Probabilistic Scheduling of Scientific Workflows in Dynamic Cloud Environments

Amelie Chi Zhou, Bingsheng He and Cheng Liu Nanyang Technological University

There have been some proposals to reduce the performance interference and unpredictability in the cloud, such as net- work performance [42] and I/O performance [43], [44], [45]. However, by the design of cloud computing, cloud is shared by many concurrent executions. Therefore, the performance dynamics caused by the resource interference is unavoidable. Some other works have devoted efforts on designing tools for assessing cloud performance. Lenk et al. [46] designed a new performance measuring method that can measure the actual performance of the virtual machines running a specific IaaS service. CloudSleuth [47], CloudHarmony [48], Cloud- stone [49] and CloudCMP [50] are four common tools and services that can help customers measure the performance of cloud services. For example, CloudCMP provides comparisons on the performance of computation, storage and network services offered by different cloud providers. Chen et al. [51] have developed WorkflowSim with consideration on system overhead and failures.

Parallel and Distributed Systems, IEEE Transactions on (Volume:PP , Issue: 99 )

Meeting Deadlines of Scientific Workflows in Public Clouds with Tasks Replication

Rodrigo N. Calheiros, Member, IEEE and Rajkumar Buyya, Senior Member, IEEE

We assume that the workflow application executes in a single Cloud data center. Since more predictable execution and data transfer times are paramount for meeting application deadlines, keeping the workflow in a single data center eliminates one possible source of execution delay. It also eliminates the cost incurred by data transfer among data centers. We also ignore overheads incurred by the workflow management system. This is because they are strongly dependent on the particular technology for workflow management in use, varying from constant time [22] (which could be modeled as additional execution time of each task) to cyclical regarding the number of tasks managed [23].

Proceedings of the 2nd ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies

DynamicCloudSim: Simulating Heterogeneity in Computational Clouds

Marc Bux, Ulf Leser, Humboldt-Universität zu Berlin

Chen and Deelman [35] recently presented WorkflowSim as another extension to CloudSim. WorkflowSim is tightly bound to the SWfMS Pegasus [20] and adds to CloudSim (1) the workflow engine underlying Pegasus and DagMan [41], (2) an elaborate model of node failures, (3) a model of delays occurring in the various levels of the Pegasus stack (e.g., queue delays, pre/post-processing delays, data transfer delays), and (4) the implementations of several workflow schedulers implemented in Pegasus (e.g., greedy task queue, HEFT [23], Min-Min, and Max-Min [24]). Parameters are directly learned from traces of real executions. WorkflowSim follows a quite different approach than DynamicCloudSim: WorkflowSim models delays in the Pegasus workflow stack and is thus tightly coupled to Pegasus. It has no notion of heterogeneous hardware or variance in available resources. In contrast, DynamicCloudSim directly models instability and heterogeneity in the environment in which a workflow (or any other collection of computationally intensive tasks) is executed and is thus independent of a concrete system. See Table 2 for a comparison of features available in CloudSim, WorkflowSim, and DynamicCloudSim.

Simulation Modelling Practice and Theory

Energy-aware simulation with DVFS

Tom Guérout, Thierry Monteil, Georges Da Costa, Rodrigo Neves Calheiros, Rajkumar Buyya, Mihai Alexandru

a CNRS, LAAS, 7 Avenue du Colonel Roche, F-31400 Toulouse, France

Model and simulation of execution of power-aware DAGs in CloudSim was enabled with a series of extensions to the basic CloudSim objects in order to make them aware of the dependencies between tasks, as well as to account for data transfer between virtual machines. This implementation is independent from, and was developed concurrently with, the implementation by Chen and Deelman [26]. This implementation targets modeling and simulation of energy consumption and DVFS during workflow execution, whereas Chen and Deelman’s focused in a realistic modeling of workflow management systems, including advanced scheduling techniques, such as task clustering and fault-tolerant scheduling, without emphasis on power-aware simulation.

The method for computing the slack-time for each non-critical task of the DAG was obtained from Kimura et al. [32], which presents a study of reducing energy consumption using slack-time and DVFS. Others studies about workflow overheads were presented by Chen and Deelman [33] and Nerieri et al. [34].

AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

IN KRAKOW

CLOUD WORKFLOW SIMULATOR

Piotr Bryk, Bartłomiej Szczepanik

WorkflowSim is a simulator based on CloudSim. It specifically aims for workflow simulations. It includes infrastructure for reading, parsing and processing workflow input files. Although it seems that it is very similar to Cloud Workflow Simulator 2, it is different in many scopes. First of all it supports only single workflows, whereas CWS2 is built for simulating ensembles (sets of workflows). WorkflowSim also focuses on different areas of experiments, such as heterogeneous distributed systems, various failure and delays models. It models task queue which in batch scheduling systems can be of chief influence on simulation. Comprehensive description of the simulator can be found in its original research paper [1].

Advanced Computing & Communication Technologies (ACCT), 2014 Fourth International Conference on

Scheduling in Grid Computing Environment

Harshadkumar B. Prajapati, Vipul A. Shah, Information Technology Department, Dharmsinh Desai University Nadiad, INDIA

Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds

A Broker-based Framework for Multi-Cloud Workflows

Foued Jrad foued.jrad@kit.edu

Jie Tao jie.tao@kit.edu

Achim Streit achim.streit@kit.edu

Karlsruhe Institute of Technology KIT Steinbuch Centre for Computing Hermann-von-Helmholtz-Platz 1 Eggenstein-Leopoldshafen 76344, Germany

The Cloud Service Bro- ker components were implemented as value-added services on top of the modeled CloudSim datacenters. For managing large scale workflows, we simply applied WorkflowSim [4], a modeled WfMS developed on top of CloudSim. Similar to the Pegasus WfMS, WorkflowSim contains a Workflow Map- per to map abstract workflows to concrete workflows, which are dependent on execution sites, a Workflow Engine to han- dle the tasks and data flow dependencies, and a Clustering Engine to reduce the number of tasks by applying differ- ent merging techniques. For the purpose of this work, we extended WorkflowSim to use the Cloud Service Broker as the scheduler instead of using an external one. In addition, a Replica Catalog keeps a list of data replicas by mapping input/output filenames to their current site locations. Cur- rently, the data transfer is initiated by workflow tasks during their execution on the respective datacenters, whereas the Replica Catalog is managed by the Data Manager.

We would like to thank Weiwei Chen from the University of Southern California for his contribution to this work by providing us with the WorkflowSim source code.

ACM Transactions on Modeling and Computer Simulation (TOMACS) Volume 23 Issue 4, October 2013

On the Validity of Flow-level TCP Network Models for Grid and Cloud Simulations

PEDRO VELHO, UFRGS, Institute of Informatics, Brazil LUCAS MELLO SCHNORR, UFRGS, Institute of Informatics, Brazil HENRI CASANOVA, Dept. of Computer and Information Sciences, University of Hawai‘i at Manoa ARNAUD LEGRAND, CNRS, Grenoble University, France

Although using valid simulation models seems a prerequisite for developing a useful simulator, it turns out that many popular grid/cloud simulators use network models that have not been (sufficiently) validated. Table I illustrates a few simple experiments that invalidate four well-known simulators: OptorSim (2.1, 02/2010) [Bell et al. 2003], GridSim (5.2 beta, 11/2010) [Buyya and Murshed 2002], GroudSim (0.11, 06/2010) [Ostermann et al. 2010], and CloudSim (3.0.2, 11/2012) [Calheiros et al. 2011]. The versions we tested were the latest at the time of writing this article. These simulators have been used to obtain results published in hundreds of research articles, and also used as building blocks to develop other simulators [Chen and Deelman 2012; Teng et al. 2011; Shi et al. 2011; Jung and Kim 2012]. Each invalidating experiment in Table I can be devised through inspection of the simulator’s source code, and some are even sometimes documented by the developers themselves. Some model validation results have been published for other simulators, but they typically consider only a few cases in which simulation models are expected to work well [Zheng et al. 2004a, 2004b] (essentially merely verifying that model implementations are correct).

Discussion: http://blog.csdn.net/lingdianjiaoni/article/details/12838187

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=17&cad=rja&uact=8&ved=0CFkQFjAGOAo&url=https%3A%2F%2Fgroups.google.com%2Fd%2Fmsg%2Faut-cloud-computing-group%2F5lBf0UEfKC8%2Frma0GrFFJssJ&ei=uhVXU8eOLM6PyASM4oGYCw&usg=AFQjCNGlbNxGmxYaGR6bpOMteG8hRo0Aew&sig2=t1MXZVu5RgN6xlc7ozsPRg&bvm=bv.65177938,d.aWw