Scaling Smart Appliances for Spatial Data Synthesis

Luis Pineda-Morales
Microsoft Research - Inria Joint Centre
Rennes, France
luis.pineda-morales@inria.fr

Balaji Subramaniam
Argonne National Laboratory
Lemont, IL, USA
balajis@mcs.anl.gov

Kate Keahey
Argonne National Laboratory
Lemont, IL, USA
keahey@mcs.anl.gov

Gabriel Antoniu
Inria Rennes - Bretagne Atlantique
Rennes, France
gabriel.antoniu@inria.fr

Alexandru Costan
IRISA / INSA Rennes
Rennes, France
alexandru.costan@irisa.fr

1. INTRODUCTION

Several scientific domains rely on the ability to synthesize spatial data, embedded with geographic references. Economists and sociologists, for instance, use spatial data to analyze and describe population dynamics [2]. As the sources of spatial data, such as sensors and social media, have become more accurate and numerous, the generated data has considerably grown in size and complexity over the past years. As a consequence, larger computing capabilities are required for storing, processing and visualizing the data. Recently, clouds have emerged as convenient infrastructures for supporting current spatial data synthesis needs, since they offer dynamically provisioned and fairly inexpensive resources. Elastic provisioning aims at optimizing resource usage in clouds by adding or removing appliances on the fly. In particular, we look into two scenarios in spatial data synthesis where elastic capabilities can improve application performance:

1. A service which has several different components within it. As a result, the amount of resources required, such as compute, network and storage, fluctuates depending upon the specific component being used.
2. These services support multiple scientific communities. Several users aggregate and integrate data from different sources. Therefore, managing concurrent user requests becomes a challenge.

1.1 Application Use Case

In collaboration with the CIGI - UIUC[1], we are providing and testing cloud resources for enabling spatial data synthesis; specifically we support an application addressing spatial index for estimating home and work relocation, and unemployment rates using geo-located twitter data.

The application in its current state consists of three stages: filtering, classifying and clustering. The *filtering* stage removes duplicate or corrupted entries from the raw geo-located Twitter data, the *classifying* stage looks for the data per unique user, and the *clustering* stage identifies the top visited locations by each user. The first stage is implemented in Pig and the last two in Hadoop.

[1]CyberInfrastructure and Geospatial Information (CIGI) Laboratory at the University of Illinois at Urbana-Champaign

1.2 Contributions

In this poster, we make the following contributions towards enabling elastic cloud appliances for spatial data synthesis:

1. We carry out a proof-of-concept evaluation of the application described in Section 1.1.
2. We provide an analysis of the performance of the different stages of the application.
3. We present details into tuning the Hadoop configuration parameters to extract the best performance from the application.

2. EVALUATION

In this section, we describe our experimental platform and inferences gained from tuning the Hadoop configuration parameters.

Experimental Platform: We used two bare-metal nodes in the Chameleon cloud. Each node has 24 physical cores (48-threads, when hyperthreading is enabled) and 128 GB memory, connected by a 10Gbits/sec network. We ran Pig scripts and MapReduce jobs on Hadoop 2.7 (YARN), using the default HDFS blocksize of 128 MB as the baseline.

In the rest of this section we will describe our experience on tuning the Hadoop configuration parameters for different stages of the application.

Data loading: We measured transfer time for different data block and dataset sizes. We notice that with larger

blocks (2GB) data is transmitted faster. However, larger blocks might incur slower execution times.

Execution time: We observe that there is a "sweet spot" where the tradeoff between block size and block count yields the best performance. This spot is the largest block size so that the number processed blocks is smaller than the number of processing units (containers).

Data replication: By varying HDFS replication factor between 1 and 2, we obtained two observations: 1) for small datasets, using a factor of 2 does not significantly outperform factor of 1, instead, it adds data loading overhead. 2) For large datasets, disk saturation due to replication can render the execution nodes ill and interrupt the computation.

Parallelism: We varied the number of parallel reducers from 24 to 120. We did not notice a significant difference in execution time. However, we did notice that each container is under-utilized (only ∼25% of memory is used).

3. CONCLUSIONS AND FUTURE WORK

We have identified pre-configurable parameters that impact the execution time of an application in the cloud. We intend to learn from our observations with such parameters to derive performance models for elastic provisioning.

The following remain in our agenda for future work:

- Scale out experiments to tens of nodes and months of tweets
- Perform tests in other infrastuctures, including virtualization environments
- Consistent support for elastic execution of the use case application's workflow as new components are released. This will likely involve coupling Chameleon with Phantom elastic provisioner [1].

4. ACKNOWLEDGMENTS

Luis Pineda-Morales' internship at Argonne National Laboratory was funded by the Data@Exascale Inria-ANL Associate Team and by the NextGN Inria-ANL PUF project. This material is based in part upon work supported by the U.S. National Science Foundation under grant numbers: 1047916, 1429699, and 1443080. Insightful comments and feedbacks received from the following members of the CyberGIS Center: Junjun Yin and Kiumars Soltani are greatly appreciated.

5. ADDITIONAL AUTHORS

Shaowen Wang[abc], Anand Padmanabhan[abc], Aiman Soliman[ab]
[a] CyberGIS Center for Advanced Digital and Spatial Studies
[b] National Center for Supercomputing Applications
[c] Department of Geography and Geographic Information Science
University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States

6. REFERENCES

[1] K. Keahey, P. Armstrong, J. Bresnahan, D. LaBissoniere, and P. Riteau. Infrastructure outsourcing in multi-cloud environment. In *Proceedings of the 2012 Workshop on Cloud Services, Federation, and the 8th Open Cirrus Summit*, FederatedClouds '12, pages 33–38, New York, NY, USA, 2012. ACM.

[2] A. Llorente, M. Cebrian, E. Moro, et al. Social media fingerprints of unemployment. *arXiv preprint arXiv:1411.3140*, 2014.