

# Benchmarking Further 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 ten different SBCs, namely Banana Pi, Beaglebone Black, Cubieboard2, Odroid-C1+, Odroid-U3+, Odroid-XU3 Lite, Orange Pi Plus, Radxa Rock Lite, Raspberry Pi Model B+, and Raspberry Pi 2 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 both for hobbyists and for 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.

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.

This paper is an extended version of our conference paper [6], where 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. Now, we extend our previous results with the testing of four further SBCs. We also disclose our plans for future research concerning SBCs.

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. Seventh, we disclose our plans for future research concerning SBCs. Finally, we give our conclusion.

## II. SELECTED SINGLE BOARD COMPUTERS FOR TESTING

For our conference paper [6], six SBCs were selected for 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. And finally, Beaglebone Black was an alternative single-core SBC.

We have selected four further SBCs for the journal version of our paper. As Odroid-U3+ was the absolute

Manuscript received November 15, 2015, revised January 27, 2016.

G. Lencse is with the Department of Telecommunications, Széchenyi István University, 1. Egyetem tér, H-9026 Győr, Hungary (phone: +36-20-775-82-67; 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).

doi: 10.11601/ijates.v5i1.138

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-C1+	http://www.hardkernel.com	ARM Cortex A5	Amlogic S805	4	1500
Odroid-U3+	http://www.hardkernel.com	ARM Cortex A9	Samsung Exynos 4412	4	1700
Odroid-XU3 Lite	http://www.hardkernel.com	ARM Cortex A15+A7	Samsung Exynos 5422	4+4	1800+1300
Orange Pi Plus	http://www.orangepi.org	ARM Cortex A7	AllWinner H3	4	1600
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
Raspberry Pi 2 Model B+	http://www.raspberrypi.org	ARM Cortex A7	Broadcom BCM2836	4	900

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	34.00
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-C1+	DDR3	792	1024	1000	microSD+eMMC, HDMI, 4xUSB 2.0	37.00
Odroid-U3+	LPDDR3	933	2048	100	microSD+eMMC, microHDMI, 3xUSB 2.0	69.00
Odroid-XU3 Lite	LPDDR3	933	2048	100	microSD+eMMC, microHDMI, 4x USB 2.0, USB 3.0	99.00
Orange Pi Plus	DDR3	480	1024	1000	8GB+microSD+SATA II, HDMI, 4xUSB 2.0, WiFi	47.50
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	32.19
Raspberry Pi 2 Model B+	LPDDR2	400	1024	100	microSD, HDMI, 4xUSB 2.0	38.71

winner from among the six SBCs, we included two other kind of Odroid SBCs: Odroid-XU3 Lite for its expected higher performance and Odroid-C1+ for its being more cost effective and also having a Gigabit Ethernet NIC. As the old Raspberry Pi showed the poorest performance among the six SBCs, now we included its improved version, Raspberry Pi 2 Model B+ to give the brand a second chance. We also found Orange Pi Plus interesting because of having 8GB built-in storage and SATAII interface. 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 as of November 13, 2015. Please note that Odroid U3+ and Odroid XU3 Lite are discontinued. The latter one was replaced by Odroid XU4 using the same Samsung Exynos 5422 CPU at 2GHz and having a Gigabit Ethernet port.

### 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 [7].

The OMNeT++ discrete-event simulation framework [4] contains a CQN implementation among its samples. We first used this model in our paper [8]. The below description of the model is taken from there.

This model consists of  $N_T$  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 [8], we used this implementation for the experimental validation of the criterion defined for good speedup in [9]. 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 [9] 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 ( $ev/sec$ ).

<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.

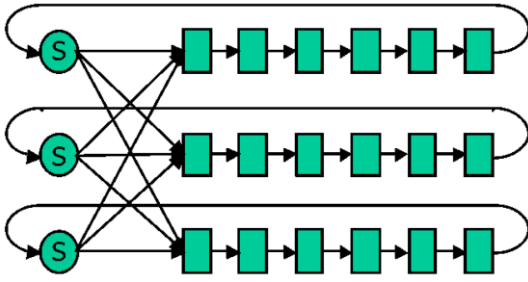


Fig. 1.  $N_T=3$  tandem queues with  $k=6$  single server queues in each tandem queue [8].

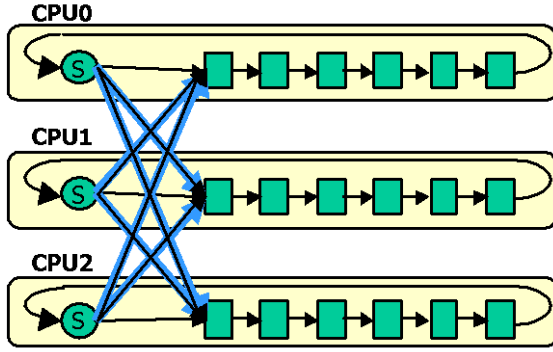


Fig. 2. Partitioning the CQN model [8].

- $E$  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.
- $\lambda$  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 [8] 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:  $N_T=24$  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 ( $T_l$ ):

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

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

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-C1+	3.10.67-55 armv7l	Ubuntu 14.04.3
Odroid-U3+	3.8.13.16 armv7l	Ubuntu 13.10
Odroid-XU3 Lite	3.10.69 armv7l	Ubuntu 14.04.2
Orange Pi Plus	3.4.39 armv7l	Debian 8.0
Radxa Rock Lite	3.0.36+ armv7l	Linaro 14.04
Raspberry Pi B+	3.12.35+ armv6l	Raspbian (Deb. 7.6)
Raspberry Pi 2 B+	4.1.7-v7+ armv7l	Raspbian (Deb. 7.8)

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-C1+	33.8	0.86	46370
Odroid-U3+	23.8	0.11	<b>65839</b>
Odroid-XU3 Lite	17.2	0.71	<b>91281</b>
Orange Pi Plus	36.6	0.78	42804
Radxa Rock Lite	28.6	0.26	54692
Raspberry Pi B+	177.4	1.46	8830
Raspberry Pi 2 B+	46.2	0.96	33946

## 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-XU3 Lite shows the best performance by processing 91281 events per second. Odroid-U3+, which one was winner in [6], is now the second best one (65839 ev/sec) whereas Radxa Rock Lite takes the third place (54692 ev/sec). Yet Odroid-C1+ (46370 ev/sec) and Orange Pi Plus (42804 ev/sec) excel somewhat from the rest of the SBCs. The performance of Banana Pi (33432 ev/sec) Cubieboard2 (33494 ev/sec) and Raspberry Pi 2 B+ (33946 ev/sec) are very close to each other. BeagleBone Black (22952 ev/sec) performed significantly worse and the old Raspberry Pi B+ (8830 ev/sec) is lagging behind all the others.

### B. Multi core results

Second, we also tested the performance of the eight multi-core SBCs 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<sup>2</sup>. 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 in  $N$  segments (by the same number of CPU cores)

<sup>2</sup> Please note that Odroid-XU3 Lite has four A15 and four A7 cores. We used only four partitions therefore the four "smaller" cores were not utilized.

and the sequential execution by 1 CPU core which is usually calculated as the ratio of the execution time of the sequential execution ( $T_I$ ) 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 ( $P_I$ ):

$$s_N = \frac{T_I}{T_N} = \frac{P_N}{P_I} \quad (4)$$

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.

Four of the SBCs 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 [10], 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 a significant change in the order (see Table V): Odroid-U3+ (279955 ev/sec) now outperformed Odroid-XU3 Lite (247956 ev/sec) and Raspberry Pi 2 B+ (167684 ev/sec) is now the third one.

As for the relative speed up, Raspberry Pi 2 B+ (1.23) is the best and Banana Pi (1.21) is very close to it. Regarding the low end, Orange Pi Plus (0.48) is the worst, and the "Light" versions of cards are also poor: Radxa Rock Lite (0.65) and Ordoid-XU3 Lite (0.68).

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. Please note that the case study was prepared for the conference version of the paper and therefore it does not include the further SBCs which are included in the journal version only.

## 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.*"

TABLE V  
ALL-CORE PERFORMANCE AND COMPARISON

Name	No. of Cores	$P_I$ (ev/sec)	$P_N$ (ev/sec)	Speedup	Relative Speedup
Banana Pi	2	33432	81160	2.43	1.21
BeagleBone Black	1	22952	22952	--	--
Cubieboard2	2	33494	76071	2.27	1.14
Odroid-C1+	4	46370	148967	3.21	0.80
Odroid-U3+	4	65839	<b>279955</b>	4.25	1.06
Odroid-XU3 Lite	4	91281	<b>247956</b>	2.72	0.68
Orange Pi Plus	4	42804	82452	1.93	0.48
Radxa Rock Lite	4	54692	142369	2.60	0.65
Raspberry Pi B+	1	8830	8830	--	--
Raspberry Pi 2 B+	4	33946	167684	4.94	1.23

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 [11]. 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)

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

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>

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 partitionings were made. For the first one, the P values from the single core measurements were used, see Table IV. For the second one, the same values were kept for the single core SBCs, but the  $P_{ICE}$  one core equivalent parallel performance from the all core measurements was calculated according to (9) taking the  $P_N$  and  $N$  values from Table V.

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

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 partitionings. 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.

### 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 ten SBC, we use their  $P_N$  all-core performance values. They are compared by using a bar chart in Fig. 3. (It is put on the same page with the relative performance comparison figures for the synoptic view and easy comparison.)

### B. Size and Power Consumption

We measured the size of the SBCs together with their overhanging parts (e.g. connectors, buttons, microSD cards), thus our results in Table IX are somewhat higher than those

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	96	..75	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-C1+	..87	..55	21	100	5.14	330	1.70	5.14	390	2.00	5.12	490	2.51
Odroid-U3+	81	..48	17	66	5.55	350	1.94	5.51	410	2.26	5.33	1000	5.33
Odroid-XU3 Lite	99	..78	22	170	5.19	560	2.91	5.11	970	4.96	5.11	1510	7.72
Orange Pi Plus	115	63	19	138	5.05	540	2.73	5.02	640	3.21	5.00	690	3.45
Radxa Rock Lite	100	..90	14	126	5.50	550	3.03	5.50	580	3.19	5.41	700	3.79
Raspberry Pi B+	90	..60	19	103	5.52	380	2.10	5.51	405	2.23	--	--	--
Raspberry Pi 2 B+	90	..60	19	103	5.16	230	1.19	5.14	300	1.54	5.11	450	2.30

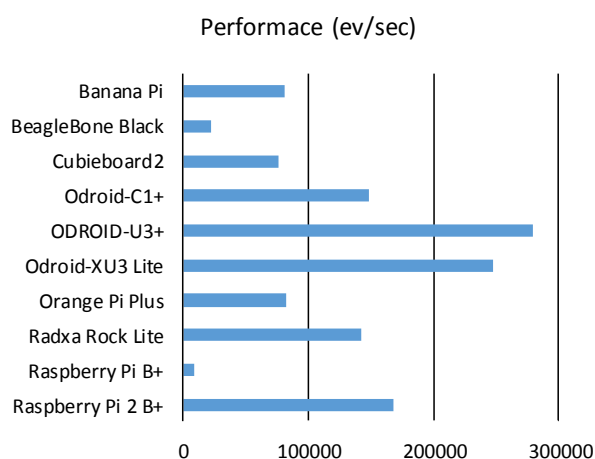


Fig. 3. Comparison of the all-core performance of the SBCs.

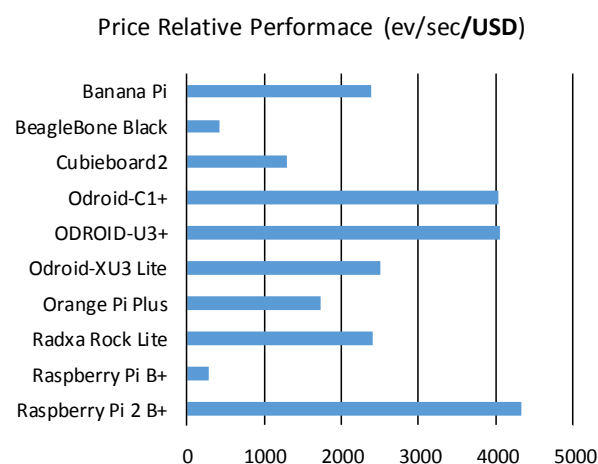


Fig. 5. Comparison of the price relative all-core performance of the SBCs.

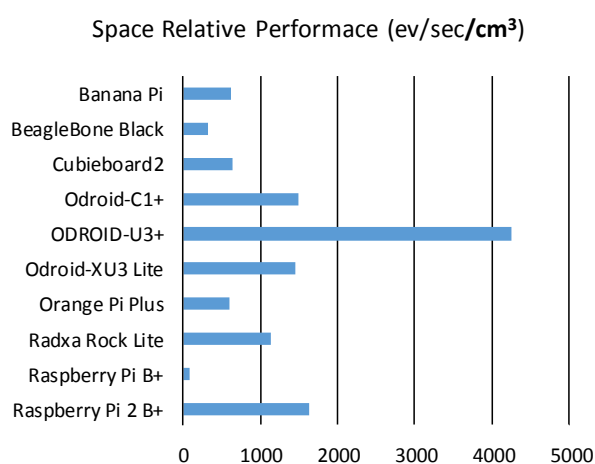


Fig. 4. Comparison of the space relative all-core performance of the SBCs.

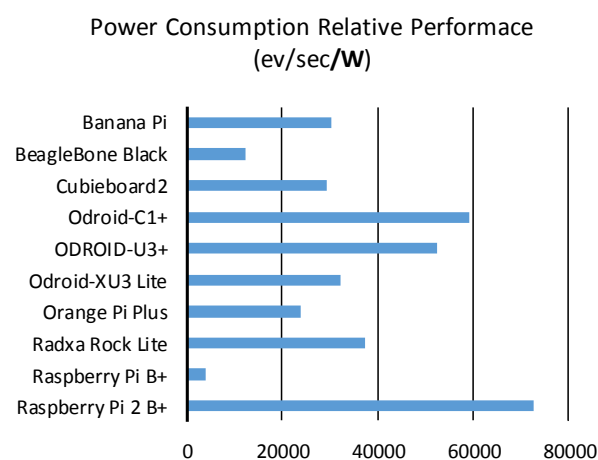


Fig. 6. Comparison of the power consumption relative all-core performance of the SBCs.

provided by the manufacturers. Please note that if SBCs are integrated into a cluster then they will occupy even larger space because both cables (Ethernet, power) and cooling need significant space.

We measured the power consumption of the SBCs under different load conditions: the system was idle, one core had full load, all cores had full load. The above detailed CQN model was used for load generation. Our results can also be found in Table IX.

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	2387	30284
BeagleBone Black	323	417	12474
Cubieboard2	645	1289	29485
Odroid-C1+	1490	4026	59349
Odroid-U3+	<b>4242</b>	4057	52524
Odroid-XU3 Light	1459	2505	32119
Orange Pi Plus	597	1736	23899
Radxa Rock Lite	1130	2413	37564
Raspberry Pi B+	86	274	3960
Raspberry Pi 2 B+	1628	<b>4332</b>	<b>72906</b>

### 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.

#### 1) Space relative performance

Concerning space relative performance, Odroid-U3+ (4242 ev/sec/cm<sup>3</sup>) seriously outperformed all other SBCs. The second one, Raspberry Pi 2 B+ (1628 ev/sec/cm<sup>3</sup>) could not reach even half of the performance of Odroid-U3+.

#### 2) Price relative performance

Raspberry Pi 2 B+ (4332 ev/sec/USD) showed the best price relative performance, but Odroid-U3+ (4057 ev/sec/USD) and Odroid-C1+ (4026 ev/sec/USD) were close to it.

#### 3) Power consumption relative performance

Raspberry Pi 2 B+ (72906 ev/sec/cm<sup>3</sup>/W) showed the best price relative performance being significantly better than Odroid-C1+ (59349 ev/sec/W) and Odroid-U3+ (52524 ev/sec/USD).

### D. Discussion of the results

The usage of different metrics resulted in different ranking

order of the tested ten SBCs. We consider that our most important result is the testing method itself and not the ranking of the ten tested SBCs. Using our testing method, one can test other SBCs and select from among them on the basis of his/her own target function.

Three of the tested SBCs have Gigabit Ethernet NICs but they could not gain advantage from it, because our benchmarking method did not test that. When the SBCs are actually used for parallel simulation then the communication speed matters. These three cards could be better ranked in a more realistic test setup when a small cluster is built for testing purposes. For more details, see our future plans for further research.

## VIII. FUTURE PLANS

### A. Building a Cluster for Simulation

We plan to build a cluster of significant size (at least of 128 elements or even more), but before determining the type of the SBC, we plan to experiment with smaller clusters of 16 elements for a more realistic benchmarking of the SBCs. These small clusters makes it possible to take also the speed of communication into consideration. In addition to that we may gain more experience on the architecture of the cluster, too. For example, if we will connect high number of elements then multiple switches will have to be used. How it will influence the behavior of the cluster? We can examine this phenomenon in a small size first, e.g. by building four sub-clusters, each of which have four elements.

### B. Other Areas of Application

Besides to parallel discrete-event simulation, we plan to use the SBCs for other purposes too.

#### 1) Load generation for DNS64 and NAT64 tests

First of all, we are going to use them as a load generator in DNS64 [12] server and NAT64 [13] gateway performance tests. We have several results in this area. As for DNS64 servers, we compared the performance of BIND and TOTD in [14]. We found a bug and a security hole in TOTD and provided a patch for correcting them in [16]. Later we also included the performance analysis of Unbound and PowerDNS in [15]. We prepared an own test program called dns64perf for the performance analysis of DNS64 servers [17]. We also have and own DNS64 implementation called MTD64 [18]. As for NAT64 gateways, we compared the performance of TAYGA+iptables and of OpenBSD PF using ICMP in [19] and later also TCP and UDP in [20]. In our further experiments, we plan to use a 16 element cluster of SBCs for load generation for DNS64 and NAT64 tests. This is an area where we expect that having a Gigabit Ethernet NIC will be an advantage.

#### 2) Testing and using as MPT servers

The MPT network layer multipath communication library [21] makes it possible to aggregate the transmission capacity of multiple interfaces of a device. Its channel aggregation capability was tested for two channels in [22], four channels in [23]–[24], and twelve channels in [25]. MPT is also a good solution for wireless network layer roaming problems [26] and changing the communication interfaces (using different transmission technologies) without packet loss [27]. We plan to test some SBCs in the role of an MPT server.

## IX. 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. Ten single board computers (SBCs) were benchmarked. Their space, price and power consumption relative performance were also calculated and compared. It was found that the usage of different metrics resulted in different ranking order of the SBCs. Odroid-U3+ gave the best absolute and space relative performance whereas Raspberry Pi 2 B+ showed the best price and power consumption relative performance. Both SBCs have only 100Mbps Ethernet NICs. Different SBCs may be optimal choice for different purposes. We also gave the directions of our planned future research including building an SBC cluster for simulation, using a cluster of SBCs for load generation in DNS64 server and NAT64 gateway performance analysis and using individual SBCs as MPT servers.

## 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, pp. 30–33, Apr. 2013. 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, pp. 349–358, Jun. 2014. 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, 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, 2013, pp. 314–310. DOI: 10.1109/TSP.2013.6613941
- [6] G. Lencse and S. Répás, "Method for benchmarking single board computers for building a mini supercomputer for simulation of telecommunication systems", in *Proc. Int. Conf. on Telecomm. and Signal Processing (TSP 2015)*, Prague, 2015, pp. 246–251. DOI: 10.1109/TSP.2015.7296261
- [7] 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, pp. 395–411, Apr. 2000. DOI: 10.1109/71.850835
- [8] G. Lencse and A. Varga, "Performance Prediction of Conservative Parallel Discrete Event Simulation", in *Proc. 2010 Industrial Simulation Conf. (ISC'2010)* Budapest, 2010, pp. 214–219.
- [9] 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)*, Delft, 2003, pp. 81–92.
- [10] 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
- [11] 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, 2013, pp. 101–107.
- [12] M. Bagnulo, A. Sullivan, P. Matthews and I. Beijnum, "DNS64: DNS extensions for network address translation from IPv6 clients to IPv4 servers", IETF RFC 6147 Apr. 2011.
- [13] M. Bagnulo, P. Matthews and I. Beijnum, "Stateful NAT64: Network address and protocol translation from IPv6 clients to IPv4 servers", IETF RFC 6146, Apr. 2011.
- [14] G. Lencse and S. Répás, "Performance analysis and comparison of different DNS64 implementations for Linux, OpenBSD and

FreeBSD”, in *Proc. IEEE 27th Internat. Conf. on Advanced Information Networking and Applications (AINA 2013)*, Barcelona, Spain, 2013, pp. 877–884. DOI: 10.1109/AINA.2013.80

- [15] G. Lencse, S. Répás, “Performance analysis and comparison of four DNS64 implementations under different free operating systems”, *Telecommunication Systems*, in press.
- [16] G. Lencse and S. Répás, “Improving the performance and security of the TOTD DNS64 implementation”, *Journal of Computer Science and Technology (JCS&T)*, ISSN: 1666-6038, vol. 14, no. 1, pp. 9–15. Apr. 2014.
- [17] G. Lencse, “Test program for the performance analysis of DNS64 servers”, *Internat. J. of Advances in Telecomm., Electrotechnics, Signals and Systems*, vol. 4, no. 3, pp. 60–65. Sep. 2015. DOI: 10.11601/ijates.v4i3.121
- [18] G. Lencse and A. G. Soós, “Design of a tiny multi-threaded DNS64 server”, in *Proc. 38th Internat. Conf. on Telecommunications and Signal Processing (TSP 2015)*, Prague, 2015, pp. 27–32. DOI: 10.1109/TSP.2015.7296218
- [19] G. Lencse and S. Répás, “Performance analysis and comparison of the TAYGA and of the PF NAT64 implementations”, in *Proc. 36th Internat. Conf. on Telecommunications and Signal Processing (TSP 2013)*, Rome, 2013, pp. 71–76. DOI: 10.1109/TSP.2013.6613894
- [20] S. Répás, P. Farnadi, and G. Lencse, “Performance and stability analysis of free NAT64 implementations with different protocols”, *Acta Technica Jaurinensis*, vol. 7, no. 4, pp. 404–427. Oct. 2014. DOI: 10.14513/actatechjaur.v7.n4.340
- [21] B. Almási, A. Harman, “An overview of the multipath communication technologies”, in *Proc. Conf. on Adv. in Wireless Sensor Networks 2013 (AWSN 2013)*, Debrecen, 2013, pp. 7–11.
- [22] B. Almási, Sz. Szilágyi, “Throughput performance analysis of the multipath communication library MPT”, in *Proc. 36th Int. Conf. on Telecommunications and Signal Processing (TSP 2013)*, Rome, 2013, pp. 86–90. DOI: 10.1109/TSP.2013.6613897
- [23] B. Almási, Sz. Szilágyi, “Multipath ftp and stream transmission analysis using the MPT software environment”, *Int. J. of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 11, pp. 4267–4272, Nov. 2013.
- [24] B. Almási, Sz. Szilágyi, “Investigating the performance of the MPT multipath communication library in IPv4 and IPv6”, *Internat. J. of Advances in Telecomm., Electrotechnics, Signals and Systems*, under review
- [25] G. Lencse and Á. Kovács, “Advanced measurements of the aggregation capability of the MPT multipath communication library”, *Internat. J. of Advances in Telecomm., Electrotechnics, Signals and Systems*, vol. 4, no. 2, pp. 41–48, May 2015, DOI: 10.11601/ijates.v4i2.112
- [26] B. Almási, “A simple solution for wireless network layer roaming problems”, *Carpathian Journal of Electronic and Computer Engineering*, vol. 5, no. 1, pp. 5–8, 2012.
- [27] B. Almási, “A solution for changing the communication interfaces between WiFi and 3G without packet loss”, in *Proc. 37th Int. Conf. on Telecommunications and Signal Processing (TSP 2014)*, Berlin, Jul. 2014, pp. 73–77.



**Gábor Lencse** received his MSc in electrical engineering and computer systems at the Technical University of Budapest in 1994, and his PhD in 2001.

He has been working for the Department of Telecommunications, Széchenyi István University, Győr, Hungary since 1997. He teaches computer networks and the Linux operating system. Now, he is an Associate Professor. He is responsible for the specialization of the information and communication technology of the BSc level electrical engineering education. He is a founding member and also core member of the Multidisciplinary Doctoral School of Engineering Sciences, Széchenyi István University. The area of his research includes discrete-event simulation methodology, performance analysis of computer networks and IPv6 transition technologies. He has been working part time for the Department of Networked Systems and Services, Budapest University of Technology and Economics (the former Technical University of Budapest), Budapest, Hungary since 2005. There he teaches computer architectures and computer networks.

Dr. Lencse is a member of IEEE, IEEE Computer Society and the Institute of Electronics, Information and Communication Engineers (IEICE, Japan).



**Sándor Répás** received his BA in business administration and management from the Corvinus University of Budapest, Budapest, Hungary in 2009, BSc in electrical engineering from the Óbuda University, Budapest, Hungary in 2011 and MSc in electrical engineering from the Széchenyi István University, Győr, Hungary in 2013.

He is a full time PhD student in information technology at the Széchenyi István University, Győr, Hungary. The main field of his research is the IPv6 implementation technologies. His other

favorite topics are computer networking, information security, and critical information infrastructure protection. He has several certificates from Cisco, ISACA, Microsoft, MikroTik, Novell, and other vendors.

Mr. Répás is a student member of the Association for Computer Machinery (ACM), and member of the Information Systems Audit and Control Association (ISACA), and the John von Neumann Computer Society (NJSZT).