Sector – Brief Definition

Sector is an open source data cloud model.
Its Assumptions: Presence of a high bandwidth data link among the racks in a Data Center and also among different Data Centers. Also, that individual applications may have to process large streams of data and produce equally large output streams.

Sector: Architecture

Sector manages its data with the help of the following components.

Security Server : Authenticates the clients and the slave servers

Master Server : Contains File Meta Data, Schedules the work among the slave servers.

Slave Servers: Contains datasets divided amongst them in the form of Linux files.

Sector: Network Protocols

UDT for data transfer and UDP for message transfer.

Sphere: A Brief Definition

Sphere is a programming model built over the Sector architecture of data cloud. It falls under the Single instruction Multiple Data Category from Flynn;s taxonomy.

Sphere : Computing Model

Sphere identifies the individual records in a file with the help of index files.

Each record or a bunch of records are treated as independent data entities that can be processed in Parallel by different slave nodes. Similar to data parallelism.

These slave nodes are manged by Sphere Processing Engines(SPE) and the SPE are scheduled by Sphere.

Multiple stages of SPE can be coupled together.

Similarity to MapReduce

Sphere model is quite similar to Map Reduce because both of them deal with data parallelism and allow coupling of at least two stages of processing elements. Map Reduce shuffles the data among the slave nodes in the second stage whereas Sphere allows the output of the first stage to be distributed among different processing nodes of the second stage.

The Google File System

Google File System(GFS) is a FS developed and employed on the clusters of Google. It is designed to scalable, reliable, provide high availability and high throughput rather than low latency. It is designed with the Google specific applications in mind and optimizes for certain common operations and data characteristics for these applications.

GFS: Google specific assumptions

GFS assumes the applications to read/write large stream of data. It provides less support for small read. Record append is more common an activity than data modify.

Applications require high sustained throughput rather than low latency.

GFS: Architecture

GFS manages its data with the help of a Master server and several chunk servers.
A file is divided among several chunks.
The size of a chunk is kept at 64MB, larger than the common block size in native Linux FS.
Chunks are placed on different chunk servers and are replicated to add reliability.
Master server holds the meta data and places the chunks over the chunk servers.
Meta Data is kept very small and kept in the memory of the server.

GFS: Reliability

Shadow Master servers exist in the system which take over in case the primary Master Server fails.

An operation log is maintained to record all the operations happening on the Master.

Generally, 3 copies of individual chunks are maintained.

Master monitors the number of chunk copies avaialble in the cluster and also checks if any of them has become stale.

GFS: Atomic Record Append

Google applications generally append records concurrently to the files. GFS lets them provide data and decides the offset by itself for such operations. Thus is allows concurrent write appends. These write operations are guaranteed to succeed once in their entiriety.