Smart Connector and Scheduler on the Cloud

Iman Yusuf, Ian Thomas, Heinz Schmidt
RMIT University, Melbourne, Australia
{Iman.Yusuf, Ian.Edward.Thomas, Heinz.Schmidt}@rmit.edu.au

INTRODUCTION

Running experiments, collecting experiment results, and data visualisation are routine tasks of many researchers. Depending on how a researcher carries out a task, they may need to execute multlple activities to complete the task. For instance, organising experiment results is no more than collecting data in a directory for one, but requires navigating multiple screens for the other who uses multiple applications, or for when the research workflow contains data curation systems like MyTardis [1]. Processing multiple screens of decisions could be tolerable if the researcher curates data only a few times. However, such activity becomes quickly tedious especially for a researcher whose experiments need to run multiple times for the purpose of, among other things, statistical confidence. Therefore, the objective of our project is, first and for most, to take these tedious activities out of the researcher's routine. Furthermore, we address the increasing use of infrastructure-as-a-service (IaaS) cloud infrastructure and the need to fit such services into the users workflow with the minimum of extra work. Overall, in this project, we aim to achieve two goals: (a) automating repetitive activities via *smart connector*, and (b) facilitating the scheduling of tasks on the IaaS systems based on researchers' needs via *smart cloud scheduler*.

SMART CONNECTOR

A *Smart Connector* is a dropbox-like component that executes a sequence of activities without external intervention. Smart connectors are designed to automate the activities that are routinely executed by a specific research group. Researchers in the School of Applied Science in RMIT University, for instance, curate data using HPC based version of MyTardis. A smart connector for this research group removes the need to navigate a dozen screens whenever data is curated. The connector executes activities on the researchers' behalf behind the scenes. Despite this connector being designed for RMIT University researchers, we expect the connector to be functional for any researcher that uses MyTardis for curating CRYSTAL [2], VASP [3], SIESTA [4] or GULP [5] data, and utilises standard and common workflows. The only requirement for using the smart connector is the placement of the data in a designated location.

SMART CLOUD SCHEDULER

A *Smart Cloud Scheduler* is a meta-scheduler that takes the concerns of researchers into account before submitting a task to a particular cloud. The smart scheduler recognises the concerns of researchers as security, cost, and execution time.

a. **Security**: Based on security requirements, tasks are scheduled on private, public or hybrid clouds. If a highly secured computation is required, for instance, then the computation is exclusively scheduled to run on a private cloud.

b. **Cost**: Another concern for researchers who use Cloud-based services to complete a task is monetary cost. Such concern becomes more important as the service involves more and more compute and/or data intensive computations. In order to address such concern, the smart scheduler thrives to ensure that a task is submitted to the Cloud provider that is most cost effective. It is important to note that the smart scheduler makes a price comparison only among Cloud providers that are available for the researchers. In this project, the comparison is made among Amazon EC2 [6], Rackspace [7] and Nectar Openstack [8].

c. **Execution Time**: The execution time of certain types of long running applications can be reduced by allocating more resources. The degree of reduction in the execution time, however, depends on the type of the application. Embarrassingly parallel applications, for example, are expected to benefit the most by the availability of abundant computing resources. The activities of an embarrassingly parallel application are independent of each other, and thus the execution of such application involves little or no communication. As the result, each activity can be scheduled

without being concerned about the location of other activities. Assuming maximal parallelism, the execution time of an embarrassingly parallel application is as small as the execution time of its longest activity. The Monte Carlo method [9], which is often used for simulating physical systems using repeated random sampling, is an example of an embarrassingly parallel application. In order for the smart scheduler to request the 'right' number of resources, the type of the application and optionally the number of activities of the application should be provided.

Researchers are at liberty to prioritise their concerns. Researchers who analyse sensitive data may put security ahead of both cost and execution time. These researchers are likely willing to pay more, and to wait longer for their task to be completed as long as the task is executed in a highly secured environment. On the other hand, researchers who analyse massive non-sensitive data may put execution time ahead of cost and/or privacy. These researchers may need to pay extra so as to access HPC facilities that enable significant reduction in execution time.

CONCLUSION

One of the goals of this project is to automate the repetitive activities of researchers in the School of Applied Science in RMIT University via discipline specific smart connectors. Despite the focus on one group, we would like to continue our work and provide other connectors for other research groups. We hope that the eResearch community gets excited about such initiative and start building smart connectors. The ultimate goal is that a smart connector that is built for specific researchers in a particular university or research centre can be used by any researcher who wishes to perform a similar task. The other goal of our project is to provide customised scheduling via the smart cloud scheduler. The baseline implementation of the smart scheduler uses course-grained policy to address the concerns of the researchers before scheduling. By fine-tuning the policy, we expect the performance of the scheduler to be improved.

REFERENCES

1. MyTardis - Androulakis S, Schmidberger J, Bate MA, Degori R, Beitz A, Keong C, Cameron B, McGowan S, Porter CJ, Harrison A, Hunter J, Martin JL, Kobe B, Dobson RC, Parker MW, Whisstock JC, Gray J, Treloar A, Groenewegen D, Dickson N, Buckle AM. (2008) Federated repositories of X-ray diffraction images. Acta Crystallogr D Biol Crystallogr. Jul;64(Pt 7):810-4
2. CRYSTAL. http://www.crystal.unito.it/
3. Martijn Marsman. *VASP ab-initio package: Vienna Ab-inition Simulation Package*. 2009.
4. E. Artacho et al. The Siesta method; developments and applicability. *J. Phys.: Condens*. Matter **20**, 064208 (2008).
5. Julian D. Gale, GULP - a computer program for the symmetry adapted simulation of solids, *J. Chem. Soc. Faraday Trans.* **93** (1997) 629–637.
6. Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/
7. Rackspace. http://www.rackspace.com.au/
8. Nectar Openstack. http://nectar.org.au/
9. Metropolis, N.; Ulam, S. (1949). The Monte Carlo Method. *Journal of the American Statistical Association* (American Statistical Association) **44** (247): 335–341.

ABOUT THE AUTHORS

Iman Yusuf is a software developer at the eResearch Office of RMIT University, with responsibly for developing cloud-based platforms in support of eResearch applications. She is also a PhD candidate in the School of Computer Science and Information Technology at RMIT University. Her research interests are grid and cloud computing, fault tolerance, reliability, and component-based software architecture.

Ian Thomas is a software developer and system administrator at the eResearch Office of RMIT University. He has worked in data curation for output of high-performance computing systems, microscopy data for materials, and screen media objects (film and television). His current work is in institutional metadata stores, decision support systems for climate change modelling, and in cloud-based platforms in support of eResearch applications.

Heinz is Professor of Software Engineering at RMIT University where he is the Director of eResearch and heads the Distributed Software Engineering and Architecture lab in Computer Science. Heinz is also an adjunct professor at Mälardalen University in Sweden. Heinz received his PhD from Bremen University, Germany. He has over 30 years experience with component-based and object-oriented architecture, especially

in parallel and distributed systems and languages in practice, research and education. Heinz is an eminent researcher who has published over 120 refereed articles, supervised over 25 higher-degree research students, and lectures in software engineering, distributed systems and enterprise architecture. Prior to RMIT Heinz held positions at Monash University, the CSIRO and ANU in Canberra, at the German National Research Centre for Information Technology and the International Computer Science Institute at the University California, Berkeley. Prof Schmidt has led large university-industry research collaborations, in the European ESPRIT program and the Australian Collaborative Research Center program, among others with SIEMENS, ABB, DEC and Olivetti, IBM and others.