Association for Information Systems
AIS Electronic Library (AISeL)

SAIS 2015 Proceedings

Southern (SAIS)

2015

High Performance Computing: Considerations When Deciding To Rent Or Buy

Russell Thackston
Georgia Southern University, rthackston@georgiasouthern.edu

Ryan Fortenberry
Georgia Southern University, rfortenberry@georgiasouthern.edu

HIGH PERFORMANCE COMPUTING: CONSIDERATIONS WHEN DECIDING TO RENT OR BUY

Russell Thackston
Georgia Southern University
rthackston@georgiasouthern.edu

Ryan Fortenberry
Georgia Southern University
rfortenberry@georgiasouthern.edu

ABSTRACT

The commercial cloud computing (CCC) industry has reached a level of maturity to make it a truly viable alternative to the traditional, in-house data center. Although there are many notable examples of CCC platforms and technologies being piloted for high performance computing (HPC) tasks, it has yet to enter the mainstream. A variety of obstacles exist which have slowed or hindered adoption of CCC platforms, including implementation complexity, cost confusion, and security concerns. This paper describes the author's experiences in using CCC for various HPC tasks and compares the results to the same tasks being executed on in-house computing resources. The breakpoint at which CCC becomes more costly than in-house equipment is identified. Lastly, a series of "lessons learned" are presented to assist researchers in effectively interacting with CCC vendors and platforms.

KEYWORDS

Cloud computing, high performance computing, costs

INTRODUCTION

A series of complex, quantum chemical computations were performed using Amazon's AWS cloud computing platform. These same computations were also conducted using a high-performance computing cluster operated by Georgia Southern University's chemistry department. The results were compared based on time and monetary costs, as well as system complexity. Based on the data gathered, various breakpoints can be identified at which the Amazon platform becomes more expensive to operate than the traditional, in-house computing cluster. In addition, various lessons were learned that may aid in choosing and interacting with the cloud computing platforms and will certainly enhance future work in this area.

METHODOLOGY

The software, hardware, and computations used for this research are widely available, well known by the quantum chemistry discipline, and highly reproducible.

Software

PSI4 is among the most well-known, open source computational software packages available to quantum chemists (Turney et al., 2012). The software may be freely downloaded as source code from the http://www.psicode.org and runs on most Linux operating systems. PSI4 relies on a math library conforming to the basic linear algebra subprogram (BLAS) API. The GotoBLAS2 v1.13 library was selected for its availability and existing experience compiling it on Linux. GotoBLAS2, in turn, relies on a linear algebra package (LAPACK) for solving linear equations; LAPACK v3.4.2 was selected for its known compatibility with GotoBLAS2. Multithreading was implemented via GotoBLAS2, rather than PSI4, for the best performance and to alleviate shortcomings within the parallelization for the version of PSI4 utilized (Beta 0.5).

Hardware

Because Amazon's AWS computing instances may be run on a variety of commodity hardware, Amazon has adopted the concept of a compute unit, which roughly corresponds to a "1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor." The AWS platform provides the customer the ability to select from a variety of predefined hardware configurations, with various amounts of memory, computing power, and storage. Table 1 lists the hardware configurations used in this research and their associated per hour costs.

Georgia Southern University's chemistry department houses a cluster of Dell R420 servers each with two, 16-core Intel XeonE5-2470 2.30GHz processors and 256 GB of RAM. The original purchase price for each server was $6,055.

Cost Models

This research is not prescriptive with respect to calculating actual in-house costs and, therefore, deliberately utilizes a simplified costing model based solely on a comparison of the billed costs of the cloud resources consumed and the hardware purchase costs of the in-house equipment. As a result, the in-house solution costs are under-reported due to the fact that a variety of costs have been omitted: electricity, HVAC, building maintenance, IT support personnel, insurance, licensing, etc. As a result, this research provides a starting point for cost comparisons upon which a consumer may build their own cost models, using data provided by their specific organization.

	Cores/vCPUs (GHz)	Memory (GB/GiB)	Per Hour Cost[1*]
AWS Medium	1	3.75	0.07
AWS Large	2	7.5	0.14
AWS XLarge	4	15	0.28
AWS 2XLarge	8	30	0.56
AWS 4XLarge	16	30	0.84
GSU HPC server	32	256	Varies[2]

Table 1 Hardware configurations

Computations

The computational method used in this research is known as "coupled cluster theory" and is considered the "gold standard" of modern quantum chemistry (Shavitt and Bartlett, 2009; Crawford and Schaefer, 2000). The computations are easily repeatable on any sufficiently capable platform and are well known throughout the computational chemistry discipline. The computations cover a wide variety of runtimes and utilized a wide spectrum of computing resources.

In total, 252 computations were performed for a total of more than 934 hours of computing time. The time required for the various computations to complete ranged from three seconds to fifty-one hours. Multithreaded operations ranged from one to sixteen cores or, in the case of AWS, virtual CPUs (vCPUs).

The computations are highly CPU-intensive and utilize a relatively large amount of "scratch" space on the local drive to complete certain mathematical calculations (e.g. numerical integrations). Contrary to the extensive amount of work performed, the required input file and generated output files are of trivial size (i.e. measured in kilobytes).

CLOUD BILLING

CCC vendors generally charge for their services following an à la carte approach, with each product or service billed as a separate line item. For example, AWS charges separately for the server instance, bandwidth used, and storage consumed, each with their own metric (i.e. hours, GB, etc.).

The only major AWS cost incurred during this research, besides the in-house HPC server, was compute time, which Amazon calculates based on per hour usage, rounded up to full hours. Therefore, the only monetary cost considered in this comparison is the per hour charges.

PERFORMANCE

In terms of computing power, the in-house system used in this research was most closely matched by Amazon's 4XLarge platform. Figure 1 shows a sample calculation, "EOM-CCSD/6-31G," used in the research. In most instances, the in-house system completed the computation slightly slower or slightly faster than the 4XLarge system. A full list of the jobs, with completion times, is available for download as an Excel spreadsheet or PDF at http://centers.georgiasouthern.edu/ccrl.

A detailed comparison of the calculations run on both the AWS 4XLarge platform and the in-house system shows that in 70% of the instances, the runtimes for the two systems were within 18% of each other. In 91% of the instances, performance was within a 31% delta. Notably, a majority of the high delta values originated from a single, highly complex calculation, with the in-house system performing consistently faster than the 4XLarge platform. However, in less complex calculations, the 4XLarge platform consistently outperformed the in-house system.

[1] AWS per hour cost as of December 29, 2014, in Amazon's U.S. East region via http://aws.amazon.com/ec2/pricing/.

[2] Varies based on lifetime utilization of the server.

In some instances, the lower end AWS platforms were unable to complete certain calculations, due to either disk or memory limitations. This was largely expected, as AWS system memory and disk space are fairly minimal when compared to typical in-house computing platforms. Amazon typically expects large disk requirements to be fulfilled via their slower, network-based storage solution marketed as "elastic block store (EBS)." Such storage is not suitably fast for the disk-intensive operation used by the math libraries in this research.

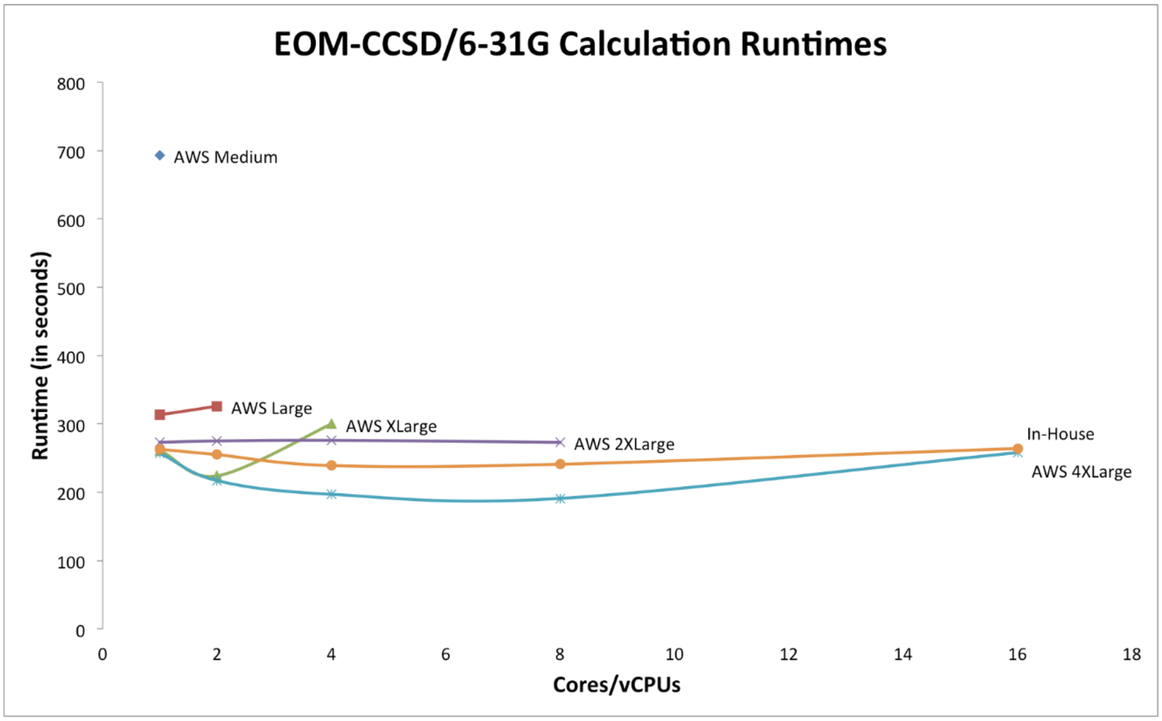

Figure 1 Example Computational Runtimes

COMPUTATIONAL COSTS

Computational costs may be evaluated as monetary or time costs, with different calculations for each platform. The monetary costs reflect either the out-of-pocket expense used to purchase computing equipment or the amount paid to a CCC vendor. On the other hand, time costs are reflective of the time spent setting up a computing environment, managing computational jobs, and waiting for jobs to run to completion.

Monetary Costs

AWS monetary costs may be easily calculated based on the per hour cost of the chosen platform and the time the job takes to complete. The total number of hours the AWS server was active – including boot time, runtime, and shutdown time – is rounded up to the nearest whole number and multiplied by the per hour billing rate of that platform.

For this research, job costs for in-house computing equipment are based on the original cost of the equipment and its utilization over its lifetime. For example, if an HPC server – purchased at a cost of $6,055 – runs 1,000 jobs over its lifetime, per job cost will be approximately $6.06, not counting overhead costs. On the other hand, if the same server completes 100,000 jobs over its lifetime, per job cost will be approximately $0.06. For a more robust comparison, a consumer should seek to incorporate all other overhead costs into the calculation, such as electricity and maintenance, which will result in a higher per job cost of in-house equipment.

Since the calculations being performed over the lifetime of an in-house server are likely to vary significantly in duration, a more useful monetary cost metric may be derived based on the server's lifetime utilization; the original cost of the server is divided by the total number of "job hours" completed by the server. For example, if the server completes an average of 2,000 hours of computations each year, its lifetime utilization will be 10,000 hours, resulting in a per hour cost of approximately $0.61. Theoretically, the maximum utilization for a server over a five-year period is an unrealistic 43,800 hours (i.e. non-stop). Therefore, the lowest possible per-hour cost for a $6,055 server is approximately $0.14.

Time Costs

Time costs may be quantified in terms of wait time or opportunity costs. Wait time reflects the amount of time spent waiting for a computation to complete. Wait time is a function of server capacity; the higher the capacity – in terms of number of servers, processor speed, memory, etc. – the shorter the wait time. Opportunity costs reflect the cost incurred when a particular calculation is not run because the computing equipment is busy with other tasks.

AWS time costs are reflective of the platform selected or the number of instances utilized. For example, a higher capacity platform – such as the AWS c4.4xlarge platform – will reduce the wait time for a single calculation but will increase the per hour monetary costs. If multiple calculations are being performed, the wait time may be reduced by using a higher capacity platform, by running each calculation on a separate instance, or through a combination of both.

In-house computing resources time costs are defined by the original equipment purchase. The computing capacity and number of servers purchased defines the number of calculations that may be performed and the speed at which they complete.

RESULTS

Logically speaking, AWS job cost is a function of job duration with a linear growth trend: the longer it takes to complete a computation, the higher the monetary cost.

Figure 2 shows the actual AWS costs incurred during this research, limited to jobs less than 12 hours for clarity. For comparison, Figure 3 shows the per-job costs for the in-house server based on the number of jobs completed over the server's lifetime.

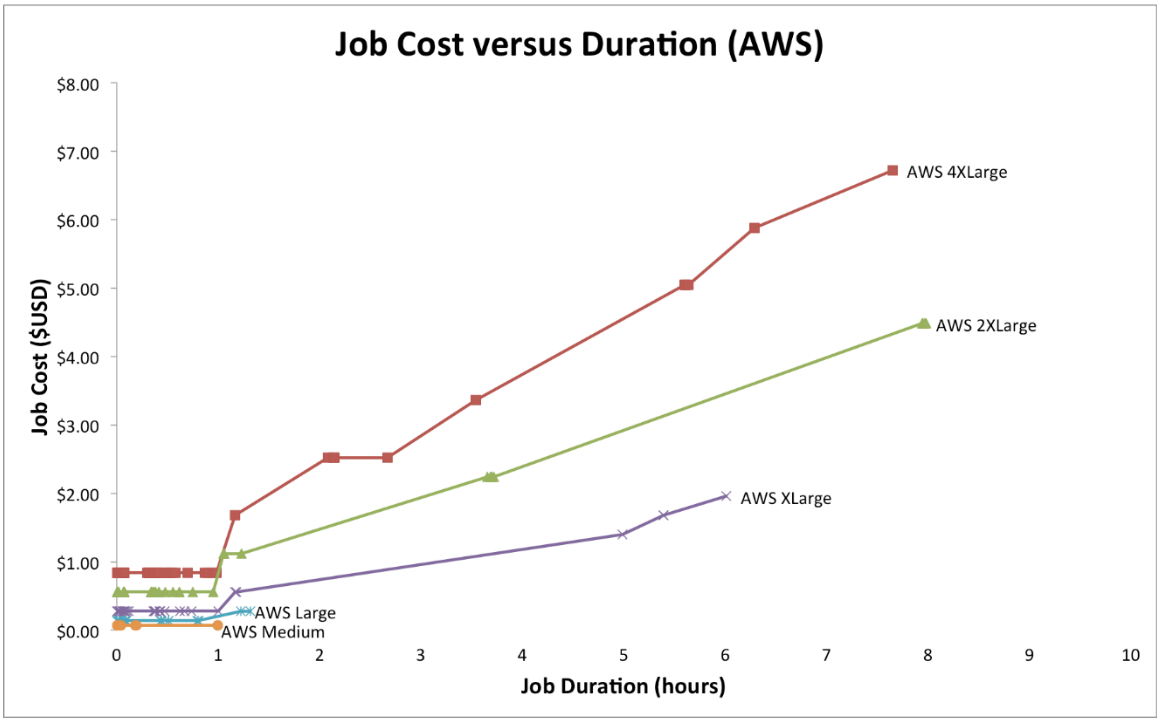

Figure 2 Job Cost versus Duration for Amazon AWS Jobs

Given these pricing scenarios, a comparison of AWS costs to in-house costs will be based on a series of questions about the expected usage of the system in question. What is the total budget for running the calculations? What types of calculations (by size) will be run on the target system? How many of each type will be run during the system's lifetime? What level of urgency is associated with each job? Can multiple calculations be run in parallel?

For example, if a researcher's total computing budget were $1,000 per year for five years, they would be unable to purchase a suitable in-house HPC system; in this case, a cloud platform would be a logical alternative. The researcher could purchase anywhere from 1,200 to 14,000 hours each year from Amazon, depending on the capabilities on the selected platform. If the calculations varied in size, a suitable platform could be selected for each job to balance completion time against its monetary cost. Furthermore, if each of the researcher's calculations were not dependent upon the results of the other calculations, then all the calculations could be completed at once using a cloud platform and multiple HPC server instances.

In an alternate example, a researcher may have a one-time budget opportunity of $6,055. This amount could purchase a reasonably capable HPC system for in-house use. If the system were highly utilized over its five-year lifetime – a total of 30,000 hours or 24 hours a day, five days a week – the job cost would be approximately $0.20/hour. This is the equivalent of a mere 4,762 hours of compute time on Amazon's 4XLarge platform, which performs roughly the same as the in-house server used in this research.

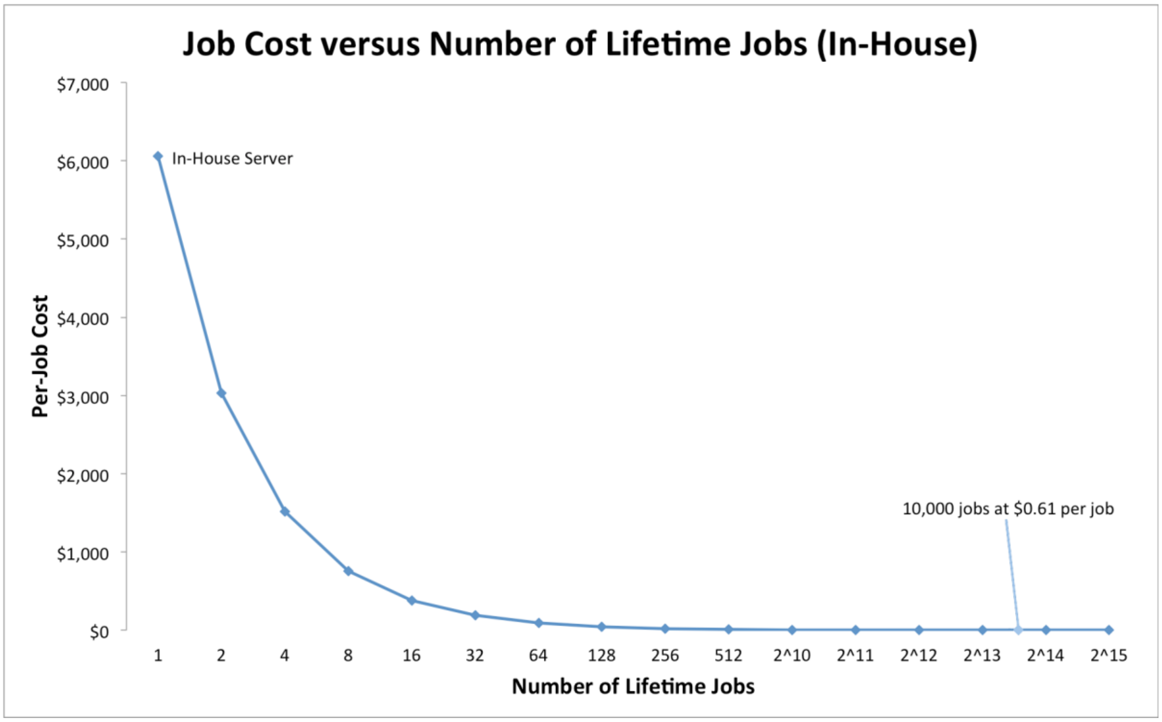

Figure 3 Job Cost versus Duration for In-House Computing Equipment

LESSONS LEARNED

In addition to the knowledge gained with respect to cost decision-making for cloud platforms, this research revealed some important lessons with regard to provisioning and managing cloud platforms in a computational setting.

Automation

First and foremost, cloud vendors charge for usage based on the total time the resources are in use. Therefore, it behooves the CCC consumer to develop methods for minimizing the amount of time a cloud instance is actively provisioned. In the case of this research, that translated to various automation tools and techniques for job creation, execution, monitoring, and termination. Furthermore, it led to the creation of a custom server image with all necessary software and scripts preinstalled, which could be instantiated with a few mouse clicks via Amazon's management console. As the project progressed, the researchers were able to minimize the amount of time an instance remained underutilized.

In a closely related lesson, the researchers developed a queuing system for jobs to further minimize slack time on the instance, especially for smaller jobs that took only a few minutes. This resulted in significant cost savings when a series of jobs could be completed within Amazon's one-hour increment billing policy.

Multithreading

In some instances multithreading reduced performance, especially at eight or sixteen cores. This is likely due to the overhead required to manage breaking the calculation into smaller parts for parallelization. Therefore, it is clear that "more equals faster" is not a safe assumption.

Steal Time

Any virtualized environment, such as AWS, will lead to a level of steal time. Steal time is the amount of time the CPU spends working on any task other than yours. The visible representation of steal time is the difference between reported processing time for a job and wall clock time. In any virtualized environment, the reported processing time – divided by the number of processors used – will always be less than the actual time taken. Such time is used to process the requests of other cloud customers with instances running on the same physical hardware.

While steal time cannot be fully eliminated, it can be monitored and managed. The Linux operating system offers several utilities – top, mpstat, /usr/bin/time, etc. – that can look for and estimate steal time. When unacceptably high levels of steal time are noticed, the customer can take action. For example, moving jobs to virtual machines in different vendor regions may result in lower steal time. Additionally, running jobs at different times of day may improve performance. However, since a customer's view of a CCC platform is more restrictive than that of in-house equipment, the available options will be limited.

CONCLUSION

This research analyzes the metrics for a series of quantum chemistry calculations recognized as standard computations for the discipline. These calculations were performed using Amazon's CCC platform and a typical in-house high performance computing system. The results of the analysis – a series of cost considerations and lessons learned - may be applied to any discipline considering cloud computing as an alternative to purchasing an in-house computing system.

REFERENCES

1. T. D. Crawford and H. F. Schaefer III, in Reviews in Computational Chemistry, Vol. 14, edited by K. B. Lipkowitz and D. B. Boyd (Wiley, New York, 2000) pp. 33–136.
2. Shavitt and R. J. Bartlett, Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory (Cambridge University Press, Cambridge, 2009).
3. "Psi4: An open-source ab initio electronic structure program," J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, N. J. Russ, M. L. Leininger, C. L. Janssen, E. T. Seidl, W. D. Allen, H. F. Schaefer, R. A. King, E. F. Valeev, C. D. Sherrill, and T. D. Crawford, WIREs Comput. Mol. Sci. 2, 556 (2012). (doi: 10.1002/wcms.93).