Tong Wu, Zhaolin Fang, Leonardo Kuffo

Group 6: Assignment 1

Paper #1: The Pathologies of Big Data

In this article, the author discusses six relevant problems and two bottlenecks when dealing with big data from the late 1980s to modern times. Simultaneously, he gives some strategies either to avoid or solve these pathologies. In the end, the author offers a meta definition of big data obtained from the process of discussion, thereby answering the question he posed at the beginning— "What is big data".

The first problem is that in the late 1980s, ***storing big data was expensive and it required manual work by operators***. The latter restricted the kinds of questions that could be asked on big data datasets (e.g. The 1980 U.S. census database). In addition to this, big data made the storage progress very slow. Nevertheless, with the continuous development of computer disks, the author believed that he would be able to store data from the "World Census Dataset" on a cheap disk nowadays and deal with it on a desktop computer. In order to prove this, the author wrote an algorithm to calculate the median age by sex and country over the entire world population and tested it on a desktop PC. The results of the experiment arose the second typical problem when dealing with big data nowadays—***The algorithm was bounded since a disk reading rate is slow and therefore leads to an underutilization of the CPU***. Furthermore, when the author loaded the dataset into PostgreSQL and tried to make queries on the big dataset, he experienced an increase in the execution time when the number of rows went over 1 million. The latter was due to PostgreSQL's "sorting by group columns" query algorithm. In terms of time, it was a viable operation if given a million rows. However, it performed very poorly when facing a billion rows. ***Hence, we can see here the third difficulty of big data was in analyzing the stored data, rather than in the storage.***

To avoid such problems of big data, we need to think about "what makes big data *big*". By doing so, the last three problems of big data arise. The 4th problem is ***"repeated observations over time and/or space*** make big data big". These types of observations include weblogs, retailer logs and scientific measurements involved with 2 or 3 dimensions of space. ***The model of relational databases*** is the 5th problem since it explicitly ignores the order of rows in tables, hence data retrieval will be non-sequential once the dataset becomes overwhelming. It is important to say that sequential access is far more efficient and fast than non-sequential access. In order to avoid this problem, we may need to consider abandoning purely relational databases to obtain acceptable performance when we execute highly order-dependent queries on large data. The 6th problem is the ***existence of random access***. The huge cost of random access has major implications for the analysis of large datasets. To avoid it, we should consider changing the substantial media (i.e. tape) rather than only improving storage in degrees since even with SSD, the random access is still there and the difference between random access and sequential access is still enormous.

In addition to the six problems previously stated, there are two bottlenecks that generally surface when handling big data. These are "Hard Limits" and the problems encountered when we process big data using "Distributed Computing". For instance, applications such as Microsoft Excel have ***hard limits*** on the size of data like rows and columns they can handle. On these types of applications, an improvement on the size of the limits is not enough since the users are likely to face a bigger dataset sooner or later. Regarding Distributed Computing, two major problems arise: non-uniform ***distribution of work across nodes*** and ***reliability***. The non-uniform distribution of work refers to all the workload being concentrated in a single node instead of evenly balancing across all the nodes. In this case, we will get no benefit at all from parallelism. In addition to this, with more nodes, the probability of failure also becomes larger. Hence, negatively affecting the reliability of applications.

Finally, the author concludes with a meta definition of big data we obtained from the above discussion— "data whose size forces us to look beyond the tried-and-true methods that were prevalent at that time."

Reference: *Jacobs. (2009). The pathologies of big data. Communications of the ACM, 52(8), 36–44. https://doi.org/10.1145/1536616.1536632*

Paper #2: Cloud Computing and Grid Computing 360-Degree Compared

"Computation as a public utility" has been an idea since many decades ago. Grid Computing was the first tangible example of this concept until the recent emergence of Cloud Computing. Both terms and concepts are overlapped while not being entirely equal. Both share the same vision (i.e. reduce computing costs and increase computing reliability) and both must deal with similar problems. However, the scale of operation, underneath technologies and utility of each one of them is different. In this article, the authors compare Cloud Computing and Grid Computing from **six** different points of view. On each one of them, they highlight the differences, similarities, and challenges that both technologies must tackle.

From a **business model** point of view, Cloud Computing is based on "pay-on-demand". Similarly to electricity or water bills, users only pay for the computing time or storage amount they use. On the other hand, Grid Computing business models are project-oriented. This means that organizations must write a project proposal in order to use the computing power of a grid.

From an **architectural** viewpoint, interoperability and security are key on Grid computing. Hence, standardized protocols and services are defined to ensure the correct operability of the Grid. On the other hand, Cloud computing consists of a more "relaxed" architecture that can be built on top of web services protocols (e.g. SOAP), Web 2.0 technologies (e.g. REST) or even on top of Grid computing protocols. The latter enables Cloud computing to provide different levels of services: Infrastructure (e.g. computing power or storage services), Platform (e.g. environments to build or deploy applications) and Software as a service (i.e. special-purpose software).

In terms of **resource management,** the authors contrast at a very high granularity Grid and Cloud computing in topics such as: compute models, data models, virtualization, monitoring and provenance. For example, regarding *data models*, the authors argue that Client Computing will be as important as Cloud Computing due to security and network concerns. However, in both Cloud and Grid computing, data locality will be a problem since moving big data across the network is becoming the main bottleneck of computation. Hence, the orchestration of computing and data management is of high importance for both Cloud and Grid computing. In addition to this, *virtualization* comprises a key ingredient for Cloud computing. These abstractions of resources enable applications to be encapsulated and managed in an easier way. In contrast, Grid computing does not rely heavily on virtualization.

Regarding **programming models**, Grid environments usually use Message Passing Interface (MPI) implementations (e.g. MPICH-G2). In addition to this, there are systems specialized for workflows executions such as the Swift system. On the other hand Cloud environments use scripting languages (e.g. Python) and have adopted HTTP and SOAP as communication protocols between Web Services.

When talking about **application models**, Grid and Cloud computing are able to achieve success in many types of loosely coupled applications and High Throughput Computing. However, due to latency limitations, Cloud computing is having a hard time supporting High-Performance Computing (HPC) applications. Which is one of the applications in which Grid computing have achieved success.

Last but not least, regarding **security models**. Cloud computing security seems less complex and less secure due to unsafe credentials management and unsafe communication mechanisms. In addition to this, information and data centres are under the control of one external organization. In contrast, Grid computing implements GSI (Grid Security Infrastructure) for authentication, communication and authorization. Furthermore, credentials are only faxed or mailed to users.

The authors close the article by remarking how notable the differences between Cloud and Grid computing are. However, they emphasized that both technologies share the same vision. They wrap up by stating that it is important to develop and tackle problems of both technologies in parallel in order to develop *"what comes next"*.

Reference: *Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008, November). Cloud computing and grid computing 360-degree compared. In 2008 grid computing environments workshop (pp. 1-10). Ieee.*