The Eleventh International Conference on
Engineering Computational Technology
23–25 August, 2022 | Montpellier, France

Proceedings of the Eleventh International Conference on
Engineering Computational Technology
Edited by B.H.V. Topping and P. Iványi
Civil-Comp Conferences, Volume 2, Paper 8.2
Civil-Comp Press, Edinburgh, United Kingdom, 2022, doi: 10.4203/ccc.2.8.2
©Civil-Comp Ltd, Edinburgh, UK, 2022

Behaviour of asynchronous parallel iterative algorithms on cloud computing type architectures

M. A. Rahhali[1] and T. Garcia[2,3] and P. Spiteri[3]

**[1]ENGIE
Saint-Ouen, France
[2]CY Tech
CY Paris University, France
[3]IRIT-ENSEEIHT
INP Toulouse, France**

Abstract

In order to solve a discretized stationary linear diffusion problem, this paper deals with a comparison between the performances of parallel iterative synchronous or asynchronous algorithms implemented on a Cloud computing architecture with several sizes of an algebraic systems to be solved.

Keywords: numerical, parallel, asynchronous, performance, implementation, cloud.

1 Introduction

Synchronous parallel iterative algorithms are computing methods in which the communications between the processors are synchronized at the end of each iteration. In this kind of method, when the load on each processor is not uniform or when the processors do not have the same performance, at each synchronization point the processing units must wait for the slowest unit. Consequently, given a heterogeneous distributed architecture, the idle times of the processing units will degrade the performance of these synchronous methods and the job restitution times will be penalized.

Asynchronous parallel iterative algorithms correspond to methods in which the communications between the processors are not synchronized at each iteration.

Therefore, when a processing unit has finished its own calculations, it starts the next cycle again using the latest interaction data computed by the other processors and received during the previous cycle, without waiting for the arrival of new results delivered by the other processing units. However, there are asynchronous parallel methods with more flexible communications between the processors where the interaction data is used without a predetermined order. Thus, the calculations are processed on each processor respecting the own rhythm of each processing unit and using the last available values calculated by the other processors. Depending on the frequency of updates of calculations between processors, the speed of convergence of these asynchronous parallel methods can be slower. But the big advantage lies in the fact that there is generally a reduction in the elapsed time to reach convergence.

The convergence of both synchronous or asynchronous algorithms can be studied by various methods; however, when the fixed point equation associated to the problem to be solved is contracting one obtains on the one hand convergence whatever be the decomposition of the problem into sub-problems and on the other hand the value of the contraction constant gives an estimate of the convergence speed. Moreover, the successive iterates are located in nested sets centered in the solution, which allows to implement efficient numerical stopping tests insofar as one can give an estimate of the diameters of each nested set and stop the iterative process when a diameter is smaller than a given tolerance. In previous studies (see [1], [2], [3], [4], [5]), we have implemented these methods on grids constituted by heterogeneous and distant machines; we could observe that asynchronous algorithms were very efficient when there was a lot of synchronization between the processors.

2 Methods

In this paper we now consider the implementation of these two kinds of methods and their comparison on a Cloud computing architecture. To test this, we solve the following Poisson problem with homogeneous Dirichlet boundary conditions defined in the unit cube:

$$\begin{cases} -\Delta u(x,y,z) + q.u(x,y,z) = f(x,y,z), (x,y,z) \in]0,1[^3, q \geq 0 \\ u(x,y,z) = 0 \text{ on the boundary} \end{cases} \tag{1}$$

Cloud computing is a way of providing access to computer resources, characterized by its self-service availability, elasticity, openness, mutualization and pay-per-use; self-service and on-demand resources of storage capacity and computing power are based on the customer's needs. This contrasts with so-called “traditional” computing where any need to change an application requires manual operations and therefore time.

In the Cloud, the need, automatically detected by the application or at the customer's request, is taken into account and satisfied immediately and we no longer speak of infrastructure components but of services (see Figure 1): IaaS (Infrastructure as a Service) which provides CPU, Storage, RAM, Network; PaaS (Platform as a Service) which provides ready-to-use middleware (database, web server, etc.) and SaaS (Software as a Service) which provides ready-to-use software.

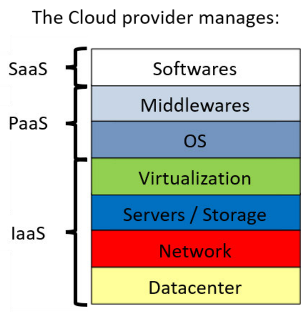

Figure 1: Cloud Services.

In this study, synchronous and asynchronous parallel algorithms have been developed in C language to solve Problem (1) and simulation was carried out on a cloud computing architecture based on a platform located in France.

This platform (called Grid'5000 see [6]), is a large-scale and versatile testbed for experiment-driven research in all areas of computer science, with a focus on parallel and distributed computing including Cloud, HPC and Big Data. This environment (see Figure 2), provides access to a large amount of resources (8 sites, 456 computers), gives an easy access to a wide variety of hardware technologies (Intel, AMD, ...) and is particularly suitable to carry out experiments.

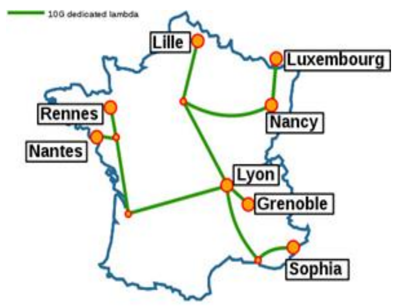

Figure 2: Platform architecture.

It can be used as an infrastructure proposed by a Cloud provider which shares mutualized resources and, a virtual computer corresponds to a quantity of virtual resources in the cloud. It also allows the execution of virtual computers with hardware virtualization capabilities as well as subnet reservation to give a routed IP range for experiments. Typically, this platform can be viewed as a service IaaS because it manages, as a Cloud provider, server hardware, virtualization layers, storage and networks and the client manage the middleware and the software (executables, settings, …). It represents a model where the customer has an infrastructure that is physically located at the Cloud provider.

3 Results

Our parallel simulations can therefore be run with reserved resources in the platform presented above with different numbers of cores, memory size and processor technologies. These resources are operating system templates and each instance start as an identical clone of all other instances. It is the user data that gives each cloud instance its personality and "cloud-init" is the tool that applies user data to instances (see [7] for more information). It uses two files: meta-data which contains configuration such as hostname, "ssh" key (ssh protocol uses encryption to secure the connection between a client and a server), instance id, ... and user-data which can contain more configuration in a different format such as a linux script or a "YAML" file (YAML is a language used for writing configuration files) that describes configuration such as creating users, mounting a device, ... The proposed disk image is in "qcow" format which uses a disk storage optimization strategy that delays the allocation of storage until it is actually needed. It uses "libvirt" which is a toolkit for managing virtualization servers. It will create a virtual interface and attach it to the bridge so that the virtual machine can reach the rest of resources and access to internet. In addition, it is possible to run and manage the guest operating system with "virsh" which is a utility for managing virtual machines from the command line and use "ssh" directly on the virtual machine from anywhere.

For our Problem (1) to be simulated, several sizes 120^3 (i.e. 1 728 000), 240^3 (i.e. 13 824 000) and 360^3 (i.e. 46 656 000) of the algebraic systems to be solved have been considered (see Table 1 below), on 64 computers resources.

In this Table 1, synchronous and asynchronous results are presented in term of elapsed time and number of relaxations. The parameter τ measures the ratio of elapsed time between synchronous and asynchronous methods. This parameter is significant in order to achieve the comparison of synchronous and asynchronous parallel methods.

It can be noticed that for the considered application and the used architecture, the asynchronous computation scheme gives better results than the synchronous scheme; on 64 machines, it is clearly faster by 10.2, 4.6 and 5.0 respectively.

	Synchronous		Asynchronous		τ
Size	Elapsed time	Number of relaxations	Elapsed time	Number of relaxations	
120^3	438	749056	43	969116	10.2
240^3	6076	2886016	1308	3815801	4.6
360^3	38460	6364544	7728	7895476	5.0

Table 1: Synchronous and asynchronous simulations on Cloud computing.

4 Conclusions and Contributions

The synchronous and asynchronous parallel experiments were performed for the solution of a Poisson problem with homogeneous Dirichlet boundary conditions; according to our experience (see [1]), similar performances would be obtained considering other boundary conditions. Moreover, the model problem considered, although linear, constitutes a sufficiently illustrative example because it acts as an auxiliary problem for the solution of more complex problems, in particular those that are strongly nonlinear. The nonlinearities that can be taken into account are of two types: on the one hand by perturbing the partial derivative operator by a diagonal operator, for example an exponential of the solution or more generally a monotonic one, and on the other hand by taking into account inequality type constraints on the solution. For example, in the first case, the target applications correspond to the simulation of solar oven or to situations involved in the study of plasmas, whereas in the second case the target applications are involved in financial mathematics or more generally in image processing modelled by the Hamilton-Jacobi-Bellman equations. Results on cloud architecture show that asynchronous parallel methods present real interest when parallel processing requires a large number of synchronizations. This occurs in particular when the convergence is slow, and especially when the communications between the machines are slow. Indeed, communications on a platform are dependent on the latency of the network. This slowdown is accentuated by the distance between machines that are not connected by direct lines. Thus, correlatively, the performance in terms of restitution time of asynchronous parallel methods is much higher than that of synchronous methods. This phenomenon of superiority of the asynchronous parallel methods is thus accentuated when the processors are heterogeneous and distant, which is the case in our simulations on cloud computing; we can thus see here the influence of the architecture of the machines on the performances of the algorithms. Considering the additional software layers, the use of asynchronous parallel algorithms is of interest in a cloud environment compared to the synchronous equivalent insofar as inactivity times are eliminated.

The list of possible applications can be extended to other situations and is not limited to the above one. Moreover, we can also extend the field of application to problems of convection - diffusion and then consider the resolution of the Navier - Stokes equations which in many cases comes down to the resolution of a Poisson equation coupled to a convection - diffusion equation. Finally, simulations are envisaged on FG Cloud of France Grilles, a service based on a federated academic cloud infrastructure (see Figure 3).

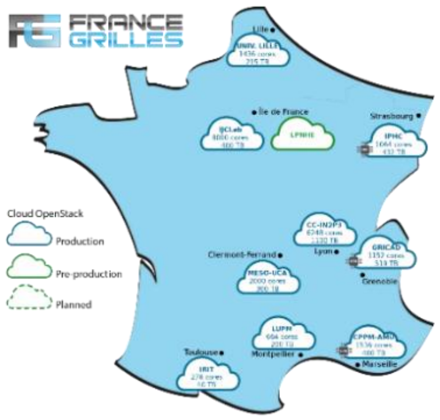

Figure 3: France federated academic cloud infrastructure.

Acknowledgements

The authors would like thank the Grid'5000 platform for allowing the reservation of resources and FG Cloud of France Grilles for future works.

References

[1] P. Spiteri "Parallel asynchronous algorithms: a survey", Advances in Engineering Software, Elsevier, 149, 2020. doi:10.1016/j.advengsoft.2020.102896

[2] T. Garcia, P. Spiteri, L. Ziane-Khodja, R. Couturier "Solution of univalued and multivalued pseudo-linear problems using parallel asynchronous multisplitting methods combined with Krylov methods", Advances in Engineering Software, Elsevier, 153, 2020. doi:10.1016/j.advengsoft.2020.102929

[3] T. Garcia, P. Spiteri, L. Ziane-Khodja, R. Couturier "Coupling parallel asynchronous multisplitting methods with Krylov methods to solve pseudo-linear evolution 3D problems", Journal of Computational Science, Elsevier, 51, 2021. doi:10.1016/j.jocs.2021.101303

[4] M. Chau, T. Garcia, P. Spiteri "Asynchronous Schwarz methods applied to constrained mechanical structures in grid environment", Advances in Engineering Software, 74, 1-15, 2014. doi: 10.1016/j.advengsoft.2014.03.005

[5] T. Garcia, G. Khenniche, P. Spiteri "Behavior of parallel two-stage method for the simulation of steel solidification in continuous casting", Advances in Engineering Software, Elsevier, 131, pp.116-142, 2019. doi:10.1016/j.advengsoft.2018.11.012

[6] R. Bolze, F. Cappello, E. Caron, M. Daydé, F. Desprez, E. Jeannot, Y. Jégou, S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, P. Primet, B. Quetier, O. Richard, E.G. Talbi, I. Touche "Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed" in International Journal of High Performance Computing Applications, 20-4, pages 481-494, 2006.

[7] Cloud-init : https://cloud-init.io/