ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 4 (2011) 1917–1926
Procedia Computer Science

International Conference on Computational Science, ICCS 2011

Linux Cluster in Theory and Practice: A Novel Approach in Teaching Cluster Computing Based on the Intel Atom Platform

Andy Georgi[1,*], Stefan Höhlig[1,**], Robin Geyer[1,**], Wolfgang E. Nagel[1,2,**]

Technische Universität Dresden, Faculty of Computer Science, Chair of Computer Architecture

Abstract

Current trends and studies on future architectures show, that the complexity of parallel computer systems is increasing steadily. Hence, the industry requires skilled employees, who have in addition to the theoretical fundamentals, practical experiences in the design and administration of such systems. However, investigations have shown, that practical approaches are still missing in current curricula, especially in these areas. For this reason, the chair of Computer Architecture at the faculty of Computer Science at the Technische Universität Dresden, developed and introduced the course "Linux Cluster in Theory and Practice" (LCTP). The main objectives of this course are to provide background knowledge about the design and administration of large-scale parallel computer systems and the practical implementation on the available hardware. In addition, students learn how to solve problems in a structured approach and as part of a team. This paper analyzes the current variety of courses in the area of parallel computing systems, describes the structure and implementation of LCTP and provides first conclusions and an outlook on possible further developments.

Keywords: teaching, practical course, systems engineering, systems administration, parallel computer systems

1. Introduction

Developments within the Top500 list [1] show, the number of processors in parallel computer systems is increasing steadily. While still 70.4 % of the systems have less than 2^{13} processors, they are accountable for only 32.1 % of the total power in the Top500 list. Already, 20 % of the total computing power in the Top500 list can be achieved by systems with more than 2^{17} processors. Also, a study on ExaScale Computing [2] shows, that this trend is expected to continue. These developments lead to more complex systems, whereat their design and administration determine

*Principal corresponding author
**Corresponding author
Email addresses: andy.georgi@tu-dresden.de (Andy Georgi), stefan.hoehlig@zih.tu-dresden.de (Stefan Höhlig), robin.geyer@zih.tu-dresden.de (Robin Geyer), wolfgang.nagel@tu-dresden.de (Wolfgang E. Nagel)
[1]URL http://tu-dresden.de/zih
[2]Head of Department

Selection and/or peer-review under responsibility of Prof. Mitsuhisa Sato and Prof. Satoshi Matsuoka
doi:10.1016/j.procs.2011.04.209

which computing performance can be achieved at the end. Hence the design and administration of such systems have a decisive role to play.

In spite of this crucial role, Section 2 will show that there is still a lack of practical approaches in the curricula of the courses Computer Science, Information Systems Technology and Systems Engineering. To close the resulting gap between study and professional practice, the chair of Computer Architecture at the faculty of Computer Science at the Technische Universität Dresden, designed and introduced the course "Linux Cluster in Theory and Practice" (LCTP). The actual delimitation of existing teaching concepts towards LCTP is given in Section 3, followed by the description of the structure of our course in Section 4. Additionally to the teaching of theoretical knowledge of the design and administration of parallel computer systems, LCTP contains, practical courses, in which the students apply what they have learned. This requires the provision of adequate hardware. The requirements for the hardware were defined primarily by low acquisition and maintenance costs, space-saving placement and high flexibility with respect to the transport and the design of the systems. The resulting challenges and their solutions are discussed in Section 5. Finally first experiences, both by the teachers, as well as students, are described in Section 6. Furthermore, an outlook on future development opportunities of the course is given.

2. Related Work

The study of the architecture, programming and analysis of parallel computer systems is an integral part of the curriculum of almost all universities. We will show how many of these courses are structured and at which points we can find conceptual problems. In addition, publications on this subject are discussed exemplary as a basis for developing a more comprehensive teaching concept. Apart from the conceptual criteria there are problems independent from the area of studies which influenced the design of LCTP.

2.1. Content of Current Courses

Courses which deal with parallel computing systems can be roughly divided into three categories:

- Architecture of Parallel Computer Systems
- Parallel Programming
- Performance Analysis

The architecture of parallel computer systems is often taught during the basic studies within the complex of themes "Computer Architecture". Many universities also offer advanced courses in the main studies. In total, the students obtain a wide basic knowledge of parallel and scalable architectures. Additionally, in some cases exercises are offered in which students can consolidate their acquired knowledge. Investigations about existing courses in this area have revealed, there is no course which offers a practical approach of the architectures taught. In detail, this means the students learn factual knowledge about the different components, but do not get the chance to build and set up a parallel computer system on their own. We assume the causes for this are high personnel expenditures and hardware costs.

In the area of parallel programming, practical courses have already been established, in addition to the lectures and exercises. Primarily, this is possible because of the broad availability of multi-core systems. Thus, parallel programming concepts can be implemented on commercial, off-the-shelf hardware. However, an interconnect network is not taken into account on a single desktop system, even though it has a major impact on application performance. Therefore, it is necessary to perform such courses on distributed computer systems. Most institutions maintaining parallel productive systems do not offer them student courses. For this purpose, the University of Stuttgart provides a dedicated system for the students [3]. The first advantage is, that the production operation is not disturbed. Secondly, shorter waiting times for the execution of a program arise. However, further acquisition and maintenance costs occur due to the additional system.

The performance analysis of computer systems is an essential component for their development, acquisition, use and evaluation. This subject is already a long time part of higher education. Accordingly, there are numerous lectures, seminars and practical courses. While there was nothing to criticize within the theoretical content, the practical courses are implemented by using simulators, desktop computers or on parallel systems. Simulation of hardware offers the advantage of flexibility and time efficiency. However, there is the disadvantage that the results usually differ from those on real systems and students get not familiarized with the handling of them. For the conduction of performance measurements, desktop systems can be used only for teaching a general approach due to the lack of massive parallelism. In principle, the execution on parallel computer systems, is evaluated positively. The system settings have a great influence on the resulting performance. Investigation of these parameters require administrative rights, which have not been provided by any of the courses found.

In summary, it can be determined that there is a sufficient provisioning of theoretical principles in all three areas. Apart from the parallel programming, deficits exist in the realization of the content taught.

2.2. Scientific Publications

The University of Colorado, Boulder (UCB) describes in a publication from 2004 [4] the development, subsequent failure and the reintroduction of a course about High-Performance Scientific Computing (HPSC). It contains important aspects which had influenced the design of our course. Firstly, it can be stated that the financing of a new course is possible, but often no further funds for the maintenance or development are provided. Secondly, the rapid technical development and the associated process of updating the teaching materials, are a major challenge. At the UCB, the problems could be solved by purchasing a Beowulf Cluster [5] and the use of Linux as operating system. Thereby costs could be reduced and an environment was created which can be used independently from other providers and stays constant over a longer period of time. It is also shown that parallel concepts can be implemented with the help of smaller systems.

In 2005, the Friedrich-Alexander University Erlangen-Nuremberg published a technical report about teaching High Performance Computing for Computational Engineering students [6]. In addition to the general structure it also addresses the challenges that arise when students from various disciplines participating the same course. Due to the different prior knowledge, the teaching materials are customized to meet the requirements of all students. This composition can lead to a active exchange of information, as a result of the different points of view, induced by the different background. Another incentive for discussion is given by the distribution of projects and the presentation of the results. Often students are interested in solving the problem even beyond the underlying task and acquire additional background information about the chosen topic on their own. The design of LCTP takes both aspects into account.

The need for purchasing a parallel computer system for teaching purposes and their benefits are discussed in a publication of the University of Stuttgart [3]. In particular, the restricted access to existing high-performance computers was the reason for acquiring its own system for teaching. In contrast, the use of the resources of the Center for Information Services and High Performance Computing (ZIH) by students is not a problem at all. However, only a remote access is possible, and it lacks of administrative privileges in order to examine various system parameters. As the students thus already have the opportunity to work on a massively parallel system, it is only necessary to have a platform which can be designed individually and can be used to measure the impact of different system settings. Hence, an installation of the order of the University of Stuttgart is not necessary at the TU Dresden.

2.3. Current Challenges in Teaching

In almost each disciplines within natural sciences and engineering the teachers are confronted with new challenges. The fact that a wealth of information is provided through the internet means that factual knowledge has not to be taught in the kind as in the past. The selection and interpretation of needed information and the way how it can be used for own problem-solving processes are the tasks whose solution should be part of educational systems.

Another challenge which arises out of the amount of available data, is the design of teaching materials. Experiences have shown that the effort for producing teaching materials increase, as soon as one deviate from the conventional concept – the teaching of factual knowledge. Thus, an abstraction is necessary to reduce the expense of time for the periodic reworking.

Furthermore, it is not possible to define a certain time at which all participating students are able to acquire knowledge. Studies like [7], [8] or [9] have shown that the retentiveness depends on the time of day. Thus, an approach is desirable, allowing the students to learn and to solve the tasks almost independent of the time.

3. Delimitation of other Teaching Approaches

In Section 2, we discussed different teaching approaches in the field of parallel computer systems. Based on these insights in this section will be described how LCTP differs in its objectives, content and implementation from existing lectures.

3.1. Objectives

The current trends in the Top500 list [1], and studies of future computer architectures, such as ExaScale computing [2] show, that the complexity of parallel architectures continues to increase. Thus, the provider as well as the system engineers and administrators facing new challenges. As shown in Section 2.1, there currently exists a lack of practical approaches within the relevant courses of studies. The main objective of this course is to provide background knowledge about the administration and design of large-scale parallel computer systems and the practical implementation of this knowledge on the available hardware. So, the gap between theory and professional practice is closed.

Furthermore, as described in Section 2.3, the development of own problem-solving processes is of major importance. To achieve this the students are led into known problem situations, which they have to cope with mostly on their own. This implies the search and filtering of relevant information out of provided or freely available materials. Learning this approach of solving problems is also an essential component of the course.

3.2. Content

Section 2.1 has shown, that a number of available courses deal with the use and analysis of parallel computer systems. LCTP takes this content and adds the practical implementation and the presentation of concepts and tools that are necessary to provide and maintain such systems. The course is roughly divided into the design and construction of the system, installing the necessary software, test of functionality by other groups and finally the usage for scientific applications. Apart from the available hardware and the defined functionality no guidelines are made, so the students have free choice in the implementation, as far as this is possible. Not all possible software-combinations can be studied in advance. Recommendations are made for different implementations, where the tutors are able to interfere in case of need. Such an approach could not be found at any other university.

3.3. Implementation

For each subject area LCTP has both, a theoretical and a practical part. The ratio of theory to practice is about 1:3. The lectures are, as described in Section 2.3, widely abstracted, in order to explain the general architecture and operation of the various components. It is not the objective to teach the greatest possible amount of factual knowledge, but the needed basics. So the students are able to read up on the current topic on their own. For this, selected sources are quoted by which additional research is possible. Hence, the participants are able to prepare themselves independently and at a freely chosen time for the practical part of the course.

The tasks for each topic are given subsequent to the theoretical instructions and the autonomous familiarization. As it is common at almost all universities, official events are available on which tutors are present to support the students in case of need. On the other hand it is also possible to solve the tasks independently from the offered dates. In order to discuss problems in this case, various communication platforms are provided. With help of these communication resources, not only the tutors but also other groups can provide assistance. The students are not tied to specific times at which the tasks must be solved. Only the date at which the solution must be present is defined.

Experiences have shown that the use of a project management tool is an advantage in respect of communication, organization and control of the students and tutors. Therewith, for example, it is possible to report encountered problems, to document the current state of the tasks or to provide additional material. Many of these tools provide statistics that help to evaluate the course afterwards. This information can be used by the teachers to optimize the course in the next semester.

4. Design of the Course

The concept discussed in this paper is subdivided into four modules. In the first module, the students have to layout a design for their cluster which is discussed individually afterwards. If it meets the requirements the design is actually set up with the given hardware. The second module contains all activities needed to make the cluster usable for a real user. This contains for example, the installation of all basic services and the provision of libraries, compilers and applications. In the third module the cluster is tested by other groups in a cross-testing procedure. Thereby, the benchmarking is of particular importance. Finally, the fourth module contains the actual usage of the cluster by the groups. Therefore they run real world scientific applications and self-developed parallel software. Also a final task is part of the this module.

As mentioned in Section 3.3, several tasks are included in each module of the course. For every task a introducing lecture is provided, the actual solving of the given tasks is done in practical courses afterwards. Accompanying to the practical courses a report has to be maintained in a version management system. The approval of every solution contains a talk with the tutor, a function test of the system and a review of the report.

4.1. Module 1 – Design and Setup

Besides organizational objects, like security instructions, the first module starts with an extensive lecture about the basics of cluster computing. Using the knowledge of these lecture the students have to design their systems. A major constraint to the designs is the preset hardware. The type and count of the given hardware is fixed and can not be altered. Hence, the layout to be made is rather the logical structure of the system than the hardware design. These logical structures are e.g. identifiers for the nodes, the network plan, the choice of the operating system and the distribution of shared directories. Later the design is accepted or declined by the associated tutor. Therefore the groups have to explain their designs. The most important corner points for a acceptable design are:

- The division of the system in a head- and more than one compute-nodes
- A numbered name scheme
- Correct network parameters
- Exporting of the home directory and one other shared folder from the head- to all compute-nodes
- Partitioning of the hard disks
- Choice of the needed basic network services (DNS, DHCP, LDAP, NTP, ...)

If the content of the design is correct, the group is allowed to set it up on the given hardware. The approach is first to install a Linux base system on the head-node and one compute-node. The commissioning of the rest of the compute-nodes is done by “cloning” later. Once no further physical access to the cluster is needed the students are allowed to install the rest of the system and services by remote access.

4.2. Module 2 – System Configuration and Services

The implementation of the designs requires the installation of additional services. The first task in this module is to secure the system. Different network security settings have to be adjusted. Therefore, port white-listing using IPTables, gray-listing against brute force attacks with suitable tools and limitation of the root login have to be implemented on the head-node. Further concepts can be implemented by the students if wished and if the overall performance of the cluster system is not affected. Once the security concept is implemented, the students are allowed to use remote access to set up the rest of the services. From this time on, the solving of the tasks is no longer restricted to the course hours. Successively all other services are installed from this point. The corresponding server daemon is always running on the head-node. The only functional compute-node acts as the test client at this time. The following services are particular notable:

- NFS Server – The compute-nodes mount /home and one additional directory from the head-node. The additional directory later provides system wide libraries, applications and data for all users.

- LDAP – The head-node offers the complete user database to the compute-nodes, except root, via LDAP. On the compute-nodes pamldap is used for login and authentication.
- DNS – The cluster get its own private domain, therefore an DNS server has to run on the head-node. This DNS server also answers requests from the compute-nodes for external domains.
- NTP – All compute-nodes synchronize their clocks via a NTP server running on the head-node.
- Batchsystem – A freely available batch system and scheduler have to be used for managing resources and the access to them.
- MPI – A MPI environment has do be set up for jobs which should spawn over more than one node.

Additional component of the second module is the installation of different compilers, libraries and applications. For instance the Intel Compiler Suite [10], the libraries libgmp [11], libmpfr [12] and miscellaneous others belong to them. The applications to be installed depending partially on the final task and can also be installed later if needed. However, for the cross-testing BenchIT [13] as a measurement environment and one scientific application is needed to be installed. With these two tools the "user-group" is able to evaluate the performance and test the usability from the users perspective.

At the end of the setup module, the students are supposed to commission the rest of the compute-nodes. The tool for this task can be chosen freely. Again, a corresponding lecture gives an overview about a subset of usable tools. The only constraint is, that all nodes are cloned simultaneously, with full bandwidth over the network.

Some problems with the commissioning have to be considered, for instance UUID entries for hard drives and udev entries for the network adapters. Later, the host names have to be automatically distributed via the DHCP server.

4.3. Module 3 – Cross-Testing

Failures which may be caused by the cloning and adverse functionality from the users perspective are investigated by an other group. The group of testers invent individual test cases with help of the measurement environment and the scientific applications. Some test cases are already given. These test cases concerning the correct accounting of the used computation time and the error-free communication with all nodes. If the testers need additional software or a special configuration, the group which "own" the system under test has to provide this software to them.

A particular part of the testing is the performance analysis of the cluster. This is a kind of competition. Both groups, the administrators as well as the testers, are interested to deliver good benchmarking results. The administrator group want to have a well performing system because otherwise they maybe have done a improper administration job. The testers want to optimize their measurement kernels as much as possible to measure good performance. This leads to an intensive tuning of both the systems and the performance analysis tools.

For the documentation of the bugs found and the measurement results, a groupware solution with a bugtracker and a wiki is offered.

4.4. Module 4 – Usage and Final Task

When the system is running properly and the tutor has conducted the acceptance testing, the system can be used for the final task. Different lectures are offered to give the students an overview over the subject of parallel programming and some scientific applications. These lectures are supposed to motivate the students for making experiences with scientific software on their own. Furthermore, the students are encouraged to program own simple parallel software.

At the end of the course the students are supposed to solve a final task. Different tasks are at choice which are particular useful from the tutors perspective. Mainly these tasks are performance evaluation, programming of larger parallel applications and programming of tools for the cluster administration. If students want to invent a task on their own, it is possible to discuss this with the tutors and phrase a clear task description. In this way it is possible to meet special interests of the students and increase the motivation.

The result of the final task is a report of about ten pages. It is appended to the general report maintained during the course. The solution of the final task is presented in front of a public audience. Therefore a separate presentation has to be elaborated. The general and final report, as well as the presentation, are accounted for the final grade.

5. Deployed Hardware

The main criteria when acquiring a cluster are peak performance, cost, dimensioning of air-conditioning and performance per watt. For educational purposes this applies only partly. Low budgets and availability of needed hardware among other things determine the chosen cluster components in that area. Because of a related situation, the idea of Beowulf Clusters [5], using cheap commercial off-the-shelf hardware, for building parallel computer systems, was invented. In case of LCTP, limited space for storing the machines, mobility and the absence of air-conditioning represent additional constraints for choosing hardware components.

5.1. Platform and Peripheral Equipment

Considering their high availability and their low costs, x86-architecture based platforms were preferred. Due to the initially mentioned facts, the clusters were created upon the Intel Atom architecture. The main advantages compared to other x86-based platforms are the high cost-efficiency concerning costs, power consumption and needed space for placement. The actual value is about 70 $ for one mainboard with CPU and cooling, which is less expensive then common x86-based hardware. Currently the processor types Atom 330 (Diamondville) [14] and Atom D510 (Pineview) [15] are used. In combination with adequate mainboards, they form the basis for the cluster-nodes. The architecture itself was developed for NetBooks and Barebone computers and has a low power consumption, which leads to a small heat generation. Hence, the costs for power supply, air-condition, operation and spare parts are marginal compared to common server-hardware based clusters. The usual form factor for Atom-based mainboards is mini-ITX. The small dimensions allow a high packing density with relatively low weight. This typically plays a minor role for fixed installations, but is important for LCTP.

In practice there are two different approaches to implement the start-up of a node. Firstly, by saving the operating system on a local permanent storage from which it can be booted. Secondly, by a server delivering the operating system by request via an interconnect network. Subsequently, the client tries to boot the received data. Both concepts exhibit advantages and disadvantages but attention was paid to leave both feasible. Therefore, 2 GiB of main memory and a hard disk are installed in each node. This offers further advantages, like saving working data locally, local checkpointing and swapping.

For the interconnect network, Gigabit Ethernet technology is used. On the one hand, the network interface integrated in each node is used for the internal connection of head- and compute nodes. On the other hand, there are external network interfaces connected via USB with the head node, for the integration into the network of the university. While all used mainboards offer a PCI slot which allows to make use of a network interface card, reasons for taking an USB-based solution are: higher costs and less availability of PCI devices fitting into the used one HU cases. Furthermore, there is an improved heat conduction because of an unhindered air flow. To connect all nodes of a cluster, a Gigabit Ethernet switch supporting the creation of logical subnets, so called VLANs, is used. This enables parallel communication of more than one cluster over a single physical device without having the applied network services interfering each other. The resulting assembly used for LCTP is shown in Figure 1.

5.2. Deployment for Lecture

An important element of LCTP is the assembling of each node by the students. This is the reason why the initial stage of the lecture requires more space than the following. Furthermore, the work of the students should not interfere the usual work flows in the server room. As a result no fixed installed 19" rack cases are used, but mobile ones, normally used for secure transport of sound equipment. Therefore, it is possible to assemble and set-up the clusters on an adequate place. After configuring a remote connection on the cluster, the integration into the infrastructure of the server room can be performed.

The used mini-ITX mainboards in combination with the correct cases allow to place two nodes into one height unit. Concretely this means eight hardware supported threads, because of dual core and HyperThreading technologies. The count of four compute nodes per cluster necessarily avoids the appearance and the analysis of certain side effects occurring in large installations. As already mentioned in section 2.2 this is not important in our case. The Center of Information Services and High Performance Computing maintains huge installations and gives access to students of the faculty of Computer Science on request. The hardware basis used for LCTP is sufficient for lectures, because the concepts of parallel programming can be implemented and analyzed. Apart from the use of the hardware for the

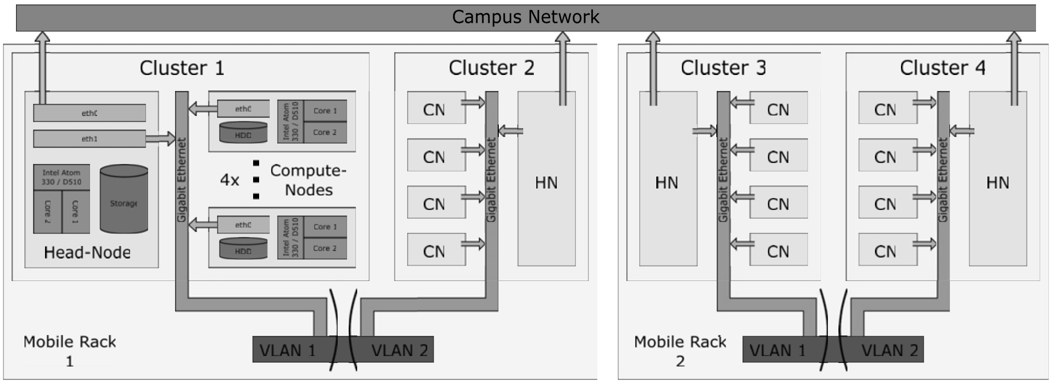

Figure 1: Layout of the current installation for LCTP

lecture there is also the possibility to give other institutions access to the systems, to run coarse-grained applications for instance.

6. Conclusions and Outlook

The majority of lectures which can be found in computer science have a theoretical approach or cover exclusively parallel programming issues. LCTP is a novel, practically aligned lecture which closes the gap between academic education and resolving real-life problems with the help of personal skills. For continuous improvement of its concept and implementation there is a constant request for feedback from the students. The resulting information, the gained experience and possible changes planned for future work will be discussed in this section.

6.1. Conclusions

Participation in LCTP requires previous knowledge of Linux and handling of consoles. Due to the fact that this is not part of the education for computer scientists at the TU Dresden, only the private interests of the students affect these needed skills. Hence, this knowledge is individual. It also complicates the estimation of time students will need to solve the given tasks because it partially requires a detailed acquisition of basic knowledge.

Nevertheless, to get a rough estimate of time and as an assistance in creating tasks for the practical course, a sample cluster was set up by the tutors during the planning phase of LCTP. The procedure, the used utilities and alternative applications, as well as arisen problems and their solutions were documented. Thus a comprehensive reference work was created for the tutors and the structuring of the lecture was simplified. Due to the fact that almost any applied software possesses comparatively short release cycles, the preparation phase is simultaneously used to update and expand the knowledge of the tutors. The resulting alterations in interacting with different applications could lead to unexpected errors which have to be resolved or even require changes in former concepts. Accordingly it is recommended to reassemble the cluster by the tutors every term. Every discovered difference is changed in the documentation immediately.

The concept of LCTP generates a high initial motivation among the students. The combination of alternating theoretical and practical components of the lecture and constantly changing problems, lead to enthusiasm in solving the given tasks and increase perceptibly the learning effect. Furthermore, the independent and creative debugging process and the unavoidable competition between different groups induce a sense of achievement and improve the motivation. In the further progress of the lecture a rapidly rising learning curve is recognizable, especially for students with less educational background in parallel computing. This leads to a much faster process of analyzing and solving problems than at the beginning of the course. The tutors are instructed to engage only in critical situations and primarily have the role of a consultant. As a consequence a lot of autonomous research for information is necessary, which trains the effectiveness in solving future problems.

The first two terms have shown that a high ratio of students to tutors is necessary. In detail one tutor is assigned to a single group. The high personal effort can be explained by the complexity of the tasks. Every group has its own approach to solve problems which leads to different system setups. To intervene in critical situations a tutor always needs to keep the track of the entire system environment. A group in turn consists of four students. A higher number would lead to problems in solving the given tasks because all of them have to work on the same system and often need the resources dedicated. At the moment the course is offered for four groups which requires the use of four tutors. However, it is recommended to offer a smaller amount of groups in the first semester to give the tutors the opportunity to get used to the work.

6.2. Outlook

Previous experiences show the high potential, effectiveness and flexibility of this educational concept. The formation of several small groups in conjunction with the cost-effective and adaptable implementation of the Intel Atom platform, allow the construction of several clusters with different characteristics. For instance there are boards with integrated graphic solutions offering CUDA support. Therefore it is possible to study the behavior of CUDA accelerated applications and their programming. Another way to enhance LCTP is the latest version of the Intel Atom E6x5C series [16] with an integrated Field-Programmable Gate Array (FPGA).

LCTP deals with the design and administration of parallel computer systems as they are common in the High-Performance-Computing area. Such installations are primarily used for processing complex computational problems or to handle a large amount of data sets. Another way in which clusters can be used is the provisioning of services and ensuring their availability. The main objective in this case is not the performance or efficiency but the reliability. So called high-availability clusters have less relevance in science, but play an important role for the industry. It is possible to extend LCTP conceptually, to treat such systems.

The combination with other courses or the introduction of a further lecture based on the current content is also conceivable. This would allow the use of the assembled clusters for programming, testing, analyzing and optimizing parallel applications in the next term. Additionally there is the possibility to learn how to secure such systems and to expand the practical experiences with the help of exemplary attacks.

Moreover, it is possible to use the existing hardware to propagate the concepts of parallel processing at other faculties and institutes, as described in a publication of the Earlham College [17], for instance. Thereby it is also conceivable to use the participants of the course as tutors in such seminars. In this case it would turn out whether the students are also able to share the acquired knowledge.

References

[1] Top500.org, Top500 Supercomputer List (November 2010).
URL `http://www.top500.org`

[2] Peter Kogge, William Dally, et al., ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, Tech. rep., DARPA (September 2008).
URL `http://www.sdsc.edu/ allans/Exascale_final_report.pdf`

[3] M. Bernreuther, M. Brenk, H.-J. Bungartz, R.-P. Mundani, I. L. Muntean, Teaching High-Performance Computing on a High-Performance Cluster, in: V. S. Sunderam, G. D. v. Albada, P. M. A. Sloot, J. J. Dongarra (Eds.), Computational Science – ICCS 2005, Vol. 3515 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2005, pp. 1–9.
URL `http://dx.doi.org/10.1007/11428848_1`

[4] E. R. Jessup, H. M. Tufo, Creating a Sustainable High-Performance Scientific Computing Course, in: International Conference on Computational Science, 2004, p. 1242.

[5] T. Sterling, D. J. Becker, D. Savarese, J. E. Dorband, U. A. Ranawake, C. V. Packer, BEOWULF: A Parallel Workstation For Scientific Computation, in: In Proceedings of the 24th International Conference on Parallel Processing, CRC Press, 1995, pp. 11–14.

[6] U. Fabricius, C. Freundl, H. Köstler, U. Rüde, High Performance Computing Education for Students in Computational Engineering, in: V. S. Sunderam, G. D. v. Albada, P. M. A. Sloot, J. J. Dongarra (Eds.), Computational Science – ICCS 2005, Vol. 3515 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2005, pp. 27–35.
URL `http://dx.doi.org/10.1007/11428848_4`

[7] C. P. May, L. Hasher, E. R. Stoltzfus, Optimal Time of Day and the Magnitude of Age Differences in Memory, Psychological Science 4 (5) (1993) 326–330.

[8] R. West, K. J. Murphy, M. L. Armilio, F. I. M. Craik, D. T. Stuss, Effects of Time of Day on Age Differences in Working Memory., The journals of gerontology Series B Psychological sciences and social sciences 57 (1) (2002) P3–P10.
URL `http://www.ncbi.nlm.nih.gov/pubmed/11773218`

[9] K. Wahistrom, Changing Times: Findings From the First Longitudinal Study of Later High School Start Times, NASSP Bulletin 86 (633) (2002) 3–21. doi:10.1177/019263650208663302.
URL `http://dx.doi.org/10.1177/019263650208663302`

[10] Intel Parallel Studio XE 2011.
URL http://software.intel.com/en-us/articles/intel-parallel-studio-xe/

[11] The GNU Multiple Precision Arithmetic Library.
URL http://gmplib.org/

[12] The GNU MPFR Library.
URL http://www.mpfr.org/

[13] S. B. e. a. Guido Juckeland, Benchit - performance measurement and comparison for scientific applications, in: PARCO2003 Proceedings.

[14] Intel corp., Intel Atom Processor 330 Series Datasheet, Tech. rep., Intel corp. (2010).
URL http://download.intel.com/design/processor/datashts/320528.pdf

[15] Intel corp., Intel Atom Processor D400 and D500 Series Datasheet - Volume One, Tech. rep., Intel corp. (2010).
URL http://download.intel.com/design/processor/datashts/322844.pdf

[16] Intel corp., Intel Atom Processor E6x5C Series Product Preview Datasheet, Tech. rep., Intel corp. (2010).
URL http://edc.intel.com/Download.aspx?id=4193&returnurl=/Platforms/Atom-E6x5C/default.aspx

[17] C. Peck, LittleFe: Parallel and Distributed Computing Education on the Move, J. Comput. Small Coll. 26 (2010) 16–22.
URL http://portal.acm.org/citation.cfm?id=1858449.1858454