Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2017-270-RC1, 2018


Geoscientific Model Development Discussions
Open Access
EGU



# *Interactive comment on* "Performance evaluation of ROMS v3.6 on a commercial cloud system" *by* Kwangwoog Jung et al.

**Anonymous Referee #1**

Received and published: 8 January 2018

General comments.

This paper presents a short performance comparison between a cloud platform (amazon web services, provisioned using HPC-specific instances) and a locally available compute cluster. To evaluate the performance differences, it uses two metrics:

1. The high performance linpack benchmark (as used for the top500) 2. A single simulation scenario in the regional ocean modelling system (ROMS).

The motivation for such a comparison is strong. As soon as one moves to simulation problems that are larger than fit on a desktop machine, the question arises: "where can I run my code?". Traditionally, the answer has been "a local compute cluster" or "a supercomputer" (depending on problem size and availability). Now, a third option is



Printer-friendly version

Discussion paper

potentially attractive, namely to use a cloud provider.

To my mind, there are number of factors that come in to play here:

1. Accessibility. How easy is to access an appropriate level of compute resource? How quickly can I get access? 2. Usability. How easy is it to get a simulation started? What expertise do I need over and above running a simulation on my desktop machine? 3. Cost. How much do I have to pay to run my simulation? 4. Time to solution. How long does it take to get the results I need? This has (at least) two factors: the speed at which the code runs; and the time it takes to get the model set up, compiled, and through any queues.

This paper addresses, the "code speed" part of time to solution (4), and, to some extent accessibility and usability (1 & 2). There is one sentence discussing cost (page 5, line 28). However, I do not feel that the analysis presented in the paper is comprehensive enough to allow readers to make an informed decision on whether cloud services would provide a suitable avenue to address their simulation needs.

My major concerns are with the level of performance evaluation carried out, and the conclusions drawn. In summary, I do not think that the performance data presented in this paper provide enough evidence to back up the many statements made about the relative performance merits of virtualised hardware.

An example from Section 4:

Page 7, line 20:

"Virtualised IT resources are easier to allocate and manage than physical resources, but performance is slower because physical resources are provided through the software layer. Because the N/W resources are provided via virtualisation, the network is slower than the physical N/W environment."

There are two unsubstantiated statements here:

"Virtualised IT resources are easier to allocate and manage than physical resources."

and, in reference to the network:

"Because the network resources are provided via virtualisation, the network is slower than the physical network environment."

I see no evidence in the paper to support either of the claims. The former is rather subjective, but the latter can surely be objectively measured. On a local system, you could demonstrate that the introduction of a virtualisation layer reduces the performance of the network. Or else, you could cite a study that does this comparison.

The analysis of the results is littered with these kind of statements.

I would encourage the authors to read this paper on benchmarking scientific codes:

@InProceedings{HoeflerBelli2015, author = {T. Hoefler and R. Belli}, title = {Scientific Benchmarking of Parallel Computing Systems}, year = 2015, pages = {73:1–73:12}, month = {Nov.}, publisher = {ACM}, note = {Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC15)}, location = {Austin, TX, USA}, }

Some (not an exhaustive list) information that is lacking:

- The spec sheet memory bandwidth of the nodes - The achievable memory bandwidth (e.g. using the STREAM benchmark) - The spec sheet floating point performance of the nodes.

HPL usually achieves close to floating point peak on desktop CPUs. e.g., the first system on the top500 list that just uses Broadwell CPUs (https://www.top500.org/system/178925) has a peak flops of 8128TFlops and achieves 7040TFlops for HPL (around 86% efficiency).

Typical low-order finite difference codes and more likely to be limited by the memory bandwidth, so measurement of that would also be of interest.

In section 5.1, discussing the HPL results:

This section does not provide enough information for a reader to attempt to reproduce the data. For example, what size of matrix was inverted? There is some discussion of HPL in section 3.1, but this does not detail the exact setup:

Page 6, line 5: "The value of N governing complexity of tasks varies from 56000 to 125312". What is N? Why are these numbers so specific? Moreover, the statement that HPL is suitable for assessing the effect of network performance is not borne out either by the cited paper (Rajan et al. 2012), or a cursory analysis of the computational complexity of dense inverse computations. For an NxN matrix, HPL globally moves O(N^2) data and does O(N^3) flops. For asymptotically O(N) flops/byte moved. Even with a very slow interconnect, one does not expect to see any real network effects once N is "large enough".

Some particular comments:

There are good *performance models* for HPL. On commodity hardware, it should be straightforward to report the expected performance. Is 100 GFlop/s on 16 cores a good number?

Page 8, Line 6. The statement that "network latency may be smaller than in the AWS cluster" is not really supported by the presented data. As mentioned above, HPL is relatively insensitive to network performance. If you want to make a statement about network latency, then measure it.

If you want to characterise the network performance, then the Intel MPI benchmarks are a good place to start (https://software.intel.com/en-us/articles/intel-mpi-benchmarks/). You can run these on both the AWS instances you have spun up, and the local cluster. These give you direct information about the quality of the interconnect and MPI implementation, rather than attempting to guess from secondary data. For example, see a Supercomputing 2016 poster comparing AWS in-

stances with the TSUBAME-KFC supercomputer (http://sc16.supercomputing.org/sc-archive/tech_poster/poster_files/post267s2-file2.pdf).

Although the two experimental testbeds give approximately the same floating point performance, it is not possible to determine if the effect of virtualisation is introducing a penalty (because the peak performance of the hardware is not reported anywhere). Are these two lines similar because the hardware is approximately the same, and being used with similar efficiency? Or, does the AWS instance in fact have twice as much "spec sheet" performance, but you lose 50% for some reason (or vice versa)?

Moving on to the discussion of the ROMS simulation:

Section 5.2.

This again has little analysis of the data, other than to note that when using higher-resolution simulations, the scaling efficiency is better. But this is not surprising. When we increase the amount of local work (higher resolution), any effects due to serial fractions (Amdahl's law) from bad parallelisation or similar become smaller. An interesting question is how many degrees of freedom (dofs) are used per core. The C-grid staggering for ROMS means that, if I have done my sums correctly, there are about 1.7e6 dofs for each of velocity and pressure on the smallest grid. On 64 cores, this corresponds to around 3e4 dofs/core. For explicit schemes that have not been highly optimised, this is generally seen to be close to a break-even point for parallel scaling. So the results at this scale are not surprising That the AWS cluster stops scaling sooner is almost certainly due to network performance. This should be possible to model if one knew what the actual network latency and bandwidth were (you could measure this!). For an example of how to make these predictions I can recommend Fischer, Heisey, and Min (2015).

@TechReport{Fischer:2015, author = {Paul F. Fischer and Katherine Heisey and Misun Min}, title = {Scaling limits for {PDE}-based simulation}, institution = {Argonne National Laboratory}, year = 2015, number = {ANL/MCS-P5347-0515}, doi = {10.2514/6.2015-

3049} }

Section 5.3

Validation of the model notes that there is a small difference in the results between local and AWS run simulations. Probably this is fine, but one wonders. Was the comparison on the same number of cores in both cases? Does ROMS offer bitwise reproducible results when run on the same number of cores (or only statistically the same)? For example, if the authors were to run multiple times on a single system, would they expect a small spread in the results, or would they expect the same numbers every time? If the latter, it is not obvious why this should not transfer to different systems (if using the same number of cores).

Section 6.1

The discussion of hyperthreading effects is somewhat confusing. It seems like you are just saying "If you want performance equivalent to X physical CPUs, you should provision X physical CPUs (rather than the X/2 physical CPUs you would get with virtualised hyperthreaded CPUs)".

It is unclear from the discussion if allocating two instances each with 16 vCPUs is twice as expensive as one instance with 32 vCPUs. Furthermore, you do not present any evidence of the performance difference between 16 vCPUs and 32 vCPUs (the referenced Figure 10 compares performance between two different compilers).

Section 6.2

This again does not provide enough information to interpret the data. For example, what compiler versions did you use in both cases, what were the compiler flags? Presumably AWS is not special, and you would expect to see the same difference on the local system as well?

Section 6.3

This compares the time to solution between two different instance types. The difference appears to only be in the frequency of the utilised CPUs. The 2.9GHz CPU is 5% faster than the 2.3GHz CPU. An interesting question is, "how much cheaper is the slower CPU". A 5% slowdown seems like it might be worth paying if the cost is much less. But, considering table 1, it is much higher (presumably because more memory is expensive).

Again, assessing the raw data from this experiment is made needlessly difficult. Given the lower cost and higher performance of the c4 instances, it is unclear why the r4 instance was chosen as a comparison.

Some minor comments:

Page 3, line 17. Chen (2017) draw, I think, bad conclusions about the parallel scaling of CESM on AWS. I suspect their local supercomputing resource would have shown the same tail-off (if they had run that far).

Page 5, line 25. It is unclear why setting up the environment for models is time consuming on local HPC services (but not on AWS).

Why does copying VMs reduce the time for large scale experiments?

Page 7, lines 30–32 are copied verbatim from the AWS documentation. Although a citation is provided, it is (to me) not sufficiently clear that these are quotes, rather than description with referenced work. I have not exhaustively checked the rest of the paper for such examples.

Page 12. The code availability section does not reference HPL at all.

Page 15. Most of the instance types listed in this table are not referred to elsewhere in the paper, why are they therefore included?

The listed memory capacity of Table 1 does not tally with Table 2. If I understand the paper correctly, most simulations were carried out with c4.8xlarge instances (which





Printer-friendly version

Discussion paper

Table 1 claims supply 60GB RAM). But Table 2 claims that 128GB RAM were available. Which is correct? Again, I have not exhaustively checked details.

---

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-270, 2017.



Printer-friendly version

Discussion paper