Massive scale data analytics with Hadoop

PROJECT PLAN

FINAL YEAR PROJECT

RUBEN CURTIS

FULL UNIT PROJECT – CS 3821

ABSTRACT

Data is simply a piece of information, by itself it is of limited value. Data inherently needs other data to provide a basis for analysis. The analysis of data is known as data analytics, where operations are performed on data, transforming it to find a desired conclusion. Hadoop is a tool that was developed by the Apache Software Foundation that can perform these necessary operations to find conclusions in data, with Hadoop being a popular implementation of the MapReduce program used in problems involving massive amounts of data and computation required to process the data [8]. This is achieved through two different processes, map and reduce [1][6], commonly referred together as MapReduce due to how often map and reduce is executed consecutively. MapReduce has a primary goal of providing optimised data analysis through splitting a large computational workload between an arbitrary number of nodes in a network [7], known as a cluster of machines, followed by reducing the results to form a final optimised solution to the request given to the MapReduce program. Allowing for large quantities of data to be analysed.

A problem of large-scale data management is typically the quantity of data [2], so much so that Big Data is a term derived from this issue, with the amount of data in these systems in the petabytes [3]. Clearly, optimisations in data management and analysis are necessary to handle this volume of data in a timely manner. MapReduce and its implementation through Hadoop provide a possible solution to help solve large scale data analysis problems [6]. MapReduce allows resources to be as efficient as possible, which is an important goal of data management [4]. This makes it possible for large infrastructures to run optimally, reducing costs for hardware and computational power requirements, or performing more operations using the same computational power. While valuable on small scale clusters, its value increases as the size of the cluster increases. This scalability makes MapReduce a powerful tool for large scale data management and by extension data analytics. The demand for data management is increasing [5], with this demand the requirement for software to handle large amounts of data will also be in demand, providing Hadoop with its opportunity to solve this problem.

This project aims to use the Map and Reduce algorithm to find solutions to both simple and complex data analytics problems, along with the implementation of a data mining algorithm to analyse the performance of the MapReduce program using Hadoop. With the goal of quantifying the performance between each problem, eventually finding optimal parameters for each problem, tested with multiple data sets. A primary objective for this project is to document learning Map and Reduce, covering the practical implementation along with technical aspects of Map and Reduce itself. Another goal is to have the program be deployable on different clusters of machines, requiring the program to adapt to differing infrastructures, simulating a real-world scenario of having the same program deployed in different systems. Another goal includes having the program be as reliable as possible, this includes contingency plans in cases of both hardware and software failures, avoiding scenarios where data cannot be operated on or at worst deleted with no option for recovery. There may be cases where the optimal parameters for a problem will change depending on the cluster it is being executed on, to compensate for this, there will be a fixed cluster which has optimal parameters along with an overall best parameter setting only in cases that require it.

TIMELINE

This project will be split into two terms. The first term will be spent prioritising understanding Hadoop and the MapReduce algorithm, implementing simple programs and documenting its development. Problem 1 is to find the total occurrences of a word given a collection of web pages. Problem 2 includes the same data set of web pages, with the goal being to find all strings matching a given regular expression.

The second term will prioritise more advanced algorithms, including data mining. Ideally the first two problems should be implemented and documented in its entirety by the beginning of the second term. The third problem is to create a frequency distribution of a large collection of numbers, along with visualising this distribution appropriately. Finally, problem 4 is to implement a data mining program that will provide insight into a data set and visualise specific characteristics of the set.

Term 1

Week 1: Initial Hadoop and MapReduce research.

Week 2: Project Plan completed. Installing Hadoop, implementing basic proof-of-concept programs.

Week 3: More proof-of-concept programs, beginning the first problem.

Week 4: Continuing the first problem, at minimum proving it is possible.

Week 5: Finding optimal parameters and beginning the visualisation of the first problem.

Week 6: Completing the first problem, ensuring that documentation of the problem and the experience of solving it is completed.

Week 7: Ensuring the first problem is complete, research for the second problem, attempting an implementation if practically feasible.

Week 8: Second problem implementation, program should be somewhat functional but not optimal, keep documentation of the program as it is being developed.

Week 9: Complete the second problem, implement an optimised solution. Research and begin the report early deliverable.

Week 10: Focus entirely on the report, ensure that both problems are entirely complete beforehand. Create a first draft.

Week 11: Create a final draft, ensure that the programs work as intended, implement the changes from the first draft to the final draft.

Term 2

Week 1: Research for the third problem, reflect on feedback from the first term.

Week 2: Begin implementing the third problem, begin documenting its implementation.

Week 3: Continuing development of the third problem, ideally beginning optimisation.

Week 4: Completing an optimised solution to the third problem, compiling the documentation of its development, to be useful in the report later. Research for the final problem if possible.

Week 5: Research for the final problem, beginning proof-of-concept programs and prototyping small sections of the problem including documenting the process.

Week 6: Attempt a working solution to the problem, begin optimizing the program only if the solution is stable.

Week 7: Have a working program along with attempts at optimizing the program. Ideally program should be completed in its entirety. Begin research for the final report if possible.

Week 8: Begin the final report only after sufficient research, ensuring that all programs are completed in its entirety, making the report the only task remaining.

Week 9: Continue the final report.

Week 10: Ideally have a first draft of most if not all the report produced.

Week 11: Finish the remaining sections of the first draft, if necessary, along with creating a final draft of the report.

RISKS AND MITIGATIONS:

The risks present in this project are both unique to Hadoop itself along with risks that are generally present in projects that involve computer programming. The following risks will be described in both the likelihood of it occurring, along with the severity of implications that would arise from the risk happening as either minor or major.

Mismanagement of time

Likelihood – Major

Severity – Variable

A risk that is somewhat difficult to quantify, as there is not exactly a definitive way to say that time has been mismanaged, it is better to quantify it on a scale, using the project plan to see deviations from the expected timeline. This risk can also be seen as mismanagement between sections of the project, such as prioritising one section of the project disproportionately over another.

File deletion/corruption:

Likelihood – Minor

Severity – Major

Files such as code ran on Hadoop clusters are at a small risk of being deleted or corrupted. In the extremely rare case that it may happen, the consequences are catastrophic. To mitigate this, all files will be stored in a GitLab repository, providing a safe, remotely accessible backup to protect against this risk. This is subject to regular commits to the GitLab repository. Non program files will also be stored on the GitLab repository, separate from program files to avoid potential conflicts.

Hadoop Implementation and debugging:

Likelihood – Major

Severity – Minor

Hadoop is an advanced tool, with the potential for the program to break in a wide variety of ways. Debugging programs such as this can take a wildly variable amount of time, this can be linked to mismanagement of time, where an excessive amount of time debugging Hadoop programs can potentially compromise the rest of the project.

Changes in project scope:

Likelihood – Major

Severity – Variable

Like many projects, there is a high likelihood that the requirements of the project can change, in the case of this project it will be very unlikely that there will be major changes in scope, the changes in

scope will likely be from specific implementations of the Hadoop programs itself, such as having programs work for a larger cluster of machines than previously planned. The project plan will help mitigate changes, if necessary, as an already structured plan will help direct the project.

Production of low-quality software:

Likelihood – Minor

Severity – Variable

Low quality software can significantly impact this project, as the likelihood that programs will be modified throughout its development is much higher than other software development projects. Therefore, code produced must remain modifiable to keep development smooth, as code that is hard to modify will also take significant time to remedy, potentially leading to mismanagement of time, another significant risk to this project.

REFERENCES

[1]. Dean, J. and Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), pp.107-113.

"Users specify the computation in terms of a map and reduce function" this reference is used to distinguish that Map and Reduce are two different programs.

[2]. Schadt, E.E., Linderman, M.D., Sorenson, J., Lee, L. and Nolan, G.P., 2010. Computational solutions to large-scale data management and analysis. *Nature reviews genetics*, *11*(9), pp.647-657.

"Supercomputing resources will be increasingly needed to get the most from the big data sets that researchers generate or analyse" This highlights the increasing demand for computing resources, implying that optimising the current computing resources will also satisfy the increasing demand.

[3]. Elgendy, N. and Elragal, A., 2014. Big data analytics: a literature review paper. In *Advances in Data Mining. Applications and Theoretical Aspects: 14th Industrial Conference, ICDM 2014, St. Petersburg, Russia, July 16-20, 2014. Proceedings 14* (pp. 214-227). Springer International Publishing.

"Big data sizes are constantly increasing, currently ranging from a few dozen terabytes (TB) to many petabytes (PB) of data in a single data set" This gives a quantifiable number to the size of data that will be getting processed, helping to understand the scale involved in data analytics.

[4]. Gustafsson, T. and Hansson, J., 2004, May. Data management in real-time systems: a case of on-demand updates in vehicle control systems. In *Proceedings. RTAS 2004. 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004.* (pp. 182-191). IEEE.

"At the same time it is important that the resources are utilized as efficient as possible, e.g., for CPU resources unnecessary calculations should be lowered as much as possible." This supports the idea that optimisation is an important tool in data management.

[5]. Qiu, J., Wu, Q., Ding, G., Xu, Y. and Feng, S., 2016. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, *2016*, pp.1-16.

“There is no doubt that big data are now rapidly expanding in all science and engineering domains” This supports the idea that there is a higher demand for data management and by extension data analytics.

[6]. Rajaraman, A. and Ullman, J.D., 2011. *Mining of massive datasets*. Cambridge University Press.

“The map-reduce framework, an important tool for parallelizing algorithms automatically.” Supports the idea that MapReduce is both a useful tool and a tool for optimizing data analytics.

[7]. Kavulya, S., Tan, J., Gandhi, R. and Narasimhan, P., 2010, May. An analysis of traces from a production mapreduce cluster. In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* (pp. 94-103). IEEE.

“MapReduce is a programming paradigm for parallel processing that is increasingly being used for data-intensive applications in cloud computing environments” Supports the idea that MapReduce is used on clusters of machines for parallel processing.

[8]. Hashem, I.A.T., Anuar, N.B., Gani, A., Yaqoob, I., Xia, F. and Khan, S.U., 2016. MapReduce: Review and open challenges. *Scientometrics*, *109*, pp.389-422.

“MapReduce is a popular tool for the distributed and scalable processing of big data. It is increasingly being used in different applications primarily because of its important features, including scalability, fault tolerance, ease of programming, and flexibility.” Supports the goals of MapReduce, along with what MapReduce does.