Modeling the accuracy of Monte-Carlo approach for Cloud based workflow simulations.

Luke Bertot and Stéphane Genaud and Julien Gossa
Icube-ICPS — UMR 7357, Univeristé de Strasbourg, CNRS
Pôle API Blvd S. Bant, 67400 Illkirch
email: lbertot@unistra.fr, gossa@unistra.fr, genaud@unistra.fr

***Abstract*—The pay as you go model of cloud operators makes exceeding the arbitrary time limits of the payment model costly. Budgeting the cost of running a scientific workflow requires users to be able to reliably predict runtimes of jobs within. Attempts to do so are hampered by the heterogeneity and opacity of cloud platform which make job runtimes both variable and hard to predict. Variable jobs call for a form of stochastic simulation, which can be achieved though either complex calculations heavily constrained to some reference cases or though a Monte-Carlo approach iterating on deterministic simulations. In this paper we study the limit of the Monte-Carlo approach by characterising the relation between the precision of the input variables and the simulations results, as well as the impact of the number of iteration for each level of input precision.**

***Keywords*-cloud computing, infrastructure as a service, simulation, montecarlo.**

I. Introduction

Whether to model impractical experiments or extract information out of large quantities of data, large scale computing is central to scientific, and sometime industrial, operations. Institutions have historically taken the burden of providing the computing power necessary for the research of it's members, usually through the acquisition of a cluster or the formation of a grid with it's existing resources, sometime pooling resource with other institution to achieve higher computing power. These resources are made available freely to members of the institutions within the constraints of time sharing and available computing power.

Over the last decade the advancement of virtualisation techniques has lead to the emergence of new economic and exploitation approach of computer resources in the form of Infrastructure as a Service (IaaS). In this model, usually referred to as cloud computing, all computing resources are be made available on demand by third-party operators and payed based on usage. The ability to provision resources on demand provided by IaaS operators is mainly used in two ways. Firstly for scaling purposes where new machines are brought online to fulfill service availability in the face of higher load, this approach is used for providing service allows for a lower baseline cost while still being able to deal by spikes in demand by provisioning machine on the go. Secondly for parallelizing tasks to achieve shorter makespan as equal cost, this approach is used for scientific and industrial workload with a clear end and where runtime is heavily dependent on computing power. This approach is made possible by the pricing model of cloud infrastructure, as popularized by AWS[1], in which payment for computing power, provided as Virtual machines (VMs), happens in increment of arbitrary length of time, billing time unit (BTU), usually of one hour. Running two Virtual machine (VM) side by side for one BTU each costs the same as running one VM for two BTU, but every BTU started is owed in full. As such within a workflow a slowed job forcing the subsequent job to run beyond the BTU limit can cause a full BTU to be invoiced for a handful of seconds of computation. Cases where such a thing might not always be avoided but IaaS to be reliably used in scientific computations the eventuality of an overrun must be reliably predicted and budgeted.

Accurate prediction of the runtime of scientific workloads is hampered by multiple factors. First IaaS operates in a opaque fashion, the exact nature of the underlying platforms are unknown and extremely heterogeneous as operators complete their data-centers over the years with new servers and equipment. Secondly cloud systems are multi-tenant by nature which ads uncertainty due to contention on network and memory accesses, depending on how VM are scheduled and the activity of your *neighbors*. IaaS operators attempt to mask these irregularities in computing power and network access by guaranteeing a minimum performance.

[1] Amazon Web Services

```
* Difficulties of simulation
- opaque platforms
- difficulties simulation precision (aka opaque codes)(cite simgrid ?)
- difficulties of stochastic simulation (cite robust DAG over possible approach)(cite elastic
- introduce Montecarlo (just lipservice)

! Related works
- Robust DAG by Jeanot et al : does the full run around of stochastic simulation approach (bac
- Elastic sim use montecarlo processes also for DAG
- Stochatsict Dag Sechduling  Zheng et Sakellariou -> looks for the best scheduling in a runti
? Simgrid
- Schlouder
? Towards a realistic performance model Lastovesky Rychkov -> but program execution variabilit

! Problem description

* Montecarlo as a workaround
- Deterministic simulator can be turned stochastic by being run on a sample of stochastic inpu

* Deterministic simulators are easy to evaluate they have 1 correct result for any set of inpu


* The stochastic component must be evaluated
- incorrect sample of inputs in the simulator will lead to wrong results
- how does the precision of the input affects the precision of the results
- how does the precision of the input affects the montecarlo process (ie. num of simulations n

! Methodology

* Rerunning experiments and simulator validation
- the schlouder Cloud batch scheduler
- simschlouder and how it relates to schlouder
- the experimental backlog (OMSSA/Montage and the platforms on with they ran

* the perfect model
- model based on real run values
- draw interval centered on real runtime

! Results
```