MIGRATION AND PERFORMANCE OF CMAQ & WRF-CMAQ IN THE PUBLIC CLOUD WITH COMMERCIAL IMAGES

Arturo Fernandez*
Odyhpc, Murrysville, PA, USA

1. INTRODUCTION

Most present air quality predictions run on on-premises hardware with only a small fraction of current predictions being generated in cloud environments. However, shifting a larger fraction of these computations to the public cloud is desirable as cloud environments accelerate the adoption of new hardware and provide a framework to integrate computations, data assimilation and high-performance storage with nearly infinite capacity. Cloud Service Providers (CSPs) provide computational and storage options in the form of Infrastructure-as-a-Service (IaaS). The extent of these services can be augmented or lowered to fit fluctuating workloads or storage needs without being tied up to a monolithic configuration. From an economic perspective, the use of cloud resources eliminates the large upfront capital needed to purchase expensive equipment. This upfront capital and how to finance it are traditionally two of the major challenges for small and medium-sized businesses to access HPC capabilities. By circumventing this cycle, smaller organizations gain access to resources traditionally reserved to large businesses, national labs, and research universities. Two additional advantages from using IaaS are advanced security features and the reduction in maintenance costs as most of these tasks are performed by CSPs' technical teams.

Since the launch of AWS by Amazon in 2006, several works including Yelick et al. (2011), Mohammadi and Bazhirov (2018), Chang et al. (2018) and Fernandez (2021) have examined HPC performance in the public cloud with the aid of benchmarks such as HPL (Dongarra et al., 2003) and HPCG (Dongarra et al., 2013) or the NASA Parallel Benchmark (Bailey et al., 1991). However, benchmarks only provide partial insights into performance and a more through quantification requires benchmarking the apps themselves. For example, Powers et al. (2021) have examined the performance of WRF in AWS IaaS. However, quantifications of CMAQ performance are still at an earlier stage.

This extended abstract present benchmarks for CMAQ and WRF-CMAQ in AWS and Azure IaaS using commercial images available to any organization or individual with valid accounts.

2. RUNNING CMAQ & WRF-CMAQ

Commercial images include precompiled executables for the apps and their dependencies, along with pre and postprocessing tools. The CMAQ images are available from the AWS and Azure Marketplaces. After subscribing, the user can launch instances following regular procedures. Fig. 1 shows how AWS launches an instance running CMAQ.

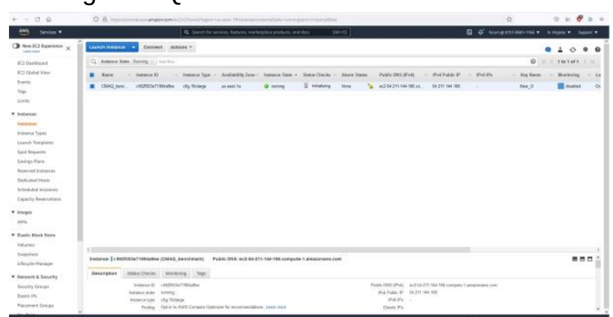


Fig. 1. AWS console showing the launch of an instance.

Once the launch is complete, the user can log in using a ssh terminal like any other Linux-based instance. The available executables include CMAQ, CMAQ- ISAM, CMAQ- DDM3D, and WRF-CMAQ and postprocessing tools. Fig. 2 shows how the CMAQ benchmark runs from a terminal.

Fig. 2. Running CMAQ from a putty (ssh) terminal.

In AWS, it is also possible to launch clusters using AWS-Parallelcluster. These clusters consist of a head instance, acting as the manager, plus one or more compute instances performing the computations. Parallelcluster also facilitates launching an elastic cluster where the number of instances is variable depending on the workloads. The scheduler is slurm so launching jobs proceeds similarly to supercomputers.

*Corresponding autor: Arturo Fernandez, odyhpc Murrysville, PA 15668; e-mail: afernandez@odyhpc.com, https://odyhpc.com.

3. PERFORMANCE EVALUATION

The present evaluation measures performance of 2 different apps in AWS and Azure IaaS: CMAQ and WRF-CMAQ. Table 1 lists the main characteristics of the hardware used to perform the benchmarks including instance family name, maximum number of cores and memory for the largest instance of that family, processor name and type/architecture of the processor.

Table. 1. Main characteristics of the instances used for CMAQ & WRF-CMAQ benchmarking on AWS and Azure IaaS.

Instance familiy	Processor	Cores	Processor type	Memory
AWS c5n	Intel Platinum 8124	36	Skylake (x86_64)	192
AWS c6i	Intel Platinum 8375	64	Ice Lake (x86_64)	256
AWS c6gn	AWS Graviton2	64	Neoverse 1 (AArch64)	128
Azure DA_v4	AMD EPYC 7452	48	Zen2 (x86_64)	384
Azure HB v2	AMD EPYC 7002	120	Zen2 (x86_64)	480
Azure HB v3	AMD EPYC 7003	120	Zen3 (x86_64)	448

The AWS choices include compute optimized instances, denoted with ‘c’ that have the lowest memory per core ratio rather than general (‘m’) or memory intensive (‘r’) instances. The first two AWS instances are powered with Intel Xeon Skylake (c5n) and Intel Ice Lake (c6i) processors using x86_64 architecture. The second of these options was introduced in the second half of 2021 and targets performance, particularly for HPC and AI applications. The last AWS instance family (c6g) is powered by Graviton2 chips, which is a custom-made processor developed by AWS itself. It uses AArch64 architecture instead of the more traditional x86_64 architecture. The main advantages of AArch64 versus x86_64 are its lower power consumption and the more modern ISA. In general, processors using AArch64 architecture are more energy efficient and cheaper, although their performance in double-precision operations has traditionally lagged those of their x86_64 counterparts. AWS introduced Graviton2 at the end of 2019 targeting cost-conscious users. Other manufacturers are presently developing their own AArch64 chips.

The Azure IaaS use AMD EPYC processors. The first choice corresponds to the standard EPYC-2 instances offered by Azure, which scale up to 48 cores. The last 2 choices are HPC specific instances that use EPYC-2 and EPYC-3 processors with 120 cores. The measurements for the latter 2 are individual points using all the available cores, whereas the other measurements utilize instances with varying number of cores.

3.1 Southeast U.S. benchmark

The first benchmarks measuring performance use the standard Southeast U.S. benchmark. This benchmark, which is well-documented in the own CMAQ website, uses a 100 by 80 by 35 grid and tracks 218 species. The benchmark covers a single day simulation in the summer of 2016.

Fig. 1 shows the measurements of computational time versus number of cores for the Southeast U.S. benchmark in all the IaaS platforms listed in Table 1. Additionally, it also includes the results listed on the CMAQ website

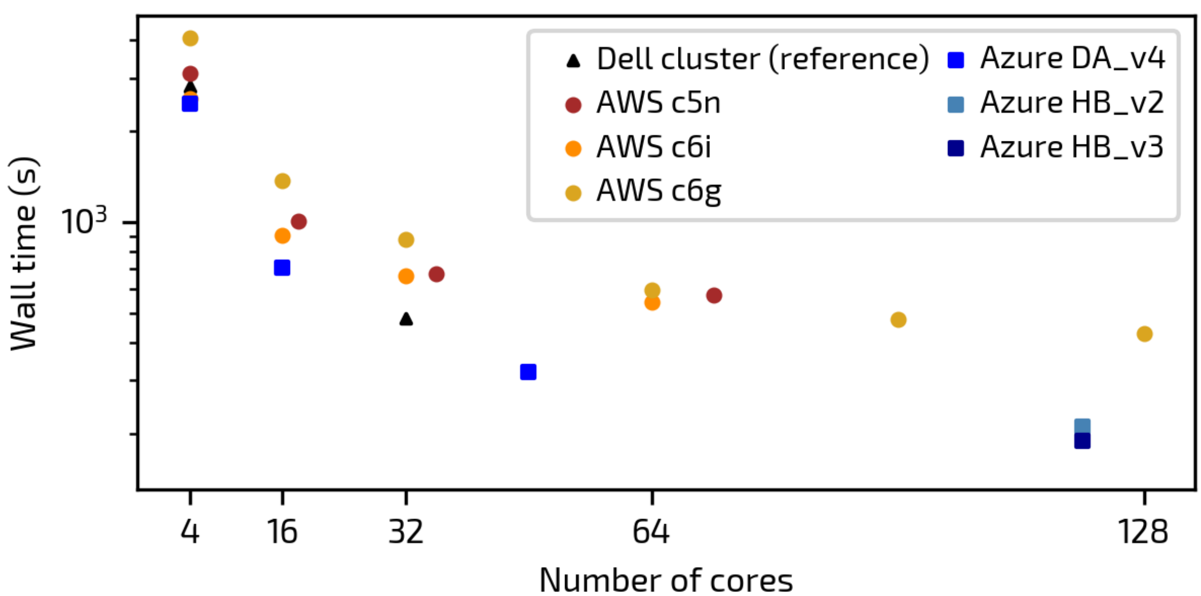


Fig. 3. Wall times for the CMAQ Southeast U.S. benchmark on IaaS from AWS and Azure versus number of cores.

with an on-premises EPA Dell cluster powered by Intel Xeon E5-2697A v4. This cluster is equipped with 16 cores per socket with a dual socket configuration for a total of 32 cores per node.

A general observation from Fig. 1 is a relatively weak scalability as increasing the number of cores leads to diminishing returns. However, Fig. 1 also shows that the performance from different instance families exhibits a relatively wide variety and requires individual evaluation. Here, it also must be noted that performance can be measured per core or per instance, which can lead to slightly different conclusions depending on the ultimate target. For a relatively low number of cores, up to 32, a single node of the EPA cluster performs better -lower computational times- than AWS instances with the same number of cores even those using Intel Ice Lake processors. In order to achieve a similar performance to the Dell cluster, it is necessary to use the largest instances with up to 64 cores. In this case, the performance of the Graviton2 instances is still somewhat weaker than the EPA cluster node but the performance of the Ice Lake instance falls very close to that of the EPA cluster node. These results also reveal that upgrading from Intel Skylake (c5n) to Intel Ice Lake (c6i) increases performance by about 20%. However, the best performers for the CMAQ Southeast U.S. benchmark are the Azure instances powered with AMD EPYC processors. These processors not only best the others at low core count number, but they also exhibit better scalability as the core count increases. The HB120rs_v2 and HB120rs_v3 instances outperform all the other hardware and are the only ones able to lower the wall time below the 200 seconds range.

3.2 WRF-CMAQ results

The next results come from the WRF-CMAQ benchmark, which also cover the Southeastern area of the continental US. The results presented here use short wave feedback, which results in computational times roughly 5 times those of the CMAQ case. Fig. 4 shows wall times versus the number of cores. The maximum value for the latter is 64 as increasing this number to the hundred range results in subdomains too small and the WRF code stops when noticing it. Therefore, the results for the Azure HBv2 and HBv3 instances use only 53.3% of the available computational capacity. It is highly desirable to have a larger WRF-CMAQ benchmark in the future to evaluate the maximum computational capacity for large instances and clusters.

The first deduction from Fig.4 is that the scalability is better than that of the CMAQ benchmark as the computational demand is greater. The evaluation of the performance of the different instances shows that the wall times with Graviton2 are very close to those of the EPA cluster for the same number of cores and using a c6g.16xlarge instance with 64 cores results in a computational time about half versus that of the EPA cluster node. The measurements with AWS c6i exhibit an even better performance and the computational time is further decreased. The results with AMD EPYC processors are also superior to the EPA cluster even for a low number of cores. For the highest number of cores used in the present tests (64), both Intel Ice lake and AMD EPYC results in computational times in the 800 seconds range.

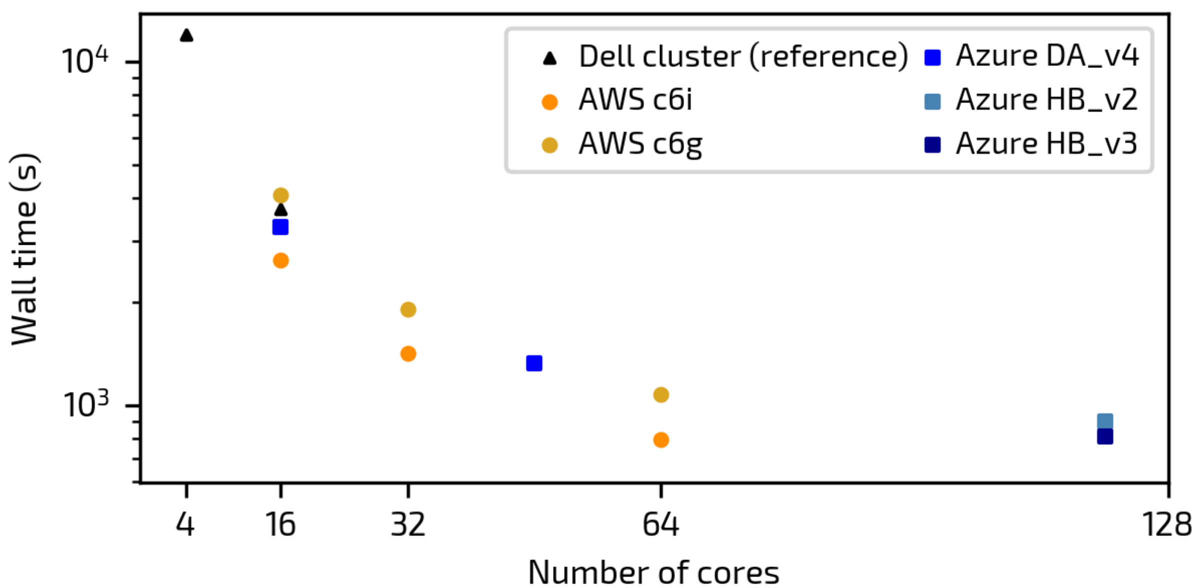


Fig. 4. Wall times for the WRF-CMAQ Southeast U.S. benchmark on IaaS from AWS and Azure versus number of cores.

4. COST ANALYSIS

A cost analysis is substantially more complex than a performance evaluation as the final cost depends on many factors with several of them being difficult to predict beforehand. A full cost analysis usually requires performing a total cost of ownership (TCO), which is complex and unique for each situation. As a minimum, it must include an evaluation of the costs associated with computational power, which usually accounts for a large fraction of the cost but not all of it, storage, outbound traffic, and any commercial fees. Furthermore, many of these categories have several tiers and conditions. The most typical example is storage with its very many options, and which can account from barely 1 or 2% to almost 20% of the total cost depending on the end-user needs. Here, things are somewhat simplified, and the cost analysis focuses on the cost associated with computational power but excluding storage or outbound traffic. Cloud providers offer different modalities of prices for their instances. The following discussion considers two tiers for the cost associated with computational power: on-demand or and spot prices. The former reflects the maximum cost associated with each instance without any discounts. Spot instances are offered by CSPs based on availability; their prices represent the maximum savings for each instance in that region, but this price varies among regions and can also fluctuate over time depending on demand. CSPs also offer other payment modalities such as reserved instances or private offers, which might involve a direct negotiation on the final price of instances and other resources. These prices are not considered here, and the on-demand and spot prices represent the upper and lower computational cost limits.

The cost estimates use the benchmark times discussed in the previous section. Fig. 5 shows the expenses incurred while performing the CMAQ and WRF-CMAQ benchmarks. These estimates are based on the largest instance for each family and takes into account how long it took to run the simulation along with the price for each resource. The prices for the latter reflect prices in the northern Virginia region for AWS and S. Central U.S. for Azure. Although spot prices are subject to change by the CSPs, we have not noticed much variability over the last few months, but potential users should be aware of this possibility.

The first and obvious conclusion from Fig. 5 is that using spot instances saves a lot of money, which is the situation not only for CMAQ but for most other apps. Looking at the results for CMAQ on AWS IaaS, another deduction is that using the c6g family with AArch64 processors saves some money versus Intel processors even though the latter is a better performer. However, these savings are not a fixed quantity and depend on the case, intended use and whether the user is running the app on on-demand, spot or even reserved instances. The results for CMAQ also show that Azure is a more economical option, at least for the case studied here, because of the better performance translating into lower cost. The regular instance family, DA_v4, is the one with the lowest cost. The cost estimate for WRF-CMAQ shows a slightly different picture. In this situation, AWS and Azure costs fall more closely with AWS c6g instances having the lowest cost based on on-demand prices but DA_v4 still being the most inexpensive for spot prices. These findings confirm that not only each user situation is unique, but that even different cases of interest to the same user might result in different conclusions from an economical viewpoint.

5. REFERENCES

Bailey D., Barton J., Lasinski T., Simon H., 1991: The NAS Parallel Benchmarks. Technical Report RNR-91-002, NASA Ames Research Center.

Chang S., Hood R., Jin H., Heistand S., Chang J., Cheung S., Djomehri J., Jost G., Kokron D. 2018: Evaluating the Suitability of Commercial Clouds for NASA's High Performance Computing Applications: A Trade Study. NAS Technical Report NAS-2018-01.

Dongarra J. J., Heroux M. A., 2013: Toward a New Metric for Ranking High Performance Computing Systems. SAND2013-4744, Sandia National Labs.

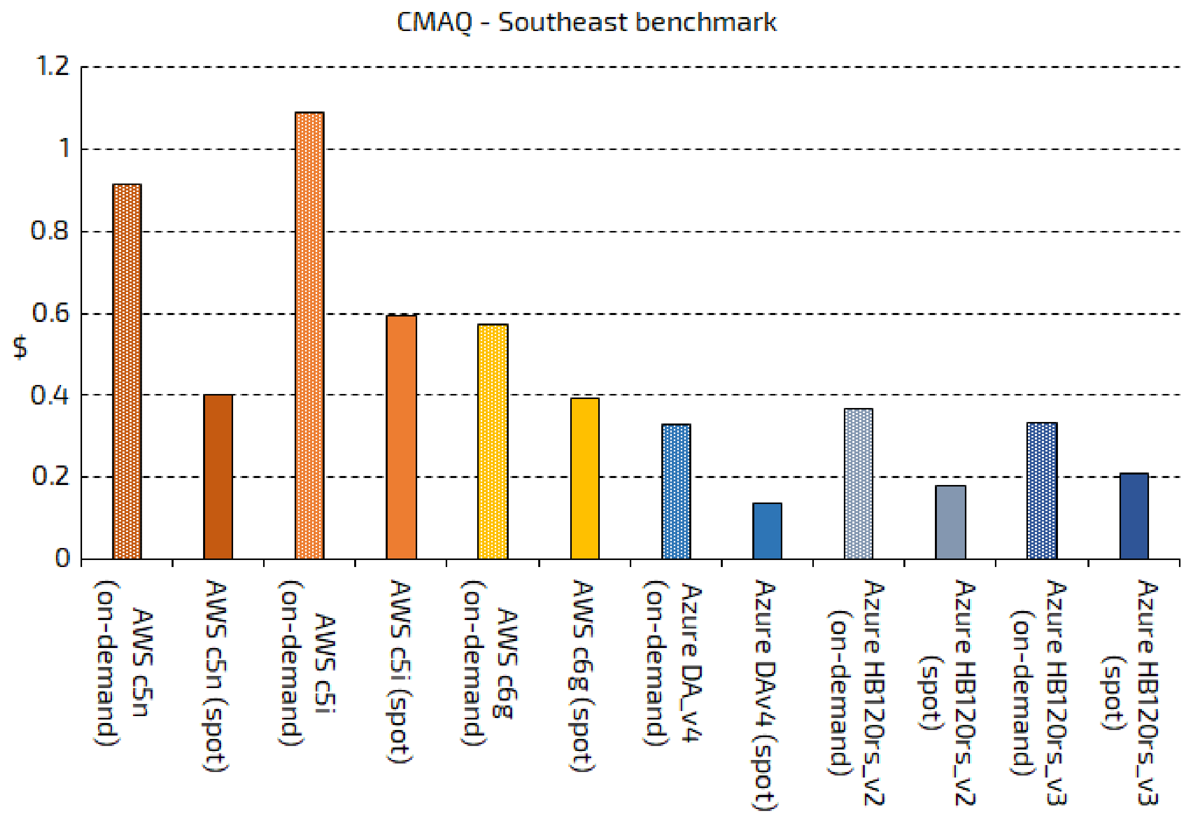
Dongarra J.J., Luszczek P., Petitet A., 2003: The LINPACK benchmark: past, present and future. Concurrency and Computation: Practice and Experience 15 (9), pp 803–820.

Fernandez A., 2021: Evaluation of the performance of tightly coupled parallel solvers and MPI communications in IaaS from the public cloud. IEEE Transactions on Cloud Computing.

Mohammadi M., Bazhirov T. 2018: Comparative benchmarking of cloud computing vendors with high performance linpack. Proc. 2nd Int. Conf. on High Performance Computation and Communication, ACM.

Powers J. G., Werner K. K., Gill D. O., Lin Y., Schumacher R. S., 2021: Cloud Computing Efforts for the Weather Research and Forecasting Model. BAMS, Vol. 102, p.E1261-E1274.

Yelick K., Coghlan S., Draney B., Canon R. S. 2011: The Magellan report on cloud computing for science. U.S. DOE.



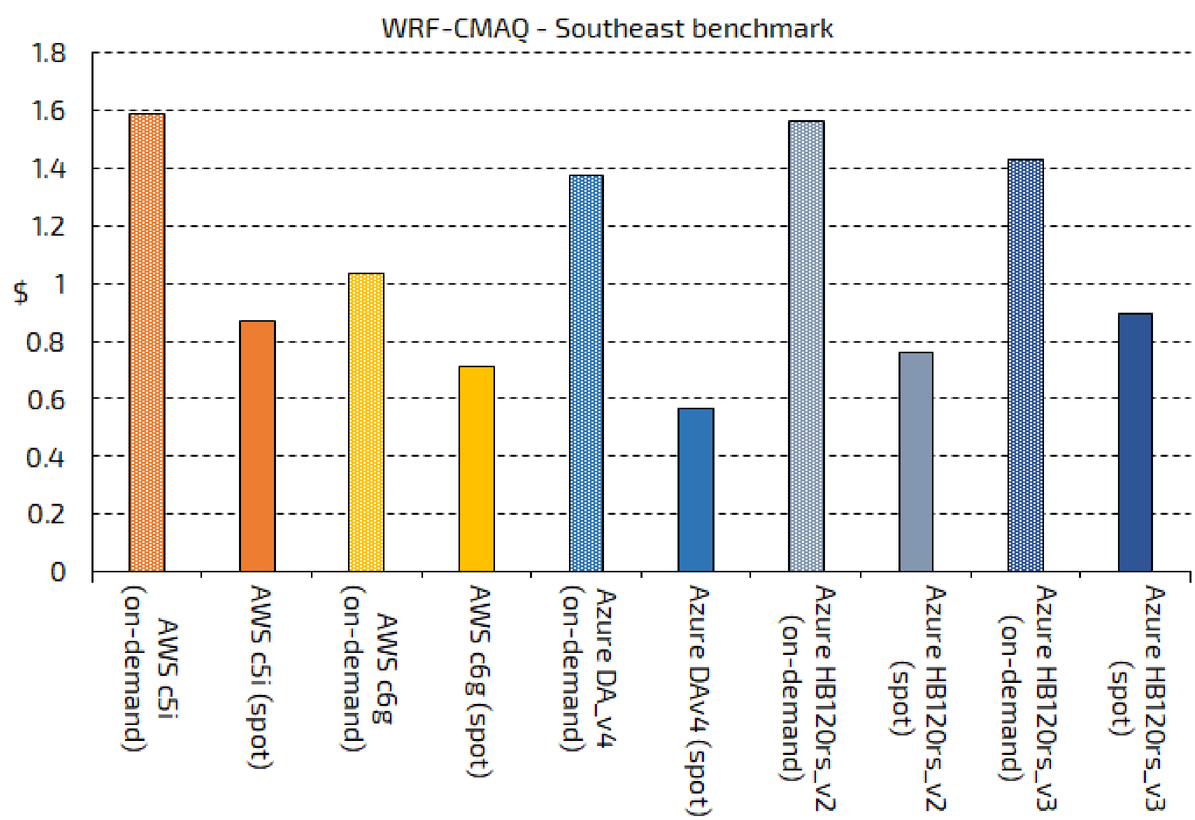


Fig. 5. Computational cost estimate for performing the WRF and WRF-CMAQ Southeast U.S. benchmark on AWS and Azure IaaS.