ISSN No:-2456-2165

Programming Language for Data Intensive HPC Applications: Applications and its Relevance

Okwu, Hachikaru Ngozi
Research Scholar, Department of Computer Science,
Ignatius Ajuru University of Education,
Rivers State, Nigeria.

Ojekudo, Nathaniel Akpofure
HOD of Computer Science,
Ignatius Ajuru University of Education,
Rivers State, Nigeria.

Abstract:- HPC is the bedrock upon which scientific, industrial, and societal progress is built. Organizations require lightning-fast, highly reliable IT infrastructure to process, store, and analyse massive amounts of data. The ability to process data in real time is critical for many applications, including streaming a live sporting event, tracking a developing storm, testing new products, and analysing stock trends. This study focuses on the relevance, and applications of programming languages for data-intensive Higher Performance Computing (HPC) to helps complete task faster and the applications they are used in. The research will serve as a basis for future research and development in programming languages for data-intensive HPC applications.

I. INTRODUCTION

The use of parallel processing for advanced application programs is referred to as high-performance computing (HPC). The term refers to systems that operate at speeds greater than a teraflop, or 1012 floating-point operations per second. Scientific researchers, engineers, and academic institutions are the most common users of HPC systems. HPC is also used by some government agencies, particularly the military, for complex applications. As the demand for processing power and speed grows, high-performance computing (HPC) will likely pique the interest of businesses of all sizes. High-performance computing (HPC) is use in processing data and performing complex calculations at high speeds, these system needs a lightning-fast, highly reliable IT infrastructure to process, store, and analyse massive amounts of data in other to get the task complete and achieve it at a faster rate. For it to work well, the system needs servers that are network together into cluster, software programs and algorithms are ran simultaneously on the servers in the cluster and this cluster is networked to the data storage to capture the output, therefore programming languages are so important to achieving this task. Programming languages are important in HPC because its helps compute data faster and more efficiently. Note that portability, performance, and usability are the most desired features of programming languages for data-intensive HPC applications.

II. RELATED WORKS

This study reviewed a Systematic Mapping Study (SMS), it showed how different programming languages for data intensive HPC applications differ in terms of Portability, performance, and usability. They carried out the study in two parts, firstly, related articles are found in eight digital libraries used an automated keyword-based search in which they used 420 sample papers then reduced it to 152 relevant articles published between 2006 and 2018 in the second phase. They were able to identify 26 programming languages from the analysis of these articles, which were mentioned in 33 of them. They compared the mapping study's findings to their 57 HPC expert's questionnaire-based survey. From their results they found that portability, performance, and usability are the most desired features of programming languages for data-intensive HPC applications and noticed that majority of programming languages used are text-based general-purpose programming languages. Also, noted that they have a steep learning curve, making them difficult to implement (Amaral et al., 2019).

Filguiera et al. (2019) introduced Dispel4py, a replacement Python framework that permits you to explain abstract stream-based workflows for distributed data-intensive applications. Data streaming improves performance, prototyping speed, and applicability to real-time observations. They presented four dispel4py mappings, three application domains, and measurements on multiple infrastructures demonstrated the optimisations achieved in providing demanding real-world applications and developing effective training. Dispel4py.org is an open-source project to which they have invited people to contribute. The efficient mapping of dispel4py to multiple target infrastructures demonstrates the use of data-intensive and high-performance computing (HPC) architectures, as well as consistent scalability.

This research looked at the evolution of computers at various levels, from desktop to large-scale supercomputers, as well as their computational power. HPC has opened up new avenues for simulations of relevant biological systems and applications in bioinformatics, biological and computational chemistry. Their research demonstrates the trends in this research field as well as its future prospects (Peréz-Sánchez et al., 2015).

Götz et al. (2017) discussed the potential for applying software engineering techniques to an emerging field in high performance computing, namely large-scale data analysis and machine learning. they argue for the employment of software engineering techniques in the development of such applications from the start, and the design of generic, reusable components. They demonstrated how, using the Juelich Machine Learning Library (JuML), a framework like this can not only simplify the design of new parallel algorithms, but also increase the productivity of the actual data analysis workflow. They are particularly interested in abstraction from heterogeneous hardware, architectural design, and aspects of parallel and distributed unit testing.

Bridges will be a new high-performance computing resource that combines advanced memory technologies with a user-centered, data-centric environment. Bridges will bring desktop-like functionality to HPC, connect campuses, and power complex workflows. It will consist of three tiers of processing nodes, each with 128GB, 3TB, and 12TB of hardware-enabled coherent shared memory. It is intended to support memory-intensive applications and ease of use, as well as persistent database and web nodes, as well as login, data transfer, and system management nodes. The system will be a resource on XSEDE, the National Science Foundation's Extreme Science and Engineering Discovery Environment (Nystrom et al., 2015).

Tuncer et al. (2017) presented a new framework for automatically diagnosing performance issues in HPC systems using machine learning. Its framework makes use of data gathered from application runs on resource usage and performance counters. To significantly reduce the overall computational burden of the technique, the collected time-series data were first converted into statistical features that retain applications. Machine learning algorithms were used to learn the anomaly features of these historical data and to identify the types of anomalies observed during application execution. They tested their framework on an HPC cluster as well as a public cloud, and found that it outperforms current state-of-the-art techniques in detecting anomalies, with an F-score of more than 0.97.

Conductor, a run-time system that intelligently distributes available power to nodes and cores to enhance performance, is introduced during this paper. Configuration space exploration and adaptive power balancing are the main techniques employed. Based on a hardware-enforced power bound, configuration exploration dynamically selects the best thread concurrency level and DVFS state. Adaptive power balancing efficiently determines critical paths and distributes more power to those paths. Greater power, in turn, allows for higher thread concurrency levels, DVFS states, or both. They described these techniques in detail and demonstrated that, when compared to the state-of-the-art technique of using statically predetermined, per-node power caps, Conductor results in a best-case performance improvement of up to 30% and an average improvement of 19.1 percent (Marathe, 2015).

Rethinagiri et al. (2015) presented two novel real-time heterogeneous platforms with three types of devices (CPU, GPU, and FPGA) in this paper. One platform is designed for efficiently accelerating computation-intensive applications. The second platform is intended for high-performance real-time embedded system applications. They carried out experiments with five real-time and high-throughput computation-data-intensive applications: cone beam computed tomography, face recognition, number plate recognition, and motion tracking. They were able to achieve an average speed-up of 21x when compared to a CPU-GPU platform, and 24x when compared to a CPU-GPU platform. When compared to a CPU-FPGA platform, and 70x when compared to quad-core CPU execution alone. We also implemented these applications by using a single programming language (OpenCL) on the trigonous platforms and achieved 6x of speed-ups on average over the quad-core setup.

III. HIGH PERFORMANCE COMPUTING (HPC)

3.1 Programming languages of HPC

HPC is the bedrock upon which scientific, industrial, and societal progress is built. Organizations require lightning-fast, highly reliable IT infrastructure to process, store, and analyse massive amounts of data. The ability to process data in real time is critical for many applications, including streaming a live sporting event, tracking a developing storm, testing new products, and analysing stock trends (NetApp, 2021). The performance or the speed at which data is to be processed is also depended on the programming language used, better programming language reduces programming time in some experiments (for non-expert programmers and short programs). The most desired features of programming languages for data-intensive HPC applications are Portability, performance, and usability which is the drive needed to exploit the performance of a new computer architecture. Programming languages commonly used by HPC are:

Fortran: is the first high-level programming language, it was created more than 50 years ago and is still used in many high-performance computing applications today. There are several possible explanations for this, one of which is that the language was designed to closely resemble Fortran algorithms. Fortran has proven to be highly resistant to new advances in other programming languages over the years. New versions are constantly being released and linked to ANSI specifications, ensuring that an application written for one operating system can be compiled and run on other operating systems using different compilers resulting in its code performing well in HPC due to some memory handling constraints and they are computationally fast. Its ability to generate efficient and optimized code is a differentiator between different compilers from different vendors; developers can determine which compiler is used results in the fastest execution time. This includes not only code generation, but also determining which loops can be vectorized. It is critical that compilers recognize where there are opportunities for optimization and that the part of the application is executed on more performance-oriented hardware (Michael, 2016).

C and C++: They are object-oriented programming languages with a high level of programming comfort. The compiler has a difficult time optimizing C because it has few syntax constraints. They are used by HPC experts because of their portability, performance and usability for the computationally intense bits. The vast majority of new HPC application code is written in modern C ++. C ++ is the only programming language that provides a high-performance solution across many different platforms. Modern C ++ has a number of modern language features, such as templates and Lambdas. All of these things are really important in performance-sensitive spaces, which HPC experts say are important for the future of the industry (Staff, 2020).

Python: Python has grown in popularity as a programming language for a variety of activities, including High Performance Computing (HPC). Python's code readability is one of its strongest advantages, enabling programs to be preserved and enhanced over time. Python is an object-oriented programming language that supports a wide range of programming models. Automatic memory management is included, as well as a broad and extensive standard library. It's lightweight and can run on a range of operating systems, including modern HPC-focused heterogeneous systems. Since its inception more than 25 years ago, the Python programming ecosystem has been continually developed and expanded. Although Python is free and open source, there are commercial compilers and software that can help you get more done. It has expanded over time to include scripting and other features that HPC developers may find useful. Python contains exponential functions and a matrix multiply feature that are extremely useful in broad scientific applications, in addition to the regular C operators. It also has a wide library of mathematical functions that can help developers build large applications faster. Python has been one of the most popular programming languages for over a decade because of its ease of use and extensibility. Python is supported by numerous implementations and development environments.

Python can be used to build a wide range of high-performance computing applications. While tight loops may still necessitate C or FORTRAN coding, Python can be used. As more devices with coprocessors or accelerators become operational, Python can be used to offload the main CPU and take advantage of the coprocessor (Michael, 2016).

3.2 Areas of Applications

Bioinformatics: it is the study of biological data. This multidisciplinary field of science includes biology, computer science, mathematics, and statistics. Bioinformatics is used to analyse large amounts of data and make sense of it all in a world where data is generated at a faster rate than we can process it therefore HPC is used in bioinformation. Molecular docking is a bioinformatic modelling technique that is used in drug development and structural molecular biology. This method is used to predict the most likely "binding scenarios" between a protein and a ligand based on their three-dimensional structure (Alejandra, 2018). Virtual screening (VS) is an important bioinformatics tool used in drug development. This method is used to identify the compounds/molecules/ligands that are most likely to bind to our target protein's receptor. The target protein's structure contains several pockets. Among certain ligands is a binding pocket (Muniba, 2021).

Health care: medical practitioner uses computers to store patient's data by tracking vital signs, analyse drug's efficacy and genome sequencing. It is used for creating virtual human heart, researchers benefit from HPC because it controls more than 25million variables used or testing drugs, implications after cardiovascular surgeries, testing different pacemaker insertion protocol. It enables researchers combed through petabytes of data to look for links between a cancer patient's genome and the composition of their tumors, aids in the art of refining cancer treatments, diagnosing diseases and organising a treatment plan (Mae, 2019).

Engineering: engineers use HPC to test design prototypes with massive computer simulations. It used in testing the functionality of airplane parts and streamlining racing bike frames.

Oil sector: In this field the drilling specialist apply HPC system to accurately spot areas where to drill for new wells and assist in developing production in older wells, which saves time and cost.

Urban planning: Cities around the world use information derived from large amounts of data to plan cities. This information includes weather, traffic patterns, and noise levels. It assists them in controlling long-term issues such as climate change and infrastructure requirements. Because of the large amount of data that must be analysed, HPC is required.

Research Centres: Scientists are always time conscious within the scope of work their depending on the urgency for instance during the pandemic, so the make use of HPC to discover sources for materials needed, in practical trials, experimental discovery and study new discoveries. The application of HPC is not limited to one area of research but diverse in any field, ranging from Engineering, robotics, biologically species, artists, geoscientists, and social media analysts.

Machine learning: with the growth in technology there are also drawbacks associated in it which we cannot allow limit the industry breakthrough technology is providing. HPC is used in detecting frauds that is on the rise on credit cards, provide technical assistance in the IT world, and aid in understanding artificial intelligence.

Medical Treatment: This computing system is applied in the improvement and advancement for long term or fatal ailment such as diabetes, dementia, and cancer to enable faster, accessible, and accurate treatment plan.

Entertainment: In the media industry, the videographers, directors used this system professionally to edit movies and include special effects into the films that attract high rating from consumers.

Note that, HPC is used and required in our daily lives to solve problems and invent new things to help and improve humanity.

IV. FINDINGS

The use of parallel processing for advanced application programs is referred to as high-performance computing (HPC). HPC is the bedrock upon which scientific, industrial, and societal progress is built. To process, store, and analyze massive amounts of data, organizations require lightning-fast, highly reliable IT infrastructure. Portability, performance, and usability are the most desired features of programming languages for data-intensive HPC applications.

Developers of high-performance computing (HPC) typically have two main objectives. The first is application efficiency (quicker time to results), and the second is development ease. Fortran is the first high-level programming language and it is used high-performance computing applications today because its language was designed to closely resemble Fortran algorithms. Fortran has proven to be highly resistant to new advances in other programming languages over the years by ensuring that an application written for one operating system can be compiled and run on other operating systems using different compilers resulting in its ability to generate efficient and optimized code is a differentiator between different compilers from different vendors which makes developers to determine which compiler is used results in the fastest execution time. C and C++ are object-oriented programming languages with a high level of programming comfort. C ++ is the only programming language that provides a high-performance solution across many different platforms. They are used by HPC experts because of their portability, performance and usability for the computationally intense bits.

Python has grown in popularity as a programming language for a variety of activities, including High Performance Computing (HPC). Python's code readability this enables programs to be preserved and enhanced over time. It is an object-oriented programming language that supports a wide range of programming models. It has an automatic memory management and extensive standard library. It's lightweight and can run on a range of operating systems, including modern HPC-focused heterogeneous systems. It has scripting and other features that HPC developers may find useful. Python contains exponential functions and a matrix multiply feature that are extremely useful in broad scientific applications. It also has a wide library of mathematical functions that can help developers build large applications faster. It is one of the most popular programming languages because of its ease of use, extensibility and is supported by numerous implementations and development environments. HPC can be used in all areas of the human existence because it is used to find solutions to problems and in the creation and testing of new technologies and innovations. Some application areas are in bioinformatics, health care, engineering, oil sector, urban planning, research Centres, machine learning, medical Treatment and entertainment.

V. CONCLUSION

High-performance computing is the bedrock upon which scientific, industrial, and societal progress is built. To process, store, and analyse massive amounts of data, organizations require lightning-fast, highly reliable IT infrastructure. The first high-level programming language was Fortran, Fortran has proven to be highly resistant over the years because it ensures that an application written for one operating system can be compiled and run on other operating systems using different compilers, resulting in its ability to generate efficient and optimized code being a differentiator between different compilers from different vendors, making it difficult for developers to determine which compiler to use. C and C ++ are the only programming languages that offer a high-performance solution on a wide range of platforms. HPC experts use them because of their portability, performance, and usability for computationally intensive bits. Python's code readability allows programs to be saved and improved over time. It is object-oriented and supports a wide variety of programming models. It has automatic memory management and a large standard library. Python includes exponential functions and a matrix multiply feature, both of which are extremely useful in a wide range of scientific applications. It also includes a large library of mathematical functions that can assist developers in building large applications more quickly. HPC can be applied to all aspects of human life because it is used in the development and testing of new technologies and innovations. Bioinformatics, health care, engineering, the oil sector, urban planning, research centres, machine learning, medical treatment, and entertainment are some of the application areas. The term High Performance Computing (HPC) is also used to refer to the use of parallel processing in scientific and industrial applications.

REFERENCES

[1]. Alejandra, Rodriguez Sosa. (2018, October 13). *Molecular Docking: Bioinformatics in Drug Discovery | TSC*. https://theskepticalchemist.com/molecular-docking-bioinformatics-drug-discovery/

[2]. Amaral, V., Norberto, B., Goulão, M., Aldinucci, M., Benkner, S., Bracciali, A., Carreira, P., Celms, E., Correia, L., Grelck, C., Karatza, H., Kessler, C., Kilpatrick, P., Martiniano, H., Mavridis, I., Pllana, S., Respício, A., Simão, J., Veiga, L., & Visa, A. (2019). Programming Languages for Data-Intensive HPC Applications: A Systematic Mapping Study. *Parallel Computing*, *91*, 102584.

[3]. Filguiera, R., Krause, A., Atkinson, M., Klampanos, I., & Moreno, A. (2017). dispel4py: A Python framework for data-intensive scientific computing. *The International Journal of High-Performance Computing Applications*, *31*(4), 316–334. https://doi.org/10.1177/1094342016649766

[4]. Götz, M., Book, M., Bodenstein, C., & Riedel, M. (2017). Supporting Software Engineering Practices in the Development of Data-Intensive HPC Applications with the JuML Framework. *Proceedings of the 1st International Workshop on Software Engineering for High Performance Computing in Computational and*

Data-Enabled Science & Engineering, 1–8. https://doi.org/10.1145/3144763.3144765.

[5]. Mae Rice. (2019, November 4). *14 High Performance Computing Applications & Examples*. Built In. https://builtin.com/hardware/high-performance-computing-applications

[6]. Marathe, A., Bailey, P. E., Lowenthal, D. K., Rountree, B., Schulz, M., & de Supinski, B. R. (2015). A Run-Time System for Power-Constrained HPC Applications. In J. M. Kunkel & T. Ludwig (Eds.), *High Performance Computing* (pp. 394–408). Springer International Publishing. https://doi.org/10.1007/978-3-319-20119-1_28

[7]. Michael, S. (2016, September 22). Fortran for HPC. *InsideHPC*. https://insidehpc.com/2016/09/fortran-for-hpc/

[8]. Michael, S. (2016, September 1). Python and HPC. *InsideHPC*. https://insidehpc.com/2016/09/python-and-hpc/

[9]. Muniba Faiza. (2021, March 21). What is Virtual Screening in Bioinformatics? *Bioinformatics Review*. https://bioinformaticsreview.com/20210321/what-is-virtual-screening-in-bioinformatics/

[10]. NetApp (2021). *What Is High-Performance Computing (HPC)? How It Works | NetApp*. https://www.netapp.com/data-storage/high-performance-computing/what-is-hpc/

[11]. Nystrom, N. A., Levine, M. J., Roskies, R. Z., & Scott, J. R. (2015). Bridges: A uniquely flexible HPC resource for new communities and data analytics. *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, 1–8. https://doi.org/10.1145/2792745.2792775

[12]. Peréz-Sánchez, H., Fassihi, A., Cecilia, J. M., Ali, H. H., & Cannataro, M. (2015). Applications of High-Performance Computing in Bioinformatics, Computational Biology and Computational Chemistry. In F. Ortuño & I. Rojas (Eds.), *Bioinformatics and Biomedical Engineering* (pp. 527–541). Springer International Publishing. https://doi.org/10.1007/978-3-319-16480-9_51

[13]. *Programming Languages—HPC Wiki*. (n.d.). Retrieved 24 March 2021, from https://hpc-wiki.info/hpc/Programming_Languages

[14]. Schmuker, Michael. (2015). Re: Which programming language support big data analysis and High performance computing both? Retrieved from: https://www.researchgate.net/post/Which-programming-language-support-big-data-analysis-and-High-performance-computing-both/54c0ef46d5a3f240688b4672/citation/download.

[15]. staff. (2020, May 7). Podcast: A Shift to Modern C++ Programming Models. *Inside HPC*. https://insidehpc.com/2020/05/podcast-a-shift-to-modern-c-programming-models/

[16]. Tuncer, O., Ates, E., Zhang, Y., Turk, A., Brandt, J., Leung, V. J., Egele, M., & Coskun, A. K. (2017). Diagnosing Performance Variations in HPC Applications Using Machine Learning. In J. M. Kunkel, R. Yokota, P. Balaji, & D. Keyes (Eds.), *High Performance Computing* (pp. 355–373). Springer International Publishing. https://doi.org/10.1007/978-3-319-58667-0_19

[17]. Rethinagiri, S. K., Palomar, O., Moreno, J. A., Unsal, O., & Cristal, A. (2015). Trigeneous Platforms for Energy Efficient Computing of HPC Applications. *2015 IEEE 22nd International Conference on High Performance Computing (HiPC)*, 264–274. https://doi.org/10.1109/HiPC.2015.19