Journal of Experimental & Theoretical Artificial Intelligence
Vol. 00, No. 00, Month-Month 200x, 1–15

Reliability of Computational Experiments on Virtualised Hardware

Lars Kotthoff (larsko@4c.ucc.ie)
Cork Constraint Computation Centre, University College Cork, Western Gateway Building, Cork, Ireland
(*Received 00 Month 200x; In final form 00 Month 200x*)

We present a large-scale investigation of the variability of run times on physical and virtualised hardware. The aim of our investigation is to establish whether cloud infrastructures are suitable for running computational experiments where the results rely on reported run times. Our application is the use of the Minion constraint solver as an example of an Artificial Intelligence experiment. We include two major providers of public cloud infrastructure, Amazon and Rackspace, as well as a private Eucalyptus cloud. While there are many studies in the literature that investigate the performance of cloud environments, the problem of whether this performance is consistent and run time measurements are reliable has been largely ignored.

Our comprehensive experiments and detailed analysis of the results show that there is no intrinsic disadvantage of virtualised hardware over physical hardware and that in general cloud environments are suitable for running computational experiments. Our meticulous investigation reveals several interesting patterns in the variability of run times that researchers using a cloud for this purpose should be aware of. We close by giving recommendations as to which type of virtual machine with which cloud provider should be used to achieve reproducible results.

Keywords: virtualisation; cloud; computational experiments

1 Introduction

Cloud computing, utility computing or on-demand computing is a computing paradigm that provisions resources as and when they are needed. Instead of buying and maintaining a large pool of computational resources that is mostly idle, capacity is added only if it is required and relinquished when demand diminishes. This capacity usually takes the form of infrastructure that is rented from a cloud computing provider.

One key enabler of this paradigm is virtualisation. Virtual resources are provisioned on top of physical resources to allow for more fine-grained control of what is used. The term *virtual machine* is commonly used to describe a bundle of virtualised resources that provide comparable functionality to a physical machine. A physical machine can be used to provide resources for several virtual machines. This increases the overall utilisation of the physical hardware but maintains the illusion that dedicated hardware is being provided to the customer.

The provision of virtualised resources has many benefits, but also raises some issues. If a piece of code is run on a physical CPU, it is relatively easy to determine its performance. If the CPU is virtualised, it becomes much harder to do so. The performance depends not only on the physical CPU that the virtual CPU runs on and the virtualisation software, but also on other virtual CPUs that happen to use the same physical CPU. The same issues apply to other resources such as network bandwidth and disk throughput.

Cloud computing providers usually give an indication of the physical hardware that backs the available virtualised resources. This may however not be a good indicator of the performance that can be achieved in practice; in particular two different virtual machines running on the same physical hardware may exhibit significantly different performance behaviour in practice.

This paper investigates the differences in CPU performance in practice for a number of different virtualised resources and compares the results to what can be achieved on non-virtualised physical resources. The focus of the investigation is not on the absolute performance that can be achieved, but rather how *consistent* that performance is.

For applications in academic and industrial research, this question is often the more important one. Scientific rigour demands that results are presented in a way that they can be verified by the scientific community. If the infrastructure used for empirical investigations does not enable this however, it becomes

unusable for this purpose.

2 Background

Advancements in Artificial Intelligence research are mostly twofold. On one hand, theoretical investigations provide an insight into problems. On the other hand, improvements may enable us to solve particular problems better in practice. A lot of research presents theoretical advances that are then used to show that it can be used to achieve performance improvements in practice. Often, practical improvements are concerned with reducing the time required to solve a problem.

To demonstrate these performance improvements, computational experiments are run that compare the new approach to previous approaches. In many cases, running these experiments requires a lot of computational resources. Unless the improvements are very clear, a large amount of empirical data may be required to build statistical models of the behaviour of a set of algorithms.

This is an application where utility computing has the potential to reduce costs significantly while providing greater flexibility at the same time. Researchers who need to carry out extensive empirical investigations face a trade-off between provisioning too few machines and not being able to gather a sufficient amount of experimental data in time for a deadline and provisioning too many and having to pay for and maintain machines that are mostly idle. This is not only an issue of financial cost, but also of research staff spending their time on system administration duties and increasing the burden on the environment through higher energy consumption. In addition to these considerations, it may not be possible for smaller groups of researchers to provide the up-front infrastructure investment to run large-scale experiments or even provide the physical space for a lot of machines.

There is another compelling reason to use virtualised environments for gathering empirical data in a scientific context. Research publications sometimes go to great lengths to make sure that interested parties will be able to reproduce experiments. Not only the physical hardware used is of interest, but also the operating system and system software. However, replicating these exact conditions is usually infeasible. This may be because the exact hardware or software is not available anymore. Even if that is the case, it requires significant commitment by either modifying the configuration of an existing machine or provisioning and configuring a new machine.

Running experiments in a virtualised environment allows to "shrink-wrap" the experiment and the conditions it was run in, for example by taking a snapshot of the disk image of the virtual machine. This shrink-wrapped experiment can be made publicly available alongside a publication to enable anyone to easily reproduce the experimental results. It does not only take care of reproducing the same environment, but also includes all other configuration steps and preliminaries that may be required. There remains the issue of replicating the environment the virtual machine was run in.

The external factors that affect the behaviour of a virtual machine, by design, cannot be controlled by a user of cloud computing. The provider of the infrastructure may change the underlying physical hardware for example. Before this becomes an issue however, we need to establish whether it is feasible at all to provide such shrink-wrapped experiments or even to run computational experiments reliably. This investigation focuses on the latter issue.

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 shrink-wrapped. 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 main questions under investigation in this paper are as follows.

- Is there inherently more variation in terms of CPU time on virtualised hardware than on non-virtualised hardware?
- Are the CPU times reported on virtualised hardware more variable than on non-virtualised hardware?
- How reliable and reproducible are the results of experiments run in the cloud?
- How does this behaviour vary across different clouds?

While our focus is on computational experiments, we believe the subject to be 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. This is one of the fundamental differences of physical and virtualised hardware – a physical machine that has the same specifications as another physical machine that has already been provisioned will most likely exhibit the same behaviour.

3 Related work

Noise and variance in measurements is an issue not only in Artificial Intelligence, but in Computer Science and indeed science in general. If the accuracy of the measurement is less than the expected effect, it cannot be observed reliably. A common approach to mitigating this problem is to repeatedly take measurements and assess the variability to be able to account for the error.

While the analysis and mitigation of measurement errors in the physical sciences has a long tradition and there are procedures to deal with them, this is much less the case in Computer Science. Cohen (1995) notes that

> "If noise factors turn out to have large effects, then variance within conditions will be larger than we like [...]."

He refers to "noise variables" as external factors that can be observed and measured reliably. The random noise introduced by empirically observing run times of programmes does not fall into that category, but the statement is still relevant – we certainly want the effects of the noise to be as small as possible.

Most publications that report improvements on previous methods by presenting empirical results demonstrate improvements by a large factor or even orders of magnitude. In this context, random noise in experimental runs is not much of an issue. However, usually the improvement is not that large across all experiments. On a fraction of the empirical measurements, the difference between two approaches is usually small enough to be influenced by random noise. While a small number of publications addresses this issue by for example repeating measurements and reporting an average, it is ignored by many.

3.1 *Cloud experiments*

In Artificial Intelligence, there are very few publications that use the cloud as a computational resource. Burkett et al. (2011) use it as a drop-in replacement for physical hardware. They repeat individual experiments many times and average the run times to account for variations. Lu et al. (2011) on the other hand present a case that looks at the specific characteristics of the infrastructure of the cloud and adapts a common Artificial Intelligence application to it. They report good empirical results for their adaptation.

Klinginsmith et al. (2011) consider eScience in general and describe how to reproduce an experimental setup with empirical results in a cloud environment. They focus on recreating the experimental environment to enable other researchers to replicate experiments and do not go as far as saving and restoring the entire virtual machine image. The variability of empirical measurements in the cloud is something they do not consider at all.

The performance of virtualised hardware and cloud-provisioned resources has been of interest to researchers since the emergence of the techniques. Accordingly, a relatively large number of publications considers this issue. In a comparison of different virtualisation techniques, Matthews et al. (2007) conclude

that in general techniques other than operating system level virtualisation achieve good performance isolation, although a badly behaving virtual machine does have a measurable impact on other virtual machines running on the same physical hardware. Koh et al. (2007) perform a similar investigation, but come to the conclusion that the investigated virtualisation technique does *not* provide sufficient performance isolation.

In a more recent analysis, Younge et al. (2011) focus specifically on comparing the performance of different virtualisation techniques that underlie cloud infrastructure. They conclude that virtual machines based on the KVM hypervisor deliver the best general performance for high-performance computing.

Walker (2008) performs one of the earliest investigations into the performance of the Amazon cloud and its suitability for high-performance computing. He concludes that the performance gap between physical and virtual hardware is substantial and inhibits the adoption of cloud computing for scientific purposes. Ostermann et al. (2009) perform a similar investigation and come to similar conclusions.

Jackson et al. (2010) compare the performance of Amazon EC2 virtual machines to other HPC systems. They come to similar conclusions as the previous studies and note in particular that performance of cloud systems decreases as the amount of network communication increases. They close by remarking that

> "Variability is introduced by the shared nature of the virtualized environment, by the network, and by differences in the underlying non-virtualized hardware."

Rehr et al. (2010) perform another investigation of cloud performance for HPC applications along similar lines. They also conclude that network performance is the main factor that inhibits replacing physical HPC infrastructure but note that for applications with low network requirements, the cloud is a feasible alternative to physical hardware.

El-Khamra et al. (2010) focus on the predictability of the performance of cloud-based systems and consider not only the Amazon cloud, but also a Eucalyptus cloud. They report large fluctuations from the mean performance for both systems. Their conclusion, similar to the other investigations, is that most of the variability can be attributed to network communications. Wang and Ng (2010) investigate the performance impact of network virtualisation specifically. Their results demonstrate that even under favourable settings, the performance can vary significantly and unpredictably.

Variability of run time measurements is one of the main issues that Schad et al. (2010) consider. They run experiments on the Amazon cloud and are able to identify two different performance levels that correspond to the two different physical hardware specifications that provide the cloud infrastructure. They note that there is considerable variation in the reported wall clock times and provide an in-depth analysis of the various factors that affect performance.

Lenk et al. (2011) show that the performance indicators given by cloud computing providers are not sufficient to determine the actual performance that a virtual machine will achieve. Some of the workloads they use in their experiments are very similar to workloads in Artificial Intelligence research. They note that starting two virtual machines of the same type may give different performance characteristics.

Our application comes directly from Artificial Intelligence research and does not consider network performance variability at all. Most of the previous studies have concluded that network communications are a significant factor that affects performance in practice, but the issue of variability *without* network communications remains largely unaddressed.

The study with most similarities to ours is Schad et al. (2010). They have the same focus of assessing the variability that virtualised hardware introduces into empirical measurements. They consider database instead of Artificial Intelligence applications however. Furthermore, they only compare the performance of physical hardware and virtualised hardware on Amazon cloud infrastructure. Our evaluation is significantly more comprehensive, both in terms of evaluating more different clouds and virtualisation techniques as well as a larger number of different virtual machine types in the Amazon cloud.

4 Methodology

We use the Minion constraint solver (Gent et al., 2006) in our investigation. Constraint solving is an area of Artificial Intelligence where advances of the state of the art are demonstrated by empirical evaluations

most of the time. The question of how variable reported results are is therefore highly relevant to this area. No background knowledge in constraint solving is required; for an introduction see for example Dechter (2003).

The Minion constraint solver suits the purpose of our investigation very well. It does not require any network communication and very little disk access and enables us to assess the variability based only on the virtualised processors. It is a state of the art system that has been used in academic and industrial applications.

We assess the performance using three constraint problems. The choice of these particular problems was motivated purely by the time they require to be solved – the first solves in a few seconds, the second in approximately a minute and the third takes several hours to solve. This large spread in run times enables us to assess both short-term and long-term effects. Short-term effects could be for example bursts of processor cycles being available when the physical system is lightly utilised while an example of a long-term effect is the general processing power provided by a virtualised processor.

The specific problems we used for the investigation are described below, in ascending order of run time to find a solution. Note that for the purpose of this paper, it is not necessary to understand the nature of these problems. We provide the description merely for the sake of completeness.

n-Queens Place n queens on an $n \times n$ chessboard such that no queen is attacking another queen. We used $n = 29$.

Balanced Incomplete Block Design (CSPLib (Gent and Walsh, 1999) problem 28) A Balanced Incomplete Block Design (BIBD) is defined as an arrangement of v distinct objects into b blocks such that each block contains exactly k distinct objects, each object occurs in exactly r different blocks, and every two distinct objects occur together in exactly λ blocks. v, b, r, k and λ are given, although b and r can be derived from the other parameters. We used $v = 7$, $k = 3$, $\lambda = 70$.

Golomb Ruler (CSPLib problem 6) For a given m, a Golomb ruler is defined as a set of m integers $0 = a_1 < a_2 < \ldots < a_m$ such that the $\frac{m(m-1)}{2}$ differences $a_j - a_i, 1 \leq i < j \leq m$ are distinct. Such a ruler is said to contain m marks and is of length a_m, i.e. the position of the last mark. The length is to be minimised. We used $m = 13$.

We used Minion version 0.12, which is available for download at its website[1].

Each problem was run 100 times. There was no effort made to intersperse runs of different problems, ensure optimal utilisation of the used processors or minimise external influences. This reflects the intention of using the cloud as a drop-in replacement for physical hardware without any adjustments – experiments are run as if the infrastructure was used exclusively for this purpose.

Our experiments serve two main purposes. First, to compare the variability of reported run times on virtualised infrastructure to the variability on physical infrastructure. Second, to assess how the reported run times vary across different virtual machines, different locations and different cloud providers.

We ran the experiments in the following four different settings.

- on three 8-core machines with non-virtualised hardware,
- on a Eucalyptus-based private cloud,
- on the public Amazon cloud and
- on the public Rackspace cloud.

The Eucalyptus cloud consisted of homogeneous hardware at the time when the experiments took place. In particular, each virtual processor was backed by a physical processor. The operating system on the physical hardware was CentOS Linux while on the virtual machines Ubuntu Linux was used.

For the Amazon cloud, we investigated the virtual machine types[2] `m1.large` (2 processors), `m1.xlarge` (4 processors), `c1.xlarge` (8 processors) and `m2.4xlarge` (8 processors) in the US East (Virginia) region. In each case, we provided 16 processors to run the experiments, i.e. 8 different virtual machines for `m1.large` and 2 different virtual machines for `m2.4xlarge`. In the Rackspace cloud, we used instances with 2048

[1] `http://minion.sf.net`

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

and 15872 MB of physical memory in the UK data centre. Each virtual machine had 4 processors and we used them in groups of three to provide 12 processors in total. In the Eucalyptus cloud, we used 5 virtual machine instances with 2 processors each. The amount of memory provided with each virtual machine was sufficient to prevent swapping or similar behaviour.

Experiments run in all four settings allow us to compare the variability of virtualised and physical hardware. The physical hardware establishes the baseline of reliability. 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.

For the purpose of evaluating the variability across different virtual machines and cloud providers, we focussed on the publicly accessible Amazon and Rackspace clouds in a second set of experiments. On the Amazon cloud, we used the additional cluster instance type `cc1.4xlarge` (8 processors) in the US East (Virginia) region and the `m1.xlarge`, `c1.xlarge` and `m2.4xlarge` instance types in the EU (Ireland) region. We did not use the smallest used type `m1.large` again. Each type of virtual machine was run 10 times. In most cases all 10 instances were started at the same time; however, a quota on the Rackspace cloud prevented us from running all virtual machines at the same time and they were run in smaller batches sequentially. Each virtual machine instance again solved all problems 100 times. No communication between the instances took place.

Our aim was to conduct experiments in as many different locations and settings as possible, but ultimately the scope of this investigation was limited by the cost of running the experiments.

For all setups, we used the Condor HPC system (Thain et al., 2005) regardless of whether a single machine was used on its own or in a group with other machines. While this setup does cause network traffic in some cases, it does not affect our measurements. We used the CPU times as collected and reported by Minion itself, which ignores any overhead incurred by using Condor. Our basic Condor configuration instructed it to use as many processors as a machine reports to the fullest extent regardless of any other activity on the system.

The experiments were conducted throughout 2011 and the results reflect the hardware that provided resources for the virtual machines at the time. The results do not necessarily reflect the performance of the current cloud configurations, but we are confident that they can be used as a guideline.

5 Experimental results

First, we will present the results for the first set of experiments. The focus is to establish whether using virtualised hardware has an intrinsic disadvantage over physical hardware in terms of variability. After that, the second set of results will be described with focus on the variability across different virtual machines in public clouds.

We use the normalised median absolute deviation as a measure of variability for the measurements in a particular experimental setting. The median absolute deviation is the median of the absolute differences of individual measurements from the median of all measurements. This number is normalised by dividing it by the median of all measurements to make the figures comparable across experiments with different median run times.

The motivation for using a measure based on the median average is that it will be able to characterise the *expected* behaviour better than other measures that are based on the mean average, which is susceptible to outliers. It is the expected behaviour that we are most interested in here.

5.1 *Comparison to physical hardware*

Table 1 presents the normalised median absolute deviations for all problems and machine types for the first set of experiments. In general, the variation is larger for the problems that take only a relatively short time to run. This is due to the fact that the operating system requires processing power for other tasks that run at the same time as the experiments. These are usually short tasks that have only small effects, but for short experiments the relative impact is larger than for experiments that take a long time to run.

experimental setting	n-Queens	BIBD	Golomb Ruler
non-virtualised	0.007	0.01	0.003
Eucalyptus	**0.003**	0.01	**0.002**
Amazon `m1.large`	0.205	0.13	0.011
Amazon `m1.xlarge`	0.13	0.175	0.034
Amazon `c1.xlarge`	0.073	0.019	0.029
Amazon `m2.4xlarge`	0.006	**0.006**	0.003
Rackspace 2048 MB	0.006	0.009	0.005
Rackspace 15872 MB	0.007	0.01	0.005

Table 1. Normalised median absolute deviation for the experiments of the first set on physical and virtualised hardware. The numbers were rounded to three decimal places. The lowest figures for each problem are in **bold**.

We were surprised that the lowest variability for all problems was not achieved on the physical hardware, but on virtual machines running in the cloud. In fact, for the n-Queens and BIBD problems, the best variability improves over the variability we observed on physical hardware by a factor of two approximately. This conclusively shows that there is no inherent disadvantage to running experiments in the cloud when it comes to reproducibility of the results.

The normalised median absolute deviations across the three different clouds vary, but no cloud is consistently better than the other ones. The Eucalyptus cloud seems to have a slight advantage with the lowest variability on two out of three problems and the Rackspace cloud consistently exhibits larger variability than the others.

Figure 1 shows the results for the first set of experiments in more detail. The machines that exhibit the smallest variation also exhibit the best overall performance with the exception of the virtual machines in the Rackspace cloud. The processors provided by this infrastructure are significantly slower than the processors of almost all the other machines.

On the `m1.xlarge` instance type, some of the run times for the n-Queens and BIBD problems were reported as 0. This is clearly incorrect and highlights the unreliability of such measurements on virtualised hardware. This only occurred in a small number of cases however.

The figure also makes clear that for the Amazon virtual machine types with lower specifications exhibit not only varying performance, but different levels of performance. The distributions show several peaks, indicating the performance levels of the different virtual machine instances that the experiments were run on. This is especially obvious in the distributions for machine type `c1.xlarge`. We ran the experiments on two instances of that type to provide a total of 16 processors and the distributions show two peaks of similar magnitude. Each of the `m1.large` and `m1.xlarge` virtual machine types provides fewer processors and we had to run more instances to achieve 16. Hence the magnitudes of the peaks of the distributions are different. The same effect does not occur on physical hardware and virtual machines with higher specifications. Even though the experiments are running on multiple instances, the performance they exhibit is consistent.

5.2 *Comparison of virtual machine instances*

After having established that the variability on virtualised hardware is not intrinsically higher than on physical hardware, we now investigate the variability across virtual machines of the same type in more detail.

Table 2 presents the normalised median absolute deviations for the second set of experiments. They are similar to those presented in Table 1. We were surprised that the lowest variability for two problems was achieved by a virtual machine type with relatively low specifications. This result is not consistent with the other results though and may be due to a specific cloud configuration that is unlikely to occur in general.

Figure 2 shows more detailed results similar to Figure 1 for the first set of experiments. Again we observe that the virtual machine types with low specifications exhibit more variability (with the exception of the `m1.xlarge` type in Amazon's EU region). In terms of absolute performance, the machines running in the

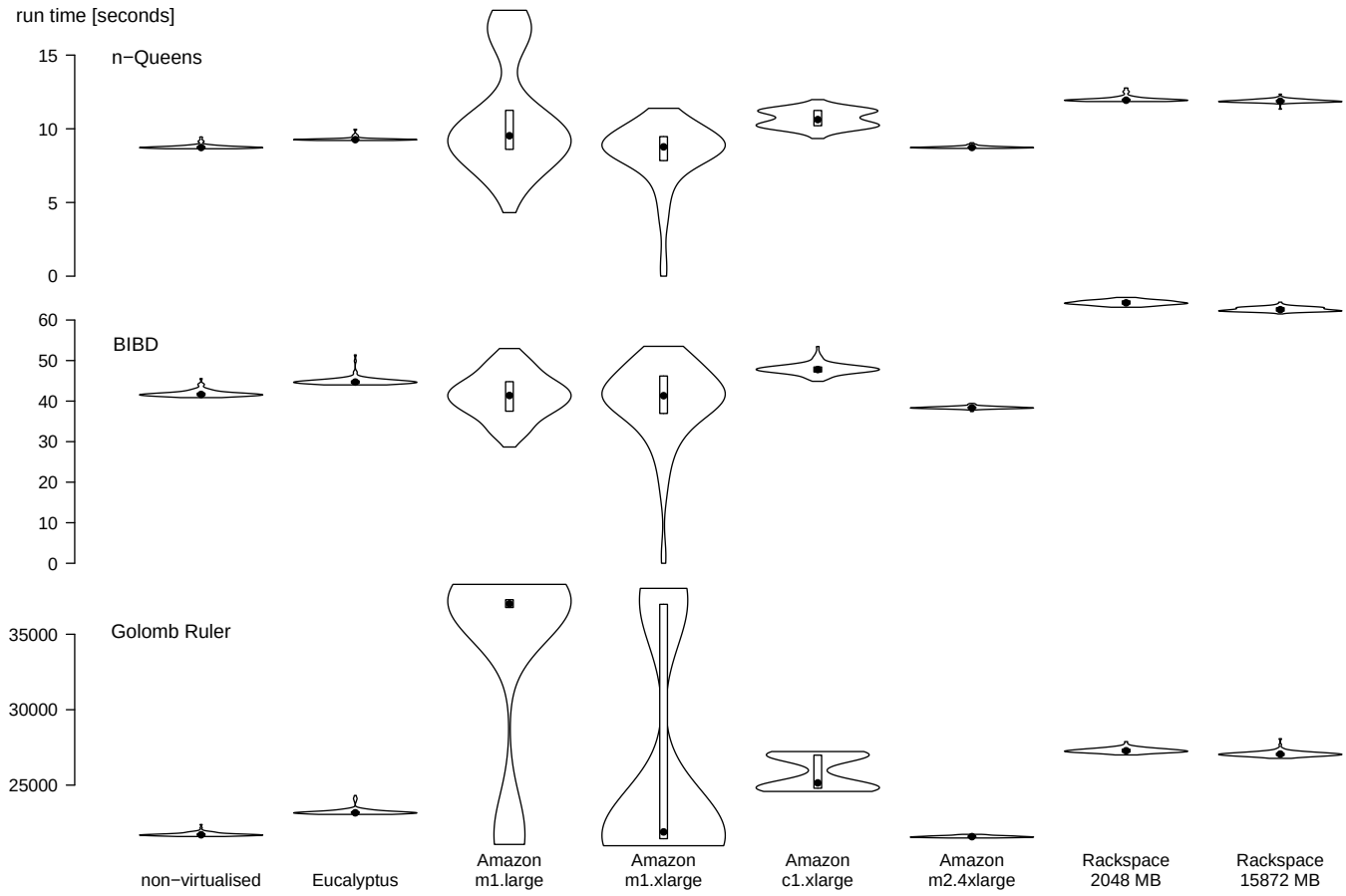

Figure 1. Distribution of run times for all scenarios of the first set of experiments. The solid dot denotes the median, the bottom and top of the boxes denotes the 25th and 75th percentile, respectively. The width of the shape denotes the number of measurements.

experimental setting	n-Queens	BIBD	Golomb Ruler
Amazon `m1.xlarge` US East	0.094	0.049	0.007
Amazon `m1.xlarge` EU	**0.005**	0.007	**0.003**
Amazon `c1.xlarge` US East	0.055	0.042	0.051
Amazon `c1.xlarge` EU	0.024	0.039	0.011
Amazon `m2.4xlarge` US East	**0.005**	0.01	**0.003**
Amazon `m2.4xlarge` EU	0.008	0.007	**0.003**
Amazon `cc1.4xlarge` US East	0.015	0.012	0.011
Rackspace 2048 MB	0.008	0.008	0.005
Rackspace 15872 MB	0.006	**0.006**	0.005

Table 2. Normalised median absolute deviation for all experiments of the second set on the Amazon and Rackspace clouds. The numbers were rounded to three decimal places. The lowest figures for each problem are in **bold**.

Amazon cloud again deliver better run times than the machines in the Rackspace cloud.

Especially the `c1.xlarge` machine type exhibits run time distributions that have several peaks again. Even though 10 different virtual machines instances were used, there are only two peaks. This indicates that two different configurations are used for the underlying physical hardware. The two peaks emerge consistently across Amazon's US East and EU regions. However, in general the distributions of run times for an individual scenario and problem is much more akin to a Gaussian bell curve than in the first set of experiments. This is probably because of the much higher number of measurements.

The `cc1.4xlarge` type in Amazon's US East region exhibits a much larger variability of run times and worse performance than we expected for an instance type of its specifications. Amazon note that the

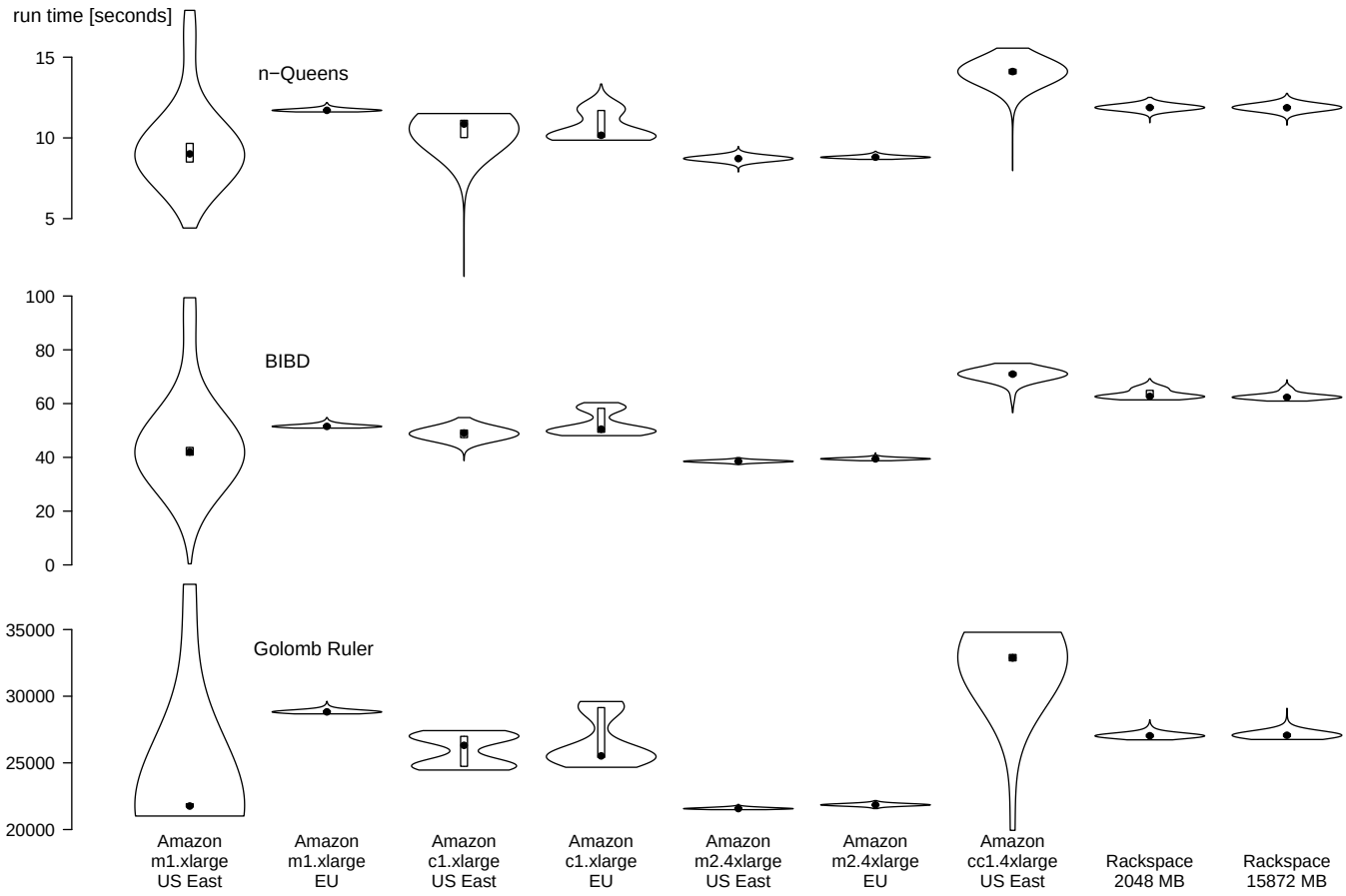

Figure 2. Distribution of run times for all scenarios of the second set of experiments. The solid dot denotes the median, the bottom and top of the boxes denotes the 25th and 75th percentile, respectively. The width of the shape denotes the number of measurements.

target customer segment for this type includes researchers who need to run compute-intensive tasks. The `cc1.4xlarge` type has eight virtual processors, but hyperthreading is enabled and instances report 16 processors. This means that our experimental setup ran 16 problems at once instead of eight that would have corresponded to the number of virtual processors. This is likely the source of the high variability and low performance. It indicates that researchers who are concerned about the repeatability of their measurements should double-check whether technologies such as hyperthreading are enabled.

For a clearer picture of the performance of the `cc1.4xlarge` type, the experiments need to be repeated with special provisions limiting the number of parallel runs to the number of virtual processors. Unfortunately, we did not have sufficient financial resources to rerun the experiments on this very expensive instance type.

Full details for all experiments are given in Figure 3. Each individual coloured box corresponds to the measurement for an individual experiment. There are two effects on variability that can be easily distinguished in this plot. The rows denote individual virtual machines instances and show performance difference across different instances. The columns denote individual runs over time and show temporal performance differences. The graph makes it obvious that both types of effects occur.

There are several scenarios where sets of virtual machine instances can be distinguished from another. This occurs for example for the `m1.xlarge` type in the Amazon US East region and the `c1.xlarge` type in the Amazon EU region. The Rackspace cloud appears to run on very homogeneous physical infrastructure; the only occasion when a single instance stands out is for the BIBD experiments on the 15872 MB type.

In multiple cases, temporal effects can be seen. The performance of all virtual machine instances increases towards the end of the experiments on the `c1.xlarge` type in the Amazon EU region for example. At this point during the experiments, it is likely that not all of the processors were utilised for experiments because there were fewer experiment runs than processors left. This is most likely the cause of the effect, although it is surprising to see it so pronounced. This effect could be an indication that the physical processors are

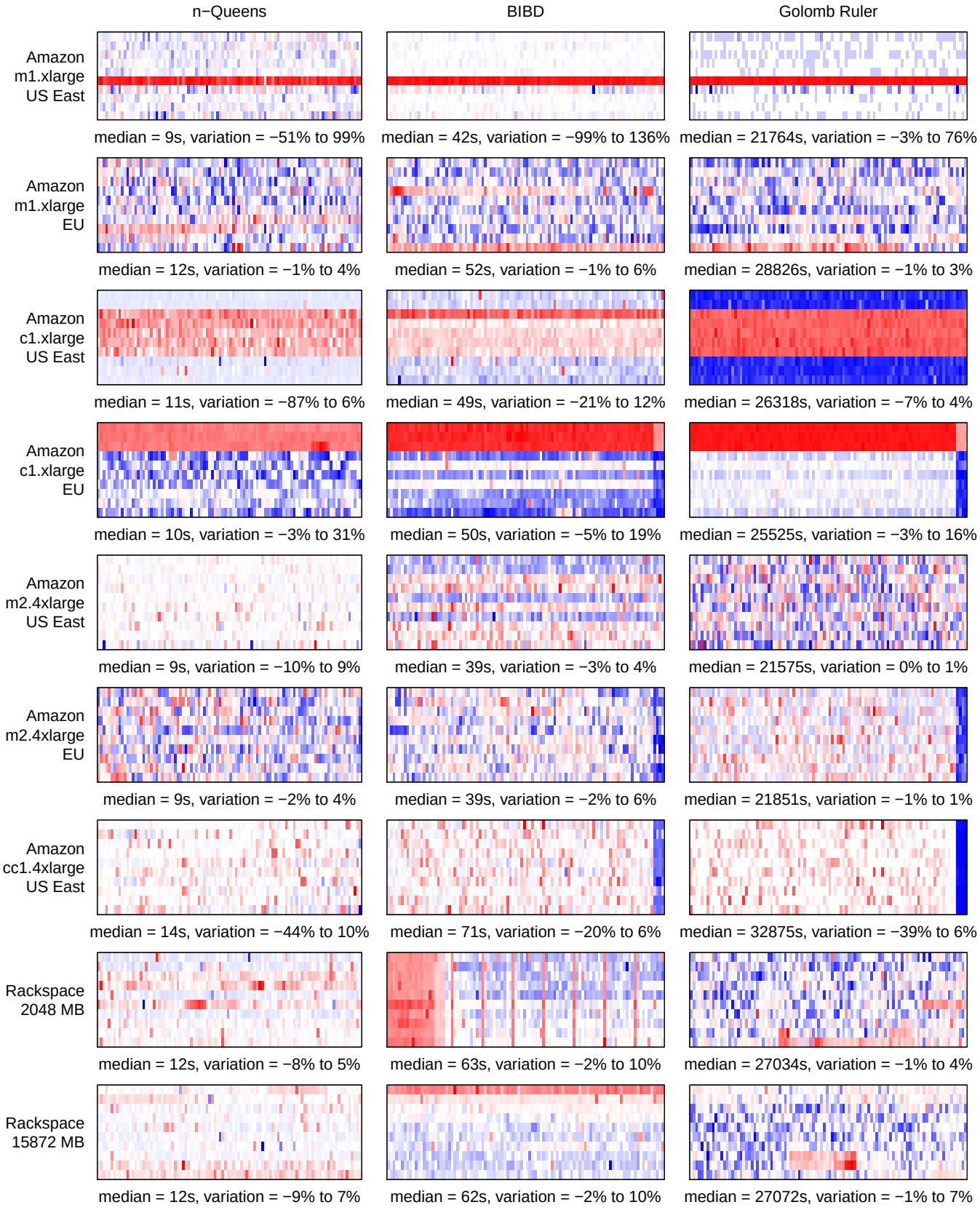

Figure 3. Variation from the median of each scenario for all runs of the second set of experiments. Within an individual box, rows represent different virtual machines and columns different runs. White denotes the median, shades of red slower than the median and shades of blue faster than the median. Red and blue themselves represent the maximum variation for each scenario. Note that while the rows are in temporal order (earliest experiment left and latest right), individual runs were not sequential but n in parallel, where n is the number of processors on the machine. The variation refers to percent difference from the median.

experimental setting	n-Queens	BIBD	Golomb Ruler
`m1.xlarge`	0.046	0.071	0.179
`c1.xlarge`	0.044	0.035	0.048
`m2.4xlarge`	0.008	0.017	0.009

Table 3. Normalised median absolute deviation across Amazon US East and EU locations. The numbers were rounded to three decimal places.

shared with other virtual machine instances and the highest performance can only be achieved when the virtual machine is underutilised.

The Rackspace 2048 MB type exhibits an interesting temporal pattern for the BIBD experiments. The performance improves after an initial period, but decreases again for short periods in regular intervals. This effect is so significant that it is most likely not caused by factors within the virtual machine instance, such as other jobs run by the operating system, but by the overall load in the data centre that houses the physical hardware. In some cases, this effect is very localised and does not only occur only for a short time period, but also affects only a subset of the virtual machine instances. This is the case for example for the Golomb Ruler experiments on the Rackspace 15872 MB type – a block of red shades in the lower middle indicates a temporary decrease in performance.

In general, the virtual machines in the Amazon cloud seem to be better isolated against differences in load on the physical infrastructure. The relative number of cases where such temporal effects occur is lower and when they occur they consistently affect all virtual machine instances for longer periods of time. This might not be because of a difference in virtualisation technologies though but merely because Amazon's data centres are larger and a much higher change in demand is required to change the load to a similar extent.

5.3 *Comparison of different locations*

When evaluating the variability across several virtual machine instances of the same type, we also have to consider the case where these instances are running in different locations or data centres. We are able to perform this evaluation for virtual machines running in the Amazon cloud by comparing the performance and variability of the same type of virtual machine in the US East and EU locations.

Table 3 shows the normalised median absolute deviations for the three virtual machine types that we evaluated in both US East and EU locations. The variability is comparable to what we measured within an individual location. The variability is significantly higher only in a few cases.

The graphs of the run time distributions in Figure 4 also show a picture that is similar to the one in Figure 2. The variation is quite large for the `m1.xlarge` type, but there are no distinguishable separate peaks in the distribution. For the `c1.xlarge` type, the range of the distribution is smaller, but at least for the Golomb Ruler problem, there are clearly two separate peaks. The `m2.4xlarge` type has the smallest variation. On close examination, two separate peaks are visible for the Golomb Ruler problem, but they are very close and do not designate a performance difference that would have a significant impact in practice.

Details for all the results are given in Figure 5. In almost all scenarios, there is a clear difference between the performance observed in the US East region and in the EU region. However, for both the `m1.xlarge` and the `c1.xlarge` type, the differences observed within a location are more significant than the differences across locations. In each case, a subset of the virtual machine instances running in one location can be distinguished more clearly from other instances in the same location than from the instances in the other location. This is especially obvious for the `m1.xlarge` type, where the full range of variation is observed in the US East location whereas all measurements in the EU location are very close to the median.

For the `m2.4xlarge` instance type, the differences across locations are more significant than the differences within a location for two out of three problems. The differences are very small however, with a range of only 3% of the median for the Golomb Ruler problem.

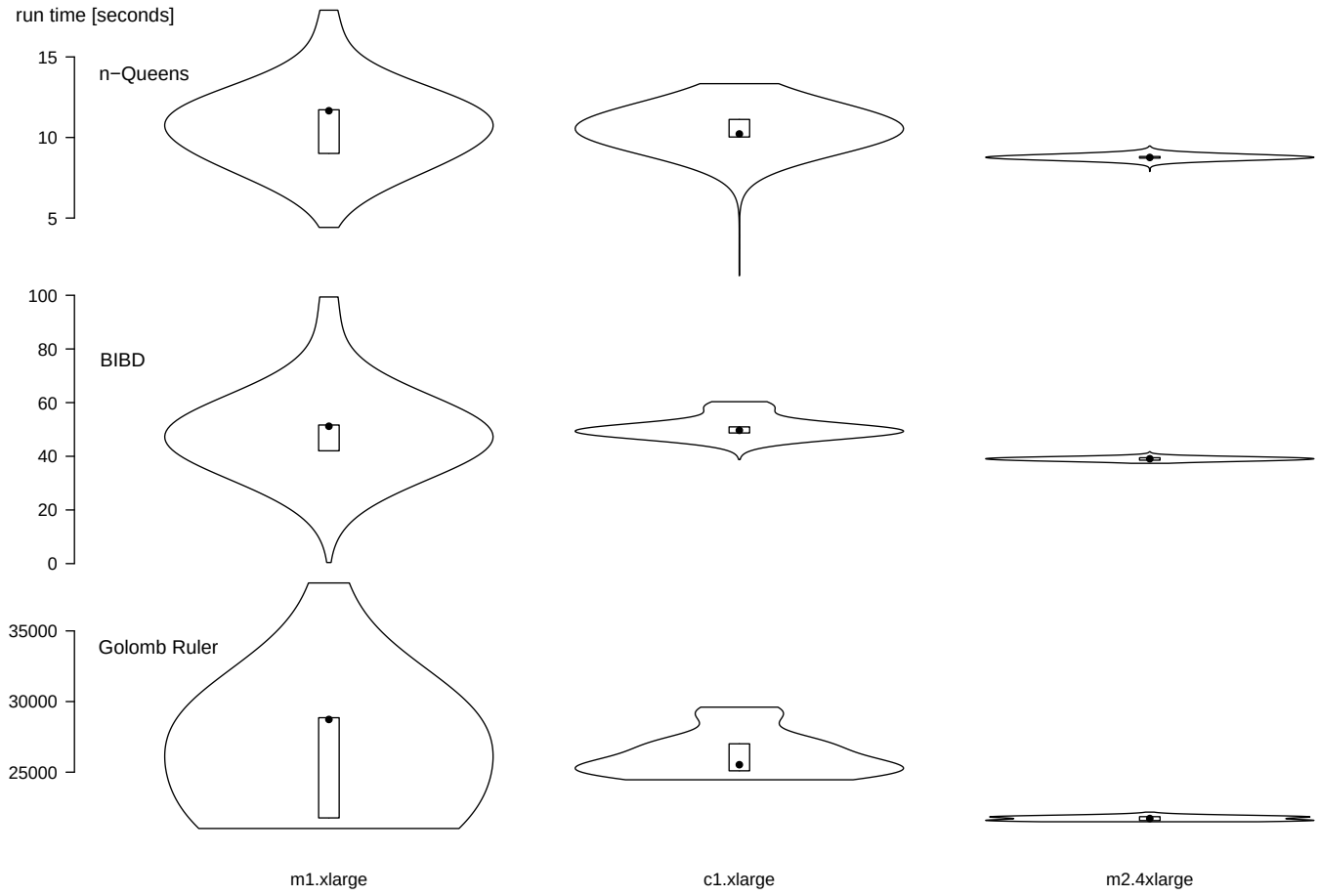

Figure 4. Distribution of run times for different virtual machine types in the Amazon cloud across US East and EU locations. The solid dot denotes the median, the bottom and top of the boxes denote the 25th and 75th percentile, respectively. The width of the shape denotes the distribution of measurements.

6 Conclusions

We have presented a detailed and comprehensive evaluation and comparison of the variability of run times for computational experiments on physical hardware and several different cloud environments. The variability of run times is crucial to the reproducibility of results, which itself is a requirement of standard scientific rigour. If reported performance improvements for an approach are less than the differences in performance to be expected for individual experiments, the results will not be able to stand up to scientific scrutiny.

The focus of this work is different from almost all previous work, which usually considers absolute performance and not reliability or variability of results. In addition, we concentrate on experiments that do not rely on high network performance, a major source of variability discovered by previous studies. This case is more common in Artificial Intelligence.

Our comprehensive experiments show that there is no intrinsic disadvantage of running on virtualised hardware over running on physical hardware. Both in terms of absolute performance and variability of reported run times, the best results were in fact not achieved on physical hardware, but in a virtualised cloud environment. This result is encouraging and we hope that it will facilitate the acceptance of cloud environments as computational resources for researchers in Artificial Intelligence.

For running reproducible experiments in public clouds, we can unreservedly recommend either the `m2.4xlarge` instance type in the Amazon cloud or both investigated instance types in the Rackspace cloud. The variation of reported run times is comparable to physical hardware and measurements never deviate more than about 10% from the median. For long-running experiments, the variation decreases even further.

The `m2.4xlarge` virtual machine type in the Amazon cloud exhibits slightly lower variation than the

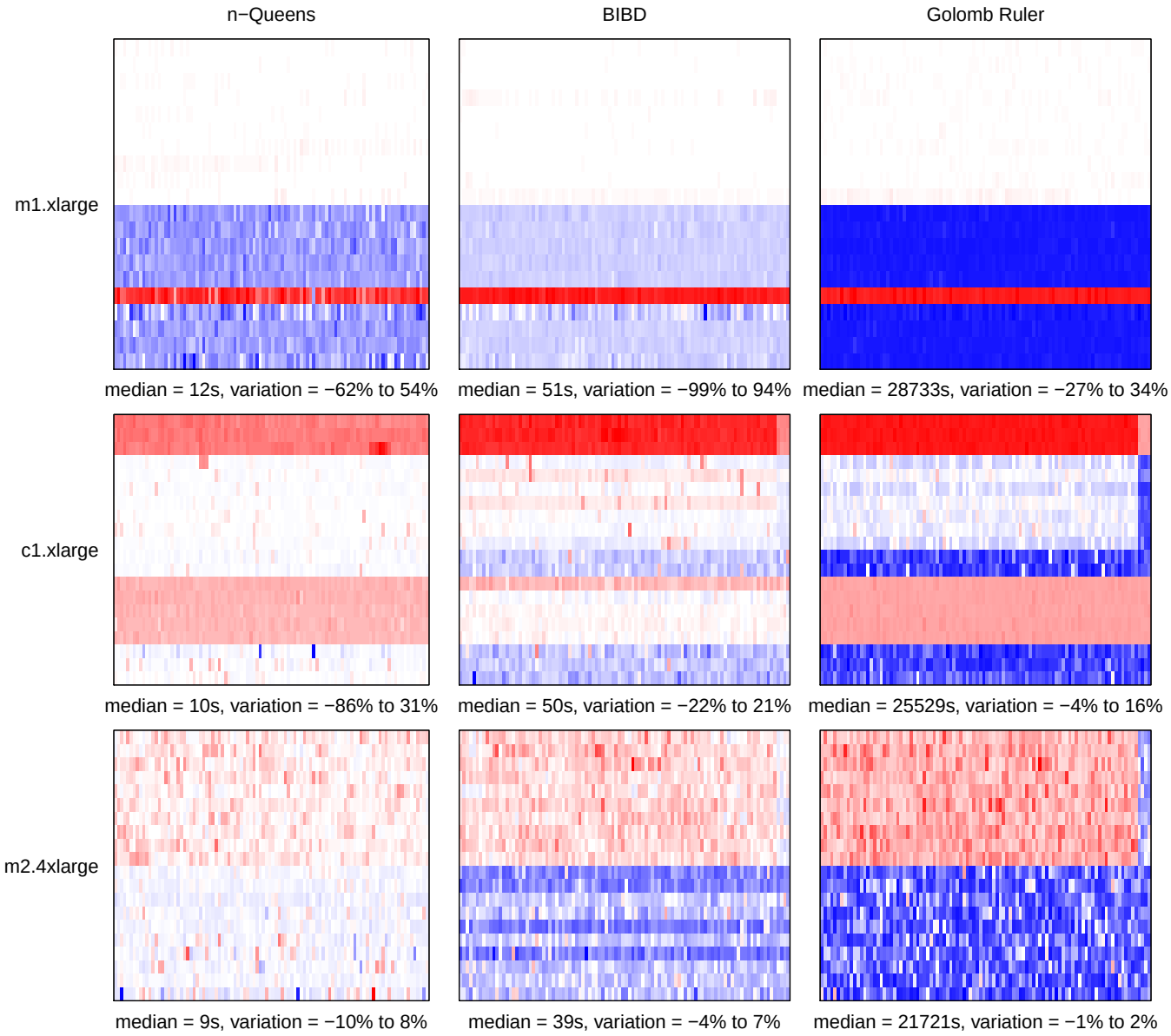

Figure 5. Variation from the median of each scenario for different virtual machine types in the Amazon cloud across US East and EU locations. Within an individual box, rows represent different virtual machines and columns different runs. The top half denotes virtual machines running in the EU region and the bottom half the US East region. White denotes the median, shades of red slower than the median and shades of blue faster than the median. Red and blue themselves represent the maximum variation for each scenario. The variation refers to percent difference from the median. Note that the top half of the boxes for the `m1.xlarge` instance type is not empty – all measurements are very close to the median and therefore the colours are very close to white.

instances in the Rackspace cloud. In terms of absolute performance, it is significantly better. In general, both performance and variability are more consistent in the Rackspace cloud however – there are huge differences between different instance types on Amazon while there are almost no differences on Rackspace.

Even the worst variation between measurements of run time that we have observed in practice might not render the virtual machine type unusable in practice if the change the researcher wants to demonstrate is orders of magnitude and the results are averaged over several repeated measurements. For improvements that are a small factor or less, the smaller virtual machine types in the Amazon cloud are not suitable because of the large variations of the reported run times.

A more serious impediment to using the cloud for computational experiments in practice are the different levels of performance different virtual machine instances of the same type can exhibit. This suggests that a set of experiments should either be run on the same virtual machine instance or, if several instances are used, care must be taken to ensure that they deliver the same level of performance. In any case, the ability

of other researchers to replicate the results would likely be diminished.

The results of the comparison of the performance across different locations suggests that the variability is not significantly higher than within a location. We were only able to compare two different locations though. If possible, it is advisable to run experiments only within a single location and indicating this in the description of the experimental methodology.

6.1 *Cost*

One of the major selling points of cloud providers is that instead of a large up-front investment in infrastructure, it can be rented as it is needed at a lower overall cost and commitment. While this is true for researchers who run occasional small-scale experiments, groups running experiments more frequently or large-scale experiments are probably better advised to make this up-front investment.

The total cost of running the experiments on the Amazon and Rackspace clouds in this paper would have been more than US-$10,000. The same amount would have been able to buy and maintain several state of the art servers to provide physical infrastructure for experiments.

Many of the experiments in the Amazon cloud used spot instances, which are usually much cheaper than the standard instances. In some cases, there were large fluctuations of the price for spot instances however. This not only diminished the advantage over regular instances, but also caused some instances to be terminated automatically and prematurely as the price increased beyond the pre-set limit.

In terms of cost, the Rackspace cloud has an advantage over the Amazon cloud as it delivers a consistently low variation even with small instance types that are cheaper than Amazon's instance types with a similar reliability. On the other hand, the Amazon cloud provides better absolute performance, both in terms of processing power and maximum memory available in an instance type. It also provides greater flexibility through more instance types and additional storage options.

6.2 *Limitations and future work*

The biggest limitation of this investigation is that we were able to evaluate the performance of virtual machines in only two of Amazon's currently seven regions and in only a single Rackspace region. In the future, we would like to extend the investigation to more locations.

There are additional opportunities to increase the scope of this study. We have focused on CPU performance, but for many applications, other factors such as disk access and network performance are also important. These issues could be addressed in a future study.

Our results represent a snapshot of the cloud infrastructures at the time the experiments were conducted, but do not inform conclusions on how this changes over time. It remains unclear whether the results reported here would be confirmed by repeating the experiments a year later.

A possible solution to the problems we have described above would be for cloud providers to remove some of the abstraction from the physical hardware and enable researchers to ask for virtual machines running on specific physical infrastructure. This, together with an option to shrink-wrap virtual machine environments, would represent a significant step forward for the reproducibility and accountability of computational scientific experiments.

Acknowledgements

This research was generously supported by a grant from Amazon Web Services and a test account provided by Rackspace. We also acknowledge support from the cloud group at the School of Computer Science of the University of St Andrews. Lars Kotthoff was supported by a Scottish Informatics and Computer Science Alliance studentship and an EPSRC Doctoral Prize fellowship.

References

David Burkett, David Hall, and Dan Klein. Optimal graph search with iterated graph cuts. In *AAAI Conference on Artificial Intelligence*, 2011.

Paul R. Cohen. *Empirical Methods for Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1995. ISBN 0-262-03225-2.

Rina Dechter. *Constraint Processing*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann, 1st edition, May 2003. ISBN 1558608907.

Yaakoub El-Khamra, Hyunjoo Kim, Shantenu Jha, and Manish Parashar. Exploring the performance fluctuations of HPC workloads on clouds. In *CloudCom*, pages 383–387, 2010.

Ian P. Gent and Toby Walsh. CSPLib: a benchmark library for constraints. Technical Report 9, APES, 1999.

Ian P. Gent, Christopher A. Jefferson, and Ian Miguel. MINION: a fast, scalable, constraint solver. In *ECAI*, pages 98–102, Amsterdam, The Netherlands, 2006. IOS Press.

Keith R. Jackson, Lavanya Ramakrishnan, Krishna Muriki, Shane Canon, Shreyas Cholia, John Shalf, Harvey J. Wasserman, and Nicholas J. Wright. Performance analysis of high performance computing applications on the amazon web services cloud. In *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 159–168, November 2010.

Jonathan Klinginsmith, Malika Mahoui, and Yuqing M. Wu. Towards reproducible eScience in the cloud. In *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 581–586, November 2011.

Younggyun Koh, Rob Knauerhase, Paul Brett, Mic Bowman, Zhihua Wen, and Calton Pu. An analysis of performance interference effects in virtual environments. In *IEEE International Symposium on Performance Analysis of Systems Software*, pages 200–209, April 2007.

Alexander Lenk, Michael Menzel, Johannes Lipsky, Stefan Tai, and Philipp Offermann. What are you paying for? performance benchmarking for Infrastructure-as-a-Service offerings. In *IEEE International Conference on Cloud Computing (CLOUD)*, pages 484–491, July 2011.

Qiang Lu, You Xu, Ruoyun Huang, Yixin Chen, and Guoliang Chen. Can cloud computing be used for planning? an initial study. In *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 1–8, November 2011.

Jeanna N. Matthews, Wenjin Hu, Madhujith Hapuarachchi, Todd Deshane, Demetrios Dimatos, Gary Hamilton, Michael McCabe, and James Owens. Quantifying the performance isolation properties of virtualization systems. In *Workshop on Experimental Computer Science*, ExpCS '07, New York, NY, USA, 2007. ACM.

Simon Ostermann, Alexandru Iosup, Nezih Yigitbasi, Radu Prodan, Thomas Fahringer, and Dick Epema. A performance analysis of EC2 cloud computing services for scientific computing. In *CloudCom*, pages 115–131, 2009.

John J. Rehr, Fernando D. Vila, Jeffrey P. Gardner, Lucas Svec, and Micah Prange. Scientific computing in the cloud. *Computing in Science Engineering*, 12(3):34–43, May 2010.

Jörg Schad, Jens Dittrich, and Jorge-Arnulfo Quiané-Ruiz. Runtime measurements in the cloud: Observing, analyzing, and reducing variance. *Proc. VLDB Endow.*, 3:460–471, 2010.

Douglas Thain, Todd Tannenbaum, and Miron Livny. Distributed computing in practice: The condor experience. *Concurrency – Practice and Experience*, 17(2-4):323–356, 2005.

Edward Walker. Benchmarking amazon EC2 for high-performance scientific computing. *USENIX Login*, 33(5):18–23, October 2008.

Guohui Wang and T. S. Eugene Ng. The impact of virtualization on network performance of amazon EC2 data center. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, March 2010.

Andrew J. Younge, Robert Henschel, James T. Brown, Gregor von Laszewski, Judy Qiu, and Geoffrey C. Fox. Analysis of virtualization technologies for high performance computing environments. In *IEEE International Conference on Cloud Computing (CLOUD)*, pages 9–16, July 2011.