SHIV SHAKTI
International Journal in Multidisciplinary and Academic Research (SSIJMAR)
Vol. 7, No. 2, April 2018 (ISSN 2278 – 5973)

MultiSite Aware Big Data Management for Scientific Workflows on Clouds

Manikantha Desu, MTech
Assistant Professor
Department of CSE
Hitam,Jntuh
Hyderabad,India
manikanthadesu@gmail.com

Manasa Bindu
Department of CSE
Hitam,Jntuh
Hyderabad, India
manasabindu6@gmail.com

Kallam Harika
Department of CSE
Hitam,Jntuh
Hyderabad, India
kallamharikareddy@gmail.com

Ch.sruthi
Department of CSE
Hitam,Jntuh
Hyderabad, India
sruthi9chundru@gmail.com

ABSTRACT

The global deployment of cloud datacentres is enabling large scale scientific workflows to improve performance and deliver fast responses. In this paper, we introduce Overflow, a uniform data management system for scientific workflows running across geographically distributed sites, aiming to reap economic benefits from this geo-diversity. Our solution is environment-aware, as it monitors and models the global cloud infrastructure, offering high and predictable data handling performance for transfer cost and time, within and across sites. Overflow proposes a set of pluggable services, grouped in a data scientist cloud kit. They provide the applications with the possibility to monitor the underlying infrastructure, to exploit smart data compression, deduplication and geo-replication, to evaluate data management costs, to set a trade-off between money and time, and optimize the transfer strategy accordingly. The system was validated on the Microsoft Azure cloud across its 6 EU and US datacentres. The experiments were conducted on hundreds of nodes using synthetic benchmarks and real-life bio-informatics applications (A-Brain, BLAST). The results show that our system is able to model accurately the cloud performance and to leverage this for efficient data dissemination, being able to reduce the monetary costs and transfer time by up to three times.

INDEX TERMS -Big Data, scientific workflows, cloud computing, geographically distributed data management.

I. INTRODUCTION

With their globally distributed datacentres, cloud infrastructures enable the rapid development of large scale applications. Examples of such applications running as cloud services across sites range from office collaborative tools (Microsoft Office 365, Google Drive), search engines (Bing, Google), global stock market analysis tools to entertainment services (e.g., sport events broadcasting, massively parallel games, news mining) and scientific workflows [1]. Most of these applications are deployed on multiple sites to leverage proximity to users through content delivery networks. Besides serving the local client requests, these services need to maintain a global coherence for mining queries, maintenance or monitoring operations, that require large data movements. To enable this Big Data processing, cloud providers have set up multiple datacentres at different geographical locations. This geographical distribution of computation becomes increasingly important for scientific discovery. In fact, many Big Data scientific workloads enable nowadays the partitioning of their input data. This allows to perform most of the processing independently on the data partitions across different sites and then to aggregate the results in a final phase. In some of the largest scenarios, the data sets are already partitioned for storage across multiple sites, which simplifies the task of preparing and launching a geographical-distributed processing.

A. The Need of Site-Aware File Management for Cloud Based Workflows

As we move to the world of Big Data, single-site processing becomes insufficient: large scale scientific workflows can no longer be accommodated within a single datacenter. Moreover, these workflows typically collect data from several sources and even the data acquired from a single source is distributed on multiple sites for availability and reliability reasons. Cloud let users to control remote resources. In conjunction with a reliable networking environment, we can now use geographically distributed resources: dynamically provisioned distributed domains are built in this way over multiple sites. Several advantages arise from computations running on such multi-site configurations: resilience to failures, distribution across partitions (e.g., moving computation close to data or vice versa), elastic scaling to support usage bursts, etc. However, in order to exploit the advantages of multi-site executions, users currently have to set up their own tools to move data between deployments, through direct endpoint to endpoint communication (e.g. GridFTP, scp, etc.). This baseline option is relatively simple to set in place, using the public endpoint provided for each deployment. The major drawback in this scenario is the low bandwidth between sites, which limits drastically the throughput that can be achieved. Clearly, this paradigm shift towards multi-site workflow deployments calls for appropriate data management tools, that build on a consistent, global view of the entire distributed data centre environment. This is precisely the goal targeted by Over Flow.

II.BIG DATA BATCH PROCESSING BY MAPREDUCE

Map Reduce is one of the most wide-spread type of computation performed nowadays on the cloud, making it an interesting candidate to evaluate the impact of Overflow. We propose the following experiment: we take an actual Map Reduce engine (i.e., the Hadoop on Azure service, denoted HOA) and a general-purpose workflow engine (i.e., the open-source Microsoft Generic Worker [12], denoted GW) and run a typical WordCount MapReduce benchmark. We used an input of 650 MB data, processed in two setups: with 20 mappers (32 MB per job) and 40 mappers (16 MB per job); the number of reducers is 3 in both cases. The results, presented in Fig. 10, show, as expected, that the specialized Map Reduce tool is more efficient. The question is: can Overflow improve the performance of a workflow engine compared to a Map Reduce engine? To answer, we replaced the Azure Blob based storage of the Generic Worker with our approach (denoted GW++) and run the benchmark. We notice that GW++ achieves a 25 percent speedup compared to Hadoop, building on its protocol switching approach which enables to use the best local transfer option according to the data access pattern.

A. Classification of MapReduce Operation Phases

A MapReduce job typically involves 6 phases, with two of them being optional:

- *Initial data distribution:* In this phase, the data to be processed is distributed among processing nodes so as to benefit from data-level parallelism inherently available in the MapReduce processing paradigm.
- *Map function:* The Map function is executed on each data block stored in each processing node. This is one of the two major data-parallel tasks involved in MapReduce paradigm.
- *Data combine:* In this optional phase, (key, value) pairs produced on each processing node during previous phase are combined (using the same Reduce function as below) on the same machine so that each node has only one pair per key.
- *Data shuffle*: the (key, value) pairs with the same key, but residing on different processing nodes, need to be moved to a single node so as to apply the Reduce function on them. This is done in the data shuffle phase.
- *Reduce function:* all pairs with the same key are now on a single processing node. The Reduce function is now applied to each set so as to obtain the final set of (key, value) pairs with unique keys. Depending on the big data job in hand, the reduce phase may not be needed and can be left empty.

III.ARCHITECTURE OF MAPREDUCE

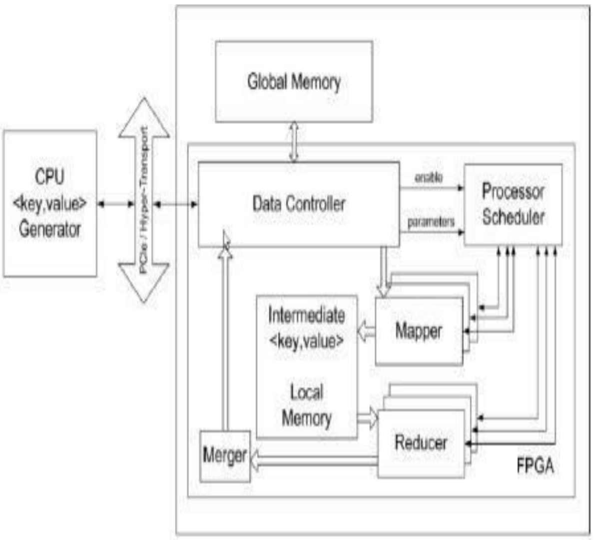

Figure 1: Architecture of MapReduce

IV.Experimental Setup

The experiments are performed on the Microsoft Azure cloud, at the PaaS level, using the EU (North, West) and US (North-Central, West, South and East) datacenters. The system is run on Small (1 CPU, 1.75 GB Memory and 100 Mbps) and Medium (2 CPU, 3.5 GB Memory and 200 Mbps) VM instances, deploying tens of VMs per site, reaching a total number of 120 nodes and 220 cores in the global system. The experimental methodology considers three metrics: execution time, I/O throughput and cost. The execution time is measured using the timer functions provided by the .Net Diagnostic library. The throughput is determined at destination as a ratio between the amount of data sent and the operation time. Finally, the cost is computed based on Azure prices and policies [19]. Each value reported in the charts is the average of tens of measurements performed at different daily moments.

V.CONCLUSION

This paper introduces Overflow of a data management system for scientific workflows running in large, geographically distributed and highly dynamic environments. Our system is able to effectively use the high-speed networks connecting the cloud datacentres through optimized protocol tuning and bottleneck avoidance, while remaining non-intrusive and easy to deploy. Currently, Overflow is used in production on the Azure Cloud, as a data management backend for the Microsoft Generic Worker workflow engine. Encouraged by these results, we plan to further investigate the impact of the metadata access on the overall workflow execution.

ACKNOWLEDGEMENT

We would like to express gratitude to our guide Mr.ManikanthaDesu for supporting us in our project.

REFERENCES

[1] http://www.windowsazure.com/en-us/case-studies/archive/

[2] https://developers.google.com/even
ts/io/sessions/333315382

[3] http://www.msr-inria.fr/projects/z-cloudflow-data-workflows-in-the-cloud

[4] http://www.codeproject.com/
Articles/380769

[5] https://www.hadooponazure.com/