AI for Data Center Management and Cloud Computing: Papers from the 2011 AAAI Workshop (WS-11-08)

Reliability of Computational Experiments on Virtualised Hardware

Ian P. Gent and Lars Kotthoff
University of St Andrews
{*ipg,larsko*}*@cs.st-andrews.ac.uk*

Abstract

We present preliminary results of an investigation into the suitability of virtualised hardware – in particular clouds – for running computational experiments. Our main concern was that the reported CPU time would not be reliable and reproducible. The results demonstrate that while this is true in cases where many virtual machines are running on the same physical hardware, there is no inherent variation introduced by using virtualised hardware compared to non-virtualised hardware.

Introduction

Running computational experiments is a task that requires a lot of resources. Especially recent research in Artificial Intelligence is concerned with the behaviour of a large number of problem-solving systems and algorithms on a large number of problems (Xu et al. 2008; Kotthoff, Miguel, and Nightingale 2010). The purpose of these large-scale experiments is to build statistical models of the behaviour of certain systems and algorithms on certain problems to be able to predict the most efficient system for solving new problem instances.

The obvious problem is that a lot of computing resources are required to be able to run this kind of experiments. Provisioning a large number of machines is not only expensive, but also likely to waste resources when the machines are not being used. Especially smaller universities and research institutions are often unable to provide large-scale computing infrastructure and have to rely on support from other institutions.

The advent of publicly available cloud computing infrastructure has provided a possible solution to this problem. Instead of provisioning a large number of computers themselves, researchers can use computational resources provided by companies and only pay for what they are actually using. Nowadays commercial clouds are big enough to easily handle the demand running large-scale computational experiments generates.

This raises an important question however. How reliable and reproducible are the results of experiments run in the cloud? Are the CPU times reported more variable than on non-virtualised hardware?

While the focus of our evaluation is on computational experiments, we believe that the results are of interest in general. If a company is planning the provisioning of virtual resources, the implicit assumption is that the performance of the planned resources can be predicted based on the performance of the already provisioned resources. If these predictions are unreliable, too few resources could be provisioned, leading to a degradation of performance, or too many, leading to waste.

Related work

There has been relatively little research into the repeatability of experiments on virtualised hardware. (El-Khamra et al. 2010) report large fluctuations of high-performance computing workloads on cloud infrastructure. (Ostermann et al. 2010) evaluate the performance of the Amazon cloud with regards to its general suitability for scientific use. The handbook of cloud computing (Furht and Escalante 2010) explores the issue in some of its chapters.

An experimental evaluation by (Schad, Dittrich, and Quian-Ruiz 2010) again showed that there is large variability in performance and care must be taken when running scientific experiments. They provide an in-depth analysis of the various factors that affect performance, but only distinguish between two different virtual machine types provided by the Amazon cloud.

Our approach is more systematic and directly compares the variability of performance on virtualised and non-virtualised hardware with a real scientific workload. Our application is lifted straight from Artificial Intelligence research.

Problem statement

We are concerned with two major problems when running experiments. First, we want the results to be **reliable** in the sense that they faithfully represent the true performance of an algorithm or a system. Second, we want them to be **reproducible** in the sense that anybody can run the experiments again and achieve the same results we did.

We can assess the reliability of an experiment by running it several times and judging whether the results are the same within some margin of experimental error. Reproducibility is related to this notion, but more concerned with being able to reproduce the results in a different environment or at a different time. The two concepts are closely related however

– if we cannot reproduce the results of an experiment it is also unreliable and if the results are unreliable there is no point in trying to reproduce them.

Running experiments on virtualised hardware gives an advantage in terms of reproducibility because the environment that an experiment was run in can be packaged as a virtual machine. This not only removes possible variability in the results due to different software versions, but also enables to reproduce experiments with unmaintained systems that cannot be built and would not run on contemporary operating systems.

The questions we investigate in this paper however are as follows.

- Is there inherently more variation in terms of CPU time on virtualised hardware than on non-virtualised hardware?
- Is the performance of virtualised hardware consistent and are we able to combine several virtual machines into a cluster and still get consistent results?
- Are there differences between different clouds that use different controller software?

Experimental evaluation

To evaluate the reliability of experimental results, we used the Minion constraint solver (Gent, Jefferson, and Miguel 2006). We ran it on the following three problems.

- An n-queens instance that takes a couple of seconds to solve (place n queens on an $n \times n$ chessboard such that no queen is attacking another queen).
- A Balanced Incomplete Block Design (BIBD) problem that takes about a minute to solve (CSPLib (Gent and Walsh 1999) problem 028).
- A Golomb Ruler problem that takes several hours to solve (CSPLib problem 006).

There is a large variation of CPU time across the different problems. This enables us to isolate short-term effects (such as virtualisation of CPUs) from long-term effects (such as other jobs the operating system runs overnight).

We ran the experiments in three different settings –

- on three 8-core machines with non-virtualised hardware,
- on the Eucalyptus-based private StACC cloud[1] and
- on the public Amazon cloud.

For the Amazon cloud, we investigated the different virtual machine types `m1.large`, `m1.xlarge`, `c1.xlarge` and `m2.4xlarge`[2]. In each case, we provided 16 cores to run the experiments, i.e. 8 different virtual machines for `m1.large` and 2 different virtual machines for `m2.4xlarge`. In the StACC cloud, we used 5 virtual machine instances with 2 cores each.

Using several virtual machines introduces an additional source of variation, but at this stage of the evaluation we are interested in the reliability of experimental results that require a large amount of resources and therefore several machines.

experimental setting	n-queens	BIBD	Golomb Ruler
non-virtualised	0.016	0.018	0.005
StACC	0.013	0.022	0.009
Amazon `m1.large`	0.333	0.13	0.183
Amazon `m1.xlarge`	0.264	0.235	0.271
Amazon `c1.xlarge`	0.055	0.028	0.042
Amazon `m2.4xlarge`	**0.008**	**0.008**	**0.003**

Table 1: Coefficient of variation for all experiments. The lowest figures for each problem are in **bold**.

The experiments on non-virtualised hardware establish the baseline of reliability we can expect. We can then compare the reliability on virtualised hardware to see if it is significantly worse. Each problem was solved 100 times. We used the coefficient of variation (standard deviation divided by mean) of the CPU time required to solve a problem across the 100 runs as a measure of the reliability of the results.

Results and analysis

The results for all problems and experimental settings are summarised in Table 1. We were surprised to find that the coefficient of variation of the reported CPU time on the largest virtual machine type in the Amazon cloud was lower than what we achieved on non-virtualised hardware. This demonstrates that running on virtualised hardware does not introduce additional variability per se.

We furthermore observed the general trend of the coefficient of variation decreasing as the experiment takes longer to run. This does not seem to be true on virtual machine types that have a large coefficient of variation though. Overall, the differences between the different experimental settings are two orders of magnitude. This is an indication that evaluations like this one are necessary and we cannot assume that the performance of any given virtual machine will be consistent and reliable.

The variation for each individual run is depicted in Figure 1 for the n-queens problem and Figure 2 for the Golomb Ruler problem. The distribution for the n-queens problem, which takes only a few seconds to solve, is more or less uniform. For the Golomb Ruler, which takes several hours to solve, however, there are distinct plateaus. We believe that these are caused by the different virtual machines we used. That is, two of the eight virtual machines used of type `m1.large` were significantly slower than the rest. Such a difference is still visible for type `c1.xlarge`, where two different virtual machine instances were used. There is no noticeable difference between the two `m2.4xlarge` instances however.

The coefficient of variation of the Eucalyptus-based StACC cloud is very similar to the one on non-virtualised hardware and not significantly better or worse than that of the Amazon cloud.

Conclusions and future work

We have presented the results of a preliminary evaluation of the variation of CPU time on virtualised vs. non-virtualised

[1] http://www.cs.st-andrews.ac.uk/stacc

[2] http://aws.amazon.com/ec2/instance-types/

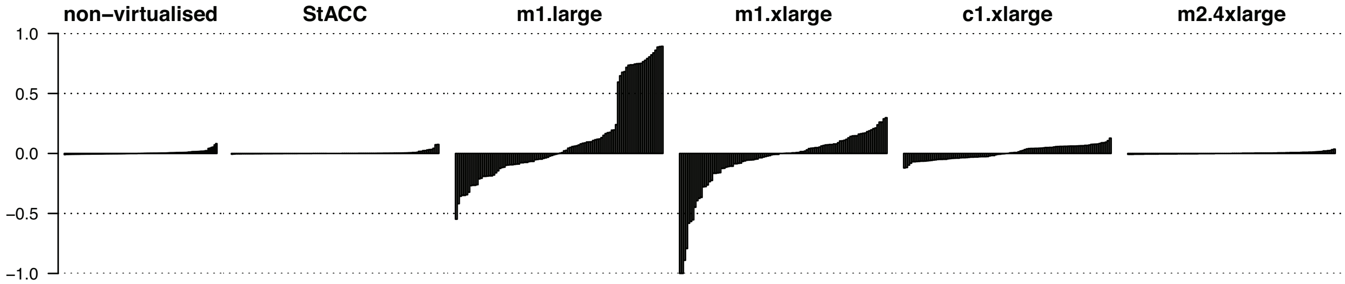

Figure 1: Relative deviation from the median CPU time for the n-queens problem for each run. A value of 1 means that the run took twice as long as the median, -1 that it took half as long.

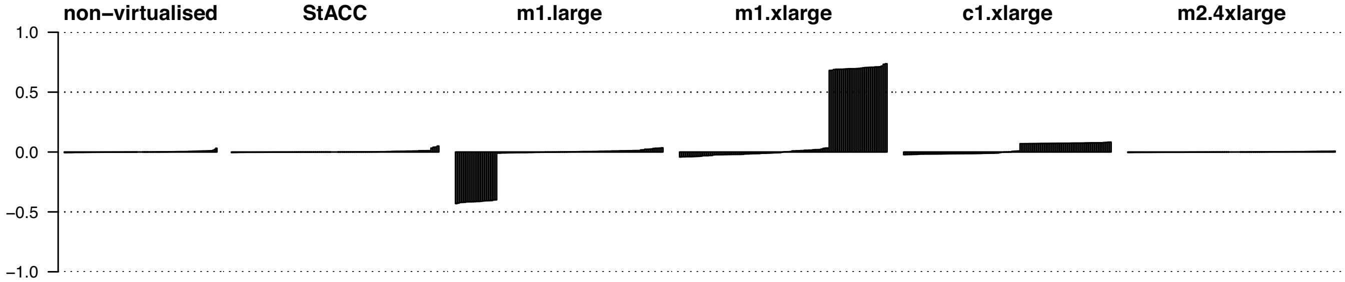

Figure 2: Relative deviation from the median CPU time for the Golomb Ruler problem for each run.

hardware. We can draw the following conclusions.

- The differences in variation across different types of virtual machines and non-virtualised hardware can be several orders of magnitude.
- Virtualised hardware does not introduce additional variation compared to non-virtualised hardware per se. This does not hold true for all types of virtual machines however.
- Performance varies across different instances of the same virtual machine type, but the variation decreases for larger virtual machine types.
- There does not appear to be a significant difference between different cloud systems (StACC Eucalyptus cloud and Amazon cloud).

The variation of CPU times on the largest virtual machine type on the Amazon cloud (`m2.4xlarge`) is at least as good as on non-virtualised hardware. In terms of reliability of results, it is therefore a feasible alternative to physical hardware to run experiments on. The high price of this instance type however eliminates some of the benefits of not having to provision hardware and paying only for what is actually used.

In the future, we are planning on investigating the variation between different virtual machines of the same type further; especially across different data centres. We are also planning on investigating the repeatability of experimental results over time. The evaluation of the financial feasibility is another important subject for future research.

Acknowledgments

This research was supported by a research grant from Amazon Web Services. Lars Kotthoff is supported by a SICSA studentship.

References

El-Khamra, Y.; Kim, H.; Jha, S.; and Parashar, M. 2010. Exploring the performance fluctuations of HPC workloads on clouds. In *CloudCom*, 383–387.

Furht, B., and Escalante, A., eds. 2010. *Handbook of Cloud Computing*. Springer.

Gent, I. P., and Walsh, T. 1999. Csplib: a benchmark library for constraints. Technical report, APES-09-1999.

Gent, I. P.; Jefferson, C.; and Miguel, I. 2006. MINION: a fast, scalable, constraint solver. In *ECAI*, 98–102.

Kotthoff, L.; Miguel, I.; and Nightingale, P. 2010. Ensemble classification for constraint solver configuration. In *CP*, 321–329.

Ostermann, S.; Iosup, A.; Yigitbasi, N.; Prodan, R.; Fahringer, T.; and Epema, D. 2010. A performance analysis of EC2 cloud computing services for scientific computing. In *CloudCom*, 115–131.

Schad, J.; Dittrich, J.; and Quian-Ruiz, J. 2010. Runtime measurements in the cloud: observing, analyzing, and reducing variance. *Proc. VLDB Endow.* 3:460–471.

Xu, L.; Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2008. SATzilla: portfolio-based algorithm selection for SAT. *J. Artif. Intell. Res. (JAIR)* 32:565–606.