COBEM-2017-2702
FINITE ELEMENT ANALYSIS USING ARM CLUSTER

Pedro Matheus Mei Antunes
Marco Lucio Bittencourt
University of Campinas, School of Mechanical Engineering - Rua Mendeleyev, 200. Campinas/SP - Brazil
pedrommantunes@gmail.com
mlb@fem.unicamp.br

Abstract. *The concept of green computer is an important initiative to keep our environment clean and safe. Following this idea we aim to apply high performance computing and data manipulation to deal with numerical simulation of structural problems. For this purpose, we assembled an ARM cluster with 8 Odroids C2 cards equipped with ARM® Cortex®-A53(ARMv8) 1.5Ghz quad core CPUs, 2 Gbytes of DDR3 SDRAM and 32 Gbytes MicroSD card, a router and a power supply. The cluster has lower cost and energy consumption, easy portability, accessibility and cooling passive environment, differently from the conventional clusters. We present comparisons of energy and processing time for the ARM and standard clusters when running the (hp)^{2}FEM software, based on the object-oriented paradigm in C++ and high-order finite elements, for structural applications.*

Keywords: *Finite Element Method, High-performance Computing, Green Computer, ARM, Computational Simulation.*

1. INTRODUCTION

Computational simulation is a powerful tool used in many fields of research, such as chemistry, biology, meteorology, and engineering. Simulation uses a numerical model described by modifiable parameters relevant to the dynamics of the problem. Therefore, simulation allows the reduction of costs, time, and resources of the testing states in the development cycle. The finite element method (FEM) is a numerical method used to obtain an approximate mathematical model of boundary value problems (Hughes, 2000). This method is applied to problems where it is not possible to find an analytical solution due, for example, to complex geometry and non-linearities. In general, simulation of complex problems requires intensive use of computer resources. High costs, infrastructure requirements and energy consumption make more difficult the availability of high-performance clusters. However, the processing demand for many applications is not very severe and smaller cluster may be used. The initial concept comes from Beowulf clusters, which exploited mass-market PC hardware and software in conjunction with cost-effective commercial network technology (Sterling, 2001).

In the last years, ARM processors have received a great attention from the computer industry because of the low energy demand and increasing processing resources. In this work, we assembled a low cost, low energy consumption and effective processing speed cluster to simulate problems approximated by the FEM.

2. COMPUTER CLUSTERS

Computer clusters are ensembles of independently operational computers integrated by means of an interconnection network and supporting user-accessible software for organizing and controlling concurrent computing tasks that may cooperate on a common application program or work-load. There are many kinds of computer clusters, ranging from among the world's largest computers to collections of throwaway PCs. Clustering was among the first computer system architecture techniques for achieving significant improvements in overall performance, user access bandwidth, and reliability. Many (Sterling, 2001).

According to Sterling (2001), Beowulf-class systems are commodity clusters that exploit the attributes derived from mass-market manufacturing and distribution of consumer-grade digital electronic components. It is a system that usually consists on client nodes (computers not equipped with a video monitor, keyboard and mouse, with only motherboard, microprocessor, memory, hard disk and network card) and a server node (computer that works as the interface to the outside world equipped with video monitor, video card, keyboard and mouse), connected by an Ethernet network or some other type of local area network (J. Rocha 2003).

In this work, the nodes are single boards computers called Odroid-C2. A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and others features required of a functional computer.

Odroid-C2 is a powerful low-cost single computer board and extremely versatile device. It can function as a home theater set-top box, a general purpose computer for web browsing, gaming and socializing, a compact tool for college or office work, a prototyping device for hardware tinkering, a controller for home automation, a workstation for software development, and much more (HARDKERNEL, 2017).

3. DESCRIPTION OF THE ODROID CLUSTER

Our cluster, illustrated in Figure 1, was assembled with 8 Odroid-C2 cards each one with 32 Gbytes MicroSd card running operating system Ubuntu Mate 16.04 (aarch64), connected to a TP-Link TL-SG1008D switch using secure shell (SSH) protocol for remote login and a power supply (Figure 2).

In order to make feasible the transportation and guarantee the energy supply for each node, we use a standard 550W computer power source (which generates 12V, 5V, 3.3V), connected separately in each port of the HUB USB. The current in the nodes does not split, which could hinder the performance. To that purpose, a voltage reducer circuit was also developed (Figure 3) in order to turn on the 9V router to the switch power. The total of energy consumption is approximately 50 Watts.

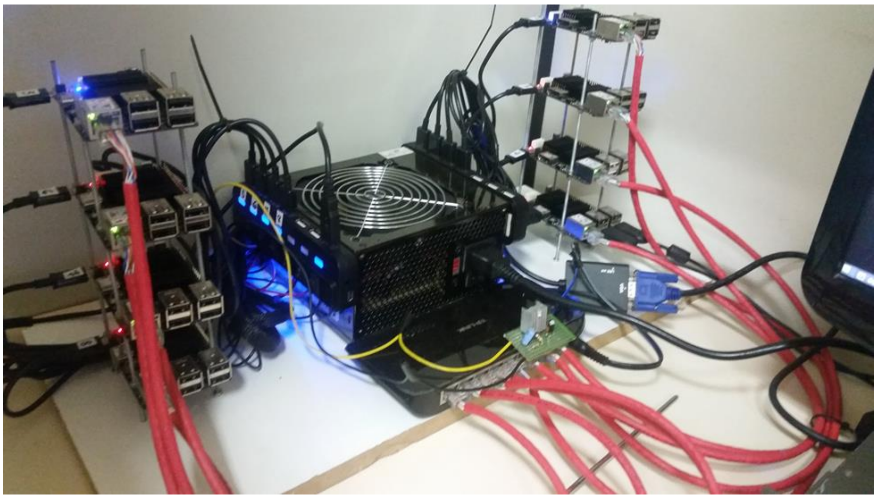

Figure 1. Complete Odroid cluster.

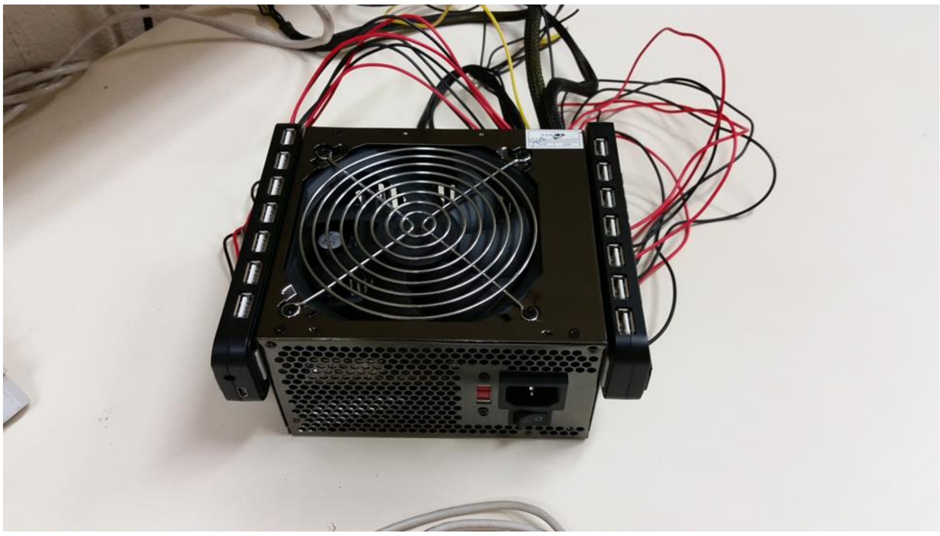

Figure 2. Power source.

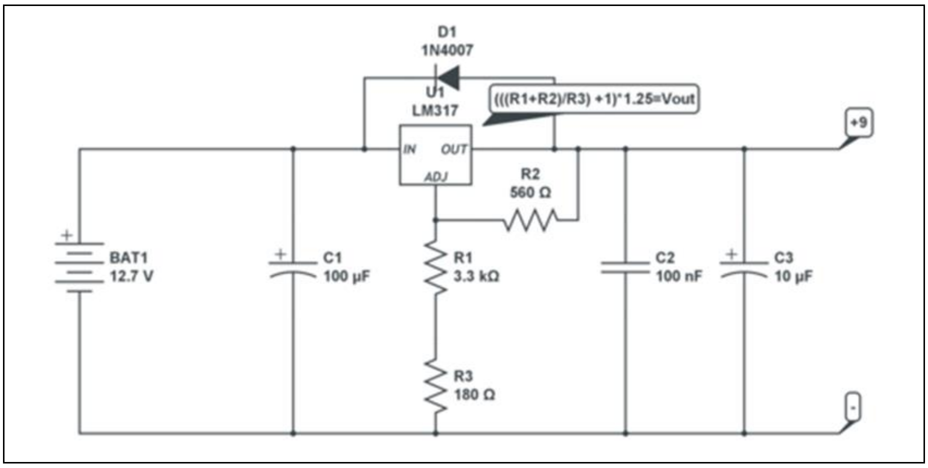

Figure 3. Voltage reducer circuit (source circuitlab.com).

4. SIMULATION SOFTWARE

We ran the (*hp*)^{2}FEM (High Performance and *hp* adaptive Finite Element Software) software framework for the high-order FEM on massively parallel computers. The (*hp*)^{2}FEM software is written in C++ and public domain. It uses local solution strategy to optimize the use of computational resources, solving the problem proposed with the same approximation of traditional approaches (G. Costa 2012).

We use two different message passing libraries: MPI (Message Passing Interface) for non-shared memory that allows the 8 nodes working together as a single core; OpenMP (Open Multi-Processing) that supports multi-platform shared memory multiprocessing programming, allowing us to split the tasks in the 4 cores of the Odroid C2 card.

5. RESULTS

We solved projection problems with meshes of hexahedral in the Odroid cluster, varying the number of elements, and polynomial order. The CPU times were compared with the previous data obtained with the IBM Blue Gene/Q, a supercomputer located at the Argonne National Laboratory (ANL) (IMB, 2017).

Table 1 shows that the BlueGene/G worked faster only in for the first mesh with 1000 elements. In general, we can say that in most of the cases the Odroid cluster using the 8 cards and MPI was able to achieve better results, being more effective. Figure 4 plots the results for the previous example and we can observe that the cluster is almost two times faster. Table 2 and Figures 5 give the results for polynomial order 6. The last case has 28,226,978 equations. Again we obtained a rate of 2 when comparing the cluster and the Blue Gene/Q. When we increased the number of equations, the performance dropped due to memory limitations.

Table 1. CPU times for the Odroid cluster and the IBM Blue Gene/Q computer with polynomial order = 1.

Number of elements	Odroid Cluster Time (s)	BlueGene/Q Time (s)
1000	0.084344	0.080721
2000	0.126164	0.16001
4000	0.177078	0.320991
8000	0.353978	0.647452
16000	0.700232	1.397667
32000	1.414806	2.962733
64000	2.902951	6.146139
128000	5.746144	12.34078

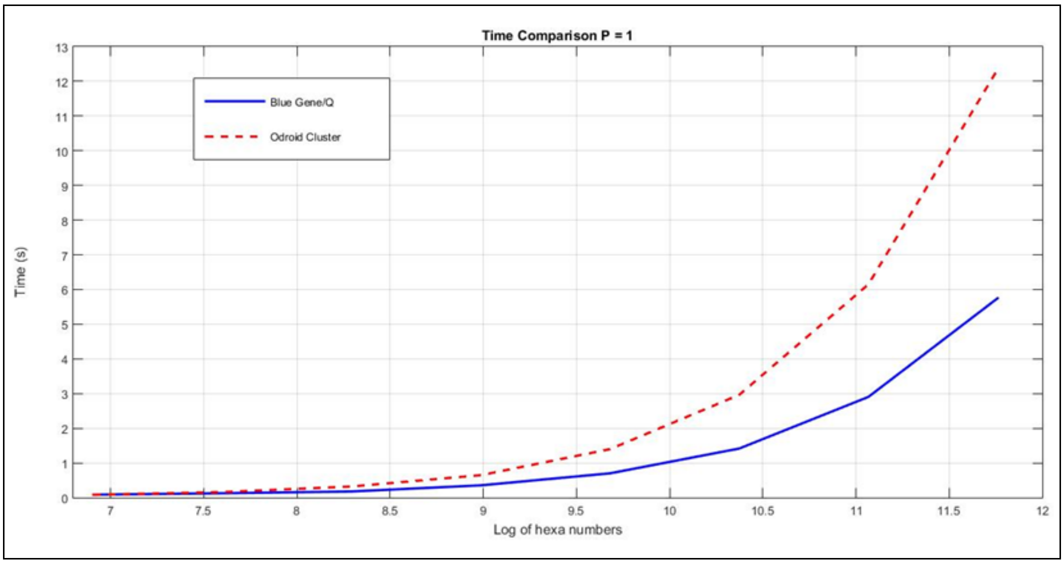

Figure 4. Comparison of CPU times for the Odroid cluster and IBM Blue Gene/Q computer for polynomial order 1.

Table 2. CPU times for the Odroid cluster and the IBM Blue Gene/Q computer with polynomial order 6.

Numbers of elements	Odroid Cluster Time (s)	Normal Cluster Time (s)
1000	3.643832	3.920967
2000	6.988388	7.833896
4000	13.761812	15.581729
8000	27.564494	30.797563
16000	53.98179	66.333333
32000	112.439062	138.328112
64000	221.498916	287.330138
128000	448.547343	577.550024

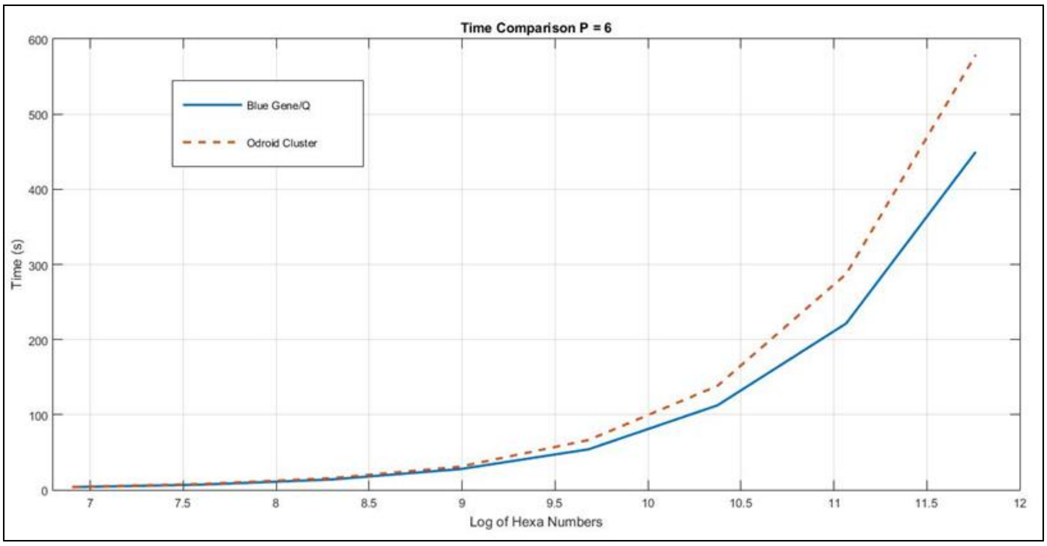

Figure 5. Comparison of CPU times for the Odroid cluster and IBM Blue Gene/Q computer for polynomial order 6.

Using the MPI and OpenMP together, we can reach the total potential of the cluster since we were able to run the (*hp*)^{2}FEM using the 4 cores of each of its 8 boards, totalizing 32 processors. According to Table 3, the speedups were 1.9 and 3.7 with two and four cores, respectively, which can be considered a very good performance.

Table 3. CPU times in seconds for the solution of the projection problem using MPI and OpenMP.

Number of elements	1 Core	2 Cores	SpeedUp 2 cores	4 Cores	SpeedUp 4 cores
1000	0.391698	0.213037	1.838638359	0.124138	3.155343247
2000	0.783379	0.402639	1.945611329	0.214006	3.660546901
4000	1.550992	0.787759	1.968866113	0.415216	3.73538592
8000	2.990833	1.583652	1.888567059	0.848361	3.525424908
16000	6.255575	3.254833	1.921934244	1.701503	3.676499542
32000	12.85274	6.580759	1.953078665	3.312205	3.88041803
64000	25.826009	13.506777	1.912077841	6.993265	3.692983034
128000	52.744692	26.513667	1.989339762	13.786691	3.825768779

6. DISCUSSION

In this work, we presented the assembling of low cost and power consumption Odroid-Cluster. We ran projection problems using the (*hp*)^{2}FEM software framework for the high-order finite element analysis. The performance obtained for the MPI and OpenMP hybrid version was very good. In addition, it was possible to solve problems with about 30 million of equations.

The Odroid-Cluster, with its low assembly costs and reduced size, facilitates transportation and makes this project an alternative idea for a variety of other studies in many areas of knowledge. Researchers may feel encouraged to assemble their own clusters capable to solve problems with very reasonable number of equations. In addition, there is a great reduction of the energy consumption for operation making them green clusters for the environment.

7. ACKNOWLEDGEMENTS

This work was supported by FAPESP and CAPES under grant numbers 2012/19922-2 and 33003017, respectively.

8. REFERENCES

da Costa, G. L. V. 2012. (*hp)*^{2}FEM : uma arquitetura de software *p* não-uniforme para o método de elementos finitos de alta ordem. Master dissertation. UNICAMP.

HARDKERNEL,2017.ODROID-C2.21set.2017<http://www.hardkernel.com/main/products/prdt_info.php?g_code=G 145457216438>.

Hughes, T.J.R, 2000. The Finite Element Method Linear Static and Dynamic Finite Element Analysis.

IMB, 2017 <https://www-03.ibm.com/systems/technicalcomputing/solutions/bluegene/>.

Rocha, K.M.G, 2003. Cluster Beowulf: aspectos de projeto e implementação”. Universidade Federal do Pará. Belem, Brazil

W. Gropp, E. Lusk, and T. Sterling, 2003. Beowulf Cluster Computing with Linux, Janusz Kowalik, MIT Press.

9. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.