Harris Simaremare <harrismare@uin-suska.ac.id>

Successful Manuscript Submission to JATIT

1 pesan

JATIT <editor@jatit.org> 30 November 2020 14.55
Kepada: Harris-Simaremare <harrismare@uin-suska.ac.id>

We have received the manuscript for review and possible publication in Journal of Theoretical and Applied Information Technology.

ID of the manuscript is " 42891 " and Password is " Jkhxy "

Please note the manuscript ID and use this as reference for correspondence regarding your submission. For updates on the status of your submission, please visit http://www.jatit.org/enter_manuscript.php.

If you want to change your password, please use this URL: http://www.jatit.org/change_password.php

Please fill the agreement form attached and mail it to mailjatit@gmail.com. Submitting this form confirms that you are in agreement with publication polices of the journal. You may also downlaod the agreement from http://www.jatit.org/agreement_form.docx

Evaluation process can take 15-60 working days depending on the time the reviewers take to convey their opinion on your manuscript.

---

agreement_form.docx
13K

Harris Simaremare <harrismare@uin-suska.ac.id>

[JATIT] Letter of Acceptance for Submitted Research Paper ID 42891-JATIT

1 pesan

editor jatit <mailjatit@gmail.com> 26 Desember 2020 00.06
Kepada: harrismare@uin-suska.ac.id, roza.novianti@students.uin-suska.ac.id, rahmad.abdillah@uin-suska.ac.id

Dear Corresponding Author **Harris-Simaremare**

We are pleased to inform you that your submission ID**: 42892-JATIT** titled **"LARGE-SCALE IMAGE EDGE DETECTION USING HIGH-PERFORMANCE COMPUTING CLUSTER"** having author(s): **HARRIS SIMAREMARE, ROZA NOVIANTI, RAHMAD ABDILLAH** has been accepted for publication in **JOURNAL OF THEORETICAL AND APPLIED INFORMATION TECHNOLOGY** (E-ISSN **1817-3195** / ISSN **1992-8645).** The acceptance decision was based on the reviewers' evaluation after double-blind peer review and the chief editor's approval.**[Attached with this acceptance intimation]**

You shall submit the OA processing fee ($450) via Credit Card/PayPal transaction through our online payment system (Use any valid credit card of Yourself / Friend / Family etc) . Please submit the dues via UK Paddle payment system at

https://pay.paddle.com/checkout/507133

so that your paper may get published in upcoming issues. (please forward us with the receipt / order number generated after the completed payment process so that we can easily track your payment). The billing info that appears on your cc statement shall have a reference of JATIT. (Any Authentic Credit Card of Yourself / Friend / Family etc can be legitimately used).

There is also an option of urgent publication fee ($900) available for urgent publication.

Kindly also submit a camera-ready copy (CRC) with updates satisfying reviewer comments in MS Word document and exact journal format [http://www.jatit.org/author_guidelines.php] along with reply to reviewer comments document and copyright to mailjatit@gmail.com after registration fee submission.

Kindly proceed with registration fee submission for slot allocation in Vol 99 March 2021 Issues of the journal, to be assigned on the first APC submission basis. The final updated copy can be submitted at a later time after slot reservation.

We shall encourage more quality submissions from you and your colleagues in the future.

Please do acknowledge receiving this notification.

Regards,

Madiha Azeem PhD
Handling Editor
Editorial office
Journal of Theoretical and Applied Information Technology

5 lampiran

42891-JATIT Final Evaluation Report.pdf
101K

Author Guidelines and Formatting Template.doc
62K

JATIT Copyright.doc
27K

JATIT Reply to Reviewer Comments.docx
16K

Preliminary Evaluation Form.pdf
127K

JATIT PAPER EVALUATION FORM-II

Paper Number: 42891-JATIT

Note: Copies of the completed evaluation form and comments will be supplied to the author(s)
Please return your review results to the Chief Editor after blind peer review. Thank You.

==

Paper Title: LARGE-SCALE IMAGE EDGE DETECTION USING HIGH-PERFORMANCE COMPUTING CLUSTER

==

EVALUATION: (please circle the appropriate rating)

Tend to reject | Tend to accept

←-------------------------|-------------------------→

Technical Content and Accuracy	1	2	3	4	5	●	7	8	9	10
Significance of the Work	1	2	3	4	5	6	●	8	9	10
Appropriate Title, Introduction, and Conclusion	1	2	3	4	5	●	7	8	9	10
Overall Organization	1	2	3	4	5	6	●	8	9	10
Appropriateness for JATIT	1	2	3	4	5	6	7	8	●	10
Style and Clarity of the Paper	1	2	3	4	5	6	●	8	9	10
Connection to Previous Research	1	2	3	4	5	6	●	8	9	10
OVERALL RECOMMENDATION	**1**	**2**	**3**	**4**	**5**	**6**	●	**8**	**9**	**10**
As a Referee how do you rate your knowledge, ability and confidence in reviewing this paper	**1** Low	**2**	**3**	**4**	**5**	**6**	**7**	●	**9**	**10** High

COMMENTS TO AUTHOR: (Please use additional sheet(s) if necessary)

1. Hypothesize the problem clearly in the paper. And reflect on the same in conclusion in light of facts identified
2. Discussion on limitations of the work is missing
3. Difference of research contribution and achievement of objectives when compared to other such studies in light of results presented in literature is missing or is not clear in text.

==

Confidential Comments to the Chief Editor:

□□al□ati□□ □□r□

JATIT

Journal of Theoretical and Applied Information Technology

T□e e□□l□□ed □a□□□□ri□t i□ □□der □□□□iderati□□ □□r t□e □□□r□al□□lea□e □r□□ide □eed□a□□ □□ t□e □□ll□□i□□ □riteria □□ t□at □□rt□er □r□□e□□ □ □□e i□itiated

Mark where appropriate	YES	NO
I□it a re□ear□□ □r re□ie□ □a□er□	□	
I□it □it□i□ t□ t□e □□□□e □□t□e □□□r□al□	□	
I□it a □□ll □a□er □□□□ i□□i□□□	□	
I□t□e la□□□□a□e □□□a□er □□□li□□□ □□□t□ □□ relaxati□□□	□	
□ ill t□e □a□er □e □□i□tere□t t□ □□□r□al reader□□i□□	□	
□a□t□e □a□er □r □art □□it alread□ □ee□ □□□li□□ed el□e□□ere□ [Based on Google Search on Tile And Abstract]		□

Recommendations: Mark where appropriate.

Re□e□ted □□ter I□ter□al Re□ie□	
□□□e□ted □□ter I□itial Re□ie□ a□d Re□□□ □e□ded □□r Detaied Te□□□i□al Re□ie□	□

□Rela□ati□□ i□ □□l□i□ □□e□ial □a□e □□ere □□e □□a□□ □t□er la□□□a□e i□ □□rtail t□ □□r□ □re□e□ted □□it□er i□ ta□le□□i□□re□ □r te□t□

Reply TO REVIEWER COMMENTS AND CHANGE LOG

Note: Indicate the updates of changes in the manuscript in red colour font so that changes/updates are easy to track.

S.No	Comment	Reply to Comment / Change Description	Page No.
1)	Hypothesize the problem clearly in the paper. And reflect on the same in conclusion in light of facts identified	Therefore, this research aims to solve the problem of real-time edge detection processing in large pixel images using HPC and to know the performance of the PC-Cluster against the detection technique The PC-Cluster processing speed on a 1000Mbps switch is 55.53% faster than a 100Mbps switch. At the same time, the maximum performance of the PC-Cluster in the experiment was 3.657E + 00.	1 13
2)	Discussion on limitations of the work is missing	To the best of our knowledge, no one has performed edge detection testing on images using PC-Cluster. Only [13] used PC-Cluster to perform clustering of MRI objects. The PC-Cluster is built to reduce computation time. This is proven by the collaboration of low computer resources that can produce extraordinary performance. The PC-Cluster we build is limited to computer resources, if we want to increase performance, we recommend using a 1000Mbps switch and upgrading the PC-Cluster such as RAM and Processor.	13
3)	Difference of research contribution and achievement of objectives when compared to other such studies in light of results presented in literature is missing or is not clear in text	The computer cluster configuration used by [12] is two computer clusters connected to each other using a wide area network with a bandwidth of 10Gbps. Meanwhile, computer cluster [13] uses 4 PC-Cluster collaboration with Pentium IV 2.6 GHz CPUs and 256 MB RAM. Each computer is connected to the FastEthernet TCP / IP network. According to CPU Benchmark (a website that provides benchmark information against CPU, RAM, Video Card, Hard Drive, Android and IOS / iPhone), the PC-Cluster specifications	2

		we use are better than the PC-Cluster specifications used by [13].	
4)			
5)			
6)			
7)			
8)			
9)			
10)			

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

LARGE-SCALE IMAGE EDGE DETECTION USING HIGH-PERFORMANCE COMPUTING CLUSTER

[1]HARRIS SIMAREMARE, [2]ROZA NOVIANTI, [3]RAHMAD ABDILLAH

[1]Electrical Engineering, Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia

[2]Informatic Engineering, Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia

[3]Informatic Engineering, Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia

E-mail: [1]harrismare@uin-suska.ac.id, [2]roza.novianti@students.uin-suska.ac.id, [3]rahmad.abdillah@uin-suska.ac.id

ABSTRACT

The limitation of computational resources for processing large-scale images makes researchers unable to work optimally. PC-Cluster is an alternative as a computing machine on limited resources. This study tested Sobel performance as an edge detection technique on large-scale images using a PC-cluster. The experimental results show that the PC-Cluster can shorten the processing time of the single technique.

Keywords: *Edge Detection, PC-Cluster, Sobel*

1. INTRODUCTION

One of the image processing techniques to determine the edge of an image is Edge detection. This method can simplify higher level examinations and is actively developing new methods [1]. In an image, the edge represents the boundary of the object, and so it can be used to identify objects in the drawing area [2]. The first recognition of objects in a visual vision are lines and the types of line feature that are key. Line features can generally be seen from the outline, texture and boundaries of an object. Meanwhile, edge detection is essential information to get an image outline [3].

One of the most widely used techniques for image processing is Sobel. The Sobel technique is most widely used to perform data extraction and image segmentation in various implementation models [4], [5]. Sobel has a better level of efficiency than other edge detectors such as Prewitt and Robert [6]

Real-time image processing is a problem in edge detection [7] The edge detection process in the image must be very fast to produce other related control actions [6]. This is a challenge for image processing, especially edge detection in dealing with images with large pixel sizes [1]. Therefore, a large pixel size will have an impact on computational complexity [7].

High-Performance Computing (HPC) is a solution to overcome computational complexity [8]. HPC consists of various techniques such as computer architecture, system software, algorithms and programs that collaborate to solve computing problems quickly [1]. HPC makes a significant contribution to reducing the processing time and analysis of medical images [9], such as the reconstruction of a tomographic microwave image of the brain [10].

Very few use HPC technology to perform edge detection on images that have large pixels. Therefore, this research aims to solve the problem of real-time edge detection processing in large pixel images using HPC and to know the performance of the PC-Cluster against the detection technique. The edge detection used is Sobel because it has a better level of efficiency.

This paper is organized as follows. In section II, there is an introduction to the High-Performance Computing Cluster and Sobel technique. In section III, a discussion about the implementation of the single technique on the High-Performance Computing Cluster. The Sobel technique performs edge detection of large-scale images. Section IV displays the results of the single performance using the PC-Cluster and the built-in PC-Cluster performance. Finally, the explanation of the conclusions of the study.

Journal of Theoretical and Applied Information Technology
31st January 2021. Vol.99. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RELATED WORKS

2.1 High-Performance Computing Cluster

Many HPC techniques are used in different research results. The use of HPC has helped a lot in various scientific fields, especially in the medical field. A useful HPC requires a considerable cost which is directly proportional to the benefits obtained in various fields of research. The most popular use of HPC is collaborating between Graphics Processing Units and CPU Cores such as the use of 32 Intel Xeon CPU cores, 6 NVIDIA cards with Tesla GPU [11]. Because the costs required are huge, the use of HPC is constrained.

Computer clusters are an alternative to solve the HPC development cost problem. A computer cluster is a collection of computers called nodes that are interconnected and work together to solve computational problems so that the cluster computer performance is better than a single computer.

Some studies use computer clusters, such as medical image analysis and processing [12], [13]. The computer cluster configuration used by [12] is two computer clusters connected to each other using a wide area network with a bandwidth of 10Gbps. Meanwhile, computer cluster [13] uses 4 PC-Cluster collaboration with Pentium IV 2.6 GHz CPUs and 256 MB RAM. Each computer is connected to the FastEthernet TCP / IP network.

According to CPU Benchmark (a website that provides benchmark information against CPU, RAM, Video Card, Hard Drive, Android and IOS / iPhone), the PC-Cluster specifications we use are better than the PC-Cluster specifications used by [13]. Our study uses the PC-Cluster concept as applied by [12], but we use evaluation techniques that are different from them. We evaluate the effect of the Switch hardware interface on the performance of HPCs such as FastEthernet and Gigabit Ethernet TCP / IP network.

-1	-2	-1
0	0	0
1	2	1

Gx

-1	0	1
-2	0	2
-1	0	1

Gy

Figure 1: Sobel Matrix

The gradient value for each pixel is obtained from the horizontal and vertical gradient values using Eq. (1).

$$M = \sqrt{G_x^2 + G_y^2} \quad (1)$$

After that, we use Eq. (2) to get the magnitude of the gradient.

$$M = |G_x| + |G_y| \quad (2)$$

3. METHOD

4.3 High-Performance Computing Cluster

The PC-Cluster that we built consists of 4 nodes as slaves and one node for the master with Intel (R) Pentium (R) dual CPU E2180@2.00GHZ specifications, 512MB RAM and FastEthernet / GigabitEthernet Switch. We also install PC-Cluster support applications such as Ubuntu Desktop (master node operating system), NFS (Network File Sharing), OpenMPI (library for parallel processing), HTOP (monitoring process Node), Network Time Protocol application and High-Performance Linpack (Benchmark. cluster). Each node is connected to form a star topology, as shown in Figure 2.

Figure 2: PC Cluster

4.4 Large Scale Image

The images used in this study come from the website http://www.spacetelescope.org/images/heic1502a/ and the website https://commons.wikimedia.org/wiki/File:View_from_eiffel_tower_2nd_level.jpg. Each image is made of changes, as in table 1.

Journal of Theoretical and Applied Information Technology
31st January 2021. Vol.99. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Large scale image

Image	Size		Pixel
Original Image	1	1,6 GB	12788 x 40000
	2	38,2 MB	11683 x 25000
	3	501 MB	14321 x 29566
	4	444 MB	12392 x 15852
	5	275 MB	13775 x 21299
Crop the Original Image	1	1,25 GB	12788 x 35000
	2	31,4 MB	10585 x 20750
	3	324 MB	11723 x 24366
	4	410 MB	10473 x 13693
	5	216 MB	13375 x 17249
Rotate the Original Image	1	1,70 GB	40000 x 12788
	2	37,6 MB	25000 x 11683
	3	502 MB	29566 x 14321
	4	439 MB	15852 x 12392
	5	275 MB	21299 x 13775
Grayscale the Original Image	1	1,6 GB	12788 x 40000
	2	38,2 MB	11683 x 25000
	3	501 MB	14321 x 29566
	4	444 MB	12392 x 15852
	5	275 MB	13775 x 21299

4.5 Propose work

Here are the methods we propose for implementing edge detection on PC-Clusters:

1) Calls up the rank information on the communicator.
2) Overall time.
3) Time to start reading pictures.
4) Read RGB image / image.
5) Convert an RGB image / image to a grayscale image / image.
6) Count the number of elements in the grayscale matrix.
7) Define rows and columns on the grayscale matrix.
8) Stop reading time of images.
9) Starting time of sending pictures.
10) Broadcast the number of matrix elements to each node.
11) Check if rank = size-1, if rank = size-1 then continue to the next step otherwise go to step 17.
12) Divide the column and image by the number of nodes
13) Send all data to each node/rank using MPI_Scatter.
14) Stop timing of sending images.
15) Start time of operation Sobel.
16) Check the boundary of the first line, if the boundary of the first line = 0, then go to the next step otherwise go to step 19.
17) Copy the value 1 to the first-row region.
18) Check the boundary of the last line, if the boundary of the last line is> -1, then go to the next step otherwise go to step 21.
19) Copy the grey object with the rows method to the last row area then subtract 1.
20) Loop lines on the whole image.
21) Calculate the gradient value x.
22) Calculate the gradient y value.
23) Calculate the magnitude of the x gradient and the y gradient.
24) Check if matrix value 255 <sum <0, if matrix value 255 <sum <0, then continue to the next step otherwise go back to step 24.
25) Access pixel value and convert matrix value to image intensity value.
26) Stop timing of Sobel operation.
27) Start image stitching time.
28) Send all data to rank 0 using MPI-Gather.
29) Stop the image stitching time.
30) Stop the whole time.
31) Check if rank = 0, if rank = 0 continue to the next step otherwise the process is complete.
32) Save the edge detection image in the specified folder.
33) Show execution time.

4. RESULT AND DISCUSSION

We made two for testing models, such as the use of FastEthernet and GigabitEtherner switches. Tests are made crosswise between the switch and the master node, such as the FastEthernet Switch with the GigabitEthernet Master Node so on.

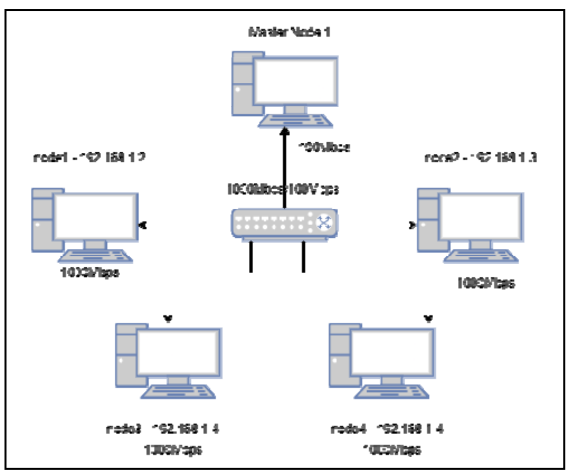


Figure 3: Cluster Topology for Node 1

In figure 3, we are using a 100Mbps switch with a 1000Mbps ethernet cable and a NIC on a 100 Mbps master node. Each node uses a 1000Mbps NIC. Star topology is used in HPC. IP address used 192.168.1.1 - 192.168.1.5.

Table 2: Edge Detection Processing Time (S) on Master Node 1 for the original image using 100Mbps switch

Node	Transfer	Switch 100Mbps							
		Time (second)							
		Sobel Operation						Combine	Final
		Master	node1	node2	node3	node4	All		
1	21,942	10,790	12,020	-	-	-	22,821	26,219	140,770
2	29,135	7,183	6,600	6,588	-	-	20,373	29,187	149,401
3	32,946	5,476	4,934	4,834	4,932	-	20,278	32,825	158,371
4	34,882	4,335	3,940	3,944	3,949	3,951	20,119	34,997	153,611

In table 2, the time required by master node 1 to use a 100Mbps switch is 153,611. Whereas in table 3, the time required by master node 1 when using a 1000Mbps switch is 150,144, so there is an increase in the single detection processing time between tables 2 and 3 by 2.31%. This is due to the use of a 1000Mbps switch which can increase data transfer. As well as the ability to use the auto-negotiation feature on the NIC (network interface card) on the master node 1. This feature can analyze the best data transfer speed based on the environment.

Table 3: Edge Detection Processing Time (S) on Master Node 1 for an original image using 1000Mbps switch

Node	Transfer	Switch 1000Mbps							
		Time (second)							
		Sobel Operation						Combine	Final
		master	node1	node2	node3	node4	All		
1	22,216	10,720	11,890	-	-	-	22,619	24,105	133,112
2	29,356	7,138	6,563	6,653	-	-	20,356	29,033	142,603
3	33,221	5,370	4,954	4,936	4,945	-	20,208	32,639	148,240
4	35,209	4,328	3,948	3,948	3,948	3,947	20,120	34,810	150,144

The average completion time of the Sobel operation in table 2 is 150.5383 (s), while table 3 is 143.5248 (s). the difference is only 7.0135 (s) in the processing time of the Sobel operation. In contrast to the average transfer time between nodes in tables 2 and 3, namely 30.0005 (s) and 29.72625 (s). the difference in data transfer is only 0.27425 (s).

Table 4: Edge Detection Processing Time (S) on Master Node 1 for a cropped image using 100Mbps switch

Node	Transfer	Switch 100Mbps							
		Time (second)							
		Sobel Operation						Combine	Final
		Master node 1	node1	node2	node3	node4	All		
1	19,189	9,350	8,710	-	-	-	18,060	19,207	80,518
2	25,481	6,306	5,740	5,761	-	-	17,808	25,545	90,778
3	28,640	4,683	4,325	4,316	4,317	-	17,642	28,711	96,263
4	30,516	3,782	3,451	3,451	3,462	3,461	17,610	30,620	102,767

In table 4, the time required by master node 1 using a 100Mbps switch is 102,767. Whereas in table 5, the time required by the master node 1 when using a 1000Mbps switch is 96,164, so there is an increase in the single detection processing time between tables 4 and 5 by 6.87%. This is due to the use of a 1000Mbps switch which can increase data transfer. As well as the ability to use the auto-negotiation feature on the NIC (network interface card) on the master node 1. This feature can analyze the best data transfer speed based on its environment.

Table 5: Edge Detection Processing Time (S) on Master Node 1 for the cropped image using 1000Mbps switch

Node	Transfer	Switch 1000Mbps							
		Time (second)						Combine	Final
		Sobel Operation							
		Master node 1	node1	node2	node3	node4	All		
1	19,230	9,372	9,372	-	-	-	19,146	20,978	82,563
2	25,483	6,224	5,752	5,765	-	-	17,742	25,415	89,804
3	28,646	4,677	4,324	4,320	4,317	-	17,640	28,559	93,330
4	30,521	3,711	3,460	3,459	3,462	3,461	17,555	30,459	96,164

The average completion time of the Sobel operation in table 4 is 92.5815 (s), while table 5 is 90.46525 (s). only 2.11625 (s) difference in the processing time of Sobel operation. In contrast to the average transfer time between nodes in tables 4 and 5, namely 25.9565 (s) and 25.97 (s). the difference in data transfer is only 0.0135 (s).

Table 6: Edge Detection Processing Time (S) on Master Node 1 for a rotate image using 100Mbps switch

Node	Transfer	Switch 100Mbps							
		Time (second)							
		Sobel Operation						Combine	Final
		Master node 1	node1	node2	node3	node4	All		
1	22,903	10,52	12,97	-	-	-	23,508	25,883	129,860
2	29,573	6,981	6,637	6,519	-	-	20,180	29,188	137,075
3	33,601	5,241	4,947	4,940	4,937	-	20,067	32,825	142,175
4	35,005	4,191	3,949	3,948	3,953	3,947	19,990	34,999	207,798

In table 6, the time required by master node 1 to use a 100Mbps switch is 207,798. Whereas in table 7, the time required by the master node 1 when using a 1000Mbps switch is 146,729, so there is an increase in the single detection processing time between tables 6 and 7 by 41.62%. This is due to the use of a 1000Mbps switch which can increase data transfer. As well as the ability to use the auto-negotiation feature on the NIC (network interface card) on the master node 1. This feature can analyze the best data transfer speed based on its environment.

Table 7: Edge Detection Processing Time (S) on Master Node 1 for the rotate image using 1000Mbps switch

Node	Transfer	Switch 1000Mbps							
		Time (second)						Combine	Final
		Sobel Operation							
		Master node 1	node1	node2	node3	node4	All		
1	21,953	10,65	11,97	-	-	-	22,633	24,386	136,627
2	29,122	7,095	6,566	6,561	-	-	20,223	29,042	139,782
3	32,750	5,298	4,938	4,930	4,937	-	20,105	32,644	145,306
4	34,890	4,250	3,951	3,949	3,948	3,953	20,052	34,810	146,729

The average time to complete the Sobel operation in table 6 is 154.227 (s), while in table 7 it is 142,111 (s). only 12,116 (s) difference in the processing time of Sobel operation. In contrast to the average transfer time between nodes in tables 6 and 7, they are 30.2705 (s) and 29.67875 (s). the difference in data transfer is only 0.59175 (s).

Table 8: Edge Detection Processing Time (S) on Master Node 1 for a grayscale image using a 100Mbps switch

Node	Transfer	Switch 100Mbps							
		Time (second)							
		Sobel Operation						Combine	Final
		Master node 1	node1	node2	node3	node4	All		
1	22,188	10,78	12,40	-	-	-	23,196	25,163	99,579
2	29,496	7,345	6,589	6,572	-	-	20,508	29,257	106,191
3	33,086	5,407	4,932	4,933	4,933	-	20,206	32,833	110,633
4	35,057	4,295	3,945	3,943	3,943	3,944	20,078	35,104	111,955

In table 8, the time required by master node 1 to use a 100Mbps switch is 111,955. Whereas in table 9, the time required by master node 1 when using a 1000Mbps switch is 115,795, so there is an increase in the single detection processing time between tables 8 and 9 by 3.32%. This is due to the use of a 1000Mbps switch which can increase data transfer. As well as the ability to use the auto-negotiation feature on the NIC (network interface card) on the master node 1. This feature is able to analyze the best data transfer speed based on its environment.

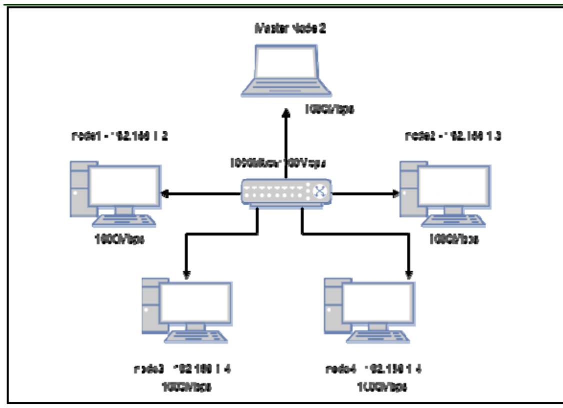
Table 9: Edge Detection Processing Time (S) on Master Node 1 for the grayscale image using 1000Mbps switch

Node	Transfer	Switch 1000Mbps							
		Time (second)						Combine	Final
		Sobel Operation							
		Master node 1	node1	node2	node3	node4	All		
1	21,953	10,72	14,38	-	-	-	25,112	26,603	101,001
2	29,160	7,190	6,584	6,574	-	-	20,349	29,044	106,948
3	32,762	5,381	4,930	4,933	4,936	-	20,182	32,652	112,417
4	34,908	4,310	3,942	3,946	3,948	3,948	20,096	34,816	115,795

The average time to complete the Sobel operation in table 8 is 107.0895 (s), while in table 9 it is 109.04025 (s). the difference is only 1.95075 (s) of the Sobel operation processing time difference. In contrast to the average transfer time between nodes in tables 8 and 9, namely 29.95675 (s) and 29.69575 (s). the difference in data transfer is only 0.261 (s).

Journal of Theoretical and Applied Information Technology
31st January 2021. Vol.99. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



In figure 4, we are using a 1000Mbps switch with a 1000Mbps cable and a NIC on a 1000 Mbps master node. Each node uses a 1000Mbps NIC. Star topology is used in HPC. IP address used 192.168.1.1 - 192.168.1.5. The difference in topology in Figures 4 and 5 is the NIC and the switch, namely 1000Mbps.

Table 10: Edge Detection Processing Time (S) on Master Node 2 for the original image using 100Mbps switch

Node	Transfer	Switch 100Mbps							
		Time (second)						Combine	Final
		Sobel Operation							
		Master node 2	node1	node2	node3	node4	All		
1	22,659	9,895	12,71	-	-	-	22,607	25,755	128,752
2	29,663	6,618	6,549	6,636	-	-	19,803	29,275	135,294
3	33,585	4,954	4,935	4,938	4,935	-	19,763	32,874	209,621
4	35,194	3,955	3,949	3,949	3,949	3,946	19,751	35,035	147,180

In table 10, the time required by master node 1 to use a 100Mbps switch is 147.18. Whereas in table 11, the time required by master node 1 when using a 1000Mbps switch is 81.917. So that there was an increase in the processing time of single detection between tables 10 and 11 by 44.34%. This is due to the use of a 1000Mbps switch which can increase data transfer. As well as the ability to use the auto-negotiation feature on the NIC (network interface card) on the master node 2. This feature is able to analyze the best data transfer speed based on the environment.

Table 11: Edge Detection Processing Time (S) on Master Node 2 for the original image using 1000Mbps switch

Node	Transfer	Switch 1000Mbps							
		Time (second)						combine	Final
		Sobel Operation							
		Master node 2	node1	node2	node3	node4	All		
1	2,478	9,920	13,34	-	-	-	23,262	7,246	90,247
2	3,095	6,590	6,555	6,654	-	-	19,801	3,034	83,115
3	3,409	4,949	4,937	4,935	4,970	-	19,792	3,363	82,403
4	3,593	3,955	3,948	3,945	3,948	3,948	19,744	3,563	81,917

The average time to complete the Sobel operation in table 10 is 155.21175 (s), while in table 11 it is 84.4205 (s). the only difference is 70.79125 (s) difference in the processing time of Sobel operation. In contrast to the average transfer time between nodes in tables 10 and 11, namely 30.27525 (s) and 3.14375 (s). the difference in data transfer is only 27.1315 (s).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 12: Edge Detection Processing Time (S) on Master Node 2 for a cropped image using 100Mbps switch

Node	Transfer	Switch 100Mbps							
		Time (second)						Combine	Final
		Sobel Operation							
		Master node 2	node1	node2	node3	node4	All		
1	19,370	8,621	8,599	-	-	-	17,221	19,417	83,218
2	25,737	5,735	5,738	5,752	-	-	17,225	25,610	92,852
3	28,805	4,321	4,313	4,315	4,314	-	17,265	28,771	96,853
4	30,705	3,464	3,461	3,453	3,452	3,463	17,297	30,174	99,174

In table 12, the time required by master node 1 to use a 100Mbps switch is 99.174. Whereas in table 13, the time required by master node 1 when using a 1000Mbps switch is 46.715. So that there was an increase in the processing time of single detection between tables 12 and 13 as much as 52.90%. This is due to the use of a 1000Mbps switch which can increase data transfer. As well as the ability to use the auto-negotiation feature on the NIC (network interface card) on the master node 1. This feature is able to analyze the best data transfer speed based on its environment.

Table 13: Edge Detection Processing Time (S) on Master Node 2 for the cropped image using 1000 Mbps switch

Node	Transfer	Switch 1000Mbps							
		Time (second)						combine	Final
		Sobel Operation							
		Master node 2	node1	node2	node3	node4	All		
1	2,934	8,609	12,47	-	-	-	21,089	6,601	50,986
2	3,337	5,735	5,737	5,790	-	-	17,261	2,669	45,425
3	3,520	4,357	4,329	4,323	4,326	-	17,327	2,953	56,950
4	3,648	3,432	3,453	3,455	3,448	3,457	17,247	3,124	46,715

The average completion time of the Sobel operation in table 12 is 93.02425 (s), while table 13 is 50.019 (s). only the difference is 43.00525 (s) the difference in the processing time of the Sobel operation. In contrast to the average transfer time between nodes in tables 12 and 13, namely 26.15425 (s) and 3.35975 (s). the difference in data transfer is only 22.7945 (s).

Table 14: Edge Detection Processing Time (S) on Master Node 2 for the rotate image using 100Mbps switch

Node	Transfer	Switch 100Mbps							
		Time (second)						Combine	Final
		Sobel Operation							
		Master node 2	node1	node2	node3	node4	All		
1	22,175	9,878	13,55	-	-	-	24,678	27,909	130,386
2	29,315	6,603	6,572	6,577	-	-	19,783	29,265	136,438
3	32,911	4,973	4,933	4,938	4,935	-	19,782	32,885	141,676
4	35,100	3,974	3,944	3,947	3,949	3,949	19,765	35,042	144,157

In table 14, the time required by master node 1 using a 100Mbps switch is 144.157 (s). Whereas in table 15, the time required by master node 1 when using a 1000Mbps switch is 82.755 (s). So there is an increase in the single detection processing time between tables 14 and 15 by 42.59%. This is due to the use of a 1000Mbps switch which can increase data transfer. As well as the ability to use the auto-negotiation

Journal of Theoretical and Applied Information Technology
31st January 2021. Vol.99. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

feature on the NIC (network interface card) on the master node 1. This feature is able to analyze the best data transfer speed based on its environment.

Table 15: Edge Detection Processing Time (S) on Master Node 2 for the rotate image using 1000 Mbps switch

Node	Transfer	Switch 1000Mbps							
		Time (second)						combine	Final
		Sobel Operation							
		Master node 2	node1	node2	node3	node4	All		
1	2,981	9,910	11,86	-	-	-	21,775	5,770	88,514
2	3,594	6,569	6,549	6,609	-	-	19,729	3,063	83,421
3	3,684	4,938	4,933	4,939	4,937	-	19,748	3,385	82,110
4	4,071	3,963	3,944	3,950	3,950	3,946	19,755	3,585	82,755

The average completion time of the Sobel operation in table 14 is 138.16425 (s), while table 15 is 84.2 (s). only 53.96425 (s) difference in the processing time of the Sobel operation. In contrast to the average transfer time between nodes in tables 14 and 15, namely 29.87525 (s) and 3.5825 (s). the difference in data transfer is only 26.29275 (s).

Table 16: Edge Detection Processing Time (S) on Master Node 2 for the grayscale image using 100Mbps switch

Node	Transfer	Switch 100Mbps							
		Time (second)						Combine	Final
		Sobel Operation							
		Master node 2	node1	node2	node3	node4	All		
1	22,124	9,878	13,55	-	-	-	23,434	26,689	99,612
2	29,332	6,603	6,572	6,577	-	-	19,754	29,276	105,712
3	32,950	4,950	4,932	4,935	4,955	-	19,773	32,885	111,225
4	35,094	3,969	3,946	3,946	3,947	3,952	19,761	35,049	114,526

In table 16, the time required by master node 1 to use a 100Mbps switch is 114.526 (s). Whereas in table 17, the time required by master node 1 when using a 1000Mbps switch is 52.104 (s). So there is an increase in the single detection processing time between tables 16 and 17 by 54.50%. This is due to the use of a 1000Mbps switch which can increase data transfer. As well as the ability to use the auto-negotiation feature on the NIC (network interface card) on the master node 1. This feature is able to analyze the best data transfer speed based on its environment.

Table 17: Edge Detection Processing Time (S) on Master Node 2 for the grayscale image using 1000 Mbps switch

Node	Transfer	Switch 1000Mbps							
		Time (second)						combine	Final
		Sobel Operation							
		Master node 2	node1	node2	node3	node4	All		
1	3,013	9,904	12,09	-	-	-	21,997	5,448	58,320
2	3,715	6,586	6,575	6,589	-	-	19,751	3,066	53,432
3	3,771	4,952	4,934	4,936	4,938	-	19,762	3,400	57,708
4	3,912	4,000	3,948	3,947	3,948	3,950	19,795	3,580	52,104

Journal of Theoretical and Applied Information Technology
31st January 2021. Vol.99. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The average completion time of Sobel operations in table 16 is 107.76875 (s), while table 17 is 55.391 (s). only 52.37775 (s) difference in the processing time of the Sobel operation. In contrast to the average transfer time between nodes in tables 16 and 17, namely 29.875 (s) and 3.60275 (s). the difference in data transfer is only 26.27225 (s).

Table 18: Edge Detection Processing Time (S) on Master Node 1 for the original image

Image		Master Node 1 (100Mbps)	
		100Mbps Switch	1000Mbps Switch
Original	1	153,611	150,144
	2	56,381	55,903
	3	103,477	101,443
	4	54,183	53,423
	5	64,678	64,428

In table 18, it can be seen that the time speed increases using 100Mbps and 1000Mbps for each image. In the first image, there is a decrease in time by 1.95%, and the second image is 0.89%, the third image is 1.97%, the fourth image is 1.40%, and the fifth image is 0.39% from the use of 100Mbps to 1000Mbps switches.

Comparison of the overall time speed generated using the master node1 using the 1000Mbps, and 100Mbps switches on each original image were 1.51% faster using a 1000Mbps switch.

Table 19: Edge Detection Processing Time (S) on Master Node 1 for the cropped image

Image		Master Node 1 (100Mbps)	
		100Mbps Switch	1000Mbps Switch
Crop the Original Image	1	102,767	96,164
	2	38,952	37,734
	3	64,623	63,927
	4	24,39	29,529
	5	48,272	48,451

In Table 19, it can be seen that the time speed increases using 100Mbps and 1000Mbps for each image. In the first image, there is a decrease in time by 6.43%, and the second image is 3.13%, the third image is 1.08%, the fourth image is 2.19%, and the fifth image is 3.31% from the use of 100Mbps to 1000Mbps switches.

Comparison of the overall time speed generated using master node1 using a 1000Mbps switch and 100Mbps on each image cropping is 3.78% faster using a 1000Mbps switch.

Table 20: Edge Detection Processing Time (S) on Master Node 1 for rotate image

Image		Master Node 1 (100Mbps)	
		100Mbps Switch	1000Mbps Switch
Rotate the Original Image	1	207,798	146,729
	2	50,011	59,198
	3	95,367	97,348
	4	50,359	48,983
	5	61,834	60,767

In table 20, it can be seen that the time speed increases using 100Mbps and 1000Mbps for each image. In the first picture, there is a time decrease of 29.39%, and the second image has a time increase of 15.52%, the third image has a time increase of 2.03%, the fourth image has a decrease in time by 3.12% and the fifth image has a decrease in time 1.73% of the 100Mbps to 1000Mbps switch usage.

Comparison of the overall time speed generated using master node1 using a 1000Mbps switch, and 100Mbps on each rotate image is 11.29% faster using a 1000Mbps switch.

Table 21: Edge Detection Processing Time (S) on Master Node 1 for grayscale image

Image		Master Node 1 (100Mbps)	
		100Mbps Switch	1000Mbps Switch
Grayscale the Original Image	1	111,955	115,795
	2	49,573	49,642
	3	82,747	81,643
	4	39,874	40,311
	5	54,438	54,661

In table 21, it can be seen the increase in time speed using 100Mbps and 1000Mbps for each image. In the first picture, there is a time increase of 3.32%, and the second image is 0.14%, the third image has a time decrease of 1.33%, the fourth image has a time increase of 1.08% and the fifth image is 0.41% from the use of switches 100Mbps to 1000Mbps.

Comparison of the overall time speed generated using the master node1 using the 1000Mbps, and 100Mbps switches on each grayscale image are 1.01% faster using a 100Mbps switch.

Table 22: Edge Detection Processing Time (S) on Master Node 2 for the original image

Image		Master Node 2 (1000Mbps)	
		100Mbps Switch	1000Mbps Switch
Original	1	147.180	81.917
	2	54.032	18.250
	3	98.026	44.012
	4	48.530	24.233
	5	61.950	24.497

In table 22, it can be seen that the time speed increases using 100Mbps and 1000Mbps for each image. In the first image, there is a time decrease of 44.34%, and the second image is 66.22%, the third image is 55.10%, the fourth image is 50.07%, and the fifth image is 60.46% from the use of 100Mbps to 1000Mbps switches.

Comparison of the overall time speed generated using the master node2 using a 1000Mbps switch and 100Mbps on each original image is 52.92% faster using a 100Mbps switch.

Table 23: Edge Detection Processing Time (S) on Master Node 2 for the cropped image

Image		Master Node 2 (1000Mbps)	
		100Mbps Switch	1000Mbps Switch
Crop the Original Image	1	99.174	46.715
	2	41.144	13.737
	3	63.897	30.168
	4	32.132	9.502
	5	49.114	19.634

In table 23, it can be seen the increase in time speed using 100Mbps and 1000Mbps for each image. In the first image, there is a time decrease of 52.79%, the second image is 66.61%, the third image is 70.43%, the fourth image is 1.40%, and the fifth image is 60.02% from the use of 100Mbps to 1000Mbps switches.

Comparison of the overall time speed generated using master node2 using a 1000Mbps switch and 100Mbps on each image cropping is 58.05% faster using a 100Mbps switch.

Table 24: Edge Detection Processing Time (S) on Master Node 2 for rotate image

Image		Master Node 2 (1000Mbps)	
		100Mbps Switch	1000Mbps Switch
Rotate the Original Image	1	144.157	82.755
	2	54.432	18.485
	3	95.345	41.803
	4	48.564	26.915
	5	68.832	25.644

In table 24, you can see the increase in time speed using 100Mbps and 1000Mbps for each image. In the first image, there is a time decrease of 42.59%, the second image is 66.04%, the third image is 56.16%, the fourth image is 44.58%, and the fifth image is 62.74% from the use of 100Mbps to 1000Mbps switches.

Comparison of the overall time speed generated using master node2 using a 1000Mbps switch, and 100Mbps on each rotate image is 52.54% faster using a 100Mbps switch.

Table 25: Edge Detection Processing Time (S) on Master Node 2 for the grayscale image

Image		Master Node 2 (1000Mbps)	
		100Mbps Switch	1000Mbps Switch
Grayscale the Original Image	1	114.526	52.104
	2	53.536	17.800
	3	84.344	33.881
	4	41.637	16.508
	5	55.250	18.816

In table 25, it can be seen that the time speed increases using 100Mbps and 1000Mbps for each image. In the first image, there is a time decrease of 54.50%, the second image is 66.75%, the third image is 59.83%, the fourth image is 60.35%, and the fifth image is 69.94% from the use of 100Mbps to 1000Mbps switches.

Comparison of the overall time speed generated using master node2 using 1000Mbps, and 100Mbps switches on each grayscale image are 60.17% faster using 1000Mbps switches.

4.1 High-Performance Computing Cluster

We use Linpack tools to assess the performance of the PC-Cluster we have built.

Journal of Theoretical and Applied Information Technology
31st January 2021. Vol.99. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Linpack can measure the compute of a cluster. Performance in question is the number of millions of floating-point operations per second measured in megaflops (Mflop s-1). In the context of the Linpack benchmark using gigaflops (Gflop s-1) as the number of billions of floating-point operations per second. The following are the parameters used on the Linpack:

- *N Parameter*

The parameter N, which shows the value of the number of problems to be tested on the cluster. The value of N is useful for knowing how much performance a computer has. For the selection of N values, use 80% of the total memory available. In this study, the amount of memory in each node is 512 MB, so 512 MB x 4 nodes, which is 2 GB, then we can find the amount of N used in the study using Eq. (3).

$$N = \sqrt{\frac{\text{Sum of RAM size each Nodes (GB)} * 1024^3}{8}} * 0.8 \quad (3)$$

- *NB Parameter*

The NB parameter, which shows the block size value used for data distribution. Usually, the block sizes give good results. The recommended ranges are [96, 104, 112, 120, 128,…, 256]. The NB value used in this research will be carried out one by one using the same N value. The highest Gflop value is at block size 176.

- *P and Q Parameter*

Parameters P and Q, which indicate the value of the number of cores that each node has on. The P value should be less than the Q value. After getting the value, then find the factors so that we get the factor values of the number. Choose the closest factor number and the value of P <Q. In this study, the number of nodes used is four and each node has 2 CPU cores on each node, so the factors for the P and Q values used are 1 x 8 and 2 x 4

Table 26: Cluster maximum performance with an additional number of nodes

N	NB	P	Q	Time (s)	Gflops	Node
13107	176	2	4	410.53	3.657E+00	4
11350	176	1	6	331.68	2.940E+00	3
9268	176	1	4	227.2	2.336E+00	2
6553	176	1	2	153.52	1.222E+00	1

Testing the maximum performance of the cluster can be seen from the results of the test, namely with the N value increasing as the number of nodes used increases, it can be seen that the resulting Gflops value increases as the number of nodes increases. The Gflops value increases when using four nodes and produces the highest Gflops value, namely 3.66E + 00.

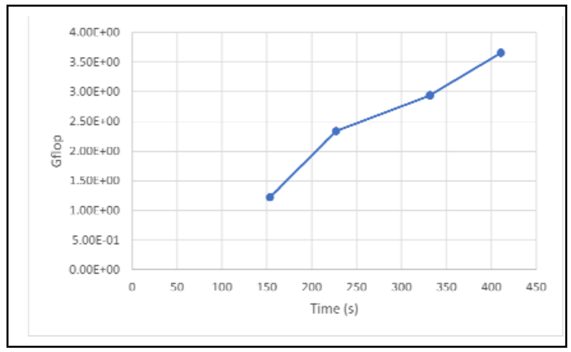


Figure 5: Value of Cluster Performance Gflops with Increase in Number Nodes

After testing the cluster performance, the next step is to test the addition of the number of nodes as in table 15.

Table 27: Cluster performance gflops value with an additional number of nodes

N	NB	P	Q	Time (s)	Gflops	Node
13107	176	2	4	410.53	3.657E+00	4
13107	176	1	6	1375.18	1.092E+00	3
13107	176	1	4	4572.32	3.284E-00	2
13107	176	1	2	-	-	1

In Figure 7, it can be seen that on N 13107, the cluster performance is achieved when the nodes are 4 in 410.53 seconds. In contrast, one node cannot measure the performance of this cluster because one node cannot run N of 130107 so that when using one node does not get results.

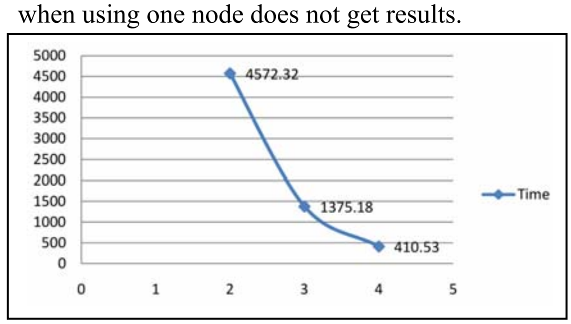


Figure 6: Relationship number of nodes with PC-cluster performance

Journal of Theoretical and Applied Information Technology
31st January 2021. Vol.99. No 2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The following is a comparison between 100Mbps switches and 1000Mbps switches on master node1.

Table 28: Edge Detection Processing Time (S) on Master Node 1 for the original image

Image	Switch	
	100Mbps	**1000Mbps**
original	86,466	85,162
Crop	56,961	54,805
Rotate	93,144	82,603
Grayscale	67,717	68,41
Average time(s)	76,07	72,75

Based on table 28, it can be seen that the comparison of the overall time speed generated using the master node1 using a 1000Mbps and 100Mbps switches on each of the overall images is 4.36% faster using a 1000Mbps switch.

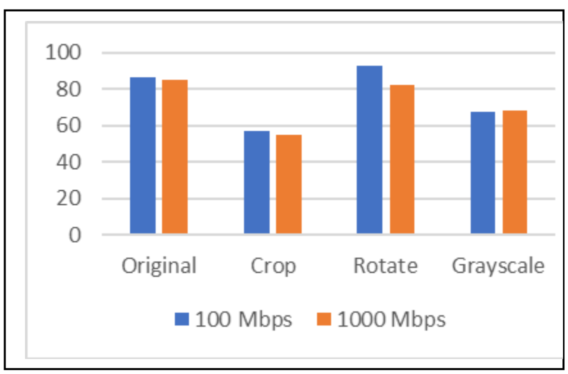


Figure 7: Comparison between 100Mbps switch and 1000Mbps switch on master node1

The following is a comparison between 100Mbps switches and 1000Mbps switches on master node2.

Table 29: Edge Detection Processing Time (S) on Master Node 1 for the original image

Image	Switch	
	100Mbps	**1000Mbps**
Original	81,944	38,582
Crop	57,092	23,951
Rotate	82,266	39,12
Grayscale	69,859	27,822
Average time(s)	72,79	32,37

Based on table 29, it can be seen that the comparison of the overall time speed generated using the master node2 using a 1000Mbps and 100Mbps switches on each of the overall images reaches 55.53% faster using a 1000Mbps switch.

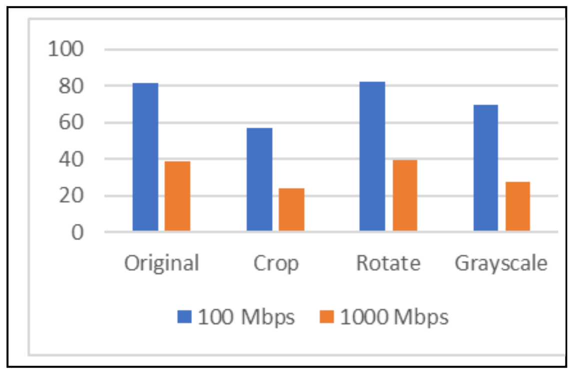


Figure 8: Comparison between 100Mbps switch and 1000Mbps switch on master node2

To the best of our knowledge, no one has performed edge detection testing on images using PC-Cluster. Only [13] used PC-Cluster to perform clustering of MRI objects. The PC-Cluster is built to reduce computation time. This is proven by the collaboration of low computer resources that can produce extraordinary performance.

The PC-Cluster we build is limited to computer resources, if we want to increase performance, we recommend using a 1000Mbps switch and upgrading the PC-Cluster such as RAM and Processor.

5. CONCLUSION

The limitation of computational resources causes researchers to be constrained to get better research results. We offer a computing concept with limited resources, PC-Cluster. We use this concept to test the performance and timing of single methods for doing edge detection on large-scale images. PC-Cluster can improve the best computation when using Gigabit Ethernet interface against 1000Mbps switch. The PC-Cluster processing speed on a 1000Mbps switch is 55.53%

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

faster than a 100Mbps switch. At the same time, the maximum performance of the PC-Cluster in the experiment was 3.657E + 00.

REFERENCES:

[1] R. Dhivya and R. Prakash, "Edge detection of satellite image using fuzzy logic," *Cluster Comput.*, vol. 22, no. November, pp. 11891–11898, 2019, doi: 10.1007/s10586-017-1508-x.

[2] A. Ali, "Coarse Classification of Terrain Image Information by Using Sobel Edge Detection Technique," *Int. J. Comput. Sci. Netw. Secur.*, vol. 18, no. 1, pp. 149–154, 2018.

[3] X. Wang, Y. Fang, C. Li, S. Gong, L. Yu, and S. Fei, "Static gesture segmentation technique based on improved Sobel operator," *J. Eng.*, vol. 2019, no. 22, pp. 8339–8342, Nov. 2019, doi: 10.1049/joe.2019.1075.

[4] P. Sikka, A. R. Asati, and C. Shekhar, "High-speed and area-efficient Sobel edge detector on field-programmable gate array for artificial intelligence and machine learning applications," *Comput. Intell.*, no. April, pp. 1–12, 2020, doi: 10.1111/coin.12334.

[5] R. G. Zhou and D. Q. Liu, "Quantum Image Edge Extraction Based on Improved Sobel Operator," *Int. J. Theor. Phys.*, vol. 58, no. 9, pp. 2969–2985, 2019, doi: 10.1007/s10773-019-04177-6.

[6] G. N. Chaple, R. D. Daruwala, and M. S. Gofane, "Comparisions of Robert, Prewitt, Sobel operator based edge detection methods for real time uses on FPGA," *Proc. - Int. Conf. Technol. Sustain. Dev. ICTSD 2015*, no. 1, pp. 4–7, 2015, doi: 10.1109/ICTSD.2015.7095920.

[7] P. Fan, R.-G. Zhou, W. Hu, and N. Jing, "Quantum image edge extraction based on classical Sobel operator for NEQR," *Quantum Inf. Process.*, vol. 18, no. 1, p. 24, Jan. 2019, doi: 10.1007/s11128-018-2131-3.

[8] E. Xie, "Research and implementation of Sobel edge detection using model-based design," *Int. J. Secur. its Appl.*, vol. 10, no. 7, pp. 345–354, 2016, doi: 10.14257/ijsia.2016.10.7.30.

[9] C. A. S. J. Gulo, A. C. Sementille, and J. M. R. S. Tavares, "Techniques of medical image processing and analysis accelerated by high-performance computing: a systematic literature review," *J. Real-Time Image Process.*, vol. 16, no. 6, pp. 1891–1908, 2019, doi: 10.1007/s11554-017-0734-z.

[10] P. H. Tournier *et al.*, "Microwave tomographic imaging of cerebrovascular accidents by using high-performance computing," *Parallel Comput.*, vol. 85, pp. 88–97, 2019, doi: 10.1016/j.parco.2019.02.004.

[11] L. Domanski, T. Bednarz, T. E. Gureyev, L. Murray, E. Huang, and J. A. Taylor, "Applications of Heterogeneous Computing in Computational and Simulation Science," in *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, Dec. 2011, pp. 382–389, doi: 10.1109/UCC.2011.64.

[12] V. S. Kumar *et al.*, "Large-Scale Biomedical Image Analysis in Grid Environments," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 2, pp. 154–161, Mar. 2008, doi: 10.1109/TITB.2007.908466.

[13] J. YEH and J. FU, "Parallel adaptive simulated annealing for computer-aided measurement in functional MRI analysis," *Expert Syst. Appl.*, vol. 33, no. 3, pp. 706–715, Oct. 2007, doi: 10.1016/j.eswa.2006.06.018.