Inexpensive Green Mini Supercomputer Based on Single Board Computer Cluster

T.K.Priyambodo, A.W.Lisan, and M.Riasetiawan
Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada.
mastri@ugm.ac.id

***Abstract*—This work focused on building a cluster computer named Wisanggeni 01 as mini-supercomputer for high performance computing, research, and educational purposes. Wisanggeni 01 was constructed from 33 node Raspberry Pi 2. Wisanggeni 01 runs Rasbian Whezzy OS with MPICH as parallel protocol. The Wisanggeni 01 performance is optimized with overclocking. The Wisanggeni 01 performance was tested with HPL benchmark, temperature test, and power consumption test. The result indicated that the peak performance of Wisanggeni 01 are 6020 MFLOPS with N=59000 at default clock and 9943 MFLOPS with N=55000 when overclocked. Average temperature when idle is 27.1°C – 30.2°C and 31.2°C – 34.6°C when running HPL benchmark. Average temperature at overclocked mode increase 2°C higher. Maximum wattage load of Wisanggeni 01 are 110W at default clock and 125W at overclocked clock. Power consumption used are 56% Raspberry Pi, 31% switch, and 13% cooler – LED.**

***Index Terms*—Green Computing; High Performance Computing; Low Power; Overclocking; Parallel Computer; Raspberry Pi.**

I. INTRODUCTION

Supercomputer is a parallel computer that uses high-level computation, which cannot be solved, by a normal computer [1]. The building of the supercomputer requires huge budget, wattage consumption, and complex infrastructure. In 2012 Raspberry Pi [2] was introduced: credit card sized computer, cheap, and low power consumption. The motivation of building the Raspberry Pi cluster is for developing a low-cost mini supercomputer for educational purpose. Raspberry Pi cluster also opens up an opportunity to build a parallel computer, which is cheap, low power, small, and requires a simple infrastructure.

The main goal of this research is building Raspberry Pi cluster for other experiments, trainings, and high-level ARM computation purposes in the laboratory. The research and training involves the application creation on the cluster, parallel computer architecture, and the learning on the parallel system subject.

II. LITERATURES

Simon Cox et.al., from University of Southampton is the first to popularize Raspberry Pi cluster project with their ‘Iridis Pi’ [3] consists of 64 Raspberry Pi 1A. The main purpose of their Iridis Pi is to build mini supercomputer prototype with low cost and low power consumption (*green computing*). Another experiment from Michael Cloutier, et.al. With comparison of 10 ARM based computer for performance per watt and price concluded that Raspberry Pi is the best [4].

The Raspberry Pi clusters development for educational and research purposes has been built by Shaun Franks and Johnathan Derby at Middle Georgia State College [5]. The Raspberry Pi development for specific and real computation purposes was done by Pezaros [6] and Mardhani for cloud computing and database server via Hadoop [7]. Those clusters were enclosured on lego case just like Cloutier done on Iridis Pi. Unfortunately the Mardhani’s cluster is not running properly because of overheating issue. Therefore, a good cooling system is needed on functional Raspberry Pi cluster for heavy computation. Kliepert with his Beowulf Raspi cluster [8] build a case with push-pull fan for cooling but immobile because the switch and ethernet cable still outside the case. The first building of ATX-sized case for all Raspberry Pi cluster hardware was done by David at 2014 but still has some problem on cooling system and enclosure material.

III. METHODS

The present research activity is to build a cluster based on Raspberry Pi 2B named ‘Wisanggeni 01’ as shown on Figure 1. Main experiment conduct on performance optimization with overclocking to prepare Wisanggeni 01 at high-level ARM computation and in depth analysis with Amdahl’s Law. A custom case will also be built with ATX-sized to enclosure all of cluster hardware into mobile supercomputer case, completed with cooling system and cable management.

Wisanggeni 01 consists of 33 node of Raspberry Pi 2B. Each Raspberry Pi is installed with 16GB Sandisk Extreme MicroSD. All of nodes are connected to HP V1810 J9660A-48 port switch through Belden CAT-6 ethernet cable. Unitek USB Hub Y2155 used for power source to each Raspberry Pi with output rating 5V/2.1A, also to prepare the power consumption space at overclocked mode. Wisanggeni 01 follows standard cluster configuration with one node act as head node while the remaining as slave node. The head node is connected to first port while the slave at 2nd to 33th port. The total cost of those hardware based on 1 USD = 13.600 IDR rating is Rp 32.922.000,00 or approx USD 2500.

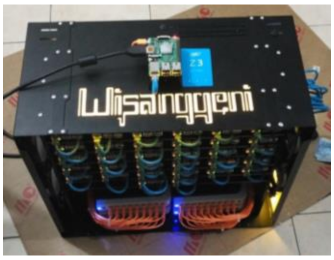


Figure 1: Wisanggeni 01Cluster

Wisanggeni 01 run Rasbian Whezzy as operating system (OS). We have tested that the newest Rasbian Jessie still inferior to old but updated Whezzy at parallel performance. The benchmark used to measure Wisanggeni 01 performance is HPL (High Performance Linpack) which compute linear algorithm Ax = B on P and Q matrix [9]. In this experiment run HPL 2.1 version with ATLAS BLAS [10] standard library. Another user for its specific cluster application should install another parallel program.

Wisanggeni 01 uses HSF (Heatsink Fan) cooling system. Each Raspberry Pi is installed with aluminium heatsink, delided base for better performance. The fan is installed on case with push-pull model with Thermaltake RGB Ring 140mm as push fan and Corsair AF-140 as pull fan. For head node, which placed on outside case, has its own fan with Delta 90mm 4500RPM. A custom case is made to Wisanggeni 01, made of metal and acrylic. The case consists of several part: Raspberry Pi rack, switch, push-pull fan as cooling system, power breakout, and cable management. This case dimensions are 254mm x 502mm x 444mm with the weight of 28 kg.

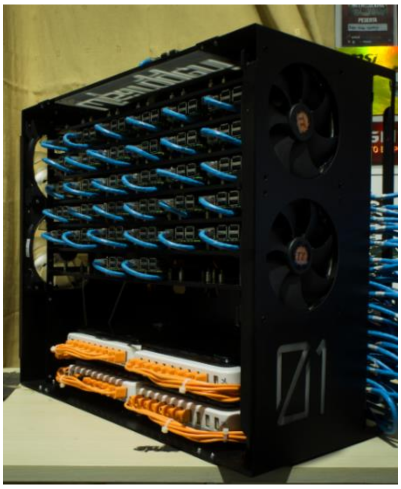
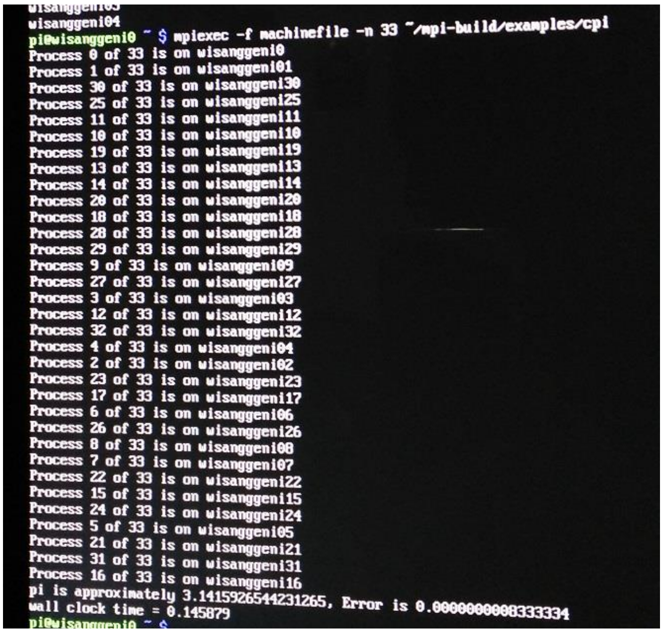


Figure 2: The complete setup of Wisanggeni 01 case

The head node is placed on outside top case where the slave node is inside the detachable case rack. Each rack could be installed to six nodes as shown at Figure 2. The left side of case accommodates all of Raspberry Pi and USB Hub. On the right side is for power breakout, cable management, and switch. The push-pull fan located on front and back case. The complete setup of case is shown at Figure 2.

IV. Implementation

The first step is build an image OS that has been installed with Rasbian Whezzy and MPICH [11], named 'cluster OS', then this cluster OS is copied into all MicroSD with Win32disk Imager software [12]. The next steps is detect the random dynamic IP address given from switch through iPerf then change the address into static and sort address. Each node communicate through SSH (Secure Shell) from Rasbian but SSH transaction on Rasbian needs login with pi username and raspberry password. To bypass that login need to make a passwordless SSH system with rewrite the password at id_rsa.pub Wisanggeni 01 into null password. The last step is testing the parallel protocol with run a phi parallel calculation example from mpich_build folder. The successful result is shown on Figure 3. For benchmarking test, HPL 2.1 is installed only on head node with Fortran compiler and ATLAS BLAS installation first.



```
wisanggeni03
wisanggeni04
pi@wisanggeni0 ~ $ mpiexec -f machinefile -n 33 ~/mpi-build/examples/cpi
Process 0 of 33 is on wisanggeni0
Process 1 of 33 is on wisanggeni01
Process 30 of 33 is on wisanggeni30
Process 25 of 33 is on wisanggeni25
Process 11 of 33 is on wisanggeni11
Process 10 of 33 is on wisanggeni10
Process 19 of 33 is on wisanggeni19
Process 13 of 33 is on wisanggeni13
Process 14 of 33 is on wisanggeni14
Process 20 of 33 is on wisanggeni20
Process 18 of 33 is on wisanggeni18
Process 28 of 33 is on wisanggeni28
Process 29 of 33 is on wisanggeni29
Process 9 of 33 is on wisanggeni09
Process 27 of 33 is on wisanggeni27
Process 3 of 33 is on wisanggeni03
Process 12 of 33 is on wisanggeni12
Process 32 of 33 is on wisanggeni32
Process 4 of 33 is on wisanggeni04
Process 2 of 33 is on wisanggeni02
Process 23 of 33 is on wisanggeni23
Process 17 of 33 is on wisanggeni17
Process 6 of 33 is on wisanggeni06
Process 26 of 33 is on wisanggeni26
Process 8 of 33 is on wisanggeni08
Process 7 of 33 is on wisanggeni07
Process 22 of 33 is on wisanggeni22
Process 15 of 33 is on wisanggeni15
Process 24 of 33 is on wisanggeni24
Process 5 of 33 is on wisanggeni05
Process 21 of 33 is on wisanggeni21
Process 31 of 33 is on wisanggeni31
Process 16 of 33 is on wisanggeni16
pi is approximately 3.1415926544231265, Error is 0.0000000008333334
wall clock time = 0.145879
pi@wisanggeni0 ~ $
```

Figure 3: Successful result of phi parallel calculation example

The overclocking is applied on Raspberry Pi with changing the clock configuration on boot config [13]. The overclocking on Wisanggeni 01 is limited to VID 1.3V or over_voltage 4 to maintain the operational lifetime usage. After we binned all of Raspberry Pi, the final setting to be applied is ARM/Core 1080/500 Mhz and SDRAM 483 Mhz, compared to default ARM/Core 900/250Mhz and SDRAM 450 Mhz. We use the golden ratio ARM:Core = 2:1.

Cluster performance test is done through HPL benchmark on 1,3,5,9,17, and 33 node. The result of HPL benchmark is FLOPS (Floating Operations Per Seconds). HPL is set on NB 128. Each P and Q matrix is different depends on node size. The benchmark begins from Ns = 5000 until the cluster fail to finish the benchmark.

We run temperature test on ARM processor and case (ambient temp). We measure the ARM temp with internal sensor on BMC2836 while ambient temp on case using laser thermometer. The result test will be compared on default clock and overclocked mode. We select certain nodes among 33 nodes: head node, slave node no. 17 & 23 (close to push fan), no 15 & 21 (in the middle), and 13 & 19 (close to pull fan). The selected node positions is shown on Figure 5. Power consumption of this cluster is measured using wattmeter with the variations of hardware: Raspberry Pi, switch, and cooling system (and other stuff like LED).

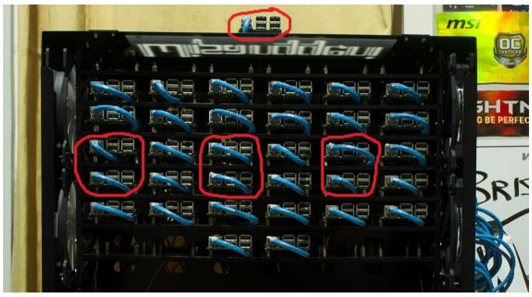


Figure 4: Selected Nodes for Temperature Test

V. Results and Discussion

The result of HPL benchmark is shown on Figure 5. Each Ns has different result on HPL benchmark. FLOPS score increased among with Ns load until reached the highest score named peak performance where the Ns load was suitable for certain nodes. Table 1 presented the peak performance for each nodes. From all of those results concluded that a cluster will reach its most efficient performance on the right load.

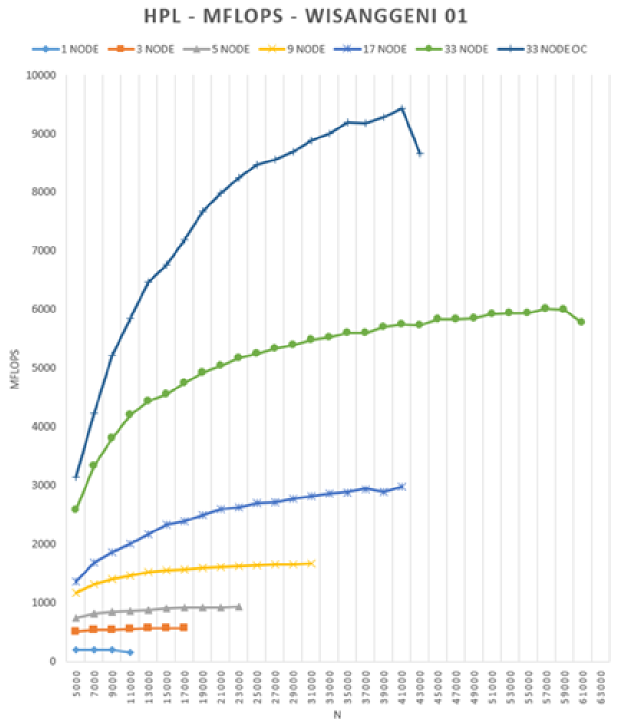


Figure 5: HPL benchmark result graph

Table 1
Peak performance of each number of nodes

Node	P	Q	Ns	SCORE (MFLOPS)	TIME (S)
1	1	1	9000	199.6	2435.9
3	1	3	15000	570.1	3947.55
5	1	5	23000	930.3	8670.18
9	3	3	31000	1674	11863.31
17	1	17	41000	2983	15403.91
33	3	11	59000	6020	22828.03

HPL benchmark testing at overclocked node only done on 33 node because safety reason. The result show significant gain difference from default clock which is 9433 MFLOPS but with lower Ns = 41000. Overclocked node failed to finish the benchmark higher than Ns 43000. There is no further testing but we concluded that overclocked cluster has much higher performance than default but not suitable for long-running computation. It's only recommended to run short computation below than 2 hours.

In depth analysis of performance is focused on speedup S(N). In this experiment we calculate the speedup based on Ns = 9000. The speedup is calculated with Equation (1).

$$Speedup = \frac{T(1)}{T(N)} \qquad (1)$$

where T(1) is calculation time at single node and T(N) is calculation time on parallel number of nodes N. The graph result is presented on Figure 6 where the orange line is the theoretical speedup with 100% parallel fraction (Pr) while the blue line is Wisanggeni 01 speedup. We used parallel fraction of 0.9598 or 95.98% based on the mean parallel of each number of nodes presented at Table 2. Parallel fraction is calculated with Equation (2).

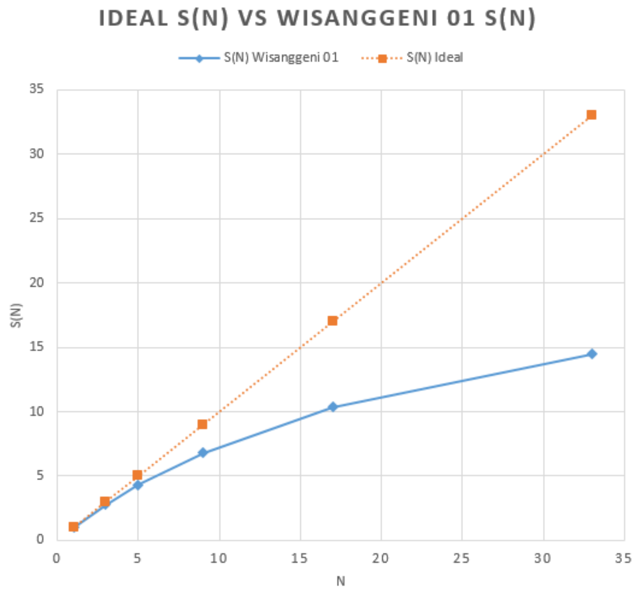


Figure 6: Ideal S(N) versus Wisanggeni 01 S(N)

Table 2
S(N) and Pr for each number of nodes

Node and Matrix		TIME (s)	S(N)	Pr from S(N)
1	P=1 Q=1	2453.9	1	0
3	P=1 Q=3	904.2	2,71	0,947
5	P=1 Q=5	573.32	4,28	0,958
9	P=3 Q=3	347.07	7,07	0,966

17	P=1
	Q=17
		260.76	9,41	0,95

33	P=3
	Q=11
		127.49	19,24	0,978

$$Speedup(N) = \frac{1}{(1-\mathrm{Pr}) + \frac{Pr}{N}} \qquad (2)$$

Next with the same parallel fraction we tried to continue the S(N) with (2) as extrapolation. The result shown at Figure 7 sounds the principal of Amdahl's Law [14] that the cluster reached the point with linear S(N) no matter with increased number of nodes. The linear S(N) begins at 128 nodes. Assumed that Ns 9000 is ideal for real ARM computation level could be assumed that 128 nodes is the maximum number for efficient Raspberry Pi cluster, unless there is a more efficient parallel program (although 95.98% parallel fraction is a very efficient parallelzation works).

The result of temperature tests on head node and slave node are summarized on Table 3 and Table 4. The idle test is conducted with duration of 30 minutes while the full load at 50 minutes. Temperature test is done on air-conditioned room that kept on 25°C ambience.

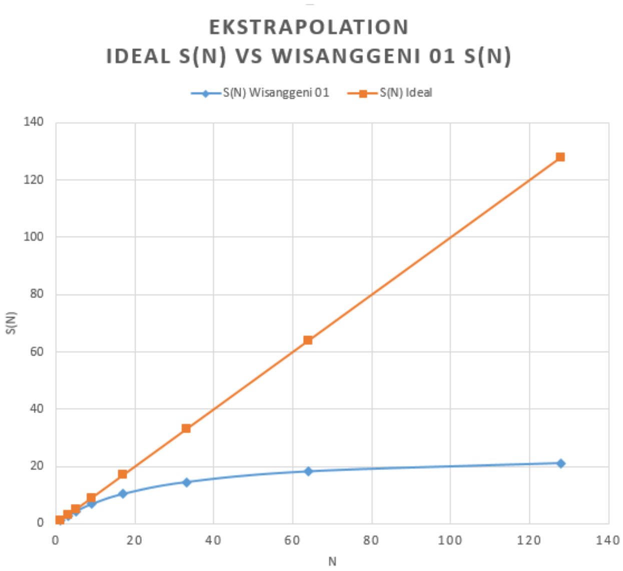


Figure 7: Extrapolation of ideal S(N) versus Wisanggeni 01 S(N)

Table 3
Temperature test result on head node

Test	DF mean	OC mean	Min DF	Min OC	Max DF	Max OC
Idle	27.1°C	29.1°C	26.1°C	28.2° C	27.7°C	29.9°C
HPL	31.2°C	31.6°C	30.4°C	31.5° C	31.5°C	32.6°C

Table 4
Temperature test result on slave node

Test	DF mean	OC mean	Min DF	Min OC	Max DF	Max OC
			Overall			
Idle	29.2°C	33.7°C	27.2° C	31.5°C	31.5°C	25.8°C
HPL	33.6°C	38°C	30.9° C	34.2°C	35.8°C	40.6°C
			Node 17 & 23			
Idle	27.3°C	31.9°C	27.2° C	31.5°C	28.2°C	32.6°C
HPL	31.6°C	35.7°C	30.9° C	34.2°C	32.6°C	36.9°C
			Node 13,15,19,21			
Idle	30.2°C	34.9°C	29.3° C	34.2°C	30.9°C	35.8°C
HPL	34.6°C	39.2°C	33.1° C	37.4°C	35.8°C	40.6°C

The cooling system shows its efficient works to drop the temperature significantly, at least 10°C from without cooling. The peak temperature ever recorded is 40.6°C on overclocked node. The cooling system is able to keep the full load temperature while HPL benchmark run at the background, with maximum 4.4°C difference of idle-full temp. Overclocked node has higher operational works compared to default but still on secure range. While testing it's also noticed that temp decreased quickly after the benchmark finished, one of the sign that the cooling system worked efficiently.

At slave node test result, it's obviously noticed that nodes near push fan (node 17 & 23) has lower temp than the rest (node 13,15,19,21) but with little difference. In the ambience test presented at Figure 8 also show that the zone near push fan has colder temp than in the middle and near pull fan. It's concluded that the cooling system at Wisanggeni 01 works perfectly. All of raspberry pi nodes never reaches temp higher than 50°C on HPL test even with hot ambience temp room.

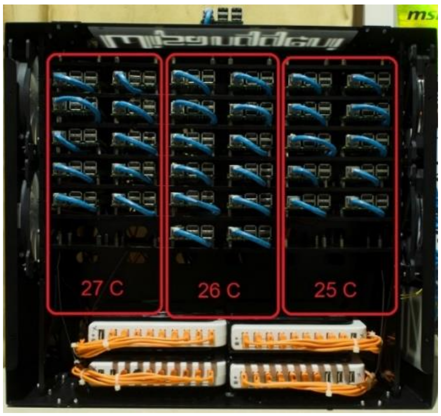


Figure 8: Ambient case temperature result

The result obtained from test is presented on Table 5. The ratios of power consumption on Wisanggeni 01 are 56% Raspberry Pi, 31% switch, and 13% cooling system and LED.

Table 5
Wisanggeni 01 Power consumption

Part	Default		Overclocked	
	Idle	HPL	Idle	HPL
Raspberry Pi	57W	66W	68W	80W
Switch	32W			
Cooler & LED	13W			
Overall	101W	110W	112W	125W

Based on those results we could calculate Wisanggeni 01 performance efficiency or GFLOPS per watt, presented at Table 6. One of the interesting result is, overclocked node has better GFLOPS per watt efficiency than default ones although has higher power consumption. This case shows that increase Raspberry Pi cluster performance through overclocking is efficient and not considered as 'brute force' way.

Table 6
Wisanggeni 01 performance per watt

Node	Peak GFLOPS	Wattage (W)	GFLOPS/W
Default	6,02	110	0,054
OC	9,43	125	0,075

For the ending we compare the Wisanggeni 01 with modern high-end processor performance like Core i5 and i7 Nehahelm and newer. Wisanggeni 01 is totally outnumbered with Intel processors on peak FLOPS and performance per watt because of architecture difference and purposes. ARM processors mainly built for as low as power consumption possible. The comparison summarized on Table 7.

Table 7
Performance comparison of Wisanggeni 01 and Intel processors

Hardware	Peak GLOPS	Full load wattage	GFLOPS/watt
Wisanggeni 01 default	6,02	110	0,054
Wisanggeni 01 OC	9,43	125	0,075
Intel Core i7 970	40	213	0,187
Intel Core i5 3570	105	112	0.937
Intel Core i7 4770K	182	148	1.229

In addition, here we compare the Wisanggeni 01 with other Raspberry Pi clusters. Unfortunately we cannot find the similar cluster (32-33 nodes Raspberry Pi 2) for fair comparison of peak GFLOPS so we compare the cluster on efficiency term (GFLOPS/Watt). The complete comparison shown on Table 8. The data found from other paper references [3][4]. Overall Wisanggeni 01 efficiency is higher than other Raspberry Pi clusters. There's an interesting result when Cloutier Raspberry Pi 1B overclocked cluster could beat Wisanggeni 01 Raspberry Pi 2B default clock both at peak and efficiency performance. It shows that overclocking is an effective way to boost up the cluster performance.

Table 8
Comparison of Wisanggeni 01 with other Raspberry Pi clusters

Cluster	Node	Wattage	GFLOPS/W
Wisanggeni 01 DF	33 Pi 2B default	110	0.055 (peak 6.02 GF)
Wisanggeni 01 OC	33 Pi 2B overclocked	125	0.075 (peak 9.4 GF)
Cox's Iridis Pi	64 Pi 1A	112	0.01 (peak 1.14 GF)
Cloutier's cluster DF	32 Pi 1B default	93	0.046 (peak 4.37 GF)
Cloutier's cluster OC	32 Pi 1B overclocked	112	0.056 (peak 6.25 GF)

VI. CONCLUSION

From all of those results and analysis, the conclusions are Wisanggeni 01 cluster is fully operational for parallel computing with decent performance on FLOPS and operational temperature. Successful overclocking configuration on Wisanggeni 01 are ARM/Core 1080/550 Mhz and SDRAM 483 Mhz. Overclocked node has significant performance gain than default clock. Peak performance of Wisanggeni 01 on default clock is 6020 MFLOPS at Ns 59000. Peak performance on overclocked mode is 9433 MFLOPS at Ns 41000. Default clock has stock performance but could run very intensive load while overclocked clock has significant performance gain but couldn't run intensive computation. Cooling system works efficiently both on default and overclocked mode. Highest temperature ever recorded on testing was 40.6°C.

Power consumption on Wisanggeni 01 at default clock are 101W idle and 110W full load. Power consumption at overclocked clock are 112W idle and 125W full load. The ratio of power consumption is 56% Raspberry Pi, 31% switch, and 13% cooling system and LED. The efficiency performance of Wisanggeni 01 is 0.054 GFLOPS/Watt at default and 0.075 GFLOPS/Watt at overclocked. Wisanggeni 01 performance couldn't reaches the same level with modern processors like Intel Core i5 and Core i7 because the different purposes of processor architecture.

ACKNOWLEDGMENT

The authors acknowledge to all contributors in building Wisanngeni 01 SBC-based cluster computer, especially thanks to Indonesian Ministry of Research, Technology and Higher Education for research grants with contract number 156/LPPM/2015 and 779/UN1-PIII/LT.DIT-LIT/2016. Thanks are also extended to the Department of Computer Science and Electronics Gadjah Mada University for the DIKE research grants.

REFERENCES

[1] Rajaraman, V., 1999, "Supercomputers", University Press: Jawaharlal Nehru Centre for Advanced Scientific Research, India.

[2] Raspberry Pi Foundation: Raspberry Pi FAQs. URL http://www.raspberrypi.org/faqs. Accessed Aug 2016

[3] Cox, S. J., Cox, J. T., Boardman, R., Johnston, S., Scott, M., Brien, N., 2013, "Iridis-Pi: a low-cost, compact demonstration cluster". University of Southampton, Southampton, UK. URL: http://www.southampton.ac.uk/~sjc/raspberrypi/raspberry_pi_iridis_lego_supercomputer_paper_cox_Jun2013.pdf.

[4] Cloutier, M.F., Paradis, C., Weaver, V.M., 2014, "Design and Analysis of a 32-bit Embedded High-Performance Cluster Optimized for Energy and Performance – Extended Edition". University of Maine. URL: http://web.eece.maine.edu/~vweaver/projects/pi-cluster/2014_cohpc_cluster_extended.pdf.

[5] Franks, S., Yerby J., 2014, "Creating a Low Cost Supercomputer with Raspberry Pi". Southern Association for Information Systems Conference, Macon, GA, USA. URL: http://saisconferencemgmt.org/proceedings/2014/FranksYerby.pdf.

[6] Tso, P.T., White, D.R., Jouet. S, Singer, J., Pezaros, D.P., 2013, "The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures". University of Glasgow, G12 8QQ, UK. URL: http://davidrwhite.co.uk/wp-content/uploads/2014/09/the_glasgow_raspberry_pi_cloud_a_scale_model_for_cloud_computing_infrastructures.pdf.

[7] Apache Hadoop. http://hadoop.apache.org/

[8] Kiepert, J., 2013, "RPiCLUSTER: Creating a raspberry pi-based beowulf cluster". Technical report, Boise State University. URL: http://coen.boisestate.edu/ece/files/2013/05/Creating.a.Raspberry.Pi-Based.Beowulf.Cluster_v2.pdf

[9] Dongarra, J.J., Luszczek, P., Petitet, A., 2003, "The linpack benchmark :past, present and future". Concurrency and Computation: Practiceand Experience. URL: http://onlinelibrary.wiley.com/doi/10.1002/cpe.728/epdf

[10] Dongarra, JJ., 2002, "Performance of various computers using standard linear equations software". Technical Report,University of Tennessee. URL: http://www.netlib.org/benchmark/performance.ps

[11] Amer, A., Balaji, P., Bland, W., Gropp, W., Latham, R., Lu, H., Oden, L., Pena, A., Raffenetti, K., Seo, S., Thakur, R., Zhang, J., 2015, "MPICH Installer's Guide Version 3.2", Mathematics and Computer Science Division Argonne National Library. URL: https://www.mpich.org/static/downloads/3.2/mpich-3.2-installguide.pdf

[12] Cox, S., 2013, "Steps to make Raspberry Pi Supercomputer". University of Southampton

[13] Plaisier, PJ., 2013, "Raspberry Pi Overclocking Guide – Cheap, Fun, and Overclockable (with HWBOT Prime)". URL: http://hwbot.org/news/9761_raspberry_pi_overclocking_%e2%80%93_cheap_fun_and_hwbot_prime_oc_guide, accessed Feb 2016

[14] Hill, M., Marty, M., 2007, "Amdahl's Law in the Multicore Era". IBM. URL: http://research.cs.wisc.edu/multifacet/papers/ieeecomputer08_amdahl_multicore.pdf