

LargeClusterTips

Below are tips for managing large clusters.

- Have a good sysadmin if you're not one yourself.
- Take a look at a presentation done by Allen Wittenauer: <http://tinyurl.com/5foamm>
- Have the LAN closed off to untrusted users. Without this, your filesystem is effectively open to everyone on the network.
- Once you are on the private LAN, turn off all firewalls on the machines, as it only creates connectivity problems.
- Use LDAP or similar to manage user accounts.
- Only put the slaves file on your namenode and secondary namenode to prevent confusion.
- Use RPMs to install the Hadoop binaries. [Cloudera](#) provide some RPMs for this, and a web site to generate configuration RPM files.
- Use kickstart or similar to bring up the machines.
- If you are trying to configure the machines one by one, step away from the keyboard. That is not the way to manage a cluster.
- Keep an eye out for disk SMART messages in the server logs. They warn of trouble.
- Keep an eye on disk capacity, especially on the namenode. You do not want the NN to run out of storage as Bad Things happen.
- Keep the underlying software in sync: OS, Java version.
- Run the rebalancer, throttled back appropriately for your bandwidth

Management Tool Support

1. See the [AmazonEC2](#) and [AmazonS3](#) pages for tips on managing clusters built on EC2 and S3.
2. See the [Management Tools](#) Page for details on integration with Management Tools
3. Other good documentation: [Patterns of Hadoop Deployment](#)

Hadoop Configuration

- Don't do it by copying XML files around by hand.
- Look at the cloudera config tools. If you use them, keep the previous RPMs around
- Consider a system configuration management package to keep Hadoop's source and configuration consistent across all nodes. Some example packages are bcfg2, [SmartFrog](#), Puppet, cfengine, etc.
- Keep your site XML files under SCM, so you can roll-back, diff changes.

NameNode Health

The [NameNode](#) is a SPOF. When it goes offline, the cluster goes down. If it loses its data, the filesystem is gone. Value it.

- <https://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-hdfs/HDFSHighAvailabilityWithQJM.html>
- Never let its disks fill up.
- Consider RAID storage here. If not, set it to save its data to two independent drives, ideally on separate controllers (just in case the controller decides to play up)
- Set the NN up to save one copy of all its data to a remote machine (NFS?), so even if the NN goes down, you can bring up a new machine with the same hostname for everything else to bind to.

Workers

- Have identical hardware on all workers in the cluster, eliminating the need to have different configuration options (task slots, data directory locations, etc.)
- Have a common user account on every machine you run Hadoop on, with a common public key in `~/.ssh/authorized_keys`
- Track HDDs, their history and their failures. Disk failures are not always independent in a large datacentre.
- Have simple hostname to rack or IP to rack mappings, so the rack detection scripts are trivial.

How to rebalance a full datanode

If a datanode is at or near 100% capacity,

1. Decommission the node: this will copy everything off. 2. Take it offline. 3. Delete the data, clean up the HDDs. 4. Add the node again.

Testing Failure

Things will go wrong. There is always SPOF. Test your failure handling processes before you go live.

- Simulate a corrupted edit log by killing the namenode process, truncating the (binary) edit log, and bringing it up. See how the team handles it.
- Turn off one of the switches, pull out a network cable. See how the cluster handles it, how it recovers. Then put the switch back on.
- Turn an entire rack off without warning. See what happens when they go offline.
- Turn off DNS. Or just the rDNS side of things.
- Turn off the entire datacenter, switch it back on. Are there any race conditions?
- Write an job that tries to generate too much data, fills up the cluster. How is it handled?