Optimizing HPC Deployment: Enhancing Accessibility and Efficiency through the OMPC Framework

Leonardo Benicio[1], **Emilio Francesquini**[2], **Alfredo Goldman vel Lejbman**[1]

[1] Department of Computer Science – University of Mathematics and Statistics
University of São Paulo, São Paulo, Brazil.

[2]Department of Computer Science – Federal University of ABC
São Paulo, Brazil.

{lbenicio,gold}@ime.usp.br, e.francesquini@ufabc.edu.br

Abstract. *As the need for parallel and distributed computing systems grows, effectively developing, deploying, and maintaining high-performance computing (HPC) applications becomes essential. This study addresses the HPC community's challenges, focusing on enhancing the use of low-level frameworks like MPI for optimal performance. We explore the Open MP Cluster (OMPC) framework to improve understanding and utilization of HPC applications. By reviewing current approaches to overcoming these framework complexities, we contribute to discussions on making HPC more accessible and efficient. Our findings aim to provide researchers and developers with the knowledge to fully exploit HPC capabilities, promoting innovation in the realm of parallel and distributed systems.*

1. Introduction

The evolution of hardware development consistently amplifies computational power, notably through a steady rise in processing units per chip and an expansion of the number of cluster nodes. This surge in accessible processing power has sparked a heightened interest in exploring parallel and distributed computing solutions. The process of constructing and managing such computing clusters, whether situated on-premises or hosted in the cloud, presents a multifaceted challenge. This complexity arises from the intricate interplay of numerous components, making these systems difficult to comprehend and maintain.

Reflecting on the intricacies of managing high-performance clusters, an article detailing the experiences at the University of Luxembourg illustrates this point. According to the authors [Varrette et al. 2014], orchestrating such an endeavor requires a diverse array of skills and tools to circumvent the prevalent bottleneck of excessive manpower demands. The investment extends beyond labor, encompassing the significant costs associated with the necessary infrastructure. The authors advocate for the strategic use of cloud spot instances to mitigate hardware expenses and recommend employing frameworks like the Message Passing Interface (MPI) for their ease of use and ability to leverage the infrastructure's full capabilities.

Further corroborating this perspective, another study [Anderson et al. 2017] demonstrates the potential for a substantial performance improvement — ranging from $3\times$ to $70\times$ — by transitioning a Spark application to an architecture underpinned by MPI. This study not only highlights the performance advantages but also outlines a strategic approach for migrating workflows from traditional Apache Spark deployments to those optimized through MPI, showcasing the practical benefits of such technological advancements.

2. Background

The deployment of high-performance computing (HPC) workloads demands a deep knowledge in infrastructure as code, enabling cloud resource and hardware management using file formats like TOML, JSON, and YAML. This knowledge allows the management of application workloads, encompassing task initiation, resumption, completion, results aggregation, and the development of parallel and distributed code to fully utilize the cluster's capabilities.

Integrating tools such as Terraform for infrastructure, Slurm for workload management, and MPI for application development addresses these requirements, yet challenges arise without a unified framework for cohesive management of cluster infrastructure and applications. The limited use of MPI's advanced features [Laguna et al. 2019] highlights a disconnect between available technology and its application, suggesting the need for an abstraction layer to simplify deployment and enhance efficiency. This need extends to integrating MPI with big data analytics tools like Apache Spark and Hadoop [Anderson et al. 2017].

Efforts to overcome these challenges, such as the toolkit developed by [Munhoz and Castro 2022], aim to simplify cluster and workload management, utilizing cloud advantages like spot instances for cost-efficiency. This progress underscores ongoing endeavors to make HPC deployment more accessible and straightforward, addressing both the technical obstacles and the broader goal of democratizing high-performance computing resources for wider application and innovation.

3. An alternative approach

3.1. Deploying a hello world application

The development of a high-performance computing (HPC) application presents developers to a myriad of conventional and innovative solutions. One such approach involves utilizing OMPC, a tool known for its ease-of-use in HPC application development. For beginner developers, OMPC offers an accessible starting point through its comprehensive documentation and all-in-one Docker images. These images serve as a robust foundation, including pre-compiled binaries, thus allowing developers to clone, compile and execute a sample application in a straightforward manner.

Transitioning from a local setup to a cloud-based environment marks the next phase in deploying HPC applications. Initially, this process begins on a single bare-metal machine to closely mirror the Hello World example within a new setting. This involves cloning the main repository, installing dependencies, and compiling the required binaries. The simplicity of running a one-node application paves the way for more complex endeavors—namely, replicating this environment across multiple machines. Achieving this requires the creation and deployment of an image from the initial machine onto others, subsequently linking these machines within a unified virtual network to facilitate seamless communication.

With the infrastructure now in place, testing the Hello World application on this networked setup is the immediate goal. This is accomplished by directing the MPI execution binary to the local network IP addresses, thus enabling distributed system task execution. This achievement signifies a pivotal moment in the journey towards deploying substantive workloads. However, advancing beyond this point introduces a new set of challenges, necessitating the resolution of increasingly complex problems as developers aim to harness the full potential of distributed computing power.

This alternative approach not only provides a pragmatic roadmap for developing and deploying HPC applications but also highlights the inherent scalability and adaptability required to navigate the complexities of cloud-based environments. As we delve deeper into the nuances of such deployments, the necessity for innovative solutions and frameworks becomes ever more apparent, underscoring the dynamic interplay between technology, infrastructure, and application development in the realm of high-performance computing.

3.2. Advancing to a real-world scenario

Transitioning an HPC application from development to production introduces significant complexities, particularly in workload management and dynamic resource allocation. The risk of failures within the cluster necessitates a robust system for managing interruptions, demanding capabilities beyond those offered by the MPI run binary. Slurm emerges as a critical solution, facilitating not just task deployment across hosts but also enabling sophisticated management of task states, aggregation of results, and efficient output generation.

Addressing the static nature of traditional cluster setups reveals another layer of complexity, especially when moving to cloud environments. This shift requires moving away from a fixed infrastructure model towards a more flexible, dynamic resource management approach. Such flexibility is crucial for leveraging the cloud's scalability and cost-efficiency, where resources are provisioned and decommissioned in response to real-time demands. Terraform supports this transition by automating the provisioning of compute, network, and storage resources, allowing for a more streamlined setup process using pre-configured images.

The integration of these strategies leads to the development of a comprehensive workflow that automates the entire lifecycle of an HPC cluster. This includes the initial creation of the cluster, deployment of workloads, and the final teardown, ensuring optimal resource utilization and cost savings. This approach not only ensures the operational efficiency of the HPC applications but also embodies the principles of cloud-native computing by embracing the transient nature of cloud resources, including the strategic persistence of only essential storage solutions post-computation.

By tackling these advanced deployment challenges with strategic use of workload managers like Slurm, infrastructure as code through Terraform, and automated lifecycle management, developers can effectively navigate the intricacies of bringing HPC applications to the production stage in cloud environments. This method enhances operational resilience, scalability, and leverages the inherent cost-effectiveness of the cloud, marking a notable advancement in the field of high-performance computing application deployment.

4. Conclusion

In this study, we delineated a structured approach for deploying high-performance computing (HPC) applications in cloud environments, emphasizing a three-tiered architecture. Initially, we leverage Terraform for infrastructure automation, followed by establishing an Open MPI cluster for uniform computing environments, and culminating in the application code enhanced with Open MP for maximizing parallel computing. This method, supported by Bash and Python for their comprehensive libraries, addresses the deployment and management challenges of HPC clusters, aligning with the observations by [Laguna et al. 2019] on MPI complexities and [Varrette et al. 2014] on the operational demands of such infrastructures.

The implementation of the framework suggested by [Munhoz and Castro 2022] has significantly streamlined this process, demonstrating the efficacy of cloud computing in enhancing scalability and cost-efficiency for HPC applications without the burden of its inherent complexities.

Nonetheless, our analysis indicates an urgent need for ongoing advancements towards a more robust and user-friendly platform. Aiming to mirror the convenience offered by Platform as a Service (PaaS) for web development, we advocate for a platform that simplifies HPC application development, deployment, and management, making it more accessible to a wider research and development audience. This endeavor not only reinforces the viability of utilizing cloud infrastructure for sophisticated computational tasks but also emphasizes the importance of an ecosystem that facilitates a straightforward HPC deployment process, thereby broadening the accessibility and efficiency of high-performance computing for researchers and developers.

References

Anderson, M., Smith, S., Sundaram, N., Capotă, M., Zhao, Z., Dulloor, S., Satish, N., and Willke, T. L. (2017). Bridging the gap between hpc and big data frameworks. *VLDB Endowment*, 10(8):901–912.

Laguna, I., Marshall, R., Mohror, K., Ruefenacht, M., Skjellum, A., and Sultana, N. (2019). A large-scale study of mpi usage in open-source hpc applications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '19, New York, NY, USA. Association for Computing Machinery.

Munhoz, V. and Castro, M. (2022). Hpc@cloud: A provider-agnostic software framework for enabling hpc in public cloud platforms. In *Anais do XXIII Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 157–168, Porto Alegre, RS, Brasil. SBC.

Varrette, S., Bouvry, P., Cartiaux, H., and Georgatos, F. (2014). Management of an academic hpc cluster: The ul experience. In *2014 International Conference on High Performance Computing & Simulation (HPCS)*, pages 959–967.