Advancing Research on Bioinformatics and Cloud Infrastructure using AWS

Alba C. M. A. Melo[1], **Lucia M. A. Drummond**[2], **Celia G. Ralha** [3], **Gustavo J. Portella** [4], **Luan Teylo** [5], **Aldo H. D. Mendes** [6]

[1]Department of Computer Science – University of Brasilia (UnB)
Campus Asa Norte - Predio CIC/EST - 70910-900 - Brasilia - DF - Brazil

[2]Institute of Computation – Federal Fluminense University (UFF)
Av. Gal. Milton Tavares de Souza, s/n - 24210-346 - Niteroi - RJ - Brazil

[3]Institute of Computation – Federal University of Bahia (UFBA)
Av. Milton Santos - Ondina - 40170-110 - Salvador - BA - Brazil,

[4]Executive Board of Information Technology – Coordination for the Improvement of Higher Education Personnel (Capes)
SBN Quadra 2, Edificio Capes - 70040-020 - Brasilia - DF - Brazil

[5]Inria Centre at the University of Bordeaux
Sud-Ouest, Talence, France

[6]University Center Unieuro (UNIEURO)
Campus Aguas Claras – 70297-400 - Brasilia - DF - Brazil

Abstract. *This paper describes joint research to accelerate Bionformatics Applications in the AWS Cloud. There were two main axes of research: parallel sequence comparion Bioinformatics tools and cloud schedulers to execute efficiently the parallel applications in the AWS cloud. Our project involved three universities and one research institute and produced as outcome 4 papers in prestigious international journals, 2 papers in international conferences and 5 PhD Theses. As results of scientific research, we developed (a) a cloud scheduler that aims to reduce both the running time and the cost of the execution; (b) a combined model that uses statistics and neural networks to predict the cost variation of spot instances; (c) a multiagent framework to provision and execute cloud applications; (d) a fault tolerant strategy to execute long running applications with GPUs in the cloud. Additionally, since the duration of the CNPq-AWS project was from 2020 to 2021, thus during the covid-19 pandemics, we were able to compare hundreds of thousands of SARS-CoV-2 sequences with our cloud schedulers and parallel sequence comparison tools. Finally, members of our group were editors of a book on High Performance Clouds, published in 2023 by Springer Nature.*

1. Introduction and Goals

Biotechnology can be defined as an interdisciplinary area that involves the study of DNA, RNA, proteins and more complex molecules; cell and tissue cultures; bioinformatics and nanotechnology, among others [Gupta et al. 2017]. Bioinformatics is an important part of Biotechnology because it consists of creating tools and algorithms to collect, store and

analyze biological data [Borin et al. 2023]. Among the popular applications of Bioinformatics, the analysis of biological sequences (DNA, RNA and proteins) stands out, which determines their structure/function, playing a fundamental role in the development of medicines and the study of diseases.

Traditionally, Bioinformatics laboratories had powerful computers, many of them containing accelerators such as GPUs, allowing the execution of intricate algorithms that will lead to complex analyses and scientific breakthroughs. However, many Bioinformatics labs have recently moved to the cloud [Banimfreg 2023], taking advantage of its stable, resilient and flexible platform.

Amazon AWS (Amazon Web Services) is a cloud provider that has been established in the market for many years and offers various types of resources suitable for a wide range of applications. Amazon EC2 (Amazon Elastic Compute Cloud) offers computing resources using basically two pricing models: on-demand, which has a fixed cost (USD/hour), and spot, which has a variable cost, generally much lower than the on-demand model. While on demand instances are only freed by the user who allocates them, spot instances may be revoked by AWS at any time, complicating their management. As of February 27, 2026, Amazon EC2 had 259 data centers spread across 39 regions and offered more than 750 different types of computing instances, including CPUs, GPUs (Graphics Processing Units), and FPGAs (reconfigurable hardware)[1].

Bioinformatics applications, particularly those of biological sequence comparison, often require high computing power. In particular, biological sequence analysis applications are complex, composed of several tasks [Sandes et al. 2016b] [Figueiredo et al. 2021], and can be classified as HPC (High Performance Computing) applications. Their execution can take hours or even days to complete and, therefore, they typically use powerful execution platforms, such as clusters of CPUs, GPUs or FPGAs.

The goal of this paper is to present the main results achieved in our BioCloud CNPq/AWS project, which aimed to execute Bioinformatics applications efficiently in the AWS cloud. Our project addressed the problem of resource allocation and management in the Amazon AWS cloud for Biological Sequence Comparison HPC applications, minimizing execution time and maximizing availability, without violating service level agreements (SLAs), in order to run biology applications efficiently and with lower financial costs for the user. In this sense, both permanent (on-demand) and transient (spot) instances [Portella et al. 2019b] will be considered, as well as multiple types of computational resources (CPU, GPU, and FPGA), increasing the range of alternatives explored. The use of intelligent agents [Ralha et al. 2019] to assist in management decisions was also investigated.

2. Overview of the BioCloud Project

In the BioCloud project, we put together a strong research team, composed of 32 researchers from three universities (University of Brasilia, Federal Fluminense University and State University of Rio de Janeiro) and one research institute (Embrapa Genetic Resources and Bioechnology), with joint and complementary research skills.

Since the project took place during the Covid-19 pandemic, we had to hold fre-

[1]https://aws.amazon.com

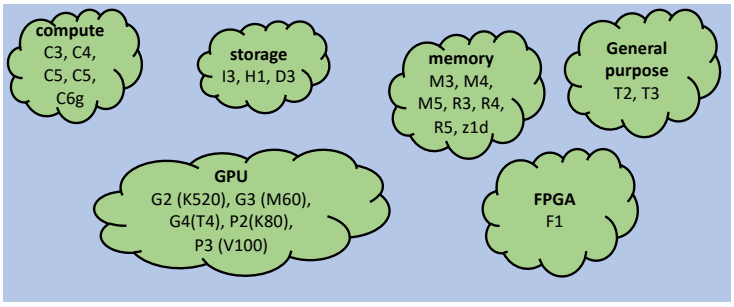


Figure 1. Instance families used in the BioCloud project

quent virtual meetings and use slack to share files. We also decided to use the parallel sequence comparison applications from the project to compare hundreds of thousands of SARS-CoV-2 sequences, which were made available in the public genomic databases.

At the beginning of the project, we decided to use a vast number of instances from different families, as shown in Figure 1. The instances on the top of the figure are CPU instances, whereas the instances at the bottom are accelerator instances (GPU and FPGA).

In our project, we also used several AWS services and tools, as shown in Figure 2. The AWS lambda service was used to investigate how the sequence comparison applications benefit from the serverless computing model. The AWS Parallel Cluster tool was used to create a dedicated HPC environment in AWS, for the applications that needed network bandwidth guarantees. We also used three types of storage (S3, EFS and EBS), that were chosen based on price, latency and capacity, according to the specific characteristics of each research study. Finally, AWS Cloud Watch was used to monitor the execution of the applications and infrastructure tools developed in the project.

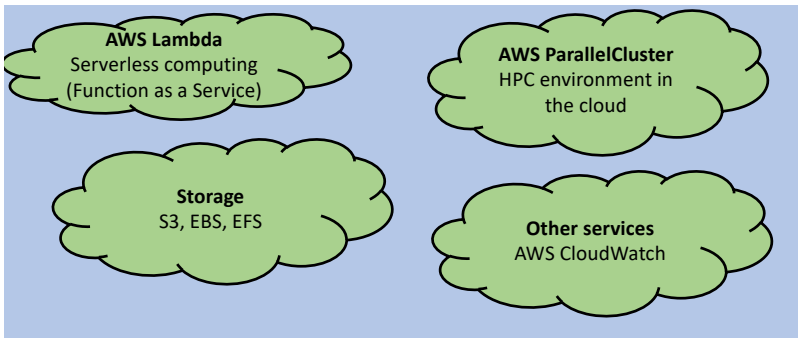


Figure 2. Services and tools used in the BioCloud project

3. Results

3.1. Bibliographic outcome

In the context of the BioCloud project, the following outcomes were produced:

- Four papers in prestigious journals, addressing cloud resource managers [Mendes et al. 2024] and schedulers [Teylo et al. 2023], function-as-a-service parallel executions [Carvalho et al. 2023] and prediction of spot instance prices [Portella et al. 2024];
- Two papers in prestigious international conferences, addressing fault tolerance with spot instances [Brum et al. 2021] and bag-of-tasks scheduling in the cloud [Teylo et al. 2021];
- One book edited by some members of the project on High Performance Computing in the Clouds, published by Springer Nature in 2023 [Borin et al. 2023]. Our research team was contacted by Springer Nature because an editor from Springer attended to our talk in the IEEE/ACM CCGRID 2021 conference [Teylo et al. 2021];
- Five PhD Theses, addressing cloud schedulers (Luan Teylo from UFF [Teylo 2022]), cloud spot price prediction and selection (Gustavo J. Portella from UnB [Portella 2021]), agent systems for cloud schedulers (Aldo Mendes from UnB [Mendes 2024]), parallel sequence comparison in CPU-FPGA systems (Carlos A. C. Jorge from UnB [Jorge 2022]), and schedulers for federated learning applications (Rafaela Brum from UFF [Brum 2024]).

3.2. Scientific Research results

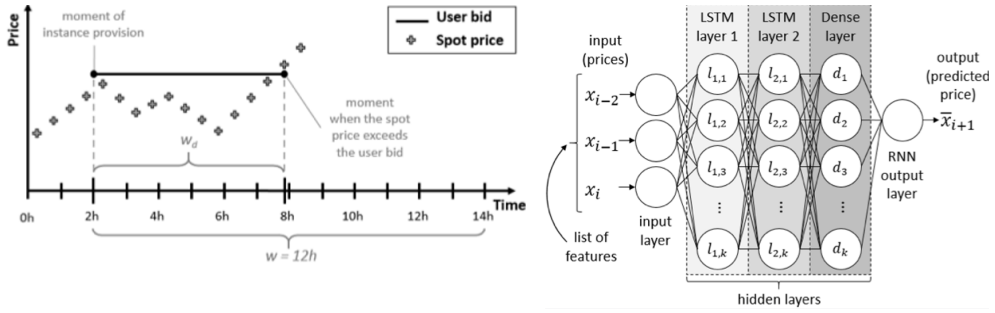
In this section, we will briefly describe four major scientific outcomes of the BioCloud project.

3.2.1. Selecting Low Price AWS Spot Instance with Low Probability of Revocation

Prior to the start of the project, the work [Portella et al. 2019a] was done to perform a statistical analysis of the AWS spot pricing model. In this work, we used time-smoothed moving averages of 12-hour periods, aiming to provide a price-availability trade-off to the user. Depending on instance type, user's bid could be set at 30% of the on-demand price, with an expected availability above of 90%. This study was important for the subsequent analyses, which used not only statistics, but also utility models and neural networks.

In the specific case of biological sequence comparison, depending on the lengths of the sequences or the number of comparisons, the parallel applications can take several hours, incurring in high cost. In the work [Portella et al. 2024], we investigated the problem of executing long running parallel biological sequence comparison applications in the AWS cloud with low price and low probability of revocation.

Our utility prediction strategy consisted of two steps: (a) short term prediction (12 hours), which used sliding windows; and (b) long term prediction (7 days), which used Long Short Term Memory (LSTM) Neural networks to do the prediction. First, we do the long term prediction, choosing a set of instances and then refine the prediction with



(a) short term – statistics

(b) long term - LSTM

Figure 3. Price Prediction for AWS Spot Instances [Portella et al. 2024]

sliding windows for the selected instances in the previous step. Figure 3 illustrates our strategy.

We used our strategy in a covid-19 case study that compared 218179 pairs of SARS-CoV-2 sequences (Alpha variant) with our MASA-OpenMP parallel tool [Sandes et al. 2016a]. Eight different spot instances were considered and we analyzed the cloud spot price behavior in the first eleven months of 2020. Then we run our analysis to execute the case study at the end of January 2021. Figure 4 presents the results for the whole study with spot instance m5.2xlarge (8 vCPUs). We can see in this figure that, if we used the default AWS approach, our spot instance would had been revoked with 7 hours and 44 minutes of use, without being able to complete the execution. Using our utility approach, we were able to execute for 8 hours and 49 minutes, i.e., about one hour longer than the default approach.

Instance (Date)	Spot Request	Max Price	Cost USD	#Cmp	Setup Time	Exec. Time	USD/ #Cmp	Time/ #Cmp
m5.2xlarge	Utility	0.152	1.43	43560	00:28:17	08:49:59	0.00003283	0.730
01/13/2021	Current	0.143	0.29	5420	00:41:46	01:06:42	0.00005351	0.738
	Default	-	1.29	37701	00:52:49	07:44:56	0.00003422	0.740

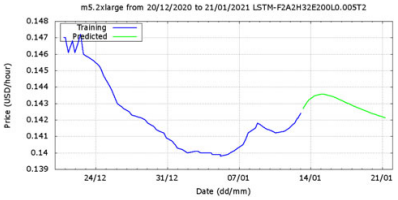


Figure 4. Results for the covid-19 study (m5.2xlarge) [Portella et al. 2024]

3.2.2. Reducing the Cost of AWS Executions with QoS Guarantees

One of the main objectives of the cloud users is to execute their applications in the cloud paying a low price. However, if only the price is considered, a weak instance could be selected, leading to extremely long execution times. This solution is clearly not appropriate.

In the work [Teylo et al. 2023], a scheduler called BURST-HADS is proposed that executes applications in the cloud considering both the price and the deadline for completing the executions. The proposed technique is composed of 2 modules. The primary

module uses Iterated Local Search (ILS) to choose the instance and the secondary module periodically monitors the execution, possibly choosing a more appropriate instance and migrating the application to it. This way, it is guaranteed that the deadline is respected.

Figure 5 shows the results obtained with jobs presenting different characteristics. We can see that, for job J100, which has 100 tasks that take about 300 seconds each, the cost was reduced in 48%, when compared to the on-demand execution. In this case, there was an increase in the execution time of 22% which did not violate the deadline.

JOB	Burst-HADS w/o Hibernation		HADS w/o Hibernation		AutoBoT-like w/o Interruptions		ILS On-demand	
	cost	makespan	cost	makespan	cost	makespan	cost	makespan
J60	$0.112	1,274	$0.067	2,290	$0.166	2,221	$0.271	1,112
J80	$0.151	1,329	$0.104	2,295	$0.199	2,266	$0.312	1,190
J100	$0.176	1,660	$0.112	2,332	$0.218	2,342	$0.371	1,462
ED200	$0.357	2,275	$0.267	2,580	$0.387	2,566	$0.698	1,887

Figure 5. Comparative execution results of BURST-HADS [Teylo et al. 2023]

In [Teylo et al. 2021], BURST-HADS was used to compare 22560 pairs of SARS-CoV-2 sequences (Alpha variant), organized into 60 supertasks, with our MASA-OpenMP parallel tool [Sandes et al. 2016a]. A total of 24 EC2 instances (136 vCPUs) were used and the comparison took about 21 minutes.

3.2.3. Comparing Huge Chromosomes in spot GPU with Fault Tolerance

Comparing huge chromosomes with exact algorithms compute large dynamic programming matrices and may take hours or even days to finish [Durbin et al. 1998]. GPUs may be used to accelerate those comparisons but the execution time remains considerable. Moreover, it is known that GPU instances may be expensive.

In the work [Brum et al. 2021], we address the challenge of reducing the cost of long running GPU executions. We proposed a framework that executes our MASA-CUDAlign GPU sequence comparison tool [Sandes et al. 2016b] [Figueiredo et al. 2021] using AWS GPU spot instances, with fault tolerance. Periodically, the last computed row of the matrix is saved, so that execution can restart from this row. Additionally, the framework monitors the execution. If it receives a notification from AWS stating that the instance will be revoked in 2 minutes, the framework selects another appropriate spot instance and restarts the execution from the last row saved.

In the experimental results, we compared the human and chimpanzee homologous chromosomes 19, 20, 21, 22 and Y in AWS spot GPUs, considering three scenarios, that vary depending on the probability of revocation. Five spot GPU instances were considered, ranging from the g2 family (1532 cuda cores) to the p2 family (4992 cuda cores). Table 1 presents our results. In this table and considering chromosome 20 in scenario S3, there was one migration (2 spot VMs). The cost was reduced from USD 3.13 (on-demand) to USD 1.14 (our framework). In this case, the execution time was increased in about 5 minutes. Considering that the execution takes about 6 hours, this is a small increase. In Bioinformatics laboratories, the researchers may execute several chromosome comparisons each day so a reduction in 2 dollars per comparison may lead to big savings.

Table 1. Framework results for comparing huge chromosomes [Brum et al. 2021]

Seq.	Simulated revocations					On-Demand VM	
	Scenario	Spot VMs	On-Demand VMs	Exec. time	Cost	Exec. time	Cost
chr19	S1	4	0	07:18:45	$1.39	04:56:49	$2.60
	S2	2	0	05:58:34	$1.18		
	S3	2.67	0	06:17:53	$1.31		
chr20	S1	4.33	0.67	08:35:08	$2.34	05:57:15	$3.13
	S2	2.67	0	07:35:34	$1.44		
	S3	2	0	06:02:27	$1.14		
chr21	S1	2	0	02:31:51	$0.46	02:31:39	$1.33
	S2	1.67	0	02:25:24	$0.43		
	S3	1	0	02:29:27	$0.39		
chr22	S1	2	0	03:19:12	$0.61	02:38:30	$1.39
	S2	1.33	0	02:50:13	$0.49		
	S3	1.33	0	02:47:40	$0.49		
chrY	S1	1.33	0	02:40:29	$0.44	02:45:52	$1.45
	S2	1	0	02:38:34	$0.41		
	S3	1.67	0	02:42:41	$0.49		

3.2.4. Multi-Agent Reasoning for instance provisioning in the AWS Cloud

MAS-Cloud+ [Mendes et al. 2024] is a framework that uses agents to select instances and execute applications with different characteristics in the AWS cloud, using one of three reasoning models: (a) heuristic model based on inference rules; (b) combinatorial optimization model solved with Integer Linear Programming (ILP); and (c) metaheuristic model based on Greedy Randomized Adaptive Search Procedure (GRASP). The goal is to reduce execution time, reduce resource waste and reduce cost. The framework includes the user interface for the application submission, cloud management with time/resource prediction, the VM provisioning/instantiation/monitoring/finish, and the application execution in a cloud provider, as shown in Figure 6.

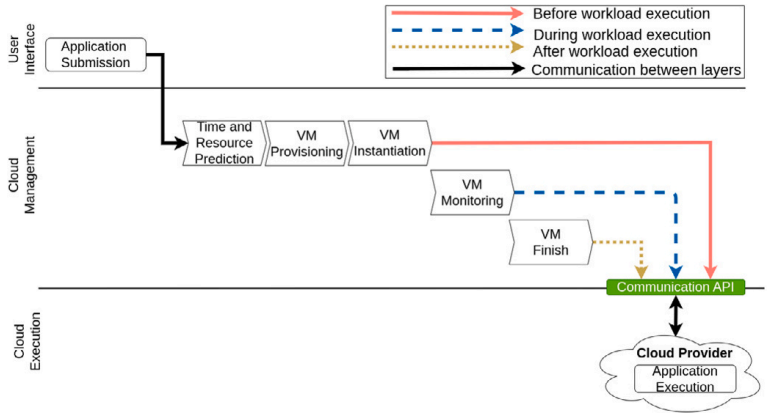


Figure 6. Workflow for application execution in MAS-Cloud+ [Mendes et al. 2024]

The parallel sequence comparison tool MASA-OpenMP [Sandes et al. 2016a] was used in the experiments, and many organisms were compared. Figure 7 presents the results for the human herpesvirus study (150K) and the plasmid study in the Rhizobium bacteria (500K). In the graphics, the first three set of bars in the left are random choices for VMs with different vCPUS. It can be seen that the random choices are not

appropriate and, for these two cases, the best model was the optimization model.

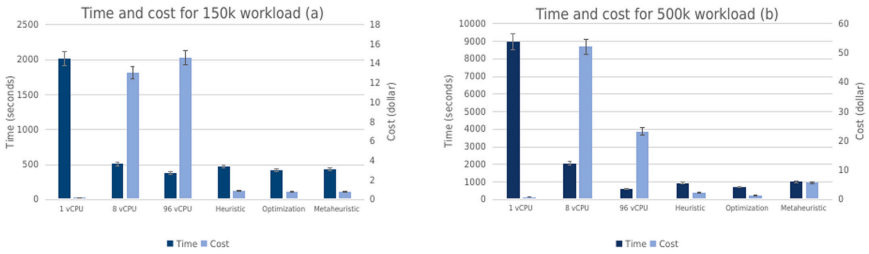


Figure 7. MAS-Cloud+ for the Bioinformatics study [Mendes et al. 2024]

3.2.5. Other Research Works in the Project

We have also conducted research on parallel biological sequence comparison with reconfigurable hardware, where the AWS FPGA (Field Programming Gate Array) F1 instance was used to execute a parallel code written on High Level Synthesis (HLS) [Jorge 2022].

In addition, a scheduler for federated learning applications for execution environments composed of multiple clouds was proposed and evaluated in [Brum 2024].

Finally, we investigated the use of function-as-a-service to execute parallel biological comparison applications in the cloud [Carvalho et al. 2023]. In this case, the AWS Lambda service was extensively used.

4. Conclusion

In this paper, we presented the main achievements of the project CNPq/AWS BioCloud. The project produced successful outcomes in both academic mentorship and scientific production. Our research generated important contributions to the fields of cloud management, cloud scheduling and biological sequence analysis, with an intensive use of parallel processing.

We would like to thank CNPq and AWS for their support of this project. AWS infrastructure allowed us to explore various types of architectures and VM instances in record time, greatly accelerating the acquisition of results and enabling us to advance our research at an unprecedented speed.

5. Acknowledgements

This work is supported by CNPq/AWS project n. 440014/2020-4.

6. Statement on the use of Artificial Intelligence

The authors state that they did not use any AI tool to write the text or to conduct the experiments.

References

Banimfreg, B. H. (2023). A comprehensive review and conceptual framework for cloud computing adoption in bioinformatics. *Healthcare Analytics*, 3.

Borin, E., Drummond, L. M. A., Gaudiot, J.-L., Melo, A. C. M. A., Alves, M. M., and Navaux, P. O. A. (2023). *High Performance Computing in Clouds: Moving HPC Applications to a Scalable and Cost-Effective Environment*. Springer Nature.

Brum, R. C. (2024). *Multi-FedLS: A Scheduler of Federated Learning Applications in a Multi-Cloud Environment*. PhD thesis, Graduate Program in Computer Science, Federal Fluminense University and Sorbonne University. Available online: https://theses.hal.science/tel-04812231v1.

Brum, R. C., Sousa, W. P., Melo, A. C. M. A., Bentes, C., Castro, M. C. S., and Drummond, L. M. A. (2021). A fault tolerant and deadline constrained sequence alignment application on cloud-based spot gpu instances. In *27th International Conference on Parallel and Distributed Computing, Euro-Par, Virtual*, pages 317–333.

Carvalho, L. R., Melo, A. C. M. A., and Araujo, A. P. F. (2023). Afmc: An alignment framework for multiple computing services and providers. *Concurrency Computation: Practice and Experience*, 35(18).

Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. J. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Figueiredo, M. A. C., Navarro, J. P., Sandes, E. F. O., Teodoro, G., and Melo, A. C. M. A. (2021). Parallel fine-grained comparison of long dna sequences in homogeneous and heterogeneous gpu platforms with pruning. *IEEE Transactions on Parallel and Distributed Systems*, 32(12):3053–3065.

Gupta, V., Sengupta, M., Prakash, J., and Tripathy, B. C. (2017). *Basic and Applied Aspects of Biotechnology*. Springer Nature.

Jorge, C. A. C. (2022). *Comparação paralela de sequências biológicas em plataformas de hardware uniformes e híbridas*. PhD thesis, Graduate Program in Informatics, University of Brasilia. Available online: https://repositorio.unb.br/handle/10482/45443.

Mendes, A. H. D. (2024). *Arquitetura multiagente com modelos de raciocínio distintos para gerenciamento de recursos em múltiplos provedores de nuvem*. PhD thesis, Graduate Program in Informatics, University of Brasilia. Available online: https://repositorio.unb.br/handle/10482/49786.

Mendes, A. H. D., Rosa, M. J. F., Marotta, M. A., Araujo, A. P. F., Melo, A. C. M. A., and Ralha, C. G. (2024). Mas-cloud+: A novel multi-agent architecture with reasoning models for resource management in multiple providers. *Future Generation Computer Systems*, 154:16–34.

Portella, G., Rodrigues, G. N., Nakano, E., and Melo, A. C. (2019a). Statistical analysis of amazon ec2 cloud pricing models. *Concurrency and Computation: Practice and Experience*, 31(18):e4451. e4451 cpe.4451.

Portella, G. J. (2021). *Precificacao em computacao em nuvem para instâncias permanentes e transientes : modelagem e previsao*. PhD thesis, Graduate Program in Informatics, University of Brasilia. Available online, https://repositorio.unb.br/handle/10482/41391.

Portella, G. J., Nakano, E. Y., Rodrigues, G. N., Boukerche, A., and Melo, A. C. M. A. (2024). A novel statistical and neural network combined approach for the cloud spot market. *IEEE Transactions on Cloud Computing*, 11(1):278–290.

Portella, G. J., Nakano, E. Y., Rodrigues, G. N., and Melo., A. C. M. A. (2019b). Utility-based strategy for balanced cost and availability at the cloud spot market. In *9th IEEE International Conference on Cloud Computing (IEEE CLOUD), Milan*, pages 214–218.

Ralha, C. G., Mendes, A. H. D., Laranjeira, L. A., Araujo, A. P. F., and Melo, A. C. M. A. (2019). Multiagent system for dynamic resource provisioning in cloud computing platforms. *Future Generation Computing Systems*, 94:80–96.

Sandes, E. F. O., Guillermo Miranda, X. M., Ayguade, E., Teodoro, G., and Melo, A. C. M. A. (2016a). Masa: A multiplatform architecture for sequence aligners with block pruning. *ACM Transactions on Parallel Computing*, 2(4).

Sandes, E. F. O., Miranda, G., Martorell, X., Ayguade, E., Teodoro, G., and Melo, A. C. M. A. (2016b). Cudalign 4.0: Incremental speculative traceback for exact chromosome-wide alignment in gpu clusters. *IEEE Transactions on Parallel and Distributed Systems*, 27(10):2838–2850.

Teylo, L. (2022). *Scheduling Deadline Constrained Bag-of-Tasks in Cloud Environments Using Hibernation Prone Spot Instances*. PhD thesis, Graduate Program in Computer Science, Federal Fluminense University. Available online: http://200.20.15.38/ ic2022/teses-e-dissertacoes/.

Teylo, L., Arantes, L., Sens, P., and Drummond, L. M. A. (2023). Scheduling bag-of-tasks in clouds using spot and burstable virtual machines. *IEEE Transactions on Cloud Computing*, 11(1):964–982.

Teylo, L., Nunes, A. L., Melo, A. C. M. A., Boeres, C., de A. Drummond, L. M., and Martins, N. F. (2021). Comparing sars-cov-2 sequences using a commercial cloud with a spot instance based dynamic scheduler. In *21st IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), Virtual*, pages 247–256.