Predicting Application Performance for Multi-Vendor Clouds Using Dwarf Benchmarks

Vegard Engen, Juri Papay, Stephen C. Phillips, and Michael Boniface

IT Innovation Centre, University of Southampton, SO16 7NS, U.K.

Abstract. Future Internet applications are becoming increasingly dynamic and can be composed of a wide range of services controlled and hosted by different stakeholders. This paper addresses the challenge of resource provisioning for applications that have specific Quality of Service (QoS) requirements and where consumers of Cloud resources want to avoid lock-in to any specific Infrastructure-as-a-Service (IaaS) provider. Application modelling can be used to predict performance of applications given certain resources, workload and configuration. However, application modelling is a significant challenge for Cloud consumers due to the limited and varying information IaaS providers disclose about infrastructure resources. We demonstrate in this paper how Dwarf benchmarks can be used as a uniform and informative way of characterising compute resources, which is successful for application modelling, achieving high prediction accuracy on a range of applications.

Keywords: Cloud Computing, Future Internet, Resource Estimation, Quality of Service, Application Modelling, Application Benchmarking, Dwarfs.

1 Introduction

Cloud computing offers the potential to dramatically reduce the cost of software services through the commoditisation of information technology assets and on-demand usage patterns. However, the complexity of determining Quality of Service (QoS) requirements for applications in such environments introduces significant market inefficiencies and has driven the emergence of service engineering tools for modelling, analysing and planning the QoS of service based applications deployed within the Cloud [1,2].

In this paper we address the problems of resource provisioning for Software-as-a-Service (SaaS) providers with guarantees on QoS whilst avoiding lock-in to any particular Infrastructure-as-a-Service (IaaS) provider. This is a significant challenge for applications deployed across federated Clouds as the resource offerings by different IaaS providers vary significantly. In practice, the approach to determining resources required for a particular application is often *ad hoc*, most likely requiring SaaS providers to run their application on different resources (on different IaaS providers) and observe the performance. This can be a very time consuming and costly exercise, which typically leads to relying on a single IaaS provider. Dwarf benchmarks have been proposed as a way to describe compute resources in a uniform manner across different IaaS providers whilst also being intended to be sufficiently information-rich to be used directly in application modelling [3].

This paper demonstrates how the Dwarf benchmarks can be used in application modelling to successfully predict the performance of several common multimedia and scientific applications. This use of the Dwarf benchmarks, therefore, enables transferability of service engineering tools to different IaaS providers, opening up the Cloud market and helps SaaS and PaaS providers exploit the potential for multi-vendor Clouds.

2 Background

2.1 Benchmarking Compute Resources

Measuring the performance of computers by benchmarking is a well-established activity and a large collection of benchmarks exists, such as SPEC, EEMBC, LINPACK and LAPACK. The issue with such benchmarks, which are not application-focused, is that the results can be uninformative and misleading [4,5].

Colella [6] proposed a Dwarf taxonomy for benchmarking aiming to capture known computational patterns. The Dwarf taxonomy was furthered developed at UC Berkeley [7,8], now comprising 13 Dwarfs: Finite State Machines, Combinatorial, Graph Traversal, Unstructured & Structured Grids, Dense & Sparse Matrices, Spectral, Dynamic Programming, Particles, MapReduce (Monte Carlo), Backtrack and Branch & Bound, and Graphical Models.

Initial results in [3] indicate that using this taxonomy of Dwarfs is a useful way to describe Cloud compute resources as they expose non-obvious differences in resources deemed to be the same by the IaaS provider.

2.2 Resource Estimation and Application Modelling

One of the motivations of the work discussed in this paper is helping Cloud consumers determine which IaaS provider(s) and specific resources are required to run their applications in the Cloud with particular QoS constraints. This work fits particularly well within the service engineering tools that a Platform-as-a-Service (PaaS) provider can offer as part of the wider role of helping the application provider develop, deploy and manage their application.

Application modelling can be used to predict the performance of an application given some specific resources. A generic application model takes as input a description of the expected static application *workload*, a description of the *resources* (physical or virtual) used to execute the application (including the resource *reliability*) and a description of any expected *user interactions* which contribute to the workload or otherwise affect the process [3]. Using a mathematical process, the model makes a prediction of the application performance.

We focus here on computing the core processing time of components in such a model. In related research, the work described in [9] achieved this by performing extensive benchmarking of the application on the same hardware the application would be run on. These benchmarks, therefore, could not support new hardware, nor be transferable to another IaaS provider.

3 Method

We investigate the use of Dwarf benchmark scores to characterise computational resources, which are used as input to an application model to predict the application performance. Using the Dwarf benchmark scores, we achieve a uniform description of compute resources, which we hypothesise will allow prediction of application performance on unseen resources.

3.1 Benchmark Suite

The benchmark suite we have adopted is described in [3], and therefore not all details are repeated here. The suite currently comprises eight out of the thirteen Dwarfs suggested by Asanovic et al. [8]: Structured & Unstructured Grid, MapReduce, Dense & Sparse Matrix, Graph Traversal, Particle and Spectral.

To calculate the Dwarf scores, we have used the as Phillips et al. [3]. Each Dwarf in the benchmark suite is executed multiple times to obtain a mean performance metric that is used to calculate the Dwarf score [3]. Thus, giving a numerical performance characterisation of a compute resource in the form of eight Dwarf scores.

3.2 Applications

Similarly to Phillips et al. [3] we make use of the following three applications for this investigation: Gromacs v. 4.0.7 (molecular dynamics), FFmpeg v. 0.6.2 (video transcoding) and Blender v. 2.49.2 (3D rendering).

For Gromacs, two different workloads have been chosen. One configuration uses a spherical cut-off for the electrostatic calculations and the other one uses the Particle Mesh Ewald (PME) method. We observe in [3] that these algorithms do correlate differently with the different Dwarfs, although computing an approximation of the same physical property.

The chosen FFmpeg computation is the transcoding of the "Big Buck Bunny" video [10] from M4V (h264 encoded) to OGV (libtheora encoded), and changing the frame size from 640x360 to 480x270. The sound is also changed from AAC to FLAC. As in [3], we have used Blender to render a bespoke animation small enough to process on resources constrained to 1GB RAM.

3.3 Computational Resources

We have conducted this investigation as part of an experiment in the BonFIRE project [11], which offers a multi-site testbed of heterogeneous Cloud resources across Europe for Internet of Services research. At the time, BonFIRE offered four infrastructure testbeds: EPCC, HLRS, IBBT and Inria This investigation also includes five public Cloud providers, all of which have different resource offerings and ways of describing them; Amazon EC2, Rackspace, CloudSigma (Zürich site), ElasticHosts and GoGrid.

The BonFIRE testbeds use a common labelling of *small*, *medium*, *large*, etc., which have defined number of cores and RAM size. However, the 100% of the CPU is given, which means the performance of resources with the same label can vary significantly between the testbeds due to heterogeneous hardware [3]. Amazon EC2 also operates with similar labels, but defines CPU performance in ECUs as well as the number of virtual cores and RAM size. Therefore, the performance of resources with the same ECUs should be the same even if they are heterogeneous as Amazon EC2 scale the CPU speed accordingly. This is not the case, however, as demonstrated in [3], as characterising the performance of a compute resource based on one parameter is not sufficient.

Other Cloud providers offer more fine-grained specifications of the VM instances, such as ElasticHosts, allowing you to determine exactly the virtual CPU speed, number of virtual cores, RAM size and storage space. CloudSigma also offers a similar scaling of CPU speed, and for both providers, this is a guaranteed minimum. Rackspace and GoGrid do not allow control of the CPU properties, but offer different server options that vary in RAM and disk space. Since the Dwarf benchmarks are invariant to the number of cores and RAM size [3], we effectively only make use of one resource from each of these providers.

To increase the number of data points for statistical analysis and to build better mathematical prediction models, we have also included five different physical hosts we had access to in-house. For these machines, we have executed the benchmarks and applications on a Ubuntu Maverick VM running on VMWare 4.0 with 1GB RAM. We have used an Ubuntu Maverick image on all the public Cloud providers, and in BonFIRE a Debian Squeeze image. In total, obtaining 23 unique computational resources on which benchmarks and applications have been executed.

3.4 Modelling Techniques and Validation Method

As discussed in Sect. 2.2, we focus on the challenge of calculating the core computation time and ignore the problems of varying application workload and user interactions. We have investigated several mathematical models/functions for predicting the performance of the different applications, some based on a single Dwarf, a combination of two Dwarfs and a combination of all Dwarfs.

Based on a single Dwarf, we have investigated 1^{st} to 5^{th} order functions to determine if there are any performance gains in increasing the complexity of the function. For the sake of brevity, we only report results with a 1st order (linear) function and 5th order polynomial.

Most applications will perform different types of computations and are, therefore, unlikely to be accurately modelled by just a single Dwarf. Therefore, we have investigated a linear combination of two Dwarfs to determine any gain in accuracy. The selection of Dwarfs for a given application could be done in different ways; for example, based on knowledge about the application, code profiling or according to correlation analysis. We present results for the latter here.

The final model we have investigated is the Moore-Penrose inverse matrix calculation [12], which can take as input all Dwarfs. Each mathematical prediction model is built on training data, to create a function that takes Dwarf score(s) as input and outputs application performance. To make best use of the data available, we conduct leave-one-out validation and report the mean percentage error. For each validation step (equal to the number of data points), the percentage error ε is calculated as:

$$\varepsilon = \frac{\mid m - p \mid}{m} \times 100$$

Where m is the real measured application performance and p is the predicted performance. The performance for all applications in this investigation is taken as the execution time.

4 Empirical Results

All the mathematical functions examined here are able to successfully predict the performance of all the applications using the Dwarf scores as characterisations of the compute resources. The accuracy varies with the complexity of the mathematical function, as expected. However, even with a simple linear regression based on a single Dwarf, the mean prediction error is as low as 16.72%, as seen in Table 1 (best results highlighted in bold). The improvements achieved with the 5^{th} order function are significant on all applications; as much as 13.01 percentage points on Blender.

The best results obtained with the 1^{st} order function are achieved with the Dwarfs that are correlated very highly with the application, which is to be expected. However, the lowest error achieved with the 5^{th} order function is with a different Dwarf compared with the 1^{st} order function for all but one application.

A linear combination of the two highest correlated Dwarfs can improve the prediction accuracy compared with using only one Dwarf, as seen in Table 2. However, not in all cases. The 5^{th} order function based on one Dwarf does give better results on FFmpeg and Blender (over 10 percentage points lower).

More complex combinations of two Dwarfs may yield better results still, as for the results with the Moore-Penrose inverse matrix. The mean error rates with this function is in the range 4.70% - 6.47%. These results are encouraging, especially considering the statistically low number of data points for this investigation (23 unique compute resources) and that the benchmark suite only considers computational benchmarks, which does not include six Dwarfs that represent patterns that could further improve these predictions.

Table 1. Mean percentage error of the 1^{st} and 5^{th} order functions on single Dwarfs.

	Gromacs cut-off		**Gromacs PME**		**FFmpeg**		**Blender**	
Dwarf	Linear	5th order	Linear	5th order	Linear	5th order	Linear	5th order
Structured Grid	19.60	15.54	19.10	14.84	22.08	8.63	25.07	15.44
Unstructured Grid	20.80	15.50	18.86	**13.72**	20.70	7.91	**22.24**	**9.23**
MapReduce	**17.98**	15.37	**16.72**	13.50	18.04	**7.61**	23.82	14.41
Dense Matrix	19.15	17.24	18.06	15.36	**17.93**	9.25	22.59	15.17
Sparse Matrix	18.63	**14.11**	18.57	13.85	19.47	7.48	25.19	14.73
Graph Traversal	25.85	20.65	25.37	20.73	19.80	9.71	29.23	22.17
Particle	18.36	16.00	18.76	15.74	20.16	8.07	27.68	17.43
Spectral	25.16	17.00	25.20	15.79	24.89	9.81	29.05	10.83

Table 2. Overview of prediction results (mean percentage error).

Application	1 Dwarf 1^{st}order	1 Dwarf 5^{th}order	2 Dwarfs	All Dwarfs
Gromacs cut-off	17.98	14.11	11.80	5.79
Gromacs PME	16.72	13.72	15.02	4.70
FFmpeg	17.93	7.61	22.29	5.24
Blender	22.24	9.23	23.20	6.47

5 Conclusions and Further Work

Based on an investigation in BonFIRE and five public Clouds, we have demonstrated that the characterisation of compute resources in the form of Dwarf benchmark scores is indeed successful for application modelling to predict the performance of two multimedia applications (Transcoding and rendering) and a scientific application (molecular dynamics).

Ultimately we could imagine each IaaS provider describing the performance of their resources in terms of a standard set of benchmark scores, such as the Dwarfs, or even agreeing SLAs in such terms. Alternatively, a PaaS provider may measure the performance of many IaaS providers, adding to one of the possible services that could be offered. This would avoid consumers of Cloud resources being locked in to a particular IaaS provider, which opens up the Cloud market and helps SaaS and PaaS providers exploit the potential for multi-vendor Clouds.

Further work on this would benefit from extending the benchmark suite by implementing the remaining five Dwarfs and addressing the challenge of using the Dwarfs in modelling and predicting application performance with varying workloads and taking into account the resource reliability in the Cloud from both a computational and networking perspective. Disk and memory performance are also important factors to be included in the future.

Acknowledgements. This work has been carried out in BonFIRE, an EC supported 7th Framework Programme ICT project (FP7- 257386).

References

1. Cucinotta, T., Checconi, F., Kousiouris, G., Kyriazis, D., Varvarigou, T., Mazzetti, A., Zlatev, Z., Papay, J., Boniface, M., Berger, S., Lamp, D., Voith, T., M., Stein: Virtualised e-Learning with Real-Time Guarantees on the IRMOS Platform. In: IEEE International Conference on Service Oriented Computing and Applications (SOCA). (2010)
2. Marquezan, C., Metzger, A., Pohl, K., Engen, V., Boniface, M., Phillips, S., Zlatev, Z.: Adaptive Future Internet Applications: Opportunities and Challenges for Adaptive Web Services Technology. In: Adaptive Web Services for Modular and Reusable Software Development. IGI Global (2012)
3. Philips, S., Engen, V., Papay, J.: Snow White Clouds and the Seven Dwarfs. In: IEEE International Conference and Workshops on Cloud Computing Technology and Science (CloudCom). (2011)
4. Seltzer, M., Krinsky, D., Smith, K., Zhang, X.: The case for application-specific benchmarking. In: 7th Workshop on Hot Topics in Operating Systems. (1999)
5. Zhang, X.: Application-Specific Benchmarking. PhD thesis, Engineering and Applied Sciences: Harvard University, Cambridge, Massachusetts (2001)
6. Colella, P.: Defining Software Requirements for Scientific Computing. DARPA HPCS Presentation (2004)
7. Asanovic, K., Bodik, R., Catanzaro, B., Gebis, J., Husbands, P., Keutzer, K., Patterson, D., Plishker, W., Shalf, J., Williams, S., Yelick, K.: The Landscape of Parallel Computing Research: A View from Berkeley. Technical Report UCB/EECS-2006-183, Electrical Engineering and Computer Sciences, University of California at Berkeley (2006)
8. Asanovic, K., Bodik, R., Demmel, J., Keaveny, T., Keutzer, K., Kubiatowicz, J., Morgan, N., Patterson, D., Sen, K., Wawrzynek, J., Wessel, D., Yelick, K.: A view of the parallel computing landscape. Communications of the ACM **52**(10) (Oct 2009) 56–67
9. Metzger, A., Boniface, M., Engen, V., Phillips, S., Zlatev, Z.: Towards Critical Event Monitoring, Detection and Prediction for Self-adaptive Future Internet Applications. In: 1st International Workshop on Adaptive Services for the Future Internet. (2011)
10. Blender: Big Buck Bunny. `http://www.bigbuckbunny.org`
11. Hume, A., Al-Hazami, Y., Belter, B., Campowsky, K., Carril, L., Carrozzo, G., Engen, V., García-Pérez, D., Ponsatí, J., Kűbert, R., Liang, Y., Rohr, C., Van Seghbroeck, G.: BonFIRE: A Multi-cloud Test Facility for Internet of Services Experimentation. In: 8th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities. (2012)
12. Penrose, R.: A Generalized Inverse for Matrices. In: Cambridge Philosophical Society. (1955) 406–413