wikipedia.org/wiki/Kernel_panic#mediaviewer/File:MacOSX_kernel_panic.png

http://www.pd4pic.com/images/tower-computer-server-cpu-processor-box-cartoon.png

http://png-2.findicons.com/files/icons/1676/primo/128/lock.png

https://www.terilogy.com/product/sevone-eng/images/img_feature01.jpg

http://www.inode.gr/img/administration_small.jpg

http://marketingmotivator.net/wp-content/uploads/2010/09/Handshake_business-man-and-woman-correct_iStock_000000306702Small.jpg

http://hadoop.apache.org/images/hadoop-logo.jpg

http://etc-mysitemyway.s3.amazonaws.com/icons/legacy-previews/icons/simple-red-square-icons-alphanumeric/128241-simple-red-square-icon-alphanumeric-question-mark3.png

Thank you

www.huawei.com