Hadoop Overview

Data analytics has become a key element of the business decision process over the last decade. Classic reporting on a dataset stored in a database was sufficient until recently, but yesterday's data gathering and mining techniques are no longer a match for the amount of unstructured data and the time demands required to make it useful. The common limitations for such analysis are compute and storage resources required to obtain the results in a timely manner.

While advanced servers and supercomputers can process the data quickly, these solutions are too expensive for applications like online retailers trying to analyze website visits or small research labs analyzing weather patterns.

Hadoop is an open-source framework for running data-intensive applications in a processing cluster built from commodity server hardware. Some customers use Hadoop clustering to analyze customer search patterns for targeted advertising. Other applications include filtering and indexing of web listings, or facial recognition algorithms to search for specific images in a large image database, to name just a few. The possibilities are almost endless provided there is sufficient storage, processing, and networking resources.

How Does Hadoop Work?

Hadoop is designed to efficiently distribute large amounts of processing across a set of machines, from a few to over 2,000 servers. A small scale Hadoop cluster can easily crunch terabytes or even petabytes of data.

The key steps in analyzing data in a Hadoop framework are:

Step 1: Data Loading and Distribution:

The input data is stored in multiple files. The scale of parallelism in a Hadoop job is related to the number of input files. For example, for data in ten files, the computation can be distributed across ten nodes. Therefore, the ability to rapidly process large data sets across compute servers is related to the number of files and the speed of the network infrastructure used to distribute the data to the compute nodes.

Figure 1: Data Loading Step

The Hadoop scheduler assigns jobs to nodes to process the files. As a job is completed, the scheduler assigns the node another job with corresponding data. The job's data may be on local storage or may reside on another node in the network. Nodes remain idle until they receive the data to process. Therefore, planning the data set distribution and a high-speed data center network both contribute to better performance of the Hadoop processing cluster. By design, The Hadoop Distributed File System (HDFS) typically holds three or more copies of the same data set across nodes to avoid idle time. A network designer can manage storage costs by implementing a high-speed switched data network scheme. High-speed network switching can deliver substantial increases in processing performance to Hadoop clusters that span more than one server rack.

Steps 2 & 3: Map/Reduce:

The first data processing step applies a mapping function to the data loaded during step 1. The intermediate output of the mapping process is partitioned using some key, and all data with the same key is next moved to the same "reducer" node. The final processing step applies a reduce function to the intermediate data; the output of the reduce is stored back on disk.

Between the map and reduce operations the data is shuffled between nodes. All outputs of the map function with the same key is moved to the same reducer node. At this point the data network is the critical path. Its performance and latency directly impact the shuffle phase of a data set reduction. High-speed, non-blocking network switches ensure that the Hadoop cluster is running at peak efficiency.

Another benefit of a high-speed switched network is the ability to store shuffled data back in the HDFS instead of the reducer node. Assuming sufficient switching capacity, the data center manager can resume or restart Hadoop data reductions as the HDFS holds the intermediate data and knows the next stage of the reduction. Shuffled data stored on a reduction server is effectively lost if the process is suspended.

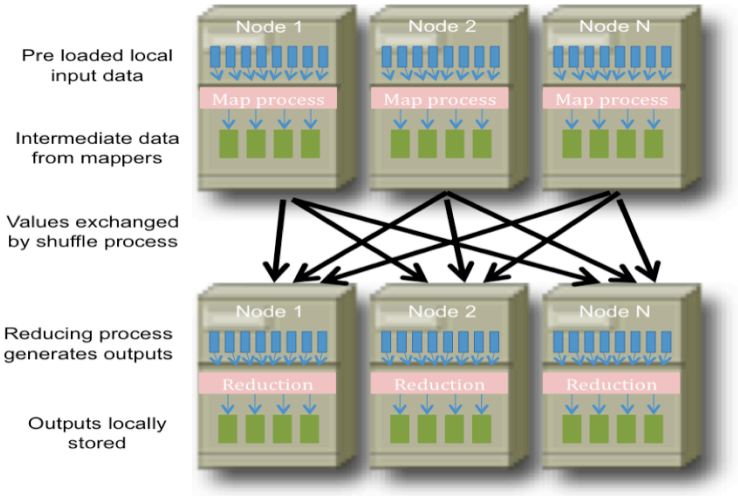

Figure 2: Map/Reduce Steps

Step 4: Consolidation

After the data has been mapped and reduced, it must be merged for output and reporting. This requires another network operation as the outputs of the reduce function must be combined from each "reducer" node onto a single reporting node. Again, switched network performance can improve throughout, especially if the Hadoop cluster is running multiple data reductions. In this instance, high performance data center switching reduces idle time, further optimizing the performance of the Hadoop cluster.

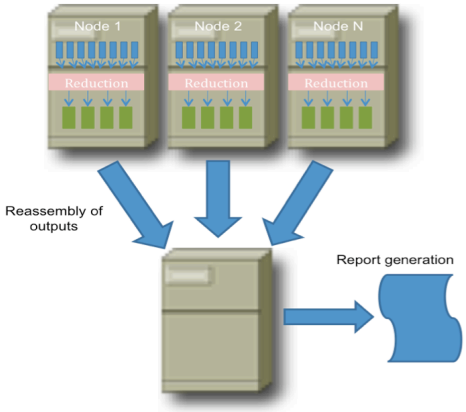

Figure 3: Consolidating Results

Impact of Network Designs on Hadoop Cluster Performance

Typically, communication bandwidth for small-scale clusters, such as those within a single rack, is not a critical issue. Nodes can communicate without significant degradation by tuning HDFS parameters. However, Hadoop clusters scale from hundreds of nodes to over ten thousand nodes to analyze some of the largest datasets in the world.

Scalability of the cluster is limited by local resources on a node (processor, memory, storage), as well as the interconnect bandwidth to all other nodes in the cluster.

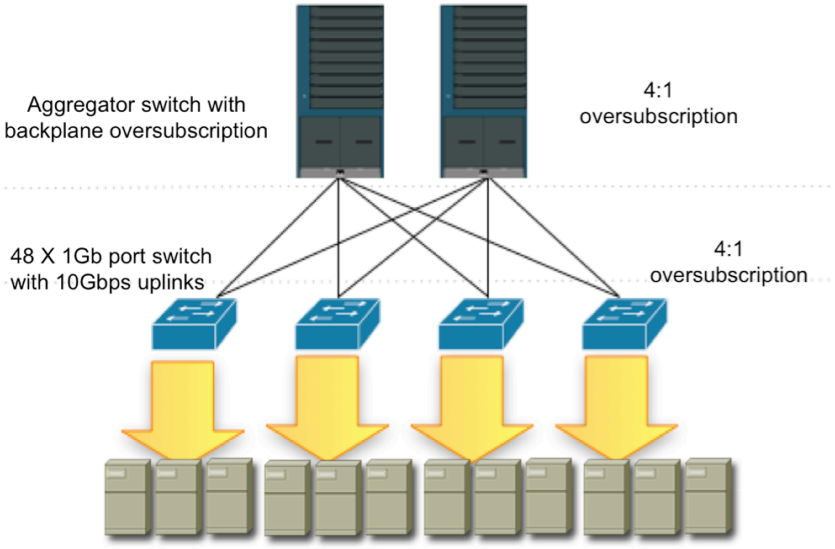

Figure 4: Legacy Network Design

Legacy network designs as shown in figure 4 lead to congestion and a significant drop in cluster performance as nodes in one rack go idle while waiting for data from another node. There is simply insufficient bandwidth between racks for the network operations to complete efficiently. In fact, the problem with these legacy designs is so significant that Hadoop 0.17 offers “rack-aware” block replication, which limits replicas of data only to nodes within the same rack. However, there is a significant performance penalty since idle resources in other racks cannot be used, thus limiting the overall efficiency of the system.

Network Designs Optimized for Hadoop Clusters

A network that is designed for Hadoop applications, rather than standard enterprise applications, can make a big difference in the performance of the cluster. The key attributes of a good network design are:

- Non-Blocking: A non-blocking design for any-to-any communication. All nodes should be able to communicate with each other without running into congestion points.
- Low Latency: A low latency, 2-tier design outperforms legacy 3-tier designs since latency at each hop is compounded by thousands of transactions within the cluster.
- Resiliency: Network node failures must not result in partitioned clusters. Additionally, network performance degradation must be smooth and predictable.
- Scale as you grow: Adding nodes or racks to an existing cluster should be seamless and not require replacing the existing network infrastructure. In essence, networking should also be viewed as a building block that can grow incrementally as additional processing nodes are added.

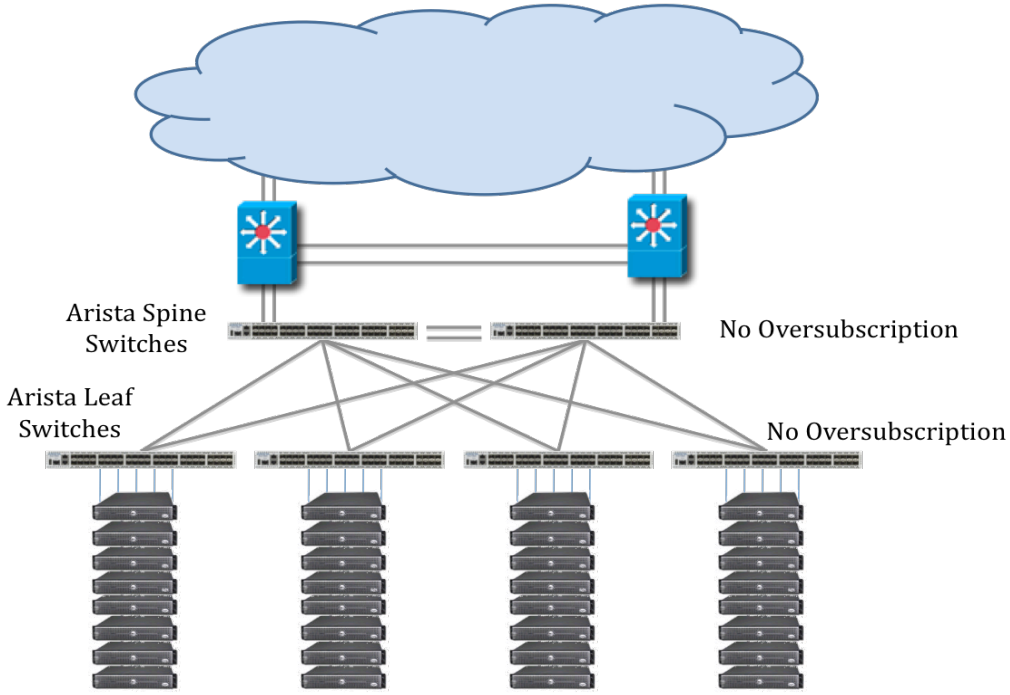

Figure 5: Non-blocking Network Design

Performance Monitoring

As a Hadoop cluster gets busy, there are peaks and troughs in the use of various resources. With careful monitoring, users can quickly identify any further optimizations in the map/reduce functions or the need to add more nodes. Ganglia[1] and Nagios[2] are two open source tools that monitor the performance of a Hadoop cluster. Ganglia can be used to monitor Hadoop-specific metrics at each individual node. Nagios can be used to monitor all resources of the cluster including compute, storage, and network I/O resources. Regardless of the tools used, a holistic view of compute and network resource usage becomes critical in monitoring a Hadoop cluster and maintaining high efficiency.

[1] ganglia.sourceforge.net
[2] www.nagios.org

Summary

Hadoop is a very powerful distributed computational framework that can process a wide range of datasets, from a few gigabytes to petabytes of structured and unstructured data. Use of Hadoop has quickly gained momentum, and Hadoop is now the preferred platform for various scientific computational and business analytics. While availability of commodity Linux based servers makes it feasible to build very large clusters, the network is often the bottleneck, resulting in congestion and less efficient use of the cluster.

Arista offers high performance 1/10 GbE non-blocking, ultra-low latency solutions that can scale from a few racks to some of the largest Hadoop deployments. In addition, Multi-Chassis LAG (MLAG) offers true active/active uplink connectivity from each rack, allowing the full bi-sectional bandwidth of the network to be utilized in a flat layer 2 network. Arista's Extensible Operating System can easily be integrated with Ganglia and Nagios. Lastly, Arista's networking solutions offer a true flat-line growth when it comes to price for server-interconnect bandwidth. All of the above factors make Arista's networking solutions ideal for any Hadoop deployment.