Implementing micro High Performance Computing (µHPC) artifact: Affordable HPC Facilities for Academia

Nkundwe Moses Mwasaga
School of Computing
University of Eastern Finland
Joensuu, Finland
nkundwe.mwasaga@dit.ac.tz

Mike Joy
Computer Science
Warwick University
Coventry, UK
m.s.joy@warwick.ac.uk

Abstract— **This Research to Practice Full Paper presents the experience of implementing micro HPC (µHPC) clusters in the education context. In developing countries, access to and experience of High Performance Computing (HPC) facilities (which are important for the economic development of those countries) has presented challenges to academic institutions due to prohibitive costs of acquisition and maintenance of HPC systems. These challenges affect the delivery of courses, which include HPC and parallel computation in their syllabi. However, due to advancements in integrated circuit technologies, credit-card-sized personal computers (PCs) have emerged with the capabilities of a fully-fledged PC. These credit-card-sized PCs offer an opportunity for system integrators to build micro HPC (µHPC) clusters which, apart from their efficiency, scalability and availability, have a number of attributes which are not shared with conventional data-center based HPC clusters These include their low power consumption, portability, diskless nodes, cost-effectiveness and excellent price to performance ratios. µHPC clusters are, therefore, ideally suited to support curricula in all educational disciplines that demand parallel computation. The purpose of this study is to investigate the implementation of µHPC to find out from students how ease of use of a µHPC system is to acquire skills and knowledge of HPC. We used focus groups and surveys to gather data in the framework of the design science research paradigm. The implementation of µHPC clusters involved a demonstration of the system in the problem context that provided valuable lessons. The results indicate that µHPC is the ease of use HPC artifact for learning about management, integration, deployment, installation, downloading, and running of parallel programs.**

Keywords— ***micro HPC, Beowulf cluster, constructivism, parallel computation, High Performance Computation, scalability, credit card-sized PC.***

I. INTRODUCTION

Knowledge and skills of High Performance Computing (HPC) systems are vital competencies required for advanced computing students in the 21st century. HPC skills and knowledge can be easily grasped when an HPC artifact is implemented or demonstrated to increase the usage to target users. The implementation of an HPC system as an educational activity involves the demonstration of HPC artifacts in the original problem context. The implementation of the designed artifact is a vital source of information about the usability of the artifact and the properties of the artifact. The implementation is about studying the interaction between the artifact, the participants, and the educational setting for the intervention in order to achieve an intended goal. The ease of use of HPC systems enables the acquisition of knowledge and skills of HPC that are required for students to access improved career opportunities in this big data era. Many types of HPC systems have been deployed and implemented in academic institutions [38], including multiprocessor simulators or emulators; supercomputers located at supercomputer centers, and accessed remotely; high performance multiprocessor systems purchased and maintained by the institution; and workstation clusters [16].

Many papers have been written about the implementation of HPC artifacts in different contexts, such as to support research and educational activities [8][12][18][19][26][28][31]. Others have articulated the desire to implement HPC systems as pedagogical tools that enable lecturers to reproduce student experiments to enhance trust in students' assignments [31]. Also, they have reported the experience of the establishment of an HPC cluster lab for a small institution that supports the curriculum [24], research activities, and student learning [36]. Moreover, the performance of programs in a Beowulf cluster for supporting rapid prototyping, teaching, and learning of distributed and parallel computing has been explored [19][26]. The Beowulf cluster architecture enables computing devices to be connected to form a single micro HPC (µHPC) system capable of achieving HPC performance [14][15]. The µHPC possesses highly parallel processing power and distributed storage capable of handling data and provide computing to tackle complex scientific and engineering challenges [4][22]. The µHPC systems support Linux open-source software [6], which includes features that are needed in parallel and distributed computing, such as full-featured multiuser and multitasking. However, the technical capability and configuration of clusters and virtual clusters as educational tools are not always presented with an explanation of their implementation [19][35].

The purpose of this study was to understand how easy to use a designed µHPC artifact as an education tool. Ease of use can be measured by implementing a µHPC system as a potential solution to the original problem [20]. This study focuses on the implementation of such a µHPC artifact in an educational context. The approach used is informed with constructivism and constructionism learning theories [1] in which individuals construct cognitions as the function of their experience in the learning context [43]. In constructivism learning theory, the interaction of situations and persons in refinement and acquisition of knowledge and skills is highlighted [44]. In this study, the learning environment was structured such that students can construct new skills and knowledge effectively [42].

In line with that, we aimed to investigate the lessons learned from Tanzanian students who participated in an HPC course using affordable µHPC clusters and to elucidate how easy to use the µHPC system was in acquiring HPC knowledge and skills. The investigation involved the evaluation of the ease of use of the µHPC artifact [21]. The research problem of this study addresses whether the implementation of a µHPC system satisfied ease of use as a usability requirement using Design Science Research (DSR). The research question of this study is "How easy to use is the implemented µHPC system for students to acquire HPC knowledge and skills?" The delivery of an HPC course and recruitment of participants for this study was from an academic institution that offers HPC modules, computer science, or computer engineering courses.

The remainder of this paper is organized as follows. In the next section, we consider the µHPC artifact and the analysis framework, followed by the research methodology, and then the results. Finally, we present discussions and conclusions.

II. ARTIFACT AND FRAMEWORK

This section is organized in two sub-sections that detail the implemented µHPC artifact and the framework used to guide the analysis of data gathered from the implementation of the artifact.

A. *Outline of the Artifact: µHPC Cluster System with 5 Nodes*

The low cost and small physical footprint of credit-card sized PCs, which can be connected together to achieve HPC performance have democratized HPC clusters[6][26][35]. The HPC clusters can be introduced and constructed as a pedagogical tool in resource-constrained areas [36][18]. Such µHPC systems form a generation of HPC systems that can be used for demonstration in terms of performance and power consumption and are small in size with a low design cost [4][6]. µHPC clusters are categorized as "micro" in terms of their size, but they can still provide high speed and density, and the attributes of µHPC clusters such as scalability, usability, availability, reliability, and affordability of commodity off the shelf (COTS) products [23].

TABLE 1. DECISION MATRIX FOR CREDIT-CARD SIZED PCS

Attributes	**BeagleBone Black[45]**	**Jetson TK1[47]**	**ODROID-C1[46]**	**Raspberry Pi 2 Model B[6]**
CPU	1GHz Sitara AM335x Cortex A8 ARM processor	Tegra K1 SOC 2.3 GHz 4-Plus-1 quad-core ARM Cortex A15 CPU	1.5GHz quad-core ARM processor	Broadcom BCM2836 900MHz quad-core processor ARM Cortex-A7 System on a Chip (SoC)
GPU	PowerVR SGX530 GPU	Kepler GPU with 192 CUDA cores	2 x ARM® Mali™-450MP 600MHz	1 x VideoCore IV 250Mhz® Multimedia
RAM	512MB DDR3 800 MHz	2 GB x16 memory with 64 bit width and 16 GB eMMC	1GB 32bit DDR3 792MHz	1GB LPDDR2 SDRAM (2x memory)
Cores	1	192 CUDA cores and four-core ARM CPU	4	4
Ethernet, with RJ45 Jack	1x 10/100Mbit/s	1x 10/100/1000Mbit/s	1x 10/100/1000Mbit/s	1x 10/100 Mbit/s
Operating Systems	Debian, Android, Ubuntu and Cloud9 IDE	Gentoo and Ubuntu Linux	Ubuntu Linux 14.04 or Android	Fedora, Ubuntu, and Debian, or FreeBSD
TDP/Power Draw / voltage	210-460 mA@5V	Up to 5A @ 12 V	2A @ 5V	800mA up to 1.2A @ 5V
Unit Price	45 USD	192 USD	35 USD	35 USD

The choice of credit card-sized PCs in the design of the µHPC system follows the modularity design trend because, as the number of credit card-sized PCs and their individual processing power increases, their size, weight, and power consumption continue to decrease. We evaluated the credit-card-sized PCs, as indicated in Table 1. The Odroid had better attributes than Raspberry Pi; however, we opted for Raspberry Pi 2 Model B due to availability and user support of the Raspberry Pi community. The Raspberry Pi 2 Model B offers computing performance as the basic node in designing and implementing the µHPC artifact[2]. The single-node consumes 3.5W of power [6]. The node supports Linux distributions as well as other operating systems, such as FreeBSD and also Microsoft Windows 10 [17], thus demonstrating the greater acceptance and support of this model in the ICT industry.

The system has a master node, a Raspberry Pi 2 Model B system, with the pre-installed Linux operating system. Linux, which is installed on Raspberry Pi, is the most common operating system used on individual nodes of 5-node Beowulf-style parallel computer systems[18]. The diskless booting of the computing nodes was handled by the BOOTP protocol in the master node as a BOOTP server. The master node contained a video display, keyboard, and a 16 GB SD card working as a hard disk. There was no disk storage at the master and computing nodes as each of the five nodes has an SD card with a capacity of 16 GB working as a hard disk. The computing nodes did not directly access input-output devices or the host's file system. The four computing nodes and one master node were arranged using a 1000 megabit-per-second Ethernet switch with each computing node communicating with each other and also the master node.

The extra storage capacity can be accessed by the µHPC system that takes advantage of a USB HDD that is faster and more robust when it comes to accessing data repeatedly. The design with USB HDD can be useful when running large MapReduce applications with Hadoop [6]. The ARM processor has four instruction units, two of which are integer units, and two are floating-point units. The instruction units are pipelined and able to produce a result on each clock cycle if the pipelines are kept full. With the clock running at 900 MHz, the ARM peak performance is 2 x 900 x 10^6 = 1.8 x 10^9 integer instructions and 1.8 x 10^9 floating-point instructions per second (i.e., a gigaflop). The actual sustainable peak performance, achieved by running the LINPACK benchmark code, is usually slightly less than theoretical peak performance.

The Python, C/C++, FORTRAN, and Java programming languages are supported by the µHPC system. The basics of programming parallel computers can be taught using the C language and three different extensions to C, i.e., Message-Passing Interface (MPI), POSIX threads (pthreads), and OpenMP [10]. These libraries and languages have a long history with parallel computing. Hence they can be used for writing parallel applications using implementations of MPI, Hadoop, and OpenMP. Java can be used to write Hadoop-based MapReduce applications. The implemented system can be powered via alternative power sources that allow them to be deployed in rural areas, e.g., battery or normal electrical power. The µHPC system can be powered by using solar panels and batteries, Li-Ion batteries, and a standard micro USB power supply. However, whatever power source is chosen should output at least 5V.

B. *The Thematic Framework for Implementation of µHPC*

The thematic framework delineates the role of a µHPC cluster as an HPC facility that integrates Piaget's and Papert's [1]learning theories (constructivism and constructionism) in delivering HPC concepts. The theories focus on student-centered learning and advocate the role of knowledge construction when students learn. The µHPC system in this framework is a cognitive tool controlled by the learner to perform activities collaboratively to contextualize learning in a meaningful way through reflection on what has been learned through conversation with other learners [9][11]. As a cognitive tool, a µHPC system can enhance learners' cognitive powers during their thinking, problem-solving, and learning HPC concepts [9][13].

The framework comprises a dependent variable, independent variables, and a mediator variable. The mediator variable is access to an affordable µHPC system. The independent variables are HPC concepts, the HPC programming environment, prior knowledge, students' motivation [7], and learning activities. The dependent variable is the acquisition of hands-on competence in parallel and distributed computing. The framework guides the investigation of ease of use in analyzing research data in learning HPC concepts, HPC programming environments, usage of prior knowledge, students' motivation factors, and learning activities. These are mediated by access to affordable µHPC facilities that enable the students to construct hands-on competence in parallel and distributed computing.

1) HPC Concepts

We demonstrated the µHPC system by devoting more time to hands-on training. We expected the students to be able to demonstrate lightweight abilities in the concepts of networking, integrating, configuring, running parallel programs, and benchmarking the µHPC facility using a Beowulf architecture. Learners can generate HPC knowledge by using available methods of constructivism to teach and learn HPC concepts. Based on constructivism, concepts can be learned when definitions are contextualized in the original setting of the problem. Hence, the understanding of HPC concepts can be constructed through hands-on activities and inquiry.

2) HPC Programming Environment

Several programming languages can be installed in a µHPC environment. These can be used for writing parallel applications using implementations of MPI, Hadoop, and Hadoop-based MapReduce applications [6]. A µHPC system provides a programming environment by installing MPICH, which provides students the opportunity to use MPI. The µHPC environment has powerful tools that enable problem-solving by using parallel libraries in parallel programming languages. However, based on constructivism learning theory, knowledge to solve parallel problems is constructed not by mastering tools but by using them through activity and problem-solving.

3) Prior Knowledge

Prior knowledge enables learners to integrate and construct new concepts during active learning [38]. Students' prior knowledge and skills are needed when introduced to HPC and include C/C++ programming, communications, computer systems architecture, and operating systems [8]. The constructivism learning theory supports the observation that students form meaning, new concepts and produce knowledge based upon their prior experience. The methodology that uses constructivism in learning HPC should prioritize the identification of students' prior experiences and knowledge. The process of identification helps avoid the challenges of interference with the learner's prior experience that persists during learning of the whole computing curriculum.

4) Students' Motivation

In using the µHPC artifact, the focus during the process of learning should be on the real world and on intrinsic motivating problems [7]. According to Ryan and Deci [5]:

> Intrinsic motivation is defined as the doing of an activity for its inherent satisfactions rather than for some separable consequence. When intrinsically motivated, a person is moved to act for the fun or challenge entailed rather than because of external products, pressures, or rewards.

The goal of students' motivation in constructivism is to engage students effectively in using µHPC as a motivation factor to learn HPC concepts [33]. In constructivism, as facilitators, teachers should prioritize the essence of students' motivation in learning because the motivated student can dedicate more time to the µHPC system, which can further improve the learning process and increase the students' engagement with the HPC paradigm.

5) *Learning Activities*

In constructivism, the replacement of lectures by sets of problem-solving activities is designed to encourage learners to construct concepts and provide solutions for autonomous thinkers and learners [3]. In this approach, the approaches to problem-solving skills are constructivist and organized around a set of activities that the student is supposed to undertake. The undertaken activity should probe the learner's grasp of concepts and encourage learning where they construct HPC techniques and concepts. In performing these activities, to help with cognitive complexity and task management, the teacher should make sure that just enough help and guidance is provided, but not too much [11]. These activities should be organized to identify students' preconceptions. The learning activities should involve activities that can introduce HPC concepts, such as parallel system integration, HPC system administration, and parallel programming, to improve the learners' prior knowledge. The essence of this is to trigger students' curiosity to investigate and learn more about HPC concepts.

III. METHODOLOGY

The factors identified for the μHPC system to be used in this implementation are price, scalability, processor power, and ease of use [28]. In this study, we have modified the definition of perceived ease of use of the system [27][33], to mean the degree to which the user expects the target system to be free of effort to acquire the knowledge and skills required to use HPC systems. In this context, perceived ease of use refers to perceptions of the participants [37].

This study's research question was, "How easy to use the implemented μHPC system for students to acquire HPC knowledge and skills?" The answer to this question consisted of explanations of why the μHPC system is easy to use and descriptive knowledge that describes how the μHPC system is easy to use in an education setting. The framework used in this study is DSR, which is the research paradigm that advocates knowing through making artifacts [20][34].

The approach used in this study used a concurrent mixed-method design. A qualitative approach is appropriate because we intended to study the interpretation of initial experience and the meaning attributed to experiences of μHPC from the participants' perspectives. A quantitative approach was also chosen to confirm the findings from a wider group of participants. Hence the triangulation of quantitative data was employed to compare the findings with those from the qualitative study [39]. Exploratory focus groups and survey questionnaires were employed to evaluate the implementation of the μHPC system.

Two focus groups were conducted in an academic institution, each with five students who had used the μHPC system as the educational tool. First, two exploratory focus groups were conducted to receive feedback on the implementation to achieve incremental improvements in the μHPC artifact design in the education environment. The first group composition was four males and one female, and the second group was five males. A representative sample was not sought because HPC students could only be found in one academic institution, and mixing groups of participants with diverse knowledge concerning the research topic could bring out inadequate depth [32].

The first focus group took a single 29-minute session, and the second took a single 20-minute session. The transcript was verified against audio recording, and then all names were removed to ensure confidentiality.

Purposive sampling was used in the survey as the participants had the experience of using the μHPC artifact during its implementation. According to the literature on usability evaluation, 50% to 85% of the problems can be identified by only five participants [40]. Other literature [41] showed that 20 users found at least 95% of the usability problems. Then the post-survey questionnaire targeted and administered questionnaires to a sample of students (n=40) who participated in the HPC course where the μHPC artifact was implemented. The survey questionnaire received a 100% (n=40) response rate. The response rate was high because the survey was conducted face-to-face with participants during the implementation of the μHPC artifact. The 40 computer science participants were from diverse education levels, including degree and diploma. The ten target questions were derived from the analysis of issues from the focus groups. The ten target questions for this study measured students' perceptions of how easy it was to use the μHPC system during the HPC course and took the form of 5-point Likert scales(1 = strongly agree, 2 = agree, 3 = neutral, 4 = disagree, 5 = strongly disagree). The questions covered topics such as usage of a cluster-based HPC system, installation of any operating system and applications in the HPC system, configuration of the network for any cluster-based HPC system, integration of the cluster-based HPC system, management of the HPC system, installation of parallel computing programs in HPC system, deployment and configuration of the cluster-based HPC system, running parallel applications in the HPC system, and benchmarking the HPC system performance.

The dataset for this study consisted of transcribed focus group data and the survey data. The management and organization of data used word processing software for the focus groups and SPSS software for the survey data. The data analysis approach used for qualitative data from the focus groups is a combination of approaches [29][30] that employ framework analysis. Basic descriptive statistics were used to analyze the surveyed data.

IV. RESULTS

The main findings of this study address the question of ease of use of the μHPC system. The results of this study are divided into two sections. The first section presents the results of the focus groups and is followed by analyzing the questionnaires in subsections A and B, respectively.

A. *Results from Focus Groups*

Initially, the users received training on the μHPC system from the facilitators. Then they practiced with the μHPC system to accomplish some parallel tasks. The focus groups' responses were collected, grouped, and coded with the emerging points in the discussion. The responses were grouped in the following themes: learning activities, students' motivation, prior knowledge, HPC programming environment, and HPC concepts

These themes were used to capture participants' perceptions regarding the ease of use of the µHPC system for students to acquire HPC knowledge and skills during the implementation of the HPC artifact.

1) HPC Concepts

Participants in the study discussed what they liked about the implementation of the µHPC system and the experience of studying HPC concepts using the system. The participants mentioned portability, affordability, computation capability, support for the Python programming language, the system is easy to learn, and support for command prompts. These attributes of the µHPC system can enable participants to acquire knowledge and skills needed to work with any HPC system, as is suggested by one participant: "One of the greatest things is portability, the look of the device that we have used, they are not expensive which means learners can gather their strength and be able to own two or three of the components and we will keep on practicing it using our expenses as they are affordable." This was also attested by the other participant as, "But due to portability those components are easy to carry, you can move with them from one point to another comparing to when if you were using computers … easy accessibility of components, it leads us to the point whereby in the middle of the learning, I was very comfortable."

In response to the question "What is the general reflection on the training of µHPC?" the participants mentioned that during the course using µHPC, they learned parallel processing, parallel programming, computer networking, how to build an HPC cluster, how to use Linux commands, and parallel programming. One participant proposed more practicals in parallel programming during the implementation of other courses as "For me, it was a new experience. I have learned parallel processing. I have learned about HPC. What I can recommend is to have more programming practicals in the course." Another participant discussed the matter at length the ease of use of µHPC to acquire HPC knowledge or supercomputer knowledge during the training course. The participant went further to mention that it was the ease of use of learning a new programming language that wasn't covered in the curriculum: "the training course involved programming using a different programming language, Python, which is not taught in our curriculum." Participants mentioned the ease of use of the µHPC system as an educational platform to learn computer networking.

Regarding ease of use of µHPC to grasp the HPC concepts by students, one participant elaborated, "At first it was a little bit difficult but after performing the first time and second time adding nodes to a cluster of three or four or five was easy. What we did is to attach the node and install the image as the same as other nodes, and apart from that, configure the same configuration as the other nodes. So it was not so complex as in the first stage and second."

In regards to a limitation of the µHPC system in terms of a number of nodes, one participant explained that "For me, I can say it was easy, but there were some limitations because when we were adding numbers of processes per number of nodes the benchmarking was not responding well, so we needed to reduce the processes to match the number of nodes. So we need to be careful of the number of processes to much the number of nodes. So the knowledge of networking and Linux helped to do that."

In regards to the challenge of power supply to the master node and computing nodes, one participant said that "Of course when we were performing benchmarking, we realized that the nodes received power from USB hubs there were not capable of performing benchmarking when all the source of power is coming from the USB hub. So we found another adapter, which was a little bit more powerful to the power that USB hub. After that, we were capable of performing benchmarking."

Regarding the challenges of deployment of the µHPC system, one participant mentioned, "I can say it was hard to deploy µHPC when moving from multimedia lab to a class that doesn't have Internet, especially the difficulties in the installation of MPICH. It reached a time that the installation has to be postponed due to slow or lack of Internet. So it was a challenge."

2) HPC Programming Environment

When the question "What do you like most about µHPC?" was posed, participants explained that during the course, they liked the built-in support of parallel programming with Python and support for command prompts in the µHPC system. As suggested by one participant: "I like it because it is the new concept to me, and I learned new things like the Python language and the whole concept of supercomputing." The same is suggested by another participant: "For myself, I can say even the programming language, python, that we have used to write programs it is a little bit simple than other languages like C we have been using. The language is easy to learn." This was quickly followed by similar accounts when participants were asked to reflect on the course of µHPC, and another participant reinforced that "… I used to believe that a cluster has to be as big as a full-size cluster. So in the µHPC course, I have learned new things like a portable supercomputer, which is very small. It uses the Linux operating system and needs one to be familiar with Linux commands."

3) Prior Knowledge

The participants discussed new HPC topics and how they integrate with their prior knowledge. They variously mentioned prior knowledge to be either detrimental or useful in learning HPC.

One participant attested to how prior knowledge of Windows terminal commands helped him switch to the Linux command prompt used in the HPC environment. One participant suggested, "The difficulty was to cram other things like Linux commands, e.g., the 'sudo' command." The other participant explained: "I had no experience to that extent using the Linux operating system, but with the supercomputer, we found ourselves interacting with the Linux operating system. From there, using terminal commands of Linux is quite different from the command prompts of Windows. Some of the commands are quite different from Windows though some commands relate to other Windows commands. If you know the Windows operating system's terminal commands, it is easy to switch to those commands." Hence the µHPC artifact supports the Windows operating system [30].

This was followed by another participant who worked in a PC Linux environment, and stated that "The Linux processes in a single processor environment are the same as those in the HPC environment."

In regards to prior knowledge, one participant attested that "For my side primarily when we were starting, it was actually challenging. I recalled back the knowledge of networking about what I am supposed to do to solve that ... So it was easy to solve networking problems which were appearing now and then."

4) Students' Motivation

The participants' motivation was displayed in the form of self-confidence that enabled them to be actively involved in problem-solving using µHPC. In response to the question, "Do you think it would have been good to spend more time in installation, benchmarking, or Linux commands during the µHPC course?" The participants mentioned that more time is needed to exercise with the benchmarking application and the interpretation of benchmarking results. Also, participants mentioned that there is a need to continue studying how to run parallel applications. One participant mentioned that "It will be better to continue running parallel applications and LINPACK benchmarking tests. Since the course, we have learned to assemble, configure, install, write programs, run programs, and benchmarking". The participant said, "The point of addition is that some days it was hard to install the benchmarking software, the LINPACK benchmarking package. It was tough to install because we took about two or three days. We can continue by learning how to run different parallel applications." Also, participants have stated that the course should start with Linux commands and how to install and execute parallel programs before starting studying HPC.

5) Learning Activities

Responding to the question "What were the challenges of installation of dependencies in the µHPC system?", the participants mentioned that the challenge was to find the dependencies on the Internet and assigning the correct path as required with benchmark software. Also, they mentioned that there was the challenge of the slow Internet speed in downloading dependencies.

When asked, "What were other challenges you faced during the installation of the µHPC system?", participants mentioned the challenges of the number of µHPC systems that were available to practice on. As one participant commented, "For me, it was not that easy or not that hard because the time that we had the tools that we had was not enough to perform the practical, then we waited for an hour or days. So the interval between hands-on practicals sometimes makes us forget some concepts. So that is the challenge."

B. Results from the Questionnaire

The frequency of responses from participants with regards to the perceived ease of use of different constructs has been analyzed. We employed SPSS to analyze statistical data from the survey questionnaire. In check reliability, Cronbach's Alpha of 0.71 was achieved, which is good because it is more significant than 0.7. That assured us the consistency of further analysis of the data in the questionnaire. The study findings of the ease of use statements have been demonstrated in Table 2.

TABLE 2. STATEMENT, SCORE AND MEAN

No.	Statement	Score	Mean
1.	I find it easy to use a µHPC system to learn skills and knowledge on how to use a cluster-based HPC system.	33/40	1.79
2.	I find it not easy to use a µHPC system to install any operating system and application.	28/40	3.66
3.	I find it easy to use a µHPC system to learn skills and knowledge on how to configure the network for any cluster-based HPC system.	35/40	1.84
4.	I find it not easy to use a µHPC system to learn how to integrate the cluster-based HPC system.	26/40	3.66
5.	I find it easy to use a µHPC system to manage an HPC system.	31/40	1.79
6.	I find it easy to use a µHPC system to accomplish the parallel computing programs in an HPC system.	34/40	1.89
7.	I find it not easy to use a µHPC system to learn how to deploy a cluster-based HPC system.	24/40	3.37
8.	I find it easy to use a µHPC system to learn how to configure a cluster-based HPC system	30/40	2.03
9.	I find it easy to use a µHPC system to learn skills and knowledge on how to run parallel applications in an HPC system.	27/40	2.26
10.	I find it easy to use a µHPC system to learn skills and knowledge on how to benchmark the performance of HPC systems.	33/40	2.00

Table 2 summarizes the aspects of the quantitative research data for this research. It seems ease of use of the µHPC system was indeed to learn how to configure the network and to accomplish the parallel computing programs in any cluster-based HPC system. However, it seems the smallest majority of the participants disagreed the least about the µHPC system not being easy to use to learn how to deploy a cluster-based HPC system.

V. DISCUSSION

This research reinforces our understanding of the ease of use of the designed HPC artifact from participants who used the µHPC system as an educational tool during the HPC course. Hence we intended to dig deeper to study ease of use of the artifact and understand all the aspects that attribute from the feedback of the participants who experienced the µHPC system.

Overall, several of the findings from the focus groups were consistent with the findings of the survey. The findings suggest that the usage of affordable µHPC clusters has influenced students who participated in the implementation of HPC, and the majority have reported positive ease of use of the µHPC system to acquire HPC knowledge and skills during the course. Overall, participants have found the µHPC system easy to use to learn knowledge and skills to benchmark HPC systems' performance, accomplish parallel computing programs, configure the network for a cluster-based HPC system, and use any cluster-based HPC system. The highest score in terms agreement from the participants (35/40) was in the construct that "it is easy to use a µHPC system to learn skills and knowledge on how to configure the network for any cluster-based HPC system." The lowest score in terms of disagreement (24/40) was in the construct, "I find it not easy to use a µHPC system to learn how to deploy a cluster-based HPC system."

The majority agreement in the statements "easy to use a µHPC system to learn skills and knowledge on how to use cluster-based HPC system" and "easy to use a µHPC system to learn skills and knowledge on how to configure the network for any cluster-based HPC system" is supported by the assertion that HPC clusters are easy to use in research [23]. Also, a small majority disagreed with the statements "I find it not easy to use a µHPC system to install any operating system and application in HPC system"; and "I find it not easy to use a µHPC system to learn how to integrate the cluster-based HPC system" is supported by the asserted characteristic of ease of use of HPC clusters in research [12][23].

In this study, it is interesting to note that the scores for ease of use to learn to configure the network or cluster-based HPC system were much greater than the scores for it not being easy to use to deploy HPC systems. One possibility is that the participants had the background of computer networking, which they studied prior to the implementation of the µHPC system. However, a small majority found the µHPC system easy to use to learn skills and knowledge on how to run parallel applications in the HPC system (although a small minority disagreed).

The analysis of data from the focus groups captures the aspect of ease of use of a sample of participants through the lenses of motivation, prior knowledge, HPC programming environment, and HPC concepts.

A. Prior Knowledge

The evidence of prior knowledge that contributed to ease of use of the µHPC system has been observed. The data has revealed that the intervention should be prior training in prerequisite knowledge and skills enabled participants to use HPC, and participants attested that there should training on the usage of HPC commands prior to the implementation of the µHPC system. There should be also prior training of the Linux operating system to include the execution commands using command prompts.

There were also challenges in the benchmarking process of the µHPC system due to a lack of prior knowledge in networking and Linux commands. This is in line with the findings of [24] with regard to the importance of a prerequisite module before studying HPC [8].

B. Students' Motivation

Participants described the contribution of motivation to the ease of use of the µHPC system. The data showed that working with the command prompt motivated them to flexibly execute more commands than using a graphical user interface and learn the concepts of HPC, i.e., the attribute of learnability [33]. The difference between the graphical user interface and command prompt was mentioned as having impressed them about working with the command prompt. This is partly aligned with research, which has found motivated students' interest in using HPC [36]. The data also revealed that with knowledge of terminal commands of the Windows operating system, it is easy to switch to those commands. This is supported by research that has analyzed the feedback from students working with the command line in the Linux environment [25].

C. Learning Activities

The emphasis on more learning activities helped parallel programming with a µHPC system become easy to learn. The data revealed difficulties in learning parallel programming, but participants mentioned that running and execution of Python parallel programs was easy. This is supported by the need to learn certain programming languages to create parallel programs [26].

The learning activities should concentrate on hands-on assignments, including benchmarking. The data showed that it was hard to install the LINPACK benchmarking application as it was a lengthy process.

The learning activities helped reinforce the ease of use of the µHPC system for participants and help solve the challenge of using the command-line interface in an HPC environment. The data revealed difficulties in remembering Linux commands, and there were challenges in writing Linux commands, especially the issue of case sensitivity. The data revealed the need for regular use of a Linux operating system and the Python language for parallel programming by students to familiarize them with the system, which mirrors the most used operating system in the HPC industry. It was revealed that those students who were familiar with the Linux operating system expressed ease of use of the µHPC system.

D. HPC Programming Environment

There were mixed perspectives on ease of use when explaining the parallel programming environment that supports parallel libraries and parallel programming languages. The research data revealed that the practical exercise in parallel programming using the µHPC system helped participants acquire programming skills, although there were difficulties in learning parallel programming.

However, the data showed that it was easy to run parallel programs written in Python using the µHPC system. Prior knowledge of parallel programming helped in using the µHPC system as a platform for writing and running parallel programs.

E. HPC Concepts

The integration process for the µHPC system has been described as being difficult initially but becomes more easy over time. This emphasizes the importance of experiential learning and hands-on activities [26]. The research data showed that ease of use of the µHPC system to manage processes in an HPC environment was initially challenging. However, it became easy with more practice as it was using computer network concepts. The networking concept of switching was handy in connecting computing nodes and other components of the µHPC system. The portability of components of the µHPC system made use of the artifact easy.

On the other aspect of learning, focus group data showed that portability; affordability; computation capability; support for command prompts; and support for the Python programming language contributed to the ease of use of the µHPC system to learn parallel programming. These attributes were consistent with the survey findings, where the majority found it easy to run parallel applications and accomplish parallel computing programs in the µHPC system.

The data showed that steady power should be available to the master node and computing nodes when running the benchmark application. The solution was to use a more powerful power adapter. This is possible as the nodes consist of low-power and low-performance processors.

The deployment of the µHPC system faced many challenges [38], ranging from access to libraries and dependencies to the Internet, and power outages, and there were difficulties in deploying in other environments where the Internet was not available. This view is supported by research on the challenges of building and running clusters in the HPC environment, including the challenge of cluster architecture, programming models, resource sharing, and running applications in an HPC environment [22].

Generally, data showed that it was easy to use the µHPC system to install and download applications and use commands to install applications. There were challenges in deploying the system to another place that did not have the Internet as it was difficult to download the applications. Further challenges were installing an application from a PC to a Raspberry Pi as some applications took much time to install.

The participants mentioned that it was easy to execute commands in the µHPC system, but only some applications were taking much time. This was described by their narrative of their experience of installing different applications. The data showed that it was easy to execute the installation commands. However, it was difficult to install some applications. Some of the links of the dependencies on the Internet were different compared to the installation manual, and participants experienced difficulties locating required dependencies packages for the parallel software.

VI. CONCLUSION

This study has demonstrated the ease of use of the implemented µHPC artifact from the perspective of participants who used it during its implementation. We used HPC concepts, the HPC programming environment, learning activities, students' motivation, and prior knowledge as the themes to organize qualitative data used to evaluate the system. The study has articulated that it is easy to use a µHPC system for managing, integrating, deploying, installing, downloading, and running program systems except for writing parallel programs. However, participants stated that it was not easy to deploy and use Linux operating systems installed in the µHPC system, which made it difficult to manage and configure. That reinforces the need for prior training to learn HPC commands. Participants also mentioned the challenges of bad Internet connections and electrical power, hindering the crucial process of configuring the HPC.

Evidence from focus groups strongly confirms the findings in the survey, which suggest that the usage of affordable µHPC clusters has influenced students who participated in the implementation of HPC. The majority have reported positive ease of use of the µHPC system to acquire HPC knowledge and skills during the course. However, a small majority found the µHPC system easy to use to learn skills and knowledge on how to run parallel applications in an HPC system.

The analysis of focus group results has indicated that there should be prior training of students in the areas of network administration, operating systems, computer architecture, and parallel programming. The µHPC system is desirable for learning HPC concepts because of ease to use, affordability, and portability.

The study has a number of strengths. The data collection was done from the actual users of the µHPC system, and analysis was done both quantitatively and qualitatively. Besides, interviews were conducted in the actual environment where the implementation of the µHPC system was done. Thirdly, the triangulation of data collection and analysis methods strengthen the findings.

This study contributes to the insight into the experience of implementation of µHPC in the context of education, but the study also has limitations. The study used 40 participants since only that number was familiar with or learning about HPC systems at the time of the study, and this limits the significance. A study with larger sample size is required to confirm the findings, but the results of the current study are highly promising.

The results of the study provide important implications for HPC education. The feedback from the experience of participants suggests the need for prior training before the intervention. The intervention's content should include system administration using the Linux operating system, computer architecture, parallel programming, and computer networks. By familiarizing participants with the contents of the intervention prior to the demonstration of the µHPC system, it is expected that participants' understanding will be improved.

This study's findings suggest a number of future directions for research aimed at understanding the usability of µHPC in an education context. In particular, future research should examine the usefulness of a µHPC system as an educational tool to support engineering education.

REFERENCES

[1] E. Ackermann, “Piaget’s constructivism, Papert’s constructionism: What’s the difference,” Future of Learning Group publication, 5(3), 438, 2001.

[2] ARM-SBC-Benchmarks-RPI-B-Plus, http://openbenchmarking.org/result/1502103-KWAR-140722031, accessed on 12th February 2015

[3] C. Chen, “A constructivist approach to teaching: Implications in teaching computer networking implications in teaching computer networking,” Information Technology, Learning, and Performance Journal, 21(2), 17, 2003.

[4] S. J. Cox, J. T. Cox, R. P. Boardman, S. J. Johnston, M. Scott, and N. S. O’brien, “Iridis-pi: a low-cost, compact demonstration cluster.” Cluster Computing 17(2), 349-358, 2014.

[5] E. Deci, and R. Ryan, "Intrinsic Motivation and Self-Determination in Human Behavior," New York, NY: Plenum Press, 1985.

[6] A. K. Dennis, "Raspberry Pi Super Cluster," Birmingham, UK: Packt Publishing, 2013.

[7] L. Faessler, H. Hinterberger, M. Dahinden, and M. Wyss, "Evaluating student motivation in constructivistic, problem-based introductory computer science courses," In E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (pp. 1178-1185). Association for the Advancement of Computing in Education (AACE), October 2006.

[8] V. Holmes, and I. Kureshi, "Developing High Performance Computing Resources for Teaching Cluster and Grid Computing Courses," Procedia Computer Science, 51, 1714-1723, 2015.

[9] D. H. Jonassen, and T. C. Reeves, "Learning with technology: Using computers as cognitive tools," In D. H. Jonassen (Ed.), Handbook of research for educational communications and technology (pp. 693-719). New York: Macmillan, 1996.

[10] P. S. Pacheco, "An Introduction to Parallel Programming," Morgan Kaufmann Publishers, Burlington, MA, 2011.

[11] D. N. Perkins, "Technology Meets Constructivism: Do They Make a Marriage?" In Duffy, T.M. & Jonassen, D.H. (Eds.), Constructivism and the Technology of Instruction: A Conversation. Hillsdale, N.J.: Lawrence Erlbaum Associates. pp. 45-55, 1992.

[12] A. J. Pounds, R. Nalluri, and B. L. Coleman, "The development of a Tri-Use cluster for general computer education, high performance computing education, and computationally intensive research," In Proceedings of the 43rd annual Southeast regional conference-Volume 1 (pp. 345-348). ACM, March 2005.

[13] G. Salomon, D. N. Perkins, and T. Globerson, "Partners in cognition: Extending human intelligent technologies," Educational Researcher, 20: 2-9, 1991.

[14] J.D. Sloan, "High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI," O'Reilly Publishers, Sebastopol, CA, 2004.

[15] T. L. Sterling, J. Salmon, D.J. Becker, and D.F. Savarese, "How to Build a Beowulf: A Guide to Implementation and Application of PC Clusters," Cambridge, MA: MIT Press, 1999.

[16] B. Wilkinson, and M. Allen, "A State-Wide Senior Parallel Programming Course," IEEE Transactions on Education, 42.3: 167 – 173, 1998.

[17] Windows 10 for Raspberry Pi 2 ,https://dev.windows.com/en-us/featured/raspberrypi2support, accessed on 12th February 2015

[18] K. L. Pokorny, "Constructing a Beowulf cluster as a pedagogical tool," Journal of Computing Sciences in Colleges, 20(4), 104-105. 2005.

[19] M. Spigarolo, and R. Davoli, "Berserkr: A virtual beowulf cluster for fast prototyping and teaching," In Proceedings of the 1st conference on Computing frontiers (pp. 294-301). ACM, April, 2004.

[20] R. J. Wieringa, "Design science methodology for information systems and software engineering," London: Springer, 2014.

[21] S. T. March, and G. F. Smith, "Design and natural science research on information technology," Decision support systems, 15(4), 251-266, 1995.

[22] V. V. Kindratenko, J. J. Enos, G. Shi, M. T. Showerman, G. W. Arnold, J. E. Stone, J.C. Philips, and W. M. Hwu, "GPU clusters for high-performance computing," In 2009 IEEE International Conference on Cluster Computing and Workshops (pp. 1-8). IEEE, August, 2009.

[23] Q. Zhou, S. Bai, C. Li, and W. Zhang, "Affordable high-performance environment for education in undeveloped regions," In 2008 IEEE International Symposium on IT in Medicine and Education (pp. 911-914). IEEE, December, 2008,.

[24] K. Stewart, "HPC undergraduate curriculum development at SDSU using SDSC resources," In Supercomputing'95: Proceedings of the 1995 ACM/IEEE Conference on Supercomputing (pp. 19-19). IEEE, 1995.

[25] T. Kendon, and B. Stephenson, "Unix literacy for first-year computer science students," In Proceedings of the 21st Western Canadian Conference on Computing Education (p. 14). ACM, May, 2016.

[26] K. Doucet, and J. Zhang, "Learning cluster computing by creating a Raspberry Pi cluster," In Proceedings of the SouthEast Conference (pp. 191-194). ACM, April, 2017.

[27] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, "User acceptance of computer technology: a comparison of two theoretical models," Management science, 35(8), 982-1003, 1989.

[28] O. Abuzaghleh, K. Goldschmidt, Y. Elleithy, and J. Lee, "Implementing an affordable high-performance computing for teaching-oriented computer science curriculum," ACM Transactions on Computing Education (TOCE), 13(1), 3, 2013.

[29] R. A. Krueger, and M. A. Casey, Focus groups: A practical guide for applied research. Sage Publications, 2014.

[30] A. Bryman, and B. Burgess, (Eds.). "Analyzing qualitative data. Routledge," 2002.

[31] H. Burkhart, D. Guerrera, and A. Maffia, "Trusted high-performance computing in the classroom," In Proceedings of the Workshop on Education for High-Performance Computing. (pp. 27-33). IEEE Press, 2014.

[32] M. Bloor, J. Frankland, M. Thomas, and K. Robson, "Focus Group in Social Research," London: Sage, 2001.

[33] V. Venkatesh, "Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model," Information Systems Research, 11(4):342-65, 2000.

[34] S. Purao, "Design research in the technology of information systems: Truth or dare," GSU Department of CIS Working Paper, 45-77, 2002.

[35] C. Peck, "LittleFe: parallel and distributed computing education on the move," Journal of Computing Sciences in Colleges, 26(1):16-22, 2010.

[36] S. Cui, Y. Wang, , L.Li, X. Peng, and B. Yalvac, "Introducing High-Performance Computing to Undergraduate Students," In American Society for Engineering Education Annual Conference and Exposition, New Orleans, LA, USA, 2016.

[37] T. Perea y Monsuwé, B. G. Dellaert, and K. De Ruyter, "What drives consumers to shop online? A literature review." International Journal of Service Industry Management, 15(1), 102-121, 2004.

[38] J. Anderson, P. J. Burns, D. Milroy , P. Ruprecht , T. Hauser, and H. J. Siegel, "Deploying RMACC summit: an HPC resource for the Rocky Mountain Region," In Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact (p. 8). ACM, July 2017.

[39] N. K. Denzin, "Triangulation: A case for methodological evaluation and combination," Sociological Methods, 339-357, 1978.

[40] J. Nielsen, and T.K. Landauer, "A mathematical model of the findings of usability problems," In Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems, 206-213, 1993.

[41] L. Faulkner, "Beyond the five-user assumption: Benefits of increased sample sizes in usability testing," Behavior Research Methods, Instruments & Computers, Vol. 35, 379-383, 2003.

[42] K.L. Schuh, "Knowledge construction in the learner-centered classroom," Journal of Educational Psychology, 95, 426-442, 2003.

[43] E. Bredo, "The social construction of learning," In G. Phye (Ed.), Handbook of academic learning: The construction of knowledge(pp. 3 – 45). New York: Academic Press., 1997.

[44] P. Cobb, and J. Bowers, "Cognitive and situated learning perspective in theory and practice," Educational Researcher, 28(2), 4-15, 1999.

[45] BeagleBone Black, http://beagleboard.org/black, accessed on 12th November 2014

[46] Credit card-sized ODRIOD-C1 is a quad-core computer for only $35, http://www.pcgamer.com/credit-card-sized-odriod-c1-is-a-quad-core-computer-for-only-35/, accessed on 19th Dec 2014

[47] Jetson TK1,https://developer.nvidia.com/embedded/buy/jetson-devkit, accessed on 23rd Jan 2015.