High Performance Computing

Authors: Bernardo Augusto, Marcelo Pereira and Daniel Marçal

Student number: 2814, 2691, 2675

Curricular Unit: High Performance Computing

Teachers: Norberto Albino and Laercio Júnior

Index

Introduction

HPC is a big study field, where we can find a lot of different areas and subfields. In this project, we focused on choosing the ones we consider the most interesting and the most enjoyable for us to share with. We start by tanking an approach on “HPC supercomputers and cluster solutions”, there will exploit an explanation of what’s a supercomputer, the differences between supercomputers and cluster computers, some cluster solutions and for last will be presented an affordable cluster solution using PS3 system, its advantages, and its disadvantages. Then “GPU, CUDA, OpenCL and CPU & GPU Coprocessing” and this one was chosen because we already talked in class about it and professor Norberto had an idea to work with CUDA, so we decided to learn how it really worked. The next subject is “HPC on Cloud and his Evaluation”, where we discuss metrics to see of an organization benefits moving to the cloud or not. And finally, the “Trends and Methods on parallel computing in HPC”, reviewing and classifying the best methods of nowadays use of parallelization.

The method used to make the project was SLR or Systematic literature review, which identifies, assesses, and interprets all findings on a research topic. SLR includes the search process, inclusion criteria, and analysis of the findings to answer research questions.

HPC supercomputers and cluster solutions

What is a supercomputer? A supercomputer is a computer that can achieve a far superior level of performance compared to a normal use computer. Supercomputers are used in many different fields besides computer science like quantum mechanics, molecular modeling, physical simulations, etc. They are used to solve specific problems. These machines tend to generate an enormous amount of heat due to its extreme power consumption and performance.

Most present supercomputers are cluster computers, but this has not always been the case. A computer cluster is a set of loosely or tightly connected computers that work together so that, in many respects, they can be viewed as a single system. In a computer cluster, these machines have each node set to perform the same task, controlled and scheduled by software. This is the reason why they are supercomputers since they are used for specific purposes, unlike general use computers that are used for a variety of tasks. The components of a cluster are usually connected through fast local area networks, with each node (computer used as a server) running its instance of an operating system. Clusters are usually deployed to improve performance and availability over that of a single computer while being more cost-effective than single computers of comparable speed or availability.

In the past Cluster computer and supercomputers were different. Cluster computers are loosely coupled parallel computers where the computing nodes have individual memory and instances of the operating system, but typically share a file system, and use an explicitly programmed high-speed network for communication. Generally, Supercomputers have a very high-performance interconnect. However, since modern datacenters can also have high performance interconnects this is no longer enough to make a clear distinction. A possible distinction could be the application. Whether the application is HPC (high-performance computation) or just large-scale computation (web-serving, big data, etc..).

There has been toon of initiatives to multiply the use of HPC among small users or users with specific requirements. These initiatives range from building a virtual cluster to the development of specialized supercomputers. Affordable and accessible supercomputing solutions have been in demand for several decades. To attract newer scientific areas for High-Performance Computing and bring scientists and researchers who are involved in HPC computing the HPC-Europa2 (2009 - 2012) project built a live DVD through which a regular computer can be bootstrapped into a virtual cluster under a hypervisor. It provides a platform to learn parallel programming with-state-of-the-art and libraries. Consequently, it provides an e-learning and teaching platform to the area of HPC.

Affordable high-performance cluster systems had been designed and implemented based on the PlayStation 3 (PS3) Cell microprocessors. This solution has a low cost when compared to the standard HPC cluster. Sony enabled the installation of Linux on the PS3 in a dual-boot setup. The GigE card is accessible to the Linux kernel through the hypervisor (computer software, hardware, and firmware that creates and runs virtual machines), which both makes it possible to turn the PS3 into a networked workstation and facilitates building PS3 clusters via network switches. This is also another advantage, of the many advantages, of this solution.

Although, PS3 has a lot of advantages, and more than the ones mentioned in this paper, it also has its limitations. One limitation that is Sony disabled the ability to run other OS, in 2010. In terms of the main memory size, the problem relies on the fact that the ps3 is equipped with only 256 MB of main memory. This can be seen as an enormous negative point because it is combined with poor network interconnection. Another limitation is that this high performance comes with a cost. It is hard to write fast and efficient code for this CELL processor since it requires a deep understanding of the processor architecture and some experience. This is a limitation because it’s not a common processor and so its architecture is not known by many.

In conclusion, the ps3 is an attractive solution for scientific computations but it relies on the application’s characteristics. For some applications, the limitations listed will label the ps3 as a not very good solution for clustering and some scientific computations.

GPU, CUDA, OpenCL and CPU & GPU Coprocessing

The GPU era started in 1999 with the launch the of the GeForce 256 which was a single chip 3D with real-time graphics. GPU stands for Graphics Processing Unit and it is commonly used to render images, videos, 3D animations and most importantly it frees up the CPU capacity to do other things as it can make very fast calculations. They got better and better over the years until 2006 when the first unified graphics and computing GPU was developed. It was the first to use CUDA, a platform developed by Nvidia for parallel computing which enables ways to speed up the computation. Although GPUs kept progressing over the years until how we know them today, CUDA stayed relevant and has played a big part in deep learning because it has helped to build the fastest computers ever made. Some of CUDA's most important are applications are: Computational finance, data science and analytics, deep learning, machine learning, media, among many others and of course, the most obvious, high end computing.

The competitor of CUDA is OpenCL and this was developed by Apple to match the needs for a platform that would not only be compatible with Intel/AMD CPU's with Nvidia developed GPU's but to match any combination of components, a more heterogenous and standard platform, if you may. The main advantage here is that for the general user he is not obliged to buy components that fits the needs of CUDA but from the programmer's perspective it is harder to code so there are some disadvantages as well.

As previously mentioned, GPUs can free up the CPU for other processes and this type of interaction between these two components is an ongoing complementary task that makes the applications run at its maximum potential. But why is that? CUDA programs execute with coprocessing tasks and what this means is that we can get the best performance out of the program. The serial parts of the program are run in the CPU, which was optimized for low latency on a single thread, while the parallel portions are executed on the GPU. Overall, we're getting the best out of each core because each core is doing precisely what it was designed to do to its maximum capability.

In the following table we can see a comparison between four different component's configurations with usage of Amdahl's law, which gives us theoretical speedup in latency of the demanded task.

	Configuration	Processing time for 1 CPU core	Processing time for 500 GPU cores	Processing time for 10 CPU cores	Processing time for 1 CPU core + 450 GPU cores
Program type	**Area**	**50**	**500**	**500**	**500**
Parallel-intensive program	0.5% serial code	1.0	5.0	1.0	1.00
	99.5% paralleliz-able code	199.0	0.4	19.9	0.44
	Total	200.0	5.4	20.9	1.44
Mostly sequential program	75% serial code	150.0	750.0	150.0	150.0
	25% paralleliz-able code	50.0	0.1	5.0	0.11
	Total	200.0	750.1	155.0	150.11

As we can see, like previously discussed, the coprocessing has the advantage but it needs to be said that serial performance is very important even in parallel code. We can also conclude that GPUs are effectively best optimized for parallel code and CPUs for serial intensive programs. The best performance was obtained because each core was running the portion for which it was developed for and this is the most important thing to take from the table.

To give us another idea of the potential speed up that Coprocessing can have we have another table that shows us exactly how faster the program could be. Please take in consideration this data is old, but it still makes truth the idea of the crucial and vital part that Coprocessing has in daily tasks such as converting a video. These daily tasks are from a personal perspective but also from the industrial perspective, which must be said that it is where it has the most impact.

Application	Field	Speedup
Two-electron repulsion integral[12]	Quantum chemistry	130×
Gromacs[13]	Molecular dynamics	137×
Lattice Boltzmann[14]	3D computational fluid dynamics (CFD)	100×
Euler solver[15]	3D CFD	16×
Lattice quantum chromodynamics[16]	Quantum physics	10×
Multigrid finite element method and partial differential equation solver[17]	Finite element analysis	27×
N-body physics[18]	Astrophysics	100×
Protein multiple sequence alignment[19]	Bioinformatics	36×
Image contour detection[20]	Computer vision	130×
Portable media converter*	Consumer video	20×
Large vocabulary speech recognition[21]	Human interaction	9×
Iterative image reconstruction[22]	Computed tomography	130×
Matlab accelerator**	Computational modeling	100×

* Elemental Technologies, Badaboom media converter, 2009; http://badaboomit.com.
** Accelereyes, Jacket GPU engine for Matlab, 2009; http://www.accelereyes.com.

As we can see there are many evidences of the big role that speeding up plays in certain tasks. Considering we are studying Bioinformatics I'd like to take a look to the "Protein multiple sequence alignment" that is represented in the table which had a 36 times speed even in 2009. If we consider that some tasks were faster 137 times the base time, I can't stress enough its importance. It also must be said that these numbers are affected by the portions of the code that are parallel and serial because CPU and GPU don't run at the same speed so the % of code present in each program is relevant to the impact of the potential speedup.

HPC in Cloud Evaluation

Cloud solutions allow users to access via Internet various types of resources such as existing applications in the Cloud, frameworks that can be used for development of custom-built applications, access to Virtual Machines for installing operating systems and also storage and sharing solutions.

The Cloud is now a significant choice for multiple types of users, common individuals, scientists or technical users so large datasets are generated, and must be processed. The scheduling algorithms used in Clouds can be improved to fit the new patterns of jobs and big data sets using hybrid approaches that will consider independent tasks, tasks with dependencies, asymptotic scale requests or smaller rates of arriving jobs

The recent Cloud computing paradigm was designed in order to provide end users and businesses with various advantages such as: self-service provisioning, broad network access, resource pooling, elasticity, measured service, pay per use. This approach is based on utility computing where we have infinite resources (as much as you need) and a concrete billing model.

The main benefits of Cloud systems are represented by the possibility to use high scale/low-cost providers, by having any time/place access via a web browser, rapid scalability (incremental cost and load sharing), and a great focus on local IT systems.

We classify the main characteristics and issues about Cloud system considering nonfunctional aspects, economic models and technological features.

Nonfunctional	Elasticity (ex: Amazon EC2)
	Reliability (ex: Vmware ecosystem)
	Quality of Service (ex: Amazon S3)
	Agility and adaptability (ex: FlexNet)
	Availability (ex: MS Azure)
Economic	Cost reduction
	Pay per use
	Improved time to market
	Return of investment (ROI)
	Turning CAPEX (capital expenditure) into OPEX (operational expenditure)
	Going Green
Technological	Virtualization (ex: Virtual Box)
	Multi-tenancy (ex: MS SQL)
	Security, privacy and compliance
	Data Management (ex: WebSphere)
	APIs and / or Programming Enhancements (ex: Hadoop)
	Tools

The evaluation metrics are presented for all described features in the previous section.

Some of this metrics are:

– Availability metrics: “flexibility, accuracy, response time”;

– Reliability metrics: “service constancy, accuracy of service, fault tolerance, maturity, recoverability”;

– Efficiency metrics: “utilization of resource, ratio of waiting time, time behavior”;

– Reusability metrics: “readability, coverage of variability, publicity”;

– Interoperability metrics: “service Modularity, service interoperability, LISI (Level of Information System Interoperability)”;

– Adaptability metrics: “coverage of Variability, other performance metrics”;

– Usability metrics: “operability, attractiveness, learnability”;

– Modifiability metric: “MTTC (Mean Time To Change)”;

– Sustainability metrics: “DPPE (Data Centre Performance per Energy) parameter, PUE (Power Usage Efficiency)”;

– Scalability metric: “average of assigned resources among the requested resources”;

– Elasticity metrics: “boot time, suspend time, delete time, provision (or Deployment) time, total acquisition time”;

– Communication metrics: “packet loss frequency, connection error rate, transfer bit/Byte speed, transfer delay”;

The new trends in modeling and simulation of Cloud Systems require performance evaluation metrics with a high level of accuracy.

This are all the metrics that the organizations must do in order to have a cloud support for their process. They need to evaluate if it fits well and using this metrics will help them to ensure that it’s a good idea to move to cloud or not.

Trends and Methods for parallel computing in HPC

Parallel Computing is one of the techniques that does computing processing by utilizing several independent computers simultaneously.

Parallel computers can be grouped according to the level at which hardware supports parallelism.

The development of information technology increasingly rapidly, it comes with the need of managing the data and organize it, in order to don't lose any information. Data is increasing more and more in the organizations, and that requires a computing device with high performance, able to process all the data and extract the information needed.

The process of doing parallel computing consists of using several computers simultaneously and using techniques that will increase performance.

That requires a parallel machine infrastructure, with all the computers connected to one network, here we are going to focus on what trends of parallelism affect the performance of computation nowadays.

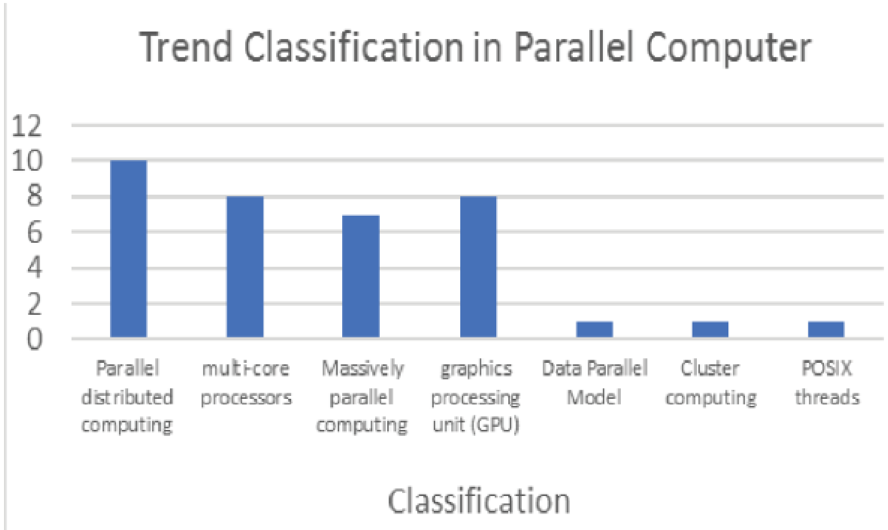
The most used methods are:

-<u>Parallel Distributed computing</u>: Form of calculation that solves problem by using multiple computer resources that are present in the number of nodes connected to each other.

-<u>Multi-core Processor</u>: This method combines two or more CPU's in the same circuit, and even each CPU can be a multicore processor.

-<u>Massively Parallel Computing</u>: A collaboration of the same program by two or more processors, and each processor handles various program threads, and what differs from Multi-core processing is the fact that each CPU has his own operating system memory

-<u>Graph Processing Unit</u>: GPU is a single chip processor with integrated transform, lighting, triangle setup/clipping, and rendering engine. GPU's can perform working process of at least 10 million polygons per second. It's more used on graphics processing and rendering based needs.



Research in this field, is developing from time to time. Here we could see a classification from the different trends. Each company and organization must evaluate and make research on what fits better for them, in order to increase performance and efficiency in the processes that they need to do.

Conclusion

High-Performance Computing is present in a lot of fields and devices. It can be found in medicine, our investments, our cellphones, in the movies we watch, etc. All sectors, in industry and the academic and scientific community, rely on HPC.

As we saw there are affordable options for high performance computing although its disadvantages. Even with its disadvantages it can be a good option for some users. We can know understand the difference between supercomputers and cluster computers. As we saw previously, nowadays, there is no clear distinction since modern datacenters can also have high performance interconnects.

Furthermore, when it comes to CUDA and Coprocessing it was a good experience to see how effective it can be both daily as well as on an industrial level even in our own study field, bioinformatics. We also want to emphasize the fact that it is really enriching to learn how these platform work and were it all started because it gives us the possibility how far the subject has come

In terms of HPC on cloud is a very recent trend and it’s evolving on a fast pace; companies are investing in software servers instead of more hardware disks.

Finally, in parallel computing, we could obtain the better classified trend, which is the distributed computing, being the most used among organizations and still has a lot of potential to be evolved into a much better performance system.

As final thoughts, we just want to leave a comment saying the realization of this research work was very challenging but turns outs to be super interesting. No regrets on the time we spend on the development of this paper and we hope this knowledge we acquired, can be used in a near future.

Acknowledgment

This paper was developed by a group of three elements, Bernardo Augusto, Daniel Marçal, and Marcelo Pereira and with the supervision of the professors, Norberto Albino and Laercio Júnior and carried out at Polytechnic Institute of Setubal, Portugal.

References

Agrawal, S. D. (2017). A Case for PARAM Shavak: Ready-to-Use and Affordable Supercomputing Solution.

Alfredo Buttari, J. D. (2008, July). *Limitations of the PlayStation 3 for High*, p. 19.

Czarnul, P. (2018, February 28). *Parallel Programming for Modern High Performance Computing Systems.* CRC Press.

Daniel Rosquete, G. L. (2009, August). *HPC for Playstation 3*, p. 8.

Eric Fernando, D. F. (2018). Classification and Advantages Parallel Computing in Process Computation: A Systematic Literature Review.

Heller, M. (2018, August 30). *What is CUDA? Parallel programming for GPUs*. Retrieved from InfoWorld: https://www.infoworld.com/article/3299703/what-is-cuda-parallel-programming-for-gpus.html

Jakub Kurzak, A. B. (2008, May/June). The PlayStation 3 for HighPerformance Scientific Computing. p. 4.

Kirt Lillywhitea, D.-J. L. (2009, August). *Lessons Learned in Developing a Low-cost High Performance Medical*, p. 7.