Big Data Research 8 (2017) 39–49

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

Chiminey: Connecting Scientists to HPC, Cloud and Big Data ☆

Iman I. Yusuf [a], Ian E. Thomas [a,*], Maria Spichkova [b], Heinz W. Schmidt [b]

[a] RMIT University, eResearch Office, 17-23 Lygon Street, 3053, Carlton, Australia
[b] RMIT University, School of Science, 124 La Trobe Street, 3000, Melbourne, Australia

ARTICLE INFO

Article history:
Received 18 April 2016
Received in revised form 15 November 2016
Accepted 20 January 2017
Available online 7 February 2017

Keywords:
Big data
Cloud
eScience
High performance computing
Parallel processing
Simulation

ABSTRACT

The enabling of scientific experiments increasingly includes data, software, computational and simulation elements, often embarrassingly parallel, long running and data-intensive. Frequently, such experiments are run in a cloud environment or on high-end clusters and supercomputers. Many disciplines in sciences and engineering (and outside computer science) find the requisite computational skills attractive on the one hand but distracting from their science domain on the other. We developed Chiminey under directions by quantum physicists and molecular biologists, to ease the steep learning curve in data management and software platforms, required for the complex computational target systems. Chiminey is a *smart connector* mediating running specialist algorithms developed for workstations with moderately large data set and relatively small computational grunt. This connector allows the domain scientists to choose the target platform and then manages it automatically; it accepts all the necessary parameters to run many instances of their program regardless of whether this runs on a peak supercomputer, a commercial cloud like Amazon EC2 or (in Australia) the national federated university cloud system NeCTAR. Chiminey negotiates with target system schedulers, dashboards and data bases and provides an easy-to-use dashboard interface to the running jobs, regardless of the specific target platform. The smart connector encapsulates and virtualises a number of further aspects that the domain scientists directing our effort found necessary or desirable.

In this article we present Chiminey and guide the reader through a hands-on tutorial of this open-source platform. The only requirement is that the reader has access to one of the supported clouds or cluster platforms – and very likely there is a matching one. The tutorial stages range in difficulty from requiring no to little technical background through to advanced sections, such as programming your own domain-specific extension on top of Chiminey application programmer interfaces.

The different exercises we demonstrate include: installing the Docker deployment environment and Chiminey system; registering resources for file stores, Hadoop MapReduce and cloud virtual machines; activating *hrmclite* and *wordcount* smart connectors – two demonstrators; running a smart connector and investigating the resulting output files; and building a new smart connector. We also discuss briefly where to find more detailed information on, and what is involved in, contributing to the Chiminey open source code base.

1. Introduction

In this article, we present the *Chiminey* platform, which provides a reliable computing and data management service. Chiminey enables domain scientists, hereafter scientists, to compute on both cloud-based, big data, and high-performance computing (HPC) facilities, handle failure during the execution of applications, curate and visualise execution outputs, share such data with collaborators or the public, and to search for publicly available data without the need to have a technical understanding of cloud-computing, HPC, fault tolerance, or data management. Many scientific experiments have a twofold challenge: they are challenging as complicated domain-specific research tasks (i.e., a complicated analysis of the quantum physics approaches), and at the same time the corresponding computations and datasets are too large-scale to be executed on a local desktop machine, i.e., cloud-based and high-performance computing (HPC) solutions are required.

Any new technology usually means not only new opportunities but also new challenges, as a technology often manages some initial knowledge acquisition task of its users. Cloud computing [1]

☆ This article belongs to HPC Tutorial for Big Data.

* Corresponding author.
E-mail addresses: iman.yusuf@rmit.edu.au (I.I. Yusuf), ian.edward.thomas@rmit.edu.au (I.E. Thomas), maria.spichkova@rmit.edu.au (M. Spichkova), heinz.schmidt@rmit.edu.au (H.W. Schmidt).

http://dx.doi.org/10.1016/j.bdr.2017.01.004
2214-5796/

enables acquisition of very large computing and storage resources, which can be integrated with big data technologies for massive scale computation. Such acquisition can be done with relatively less specialised knowledge beyond that used for a single PC.

Nevertheless, failure while setting up a cloud-based execution environment or during the execution itself is arguably inevitable: some or all of the requested virtual machines (VMs) may not be successfully created/instantiated, or the communication with an existing VM may fail due to long-distance network failure – given cloud data centers are typically remote and communication crosses many network boundaries. Also, one has to realise that all tasks of such parallel computations are required to complete, therefore the failure of any one of them may corrupt the result in some way. Statistically this means that the reliability of the overall task completion is the product of that of the individual tasks – and with very many thousands or millions of compute tasks this may quickly become a vanishingly small number.

When using cloud computing platforms and big data technologies like Hadoop, scientists require operational skills and to some extent knowledge of aspects of fault tolerance. Scientists need to learn, for example, how to create and set up virtual machines (VMs), collect the results of experiments, and finally destroy VMs. Such challenges distract the user from focusing on their core goals. Thus, there is a need for a platform that encapsulates these problems and isolates them from the user. This would enable the user to focus on domain-specific problems, and to delegate the tool to deal with the detail that comes with accessing high-performance and cloud computing infrastructure, and the data management challenges posed. For these reasons, we propose a user-friendly open-source platform that would hide the above problems from the user by encapsulating them in the platform's functionality.

The proposed open-source platform has been applied across two research disciplines, physics (material characterisation) and structural biology (understanding materials at the atomic scale), to assess its usability and practicality. The domain experts noted the following advantages of the platform: time savings for computing and data management, user-friendly interface for the computation set up, and visualisation of the calculation results as 2D or 3D graphs.

Previous work: The first prototype of Chiminey was discussed in [2]. A formal model of a platform for scalable and fault-tolerant cloud computations as well as the implementation of this platform as the *Chiminey* platform was introduced in [3]. The model allows us to have a precise and concise specification of the platform on the logical level. We also presented the refined formal model of a cloud-based platform and the latest version of its open-source implementation [4], with the emphasis on usability and reliability aspects. The feasibility of the *Chiminey* platform is shown using case studies from the Theoretical Chemical and Quantum Physics group at RMIT university.

Outline: The rest of the article is organised as follows. Section 2 provides background information, links or contrasts our work with related work. Section 3 introduces one of the core artifacts of Chiminey, Smart Connectors, as well as the resources the platform provides. Section 4 presents the tutorial, targeting the different types of Chiminey users. Section 5 concludes the article and presents the core directions of our future work on *Chiminey*.

2. Background

In 2009, Leavitt in his widely cited[1] paper [5] analysed advantages and challenges related to cloud computing, highlighting that this type of deployment architecture becomes appealing to many companies. Now, almost 8 years later, we can see that this paradigm becomes even more and more appealing. Another widely cited[2] paper on the cloud computing paradigm [6] presents a survey done by Zhang et al. The survey highlights the key concepts of cloud computing, its architectural principles, state-of-the-art implementation as well as research challenges.

Cloud computing provides many benefits, e.g., provisioning of virtual machines (VMs) within literally 15 minutes, when purchases of physical servers took days or weeks; access to online storage and computing resources at a moment's notice; cost savings by turning virtual servers and hence charges for them on and off at will; and not least, improved resource utilisation, across large numbers of users in one or more data centres.

However, failure in cloud services is arguably inevitable due to configuration errors, continuous upgrades somewhere in the cloud software stack or application layers, the unreliability of networks that remote services depend on, and thus generally the heterogeneous character of widely distributed systems. Yusuf and Schmidt [7] have shown in formal reliability and performance studies, that fault-tolerance is best achieved by reflecting the static and dynamic (behavioural) architecture of high-performance computational programs. Compared to architecture-agnostic replication, architecture-aware fault-tolerance can achieve higher reliability at lower costs, but needs to be tuned to different architectural/behavioural patterns such as stream processing, map-reduce, randomised access etc.

The development of formal models and architectures for systems involved in cloud computing is a more recent area of system engineering. Vaquero et al. [8] studied more than 20 definitions of the term *cloud computing* to extract a consensus definition as well as a minimum definition containing the essential characteristics. As a result, they consolidated the following definition:

> *Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized Service-Level Agreements.*

Buyya and Sulistio [9] presented a discrete-event grid simulation toolkit, GridSim, that can be used for investigating the design of utility-oriented computing systems such as Data Centers and Grids.

Ostermann et al. in their paper [10] stated a research question on whether the performance of clouds is sufficient for scientific computing. They analyzed the performance of the Amazon EC2 platform using micro-benchmarks and kernels, and came to the conclusion that the performance and the reliability of the tested cloud are low, and probably insufficient for scientific computing at large.

As the cloud-based systems deal with safety and security critical data, the formal modelling and verification of cloud architectures becomes more and more important. Su et al. used the CSP framework to model MapReduce system, cf. [11]. Reddy et al. [12] proposed an approach to verify the correctness of Hadoop systems (open source implementation of MapReduce) using model checking techniques. Our previous work on a formal model of the Chiminey system was presented in [3,4].

Several approaches have proposed or compared different map-reduce approaches for cloud computing, others data stream processing systems, and yet others parametric parallel solvers using

[1] More than 500 citations.

[2] More than 600 citations.