Architecture of Clusters

Cluster is a configuration of nodes that interact to perform a specific task (Map Reduce)
Clusters have in the past provided high-performance processing solutions for aerospace and defense, medical, scientific, oil and gas, financial, communications and life science applications.
With the advent of the internet clusters now provide the backbone of an increasing number of important commercial entities.
Average Cluster size ~150 nodes FaceBook approx 4000 nodes.
With the advent of the Cloud it is now possible to have multilayer clusters

Virtual Cluster (Cluster of Virtual Machines) (Hadoop here)
OpenStack The core software for Cloud clusters is now open source.
Real cluster

OpenStack Software delivers a massively scalable cloud operating system.
OpenStack or 'Cloud' provides Virtual Machines' we can spin up for different types of cluster applications
Examples include Amazon(EMR) Elastic Map Reduce and Rackspace
The Sahara project provides a simple means to provision a Hadoop cluster on top of OpenStack.

Real Clusters

Compute(Cores) < -- > Network (Switch) < -- > Storage(Disk)
Nodes are core processes, can be a box/server but not always

Compute	Network (High throughput)	Storage
cores cpus algorithm high performance parallel processing capabilities take a heavy toll on cost and power consumption Intel offers QPI	switch throughput 10G to 40G to 100G Ethernet Low-latency Ethernet MACs TCP/IP protocol IP headers and addresses IP cores TCP packet routing Fiber Backbone	size latency SSD need high-speed, low-latency interfaces terabyte drives raid

High-bandwidth, low-latency servers, networking, and storage applications
no single point of failure (hadoop namenode is a single point of failure)

Why Clustering

Market capitalisation

Company	Capital (billion usd)
Facebook (use hadoop)	150
Google (invented hadoop)	363
Amazon (use hadoop)	145

Unstructured data ---> Data Processing --> Structured data
Structured data → Advanced Analytics -> Eg Visualisation(Google Maps)

Design Considerations

1. High Availability (Replication) Failover (detect when node is down)
    - no single point of failure (raid, multiple nic's ect ..)
    - multiple paths to shared storage (Fencing)
    - redundant ups
    - network should support multicasting (Nodes Interact)
    - ACPI turned off in bios(Adv Config Power Interface Sharing)
    - Linux Network Manager Service switched off)
2. Real Time Performance (Concurrent calculations)
3. Online Scalability for Operation and Maintenance (Elastic)
4 Parallel (Concurrent, ACID transactions)((Atomicity, Consistency, Isolation, Durability)
ACID properties in a distributed transaction where no single node is responsible for all data affecting a transaction presents complications solutions (two-phase commit protocol)
5. Cache memcached
Memcached is a general-purpose distributed memory caching system. Applications using Memcached typically layer requests into RAM before falling back on a slower backing store
6. Cluster to Cluster replication (Cluster of Clusters)
7. Load Balancing (Dispatch network service sends requests to multiple node to balance the request load) ensures scalability, is a algorithm
8. Multi Master replication clusters or master-slave replication clusters
10 Checkpoints backup strategy
11 Shared Nothing Architecture (Commodity Hardware)

Components of a cluster

1.Cluster Nodes {Managers, Servers Data nodes}
2.Cluster infrastructure (Basic set of cluster services that should run on any node)
3 Distributed File System (same file system on each node) (node has local linux)
4. Master Manager Server (FailOver of services running on nodes, stores this as Metadata)
5. Checkpoint backup for Disaster recovery

Master

NameNode Core Switch (Hadoop this is a single point of failure) metaData in Memory

Rack

Switch Heartbeat (I'm online)
Rack Master (High redundancy)
Server 1 Data Nodes local disks blocks (metadata for blocks parent file in HDFS stored here in local)
Server 2 local disks Data Nodes temp.dir(shuffle step data) (low redundancy)
Server n local disks Linux local filesystem

Nodes
1. node is a process not a computer but often both
2 data nodes
- generally have a lot of memory
- local file system

3. management nodes
- back up power supply, more raid ect ...
- more throughput switch bandwidth

Summary(Hadoop Focus)

dataset(huge)
---> partition dataset smaller units
---> each unit runs in parallel in its own thread
---> when finished return to master thread
lots of synchronisation problems on a single box
distribute it to many independent nodes many of these synchronisation issues vanish
need more processing power add more machines
HDFS block optimised for throughput, put get delete and append
block replication allows multi job applications, availability
block replication protects machine and rack failure

say 100TB disk space cluster n nodes . with default replication 100/3 = 33.3 TB storage capacity
ratio processors/disks => throughput perfomance