Benchmarking the scalability of MPI-based parallel solvers for fluid dynamics in low-budget cloud infrastructure

Vanderlei M. Pereira Filho[1], **Márcio Castro**[1]

[1]Universidade Federal de Santa Catarina (UFSC)
Centro Tecnológico – Departamento de Informática e Estatística (INE)
R. Delfino Conti, s/n - Trindade, Florianópolis – SC, 88040-900 – Brazil

vanderlei.filho@posgrad.ufsc.br, marcio.castro@posgrad.ufsc.br

Abstract. *This ongoing study presents an analysis of the scaling of MPI-enabled partial differential equations (PDE) systems solver implementations with public cloud infrastructure. Shallow-Water equations are used as case study for analysis. Results indicate that public cloud can be a cost-effective solution for highly coupled problems, given a certain problem size threshold and appropriate infrastructure configuration. However, adequate software tooling and models are required for estimating and dimensioning optimal clusters.*

1. Introduction

Public cloud infrastructure is an affordable and easily available alternative for scientists and small organizations, allowing the execution of HPC workloads without relying on expensive on-premise specialized hardware. Nevertheless this approach incurs in several challenges, such as unreliable and slow networks, "noisy" tenants, and a relatively cumbersome configuration process [Netto et al. 2018]. This study aims to evaluate the viability of budget-constrained cloud environments for highly coupled parallel tasks using fluid dynamics simulation as case study.

Shallow-Water equations describe a thin layer of a fluid of constant density in hydrostatic balance, and are derived from the Navier-Stokes equations. To solve these systems numerically, most methods are based on computing the solution in discrete places inside a meshed geometry, which can then be divided and distributed for parallel processing. However, to compute the solution at mesh cells in the domains borders, information not locally available is needed, making this a not embarrassingly parallelizable problem.

2. Experiments and Results

We evaluated the distributed Shallow-Water model provided by the Oceananigans[1] Julia package (v0.60.0). Realistic models include detailed information describing the ocean shape and floor topography, however we confine our experiments to a $4\pi km^2$ square region with a flat floor and reflective borders. We also consider two problem-sizes in terms of resolution: problem A with 128^2 cells and problem B with 1024^2 cells. The experiments are set to simulate 100 seconds of shallow-water physics with a time-step of $0.001s$, totalizing 100000 iterations. The MPI topology applied is cartesian and load-balancing follows a one-dimensional slab decomposition strategy.

Cloud environments were built with Amazon Web Services EC2[2] instances at availability zone `us-east-1`. Instance types used in this study are described in detail at Table 1.

Table 1. Tested Amazon Web Services EC2 Instance Types.

Instance Type	Number of vCPUs	Sustained Clock Speed	RAM Memory	Network Performance	Hypervisor	On-Demand Pricing
t3.2xlarge	8	2.5GHz	32GB	Up to 5Gbps	Nitro	0.3328 USD/h
c4.2xlarge	8	2.9GHz	15GB	Up to 1Gbps	Xen	0.398 USD/h
c5n.2xlarge	8	3.4GHz	21GB	Up to 25Gbps	Nitro	0.432 USD/h

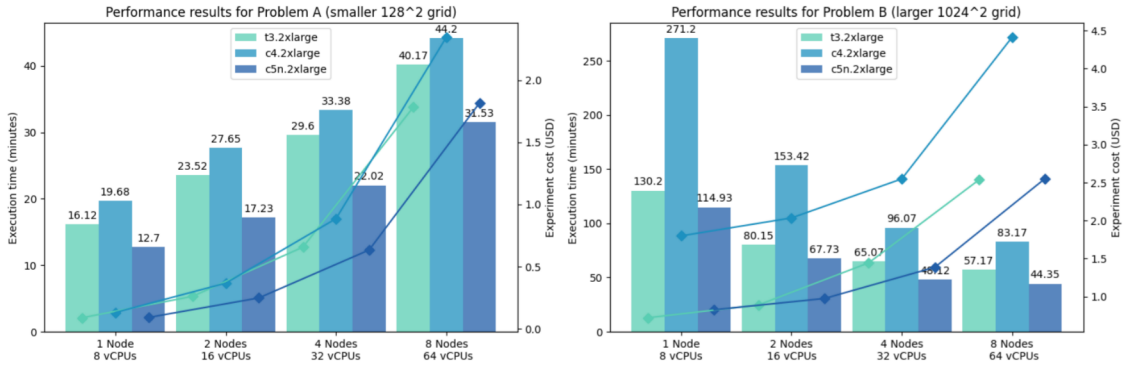

Figure 1. Experimental results (left-side: Problem A; right-side: Problem B).

The poor scaling for problem A shows that there is a lower bound on problem size to justify the use of multiple nodes. The reason is due to the network bandwidth bottleneck, which is at most 25Gbits for `c5n.2xlarge` machines and much worse for cheaper nodes. `c5n.2xlarge` is 23% more expensive than `t3.2xlarge`, but results show that using it is actually slightly cheaper because of less compute time (see results for Problem B executed with 4-node clusters). A problem-size threshold for efficient use of cluster resources is difficult to model given the amount of variables involved. Nevertheless, it is possible to find a rough cost rate estimate empirically before committing infrastructure for a complete solution, by executing the same simulation with a limited number of iterations for example and extrapolating results to the total expected number of iterations. Knowing execution time beforehand is of critical importance for optimizing cluster configurations, and thus cloud HPC cost-effectiveness.

The addition of fault tolerance characteristics into the distributed solvers may also allow the exploration of considerably cheaper spot machines, which are transient resources that can be revoked by the cloud provider at any time. Spot instances are usually between 30% to 70% cheaper than the on-demand instances used in this study. We are currently studying two approaches for fault-tolerant MPI applications, the first based on process-level checkpoint restart with external libraries, and a second more intrusive approach based on User-Level Failure Mitigation (ULFM), the latest effort of the MPI Forum[3] for standardizing error handling in the MPI standard.

References

Netto, M. A. S., Calheiros, R. N., Rodrigues, E. R., Cunha, R. L. F., and Buyya, R. (2018). HPC Cloud for Scientific and Business Applications: Taxonomy, Vision, and Research Challenges. volume 51. Association for Computing Machinery, New York, NY, USA.

[1]https://github.com/CliMA/Oceananigans.jl
[2]https://aws.amazon.com/pt/ec2/
[3]https://www.mpi-forum.org/