Virtual Hadoop

Virtual Hadoop

A recurrent question on the user mailing lists is "can Hadoop be deployed in virtual infrastructures", or "can you run Hadoop 'in the Cloud'", where cloud means "separate storage and compute services, public or private".

These are actually two separate, but interrelated, questions, since many cloud infrastructure depend upon virtualization to manage and present an aggregation of infrastructure components that can be quickly configured to meet a user's need. Cloud and virtualization need to be examined separately, but in all cases the answer is "Yes you can virtualize, and yes, you can deploy to the cloud, but you need to know the consequences and plan accordingly".

First, some definitions and background:

Virtualization is the process of converting from a purely physical implementation to one using a hypervisor (examples include VMware's ESXi and the Xen hypervisor) which abstract the underlying physical hardware and provide an idealized, or virtual, implementation upon which some higher-level services and /or implementations can be designed and built. Once a physical cluster is virtualized, then higher level services, such as cloning a data node, or providing high-availability to a specific node, or providing user controlled provisioning, can be built.

A Private Cloud is a collection of virtualized physical hardware that has added services such as catalogs of software or defined platforms that a customer can control. A private cloud differs from a public cloud in that it is generally owned and or managed by the same company or group as the customer. As an example, if I am responsible for a 100-node physical cluster, and I need to share it between Sales and Marketing that wish to perform advanced analytics with Hadoop and Engineering that wants to perform modelling of a new production plant, with each getting 50% of the capacity, I could virtualize the physical architecture and allow the pool of capacity to be shared between the competing groups, perhaps on a shared capacity or a swap-in/swap-out basis.

A Public Cloud is like a Private Cloud but owned and/or managed by an outside entity, for example Amazon Web Services. Public Clouds can provide cost benefits, either because you only pay for your use, or other's pay for their use, but at the loss of control or of intermingling of data or other undesirable issues. Not being able to prove constant custody of some types of data might be a legal liability for certain types of data or industries (PCI, HIPAA).

Strengths of VM-hosted Hadoop

- A single image can be cloned - lower operations costs.
- Hadoop clusters can be set up on demand.
- Physical infrastructure can be reused.
- You only pay for the CPU time you need.
- The cluster size can be expanded or contracted on demand.

This sounds appealing, and is exactly what Amazon Elastic MapReduce offers: a version of Hadoop on a Pay-as-you-go-basis.

For more customized deployments, Apache Whirr can be used to bring up VMs, something documented by Cloudera. There have also been demonstrations of alternate systems running on different infrastructures, such as shown in Farming Hadoop in the Cloud.

VMware has been active in the area of supporting Hadoop on in virtual infrastructures. You can read their take on the benefits of virtualizing Hadoop and also other resources about deploying and running Hadoop in virtual infrastructures. It works with Hadoop community on Hadoop Virtualization Extention to enhance Hadoop's topology awareness on virtualized platform, which is part of the Apache Hadoop release 1.2.0+.

Does this mean that Hadoop is ideal in virtualized infrastuctures? It can be when properly provisioned. Cloud? It depends on the cloud providers.

Hadoop's Assumptions about its infrastructure

Hadoop was written expecting to run in large physical datacenters. Such datacenters and physical hardware, storage and networking has some specific features.

1. A large cluster of physical servers, which may reboot, but generally recover, with all their local on-server HDD storage.
2. Non-RAIDed Hard disks in the servers. This is the lowest cost per Terabyte of any storage. It has good (local) bandwidth when retrieving sequential data: once the disks start seeking for the next blocks, performance suffers badly.
3. Dedicated CPUs; the CPU types are known and clusters are (usually) built from homogeneous hardware.
4. Servers with monotonically increasing clocks, roughly synchronized via an NTP server. That is: time goes forward, on all servers simultaneously.
5. Dedicated network with exclusive use of a high-performance switch, fast 1-10 Gb/s server Ethernet and faster 10 + Gb/s "backplane" interconnect between racks.
6. A relative static data network topology: data nodes do not move around.
7. Exclusive use of the network by trusted users.
8. High performance infrastructure services (DNS, reverse DNS, NFS storage for NameNode snapshots)
9. The primary failure modes of machines are HDD failures, re-occurring memory failures, or overheating damage caused by fan failures.
10. Machine failures are normally independent, with the exception of the failure of Top of Rack switches, which can take a whole rack offline. Router /Switch misconfiguration can have a similar effect.
11. If the entire datacenter restarts, almost all the machines will come back up along with their data.

Hadoop's implementation details

This translates into code features.

1. HDFS uses local disks for storage, replicating data across machines.
2. The MR engine scheduler that assumes that the Hadoop work has exclusive use of the server and tries to keep the disks and CPU as busy as possible.

3. Leases and timeouts are based on local clocks, not complex distributed system clocks such as Lamport Clocks. That is in the Hadoop layer, and in the entire network stack - TCP also uses local clocks.
4. Topology scripts can be written to describe the network topology; these are used to place data and work.
5. Data is usually transmitted between machines unencrypted
6. Code running on machines in the cluster (including user-supplied MR jobs), can usually be assumed to not be deliberately malicious, unless in secure setups.
7. Missing hard disks are usually missing because they have failed, so the data stored on them should be replicated and the disk left alone.
8. Servers that are consistently slow to complete jobs should be blacklisted: no new work should be sent to them.
9. The JobTracker should try and keep the cluster as busy as possible, to maximize ROI on the servers and datacenter.
10. When a JobTracker has no work to perform, the servers are left idle.
11. If the entire datacenter restarts, the filesystem can recover, provided you have set up the NameNode and Secondary NameNode properly.

How a virtual infrastructure differs from a physical datacenter

Hadoop's assumptions about a datacenter do not always hold in a virtualized environment.

1. Storage could be one or more of transient virtual drives, transient local physical drives, persistent local virtual drives, or remote SAN-mounted block stores or file systems.
2. Storage in virtual hard drives might cause a lot of seeking if they share the same physical hard drive, even if it appears to be sequential access to the VM.
3. Networking may be slower and throttled by the infrastructure provider.
4. Virtual Machines are requested on demand from the infrastructure: the machines could be allocated anywhere in the infrastructure, possibly on servers running other VMs at the same time.
5. The other VMs may be heavy resource (CPU, IO and network) users, which could cause the Hadoop jobs to suffer. OTOH, the heavy load of Hadoop could cause problems for the other users of the server, if the underlying hypervisor lacks proper isolation features and/or policies.
6. VMs could be suspended and restarted without OS notification, this can cause clocks to move forward in jumps of many seconds.
7. If the Hadoop clusters share the VLAN with other users (which is not recommended), other users on the network may be able to listen to traffic, to disrupt it, and to access ports that are not authenticating all access.
8. Some infrastructures may move VMs around; this can actually move clocks backwards when the new physical host's clock is behind that of the original host.
9. Replication to transient hard drives is no longer a reliable way to persist data.
10. On some cloud providers, network topology may not visible to the Hadoop cluster, though latency and bandwidth tests may be used to infer "closeness", to build a de-facto topology.
11. The correct way to deal with a VM that is showing re-occuring failures is to release the VM and ask for a new one, instead of blacklisting it.
12. The JobTracker may want to request extra VMs when there is extra demand.
13. The JobTracker may want to release VMs when there is idle time.
14. Like all hosted services, a failure of the hosting infrastructure could lose all machines simultaneously though not necessarily permanently.

Implications

Ignoring low-level networking/clock issues, what does this mean? (Only valid for some cloud vendors, it may be different for other cloud vendors or you own your virtualized infrastructure.)

1. When you request a VM, it's performance may vary from previous requests (when missing isolation feature/policy). This can be due to CPU differences, or the other workloads.
2. There is no point writing topology scripts, if cloud vendor doesn't expose physical topology to you in some way. OTOH, Project Serengeti configures the topology script automatically for Apache Hadoop 1.2+ on vSphere.
3. All network ports must be closed by way of firewall and routing information, apart from those ports critical for Hadoop, which must then run with security on.
4. All data you wish to keep must be kept on permanent storage: mounted block stores, remote filesystems or external databases. This goes for both input and output.
5. People or programs need to track machine failures and react to them by releasing those machines and requesting new ones.
6. If the cluster is idle. some machines can be decommissioned.
7. If the cluster is overloaded, some temporary TaskTracker only servers can be brought up for short periods of time, and killed when no longer needed.
8. If the cluster needs to be expanded for a longer duration, worker nodes acting as both a DataNode and TaskTracker can be brought up.
9. If the entire cluster goes down or restarts, all transient hard disks will be lost (some cloud vendors treat VM disk as transient and provide other reliable storage service, but others are not. This note is only for previous vendor), and all data stored within the HDFS cluster with it.

The most significant implication is in storage. A core architectural design of both Google's GFS and Hadoop's GFS is that three-way replication onto local storage is *a low-cost yet reliable way of storing Petabytes of data.* This design is based on physical topology (rack and host) awareness of hadoop so it can smartly place data block across rack and host to get survival from host/rack failure. In some cloud vendors' infrastructure, this design may no longer valid as they don't expose physical topology (even abstracted) info to customer. In this case, you will be disappointed when one day all your data disappears and please do not complain if this happens after reading this page: you have been warned. If your cloud vendor do expose this info in someway (or promise they are physical but not virtual) or you own your cloud infrastructure, the situation is different that you still can have a reliable Hadoop cluster like in physical environment.

Why use Hadoop on Cloud Infrastructures then?

Having just explained why HDFS might not protect your data when hosted in a cloud infrastructure, is there any reason to consider it? Yes.

- For private cloud, where the admins can properly provision virtual infrastructure for Hadoop:
    - HDFS is as reliable and efficient as in physical with dedicated and/or shared local storage depending on the isolation requirements
    - Virtualization can provide higher hardware utilization by consolidating multiple Hadoop clusters and other workload on the same physical cluster

    - Higher performance for some workload (including terasort) than physical for multi CPU socket machines (typically recommended for Hadoop deployment) due to better NUMA control at hypervisor layer and reduced OS cache and IO contention with multi-VM per host than the physical deployment where there is only one OS per host.
    - Per tenant VLAN (VXLAN) can provide better security than typical shared physical Hadoop cluster, especially for YARN (in Hadoop 2+), where new non-MR workloads pose challenges to security.
- Given the choice between a virtual Hadoop and no Hadoop, virtual Hadoop is compelling.
- Using Apache Hadoop as your MapReduce infrastructure gives you Cloud vendor independence, and the option of moving to a permanent physical deployment later.
- It is the only way to execute the tools that work with Hadoop and the layers above it in a Cloud environment.
- If you store your persistent data in a cloud-hosted storage infrastructure, analyzing the data in the provider's computation infrastructure is the most cost-effective way to do so.

You just need to recognize the limitations and accept them:

- For vendors like AWS, treat the HDFS filesystem and local disks as transient storage; keep the persistent data elsewhere.
- For public cloud, expect reduced performance and try to compensate by allocating more VMs.
- Save money by shutting down the cluster when not needed.
- Don't be surprised if different instances of the clusters have different performance, or the a cluster's performance varies from time to time.
- For public cloud, the cost of persistent data will probably be higher than if you built up a physical cluster with the same amount of storage. This will not be an issue until your dataset is measure in many Terabytes, or even Petabytes.
- For public cloud, over time, dataset size grows, often at a predictable rate. That storage cost may dominate over time. Compress your data even when stored on the service provider's infrastructure.

Hosting on local VMs

As well as large-scale cloud infrastructures, there is another deployment pattern (typically for development and testing): local VMs on desktop systems or other development machines. This is a good tactic if your physical machines run windows and you need to bring up a Linux system running Hadoop, and /or you want to simulate the complexity of a small Hadoop cluster.

- Have enough RAM for the VM to not swap.
- Don't try and run more than one VM per physical host with less than 2 CPU cores or limited memory, it will only make things slower.
- Use host shared folders to access persistent input and output data.
- Consider making the default filesystem a file: URL so that all storage is really on the physical host. It's often faster (for Linux guests) and preserves data better.

Summary

You can bring up Hadoop in virtualized infrastructures with many benefits. Sometimes it even makes sense for public cloud, for development and production. For production use, be aware that the differences between physical and virtual infrastructures could pose additional gotchas to your data integrity and security without proper planning and provisioning.