Automated cloud brokerage based upon continuous real-time benchmarking

Thomas Richard Connor
Cardiff University School of Biosciences
Sir Martin Evans Building, Museum Avenue
Cardiff, UK
ConnorTR@cardiff.ac.uk

Joel Southgate
Cardiff University School of Biosciences
SouthgateJA@cardiff.ac.uk

***Abstract*—Over the last few years there has been a massive proliferation of cloud providers, all using a set of different metrics to describe the service solutions that they offer. This results in a lack of comparability within and between services that precludes end users being able to select the most appropriate service for their needs, based upon their requirements. Here we outline an automated real-time benchmarking platform that can interact with cloud brokers to automatically select the most optimal cloud service provider for a given workload, based upon up to the minute benchmarking results generated, stored, collated and compared by the platform itself. This software package could save end users and enterprises significant amounts of time and money by ensuring that they always use the most appropriate VM flavor, on the most appropriate cloud service, every time they run a workload.**

***Keywords*—cloud benchmarking; virtualisation; cloud brokerage;**

I. Introduction

With the proliferation of free and commercial public cloud services, it has become increasingly difficult for users to make a rational choice about the service that is most appropriate for them. The selection of a suitable instance (Amazon offers 38 flavors for example [1]) and a service itself (Gartner compare 15 providers in their 2015 Magic Quadrant worldwide Infrastructure as a Service report [2]) conspires to create a situation where many users are unlikely to be able to select the most appropriate service for their workload, based upon their performance and cost requirements. Nor is it a simple choice of selecting a suitable cloud and then simply loading a VM onto that system. Differing infrastructures and hypervisors reduce VM portability, while a wide range of often basic interfaces reduces the likelihood that a user will use multiple cloud services to process their workloads. In answer to these requirements an increasing number of products from vendors such as Dell [3] and HP [4] as well as open source projects such as CloudBroker [5] seek to provide a layer that interacts with multiple local and external cloud provider APIs in order to provision VMs on a range of cloud services based upon user defined rules (e.g. price), with minimal interaction from the user.

II. Cloud Background

The concept of offering services derived from a shared computational resource has grown over the past 5 years into a multi-billion dollar industry. Cloud-based – Infrastructure and Software as a Service – offerings ranging from Dropbox [6] to AWS [1] are now firmly entrenched in both the consumer and business world. For many organizations, large efficiency savings can be achieved through pooling computational resources and utilizing virtualization technologies to efficiently pack VMs across a newly shared IT estate [7]. In addition to these private clouds, many of the worlds largest technology companies have built upon their own considerable infrastructure to develop public clouds. These public clouds offer the promise of lower cost, superior reliability, and less administration overhead than would be required by deploying and managing infrastructure/software locally. Within the last 18 months, software packages are beginning to become available that promise the best of both worlds ; hybrid approaches that can utilize a shared local resource for most activities, with the capacity to "burst" into the public cloud for particular workloads or at particular times, in response to business need [8].

III. Statement of the Problem

However, despite the development of brokerage solutions for controlling the provisioning of VMs across multiple clouds (both public and private), these offerings remain basic, and do not take into account the fact that there is no *a priori* way to link the specifications of a given cloud provider flavor to an expected level of performance by which that service can be compared with others. In the first instance what is required is a point of comparison between services that will allow a naïve user to identify the most suitable service for their workload. However, one-off benchmarking of cloud infrastructures that, by their nature, are frequently and silently updated would not provide a suitable basis for comparison. Therefore, what is required is a system that can provide real-time or semi-real-time data collected from multiple providers simultaneously in order to identify the "best" performing service for a given workload. In addition to this, simply knowing the "best" service for a given task is not likely to be of use if, for each workload, the User has to manually instantiate different VMs on different clouds. Therefore, for most users what is required is a system that in addition to benchmarking cloud performance automatically in real time, is also able to make scheduling decisions based upon that information, automatically provisioning the VM for the user, with a minimum of interaction. By storing historical data it would also be possible to study temporal and historical trends, in order to ensure optimal placement of VMs onto commercial clouds, taking into account variations in upgrade schedule and stability of VMs over time. Finally, for some workloads it may be the case that a cloud service is not appropriate and the performance hit by using a cloud is sufficient that a user is much better off running this workload locally, the system should be able to recognize this and provide an indication to the user of this fact. This paper describes an idea to integrate an automated benchmarking system with a cloud broker to ensure that the most optimal choice is made when scheduling VMs in the cloud. This idea is illustrated with a number of preliminary results of a benchmarking exercise conducted across a number of clouds, before outlining a solution that seeks to automate the process of benchmark collection and brokerage based on the stored benchmarks.

IV. Preliminary Data and Solution Proposed

A. Benchmarking

Undertaking a basic benchmarking exercise (Figure 1) comparing two commercial cloud providers reveals the difficulty inherent in comparing VM flavors within and between providers. Comparing the two entry level 2 core VMs from Microsoft [9] and Google [10] (Azure A2 and Google n1-standard-2 respectively) reveals a large difference in performance; with the Google instance being considerably faster in every benchmark, despite costing less (~$57/month compared to £69/month for the A2 image). Increasing the performance by scaling up to instances that run either on faster processors (Azure A8) or have more RAM (Google n1-standard-highmem) also demonstrates the care that needs to be taken when selecting an image. The A8 instance offers considerably more performance (at greatly increased cost - £1114/month), while the n1-standard-highmem instance offers no added performance. In the case of the n1-standard-highmem node, the cost of overestimating the RAM requirement for a job would result in a cost of ~$20/month ; which although considerably less than the cost of scaling from an A2 to an A8 image on Azure, could still result in considerable cost increases in production environments that make heavy use of cloud images.

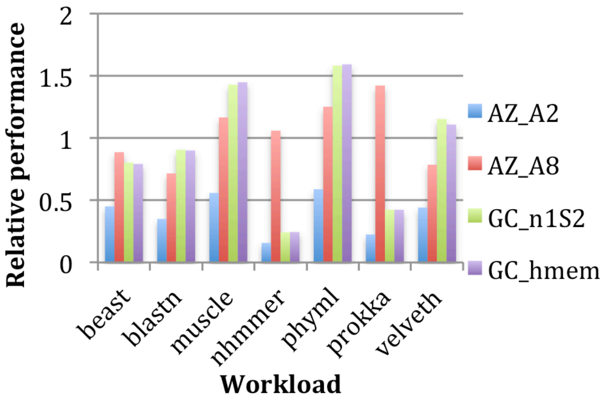

Figure 1 showing the relative performance between bioinformatics software workloads on two flavors of VMs on Google Cloud and Azure. Performance is scaled such at 1.0 for each benchmark is the performance recorded by the Cardiff University “Raven” system [11], running Intel Ivy Bridge processors

B. Value Measurement

Critical within the calculation of cost is the calculation of value. There are a large number of confounders that could affect the value proposition of the systems benchmarked above. For example, while an Azure A8 node may offer the highest performance, for many workloads, even the limited benchmarking here demonstrates that it has a disproportionately high cost. Given that A8 nodes also have enhanced connectivity, it may be that for some workloads where high speed, low latency interconnects are required, the A8 would represent best value. This requires extensive benchmarking across a wide range of use cases in order to correctly benchmark the cloud systems that would be compared.

C. Continuous Monitoring

An additional consideration is the fact that cloud systems often represent extremely large installs of hardware, and as a result may be subject to upgrades over time that will result in differences in performance between regions or providers. Additional to this is the fact that due to infrastructure and usage the stability offered by different cloud services may also vary over time. To date there is little public understanding of how cloud VM service varies depending on factors such as the time of day when a workload is run on a given VM, and this would need to be captured as part of the implementation. The solution is to enable the benchmarking software to interact with cloud APIs in order to spin up VMs and perform continuous monitoring of cloud system performance.

Continuous monitoring allows for the detection of time effects, providing a real time traffic report for each cloud service, while the frequent spinning up/down of VMs also allows for the detection of upgrades to infrastructure as soon as they happen; ensuring the best system is always picked by the system.

D. System Outline

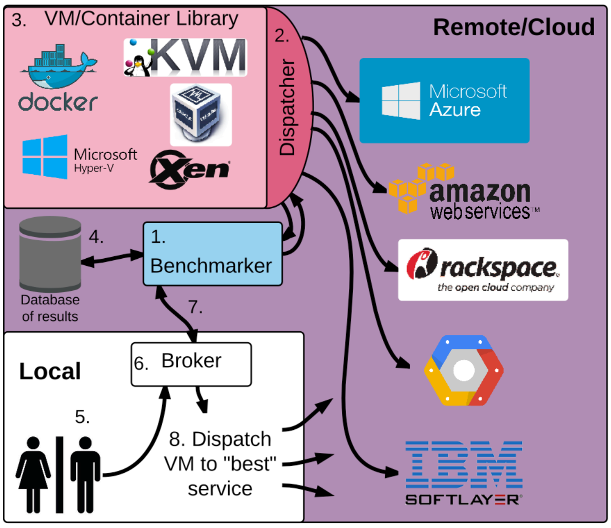

Figure 2. Schematic overview of system. Benchmarker (1.) continuously collects data from cloud services by calling a dispatcher (2.) that selects a suitable VM/container (3) and spins this up on relevant clouds. For each benchmarker VM/container the benchmarker records and stores the benchmarking results in a database (4.). When a user wants to run a workload (5.) the broker (6) queries the benchmarker (7.) to establish the best value service. Having established the most appropriate service for the workload, the broker then dispatches the VM/container to this service (8).

Figure 2 displays an overall schematic for the operation of the system. The user interacts directly with a broker, which would typically be run locally/separately from the benchmarker. The rationale for separating these two elements is simple; running benchmarks locally could rapidly become expensive for a single organization; while running benchmarks centrally and serving out the information to many organizations would be far more cost effective. Additionally, separating the broker/benchmarker means that many different providers could make use of the benchmarker via a common API, Having the user interact with a broker directly also offers a number of additional advantages. Firstly, while the benchmarker VMs (3) can be specifically created for each cloud service, user VMs may need converting before dispatch – something the broker can handle automatically. Secondly, the broker can provide a simple interface with which the user can interact with, which will also point the user at the newly running VM. Finally, and importantly for many organizations, brokers also commonly track usage for accountancy purposes.

E. Metrics to be Measured

Lastly it is important to consider the metrics that are to be gathered, stored and used by the benchmarker software. Some basic metrics such as the wall time, the system time and the user time per job can trivially be gathered as software is run, and these metrics already provide a clear basis for the computation of relative performance and value between clouds. Advantageously, these metrics can be gathered without affecting performance, and can potentially provide a performance benchmark for a given workload on a given flavor. Within a basic implementation these values would provide a simple means of comparison between systems, providing users with an expectation of relative performance. The addition of profiling tools for performance analysis would provide a natural extension to this, and would have the potential to enable the accurate prediction of performance on a given cloud provider based upon a particular workload with known requirements. In the first instance we focus on basic metrics – wall time, system time and user time – with the intention of the addition of extra metrics planned in the near future to provide a mechanism for more detailed prediction of cloud performance, and to trace performance bottlenecks on a given cloud service. Without more extensive benchmarking it is not possible to accurately predict what the key metrics are, however, over time, analysis of the collected benchmarking data should provide a clear indication of the key metrics for determining the optimal service for placement of a given VM/workload.

V. Demonstration of System

We implement a demonstration version of the system that will be accessed via a website. By the time the system is demonstrated it will have already amassed a number of automatically collected benchmarks, performed over several months prior to the demonstration. The website will allow the results to be visualized, and will also enable cloud services to be recommended based upon the specification of likely job requirements by the user. The recommendation of a given service will provide an indication of the value of the proposed service over other competitors, demonstrating the potential utility of the system for ensuring best-value for multiple workloads. Additionally, the system will make use of the biocloudcentral [12] VM launching software to enable a user to launch a VM on the most appropriate cloud for a limited number of demonstration VMs/workloads. The system will allow the selection of a pre-defined a VM/workload, and following the selection, the system will then determine the best cloud and flavor for that VM/workload. Finally the system will instantiate a suitable VM to undertake the analysis, providing the user with the required details to directly access and use that VM. This demonstration will emphasize the extent to which automated benchmarking could be incorporated into cloud brokerage. Collectively these demonstrations will provide an indication of the potential of the software, and the way in which a similar system might be expected to interact with a naïve user.

VI. Discussion

A number of solutions already exist to facilitate the provisioning of VMs onto multiple clouds, based upon a simplified criterion such as cost. The process of developing a layer such as that provided by CloudBroker that operates across multiple clouds is not trivial in of itself, and so at present it is, perhaps, not surprising that the schedulers/brokers that exist use relatively simple methodologies for determining the "best" provider for a given workload. The proposal here is a novel addition to this existing paradigm, outlining a system that provides a more nuanced mechanism for the selection of the "best" provider for a given workload, based upon user needs (cost, uptime, performance), that are all informed by an up to date archive of benchmarks that cover multiple cloud providers. The combination of real time benchmarking and a historical database opens up the opportunity for users to develop and define their own criteria for cloud selection, in addition to measures of performance per unit price that will automatically be calculated by the system for a given workload requirements. A system that is able to make scheduling decisions based upon performance information is of considerable potential utility to users of any size, and would have the potential to save cloud users significant sums of money, as well as simplifying the process of cloud service selection. The capacity outlined in this paper is not yet available within other products, and, by providing this as a central service, brokers from multiple organizations will be able to make use of the system in order to inform their scheduling decisions.

VII. Future Potential Additions

The focus of the outlined system here is predominantly around VMs, building upon the preliminary benchmarking exercise we have already carried out. However, the work could trivially be extended to two other areas that would also be of value. Firstly, the system could also perform benchmarking using containers. Within figure 2 we include Docker containers along with VMs as an indicative object that could be stored within the benchmarker library, and the addition of containers would provide an added layer of utility that could be utilized by broker software. Secondly we could also explicitly consider data within the benchmarking process. The costs of data storage and data transfer are often unclear for many cloud providers, and the performance of different storage options are also often difficult to compare. Therefore an additional element that could be considered within the scope of the benchmarker software is the benchmarking of tasks dependent upon long-term file storage, to inform both the choice of VM/container placement, and for tasks that require the longer-term storage of data in the cloud. Additionally we could consider the addition of more detailed software performance profiling in addition to the "real world" metrics that are already gathered by the system. Being able to identify likely performance bottlenecks on any given system could usefully inform cloud service choice, while the measurement of aspects such as the number of I/Os in addition to storage requirements for a given workload could also be used to inform VM placement; as this is another aspect of the charging model of some cloud providers that could add considerable cost to a VM. Finally the system could also integrate other aspects in order to suggest systems that provide the best fit for a given service. Marketplaces such as the UK Government G-Cloud provide a place for providers to offer services that in addition to cost, can be compared based upon aspects such as the security/importance level of data that the cloud is authorized to hold. Integrating aspects such as measures of data security, data auditing and legal/policy aspects within the benchmarker could provide a further aspect of added value that could also be of wide utility and impact.

References

[1] Amazon Web Services EC2 Pricing, as of September 2015, https://aws.amazon.com/ec2/pricing/

[2] L. Leong, D. Toombs, and B. Gill, "Magic Quadrant for Cloud Infrastructure as a Service, Worldwide" Gartner, May 2015.

[3] Dell Cloud Manager, http://software.dell.com/products/cloud-manager/

[4] HP Hybrid Cloud Management, http://www8.hp.com/uk/en/software-solutions/hybrid-cloud-management/

[5] CloudBroker, http://cloudbroker.com/

[6] Dropbox ; http://www.dropbox.com

[7] M.F. Bari *et al.* "Data Center Network Virtualization: A Survey", Communications Surveys & Tutorials, IEEE (Volume:15 , Issue: 2), September 2012

[8] B. Sotomayor, R.S. Montero, I.M. Llorente and I. Foster "Virtual Infrastructure Management in Private and Hybrid Clouds" Internet computing (Volume:13 , Issue: 5), September 2009

[9] Microsoft Azure, http://azure.microsoft.com

[10] Google Cloud, https://cloud.google.com/

[11] Introduction to Raven, http://www.cardiff.ac.uk/arcca/services/equipment/ravenintroduction.html

[12] N. Goonasekera, Biocloudcentral, https://github.com/nuwang/biocloudcentral

Acknowledgements

The generation of preliminary data included in this paper was funded by a Cardiff University CUROP placement. We also acknowledge the help/advice from Dr Christine Kitchen and Professor Martyn Guest of the Advanced Research Computing @ Cardiff HPC center and use of their system "Raven" to provide a baseline to benchmark the cloud flavors examined.