March 9, 2015

The Performance of Low-Cost Commercial Cloud Computing as an Alternative in Computational Chemistry

Russell F. Thackston
Ryan C. Fortenberry, *Georgia Southern University*

The Performance of Low-Cost Commercial Cloud Computing as an Alternative in Computational Chemistry

Russell Thackston[*], Ryan C. Fortenberry[†]

February 16, 2015

Abstract

The growth of commercial cloud computing (CCC) as a viable means of computational infrastructure is largely unexplored for the purposes of quantum chemistry. In this work, the PSI4 suite of computational chemistry programs is installed on five different types of Amazon World Services CCC platforms. The performance for a set of electronically excited state single-point energies is compared between these CCC platforms and typical, "in-house" physical machines. Further considerations are made for the number of cores or virtual CPUs (vCPUs, for the CCC platforms), but no considerations are made for full parallelization of the program (even though parallelization of the BLAS library is implemented), complete HPC utilization, or steal time. Even with this most pessimistic view of the computations, CCC resources are shown to be more cost effective for significant numbers of typical quantum chemistry computations. Large numbers of large computations are still best utilized by more traditional means, but smaller-scale research may be more effectively undertaken through CCC services.

Keywords: cloud computing, PSI4, hardware performance, computational optimization ■

1 Introduction

It is no secret that the cost of computer hardware significantly hampers the ease of entry for new and emerging researchers in computational chemistry. With state-of-the-art computing blades running in excess of $5,000 (US) each, an adequate high-performance computing cluster can cost more than what many small or even medium-sized universities are able to provide. The most immediate alternative is to rely on older machines that are often too small for real research. Computer time can be granted by remotely accessed supercomputing facilities. Scientific endeavors that require substantial computational power are useful

[*]Department of Information Technology, Georgia Southern University, Statesboro, Georgia 30460-8150, U.S.A, rthackston@georgiasouthern.edu

[†]Department of Chemistry, Georgia Southern University, Statesboro, Georgia 30460-8064, U.S.A, rfortenberry@georgiasouthern.edu

applications of such, but it limits the researcher to stay within the scope of the grant and cannot be utilized as a "sand box" for one to explore other avenues of research as pilot or exploratory studies.

An alternative approach to the "in-house" high-performance computing cluster (HPC) or supercomputer allocation approaches is commercial cloud computing (CCC). Several companies who need a major interface with the public rely on large, well-maintained server facilities. However, the maximum needs of a facility are rarely commensurate with the average needs. As such, companies like Amazon.com grant access to their servers for the paying customer. This approach marries the needs of the researcher to have access to quality computer hardware with a relative minimum of cost. Granted, in order to utilize adequate computational power, some cost will need to be absorbed by the researcher, but this cost is hypothesized here to be far less than the purchase of comparable hardware. Additionally, the hardware available on these CCC facilities is constantly being updated and will be close to or at the cutting-edge. On the contrary, a purchased HPC becomes obsolete very shortly after purchase. In the past, the best approach has probably been an amalgamation of in-house with supercomputer solutions, but the emergence of CCC and its role in expanding this paradigm has not been fully explored.

Some studies have applied and analyzed the use of CCC for applications to computational chemistry. Mohammed and coworkers[1] have demonstrated that proteomic mass spectrometric data can be analyzed efficiently with cloud-based solutions. In so doing, CCC resources greatly decrease the amount of time it takes to fully characterize various biochemicals analyzed in the laboratory since thousands of processors can be utilized at once for the same monetary cost as would be incurred by utilizing one processor thousands of times. Additionally, Wong and Goscinski[2] compare the performance of different cloud environments specifically for computational chemistry. They find that CCC can reduce the financial and temporal cost of scientific computing as long as the user can navigate the implementation of software distributed across many different types of systems or if the necessary software image has already been created. As a result, CCC can be an effective computational chemistry tool.[2]

Buyya and coworkers[3] define cloud computing as "a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service provider and consumers." In practical terms, CCC vendors provide clients with a variety of fee-based, on-demand computing resources, but Hardware as a Service (HaaS)[4] is the only means that is practical, at the moment, for accessing pre-built disk images with quantum chemical software pre-installed. In this work, the performance of CCC computational hardware is evaluated based on time and monetary costs in order to determine the cost-effectiveness of purchasing an HPC or simply running computations via CCC services. This research focuses on the system as a whole (CPU, memory, disk) due to the the fact that the consumer has little to no knowledge of the underlying hardware and no ability to select hardware combinations beyond the "standard" ones provided. The PSI4[5] quantum chemistry suite is a free, open-source computational chemistry package that performs high-level quantum chemical computations. As such, a variable in the determination of the operating costs within computational chemistry is immediately lowered. Three of the largest CCC vendors – Amazon, Google, and

Table 1: Specifications for the Computing Equipment Utilized.

Name	Cores/vCPUs	Processor Speed (GHz)	Memory (GB)
Surplus	1-2	3.16	8
Dedicated	1-16	2.3	256
AWS Medium	1	3.0 (3 ECUs)	3.75
AWS Large	1-2	6.5 (6.5 ECUs)	7.5
AWS XLarge	1-4	13 (13 ECUs)	15
AWS 4XLarge	1-16	55 (55 ECUs)	30

Microsoft – offer HaaS in the form of virtual computers. For this project, the Amazon AWS platform is selected to host the initial tests, as it offers the widest variety of operating system and hardware platforms among the HaaS options.

2 Methodology

2.1 Test Platforms and Metrics

This research utilizes three general categories of computing equipment that may commonly be used to run PSI4: *Surplus*, *Dedicated*, and *CCC*. The *surplus* category represents functional, yet relatively outdated, computing equipment readily available at most large organizations. In this case, the desktop computer is a Lenovo machine with an Intel Core2 Duo E8500 CPU with a maximum speed of 3.16 GHz. Next, well-funded research labs typically purchase *Dedicated* computers or clusters, which are installed and configured for the specific purpose of providing researchers an HPC environment for their computations. This class of platforms is a single Dell R420 machine with two, 16-core Intel Xeon E5-2470 2.30GHz processors purchased through start-up funds provided to one of the authors (RCF). Lastly, PSI4 may be installed on virtual computers using a *CCC* platform, with computing resources purchased "on demand." Table 2.1 lists the platforms selected for testing along with their technical specifications.

The number of processors is measured either in *cores* or *vCPUs*. The terms *cores* refers to multi-core processors, such as dual-core Pentiums. The term *vCPU* is Amazon-specific and refers to the number of virtual processors an instance has access to for multithreading. Processor speed is measured in GHz for the Surplus and Dedicated platforms. Because virtual servers on a CCC platform may run on a variety of hardware with varying processor speeds, Amazon has created the concept of a "compute unit," which roughly corresponds to a "1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor".

RedHat Linux is the chosen operating system, as it is readily available on all the test platforms. The public Beta 0.5 version of PSI4 is utilized for the performance test, with LAPACK (v3.4.2) and GotoBLAS2 (v1.13) as the supporting libraries. It should be noted that the GotoBLAS2 library is built on each system for the Intel Nehalem architecture for the sake of consistency and due to a lack of transparency as to what the AWS hardware may actually be. Hence, perceived inefficiencies in the computations are likely the result of older BLAS technology implemented in new hardware.

The Linux operating system provides the utility *time*, which measures wall clock, user, and system times for a particular program's execution. The *time* utility is used track and compare wall clock, user, and system time for each job execution. This is particularly important, given that virtual computers may only receive a fraction of the total CPU time of a physical computer, creating significant disparities between wall clock time and system time. *Wall clock time* is the most granular measure of execution time. It measures the total time spent by the CPU processing instructions for a particular program, such as PSI4, and will be the time unit utilized and discussed. For simplicity, this research reports per hour costs based on actual billed amounts and based on the fee schedule provided. In addition, each job's individual costs are reported as constrained by the provider's minimum charges (i.e. 1 hour).

2.2 Computational Details

Coupled cluster theory[6,7] is considered to be the "gold standard"[8,9] of modern quantum chemistry. The set of correlation consistent basis sets[10–12] are also standard use. Single-point energies are the most common type of computation undertaken in quantum chemistry. These can be strung together to create numerical derivatives producing geometry optimizations or vibrational frequencies[13,14] or left to stand alone as is the case for the computation of electronically excited states. The equation of motion (EOM) formalism[15,16] coupled to CCSD is a standard means of computing energy differences producing theoretical electronic spectra. Electronically excited states often represent a challenge to modern quantum chemical tools due to the inherent complexities of the constructing the ground and subsequent excited state wavefunctions.[17] Additionally, coupled cluster computations are among the most expensive class available in quantum chemistry. Hence, this type of quantum chemical computation will benchmark an upper-bound for CCC efficiency and cost that should be reduced with less exspensive methods like density-functional theory.

The computations chosen for this study represent a complete subset related to the electronic properties of closed-shell quinoline and isoquinoline deprotonated anion derivatives of interest for a current study of dipole-bound excited states of anions.[18–20] Hence, "real-world" research sample computations are employed for this study. Restricted reference wave function[21] EOM-CCSD prediction of the 2 $^1A'$ state of the quinoline anion deprotonated at the meta position[20] are employed using the 3-21G,[22] cc-pVDZ, aug-cc-pVDZ, and d-aug-cc-pVDZ basis sets with the latter three abbreviated as pVDZ, apVDZ, and dapVDZ in Table 2. These vertically excited state (XS) computations increase the number of basis functions (102, 170, 284, and 398, respectively) in order to highlight how the sizes of single-point energy computations, with excitation energies as an example, affect the computational cost as well as the choice and monetary costs of potential computational platforms. The PSI4 DCFT6 test case is also examined since it is run during the build of the standard PSI4 release and can serve as a benchmark for other users' AWS instances. These computations are analyzed utilizing different numbers of processors, as well, in order to find the most efficient combination. In all, over 100 different computations have been performed on the various test platforms to provide our data set.

Even though a parallel version of PSI4 is available, it has, in the utilized version, limited implementation, and the simplest way to employ multithreading within PSI4 is simply to allow the BLAS library to run in parallel while the main PSI4 executable is nominally running in serial. The GotoBLAS2 library can utilize up to 16 threads at a time giving a range of cores to test from 1 (serial) all the way to 16. The procedure is effective since a majority of the computations done in PSI4 employ D_GEMM, D_AXPY, and similar BLAS calls. The major exception is the first computation of the two-electron integrals which cannot make use of the multithreaded BLAS library and are not coded in parallel for the public release (Beta 0.5) version of PSI4 implemented for this study. While this is a drawback for CCC efficiency, it is a necessary factor in order to employ this free and open-source software.

The actual installation of the software proceeds through standard RedHat installation of PSI4, but CCC platforms are generally constructed using commodity hardware, making it difficult or impossible to know exactly what type of processor underlies a particular instance. This is particularly a problem with PSI4, which is distributed as source code and must be compiled on the target platform; the process of compiling typically involves specifying the type of processor in order to properly optimize the software. Here, the Intel Nehalem architecture is used for installation of the program since the dependent BLAS and LAPACK libraries are backwards-compatible with this hardware regardless of what the actual processor is. Hence, if one assumes the virtual computer as a stand-alone machine (due to the use of HaaS implementation) and the processor architecture can be generalized adequately, the installation process is the same as it would be on a standard, in-house computer.

3 Results

3.1 Time Costs

Table 2 contains a listing of the runtimes for our five test computations across an exponential sampling for the number of available cores/vCPUs for each computation. These raw numbers highlight the difference in the performance for the various CCC options compared to the standard Dedicated blade and Surplus box. Those columns for which there are no data indicate that these computations could not run on these systems due to storage or memory limitations or errors within the BLAS library for the number of cores requested. Again, these values do not represent the most efficient implementation of the software (especially for the dedicated servers) but are consistently constructed across all of the tested systems in order to minimize variables.

The AWS Medium platforms posts the slowest (worst) results in all non-trivial computations (i.e. computations other than DCFT6). The performance of the Surplus platform consistently falls between the AWS Medium and AWS Large platforms in all non-trivial computations. In about 10% of the cases, computations take longer when moving to a more powerful AWS instance for the same computation/vCPU combination. For example, when using only a single vCPU, the AWS 4XLarge platform takes 8% longer than the AWS 2XLarge platform with 69,586 seconds and 64,254 seconds, respectively.

As Figure 1 demonstrates, the relative performance between platforms for the 3-21G computations is largely expected with performance improving as the computing power increases.

Figure 1: A graph demonstrating the performance differences across platforms and cores/vCPUs for the EOM-CCSD/3-21G computation.

Figure 2: A graph demonstrating the performance differences across platforms and cores/vCPUs for the EOM-CCSD/cc-pVDZ computation.

The Surplus platform produces a performance gain moving from one to two cores. The Large and 2XLarge performance curves remain relatively flat as additional vCPUs are added. The XLarge platform gives an immediate performance gain from one to two vCPUs, but reaches a point of diminishing returns when moving to four vCPUs. The 4XLarge platform sees performance improvements from one to two to four vCPUs; eight vCPUs demonstrates no performance gain. Sixteen vCPUs actually increases the total time. The reasoning for this is largely attributable to the small number of basis functions where the parallelization intrinsic to the BLAS library is of little consequence compared to the mandatory serial bottleneck in the formulation of the integrals. Hence, only the single-processor speed and efficiency brings about any change in the timings of these relatively small single-point energy computations.

The Surplus platform improves its performance for the pVDZ computations moving from one to two cores (Figure 2), whereas the Large platform does not see any performance improvement moving from one to two vCPUs. The XLarge, 2XLarge, and 4XLarge platforms decrease their computational times, but this leads to diminishing returns at four, eight, and sixteen vCPUs, respectively. Performance for the Dedicated platform flattens out between eight and sixteen cores.

When increasing the number of basis functions from 170 to 248 for the EOM-CCSD/apVDZ computation, shown in Figure 3, the Surplus platform gives a significant performance improvement moving from one to two cores. The XLarge platform reaches a point of diminishing returns at four vCPUs. The 2XLarge platform's performance remains flat, regardless of the number of vCPUs. The 4XLarge and Dedicated performance curves see modest improvements between one and eight cores/vCPUs, before flattening out. Note that the Large platform is unable to complete the computation due to "out of memory" errors; the Medium platform is not attempted due to the failure on the Large platform.

The 2XLarge platform's EOM-CCSD/dapVDZ performance remains flat in Figure 4, regardless of the number of vCPUs, while the 4XLarge performance curve has modest improvements between one and eight vCPUs before reaching a point of diminishing returns with sixteen vCPUs. The Dedicated performance curve gives modest improvements between one and four vCPUs, before flattening out. Unfortunately, the Large and XLarge platforms are unable to complete the computation due to insufficient disk space for the necessary scratch files required to treat 398 basis functions; the Medium platform is not attempted due to the failure on the Large and XLarge platforms. Additionally, the $n = 8$ and $n = 4$ Dedicated

Figure 3: A graph demonstrating the performance differences across platforms and cores/vCPUs for the EOM-CCSD/aug-cc-pVDZ computation.

Figure 4: A graph demonstrating the performance differences across platforms and cores/vCPUs for the EOM-CCSD/d-aug-cc-pVDZ computation.

Figure 5: A graph demonstrating the performance differences across platforms and cores/vCPUs for the DCFT6 test.

and 2XLarge, respectively, jobs could not complete due to issues within the compiled BLAS libraries at these number of cores for these memory requirements.

As Figure 5 demonstrates, the DCFT6 computations are extremely small tests with highly variable results. With the exception of the Dedicated platform, these jobs complete in less than two minutes. The Dedicated platform timings approximately double each time additional cores are added due to the system administration costs overshadowing the actual computational speed-up. The Dedicated platform timings are not shown in Figure 5 as the values are too large for a meaningful visual comparison. The overhead associated with adding the additional processors far outweighs the time required to perform the actual calculation itself.

3.2 Monetary Costs

Logically speaking, the cost per job for the AWS platforms will be fixed, regardless of how many of jobs are performed. On the other hand, the Dedicated platform has a single lifetime cost of \$6,055, resulting in a per job cost that reduces as more jobs are run over the computer's lifetime. Note, all AWS costs have been calculated based on Amazon's one hour minimum billing rate. While it is possible to execute a series of jobs back-to-back and realize a significant cost savings, this research assumes the most pessimistic view of usage: any job that takes less than one hour will be billed a full hour. This approach accounts for the likely possibility that a researcher is running jobs on an as-needed basis, analyzing the results, then running additional jobs. Within the AWS platform, such behavior would result in maximum billing amounts taking effect.

Table 3 lists the point at which the per job cost for the Dedicated platform intersects with the fixed per job cost of the AWS platforms. To illustrate, Figure 6 depicts the Dedicated platform's per job cost – the dashed line – reducing as more jobs are run over the lifetime of the computer. The solid lines – representing the various AWS platforms – show fixed per job costs ranging from \$0.20 to just under \$1.00, depending on the type of platform selected. For reference, the Surplus platform – with a fixed cost of \$0.00 – is shown as a dotted line. Note the graph includes a small inset in the top-right corner to provide perspective for an x-axis beginning at zero. This trend is qualitatively the same for each test case, but the more time – which is largely related to the number of basis functions – is required to complete a given computation, the smaller the breakpoint becomes.

At the breakpoint, neither platform provides a cost benefit over the other. For example, the lines for the Dedicated and AWS 4XLarge platforms meet at 6,242 jobs as given in Table 3. At this point, both platforms have a per job cost of \$0.97. If a researcher purchases a Dedicated server for our \$6,055 purchase price and runs exactly 6,242 N1H2 XS 3-21G-type

Figure 6: A graph demonstrating the monetary cost per job across platforms for the EOM-CCSD/3-21G computation based on the number of similar jobs being run at the same time. The inset in the top-right corner is included to demonstrate context from zero.

(102 basis function single-point XS energy) computations over the server's lifetime, the total cost will be equal to that of running the same jobs on the AWS platform's 4XLarge platform. The only major difference between the two platforms would be the "wall clock" time required to run the computations, since the AWS platform would allow the researcher to run a very large number of the jobs concurrently if desired or feasible to do so.

Additionally, Table 3 clearly indicates that the larger the job (the more basis functions), the faster the breakpoint is reached. Furthermore, the more expensive the CCC platform, the faster the breakpoint is reached, as well. It will take an incredibly large number of small jobs to justify the purchase of a Dedicated-like blade server, more than can be conceived and completed over the course of a research project. However, larger jobs require a significant monetary investment regardless of whether the researcher elects to use the standard HPC or even CCC. If more than 480 EOM-CCSD/dapVDZ-sized computations are required, the HPC or Dedicated-like server is a more financially responsible decision if the time required to run those computations is also not a major consideration.

4 Conclusions

The data clearly demonstrate that additional computing power – cores/vCPUs, memory, disk space – does not automatically produce faster results. Eventually, the overhead associated with multithreading leads to flat performance or diminishing returns. Hence, single-point energy computations with 250 or less basis functions most often are most efficient utilizing 4 vCPUs while larger computations can benefit from 8 but not 16. Dedicated servers are most efficient with 8 cores in this installation of serial PSI4 with the parallel BLAS library for the older and less efficient but consistent Nehalem build option. It should be noted that while CCC platforms can match stand-alone hardware, significant computations may still require Dedicated-like systems or supercomputer allocations for the highest efficiency. In the extreme example of the 398 basis function EOM-CCSD/dapVDZ computation, the Dedicated server is over twice as fast in its completion time for the same job.

In deciding whether to rent CCC resources or purchase dedicated servers, researchers must consider the these two primary costs: *time* and *money* – and determine how to strike a balance between the two. More powerful computers allow for faster computations, but are expensive to leave idle. Logically speaking, a Dedicated-like server must be utilized over a certain percentage to justify the upfront cost. For example, Dedicated and 4XLarge platforms have the same per job cost for an N1H2 XS 3-21G-type computation at 6,242 jobs. Theoretically, the AWS platform could complete all 6,424 jobs in just under 3 minutes (see Table 3). On the other hand, a dedicated server should run the jobs sequentially in our pessimistic view. Additionally, the speed-up for using more cores is not linear, especially for jobs that require significant memory/disk. Since the Dedicated platforms takes approximately 234 seconds to complete a single instance of this type of computation, the total execution time would be approximately seventeen days. Therefore, the out-of-pocket costs for both platforms is the same after running 6,424 N1H2 XS 3-21G-type computations. However, purchasing a dedicated server provides the researcher with an additional 4.95 years of compute time.

Table 4 lists the Dedicated platform's utilization, in years, at each per job cost breakpoint. For all computations used in this research, the Dedicated platform is capable of reaching the price breakpoint within the server's lifetime. This demonstrates that any heavily-used Dedicated-like platform is more cost-effective than the alternative Amazon platforms. However, if hardware is not going to be used on significant numbers of significantly-sized computations, CCC resources are more cost-effective. Larger-scale research will probably still require in-house HPC resources and/or supercomputer time. However, the size of the project or even the size of the substituent parts of the project determines the platform to utilize. The stratification of those considerations across platforms is provided here for a researcher's own personal application.

The PSI4 implementation used in this research relies on parallelization of the BLAS library, rather than of the PSI4 program itself. Additionally, this work does not utilize full-HPC architectures whether in the cloud or in-house. Finally, we do not analyze *steal time* – CPU time spent working on other virtual user's jobs – to determine its impact on computation times and monetary costs. As such, we have provided the most pessimistic view of CCC-resources and compared it to typical, non-virtualized physical hardware functioning as a single compute unit. We also make no consideration for electricity, physical storage, or cooling for the Dedicated server. Even so, the CCC resources still demonstrate more financial efficiency (i.e. "bang-for-the-buck") over the typical Dedicated server for typical computational chemistry problems. Hence, even though in-house HPCs or blade servers are available for use until they physically break down, CCC resources may even be more cost-effective than our data indicates for research groups, especially at small and underfunded universities. Surplus machines are by far the most cost-effective means of computational chemistry research, but they are largely inadequate for cutting-edge research. In any case, the researcher must utilize the necessary resources to accomplish his or her vision.

5 Acknowledgements

Both RT and RCF are grateful for the funding provided by Georgia Southern University necessary to perform this investigation. The authors also acknowledge Prof. C. David Sherrill of the Georgia Institute of Technology for very useful discussions and insights.

Table 2: Job Execution Times in Seconds

	cores/ vCPUs	N1H2 XS 3-21G	N1H2 XS pVDZ	N1H2 XS apVDZ	N1H2 XS dapVDZ	DCFT6
AWS Medium	1	653				137
AWS Large	1	287	1,835			75
	2	281	1,836			96
AWS XLarge	1	298	1,701	21,639		95
	2	203	1,379	17,950		102
	4	271	1,517	19,405		132
AWS 2XLarge	1	251	1,715	13,319	64,254	70
	2	252	1,381	13,375	64,502	67
	4	252	1,212	13,161		68
	8	251	1,356	13,275	63,499	68
AWS 4XLarge	1	241	1,711	12,742	69,586	67
	2	197	1,490	9,616	55,561	120
	4	177	1,355	7,711	51,254	103
	8	178	1,076	7,502	45,617	64
	16	233	1,343	7,724	68,232	143
Dedicated	1	291	1,567	12,689	48,200	69
	2	249	1,525	11,450	38,179	874
	4	234	1,341	9,712	31,274	1,448
	8	237	1,191	8,728		2,747
	16	218	1,199	8,773	28,070	5,753
Surplus	2	359	3,652	35,698	158,355	311

Table 3: Per Job Cost Breakpoints in Number of Like Jobs

	N1H2 XS 3-21G/ DCFT6	N1H2 XS pVDZ	N1H2 XS apVDZ	N1H2 XS dapVDZ
AWS Medium	46,577			
AWS Large	30,275	30,275		
AWS XLarge	17,809	17,809	3,562	
AWS 2XLarge	8,775	8,775	2,194	
AWS 4XLarge	6,242	6,242	2,081	480

References

[1] Y. Mohammed, E. Mostovenko, A. A. Henneman, R. J. Marissen, A. M. Deelder, and M. Palmblad, J. Proteome Res. **11**, 5101 (2012).

[2] A. K. L. Wong and A. M. Goscinski, Future Gener. Comp. Sy. **29**, 1333 (2013).

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, Future Gener. Comp. Sy. **25**, 599 (2009).

Table 4: Dedicated Platform Utilization in Years

	DCFT6	N1H2 XS 3-21G	N1H2 XS pVDZ	N1H2 XS apVDZ	N1H2 XS dapVDZ
AWS Medium	0.10	0.3	-	-	-
AWS Large	0.07	0.2	1.1	-	-
AWS XLarge	0.04	0.1	0.7	1.0	-
AWS 2XLarge	0.02	0.1	0.3	0.6	-
AWS 4XLarge	0.01	0.0	0.2	0.6	0.5

[4] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, SIGCOMM Comput. Commun. Rev. **39**, 50 (2008).

[5] J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. A. 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 III, R. A. King, E. F. Valeev, C. D. Sherrill, and T. D. Crawford, Wiley Interdisciplinary Reviews: Computational Molecular Science **2**, 556 (2012).

[6] I. Shavitt and R. J. Bartlett, *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory* (Cambridge University Press, Cambridge, 2009).

[7] 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.

[8] T. J. Lee and G. E. Scuseria, in *Quantum Mechanical Electronic Structure Calculations with Chemical Accuracy*, edited by S. R. Langhoff (Kluwer Academic Publishers, Dordrecht, 1995) pp. 47–108.

[9] T. Helgaker, T. A. Ruden, P. Jørgensen, J. Olsen, and W. Klopper, J. Phys. Org. Chem. **17**, 913 (2004).

[10] K. A. Peterson and T. H. Dunning, J. Chem. Phys. **102**, 2032 (1995).

[11] R. A. Kendall, T. H. Dunning, and R. J. Harrison, J. Chem. Phys. **96**, 6796 (1992).

[12] T. H. Dunning, J. Chem. Phys. **90**, 1007 (1989).

[13] X. Huang and T. J. Lee, J. Chem. Phys. **129**, 044312 (2008).

[14] R. C. Fortenberry, X. Huang, A. Yachmenev, W. Thiel, and T. J. Lee, Chem. Phys. Lett. **574**, 1 (2013).

[15] J. F. Stanton and R. J. Bartlett, J. Chem. Phys. **98**, 7029 (1993).

[16] A. I. Krylov, Ann. Rev. Phys. Chem. **59**, 433 (2007).

[17] R. C. Fortenberry, R. A. King, J. F. Stanton, and T. D. Crawford, J. Chem. Phys. **132**, 144303 (2010).

[18] J. Simons, J. Phys. Chem. A. **112**, 6401 (2008).

[19] R. C. Fortenberry and T. D. Crawford, J. Chem. Phys. **134**, 154304 (2011).

[20] M. L. Theis, A. Candian, A. G. G. M. Tielens, T. J. Lee, and R. C. Fortenberry, (2015), *in prep.*.

[21] A. C. Scheiner, G. E. Scuseria, J. E. Rice, T. J. Lee, and H. F. Schaefer III, J. Chem. Phys. **87**, 5361 (1987).

[22] W. J. Hehre, R. Ditchfeld, and J. A. Pople, J. Chem. Phys. **56**, 2257 (1972).