Asian Journal of Information Technology 9 (5): 263-271, 2010
ISSN: 1682-3915

Design and Simulation of Indonesian Education Grid Topology Using Gridsim Toolkit

Ivo Bahar Nugroho and Heru Suhartanto
Faculty of Computer Science, University of Indonesia, Jakarta, Indonesia

Abstract: This study discusses the design and simulation of an e-learning computer network topology based on Grid computing technology for Indonesian schools called the Indonesian Education Grid (abbreviated as IndoEdu-Grid). The usage of such a grid has its advantages such as supporting the use of heterogeneous resources (hardware or software) and resources sharing. The establishment of such network without Grid computing capabilities will lead to redundancies of the idle resources. The simulation which is built using GridSim toolkit will handle the proposed two conditions or scenarios that have different network topologies based on their routers and links configuration. Each scenario will be run in the simulator using two packet scheduling algorithms, one will be FIFO (First In First Out) Scheduler and the other SCFQ (Self-Clocked Fair Queuing) Scheduler. The processing time of the job's packets will be evaluated to determine the most effective network topology for IndoEdu-Grid. The simulation result showed that if SCFQ scheduling algorithm is used then the most appropriate network topology is the topology which allows its packets with the same priorities to have less possibility to collide one another in one router or link which is the first scenario. The simulation results also showed that SCFQ scheduling algorithm can reduce the packets lifetime at routers that have very crowded traffics. This fact implies to the decrease of the whole job processing time.

Key words: Discrete-event simulation, FIFO, Grid computing, GridSim toolkit, IndoEdu-Grid, network topology, SCFQ

INTRODUCTION

E-learning as a trend has been developed so rapidly that many educational organizations and institutions in numerous countries including Indonesia adopts it. An example of an infrastructure that can be used for e-learning in Indonesia is Jardiknas (Jejaring Pendidikan Nasional or Indonesian ICT Network). Jardiknas is a national-scaled WAN (Wide Area Network) that facilitates educational activities in Indonesia. This network consists of Institutional/Official Jardiknas that serves online data transaction between educational institutions, College Jardiknas-Indonesian Higher Education Networks (INHERENT) that serves science and technology research and development, School Jardiknas that serves information and e-learning accesses in schools and Teacher and Student Jardiknas that serves personal information and e-learning accesses (Pustekkom, 2009). Jardiknas is established with a main purpose to serve administration in central National Department of Education (Departemen Pendidikan Nasional) and many domestic or foreign related work units and to serves learning processes in primary, junior high and senior high schools based on information and communication technology. Jardiknas covers the whole areas of Indonesia but it still has some problems as follows:

- The current network has not been equipped with capabilities to facilitate huge data processing and cannot maximize the distributed potential resource in the whole areas of Indonesia
- The number of students and educational institutions that require accesses to Jardiknas will increase every year so that the needs of wider and easier to access network have risen
- Educational activities are advancing where the education subjects (teachers, students and instructors) will interact with each other, share data and perform complex calculations and simulations such as mathematics, physics or biomolecular models

To solve these problems, we propose the usage of Grid computing technology for Indonesian Education Networks called IndoEdu-Grid. Grid computing (or Grid) is a system that can provide resources sharing among organizations. Grid infrastructure will provide a capability to connect the resources dynamically as an ensemble to support large-scale, resource-intensive and distributed applications (Berman *et al*., 2003).

IndoEdu-Grid itself is a national-scaled Grid network used for educational purposes in Indonesia especially for high schools and universities. This network consists of ten resources with many processors located in each

Corresponding Author: Ivo Bahar Nugroho, Faculty of Computer Science, University of Indonesia, Jakarta, Indonesia

province that connect to each other. However, building a suitable and reliable national Grid infrastructure is very expensive, thus this encourages us to study some possible topologies and perform some experiments on their performances.

It is impossible to perform such an experiment using the trial and error method because it will be very expensive and time-consuming. In addition, we have to consider both political and technical policies are considered in each province in Indonesia.

Thus, prior to the implementation, the experiments must be designed, done and analyzed using a simulator that can be configured and adjusted to THE needs. This simulator has to mimic the real system and predict system's behavior. In this study we use GridSim toolkit which is a Java-based simulator and supported with some additional libraries.

GridSim toolkit (or GridSim) is a toolkit developed with Java and is used to build discrete-event simulations of Grid systems. GridSim is an open-source application and licensed under GPL license, thus it encloses its source codes in its distribution package. GridSim's rationale is that creating a testbed infrastructure for Grid system is expensive and time-consuming, even an existing testbed infrastructure is also limited in size to a few resources and domains and testing scheduling algorithms for scalability and adaptability and evaluating scheduler performance for various applications and resource scenarios is harder and impossible to trace (Buyya and Murshed, 2000, 2002; Sulistio *et al*., 2005).

MATERIALS AND METHODS

We design the IndoEdu-Grid, it consists of entities such as resources, users and jobs or Gridlets; class diagram and topological scenarios such as:

Resources: Resource entities are responsible to perform computation on job entities in form of Gridlets sent by one or more users and send it back to the user.

The research uses one resource for each province; each resource consists of one Machine and each Machine consists of 4 PEs (processing elements). These Machines have identical processing speed (300 MIPS), operating systems and architectures (Debian Linux on Intel architecture).

To reduce the level of heterogeneity and make the understanding of system behavior easier, all provinces will use the same resource characteristics. The details of name and location of resources and the amount of each PE for the entire province are shown in Table 1.

Table 1: Resources for each Province

Resources name	Resources location	Number of PEs
NAD_Res	Nanggroe Aceh Darussalam	4
Sumut_Res	North Sumatera (Sumatera Utara)	4
Sumbar_Res	West Sumatera (Sumatera Barat)	4
Riau_Res	Riau	4
Kepriau_Res	Riau Archipelago (Kepulauan Riau)	4
Jambi_Res	Jambi	4
Sumsel_Res	South Sumatera (Sumatera Selatan)	4
Babel_Res	Bangka-Belitung	4
Bengkulu_Res	Bengkulu	4
Lampung_Res	Lampung	4
Banten_Res	Banten	4
DKI_Res	Special Capital District of Jakarta (Daerah Khusus Ibukota Jakarta)	4
Jabar_Res	West Java (Jawa Barat)	4
Jateng_Res	Central Java (Jawa Tengah)	4
DIY_Res	Special District of Yogyakarta (Daerah Istimewa Yogyakarta)	4
Jatim_Res	East Java (Jawa Timur)	4
Bali_Res	Bali	4
NTB_Res	Western Lesser Sundas (Nusa Tenggara Barat)	4
NTT_Res	East Lesser Sundas (Nusa Tenggara Timur)	4
Kalbar_Res	West Borneo (Kalimantan Barat)	4
Kalteng_Res	Central Borneo (Kalimantan Tengah)	4
Kaltim_Res	East Borneo (Kalimantan Timur)	4
Kalsel_Res	South Borneo (Kalimantan Selatan)	4
Sulut_Res	North Sulawesi (Sulawesi Utara)	4
Gorontalo_Res	Gorontalo	4
Sulteng_Res	Central Sulawesi (Sulawesi Tengah)	4
Sultra_Res	Southeast Sulawesi (Sulawesi Tenggara)	4
Sulsel_Res	South Sulawesi (Sulawesi Selatan)	4
Sulbar_Res	West Sulawesi (Sulawesi Barat)	4
Maluku_Res	Moluccas and North Moluccas (Maluku dan Maluku Utara)	4
Papua_Res	Papua	4

Users: Users are entities responsible to submit jobs in form of Gridlet objects to the resources. In this research, the users are programmed to send jobs to a particular resource at the same time, thus we are able to gain more knowledge on the performance of Grid system in its peak load when all the users are accessing the resource at the same time.

The selection of resource is done randomly, for example the users located in Banten may choose the resource located in Lampung and send jobs to that resource.

The number of users in each province is adjusted by the real conditions according to the number of high school students taken from the data published by National Department of Education. The most recent data published by the National Department of Education is the data in 2007/2008 academic year.

Therefore, the research used the number of high school students in 2007/2008 academic year. The number of users are adjusted to fit with the simulation. This adjustment is performed by finding the actual ratio of the number of students in each province to the number of students in Special Capital District of Jakarta. This

adjustment is made to prevent the occurrence of OutOfMemory runtime error in Java. The real total number of all students is 15, 112, 161 so we think that this adjustment is important. The ratio is then applied to the simulation so that the number of users in each province can be obtained. The obtained number of users after adjusted is 958.

Jobs (Gridlets): Jobs in GridSim are represented as the objects of the class Gridlet provided by GridSim. In the research, each user will create three Gridlets having different lengths-5000 MI (millions instructions), 3000 MI and 1000 MI. This was aimed to simulate the real situation where a user does not just send one job but it can also send more than one job with different sizes and needs of computation powers.

A user will send the three Gridlets successively to a resource that has been chosen randomly. Furthermore, the resource will send an ACK (acknowledge) signal as a sign that the Gridlet been received and ready to be processed. Resources will process the Gridlet and send it back to the user.

The class diagram: The class diagram is shown in Fig. 1. This simulation consists of four main classes, namely class Main, class Islands, class NetUser and class Randomizer and several other supporting classes available in Java's default library and the GridSim's additional library such as ArrayList, Router, GridSim, GridResource and GridletList. The function of the classes are as follows:Main class is the class containing the Main method that will run the simulation and the printGridletList method that will print the processed Gridlets. In this class, the construction of the network topology for each scenario will be done with the following stages:

- Create lists containing resources for each island or region (sub-resource list) and for all resources (main resource list). ArrayList is used because it consumes less memory resources on a computer running the simulations and more flexible for use as storage for GridResource objects
- Call createGridResourcesList method to create GridResource objects in the class Islands. This method has a parameter of the island's name whose resources will be created and a return value of a list of resources of that island. The created GridResource objects are then added to the main list (main resource list). For Moluccas and Papua regions, createGridResource method returns the GridResource objects directly without adding them to the sub-resource list. Moluccas and Papua regions receive this special treatment because we think that there is no need to make some resources to these areas. Geographically, Moluccas can be regarded as a provincial area as well as with Papua. In addition, users in Papua are not too many for the size of an island, so it would be better if the region is considered as a whole
- Create lists to store the objects of user and then create objects of user as many as the total number of users that has been determined previously. These objects of user are the objects of the class NetUser that represents the users in Grid systems

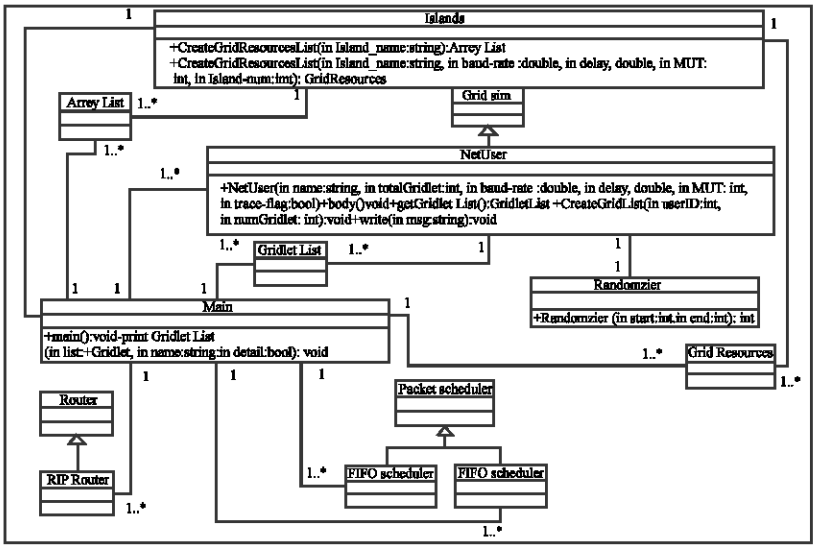

Fig. 1: The class diagram for simulation program

- Create a router for each island (called leaf routers) and their supporting routers such as routers connecting each island with the main network (called edge routers) and routers for the Western, Central and Eastern Indonesian territories (called core routers). Leaf routers are routers at the lowest level that serve as the first routers that will forward packets to and from users and resources. At higher levels, edge routers and core routers are responsible for managing the data traffic to and from more distant areas which can be a different province, island or region (Western, Central and East Indonesia)
- Connect each user and resource with the appropriate routers, e.g., users and resources in South Sumatra are connected to a router in Sumatra. This can be done using attachHost method that has parameters of the entity that will be connected to the router and scheduler that serves as the scheduling policy
- Connect all routers into one large network. This process is divided into three types according to the simulation scenarios that will be done as follows
- Involving leaf routers, edge routers and core routers. This process can be considered as forming a hierarchical network where core routers are in the highest position and leaf routers are in the lowest position. A core router consists of some edge routers and an edge router consists of some leaf routers as shown in Fig. 2a
- Involving leaf routers and edge routers. In this process, core routers are not used. Their role was replaced directly by the edge routers. Figure 2b shows this type of process. Dashed line is a proponent network (called backbone link)
- Involving only the leaf routers. In this connection, leaf routers will be connected each other directly. Figure 2c shows this type of process. Dashed line is a proponent network (called backbone link)

Islands class is a class that contains methods for creating GridResource objects. In this class, there are two methods, they are createGridResourcesList and createGridResource. NetUser class is a class that represents the users in Grid system. In this class, the behavior of the objects of user in the Grid system is set through the body method. Randomizer class is a class that functions to do randomization between two integers. This class has only one method, Randomize method which has two integer parameters which serve as initial boundary and final boundary for the random numbers.

GridSim class is a class responsible for initializing, running and stopping the whole process of simulation. In this class, there are important methods such as init to initialize GridSim, startGridSimulation to run simulations and stopGridSimulation to stop the simulation process. In addition there are also the methods for logging the statistics of simulation obtaining the status of Gridlets, stopping the simulation process for a while (pausing the simulation), sending (submitting) the Gridlets to specific resources and obtaining the ID of the entities. GridletList class is a class responsible for storing some Gridlets into a list. This class extends LinkedList class.

GridResource class is a class responsible for simulating resources in Grid systems. The properties of GridResource are defined in the objects from ResourceCharacteristics class. Router class is a class that simulates routers on the network.

This class is extended by some classes that implement different routing method such as FloodingRouter, FlowRouter, FnbRIPRouter, RIPRouter and RateControlledRouter classes. RIPRouter class is a class that implements the use of RIP (Routing Information Protocol).

This class is actually a part of the GridSim library but some parts are adjusted to the needs of this simulation in order to be able to write the routing table of routers to files named [router_name].out. PacketScheduler class is an interface that provides a template for the schedulers that will be used on the router.

In the research, this class is implemented by two scheduler classes, namely FIFOScheduler and SCFQScheduler. FIFOScheduler and SCFQScheduler classes as the names imply, respectively implement FIFO and SCFQ scheduling algorithms to schedule packets at routers. ArrayList class responsible

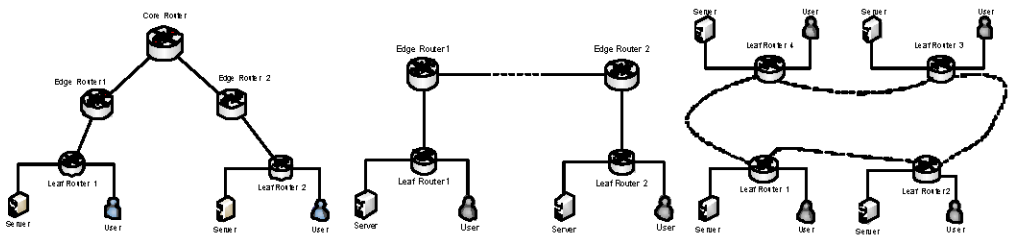

Fig. 2: Three types of router linking

in storing objects into a list. This class represents a standard data structure that is owned by the Java programming language.

The topology scenarios: The simulation consists of two scenarios divided based on the configurations of the router and link connections. We use two different types of scenarios to see whether the effects of the formation or hierarchy of links and router configurations which connect the provinces on one island and then connect and divide these islands based on the Western, Central and Eastern Indonesian zone have significant impacts on the performance of Grid systems. The two configurations are based on the thought that the territory of Indonesia is composed by three components or areas the western parts of Indonesia, the central parts of Indonesia and the eastern parts of Indonesia. These three major areas can then be viewed also as entities composed by several islands and/or archipelagos. These islands and/or archipelagos can be viewed as entities composed by one or several provinces. The concept of this dividing will be used in the router configuration for each scenario. The scenarios can be described as follows:

Scenario 1: This scenario is a representation of the proposal that divides the whole territory of Indonesia into three main sections the western, central and eastern part of Indonesia.

Each of these three sections will be subdivided into parts or units that are smaller the islands and/or archipelagos. These units of islands and/or archipelagos are subdivided again into the smallest units namely the provinces. Thus, this scenario will create a multilevel (hierarchical) dividing which starts from three main units (western, central and eastern part of Indonesia). These main units will consist of seven islands and/or archipelagos units (Sumatra Island, Java Island, Kalimantan/Borneo Island, Sulawesi Island, Bali Island- Western Lesser Sundas-Eastern Lesser Sundas, Moluccas Archipelago and Papua Island). In the end, these seven islands and/or archipelagos units will consist of 31 province units (ten provinces in Sumatra Island, six provinces in Java Island, four provinces in Kalimantan/Borneo Island, 6 Provinces in Sulawesi Island, three provinces in Bali Island-Western Lesser Sundas-Eastern Lesser Sundas, one province in Moluccas Archipelago and one province in Papua Island). Figure 3 shows the network topology for the first scenario. The points between the leaf routers symbolize the meaning of until.

Scenario 2: This scenario is a representation of the proposal that divides the whole territory of Indonesia directly into islands and/or archipelagos units. These islands and/or archipelagos will be divided again into province units. This dividing mechanism will create seven islands and/or archipelagos units and 31 province units. Each islands and/or archipelagos unit will consist of some provinces units. Figure 4 shows the network topology for the second scenario. The dash points between the leaf routers means until.

Assumptions in the Simulation: There are some assumptions during the simulation and these are as follows:

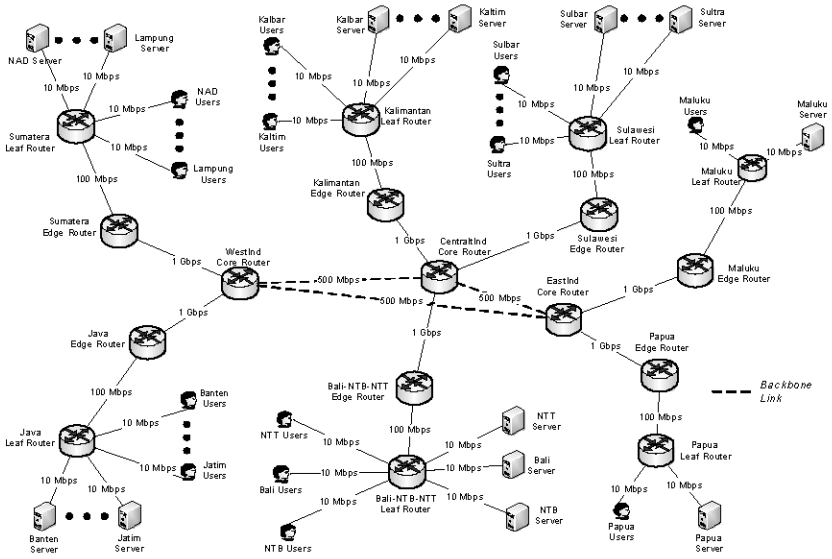

Fig. 3: Network topology for the first scenario

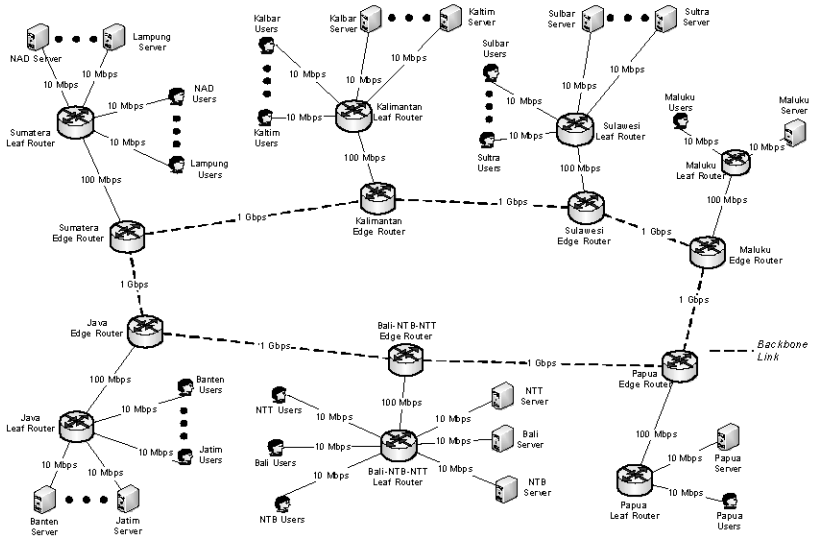

Fig. 4: Network topology for the second scenario

- User location was determined with the router connected to it, e.g., Jateng_User_519 user will be connected to jatengRouter. Thus, the user's geographic position is not simulated
- Users select the resources randomly
- Users send Gridlets at the same time. Same time in this case means all objects of the NetUser class execute the body method together
- Leaf routers, edge routers and core routers only distinguished by entities connected to them and the configuration of the links. Leaf routers are connected to the users and GridResource entities and have links with the smallest baud rate. Edge routers are connected to leaf routers and core routers and have links with baud rate greater than the links at the leaf routers. Core routers are connected to edge routers in its area and the other core routers and have links with baud rate greater than the link at the leaf routers. Thus, there are no additional characteristic that distinguish the functionality of these routers. These three types of router are the objects of the RIPRouter class
- The three simulation scenarios use FIFO and SCFQ scheduling algorithm to schedule packets on the network
- All the resources use time-shared scheduling system
- Processing time is measured by obtaining the difference between the sending and receiving time of a Gridlet. Thus, the processing time includes the propagation time of Gridlet packets in the network and execution time of Gridlets. Processing time will be more influenced by the propagation time of packets on the network because the execution time of all Gridlets is the same (all resources have the same specifications of PE). Information was obtained from the CSV files created after the simulation finished
- Characteristics of resources its name, its operating system, its architecture are static properties of the resource and have no impact in the simulation
- The load of processors used to perform the simulations is ranging from 1-10% with the amount of free physical memory is about 43-50%

The computing environment: The simulation is run under Intel® Core™ 2 Duo T5800 processor with 2.0 GHz clock speed, 800 MHz FSB (Front Side Bus) and 2 MB L2 cache 2048 MB RAM (Random Access Memory) with shared dynamically with Mobile Intel® Graphics Media Accelerator 4500MHD and 320 GB Fujitsu MHZ2320BH G2 SATA harddisk with 5400 rpm rotation speed. While the software are 32-bit Microsoft Windows Vista™ Business operating system. JDK (Java Development Kit) version 1.6.0_05 with Java™ Runtime Environment 1.6.0_05-b13. GridSim version 5.0 beta.

RESULTS AND DISCUSSION

Simulation results: In the simulation, the -Xmx300 m parameter is used to set the Java heap memory usage to 300 MB. The use of heap memory of 300 MB is done to ensure the simulation running smoothly without the occurrence of OutOfMemoryError runtime exception. In addition, the cp parameter followed by the gridsim.jar file

Table 2: Average simulation results data for the entire Provinces per Gridlet using FIFO and SCFQ scheduling algorithm

Scheduling algorithm	Scenario	Processing time (simulation seconds) Gridlet#0	Gridlet#1	Gridlet#2
FIFO	Scenario 1	239.76471	184.89620	124.45739
	Scenario 2	240.23045	185.26774	124.11812
SCFQ	Scenario 1	235.50311	180.73233	124.67395
	Scenario 2	235.78695	181.59782	124.05540

Table 3: Average processing time for the entire provinces and Gridlets using FIFO and SCFQ scheduling algorithm

Scenario	Processing time (simulation seconds) FIFO	SCFQ
Scenario 1	183.03943	180.30313
Scenario 2	183.20544	180.48006

is used again so that JVM can find where the required classes are located. Main method is located in Main file, so that Main.class will be executed.

The research compares the simulation results of the average processing time of the three types of Gridlets sent to the resource through three network topologies. The simulation will show which topology that produces the lowest and highest average processing time using FIFO and SCFQ algorithm for scheduling packets. The simulation was run 10 times in each scenario to increase the validity of simulation results and then the results were averaged.

The simulation results data per Gridlet which is the average of all provinces using the FIFO and SCFQ algorithm are shown in Table 2 while the average of all data is shown in Table 3. From Table 3, find that Scenario 1 with SCFQ algorithm gives the best performance (180.30313 simulation seconds) while Scenario 2 with FIFO gives the worst performance (183.20544 simulation seconds).

Analysis: In Scenario 1 with FIFO algorithm, it appears that processing time is large enough, i.e., 183.03943 simulation seconds. This processing time is 0.09% lower compared to the processing time of Scenario 2 using the same algorithm, i.e., 183.20544 simulation seconds. This happened because the load distribution was not done evenly. The large number of users and resources, i.e., 238 users (24.84% of total users) and 10 resources (32.26% of total resources) in Sumatra Leaf Router and 505 users (52.71% of total users) and 6 resources (19.35% of total resources) in Java Router Leaf make data traffic loads in those areas become very large. This also happened to other routers in the western region, such as WestInd Core Router, Java Edge Router and Sumatra Edge Router. In addition, the packets were served without any priorities so that all packets had to wait a long time especially those that came from and go to resources in Java. Eventually, bottlenecks occured.

In Scenario 1 with SCFQ algorithm, it appears that the processing time is 180.30313 simulation seconds or 1.49% lower compared to the processing time when using FIFO algorithm. This processing time is the lowest processing time, so that this scenario is the most appropriate scenario for IndoEdu-Grid. This happened because the load distribution was done more evenly. As mentioned in the previous paragraph, the packet traffics were large enough in the western routers.

However, arrived packets have different priorities, indicated by their weight; packets with weight 1 (high priority packets) will be served first while the packets with weight 0 (normal priority packets) will be served after the high priority packets. This differentiated service will not make the packets wait for a long time and reduce the possibility of the occurrence of bottlenecks.

The fact in Scenario 2 with FIFO algorithm can be explained with the same reasons in Scenario 1, i.e., the uneven load distribution where the routers' loads in the western part are greater than the other routers and the lack of packets priority made the packets get the same treatment.

The possibility of bottlenecks is very large. However, Scenario 2 has a difference than Scenario 1 in terms of the existence of core routers. The configuration of Scenario 2 does not involve core routers, so the packets with far destination will have greater number of hops. By observing Fig. 3 and 4, this fact can be explained by the following examples. If a user in Lampung wants to send jobs to a resource in Papua, job packets will be propagated through channels as follows:

- In Scenario 1: Lampung_User →Sumatera Leaf Router→Sumatera Edge Router→WestInd Core Router→EastInd Core Router →Papua Edge Router→Papua Leaf Router→Papua_Res→Papua Leaf Router→Papua Edge Router→EastInd Core Router→WestInd Core Router →Sumatera Edge Router→Sumatera Leaf Router→Lampung_User
- In Scenario 2: Lampung_User→Sumatera Leaf Router→Sumatera Edge Router→Java Edge Router→Bali-NTB-NTT Edge Router→Papua Edge Router→Papua Leaf Router→Papua_Res→Papua Leaf Router→Papua Edge Router→Bali-NTB-NTT Edge Router→Java Edge Router→Sumatera Edge Router→Sumatera Leaf Router→Lampung_User

Table 4: The packet lifetime in routers using SCFQ algorithm relative to that using FIFO algorithm

Scenaria	Values
Scenario 1	
WestInd core router	0.998138653
Java leaf router	0.887930488
Java edge router	0.994471472
Sumatera leaf router	0.535667588
CentralInd core router	1.004897683
Scenario 2	
Sumatera edge router	1.05569257
Java leaf router	0.858065202
Java edge router	0.959062559
Sumatera leaf router	0.673667196
Kalimantan edge router	0.995049940

The number of hops in Scenario 1 example is 6 hops, while in Scenario 2 example is 7 hops. The maximum amount of hops in this configuration is 7 hops. Thus, the inexistence of core routers will increase the number of hops for the packets with far destination. This is why the processing time of Scenario 2 with FIFO algorithm is higher than Scenario 1 with the same algorithm and makes this scenario to be the worst performance scenario Table 4.

We also computed packet lifetime the difference between the enqueuing and dequeuing time of packets in routers using SCFQ algorithm relative to the ones using FIFO algorithm in all scenarios. The value less than one indicates that the lifetime in routers using SCFQ algorithm is less than that in the routers using FIFO algorithm. In general, SCFQ algorithm causes lower packet lifetime, except only in three routers that are in CentralInd Core Router (in Scenario 1) and Sumatera Edge Router (in Scenario 2). This happens because the use of SCFQ algorithm caused overhead time to select the packets to be processed. These three routers and the other routers with value close to one are connected with high-speed link (100 Mbps, 1 Gbps and 500 Mbps), so that the packets through these routers can be transmitted directly without having to wait for long time in the queue. This also eliminates such routers need to the use of SCFQ algorithm.

CONCLUSION

In this study, IndoEdu-Grid network using GridSim toolkit is stimulating. The simulation was conducted with two different types of topologies in terms of link and router configurations and two types of scheduling algorithms FIFO and SCFQ. The average processing time is studied to obtain the most effective topology for each scheduling algorithm and the final results of the packet lifetime is used to analyze the phenomenon in the simulation.

If the FIFO scheduling algorithm will be used in establishing IndoEdu-Grid network, then the most appropriate topology is the topology that allows the packets to have a low number of hops. In this simulation, the network with the lowest number of hops is provided by the third scenario. The use of topology in the first and second scenario only makes the processing time becomes longer because the packets will have a greater number of hops.

If the SCFQ scheduling algorithm will be used in establishing IndoEdu-Grid network, then the most appropriate topology is the topology which makes the data packets with the same priorities have little chances to meet each other in a single router or link. In this simulation, topology that meets these conditions is the topology on the first scenario. Topology on the second does not meet these conditions, so it is the not appropriate topology for SCFQ scheduling algorithm.

SCFQ scheduling algorithm tends to make packet lifetime in routers with crowded traffic becomes shorter. Packet lifetime shows the difference between the enqueuing and dequeuing time of packets. This is because there are packet priority settings where the packets with higher priority will be served first, so the overall packet lifetime will be reduced.

RECOMMENDATIONS

Based on simulation results in the research, we recommend that the network topology in the first scenario with the implementation of SCFQ scheduling algorithm is used as a reference for establishing IndoEdu-Grid. The first scenario with the implementation of SCFQ scheduling algorithm has the highest level of effectiveness in terms of job packets propagation, although the achieved level effectiveness is very small (<1%). This is because this research used the size and number of Gridlets that are not too large (5000 MI, 3000 MI and 1000 MI), so the savings or effectiveness only slightly visible.

ACKNOWLEDGEMENT

We thank Muhammad Rifki Shihab for reading the first draft of this study and suggesting important improvement.

REFERENCES

Berman, F., G. Fox and H.J.G. Anthony, 2003. Grid Computing: Making the Global. John Wiley and Sons, West Sussex.

Buyya, R. and M. Murshed, 2000. Using the gridsim toolkit for enabling grid. http://www.buyya.com/papers/gridsimedu.pdf.

Buyya, R. and M. Murshed, 2002. GridSim: A toolkit for the modelling and simulation of distributed management and scheduling for grid computing. Concurrency Comput: Practice Exp., 14: 1175-1220.

Pustekkom, 2009. Jardiknas-Indonesian ICT Network. http://jardiknas.depdiknas.go.id/index.php/tentang-kami.

Sulistio, A., G. Poduval, R. Buyya and T. Chen-Kong, 2005. Constructing a grid simulation with differentiated network service using gridsim. Proceeding of 6th International Conference on Internet Computing, (ICIC'2005). Las Vegas, pp: 2-25.