The Case for a Cost-Effective General-Purpose Computer Cluster for Small Colleges

https://doi.org/10.3991/ijet.v16i04.18837

Stefano Colafranceschi (✉)
Eastern Mennonite University, Harrisonburg (VA), USA
stefano.colafranceschi@emu.edu

Emanuele de Biase
Cornell University, Ithaca, NY

Abstract—The computational capabilities of commercial CPUs and GPUs reached a plateau but software applications are usually memory-intensive tasks that commonly need/utilize most recent hardware developments. Computer clusters are an expensive solution, although reliable and versatile, with a limited market share for small colleges.

Small schools would typically rely on cloud-based systems because they are more affordable (less expensive), manageable (no need to worry about the maintenance), and easier to implement (the burden is shifted to the datacenter). Here we provide arguments in favor of an on-campus hardware solution, which, while providing benefits for students, does not present the financial burden associated with larger and more powerful computer clusters. We think that instructors of engineering/computer science faculties might find this a viable and workable solution to improve the computing environment of their schools without incurring the high cost of a ready-made solution.

At the basis of this proposal is the acquisition of inexpensive refurbished hardware and of a type1 VMware hypervisor with free licensing, as well as the development of a custom-made web platform to control the deployed hypervisors. VMware is a global leader in cloud infrastructure and software-based solutions. In particular, the adoption of a customized "Elastic Sky X integrated" as hypervisor together with Virtual Operating Systems installed in the very same datastore, would constitute an interesting and working proof-of-concept achieving a computer cluster at a fraction of the market cost.

Keywords—Paper publishing, HPC, computer cluster, virtualization, small-size college, student engagement

1 Introduction

1.1 Background

Out of 4,298 US higher education institutions, 20% or 800 are small colleges. The focus of this manuscript is these colleges' computational infrastructure. Many of the

students who wish to pursue a successful career in the high-tech domain choose faculties such as Computer Science and Computer Engineering. The issue small colleges encounter is the lack of sufficient computational resources needed to provide the students with the preparation and training appropriate for their future profession. Often a small college is too small to afford and justify industrial high-performance clusters (HPCs) because of the initial investment cost, its long-term maintenance, and the required expertise to utilize the system [1]. Because of their high cost, the use of clusters in academia is limited to higher education entities of a size that can justify such costs. In most cases small colleges rent cloud servers, which are ready-made and have time and bandwidth limitations; as this paper documents, this cloud-based solution is far from optimal from the students' perspective.

1.2 Paper organization

The paper describes the case of an efficient implementation of a general-purpose computer cluster for small colleges.

Section 2 constitutes the core of the work, as we discuss and clarify the problem definition, we educate the reader about our motivations (pedagogical, inspirational, and economical). We also discuss technical issues such as control, security, and scalability as key aspects of a vision/winning strategy. In Section 2 we provide detailed solutions that answer the problem definition and help the reader understand what and how to do. We also present a compiled descriptive list so that the understanding is clear and straightforward.

In Section 3 we present a real case from our teaching experience. We reflect on the motivations that pushed us into these directions and describe the steps taken to implement the above-mentioned solution details (Section2) into the discussed case.

In Section 4 we provide a discussion of the project's business benefits, together with dedicated sub-sections that handle several business and resource aspects, such as storage, web-services, cloud-local repositories, and Integrated Graphical Environment for software production.

Section 5 summarizes the manuscript and Section 6 constitutes the conclusions underlining the impact of the developed and implemented technology in the field.

2 Problem Definition

In this document, which uses a student's perspective as its point of view, we identify the issues and the circumstances of small colleges struggling to offer HPC experience to students. We intend to offer a viable solution that allows small colleges to avoid costs connected with rent cloud servers or the acquisition of expensive and underutilized HPC systems. Our aim is to establish a paradigm and a methodology to offer an inexpensive HPC tailored to the students' needs to support their learning and their acquisition of technical preparation [13].

2.1 Overarching architecture

When addressing the implementation of computer clusters for small colleges, two main different paths are possible: on-campus [14] vs. cloud-based [2] (rented). These two high level competing technologies [16] offer advantages and disadvantages depending on who is the final user. In general, the centralized services of an institution can be conveniently handled through cloud-based solutions [3]. However, if we look at the problem definition from a student's perspective, we should value the learning experience rather than convenience. The rationale for an on-campus cluster has three most important and pedagogical reasons described in the following sections 2.1, 2.2, 2.3.

2.2 Inspirational motivation

It is a widespread misconception that the cloud-based solutions be the solution to all problems [15], yet from the students' perspective, the complexity of such an online-system is overwhelming. At the same time, the complexity is so invisible, that an undergraduate student might end up oversimplifying the situation. We believe that it is inspirational to provide students not just with an engineering black or grey box, but rather a practical tool for them to use. This approach instills curiosity and helps critical thinking. A better understanding of the hardware setup results in the acquisition of more basic knowledge, which is the baseline for getting new and breakthrough ideas. It is worth noting that even at a small hardware scale, software implementations, scripts customization, and general troubleshooting are still comparable to larger and professional systems.

2.3 Hand-on approach

The danger of teaching Computer Science and Engineering courses, without hardware access, is to educate professional coders who lack a comprehensive vision of the technology [4]. Every piece of technology is an integrated object in which the hardware, the firmware, and the software work together, albeit at different levels. Pure software without any knowledge of firmware and software is very abstract and usually leads to poor resource utilization or extremely inefficient applications. The ability and the success of a developer do not just depend on their ability to deliver a good technical solution, but to make sure that the algorithm is correctly implemented and runs with benchmarked and effective performance [5]. In this framework, we believe that having a system on campus makes it possible to provide a number of very helpful hands-on activities. In short, an on-campus system requires:

- Understanding and monitoring network activities (access control, load balancing, availability and quality of the service)
- Control of the involved hardware (CPUs, disks, network cards)
- Decision-making policies (software packages upgrade, hardware improvements)
- Weekly troubleshooting (software crashes, incidents, general problems)

These required activities constitute in themselves a deep learning experience, which can be bypassed through the adoption of a cloud-based system [6].

2.4 Control and scalability arguments

Ready-made solutions widely available in cloud-based systems are protected/controlled by the vendor. These solutions are also very scalable but the control and the scalability help the IT manager to deliver and get things done rather than providing insights on what control and scalability means. The case for cloud-system is based on the goal of providing an efficient solution in terms of IT/helpdesk time/financial management. An in-house solution requires effort and work to achieve complete control and the required level of scalability to ensure future operability. We believe that giving students complete control of the system is an invaluable professional experience [7]. Students, as they get ready to become professionals, should be prepared to face questions and address problems using a rationale. By adopting a solution where control is required, students learn this important skillset.

Scalability in cloud-system is often coupled with the ability to increase the performance/specification of the system at runtime, which is an invaluable feature for a company because of the difficulty to predict network traffic and application success. Changing the performance of the system at runtime allows us to make a more efficient use of resources. For a higher education institution, we believe scalability should mean a slightly different thing. It's unlikely that a campus has applications/services that would dramatically change performance/networking requirements, but the purpose and the configuration of the hardware might likely be changed/adjusted to the circumstances [8]. An easy example is to address capstone projects that might require very different configurations, software, middle-layers; the computational and network bandwidths do not change every semester or every year. Clearly, thinking about scalability in these terms, the in-house solution meets the ability to reconfigure the system as needed.

3 Solution Details

In this section, we describe the proposed solution in greater detail with the help of a case study to support our argument. The core idea is to prepare a cluster based on a type1 hypervisor and several Virtual Machines (VMs) that run on top of the hypervisor. After the proper configuration of one node of the cluster, a bit-by-bit carbon copy of the master node creates an image file that can be deployed into all the nodes. The proposed implementation (Fig.1) relies on a licensing-free VMware software-based ESXi [9] that provides the required type1 hypervisor at no cost and with an easy HTML5 GUI configuration page.

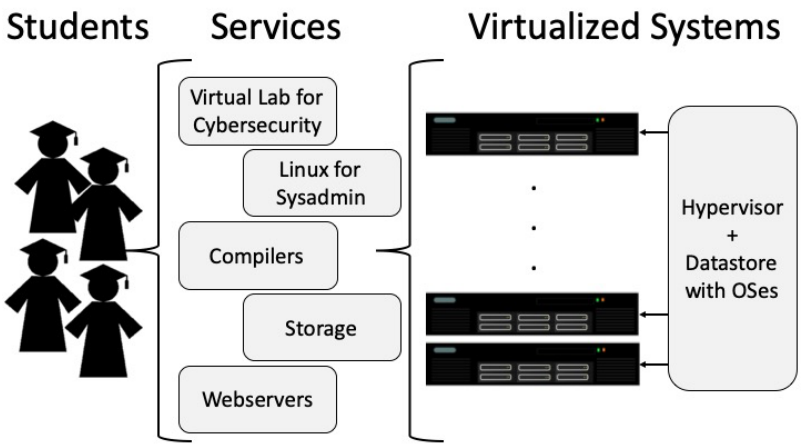

Fig. 1. Layout of the system, services, and users.

3.1 Motivation

We here describe a sample configuration of a small-scale system with 20 nodes, established for these classes: System Administration, Linux 101, Operating Systems, Networking, WebApps, Intro to CyberSecurity, Programming (C++, Java, Python). This work stems from the need of providing students a dedicated server, with hardware and software managed on campus. In our use-case we recycled old machines (DELL Optiplex 390) refurbished with additional primary memory (12GB per computer).

3.2 Implementation

The VMware ESXi is an enterprise-class, type-1 hypervisor that is installed on every node of the cluster. ESXi features a bootloader and minimal kernel that runs all the other OSs, by abstracting all computer resources. Once properly configured, the ESXi installer runs smoothly and can be efficiently managed via the web.

Since ESXi is the primary component in the VMware Infrastructure software suite, hardware limitations are present for general-purpose hardware (mostly network cards). Running your own version of ESXi on your available hardware can be troublesome if specific drivers are not supported by VMware. However, it is possible to add drivers into the ESXi installation ISO image and, even though the original ISO is certified and supported for most popular server models, it doesn't contain hardware drivers for all of them.

A third-party tool ESXi-Customizer is a graphical tool with a simple interface that makes possible the realization of the integration of driver files into ESXi ISO automatic. This tool works by including into the VMware standard ESXi any custom and/or available driver of hardware boards not supported by the hypervisor. ESXi can be licensed at no cost with the following limitations:

- No support
- Impossibility to add the nodes to a vCenter Server
- 2 physical CPUs
- Max. 8 vCPU per VM

However, the free license does support unlimited cores per CPU and unlimited physical memory. In our implementation, we adopted a flexible and versatile strategy to offer students a wide and diversified software/OS platforms. We implemented into VMFS these VMs:

- 2 Juno (BSD-like OS that powers Juniper router and switches for the CyberSecurity courses
- 1 CentOS (version 8)
- 1 Ubuntu (version 20)

ESXi runs on top of the previous list and its requirement for the primary memory is 4GB, so that the hypervisor itself can manage and orchestrate the other VMs. As concerns the secondary memory, since every computer node has only one physical disk drive, ESXi is installed in the first 1GB of the disk. The standard VMware practice is to install one VM per physical disk (datastore), and ESXi complies with this practice. However, by logging via root account in the hypervisor, one can manually override that limitation and use "vmkfstools" to create one or more VMFS partitions on any disk drive (also SSD). A 512GB disk (HDD or SSD) allows the creation of several 80GB VMs, also a dedicated 100GB VMFS could contain a datastore with the ISO installation images of most common OSes to facilitate any OS reinstallation.

Finally, we describe the operating environment to deploy the ESXi into several nodes. VSphere is a VMware solution designed to accelerate the digital transformation for evolving IT environments, given the non-availability of that application (due to expensive licensing) an alternative solution is described as follows. Once one node is totally installed with ESXi and configured with the VMs in their VMFS, the entire disk drive can be carbon copied bit by bit very easily with any low-level tool such as "dd". The result of this process is a prototype image file that can be deployed into all the other nodes. At boot time DHCP will make sure that every node will get a different IP so that every node is accessible via network.

Instead of using VSphere to control all the nodes of the cluster, we developed a Pythonic web-application that encapsulates all the HTML5 GUIs of all the nodes. In this way we achieve a close monitoring and control of every machine of the cluster.

4 Business Benefits

We believe that the proposed solution is the best compromise between industrial-grade professional HPC and consumer-level computing. It's a win-win strategy because we intend to use a consumer level hardware component with an advanced type1 hypervisor as a software layer between the hardware and the VMs [10]. After establishing the configuration of one node, the entire cluster can be deployed and re-

deployed in case of issues or at the end of each semester after student utilization. A few more beneficial aspects, add-on services, and utilities, are listed in the following subsections.

4.1 Storage

Commonly small- and mid-size schools rely on Google services for emails, calendars, and storage. Let's discuss the storage situation to understand the implication of cloud services like Google File Stream vs. on-campus *NIX solutions [11]. The first aspect of the Google cloud system is the *NIX interoperability and impossibility to directly mount the storage into a mount-point. The lack of a native *NIX client makes the cloud space useful for administrative work but not so appealing for a student learning/using a *NIX based system. Clearly, the on-campus storage option will have its limitations such as available space, however, we recognize that the price per GB is indeed very low and most of the students will deal with ASCII project files. It's also interesting to note that the File Stream (and similar) apps rely on a UI designed to meet the common user, not the student. A computer cluster that features on-campus storage allows establishing a shared file-system that can be directly mounted by students with the right privileges to do so [12].

In our implementation with only 20 nodes, all the VMFS disks that run CentOS8 are merged into a disk pool using GlusterFS and ZFS. This approach is convenient because it ensures disk over the network aggregation and data preservation/consistency in case of hardware failure of one or more disks (according to how the ZFS is configured).

4.2 Web-services

The WYSIWYG paradigm is established and many platforms/frameworks feature quick and easy utilization/deployment of web applications. The benefit of the in-house computer cluster, as concerns web services, is the ability to run any web related technology, not just the WYSIWYG universe. It's so common for a student to experiment with different programming languages, different encryption systems, different technologies to interchange data, that having a custom solution on campus is the natural way to experiment with this freedom of choice.

Another case is being able to use the on-campus cluster computers for the development of an API framework. As already mentioned, the computer nodes are easily initialized with a carbon copy from a master node, so we envision giving students a dedicated machine (for instance in a WebApp semester course) that can be used in several ways and finally scratched at the end of the semester to be used by someone else.

4.3 Local repositories

Several (free) cloud-based solutions offer limited space repositories (like BitBucket, GitHub), something that is convenient for large/professional and open-source

software/applications. In our view, local repositories (git or svn based) can play the very same role with the additional feature of ensuring privacy and copyright protection. It is not uncommon that a capstone project culminates into a provisional patent pending manuscript and it is the instructor's role to instill and stimulate copyright protection.

4.4 Scripting

For a sysadmin, a *NIX system is powerful thanks to several scripting languages that interface almost directly with the OS. Scripting often involves some other device on campus to be read or to be interfaced with (logging people, detecting objects, measuring a stream of data). It is convenient to have an on-campus solution within the same network to quickly prototype apps and ideas with student scripts. Since the storage is also *NIX mountable directly on campus, script results can be conveniently saved into dedicated student folders.

4.5 IDEs

While teaching programming languages, it is very important that students understand the lifecycle of a piece of code [17]. Connecting students to their on-campus server, equipped with compilers, facilitates this learning process. Modern laptops are no doubt equipped with the latest technologies and run heavy IDEs with advanced features but most of these features simply overwhelm students and hide the real job of the linker/compiler. Students must use a simple/light IDE and a command-line base *NIX compiler to understand and to practice with the process behind the generation of a binary executable.

5 Conclusion

This manuscript aims to help to understand the importance of a small-scale cluster-based setup for small colleges. We claim that, even for non-research schools, where the financial burden of a high-performance cluster is prohibitive, it is beneficial for students to have such technology for learning purposes, student project administration/implementation and to better prepare students for their future professional life. If you are a CS Instructor of a teaching college consider turning inexpensive and perhaps outdated hardware into a cluster solution powered by a Hypervisor to offer students handy virtual labs.

6 Acknowledgement

This work was supported by the Thomas F. and Kate Miller Jeffress Memorial Trust.

7 References

[1] Miłosz Ciznicki et at. Benchmarking data and compute intensive applications on modern CPU and GPU architectures. Procedia Computer Science 9 (2012) 1900 – 1909. https://doi.org/10.1016/j.procs.2012.04.208

[2] Atul B Naik et al. Applicability of Cloud Computing in Academia, Indian Journal of Computer Science and Engineering (IJCSE)

[3] Nasim Abdulwahab Matar, Tirad AlMalahmeh, Mohammad Al-Adaileh, Saheer Al Jaghoub Factors Affecting Behavioral Intentions towards Cloud Computing in the Workplace: A Case Analysis for Jordanian Universities https://doi.org/10.3991/ijet.v15i16.14811

[4] Violeta Holmes and Ibad Kureshi Developing High Performance Computing Resources for Teaching Cluster and Grid Computing courses https://doi.org/10.1016/j.procs.2015.05.310

[5] Fred L. Kitchens, Sushil K. Sharma, and Thomas Harris Integrating IS Curriculum Knowledge through a Cluster-Computing Project, A Successful Experiment Journal of Information Technology Education, v3 p263-278 2004. https://doi.org/10.28945/301

[6] Preeti Jaiswal, Abdulghani Ali Al-Ha Enhancing Learners Academic Performances Using Student Centered Approaches, https://doi.org/10.3991/ijet.v15i16.14875

[7] Meruert Serk, Nursaule Karelkhan Setting Up and Implementation of the Parallel Computing Cluster in Higher Education https://doi.org/10.3991/ijet.v14i06.9736

[8] G. Chiola, "Some research projects on clusters of personal computers," Proceedings. 24th EUROMICRO Conference (Cat. No.98EX204), Vasteras, Sweden, 1998, pp. XLVII-XLLIV vol.2, https://doi.org/10.1109/eurmic.1998.708060.

[9] https://www.vmware.com/pdf/hypervisor_performance.pdf

[10] T.N. Sugumar and Dr. N. Rajam Ramasamy Literature Survey and Review: Virtualization Technologies International Journal of Printing, Packaging & Allied Sciences, Vol. 4, No. 2, December 2016

[11] S. Rajkumar Patil, V. R. Kulkarni, R. M. Jogdand and S. Batlad, "A comparative review on Ceph and Swift open-source cloud storage platform," 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, 2016, pp. 213-218, https://doi.org/10.1109/icgtspicc.2016.7955300.

[12] Imran Sarwar Bajwa, A. Chaudhri, M. Naeem, Processing Large Data Sets using a Cluster Computing Framework

[13] Becker, H. "Pedagogical Motivations for Student Computer Use That Lead to Student Engagement." Educational Technology archive 40 (2000): 5-17.

[14] A. A. Frohlich and W. Schroder-Preikschat, "SMP PCs: a case study on cluster computing," Proceedings. 24th Euromicro Conference (Cat. No.98EX204), Vasteras, Sweden, 1998, pp. 953-960 vol.2, https://doi.org/10.1109/eurmic.1998.708127.

[15] Douglas K. Barry, David Dick, Cloud Computing in Web Services, Service-Oriented Architectures, and Cloud Computing (Second Edition), 2013. https://doi.org/10.1016/b978-0-12-398357-2.00024-5

[16] Shiv Shankar, Ashish Kumar Sharma, 2017, A Comparative Performance Analysis of Cloud, Cluster and Grid Computing over Network, International Journal of Engineering Research & Technology (IJERT) ICADEMS – 2017 (Volume 5 – Issue 03),

[17] Schindler, L.A., Burkholder, G.J., Morad, O.A. et al. Computer-based technology and student engagement: a critical review of the literature. Int J Educ Technol High Educ 14, 25 (2017). https://doi.org/10.1186/s41239-017-0063-0

8 Authors

Stefano Colafranceschi is faculty at the Eastern Mennonite University (EMU), 1200 Park Road, Harrisonburg (VA). He is an active IEEE member and author/reviewer of several articles and manuscripts

Emanuele de Biase is a New York-based consultant, with experience in the Education and Entertainment industries. He holds degrees from the University of Oxford, CUNY and the University of Roma Tre, where he was awarded his PhD. He is currently an MBA candidate at Cornell's SC Johnson Graduate School of Business.

Article submitted 2020-09-24. Resubmitted 2020-10-26. Final acceptance 2020-10-27. Final version published as submitted by the authors.