

# Method for Benchmarking Single Board Computers for Building a Mini Supercomputer for Simulation of Telecommunication Systems

Gábor Lencse and Sándor Répás

**Abstract**—Parallel Discrete Event Simulation (PDES) with the conservative synchronization method can be efficiently used for the performance analysis of telecommunication systems because of their good lookahead properties. For PDES, a cost effective execution platform may be built by using single board computers (SBCs), which offer relatively high computation capacity compared to their price or power consumption and especially to the space they take up. A benchmarking method is proposed and its operation is demonstrated by benchmarking six different SBCs, namely Banana Pi, Beaglebone Black, Cubieboard2, Odroid-U3+, Radxa Rock Lite and Raspberry Pi Model B+. Their benchmarking results are compared to find out which one should be used for building a mini supercomputer for parallel discrete-event simulation of telecommunication systems. The SBCs are also used to build a heterogeneous cluster and the performance of the cluster is tested, too.

**Keywords**—benchmarking, closed queueing networks, cluster computing, discrete-event simulation, OMNeT++, single board computers

## I. INTRODUCTION

RASPBERRY Pi [1] was originally aimed of encouraging basic computer science in schools, but having shipped one million units in the first year [2], its success also encouraged several vendors to design similar single board computers with somewhat better performance characteristics for both hobbyists and commercial class applications.

Whereas a demonstration cluster made up by 64 Raspberry Pi single board computers was reported in [3], our aim is to test a number of SBCs (single board computers) from different vendors, to find out which one should be selected for building a cluster for parallel discrete-event simulation. For building such a cluster, several factors must be taken into consideration. Computing power, memory size and speed, as well as communication speed are primary factors. Heat dissipation is also important both for operation costs and especially for cooling. Size also matters, if high number of elements are built together. As for usability, the support of standard Linux distributions (e.g. Debian or Ubuntu) is essential. Last but not least, the price of the devices must also be considered.

Manuscript received February 5, 2015.

G. Lencse is with the Department of Telecommunications, Széchenyi István University, 1. Egyetem tér, H-9026 Győr, Hungary (phone: +36-30-409-56-60; fax: +36-96-613-646; e-mail: lencse@sze.hu).

S. Répás is with the Department of Telecommunications, Széchenyi István University, 1. Egyetem tér, H-9026 Győr, Hungary (e-mail: repas.sandor@sze.hu).

Though vendors publish the main parameters of their devices (e.g. CPU type and clock speed, DRAM size, technology and clock speed, NIC type, etc.) we believe that their performance concerning discrete-event simulation can be estimated the most appropriate way if we benchmark them by executing discrete-event simulation. For benchmarking, we used the OMNeT++ discrete event simulator [4] and its CQN (Closed Queueing Network) sample model. We have first used the proposed benchmarking method for estimating the computing power of the different members of a heterogeneous cluster in [5] where we also proved that PDES with the conservative synchronization method can be efficiently used in the simulation of telecommunication systems because the delay of the long distance lines ensures the good lookahead.

Even though we used the proposed method to benchmark six SBCs to find out which one would be the best choice to build a suitably large cluster for simulation, however, our main aim was to validate the proposed method. The validation of our choice between the two possible performance metrics (the sequential and the parallel performance, see their details later) was done by testing the performance of a small heterogeneous cluster of the different tested single board computers.

The remainder of this paper is organized as follows. First, we give the tested SBCs with their most important parameters. Second, we summarize the method of benchmarking with the CQN model. Third, we present the benchmarking results and discuss them. Fourth, we summarize the theoretical background of heterogeneous simulation clusters. Fifth, we present our experiments and results with the experimental heterogeneous cluster. Sixth, we present our size and power consumption measurement results and do a final comparison of the tested devices using these values, too. Finally, we give our conclusion.

## II. SELECTED SINGLE BOARD COMPUTERS FOR TESTING

Six SBCs were selected for the comparison. Raspberry Pi was a must, as it was the first popular one. Banana Pi was chosen because it has a Gigabit Ethernet NIC, which one is not yet very common for SBCs today. Odroid-U3+ was chosen because of its high clock frequency quad-core CPU. Radxa Rock Lite was selected as an alternative with quad-core CPU. Cubieboard2 contains built in storage and also SATA II interface, which can be used for connecting SSD. Table I and Table II give their most important CPU, memory and network parameters, as well as the storage and connection possibilities and what is also important, their current prices.

TABLE I  
SURVEY OF SINGLE BOARD COMPUTERS – BASIC CHARACTERISTICS

Name	Vendor URL	CPU architecture	CPU Type	Number of cores	CPU speed (MHz)
Banana Pi	http://www.lemaker.org	ARM Cortex A7	AllWinner A20	2	1000
BeagleBone Black	http://beagleboard.org	ARM Cortex A8	TI AM3359	1	1000
Cubieboard2	http://cubieboard.org	ARM Cortex A7	AllWinner A20	2	1000
ODROID-U3+	http://www.hardkernel.com	ARM Cortex A9	Samsung Exynos 4412	4	1700
Radxa Rock Lite	http://radxa.com	ARM Cortex A9	Rockchip RK3188	4	1600
Raspberry Pi Model B+	http://www.raspberrypi.org	ARM1176JZ(F)-S	Broadcom BCM2835	1	700

TABLE II  
SURVEY OF SINGLE BOARD COMPUTERS – ADDITIONAL DATA

Name	DRAM Technology	DRAM speed (MHz)	DRAM Size (MB)	NIC speed (Mbps)	Storage, Ports, etc.	Price (USD)
Banana Pi	DDR3	480/432	1024	1000	SD+SATA II, HDMI, 2xUSB 2.0	39.50
BeagleBone Black	DDR3	606	512	100	2/4GB+microSD, microHDMI, USB 2.0	55.00
Cubieboard2	DDR3	480	1024	100	4GB+microSD+SATA II, HDMI, 2xUSB 2.0	59.00
ODROID-U3+	LPDDR3	933	2048	100	microSD+eMMC, microHDMI, 3xUSB2.0	69.00
Radxa Rock Lite	DDR3	800	1024	100	microSD, HDMI, 2xUSB 2.0, WiFi	59.00
Raspberry Pi Model B+	?	500	512	100	microSD, HDMI, 4xUSB 2.0	35.00

### III. BENCHMARKING METHOD

#### A. Theoretical Background

Closed Queueing Network (CQN) was originally proposed for measuring the performance of parallel discrete-event simulation using the conservative synchronization method [6].

The OMNeT++ discrete-event simulation framework [4] contains a CQN implementation among its samples. We first used this model in our paper [7]. The below description of the model is taken from there.

This model consists of  $M$  tandem queues where each tandem queue consists of a switch and  $k$  single-server queues with exponential service times (Fig. 1). The last queues are looped back to their switches. Each switch randomly chooses the first queue of one of the tandems as destination, using uniform distribution. The queues and switches are connected with links that have nonzero propagation delays. The OMNeT++ model for CQN wraps tandems into compound modules.

To run the model in parallel, the tandems should be assigned to different segments (Fig. 2). Lookahead<sup>1</sup> is provided by delays on the marked links.

As for the parameters of the model, the preset values shipped with the model were used unless it is stated otherwise. Configuration B was chosen, the one that promised good speedup.

In our paper [7], we used this implementation for the experimental validation of the criterion defined for good speedup in [8]. This criterion gives a simple and straight forward method for the estimation of the available parallelism on the basis of values which can be easily measured in

sequential execution of the simulation. Ref [8] uses the notations  $ev$  for the number of events,  $sec$  for real world time (also called execution time or wall-clock time) in seconds and  $simsec$  for simulated time (model time) in seconds.

The paper uses the following quantities for assessing the available parallelism:

- $P$  performance represents the number of events processed per second ( $ev/sec$ ).
- $E$  event density is the number of events that occur per simulated second ( $ev/simsec$ ).
- $L$  lookahead is measured in simulated seconds ( $simsec$ ).
- $\tau$  latency (sec) is the latency of sending a message from one segment to another.

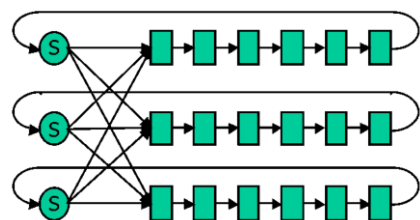


Fig. 1.  $M=3$  tandem queues with  $k=6$  single server queues in each tandem queue [7].

<sup>1</sup> Lookahead is an important parameter of the conservative discrete-event simulation: it expresses a time interval while the given segment will surely not receive a message from another segment.

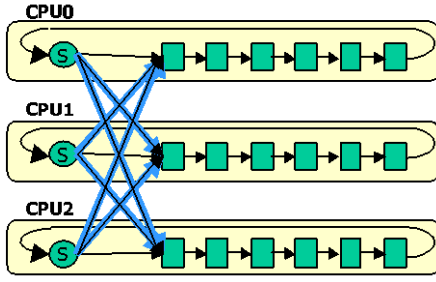


Fig. 2. Partitioning the CQN model [7].

- $\lambda$  coupling factor can be calculated as the ratio of  $LE$  and  $\tau P$ :

$$\lambda = \frac{L \cdot E}{\tau \cdot P} \quad (1)$$

We have shown in [7] that if  $\lambda$  is in the order of several hundreds or higher then we may expect a good speedup. It may be nearly linear even for higher number of segments ( $N$ ) if  $\lambda_N$  is also at least in the order of several hundreds, where:

$$\lambda_N = \frac{\lambda}{N} \quad (2)$$

### B. Parameters of Benchmarking

We benchmarked all the single board computers by executing the CQN model sequentially (thus using only one core even if multiple cores were available) with the following parameters:  $M=24$  tandem queues,  $k=50$  single server queues, with exponential service time (having expected value of 10s),  $T=10000$  simsec,  $L=100$  simsec delay on the lines between the tandem queues.

We measured the execution time and calculated the average performance ( $P$ ) as the ratio of the number of all the executed events ( $N_E$ ) and the execution time of the sequential simulation ( $T_1$ ):

$$P = \frac{N_E}{T_1} \quad (3)$$

The used Linux kernel versions and distributions are listed in Table III. OMNeT++ 4.6 and OpenMPI 1.8.4 were used.

## IV. BENCHMARKING RESULTS

### A. Single core results

First, we measured the performance of a single core only. The performance results are shown in Table IV. Odroid U3+ (65839) is the winner, and Radxa Rock Lite (54692) is the second one. Cubieboard2 (33494) is the third one but Banana Pi (33432) is very close to it. Raspberry Pi B (8830) is lagging behind all the others.

### B. Multi core results

Second, we also tested the performance of the four multi-core ones using all their available cores. The CQN model was compiled with the MPI support and the simulation model was shared into the same number of partitions as the number of CPU cores of the given single board computers had, that is two or four. Table V shows the results. We also included the speedup and the relative speedup values. According to its conventional definition, the speedup ( $s_N$ ) of parallel execution

is the ratio of the speed of the parallel execution by  $N$  CPU cores and the sequential execution by 1 CPU core which is usually calculated as the ratio of the execution time of the sequential execution ( $T_1$ ) and that of the parallel execution ( $T_N$ ), however now we used the ratio of the multi core performance ( $P_N$ ) and the single core performance ( $P_1$ ):

$$s_N = \frac{T_1}{T_N} = \frac{P_N}{P_1} \quad (4)$$

The relative speedup ( $r_N$ ) can be calculated as the ratio of the speedup and the number of the CPU cores that produced the given speedup:

$$r_N = \frac{s_N}{N} \quad (5)$$

The relative speedup measures the efficiency of parallel execution. A relative speedup value of 1 means that the speedup is linear that is the computing power of the  $N$  CPU cores can be fully utilized.

Except for Radxa Rock Lite, all the other three ones show super-linear speedup, that is the relative speedup is higher than 1. This phenomenon is usually caused by caching. (E.g. the cores have they own L1 cache and partitions better fit in them than the whole model fitted into just one of them. Similar phenomenon was reported in [9], see page 95.) Now, we do not go deeper, but we plan to do further analysis of this phenomenon.

As for the ranking of the different single board computers, there is only a little change in the order: Banana Pi (81160) is now at the 3rd place as it has overtaken Cubieboard2 (76071) but the difference is not very significant. What is much more significant, Odroid U3+ (279955) at the first place now seriously outperformed Radxa Rock Lite (142369) at the second place. Therefore Odroid U3+ proved to be far the best performing one from among the six tested single board computers.

We believe that the results of the multi core benchmark using all the cores are to be used for characterizing the performance of the SBCs for parallel simulation because we would like to use their all cores in the simulation. We will support this in a case study with heterogeneous clusters.

TABLE III  
LINUX KERNEL VERSIONS AND DISTRIBUTIONS

Name	Kernel version	Distribution
Banana Pi	3.4.104+ armv7l	Debian 7.8
BeagleBone Black	3.8.13-bone50 armv7l	Debian 7.8
Cubieboard2	3.4.43+ armv7l	Linaro 13.04
Odroid-U3+	3.8.13.16 armv7l	Ubuntu 13.10
Radxa Rock Lite	3.0.36+ armv7l	Linaro 14.04
Raspberry Pi B+	3.12.35+ armv6l	Raspbian (Deb. 7.6)

TABLE IV  
SINGLE-CORE PERFORMANCE

Name	Execution Time (s)		P (ev/sec)
	average	std. dev.	
Banana Pi	46.9	0.92	33432
BeagleBone Black	68.3	1.51	22952
Cubieboard2	46.8	0.64	33494
Odroid-U3+	23.8	0.11	<b>65839</b>
Radxa Rock Lite	28.6	0.26	54692
Raspberry Pi B+	177.4	1.46	8830

TABLE V  
ALL-CORE PERFORMANCE AND COMPARISON

Name	No. of Cores	P <sub>1</sub> (ev/sec)	P <sub>N</sub> (ev/sec)	Speedup	Relative Speedup
Banana Pi	2	33432	81160	2.43	1.21
Cubieboard2	2	33494	76071	2.27	1.14
Odroid-U3+	4	65839	<b>279955</b>	4.25	1.06
Radxa Rock Lite	4	54692	142369	2.60	0.65

## V. THEORETICAL BACKGROUND FOR HETEROGENEOUS CLUSTERS

### A. Load Balancing Criterion

We discussed the conditions necessary for a good speedup of the parallel simulation using the conservative synchronization method in heterogeneous execution environment in [5]. There we defined the logical topology of heterogeneous clusters as a *star shaped network of homogeneous clusters* where a homogeneous cluster may be built up by one or more instances of single-core or multi-core computers. In addition to the before mentioned coupling factor criterion that  $\lambda_N$  should be in the order of several hundreds, we defined another very natural criterion of load balancing that “all the CPUs (or CPU cores) should get a *fair share* from the execution of the simulation. A fair share is proportional to the computing power of the CPU *concerning the execution of the given simulation model.*”

Now, we have already benchmarked the CPUs by the CQN model.

### B. Measuring the Efficiency of Parallel Simulation Executed by Heterogeneous Systems

We extended the definition of the relative speedup of parallel program execution (not only simulation) for heterogeneous execution environments in [10]. There we applied it for measuring the efficiency of *heterogeneous simulation* (that is parallel simulation executed by heterogeneous systems) and received the following formula:

$$r_h = \frac{N_E}{T_h \cdot P_c} \quad (6)$$

where the letters denote the following values:

- $r_h$  – the relative speedup of the heterogeneous simulation compared to the sequential simulation
- $N_E$  – the number of events in the sequential simulation
- $T_h$  – the execution time of the heterogeneous simulation
- $P_c$  – the cumulative sum of the performance of all the cores in the heterogeneous execution environment,

which can be calculated as:

$$P_c = \sum_{j=1}^{N_{CT}} P_i \cdot N_i \quad (7)$$

where the letters denote the following values:

- $N_{CT}$  – the number of the CPU core types
- $P_i$  – the performance of a single core of type  $i$
- $N_i$  – the number of cores of type  $i$

Similarly to the homogeneous case, the maximum (and the desired ideal) value of the relative speedup equals to 1.

## VI. PERFORMANCE OF OUR HETEROGENEOUS CLUSTER

The six single board computers were interconnected by a TP-Link 26-port Gigabit Ethernet switch (TL-SG5426).

### A. Partitioning of the CQN model

The performance proportional partitioning of the CQN model was done using the following formula:

$$n_i = N_T \frac{P_i}{P_c} \quad (8)$$

where the letters denote the following values:

- $n_i$  – the number of tandems to put into a segment executed by a core of type  $i$
- $N_T$  – the number of tandems in the CQN model
- $P_i$  – the performance of a single core of type  $i$
- $P_c$  – see (7)

The number of the tandem queues was increased to 96 to be large enough for an approximate performance proportional partitioning. Whereas (8) defines the theoretically optimal values, the number of the tandems must be integers, therefore we rounded them. Two different partitioning were made. For the first one, the P values from the single core measurements were used, see Table III. For the second one, the same values were kept for the single core SBCs, but the *P<sub>ICE one core equivalent parallel performance</sub>* from the all core measurements was calculated according to (9) taking the  $P_N$  and  $N$  values from Table IV.

$$P_{ICE} = \frac{P_N}{N} \quad (9)$$

The division of the 96 tandem queues among the cores of the single board computers using the first and the second method are shown in Table VI and Table VII, respectively. Note that the usage of the mathematical rounding would have resulted in 97 tandem queues in Table VII therefore the number of tandem queues to be put into the segment executed by the BeagleBone Black SBC was rounded from 3.6 to 3 and not to 4.

A 10000 simsec long simulation was executed by the heterogeneous cluster 11 times and the execution time was measured for both partitioning. The relative speedup was also calculated according to (6), where the number of events in the sequential simulation was  $N_E=6260606$  and  $P_c$  was calculated according to (7) taking the  $P_i$  values from Table VI and the  $P_{ICE,i}$  values from Table VII for the first partitioning and for the second partitioning, respectively.

TABLE VI  
THE DIVISION OF THE 96 TANDEM QUEUES AMONG THE SBCs USING THE  
SINGLE CORE BENCHMARK RESULTS

SBC type	$P_i$	$N_i$	$n_i$	tandems /core	cumulated tandems
Banana Pi	33432	2	4.95	5	10
BeagleBone Black	22952	1	3.40	3	3
Cubieboard2	33494	2	4.96	5	10
Odroid-U3+	65839	4	9.76	10	40
Radxa Rock Lite	54692	4	8.11	8	32
Raspberry Pi B	8830	1	1.31	1	1
Total number of the cores:	14	Total no. of the tandems: 96			

TABLE VII  
THE DIVISION OF THE 96 TANDEM QUEUES AMONG THE SBCs USING THE  
ALL CORES BENCHMARK RESULTS

SBC type	$P_{ICE,i}$	$N_i$	$n_i$	tandems /core	cumulated tandems
Banana Pi	40580	2	6.37	6	12
BeagleBone Black	22952	1	3.60	3	3
Cubieboard2	38036	2	5.97	6	12
Odroid-U3+	69989	4	10.99	11	44
Radxa Rock Lite	35592	4	5.59	6	24
Raspberry Pi B	8830	1	1.39	1	1
Total number of the cores:	14	Total no. of the tandems: 96			

### B. Results

Table VIII shows the results. Both the average execution time and the relative speedup values are significantly better for the second method. Though someone might challenge the relative speedup values stating that they were calculated using smaller  $P_c$  values in the denominator of (6), the average execution time values are unquestionably show the superiority of the second method for partitioning.

Therefore, our results justified that if there is a significant difference between the single core benchmark values and the one core equivalent parallel performance benchmark values then the latter ones are better anticipate the performance of the cores in a parallel simulation thus the latter ones are to be considered as the valid metrics.

## VII. FINAL COMPARISON OF THE TESTED SBCs

### A. Absolute Performance Comparison

For the comparison of the absolute performance of the six SBC, we use their  $P_N$  all-core performance values. They are compared by using a bar chart in Fig. 3.

TABLE VIII  
EXECUTION TIME AND RELATIVE SPEEDUP AS A FUNCTION BENCHMARKING  
METHOD

Benchmarking Method	$P_c$ (ev/simsec)	Execution time (s)		relative speedup
		average	std. dev.	
Single core	647748	24.3	1.26	0.398
All cores	611337	18.7	0.66	<b>0.548</b>

### B. Size and Power Consumption

We measured the size of the SBCs together with their connectors, thus our results are somewhat higher than those provided by the manufacturers. We measured their power consumption under different load conditions: the system was idle, one core had full load, all the cores had full load. The above detailed CQN model was used for load generation. Our results can be found in Table IX.

### C. Relative Performance Characteristics

We used the all core parallel performance values of the SBCs. (One may also calculate with the single core results, as we provided the necessary data for that, too.) Our results can be found in Table X. Their space, price and power consumption relative performance values are compared in Fig. 4, Fig. 5 and Fig 6, respectively. In all the relative performance metrics, Odroid-U3+ became the absolute winner and Radxa Rock Lite is the second best one. As for performance per occupied space, Odroid-U3+ 3.1 times outperformed Radxa Rock Lite. As for price and power consumption relative performance, this proportion is only 1.7 and 1.4, respectively. Banana Pi received the third place both in the performance per price and in the performance per power consumption race. This is the only card with Gigabit NIC among the tested ones but it could not gain much advantage from it, because our benchmarking method did not test that. It could be better ranked in other test with high data transfer rates.

TABLE IX  
DIMENSIONS AND POWER CONSUMPTION OF THE SINGLE BOARD COMPUTERS

Name	Dimensions			V (cm <sup>3</sup> )	CPU is Idle			1 Core is Used			All the Cores are Used		
	(mm)	(mm)	(mm)		U (V)	I (mA)	P(W)	U (V)	I (mA)	P(W)	U (V)	I (mA)	P(W)
Banana Pi	75	96	18	130	5.54	310	1.72	5.50	390	2.15	5.47	490	2.68
BeagleBone Black	85	52	16	71	5.02	250	1.26	4.96	370	1.84			
Cubieboard2	102	58	20	118	5.57	230	1.28	5.53	345	1.91	5.49	470	2.58
ODROID-U3+	48	81	17	66	5.55	350	1.94	5.51	410	2.26	5.33	1000	5.33
Radxa Rock Lite	80	100	13	104	5.50	550	3.03	5.50	580	3.19	5.41	700	3.79
Raspberry Pi Model B+	60	90	13	70	5.52	380	2.10	5.51	405	2.23			

TABLE X  
RELATIVE ALL-CORE PERFORMANCE CHARACTERISTICS

Name	$P_N / V$ (ev/sec/cm <sup>3</sup> )	$P_N / \text{Price}$ (ev/sec/USD)	$P_N / \text{Power Cons.}$ (ev/sec/W)
Banana Pi	624	2055	30284
BeagleBone Black	323	417	12474
Cubieboard2	645	1289	29485
Odroid-U3+	<b>4242</b>	<b>4057</b>	<b>52524</b>
Radxa Rock Lite	1369	2413	37564
Raspberry Pi B+	126	260	3960

Power Consumption Relative Performance  
(ev/sec/W)

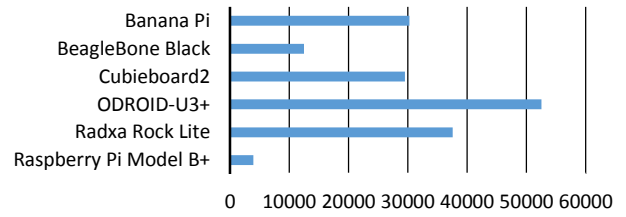


Fig. 6. Comparison of the power consumption relative all-core performance of the SBCs.

Performance (ev/sec)

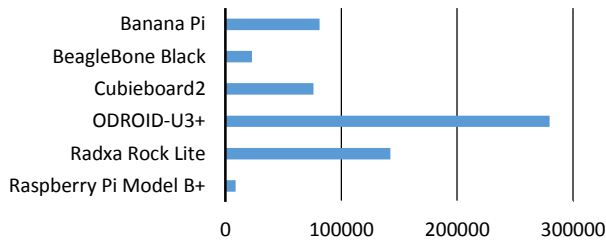


Fig. 3. Comparison of the all-core performance of the SBCs.

#### D. Discussion and Future Plans

Many more SBCs exist. We consider that our most important result is the testing method itself and not the ranking of the six tested SBCs. We have already collected the parameters of many more other SBCs and plan to select another set of them for benchmarking. We plan to select an SBC for building a homogeneous cluster of 128 elements.

Space Relative Performance (ev/sec/cm<sup>3</sup>)

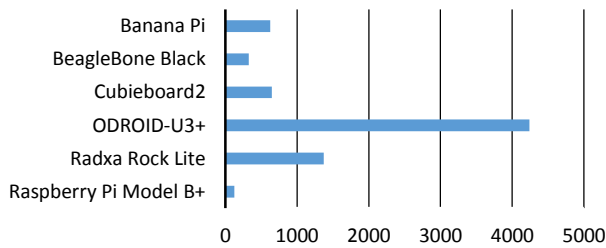


Fig. 4. Comparison of the space relative all-core performance of the SBCs.

Price Relative Performance (ev/sec/USD)

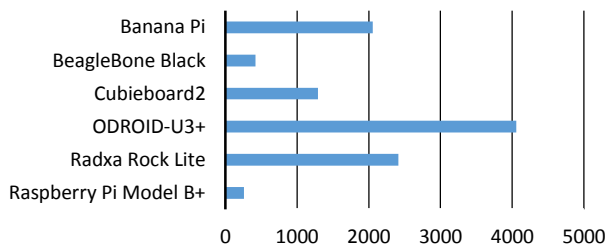


Fig. 5. Comparison of the price relative all-core performance of the SBCs.

## VIII. CONCLUSION

A method with two variants (single core and all cores test) was described for benchmarking different computers for parallel simulation. It was shown that the values of the all cores method characterize better the parallel simulation capabilities of the computers. Six single board computers (SBCs) were benchmarked. Their space, price and power consumption relative performance were also calculated. Odroid-U3+ was the absolute winner and Radxa Rock Lite took the second place both in the absolute and in the relative performance race.

## ACKNOWLEDGMENT

The authors express their thanks to Szilárd Lovas for lending a Raspberry Pi SBC and also for the practical help provided by him as well as by Sándor Major and Tamás Sass.

## REFERENCES

- [1] E. Upton and G. Halfacree, *Raspberry Pi User Guide*, 2nd ed., Wiley, 2013.
- [2] C. Edwards, "Not-so-humble Raspberry Pi gets big ideas", *Engineering & Technology*, vol. 8, no. 3, April 2013, pp. 30–33 DOI: 10.1049/et.2013.0301
- [3] S. J. Cox, J. T. Cox, R. P. Boardman, S. J. Johnston, M. Scott, N. S. O'Brien, "Iridis-pi: a low-cost, compact demonstration cluster", *Cluster Computing*, vol. 17, no. 2, June 2014, pp. 349-358. DOI: 10.1007/s10586-013-0282-7
- [4] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment", in *Proc. 1st Intern. Conf. on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, (Marseille, France, March 3-7, 2008) pp. 1-10.
- [5] G. Lencse, I. Derka and L. Muka, "Towards the efficient simulation of telecommunication systems in heterogeneous distributed execution environments", in *Proc. Int. Conf. on Telecommunications and Signal Processing (TSP 2013)*, (Rome, Italy, 2013. July, 2-4.) Brno University of Technology, pp. 314-310. DOI: 10.1109/TSP.2013.6613941
- [6] R. L. Bagrodia and M. Takai, "Performance evaluation of conservative algorithms in parallel simulation languages", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11. No 4. April 200, pp. 395–411. DOI: 10.1109/71.850835
- [7] G. Lencse and A. Varga, "Performance Prediction of Conservative Parallel Discrete Event Simulation", in *Proc. 2010 Industrial Simulation Conf. (ISC'2010)* (Budapest, Hungary, 2010. June 7-9.) EUROSIS-ETI, pp. 214-219.
- [8] A. Varga, Y. A. Sekercioglu and G. K. Egan. "A practical efficiency criterion for the null message algorithm", in *Proc. European Simulation Symposium (ESS 2003)*, (Oct. 26-29, 2003, Delft, The Netherlands) SCS International, pp. 81-92.
- [9] John Benzi, M. Damodaran, "Parallel Three Dimensional Direct Simulation Monte Carlo for Simulating Micro Flows", in *Parallel Computational Fluid Dynamics 2007*, Springer Lecture Notes in Computational Science and Engineering, Vol. 67, pp 91-98. DOI: 10.1007/978-3-540-92744-0\_11
- [10] G. Lencse and I. Derka, "Testing the Speedup of Parallel Discrete Event Simulation in Heterogeneous Execution Environments" in *Proc. ISC'2013, 11th Annu. Industrial Simulation Conf.*, (Ghent, Belgium, 2013. May 22-24.) EUROSIS-ETI, pp. 101-107.