Further Investigation on Building and Benchmarking A Low Power Embedded Cluster for Education

Sritrusta Sukaridhoto[1], Achmad Subhan KHalilullah [2], Dadet Pramadihanto [3]
[1]Author name is with Departement of Creative Multimedia, Electronics Engineering Polytechnic Institute Surabaya, Surabaya, Indonesia. E-mail:dhoto@eepis-its.edu
[2]Author name is with Departement of Creative Multimedia, Electronics Engineering Polytechnics Institute Surabaya, Surabaya, Indonesia. E-mail:subhankh@eepis-its.edu
[3]Author name is with Departement of Information and Computer Engineering, Electronics Engineering Polytechnic Institute Surabaya, Surabaya, Indonesia. E-mail:dadet@eepis-its.edu

Abstract— **Embedded parallel computing become popular, and the future of innovation in the semiconductor industry will be in ubiquitous computing. Many researchers built embedded cluster system with limited number of devices, but we utilize the device from embedded classroom to build more number of parallel computing unit. In this paper we built low power cluster consisting 32 ARM boards with low-cost customized power supply for high performance computing class for education purpose, tested with several benchmarks on embedded cluster system and analyse the raw performance.**

Keywords— embedded cluster system, ARM board, benchmark, high computing cluster for education.

Abstrak— **Komputasi paralel menggunakan sistem terbenam menjadi sangat pupular, dan untuk kedepannya inovasi di dunia semi kondukdor adalah ubiquitous computing. Beberapa peneliti telah membangun sistem cluster dengan menggunakan sistem terbenam dengan jumlah yang terbatas, tetapi dengan menggunakan sistem terbenam yang berada di kelas mata kuliah sistem terbenam dapat dibangun komputasi paralel dengan lebih banyak unit. Pada paper ini, kita membangun sistem komputasi paralel yang terdiri dari 32 unit sistem terbenam berbasis ARM dan juga sistem supplai yang sudah dimodifikasi untuk bidang pendidikan, melakukan beberapa test dan benchmark untuk menguji performa dari sistem yang dibangun.**

Kata kunci— embedded cluster system, ARM board, benchmark, high computing cluster for education.

I. Introduction

The large system used today in HPC are dominated by processors that use the x86 and Power instruction sets supplied by big vendors such as Intel, AMD, and IBM. These processors have been designed to mainly cater to the server, desktop PC and laptop market. The processors provide very good single thread performance but suffer from high cost and power usage. One of the main goals in building a HPC system is to stay within a power budget. In order to achieve this ambitious goal, other low power processor architectures such as embedded system are currently being explored since these processors have been primarily designed for the mobile and embedded device market.

Embedded processors (CPUs) can be found in a vast variety of products from cellular phones, digital cameras and up to network-connected household appliances. Some of these embedded processors can run advanced operating systems, such as Linux, to achieve flexible network connectivity and to have logically same functionality as that of high-end processors designed for PC and workstations. With these abilities, to build HPC system using embedded system also possible.

The issue of providing the High Performance Computing for education has been widely investigated in the literature, in particular with reference to embedded systems. In [1][2], the authors made the UCC embedded parallel computing based on SH4 processor, with 4 nodes. Sasaki, et [3], provides about M32RUCC parallel computing based on ARM processor. The node in that modules only using 4 nodes. In [4], they used Virtual Machine (VM) to learn parallel and distributed system. But still to build that system with VM it cost so much money. Balakrishnan [5] built ARM cluster but they built only less then 10 nodes and it will difficult if we use adaptor to give power supply.

One of the problems in education for HPC system is the lack of cost-effective standardized platform for prototyping, testing and evaluating application programs on network-connected embedded CPUs. Addressing this problem, in this paper, we present a compact high performance computing cluster system with embedded CPUs, called “EEPIS Embedded Cluster Computer (EECC)” which provides a rapid-prototyping environment for high performance computing at very low cost and low power consumption compared with conventional PC clusters.

EECC consists of 32 embedded computing nodes and network switches (100Mbps Fast Ethernet) from Embedded Class and customized power supply. The key idea is to fully utilize System on Chip (SoC) embedded products to realize cost-effective prototyping environment. In this context, we selected a commercially available embedded system as a computing node for EECC, which consists of an Dual Processor embedded CPUs, a memory, a storage, a network adapter and I/O interfaces. The computing nodes run Linux with some daemons and libraries required to make inter-processor communication for parallel processing, where MPI (message Passing Interface), and PVM (Parallel Virtual Machine) could be employed for parallel programming.

By the use of SoC products and customized power supply, we achieve compact size of 650mm x 480mm x 130mm similar with 2U rack mount system, low power consumption of 400W to drive 32 embedded CPUs and low cost. Thus, EECC can be easily introduced to educational programs in universities for Linux-based cluster computing. Another interesting feature of EECC is that every computing node has 2 USB interfaces, and hence EECC could be easily extended to various real-world application systems using USB-based sensors, such as USB cameras.

This paper is organized as follows: Section 2 gives the system overview of EECC. Section 3 describes performance test for basic data transfer bandwidth through MPI. Section 4 discusses an implementation of EECC as educational module in the class. In Section 5, we end with some conclusion.

II. System Overview

Figure 1. shows the system overview for “EEPIS Embedded Cluster Computer (EECC)”. It consists of 16 computing nodes from #1, #2, until #16, where the node #1 works as server node for various applications, such as NIS, NFS, and SSH servers, etc., and is directly accessible from terminal outside or from node #1 which connected with monitor and keyboard. These 16 computing nodes, a network switch, and 5V power supply are mounted together. The computing nodes are connected over conventional 100Mbps Fast Ethernet. Multiple EECC can be easily stacked to extend the number of computing nodes.

We decided to use System on Chip (SoC) product with embedded CPUs in building EECC. A PandaBoard ES [7] as shown in Figure 2. is employed as a computing node, is a low-power, low-cost single board computer development platform based on the Texas Instruments OMAP4460 system on a chip (SoC). The OMAP4460 SoC on the PandaBoard features a dual-core 1.2 GHz ARM Cortex-A9 MPCore, 384 MHz PowerVR SGX540 GPU, IVA3 multimedia hardware accelerator with a programmable DSP, and 1 GB of DDR2 SDRAM. Primary persistent storage is via a SD Card slot allowing SDHC card class 10 with 8GB capacity. The board include wired 10/100 Ethernet as well as wireless Ethernet and Bluetooth connectivity. Its size is slightly larger than the ETX/XTX Computer form factor at 4 × 4.5 in (100 × 110 mm). The board can output video signals via DVI and HDMI interfaces. It also has 3.5 mm audio connectors. It has two USB host ports and one USB On-The-Go port, supporting USB 2.0. The PandaBoard has a real-time clock can be synchronize with NTP server, and runs the Linux Kernel with ARM architecture. The detailed specification of PandaBoard used for computing node is shows in Table 1.

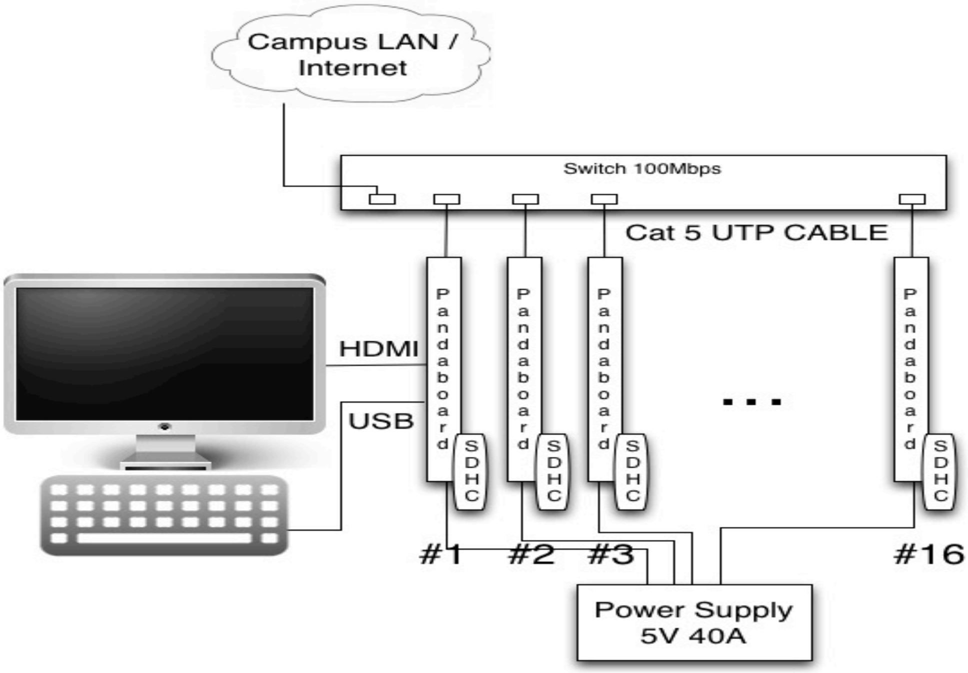

Figure 1. Architecture of the EECC.

The ARM Cortex-A9 in PandaBoard is a 32-bit multi-core processor which implements the ARMv7 instruction set architecture. The cortex-A9 can have a maximum of 4 cache-coherent cores and clock frequency ranging from 800 to 2000 Mhz. Each core in the cortex-A9 CPU has a 32 KB instruction and a 32 KB data cache. One of the key features of the ARM Cortex-A series processors is the option of having Advanced SIMD (NEON) extensions. NEON is a 128-bit SIMD instruction set that accelerates applications such as multimedia, signal processing, video encode/decode, gaming, image processing etc. The features of NEON include separate register files, independent execution hardware and a comprehensive instruction set. It supports 8, 16, 32 and 64 bit integer as well as single precision 32-bit floating point SIMD operations.

Table 1. PandaBoard specification.

PandaBoard ES	
CPU	Dual ARM Cortex-A9 MPCore 1.2Ghz
Memory	1 GB DDR2 SDRAM
Storage	SDHC slot up to 32GB
Network	100Mbps Ethernet 802.11n Wifi Bluetooth
Interface	DVI HDMI 2x USB 2.0 1x USB On-The-Go
Graphics	SGX540 GPU
Audio	3.5 mm
Size	4 x 4.5 in
Weight	81.5 gr
Power Supply	DC 5V, Min 2A ~ Max 4A
Clock	Real-time
OS	Debian Linux 3.7 armhf

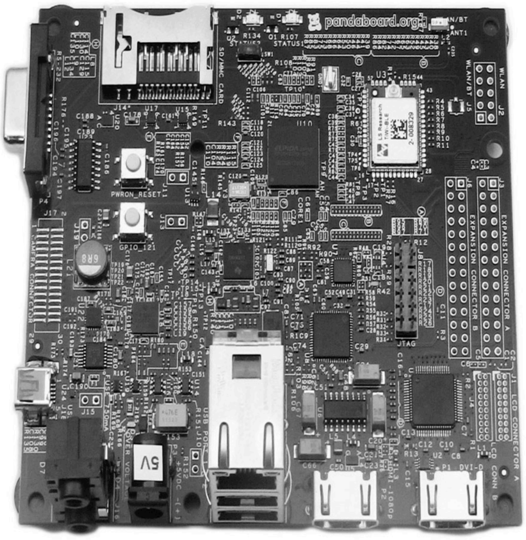

Figure 2. PandaBoard.

EECC uses SDHC Class 10 with 8GB storage as shown in Figure 3. By using this SDHC can be easily to install Operating System such as Linux. SDHC Class 10 provides 30 MB/s transfer rate of data. With this speed gives better performance for running parallel computing inside the node.

Figure 3. SDHC as Storage.

We used DC 5V 40A 200W Transformer Switch Power Supply with cooling fan. With this power supply we can power up 16 nodes of PandaBoard. In each PandaBoard need 5V DC with 2A minimum for requirement to run. Figure 4. shows the type of power supply. To distribute the power supply, we made custom cables distribution board as shown in Figure 5, 6, and 7 to parallelize the supply. With this board also, we can also build stacking power supply to provide power supply for more PandaBoard.

Table 2. Power Supply Specification.

Model		PS200-W1V
Output	DC Voltage	5V
	Current Range	0 ~ 40A
	Rated Power	200W
Input	AC Current	2.5A/230VAC
	Voltage range	170~260VAC
	Efficiency	81%

Figure 4. Power Supply DC 5V 40A to power up 16 nodes of PandaBoard.

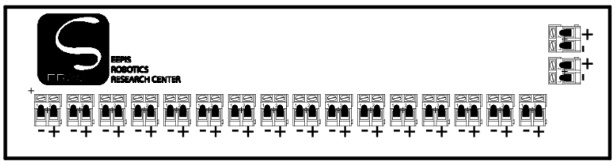

Figure 5. Top view of customized cables distribution board.

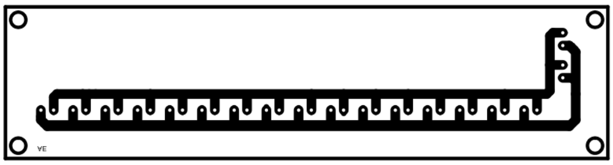

Figure 6. Bottom view of customized cables distribution board.

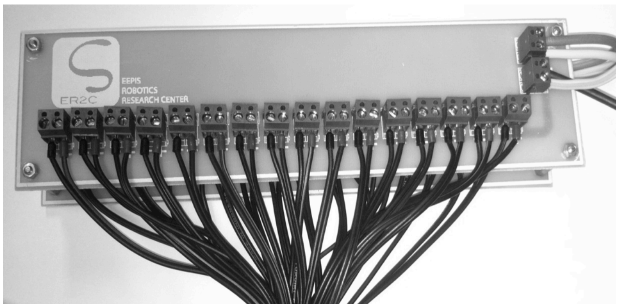

Figure 7. Power supply cables distribution.

For switching device we decided to use 24 ports 10/100Mbps. 16 ports in the switch used for PandaBoard as computing node, the rest of the port can be use as stackable network with other EECC module or can be use to connect to Internet or other LAN. Figure 8. shows the switching device. The dimension for the switch is 28cm x 12.5cm x 4cm, which this size is relatively small.

Figure 8. Switching Device.

The nodes run Debian GNU/Linux [8] 3.7 configured for ARMv7 on SDHC and daemons. Inside EECC there are several applications, which can support for programming, tools and parallel computing.

1. EECC supports a variety of development tools and libraries for parallel programming including GNU C, C++ compiler, and Message Parsing Interface (MPI) by using OpenMPI [9].
2. EECC provides with a variety editor application such as Vim, nano and etc.
3. EECC also includes remote execution communication with secure shell (SSH).
4. EECC uses NFS system to share files between nodes. With midnight commander application user is able to manage file and also send to the outside of parallel computing environment.
5. EECC implements NIS to administer user accounts to be use in nodes.
6. EECC connected with EEPIS's Debian Linux local mirror to update and upgrade software.
7. With shell scripting provided by BASH, user can easily manage and control many nodes.

Figure 9. shows a photograph of a prototype of two 16 nodes EECC modules. We could achieve extremely compact size of 650mm x 480mm x 130mm similar with 2U rack mount system, low power consumption of 400W in total and low cost by employing the SoC embedded devices.

Table 3. Overall specification of EECC.

Employed computing nodes		SoC PandaBoard ES
Number of computing nodes		16
Network interface		100Mbps Fast Ethernet
Power consumption		160W~200W
Size (mm)		650 x 240 x 130
Software	Operating system	Debian GNU Linux 3.7 armhf
	Server function	NIS NFS
	Communication	SSH
	Development	GNU C, C++
		Vim
		OpenMPI

III. Benchmark

In this section we will discuss about the various benchmark that were run on the EECC. The benchmark system chosen based on various performance metrics of the system and cluster that is of interest to HPC applications. We describe performance of basic MPI function on EECC.

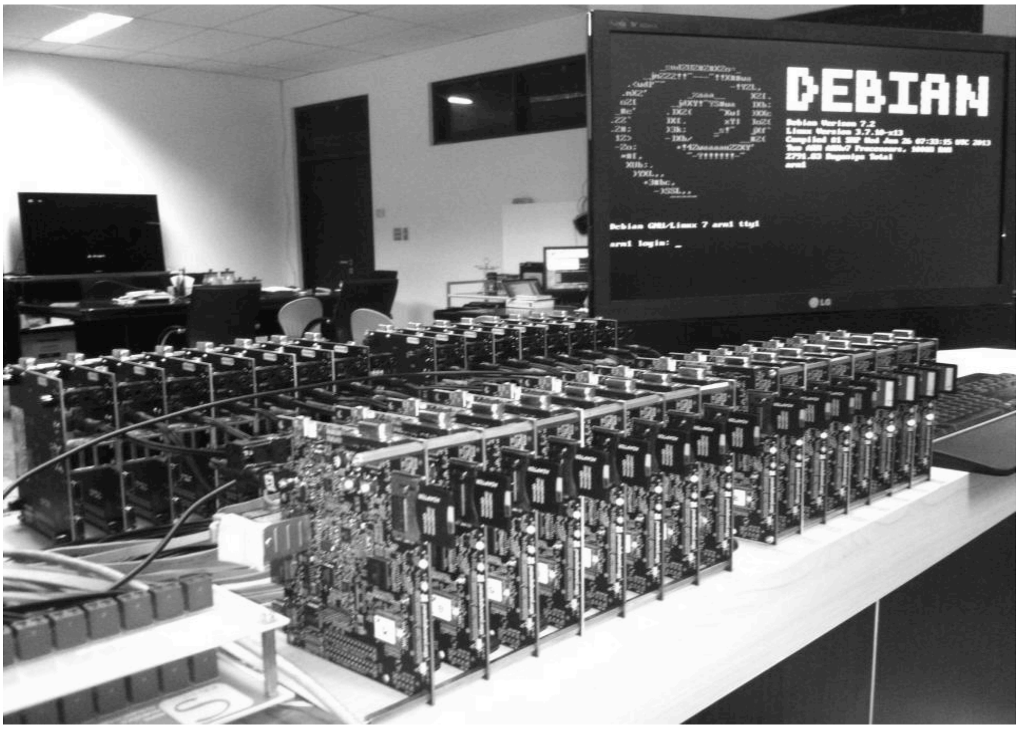

Figure 9. Two modules of EECC.

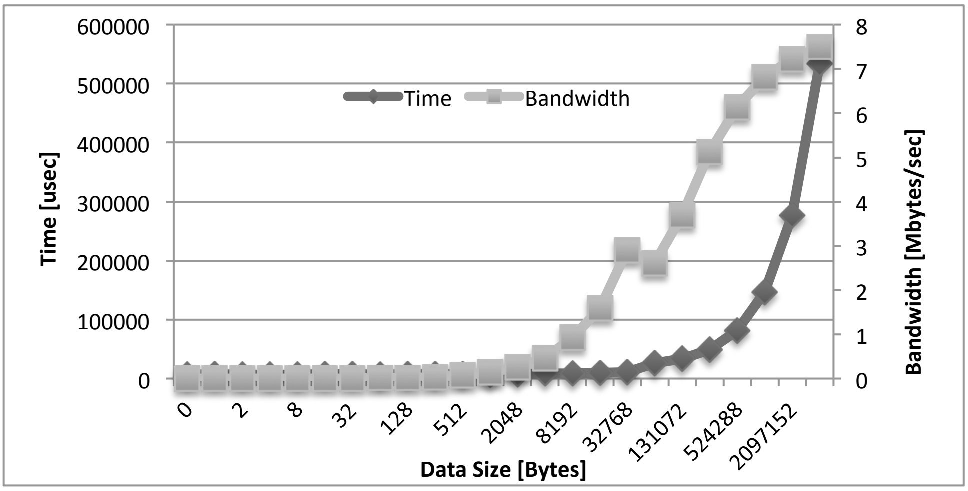

Figure 10. Benchmark for Ping-Pong.

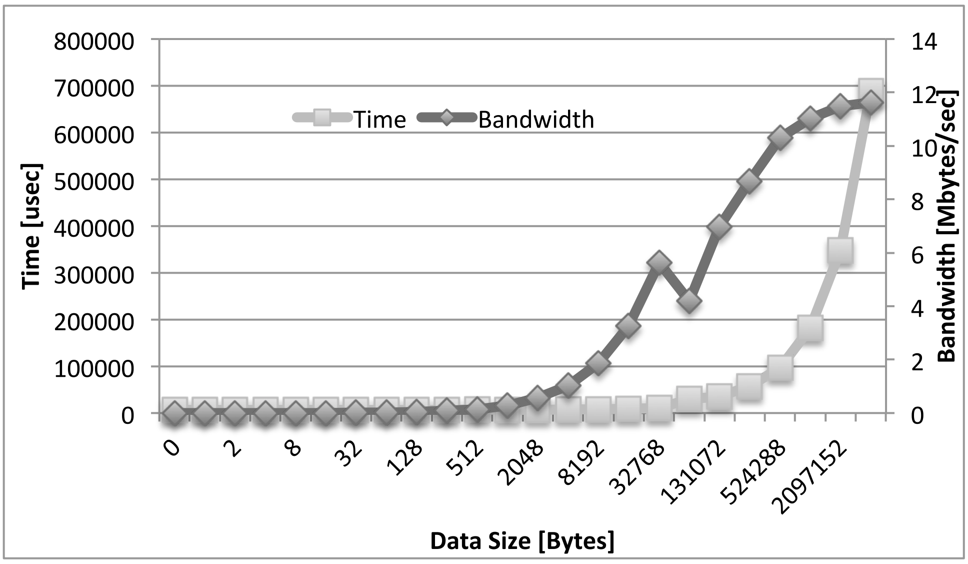

Figure 11. Benchmark for Send-Recv.

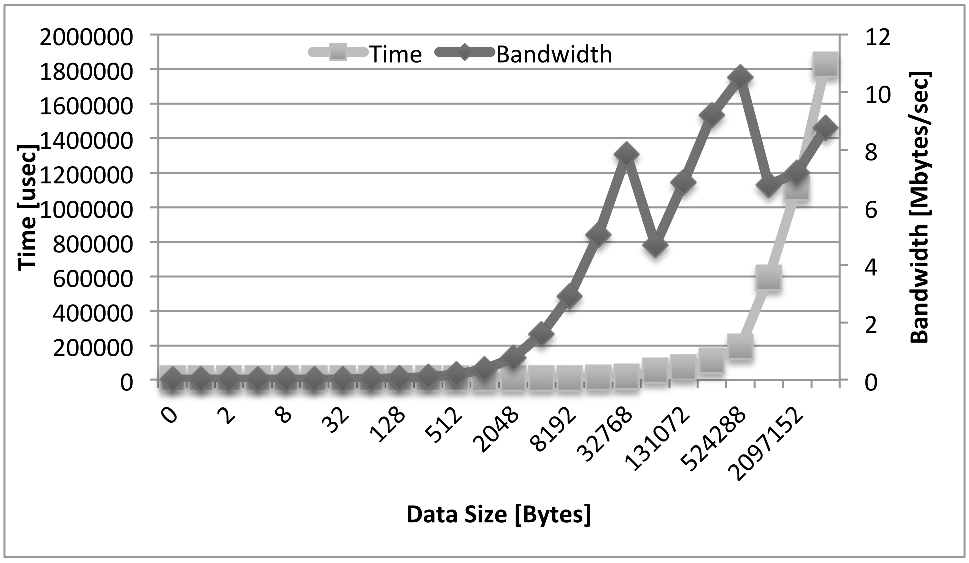

Figure 12. Benchmark for Exchange.

A. Pallas MPI Benchmark (PMB)

Pallas MPI Benchmark (PMB) [9] is employed to evaluate data transfer bandwidth and latency of basic MPI ping-pong, sendrecv, and exchange communications.

Figure 10 shows the result of ping-pong test, where two processors send and receive alternatively. Peak performance of ping-pong communication is estimated

Table 4. HPL results.

Test	Single	MPI		
		min	avg	max
Random Access (GUP/s)	0.009839	0.005963	0.009027	0.010214
DGEMM (Gflop/s)	0.602819	0.48892	0.560694	0.609617
STREAM copy (GB/s)	0.672	0.661977	0.664542	0.672081
STREAM Scale (GB/s)	0.672404	0.662056	0.669736	0.672404
STREAM Add (GB/s)	0.662029	0.655306	0.658719	0.704745
STREAM Triad (GB/s	0.697131	0.682608	0.693598	0.704745

as about 7.5Mbytes/sec (~60Mbps), with this result shows that EECC can communicate with enough bandwidth.

Figure 11 shows the experiment result from send-recv test, where 4 processors do communication send and receive as full-duplex. With all processors sending and receiving packet data, EECC can utilize about 12Mbytes/sec (~96Mbps). It means that EECC system can do high performance communication.

Figure 12 shows the result from MPI exchange communication, where 4 processors send and receive packet data as round robin topology. EECC system gave performance about 10Mbytes/sec (~80Mbps).

B. High Performance Linpack (HPL)

High Performance Linpack (HPL) [10] is a parallel implementation of the Linpack benchmark and is portable on a wide number of machines. HPL uses double precision 64-bit arithmetic to solve a linear system of equations of order N. It is usually run on distributed memory computers to determine the double precision floating-point performance of the system. The HPL benchmark uses LU decomposition with partial row pivoting. It uses MPI for inter-node communication and relies on various routines from BLAS and LAPACK libraries.

Table 4. shown that EECC passed the test comparing with single node in HPL test.

IV. HPC COURSE

The course provides an introduction to advanced computer architectures, parallel algorithms, parallel languages, and performance-oriented computing, and uses real-world case studies from computational science and engineering application domains. As hardware designers turn to multi-core CPUs and GPUs, software developers must embrace parallel programming to increase performance. No single approach has yet established itself as the "right way" to develop parallel software, especially as the hardware evolves so rapidly.

We design the course that starts by overviewing the architecture of modern processors, including multi-core, many-core, and general APUs. This discussion includes not only the computation cores themselves, but also the importance of understanding the memory hierarchy and caching. The course then turns to the programmability of these systems, and works from the ground up: multithreading, higher-level directive and task-based, message-passing, and map-reduce. The course also moves from shared memory to distributed memory to the cloud, showing examples of C++11, CUDA, Thrust, OpenMP, PPL, MPI, and Hadoop to program these systems. Additional topics include measuring

Figure 13. Students with EECC system.

Table 5. Comparison of old HPC system with EECC.

No	Statement	HPC with PC/Mac	EECC	explaination
	Economics			
1	Cost	10	5	To build HPC system with PC/Mac it cost @Rp. 5.000.000, but EECC only @2.500.000
2	Space	10	2	EECC size is same with 2U rack server smaller then PC
3	Power consumption	10	1	PC power consumption more than 550W/PC, but EECC is around 400W for 32 nodes
4	Device replacement	10	10	PC waiting time for new device is longer rather than EECC
5	Easy to install	5	10	In EECC we can unplug the SDHC, reinstall. But PC need to prepare storage and other stuff
6	Number of nodes	5	10	EECC are 32 nodes, PC are 16 nodes
	Performance			
1	Parallel mechanism	10	10	Can do parallel computing using MPI, PVM etc
2	Speedup factor	10	9	Our PC specification is almost the same with PandaBoard
3	Networking	10	10	Run on 100Mbps
4	Storage	10	8	PC have 500GB, EECC have 8GB
5	Interface	8	9	EECC also have HDMI and Wireless communication, not PC
	Education process			
1	Preparation	8	9	Turn on the system
2	Teaching material	10	10	use the same material
3	Student knowleadge	9	9	student can understand the same way
4	Long-life education	9	9	student can find the solution if they found some problems
5	Research	7	9	with embedded system we can try different field of research
6	Exam	8	8	Student can do the task or exam
7	Use devices for different subject	4	9	HPC from PC is dedicated only for HPC subject, while EECC can be use in different subject

performance, linear speedup, Amdahl's law, profiling and debugging tools, types of parallelism (data, task, dataflow, embarrassing), and common patterns (fork-join, reduction, and map-reduce).

Hands-on lab exercises in C and C++ are an integral part of the course; attendees should expect to bring a laptop. The syllabi of the course are as follows:

- Intro to HPC and multicore hardware
- Modern many-core hardware
- Types of parallelism
- Parallel programming in C++11
- The dangers of parallel programming
- OpenMP: a higher-level abstraction
- Task-based parallelism with the TPL
- Tools: debuggers, profilers, and analyzers
- Parallelism at scale: clusters and MPI
- Parallelism at scale: cloud and Hadoop
- Future research directions

The approach is hands-on, the students are expected to use the lecture information, a series of assignments and a final project to emerge at the end of the class with parallel programing knowledge that can be immediately applied to their research projects. The situation when students learn about HPC by using EECC system can shows in Figure 13.

In this section we compared our old HPC system with EECC system for HPC Course. The specification of our old HPC system is G4 Power Macintosh G4 (2003) as shown in Figure 14. We compared in the matter of economics, performance and education process.

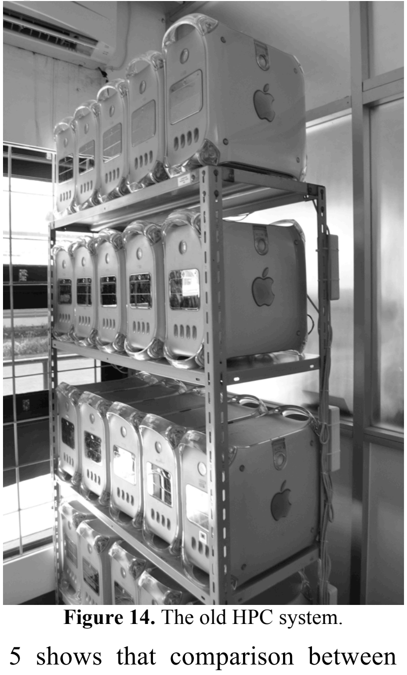

Figure 14. The old HPC system.

Table 5 shows that comparison between old HPC systems with EECC. We give values from 1 to 10 with 1 is worst and 10 is the best value. In this comparison sometimes the bigger value is not the better meaning.

In economics way, EECC is better rather than old HPC. To build HPC system with 32 nodes using PC, we need to make investment around Rp. 160.000.000,-. But when we build HPC system using EECC it will cost around half from HPC system using PCs. The HPC system using PCs need more space comparing with EECC, because EECC size is same with 2U rack server. For power consumption EECC only need 400W for 32 nodes, but for PC with 550W each it will take power consumption around 17000W to power up 32 nodes.

In performance way, EECC can give almost the same performance with the old HPC system. Significant different is in the matter of storage, EECC used 8 GB SDHC comparing with old HPC system used 500GB HDD. Although EECC using only 8GB storage, the education way to learn parallel computing is already enough. Basic installation of EECC system with parallel programming applications used about 2GB of storage.

In education process way, EECC gives better value rather than old HPC system. The reason is, EECC system can be use for different subject. For preparation, EECC and old HPC system are need the same action to turn on the node. We used the same material course and we added more information about embedded system. Student can learn how to make research, to find solution in HPC course.

V. Results and Discussion

In this paper, we presented a new parallel computing using embedded systems, called “EEPIS Embedded Computing Cluster (EECC)”, with purpose to provide a cost-effective prototyping environment for design and test of parallel computing in education program. EECC gives a better design, low cost and also low power consumption.

For future work, we want to make EECC with better packaging and more examples for HPC course. And also we want to try EECC system in robotics system.

Acknowledgement

This work is partially supported by “Penelitian Unggulan” from DIKTI 2013. The authors also want to say thank you to Mr. Nugraha Akbar and Mr. Eko Budi Utomo who helped us for designing customized power supply for this project. This research conducted at EEPIS Robotics Research Center (ER2C).

References

[1]. Sritrusta Sukaridhoto, Yoshifumi Sasaki, Koichi Ito, and Takafumi Aoki, “Development of a Compact Cluster with Embedded CPUs”, Proceeding of the Sixth Industrial Electronic Seminar 2004, pp.340-343, October 2004.

[2]. Yoshifumi Sasaki, Koichi Ito, Takafumi Aoki and Tatsuo Higuchi,” A compact cluster computer with embedded CPUs and its application to rapid prototyping of fingerprint verification system”, IEICE Electronics Express, Vol. 2, No.17, pp. 465-470, September 2005.

[3]. Yoshifumi Sasaki, Yoshiaki Abe, “Development of an Embedded CPU Board for Cluster Computing”, SICE Tohoku branch, 2007.

[4]. Elizabeth Shoop, Richard Brown, Eric Biggers, Malcolm Kane, Devry Lin, Maura Warner, “Virtual cluster for parallel and distributed education”, Proceeding of the 43rd ACM technical symposium on Computer Science Education (SIGCSE), pp. 517-522, 2012.

[5]. Nikilesh Balakrishnan, “Building and benchmarking alow power ARM cluster”, MSc Dissertation, The University of Edinburgh, 2012.

[6]. “PandaBoard”, http://www.pandaboard.org.

[7]. “Debian – The Universal Operating System”, http://www.debian.org.

[8]. “Open MPI: Open Source High Performance Computing”, http://www.open-mpi.org.

[9]. “Pallas MPI Benchmark”, http://www.ipacs-benchmark.org.

[10]. “Linpack / HPL”, http://www.ipacs-benchmark.org.