Internet & Web System
Term Paper

Utpal_patel@student.uml.edu
Utpal Patel (02006609)

Title: Scaling for Big Data Analysis

1. Abstract:

The big data contains great variety, huge volume data with more velocity which are way difficult for traditional data processing software to manage. Finding platform for data processing that includes both hardware and software share critical characteristics like data I/O rate, fault tolerance, real-time processing, data size support, iterative task support and most important scalability. For any system, to adopt with the increased demand of processing is called Scalability which can be categorized as: Horizontal Scaling and Vertical Scaling. Vertical scaling is nothing but increasing memory, processors and hardware within the given single server which also called 'Scale up'. On the other hand, distributing the overall workload across so many independent servers is known as Horizontal Scaling or 'Scale out'. The user will need to research the needs of their application or algorithm when deciding which platforms to use between the two scaling that is again based on i.) How fast the user wants the result? ii.) Size of the data and iii.) Number of iterations required for building model.

2. Introduction:

Big Data has become one of the dominant domains to research on, right now. Traditional data analysis systems are undergoing drastic modifications as a result of big data. It is import to scale up the hardware platforms to conduct any kind of analysis on such large and complicated data. Selecting the appropriate hardware/software platforms is essential if the user's expectations need to be satisfied in a reasonable amount of time.

Big data platforms come in a variety of forms, each with unique features, so picking the best option demands a clear understanding of what each platform is capable of. When determining whether it is appropriate to construct analytics-based solutions on a specific platform, the platform's flexibility in response to changing data processing requirements is particularly important. To find platform for data processing including hardware and software, we need to consider some critical characteristics like data I/O rate, fault tolerance, real-time processing, data size support, iterative task support and most important scalability.

3. Scalability:

Scaling refers to a system's capacity to adjust to changing data processing needs. Different platforms implement scaling in different manners to support large data processing. In general, the big data platforms are classified into the following two types of scaling:

Horizontal Scaling: This scaling distributes the workload across numerous servers. Scaling out is another name for the horizontal scaling which is the combination of numerous independent units

to increase the processing power. The operating system typically runs in numerous instances on various machines.

Vertical Scaling: Vertical scaling means adding more processors, memory, and faster hardware in a single server. It is also referred to as "scale up," because it typically just includes one operating system instance.

4. Horizontal Scaling:

This scaling involves distributing the workload among a number of servers, some of which might even be inexpensive computers. It includes dividing the overall work into numerous independent devices to increase their data processing capacity.

4.1. Apache Hadoop

Hadoop was created primarily for the HDFS and MapReduce concepts, both of which are frequently connected to distributed computing. It is well known that the central component of Hadoop called MapReduce can handle distributed data simultaneously. Hadoop's UNIX-based data storage layer (HDFS) is considered as the file system for Hadoop. The Google file system concept is the core of HDFS. Hadoop's capacity to run parallel computations on applications close to their data and to divide computational activities across multiple hosts are two of its key features. Block sequences of HDFS data files are copied throughout the cluster.

Since HDFS often offers Java API for usage in applications, there are numerous ways for programs to access HDFS. For instance, The Yahoo, Hadoop clusters have around 40,000 servers and can hold 40 petabytes of application data. 4,000 machines make up the largest Hadoop cluster. A hundred other companies are known to employ Hadoop worldwide as well.

4.2. Apache Kafka

Apache Kafka was created by the Apache Software Foundation as an open source event processing system that can be deployed because of its storage layer's great scalability and ability to handle several real-time sources of data. It can be used as a publish-subscribe messaging system and can be distributed, partitioned, or replicated to increase accuracy. Scalability along with high throughput are some of its key characteristics. One of its brokers is capable of handling many megabytes of writes and reads per second from a variety of applications. By including extra nodes in a cluster, it may be easily scaled up and by storing data on disk and executing it as a network of nodes, or brokers, data persistence is maintained.

4.3. Apache Spark

Spark has become the most active open source project for big data and has received more attention than Hadoop. It is an in-memory distributed processing framework for large-scale data analysis. It provides a straightforward programming interface so that programmers can quickly handle huge datasets across multiple servers using the CPU, memory, and storage capabilities. The Spark is

scalable, quick, tolerant, multipurpose, and user-friendly. Resilient Distributed Datasets (RDD), which can hold data and endure errors without replication, are the main abstraction used in Spark. RDDs are read-only distributed shared memory, however throughout the machine learning process, a variety of Spark implementations are offered by Spark, which also comes with the most recent versions of the Mahout, MLlib, and GraphX libraries.

5. Vertical Scaling:

In the IT industry, vertical scaling denotes the addition of resources. The administrators' addition of additional power or capability to a single component is a straightforward way to conceptualize vertical scaling.

5.1. High-performance computing (HPC) clusters

This involves parallel computations using many computing devices (CPU, GPU), as well as a quick network to link them. The most popular HPC system framework is clusters. Instead of using specialized systems, parallel computing is more effective when done on multiple servers. Machines with several cores are known as HPC clusters [59], sometimes known as blade or supercomputers. They differ in terms of their disk organization, communication method, and cache. They employ robust, powerful gear that is throughput and speed optimized. For HPC simulations, high-performance computing clusters provide the most adaptable, effective, and economical HPC platform.

5.2. Multicore CPU

A recent technology built on advancements in processing and network technologies is the multi-core processor. The substantial improvements in CPU chips over the years led to multi-core architectures, which are essential for the processing power needed for big data. It is a CPU chip architecture in which a single CPU chip contains two, four, six, or more processor cores. Due to the fact that all CPU cores share a common memory, multi-threading is the primary method for achieving CPU parallelism. The task needs to be divided into threads that can run concurrently on several CPU cores. The majority of computer languages offer libraries for CPU parallelization and thread generation. Java is the most widely used of these languages. The issue with CPUs is their limited number of processing cores and reliance on the system's memory for data access. The average system's memory is only a few hundred gigabytes, which limits the amount of data that CPUs can handle effectively. Anytime the amount of data on the CPU exceeds the capacity of the system memory, disk access becomes a challenge. The system may save the data, but because of the significantly slowed processing, memory access becomes a challenge. DDR5 memory, which is faster than the DDR3 memory utilized in computers, is employed in the GPU to circumvent this. The speed of data access is also increased by the GPU's high cache speed for each multi-processor.

5.3. Graphics Processing Unit (GPU)

A GPU is a multi-processor chip that is specifically designed for graphics. Up until recently, GPUs were primarily employed to accelerate graphics-related processing and for graphical tasks like image and video editing. However, the General-Purpose Computing on Graphics Processing Units

(GPGPU) has emerged as a result of their parallel framework and recent improvements in GPU technology. Nvidia's release of the CUDA framework has made it easier for programmers to access the GPU without needing to know specific hardware characteristics. These changes suggest that GPGPU will gain popularity. A few machine learning algorithms are implemented on GPUs using the CUDA framework by several libraries, such as GPUMiner. According to experimental tests, utilizing a GPU is faster than using a multicore CPU. GPU has various drawbacks, such as memory space limitations. Terabyte-scale data cannot be handled effectively due to the current generation's maximum memory space limit of 12 GB per GPU. Performance is greatly decreased, and disk access becomes more difficult if the data size exceeds the GPU memory. The majority of currently used analytical methods are difficult to integrate with GPUs because of the necessity to decompose workloads in GPUs.

5.4. Field programmable gate arrays (FPGA)

Since their original introduction more than 20 years ago, FPGAs have expanded quickly and developed in use in digital circuits. The logic capacity of FPGAs has been significantly improved by technological developments, making them a competitive choice for bigger designs. Additionally, the space, power consumption, and speed of the finished device are significantly impacted by the configurable nature of their routing and logic resources. Hardware Descriptive Language (HDL) was used in the programming. Due to their specialized hardware, their development costs are typically higher than those of other platforms. The expense of constructing the algorithm rises since their software must be programmed in HDL and requires a basic understanding of the hardware. Because FPGAs are only successful in certain situations, users must carefully research any application's suitability before implementing it (effective on a certain set of applications). When scanning massive volumes of network data, FPGA outperforms software firewalls. It is utilized as a hardware firewall.

6. Conclusion

This study examined the various data processing platforms that are currently in the field. We offered information on various hardware platforms as well as some well-known software frameworks like Spark and Hadoop. This study served as a foundation for the analysis of the examined platforms' performance, particularly in terms of their prowess in managing real-world applications. Additional research will concentrate on determining whether it is possible to integrate multiple platforms to address a specific application challenge, such as an effort to combine Hadoop (a horizontal scaling platform) with GPU (a vertical scaling platform). Investigating the use of machine learning in conjunction with contemporary big data platforms is also necessary. For several real-world issues in healthcare, IoT, smart cities, smart health applications, and large genomic datasets, combining different platforms can be a preferable approach.

References:

[1] https://link.springer.com/article/10.1007/s12599-013-0249-5

[2] https://dl.acm.org/doi/10.1145/1327452.1327492

[3]https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/en-us-events-2009summerschool-roger_barga_dryad.pdf

[4] https://link.springer.com/article/10.1007/s41060-016-0027-9

[5] https://ieeexplore.ieee.org/abstract/document/6949336

[6] https://ieeexplore.ieee.org/document/5446251

[7]https://ieeexplore.ieee.org/abstract/document/1303120?casa_token=BlnufK6E8WkAAAAA:kpZnkfsn4erbHlASRk3Rwtghbcrnmbd1OJnV-9p21HeToXoyNIkJnDuAYa4uObZMt-Q_F6ndNQ