

Assessing GridSim for Modeling the Global Distribution of Next-Generation Astronomy Data

James K. Tunbridge[†]

Department of Computer Science,

University of Cape Town



Dissertation

presented in partial fulfilment of the

requirements for the degree of

MASTER OF PHILOSOPHY Information Technology

Prof. Robert Simmonds *

April 3, 2025

[†] tnbjam001@myuct.ac.za

* Simmonds@cs.uct.ac.za

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

The copyright of this thesis vests in the author. Neither quotations nor information derived from it may be published without the full acknowledgment of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

Plagiarism Declaration

I, James Tunbridge, hereby declare that all the work in the dissertation (or thesis), save for the properly acknowledged, is my own. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own. I have used the APA 7th edition convention for citation and referencing. Each contribution to, and quotation in, this assignment from the work(s) of other people has been attributed, cited, and referenced. This assignment is my work. I have not allowed, and will not permit, anyone to copy my work to pass it off as his/her work.

Signature:

Signed by candidate

03.04.2025

James Keith Tunbridge

Abstract

The transfer of big data between geographic locations incurs various costs that are better managed when computing resources are used efficiently. Measuring the energy used by a computing facility is a mechanism for managing computational efficiency because the energy provided to the facility can be measured and managed. The Square Kilometer Array (SKA) radio telescope will share large volumes of science-ready astronomical data with the project collaborating partners. This dissertation attempts to address the weaknesses of the GridSim simulation toolkit for the configuration of the SKA data grid. Some of the GridSim features suited for the simulation project are: a) a network extension claiming realistic network communication; b) an extendable application programming interface because of the Java programming language; c) a datagrid extension that simulates distributed data storage, and tasks for managing the distributed files; d) packet- and flow-level network extensions and e) GridSim is used in simulations of similar real-world networks e.g., the Australian GrangeNet Gigabit network. GridSim was built primarily for modeling resources and application scheduling of parallel computing and distributed computation grids, and to assess different job scheduling policies.

The SKA wide area collaborative network will send data to its distributed partners who have their own network and energy-related policies. This work proposes a design to implement, in GridSim, a prototype of the end-to-end energy cost model for large scale networks, ECOFEN (Orgerie, 2015). The purpose of this work being to demonstrate the utility of the GridSim toolkit in spite of a few known problems with the software. Invalidation exercises were performed to determine the cause of lost events in a

network extension simulation, and to assess the implementation of the Routing Information Protocol, in GridSim, in multiple executions of the same simulation and configuration.

In this work, GridSim simulations lose events for which a solution is suggested. In addition, the work found that routing tables do not always contain matching shortest path information for multiple executions of a simulation. The implementation of the proposed design for an ECOFEN model extension in GridSim is a project for future work after one unsuccessful attempt to implement the model in GridSim. This work considered other simulation tools as potential alternatives to the GridSim toolkit, finding SimGrid to be a likely candidate. Modern computational systems are just too complex for popular software simulation tools to copy dependably which has supported a return to live network emulation testbeds for the accurate and scalable modeling of real-world systems.

Keywords: Astronomy, big data, Datagrid, ECOFEN, GridSim, SimGrid, SimJava, simulation, SKA, Square Kilometer Array Radio Telescope, VLBA, VLBI

Contents

Copyright and license	i
Plagiarism Declaration	ii
Abstract	iii
Contents	iv
List of Acronyms	vi
List of Tables	vii
List of Figures	viii
Annexures	ix
1 Introduction	1
2 Literature and Related Technologies	5
2.1 Motivation for power consumption modelling the SKA network ----	5
2.2 Major Topics of the SKA and large-scale science networks	6
2.2.1 Impacts on Big Data transfer	6
2.3 Network energy modelling	7
2.4 Simulation as a mechanism for modelling the network	9
2.5 Simulation software	10
2.6 The GridSim toolkit	14
2.6.1 The flow of a SimJava event	18
2.6.2 Problems with GridSim	19
2.7 Source code analysis	22
2.7.1 Solutions for some known GridSim problems	23
2.8 Reasons to use GridSim	24
Summary	25
3 Methodology	27
3.1 Research approach	27
3.1.1 Research Method	27
3.2 The Invalidation Experiments	30
3.2.1 Packet loss	30

3.2.2	GridSim, shortest path invalidated -----	32
3.2.3	In/Out port limitation in GridSim invalidated -----	34
3.2.3.1	ECOFEN implementation criteria -----	34
3.2.3.2	ECOFEN in GridSim -----	35
3.2.3.3	Code updates -----	35
	Summary -----	44
4	Results -----	45
5	Conclusion and Future Work -----	48
	References -----	50

List of Acronyms

API	Application Programming Interface
CERN	CERN Conseil Européen pour la Recherche Nucléaire
CTA	Check Then Act
IDE	Integrated Development Environment
IDIA	Inter-university Institute for Data Intensive Astronomy
LFN	Long Fat Network
LHC	Large Hadron Collider
LHCONE	LHC Optical Network Environment
LHCOPE	LHC Open Private Environment
MTU	Maximum Transmission Unit
RFC	Request For Comment
RIP	Routing Information Protocol
RTT	Round Trip Time
SKA	Square Kilometer Radio Telescope Array
SNR	Signal-to-Noise Ratio
TCP	Transmission Control Protocol
VLBI	Very Long Baseline Interferometry
VLBA	Very Long Base Array
VPN	Virtual Private Network
WAN	Wide Area Network

List of Tables

Table 1: An Abridged Comparison of Various Toolkits for Cloud Environments.-----	13
Table 2: Grid design challenges. -----	14
Table 3: Machine host for GridSim -----	28
Table 4: The energy profile of a NIC port -----	35
Table 5: A summary of some problems affecting the GridSim toolkit -----	47

List of Figures

Figure 1: The layered architecture of the GridSim API on top of SimJava2.-----	15
Figure 2: Message flow between objects of the packet level network extension.-----	16
Figure 3: The Objects of the GridSim network extension -----	17
Figure 4: The Square Kilometre Array collaborative Wide Area Network -----	28
Figure 5: The SKA Wide Area Network simulation-----	29
Figure 6: A dumbbell network-----	31
Figure 7: Network topologies for invalidation of shortest path in GridSim-----	33
Figure 8: Example of a network configuration file, network.txt-----	37
Figure 9: Example of the updated grid resource configuration file, resource.txt -----	38
Figure 10: Example of the new equipment.txt configuration file -----	39
Figure 11: Association of the equipment file loader and other GridSim file loaders-----	40
Figure 12: Network extension object associations for multiple ports -----	41
Figure 13: Multiple port class in GridSim -----	42

Annexures

Annexure A:	GridSim 5.2 Roadmap -----	57
Annexure B:	GridSim 5.2 Release Note-----	58

1 Introduction

This is a project to extend the GridSim toolkit with a model to simulate the power and cost of Big Data transfers in the SKA-WAN. Considered to be the largest data management challenge, the next few paragraphs briefly describe the SKA astronomical radio telescope before expanding on the purpose of this work.

The Square Kilometer Array (SKA) radio telescope project is a phased plan to create a geographically distributed astronomy antenna. The telescope will receive a broad spectrum of galactic radio frequencies than is currently possible with existing telescope technologies, collecting frequencies in the centimetre and meter wavelengths. The SKA phase 1 telescopes include the SKA1-mid of South Africa, and the SKA1-low (ASKAP) of Australia. The Karoo Array Telescope, MeerKAT, is the precursor to the SKA radio telescope (Anonymous, 2014). The 133 dishes of the SKA1-mid plus the 64 precursor dishes of MeerKAT (Raynard & Rowland, 2016) will listen for radio signals in the 0.3 to 3GHz range and the dipole antennas of the SKA1-low array listen for signals in the 0.1-0.3GHz frequency range. The SKA2-high, an array of small dish-antennas, will extend the capabilities of the Meerkat installation to include the 3-25GHz radio frequencies (Lazio, 2008).

The SKA radio telescope establishes a continent-spanning aperture for astronomy observations for studying the evolution of galaxies using their radiation emitted from sources beyond Earth's nearest galaxies. The raw data from the collectors is reduced for use in pulsar search surveys in the detection of blackholes. Other astronomy projects include cosmic magnetism (red shift studies), transients in detecting fast radio bursts, low-frequency cosmology in Epoch of Re-ionization and Cosmic Dawn, multi-messenger astronomy in Gravitational Wave events in neutron merger, and planetary system formation (Spekkens, K., 2019). In an effort to support the long-term requirements for SKA the African Very Long Baseline Interferometry (VLBI) project has established an extendable network of radio antennas on the African continent to support the development of scientific and technical skills in the region.

The Large Hadron Collider (LHC) and Large Synoptic Survey Telescope (LSST) are projects similar in scale to the SKA telescope. Unlike the LSST, the signals from SKA

sensors form a collage of layered images of the universe from the focal sector to render an image with increasing magnitudes of detail i.e. depths of field. The petabytes of data are processed and stored to create images many orders of magnitude larger, requiring novel approaches for data distribution, storage, processing, and data access (Wicenec, 2015). A small portion of the raw data (science-ready data) from the collection of in-depth-of-field images, including any applicable metadata, are stored for future analysis. The science-ready data from the Australian Pathfinder is kept at the Regional Data Center at Pawsey, Perth, Australia. The transformed Meerkat science data products are housed at the Karoo Array Processing Building (KAPB) and some of the archived data is transferred to the Regional Data Center (RDC) at the Center for High Performance Computing in Cape Town, South Africa (SKA Media, 2014). The regional inter-university Institute for Data Intensive Astronomy (IDIA) centrally manages the scheduling of science-ready data transfers to SKA Regional Science and Engineering Centers in Australia, Europe, Canada, China, Holland, India, Italy, New Zealand, and South Africa (Aikema, 2017). The rate of data transfer, which includes the repeaters for signal regeneration, will be at least 100 Gbps. The simulation of an energy model for the SKA network is of general interest to the SKA project because power is a component of the financial cost of the data communications.

Software simulation is a mechanism for the study of open research questions, being both convenient and cost-effective for in silico experiments (Archijya & Krauser, 2016). The software simulator, GridSim (Buyya & Murshed, 2002), was the prescribed toolkit for this work. Some concerns include, the age of the tool, the limited support for the tool and the lack of independent validation of the toolkit's reliability. The GridSim toolkit and SimJava (Assunção, 2009a) libraries are written with version 2 of the Java programming language which was released in 1998. Some idioms in the Java source code might not be suited to a multi-threaded (process based) simulation tool like GridSim which is known sometimes fail with an out-of-memory error (OOM) (Adineh, 2012; Bedwal et al., 2003; Chelladurai 2016; Grace et al., 2012; Ye, 2009). This work will use version 5.2 of the GridSim toolkit released in 2010. In its release note it claims improved compatibility with versions 5 and 6 of the Java programming language. The current version of the Java programming language is version 9 with its deprecated and retired classes. Accordingly, this work also considers GridSim's compatibility with the current *stable* version of the

Java programming language, version 8. GridSim was originally intended for the simulation of application schedulers and resource reservation of computational grids in distributed computing systems and clusters. Several reviews of software simulation tools have favourable opinions of GridSim (Mehta, 2010; Buyya & Murshed, 2008; Chelladurai, 2016; Taheri et al., 2012). The reviews give an impression of a reliable and versatile software simulation tool. Yet, in its road map document (Buyya, et al., 2009) for both versions 5.0 and 5.2 of the GridSim toolkit there is stated without explanation (personal correspondence Tunbridge, 2007),

“For version 6.0:

+ Complete new design for the simulation core. Remove dependency on SimJava”.

Reasons for the removal of GridSim’s dependency on the SimJava engine might include: a) poor simulation performance (Bambrick, 2020); b) simplified packet simulations produce nonsensical simulated communication times (Casanova et al., 2011), and c) the result of the multi-threaded simulation approach is not repeatable, because of the uncertainty introduced by multiple threads (Chen et al., 2005). In 2009, Salustiano and dos Reis Filo, saw the need for pseudo-parallel processes in multi-threaded simulation tools like GridSim. Unlike process-based simulation tools, single thread discrete-event simulation tools are preferable for repeatable results. This work asks whether the GridSim toolkit is the appropriate tool for the simulation of SKA-WAN Big Data transfers.

The collaboration in future research inquiry into next-generation astronomical data of the SKA radio telescope mandates that science-ready data products and their metadata be stored for current and future collaborative research projects in science. To date, SKA is considered to be the largest known data management challenge for science to resolve. The power for the transfer of big data in the WAN is largely determined by network latency. With the cost of leasing the long-haul networks the bandwidth capacity of the core network should work to be fully subscribed to achieve an acceptable ROI. The central coordination of all scheduled transfers of science products helps to prioritise the individual transfers according to the subscriber project criteria. The geographically distributed SKA regional centres have their own characteristics viz. policies, network

configurations and politics. The characteristics of a location contribute to the cost of the data transfer. This dissertation is concerned with the extending the GridSim toolkit with an energy model and simulation of the SKA-WAN to determine the energy consumption of Big Data transfer in a long-haul network.

Few reviews of the GridSim toolkit appear to draw their conclusions from practical experience. A critical review of the SimJava library appears to be also absent from the available literature. Therefore, this work cannot disregard the SimJava library either in explaining some of GridSim's known problems.

The value of this work lies in assessing GridSim for the SKA WAN extension for simulating the power consumed and the cost of Big Data transfers of next-generation astronomy data.

2 Literature and related technologies

This Section gives an overview of some of the published materials related to the GridSim toolkit and its SimJava discrete-event library. Software tools and related libraries used in this dissertation are also covered. The Section is organised as follows: Section 2.1 presents some motivations for modelling the power of the SKA_WAN; Section 2.2 introduces some of the major topics of SKA and large-scale networks; Subsection 2.2.1 explores some of the factors impacting big data transfers; Section 2.3 looks at modelling energy in a network; Section 2.4 introduces software simulation as a mechanism to model the energy of a network; Section 2.5 discusses software simulation tools; Section 2.6 is about the GridSim toolkit; Subsection 2.6.1 describes the flow of a message in a GridSim simulation; Subsection 2.6.2 deals with some of the problems with the GridSim toolkit; Section 2.7 considers the practice of source code analysis as a technique for improving software; Subsection 2.7.1 offers solutions for some known GridSim problems; Section 2.8 motivates for GridSim as simulator of choice for this work.

2.1 Motivation for power consumption modelling the SKA network

The sensors of the SKA radio telescope generate data at rates far greater than can be affordably analysed and stored using conventional computing practices. According to McKee (2017), the SKA plans to run at 200 times the scale of the LHC. The large data transfers of science-ready data products include their metadata. The transfer of data takes up allot of time, power, and network bandwidth capacity. The distributed regional data storage facilities manage their own data storage and processes (Tate et al., 2016); they have their own policies, network configurations and politics. The countries hosting the radio telescopes are expected to consume less power than the regional science centres (Aikema & Simmonds, 2017). Accordingly, the regions are allocated a portion of the project's annual budget. The long-haul WANs are costly to lease which means the core network capacity should be properly utilised to benefit the most from the resource. Some communication problems in a WAN include the participation of different bandwidths in the grid network, the inefficient use of the network, and path packet loss. Inefficient networks can increase the latency of the network which determines the power used for data transfers. This dissertation is concerned with the design of a network

energy model, ECOFEN, as an extension to the GridSim toolkit. The purpose of the extension is to simulate the power consumed by big data transfers over the SKA long-haul network.

2.2 Major Topics of the SKA and large-scale science networks

The management of SKA data is via a data transfer platform which catalogues the science-ready data as the data is scheduled for transfer to IDIA systems in South Africa and ASTRON facilities in the Netherlands for analysis (Aikema & Simmonds, 2017).

The SKA radio telescope project adopted the CERN Large Hadron Collider (CLHC) tiered data distribution model (Newman & Tsen, 2011). Tier-0 facilities receive information and perform computations, the tier-0 data products are passed to national Tier-1 data storage and computing nodes. The Tier-1 facilities replicate the data to regional Tier-2 geographic, localized facilities over fibre-optic connected endpoints where the data can be accessed for scientific study (Ernst, 2015). Aikema et al. (2017) elaborate further about the data transport of the SKA Meerkat (mid-tier) antenna at Carnarvon, South Africa; the delivery of data to observatory staff and to SKA regional centres. The SKA radio telescope adopts a similar Cern LHC Open Network Environment (LHCONE) to connect its Tier-2 to Tier-1 locations. The LHCONE core service is a Virtual Private Network (VPN), a point-to-point service for on-demand linkages for the orchestration of scheduled transfers to best utilise the leased network bandwidth capacity (Stanu & Martelli, 2015). The scheduling of data transfers is determined by the criteria provided by the science project. In the document, *HL-LHC network needs and data transfer challenges* (n.d.) some of the challenges with managing bandwidth capacity, in a stateful network management system, are listed.

2.2.1 Impacts on Big Data transfer

Several reasons may explain packet loss in a network. The transfer of big data is influenced by I/O data rates which are in turn affected by data compression, coding techniques, and long-distance data transport (Jones et al., 2012). WAN traffic congestion can be significant because the many nodes use links with non-uniform traffic load patterns. Big data travels over Long Fat Networks (LFN); long link distance, lengthy delays, and large bandwidth. Bandwidth congestion in any part of the Long Fat Network (LFN) and network errors can affect packet throughput causing

performance bottlenecks.

Determining the energy consumption for data transfer depends on the time a packet spends in the wide area network and, therefore, network congestion. Causes of network congestion include bandwidth bottlenecks, link delays and propagation delays. The scheduled data transfers of the SKA data network serve to mitigate these effects while also ensuring the optimal subscription of core network capacity (Aikema, 2017). Network repeater equipment amplify the signal to compensate for signal attenuation in the WAN. The repeaters receive their power from the computing facilities of the sender and the receiver. This simulation does not model the power of the repeater equipment, which the ECOFEN model acknowledges is not material to its calculations.

Network latency (Patibandla, 2020) is the time it takes for a network packet to travel over the network path from sender to receiver. Queue delays, process delays, and the physical distance between communicating devices determine network latency. Although there are strategies to keep processing delays to a minimum, the transmission errors are mostly spurious or caused by network switch buffer overflows. Spurious errors arise when the receiver misinterprets one or more bits of the packet, this type of error cannot be corrected which means the packet must be resent. Resending packets causes congestion in the network. If Network congestion errors are not corrected by link-layer protocols packet loss becomes likely. Network traffic effects (Bonaventure, 2011) have an impact on certain layers of the Open Systems Interconnection (OSI), affecting each layer slightly differently. Layer 1 is affected by network latency, where the span distance is the primary contributor, and spurious errors are caused by the nature of the physical media like signal-to-noise ratio (SNR). Secondary contributors such as the protocol used, and queuing latencies impact layers 2, 3 and 4 of the OSI. Network switches and routers are susceptible to congestion errors from their queuing structures and the traffic in the queues.

2.3 Network energy modelling

ECOFEN is an end-to-end energy cost model and simulator for evaluating power consumption in large-scale networks (Orgerie et al. 2015), it is suitable for any type of network topology with any equipment. The model implements energy conserving

techniques which include Adaptive Link Rate (ALR) and selectively turning nodes off. The energy consumption of each “crossed-equipment” – for example, multiple flows transiting a router - are part of the end-to-end cost of communication calculation.

Assumptions about the ECOFEN model:

- 1) Energy consumption of equipment (router, repeater, NIC) depends on the variable power consumption over time, which is a function of network traffic volumes.
- 2) Only plugged facilities consuming electricity are included in the model. Links are excluded from the model; their “cost” is reflected in the equipment they link.
- 3) Equipment like routers do not have energy saving features; all ports are full power.
- 4) The optimization of equipment *halt* and *boot* processes is out of scope for the model.

Equipment has two cost components, namely a fixed energy cost i.e. the energy consumed at idle, and a variable energy cost i.e. the part that depends on the amount of network traffic. Both cost components depend on bandwidth capacity (B), the duration of the transfer (L), the amount of cross traffic (the model implements fair sharing, B/n_f) and the equipment type. For a given equipment, fixed and variable cost components are defined further: 1) the power consumed by the ports during transfers is the variable part of energy used; 2) the power consumed by the equipment itself with zero data transfers is the fixed portion of used energy. In Adaptive Link Rate (ALR), a port can accept several rates viz. the transmission rate is not a discrete function. A port consumes the energy it needs to adapt to the link rate and some extra energy for the actual traffic. The power consumption of a port is expressed as the affine function (f_1), if given the slopes α ($\alpha_i > 0$) to the plot of Power consumption (in watts) vs. the range of Bandwidths (BD_i) for the port(s) of the given equipment, where P_i is the start power for each BD level.

$$P_{work} = \begin{cases} P_0 & \text{if } BD = 0 \\ \alpha_1 BD + P_1 & \text{if } BD \in]0; BD_1] \\ \vdots & \\ \alpha_i BD + (P_i - \alpha_i BD_{i-1}) & \text{if } BD \in]BD_{i-1}; BD_i] \\ \vdots & \\ \alpha_n BD + (P_n - \alpha_n BD_{n-1}) & \text{if } BD \in]BD_{n-1}; BD_n] \end{cases} \quad (f1)$$

Note. Piecewise affine function for the power consumption of a port, P_{work} (Orgerie et al., 2011).

The total energy for the transfer ($E_{Transfer}$) is calculated by function (f2) for two nodes and two routers.

$$\begin{aligned} E_{transfer} = & E_{EthernetCard}(\text{NodeA}, BD, L) + E_{Router}(\text{Router1}) \\ & + E_{Port}(\text{In}, \text{Router1}, BD, L) + E_{Port}(\text{Out}, \text{Router1}) \\ & + E_{Port}(\text{In}, \text{Router2}, BD, L) + E_{Port}(\text{Out}, \text{Router2}) \\ & + E_{Router}(\text{Router2}) + E_{EthernetCard}(\text{NodeB}, BD, L) \end{aligned} \quad (f2)$$

Note. Orgerie et al., (2015)

2.4 Simulation as a mechanism for modelling the network

Software simulation is a mechanism for prototyping algorithms of academic interest and assessing their application in real-world computing systems. The software simulation is a convenient substitute for real-world systems that are often inaccessible or too complex to reproduce in a laboratory test-bed environment (Taheri et al. 2012). Bottom-up network simulation models originate from the network protocol community. The bottom-up network approach to simulation is not recommended for simulation because the components are too complex to implement. In a network simulation the parameters of the physical network are hard to measure. Network simulations, therefore, concern themselves with the macroscopic behaviour of the network (Casanova et al., 2013b).

Function (f3) explains how end-to-end latency of a network path is calculated as the

sum of all the intervals.

$$\text{latency} = \underline{D_p} + \underline{D_e} + \underline{D_l} + \underline{D_s} \quad (f3)$$

- D_p Time spent in the link, processing delay. Propagation delay.
- D_e Time for the processor to parse the data packet.
- D_l Time to move data from the processor to the link.
Link delay or serialization.
- D_s Time spent scheduling the packet(s). Scheduling delay.

For large data transfers in a large RTT fibre-optic Wide-Area-Network (WAN), the RTT calculation is affected primarily by the propagation delay (D_p). Propagation delay is the time it takes for a signal to traverse the distance separating the endpoints in a network, usually at the speed of light. According to Taheri et al. (2012), only latency and bandwidth measures are required for the simulation of a computer network.

The SKA WAN connects several geographic locations and their heterogeneous networks. Building a test-bed simulation would be impractical, leaving software simulation as the viable alternative.

2.5 Simulation software

The software simulation tools, SimJava (1998-2002), GridSim (2002-2010) and SimGrid (2001-2023) are some early attempts at grid computing simulation. By 2010 the attentions of the GRIDS laboratory team had also shifted to the development of the cloud simulator, CloudSim (Buyya et al., 2009). In 2011, Bąk S. et al. compare the simulation objectives of a few cloud capable simulation tools. In 2016, Patel, N. & Patel, H. compared various simulation toolkits for the cloud environment, some are listed in Table 1. The CloudSim toolkit features prominently as the underlying platform as an engine for several simulators. Platforms OMNeT MPI (Varga, 2021), Bluetool and NS2/NS3 (Issariyakul, 2012) are the core components for some other simulators (Table 1). The simulators FlexCloud and CloudSim have no underlying platform. According to Bambrick (2020, p. 30), most cloud simulators are implemented in Java which hinders their scalability and performance; implementing simulators in C++ improves scalability. Some simulator tools give more detailed simulation results than other toolkits. For instance, the simulator iCanCloud is more detailed than CloudSim, but CloudSim is more

adaptable (Bambrick, 2020, p. 6). The energy models in Cloud computing assess the operational costs of Datacentres (DC's) hosting the cloud services. Recent energy models assess the performance and energy for the migration of virtual machines (VM's). Both CloudNetSim++ (Bambrick, 2020, p. 13) and Nutshell (Bambrick, 2020, p. 19) simulation toolkits claim to simulate accurate packet networks.

The proposed future release of a version 6 of the GridSim toolkit without its dependencies on the SimJava library appears to have been realised with the development of CloudSim. Some alternatives to the early grid simulation tools like GridSim include the packet level simulators OptorSim (Cameron, et al., 2010) and GangSim (Dumitrescu et al., 2005); however, both lack TCP realism. Other packet level simulators are OMNet++ (Chen, 2015; Varga, 2020), GTNetS (Riley, 2003), and NS2/NS3 (Issariyakul, 2012), for studying wired and wireless TCP communications. The network simulator NS2/NS3 can import the Brite (Medina, 2001) network topology design products to help with the simulation configuration. The main problem with packet-level simulators is the duration of simulation which can be orders of magnitude larger than simulations with realistic topologies that are modelling many data flows. The packet-level simulation can be prohibitive for studies of distributed systems with large volumes of communication, like the scientific collaborations of the SKA network. The GridSim network extension models both packet- and flow-level communication. GridSim has found application in GSSIM (Bak, 2011), Alea2 and, for a brief period in CloudSim.

The available literature offers two popular flow-level network simulator alternatives to the GridSim toolkit these are OptorSim and SimGrid. In 2015, Lebre et al. proposed adding storage to SimGrid; the latest SimGrid 3.35 has one storage model which is in the File System plugin. The File System plugin has the notion of host storage, simulated Files on top of storage: disk I/O and files; copy/move a file to another location on a remote host; writes to a file; positioning the file pointer; etc. The File System plugin operations as they are described by SimGrid' API documentation are like the GridSim datagrid tasks. However, unlike GridSim, the accuracy of the SimGrid network has been the target of invalidation studies (Casanova & Fujiwara, 2007; Casanova et al., 2008; Casanova et al. 2013). The SimGrid simulator also has a Link Energy plugin for measuring the energy consumption of the current load in a Link (The SimGrid Team,

2023). The GTNetS simulator verifies the accuracy of the SimGrid flow-level network implementation.

Table 1.

An Abridged Comparison of Various Toolkits for the simulation of Cloud Environments.

Simulator	Pub. Year	Cited By (as in Oct 2016)	Underlying Platform	Available As	Programming Language	Model				Federation Policy	Graphical Support	Migration Policy
						Cost	Energy	Comm.	Security			
FlexCloud	2015	5	Any	Opensource	JAVA	Yes	Yes	NA	No	NA	Yes	Yes
GDCSim	2014	9	Bluetool	Opensource	C++/XML	NA	Yes	NA	No	NA	No	No
EMUSim	2013	73	CloudSim AEF	Opensource	JAVA	Yes	Yes	Limited	No	No	No	No
GreenCloud	2012	397	NS-2	Opensource	C++ Otcl	No	Yes	Full	No	No	Limited	No
MR-CloudSim	2012	18	CloudSim	NA	JAVA	Yes	Yes	Limited	No	Yes	No	No
CloudSim	2011	189	Any	Opensource	JAVA	Yes	Yes	Limited	No	Yes	No	Yes
NetworkCloudSim	2011	152	CloudSim	Opensource	JAVA	Yes	Yes	Full	No	Yes	No	May
CloudAnalyst	2010	312	CloudSim	Opensource	JAVA	Yes	Yes	Limited	No	Yes	Yes	No
GridSim	2002	170	SimJava2	Open Source	JAVA	Yes	No	No	No	No	No	No

Note. The simulators listed here, from Table 1 of Patel, N. and Patel, H. (2016) on p. 18976, each have an Energy Model.

2.6 The GridSim toolkit

The GridSim software toolkit (Buyya 2002, 2008), was developed by the GRIDS lab team at the Monash University in Melbourne in 2001 as an extension to the SimJava discrete-event, process-based library (Simatos, 2002). Table 2 describes a few of the grid computing challenges GridSim claims to address. Figure 1 illustrates the layered and modular architecture of GridSim.

The latest version of the GridSim toolkit is on the official project website at Sourceforge – <https://sourceforge.net/projects/gridsim>. The project was listed for the first time on the 19th of August 2007 and this dissertation uses version 5.2 dated the 25th of November 2010. The bundled gridsim.jar and simjava.jar libraries are dated the 26th of November 2010 and 22nd of October 2009, respectively. Two earlier versions of the GridSim toolkit, GridSim 5.0 beta dated 25th of September 2009 and GridSim 4.2 beta dated 12th of October 2008 are still available for download.

Table 2.

Network Grid design challenges.

Grid Challenge	Definition
Multiple layers of schedulers	The higher-level scheduler has less information about the distributed resources since local resource managers control the resources.
Lack of Control over resources	Grid schedulers do not have ownership of the underlying resources.
Shared resources and variance	Grid does not supply dedicated access to resources as resources are shared which leads to a high degree of unpredictability and variance (uncertainty)
Conflicting performance objectives	Owners and end users have conflicting preferences. They may have different local policies, cost models, security mechanisms. Multiple layers of schedulers
Heterogeneous Network protocols	Grid resources and networks have different configurations, capabilities, policies, and domains.

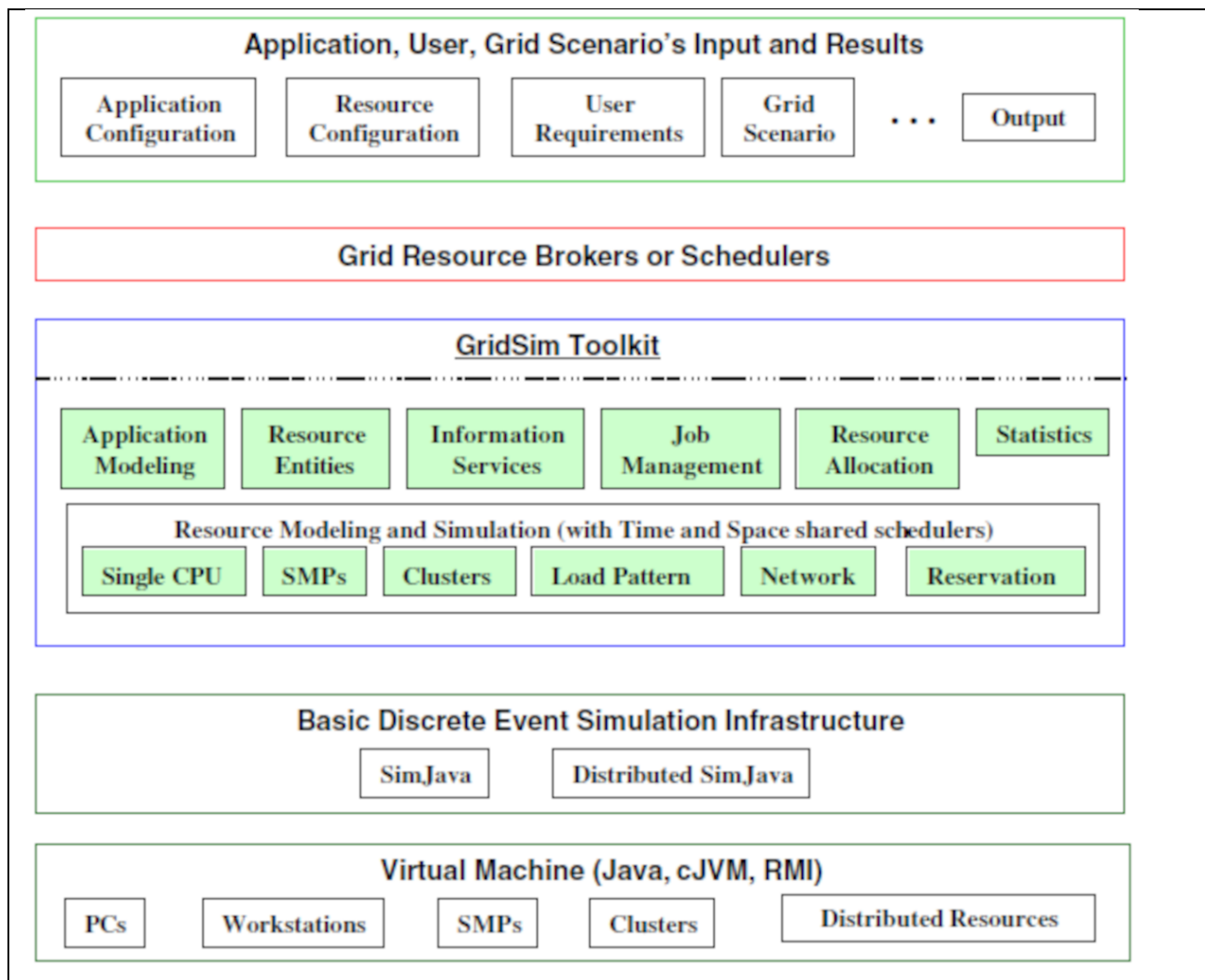
Note. A list of network grid design challenges the GridSim toolkit addresses.

Reproduced from Bedwal et al., 2014.

The GridSim toolkit: a) assesses a scheduler' performance with any combination of heterogeneous resources and users; b) composes application tasks, allocates tasks to resources and performs end-to-end task management, and c) performs wired-network data grid file replication. The core of GridSim is the generic discrete-event SimJava library which holds the i) time ordered Future Event List (FEL), ii) advances a simulated clock variable which steps through the FEL by event-execution time, iii) manages the state of the system, iv) updates the destination resource buffers with FEL content, v) updates the state of SimJava entities, and vi) synchronises all the threads of the simulation.

Figure 1.

The layered architecture of the GridSim API extension of SimJava2.



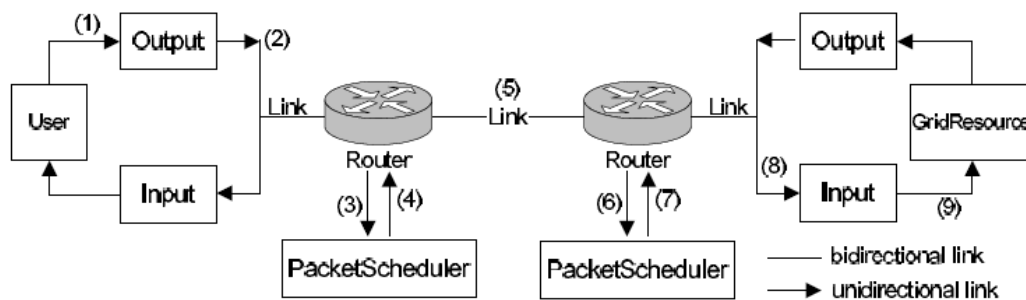
Note. A layered and modular architecture for grid simulation which leverages existing

technologies as separate components (Buyya, 2002).

Written in Java, GridSim uses Java's Object-Oriented (OO) language, like polymorphism, to extend its range of modular capabilities. The various heterogeneous entities - threads - communicate by exchanging messages called events in SimJava. Communication between directly connected nodes is an exchange of IO_event message objects over their respective NetIO non-buffered ports. A UDP-like network extension for packet communication in GridSim simulates network communication over a user-defined topology of routers, links and buffered Input- and Output-ports (Figure 4). The GridUser in GridSim initiates the simulation with a series of predetermined tasks.

Figure 2.

Message flow between objects of the packet level network extension.



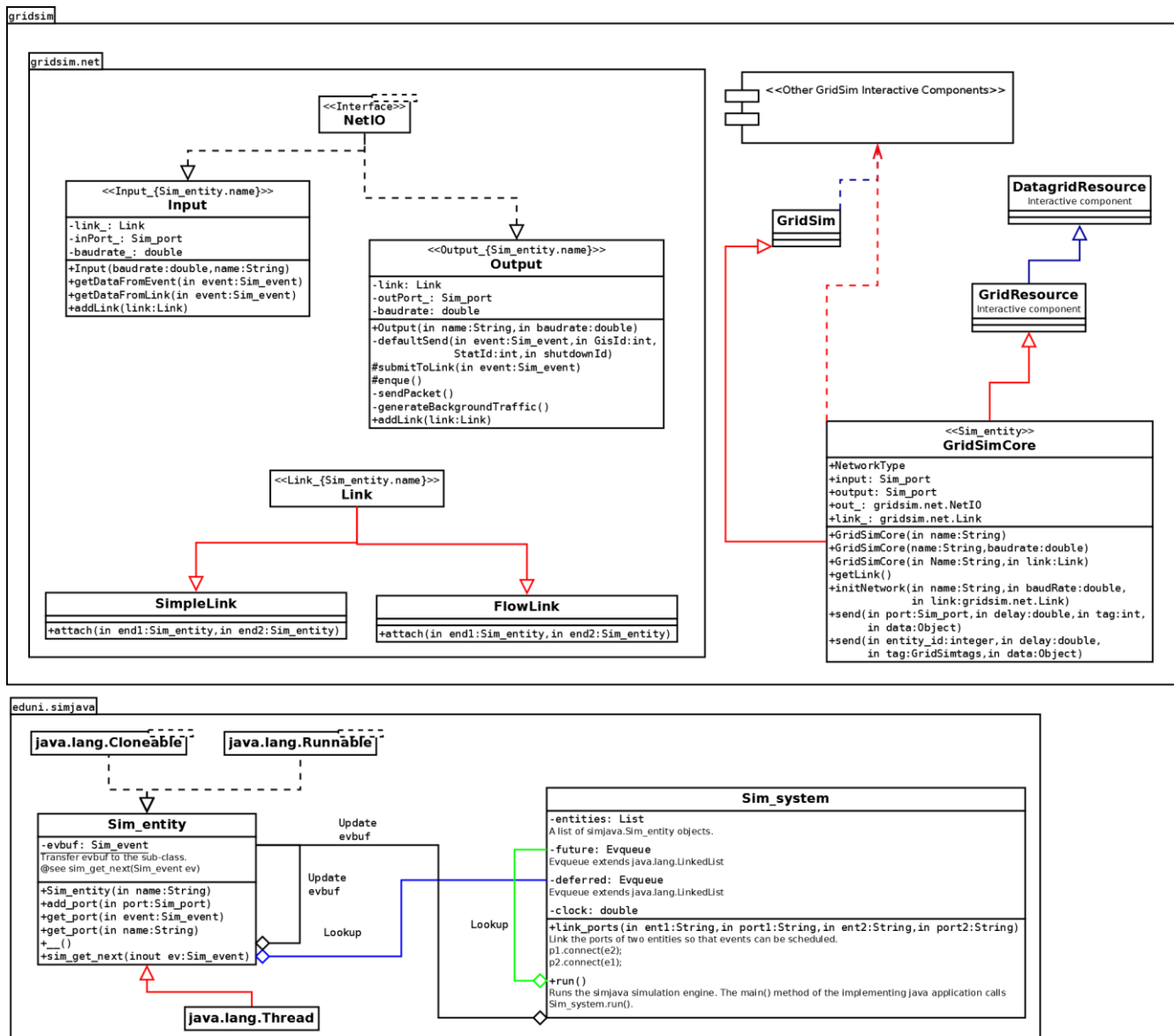
Note. Sulistio et al. (2005)

GridSim's simplified packet network model sacrifices accuracy for improved performance viz. only does coarse-grained network simulations. In Figure 2, the relationships of the objects in a packet-level network are illustrated and the data flow sequence numbered, from sender (GridUser) to receiver (GridResource). The shortest-path-first Routing Information Protocol (RIP), RFC1038 (Hedrick, 1988), is the default setting for populating the routing tables at the start of the simulation after all the configured resources have been front-loaded viz. cached. Front-loading the configured resources registers the resources with GridSim's Grid Information Service (GIS). In network simulations the GridSim network extension does not implement a Maximum Transmission Unit (MTU) for the network path. A path MTU

refers to the configuration setting which is the largest datagram size for the network that does not fragment the packets in-flight. In a datagrid simulation, the sender's Output port creates MTU-sized chunks of data; the datagram is fragmented and the individual fragments

Figure 4.

The Objects of the GridSim network extension



packetized. Similarly, a network router creates additional MTU-sized packets when the size of any in-flight packet is larger than the MTU size of the downstream link. A packet scheduler determines the in-flight order of packets in the router's queue. The GridSim network extension (Figure 4) does not model bandwidth sharing for packet-level networks. Although, fluid (flow) level networking models have been extended to do bandwidth sharing in

GridSim (Broberg and Buyya, 2009). Unlike packet-level networks, the flow-level network extension is an abstraction of network activity.

A single FlowPacket represents the activity of the entire flow for the duration of the flow's lifespan and a MIN-MAX bandwidth fair sharing algorithm allocates equal portions of the available bandwidth capacity to each of the active flows. The MIN-MAX algorithm approximates TCP bandwidth sharing in the network which is simple to implement for acceptable results in most network simulations. Data is not fragmented and there are no packet schedulers. The flow-level network: a) uses a fair bandwidth sharing algorithm when many flows are active over one or more links, b) updates the remaining active links of the flow with the lowest bandwidth (a.k.a. bottleneck bandwidth), and c) forecasts the duration of each active flow, to determine which will complete first.

2.6.1 The flow of a SimJava event

The entities of the GridSim extension insert scheduled events (messages) into SimJava's Future Event List (FEL). The first event of the FEL sets the value of SimJava's simulated clock to match the event's scheduled execution time, an event attribute. The simulation removes and processes the events with a scheduled execution time matching the value of the simulated clock. The iteration of the FEL ends when the next event has an execution time greater than the value of the simulated clock, and the simulation ends when no more events appear in the future queue. Events with an execution time matching the value of the simulated clock but cannot be processed because of a change in their destination status are inserted to the Deferred Event List (DEL). After parsing the FEL, the entities (Java threads) that have changed state from waiting to runnable are started; the wait state of the thread is interrupted. The entity processes the event in its buffer, before it fetches its next events from the DEL. The predicates for loading events from the DEL are `Sim_type_p`, `Sim_from_p`, `Sim_any_p` and `Sim_none_p`. Note, the predicates in SimJava do not use the scheduled event time as the criterion, because their execution time has to match the current value of the simulated clock. When the entity has processed its events, the system updates the semaphore and changes the state of the entity to waiting; the thread is in a wait state. The SimJava engine verifies that all threads are completed and begins a new iteration over the FEL. The first FEL event updates the value of the simulated clock with its scheduled execution time value, advancing the system time; one tick of the SimJava clock.

A GridSim simulation continues while, 1) the SimJava “running” attribute is true; 2) the message tag of the Sim_event buffer is not -1; 3) and, the FEL has content. An empty FEL causes SimJava to end the simulation. In addition, the GridSimShutdown Service (GSS) thread waits for all GridUser objects to complete their processes before changing its state to signal an end to the simulation. Upon ending a simulation, the GridInformationService (GIS) processes an “end of simulation message” closing each of its registered SimJava entities. The number of users configured for the simulation is a constant value, not a value calculated by counting the configured user objects at runtime. A simulation can remain active while the count of user objects is greater than the parameter for the number of configured GridUser objects.

2.6.2 Problems with the GridSim toolkit

In their 2007 seminar paper, Fahringer et al. described as a major concern the questionable quality of GridSim results. They expressed their concern at the number of research articles using the results from GridSim simulations (Casanova et al., 2013a). Simulators, like GridSim, that follow a simplified network simulation model, often give nonsensical simulation timings, and the simplified network models are not scalable (Casanova et al., 2013a, Casanova et al., 2013b). A well-documented drawback of package-based simulations is the poor performance in the simulation of multiple grid sites (Ostermann et al., 2011). However, some literature reviews like Curiel et al. (2007) and Taheri et al., (2012) fail to disclose GridSim’s potential flaw which does little to assist in the decision of an appropriate simulation tool for a particular project. Support and development of the GridSim toolkit ended in 2010 with the release of version 5.2 (CLOUDS, 2010). Curiously, the display of the version number on the command-line-interface (CLI) has not changed, it is still version 5.0.

A decline in grid simulation performance is a well-documented drawback of package-based simulations with many configured nodes (Ostermann et al., 2011). In 2012, Quinson reported a network problem in the bundled GridSim example of a flow-level network, claiming that active data flows are not allocated correct bandwidths resulting in a doubling of the transmission time. In 2013, Velho et al. compared the simulators GridSim, OptorSim (Bell et al., 2003), and CloudSim (Buyya et al., 2009c); their work also invalidated the

GridSim flow-level network model. To illustrate the problem, let us assume there are three Flows ($n_f = 3$) scheduled for a link with a bandwidth, B (in bps). The incorrect allocation of bandwidth in the flow of packets is illustrated as follows: Flow #1 receives B , Flow #2 receives $(B/2)$ bps, Flow #3 receives $(B/3)$ bps, which are incorrect allocations of the bandwidth capacity. The correct calculation should allocate (B/n_f) bps to each active flow, viz. $(B/3)$ bps.

In 2010, Klusáček et al. noticed that GridSim stages a constant set of resources for the simulation whereas simulators like Alea (Srivastava, 2011), an extension of the GridSim library, adopted a more platform friendly, dynamic provisioning of resources approach. Dynamic provisioning creates instances of the grid resources, as they are needed to conserve platform resources like memory and the OS management of threads. Some challenges with the multi-threaded approach of the SimJava2 library are well-known. In 2002, Simatos et al. suggested minimising the many context switches amongst the active threads of SimJava simulations to reduce the overhead on the host platform and to improve the performance of SimJava-based simulations. Simulators like Alea and GroudSim attempt to overcome the scalability challenge of multi-threaded simulators with a single thread per simulation process approach (Ostermann, 2011). A similar approach appears to have been used by CloudSim. In the early years of CloudSim, GridSim provided the core thread, the datagrid, and the framework for task composition and lifespan management. Newer versions of CloudSim do not depend on the GridSim extension (Bambrick, 2020).

In 2016, Ilyas & Elkhailifa claimed that some users of the GridSim toolkit were having difficulties configuring their simulations. Also in 2016, Eng reported a problem with the accuracy of the time value of the SimJava clock variable. In 2009, de Frutos noticed that changing the channel bandwidth assigned to the resource clusters in a gridlet simulation that the simulated time did not change. In 2008, in a reply to Cunha, Marcos de Assunção referred to the unsynchronized “future.add_event(Sim_event ev)” method of the enclosing “Sim_system.add(Sim_entity e)” method which sometimes triggered a concurrent modification exception.

The GridSim Toolkit is also known to fail with a Java “out of memory” error (Klusáček, 2008; Ostermann et al., 2011; Ye et al., 2008; Adineh, 2012). The GSSIM simulator, which is based

on GridSim, is known to always end with an out-of-memory error, allegedly because of the Java Virtual Machine memory limitation of 1GB. The Masaryk University aimed to resolve the problem of large simulations with their GridSim-based Alea simulation tool (Klusáček & Hana, 2010). However, in GridSim, the Jobloader mechanism caches all the resources and tasks for the simulation upfront, which can trigger a similar exception if host platform resources are depleted

Concurrency refers to the mechanism of independent processes that attempt to simultaneously access a member of the Java Collection Framework (JCF). To preserve data integrity the Java language works to prevent violations of its concurrency rules. The monitor mechanism for a thread class in Java's mutual exclusion mechanism, acquires a semaphore permit or releases the lock on a block of code for a process to access the shared code in memory. A violation triggers either a `ConcurrentModificationException` (CME) or a `ConcurrentTimeoutException` (CTE) error. Giancarlo (2009) and Hojiyev (2012) claim that concurrency will remain a problem for GridSim when many events are sent from different senders to a common destination.

GridSim cannot perform a simulation multiple times in one process. Forward-loading entities which can be used by the next simulations in one process uses the same configuration for the simulation environment. In 2008, De Assunção described a change to the SimJava library to accomplish this but found that not all GridSim classes implemented the requisite `Cloneable` interface (De Assunção, 2008b). A similar update is described by Klusáček (2014) which lets GridSim reset the value of SimJava's internal state variable to "incomplete". The Alea simulator uses the latter mechanism.

Other problems include,

In 2009, de Assunção described the presence of a bug in a cluster network simulation.

"The problem arises when there are two events in the deferred queue, one for GridResource and the other for the policy. Both are scheduled for the same time, and both will result in making the entities access internal variables of the policy".

The error is unlikely to affect the simulation of the SKA WAN because there is no resource reservation in the ECOFEN model.

2.7 Source code analysis

The longer an error remains undetected in a program, the more costly it can be to rectify. Detecting and correcting bugs early in the development cycle is the best approach to improving software production. The approaches to error detection are static code analysis and dynamic software analysis. Static code analysis is an analytical method for examining source code to find implementation errors before the software is used. Detection of errors with static code analysis improves when combined with dynamic software analysis. The latter detects implementation errors at runtime. Automated static analysis tools compare favourably with the manual approach, are faster and may harness the knowledge obtained from earlier work. Regardless of their convenience, automated tools still need human intervention to decide the appropriate intervention (Gomes et al., 2009; Ilyas & Ekhalifa, 2016). In 2015, Lin et al. supported the utilitarian use of a tool to detect incorrect semantics for replacing with higher-order idioms. The check-then-act detector (CTADetector) is a static code analysis tool for interactive analysis of Java source code (Lin & Dig, 2013). The CTADetector found no check-then-act idioms in the GridSim source code requiring correction.

This version of the GridSim toolkit did not include a copy of the SimJava source code; also confirmed by Teng, 2009. The source code used for this work was from the Edinburgh University website (de Assunção, 2009a). The GridSim's version of the SimJava library is dated 22 October 2009 and its compiled classes are dated May 2009. The corresponding dates of the 2002 SimJava2 archive differ and the GridSim extension did not compile with the 2002 release of the SimJava2 library. Decompiling GridSim's SimJava bytecode was, therefore, necessary.

Comparison of the 2002 version of SimJava with GridSim's decompiled source code found that the ordered Evqueue class, a linked list, had been substituted for a novel FutureEvqueue class. The FutureEvqueue class has a Java TreeSet Container object for caching its events. The FutureEvqueue lacks a Comparator class for its TreeSet attribute. The lack of a Comparator causes a simulation to fail when an event in the FEL cache is not in event

execution time sequence.

The GridSim network extension implements a single Output and Input port per GridResource and GridUser object, respectively. In 2008, Sulistio wrote:

“Unfortunately, due to a design issue, an entity in GridSim can only connect through one Input and Output port. This is also due to SimJava definition of ports, i.e. they are unidirectional (1-way) communication links”.

The sender (usually a GridUser) and the receiver (a GridResource) being limited to one in- and one out-port would be an impediment for the implementation of the ECOFEN energy prototype in GridSim. Furthermore, a GridSim router abstracts the notion of ports; a bidirectional link binds the router directly to a peer or host, but a router can bind more than one link. An inspection of the source code suggests that the number of ports can be increased for a resource (an entity), more about this in the methodology section.

The MTU values of a network can affect Round-Trip-Time (RTT) which occurs when in-flight packets are fragmented, and each new packet is encapsulated with a new header. The new headers increase the volume of in-flight data. Inspection of GridSim’s source code found no new headers for the fragmented in-flight packets viz. simulations do not model the additive effects of a packet header on network latency.

2.7.1 Solutions for some known problems

Inspection of the source code to understand the concurrent modification exception queried by Cunha in 2008 mentioned in the literature, found that GridSim Semaphores were not updated when an entity was created dynamically at runtime. The purpose of the Java Semaphore object is to allow multiple threads to access a shared resource. The problem is resolved by adding the “p” method of the onestopped semaphore class to the “add_entity_dynamically” method.

In Ye (2008), Marcos de Assunção explains the rationale for using the TreeMap object in SimJava’s FutureEvqueue class. Unfortunately, the FutureEvQueue class uses the similar java.util.TreeSet and a there is no Comparator class. Consequently, the events of the future queue are inserted in a natural order, but not necessarily in the order of their event

execution times. When an event in the FEL has a scheduled execution time less than the value of SimJava's simulated clock, the simulation will fail with an error message.

The RIP network implementation and the registration of resources to GridSim's Grid Information Service (GIS) use SimJava's pause method for the configuration of the simulation. Upon inspection of the source code, GridSim's Round-Trip-Time (RTT) value was found to include the intervals from the pause methods. Consequently, GridSim's RTT value is unlikely to compare favourably with values from the real world for a similar set of network parameters.

Inspection of GridSim's Router object's "hostTable" found entries that should only be in the "routerTable". The Application Programmers Interface (API) documentation for the Router class describes the correct data contents for the respective routing tables of the Router object.

2.8 Reasons to use GridSim

The SimGrid toolkit was written in the C/C++ programming language, the objects are expanded with plugins bringing novel features to the simulation platform. The GridSim toolkit is written in the Java programming language which can add Java packages to extend GridSim's range of modular features. Some of the beneficial features of GridSim: many users and heterogeneous resources can run concurrently (in parallel) in a simulation; has a flow- and packet-level network implementation; measures Quality-Of-Service (QOS – bottleneck Bandwidth, latency and RTT); can reserve resources by calendar; assesses scheduler algorithms for job reservation; and can produce background traffic (cross-traffic) for better network realism. GridSim has a Data Grid package for studying scientific collaboration over wired geographic networks. For this work, the GridSim toolkit has some extensions for the simulation of the SKA WAN and inspection of the source found that GridSim could be updated to accommodate multiple ports. The GridSim toolkit was, therefore, chosen for the proposed design and future implementation of a prototype of the ECOFEN model.

Summary

The Square Kilometer Array radio telescope data grid follows the Large Hadron Collider Open Network Environment (LHCONE) and the LHC Virtual Private Network (VPN) data distribution model for maximizing bandwidth use and reliable data transmission. The GridSim toolkit is an extension of the SimJava discrete event simulation library. The reliability, scalability, accuracy, and coarse-grained UDP-like network are some of the concerns mentioned in the literature.

Validation experiments done on some simulation tools including GridSim emphasized their respective shortcomings. In GridSim the max-min fair bandwidth sharing algorithm allocated the incorrect bandwidth to flows in the flow-level simulation. Other problems allude to GridSim's performance because it is a process-based (multi-threaded) discrete-event simulation tool, written in the Java programming language. The purpose of GridSim was not for network analysis simulation, offering a very simplified network extension and coarse-grained simulation. GridSim added flow-level networking to overcome challenges with the simulation of large real-world distributed grid computational systems. Concerns exist in the available literature about the quality of the results from simulation and their use in research papers. The submissions alleging problems with the GridSim toolkit are seldom mentioned by the published reviews of simulation tools.

Code analysis is an accepted practice, a useful tool for reducing costly problems in software code. Both static and automated techniques are complementary in their use to detect and in their resolution of new and known coding errors. The CTADetector is a tool used in this work to find CTA idioms in GridSim.

This dissertation encountered a number of problems in GridSim offering solutions for some. They include, a) concurrent modification exceptions which this work believes are related to the failure to increment a semaphore counter in SimJava, b) a missing novel TreeMap class or Comparator, c) unresolved Out-Of-Memory Error, and d) inaccurate RTT values. Other problems identified by this work are the incorrect post configuration routing table entries of the Router class.

For greater accuracy in simulation some tools resort to single thread discrete-event simulation engines. Despite these problems, GridSim found application as the core in

other simulation tools like Alea2, GSSIM and CloudSim. However, CloudSim removed its dependency on the SimJava discrete event simulation library. There are few alternatives to GridSim that offer similar features. The SimGrid simulation tool has been routinely extended to have many of the features of the older GridSim toolkit. SimGrid has also been studied to improve the accuracy of its flow-level network simulations.

GridSim is chosen because it was developed to perform distributed and cluster collaborative grid computational simulations. The datagrid extension has many of the tasks required for the simulation of the distribution of SKA science products in a WAN. If bringing multiple ports and the notion of equipment types in GridSim is achieved, then GridSim would be a plausible candidate for the simulation of energy models like ECOFEN.

3 Methodology

In this Section, the configuration of the data grid network topology for the Square Kilometer Array (SKA) simulation is explained (Aikema, 2010). Described are invalidation exercises for the confirmation of; 1) packet loss in SimJava; and 2) failure of shortest path first routing in GridSim. The invalidation exercises use a smaller dumbbell network topology for this purpose. A design for implementing a prototype of the ECOFEN model is used to demonstrate the use of multiple ports in GridSim.

The parameters in GridSim have predefined units of measure which makes simulations in GridSim less scalable than in other simulators like NS2 which do accommodate some degree of unit conversion for the configuration of their parameters. A change to the SKA simulation can be done to accommodate industry standard network constants for some parameter values with a corresponding unit of measure. Changing the unit of measure would change the scale of the simulation in GridSim.

3.1 Research approach

Efficient algorithms, abstraction of the real world, and encouraging interdisciplinary cooperation enhance the pure science content of computing research (Gupta & Rani, 2019). This dissertation places value in the experimental research method which frames computing science as an applied science. This work performs an empirical study using a software simulation model of the SKA data grid environment for the transfer of science-ready astronomical data over a large Round-Trip-Time (RTT) wide area network or LFN.

3.1.1 Research Method

This dissertation configures a data grid network in GridSim to simulate the transfer of data from one or more data centres over greater than 100Gbps network linkages to distributed SKA regional centres, Figure 5. The computer system components for hosting the GridSim simulation are listed in Table 3. The simulation uses GridSim's loader classes to configure the simulation resources from the set of flat files the researcher provides which contain the necessary parameters for the simulation. The GridSim file loaders separate parameter values from the source code to avoid having to recompile the source code whenever a parameter is changed. Furthermore, arranging configuration files in collections is useful for categorizing simulation scenarios which

the GridSim simulation can retrieve from settings in its parameter.txt file.

Table 3

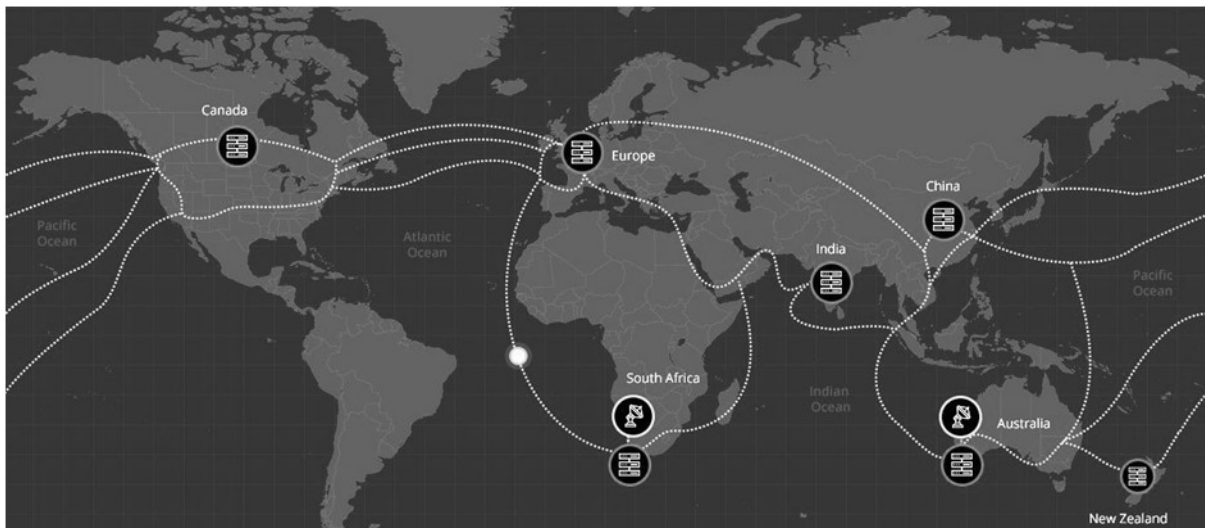
Machine host for GridSim.


Processor	Intel i5 2500S incl. msr support for RAPL
Memory	8 GB, 1600 MHz DDR3
GPU	On chip
Operating System	Linux CentOS 7
Shell	Bourne shell
Language support	Java version 8

The network topology for the simulation, models the Global SKA Regional Center network, which is the ten-year IRU (an Indefeasible Right to Use) >100 Gbps leased network shown in Figure 3. The simulation sends data from the South African or Australian Science Processing Centers (SPC) to the regional centres located in the United Kingdom, Europe, India, Canada, China, Australia, and New Zealand (Aikema, 2017, Huyn, 2019), participating in the SKA WAN (Figure 5).

Figure 3.

The Square Kilometre Array Collaborative Wide Area Network



Note. The astronomical pathfinders of Carnarvon, South Africa and Murchison, Australia supply science-ready data for the SKA Regional Centres. Diagram by Huyn (2019)  from

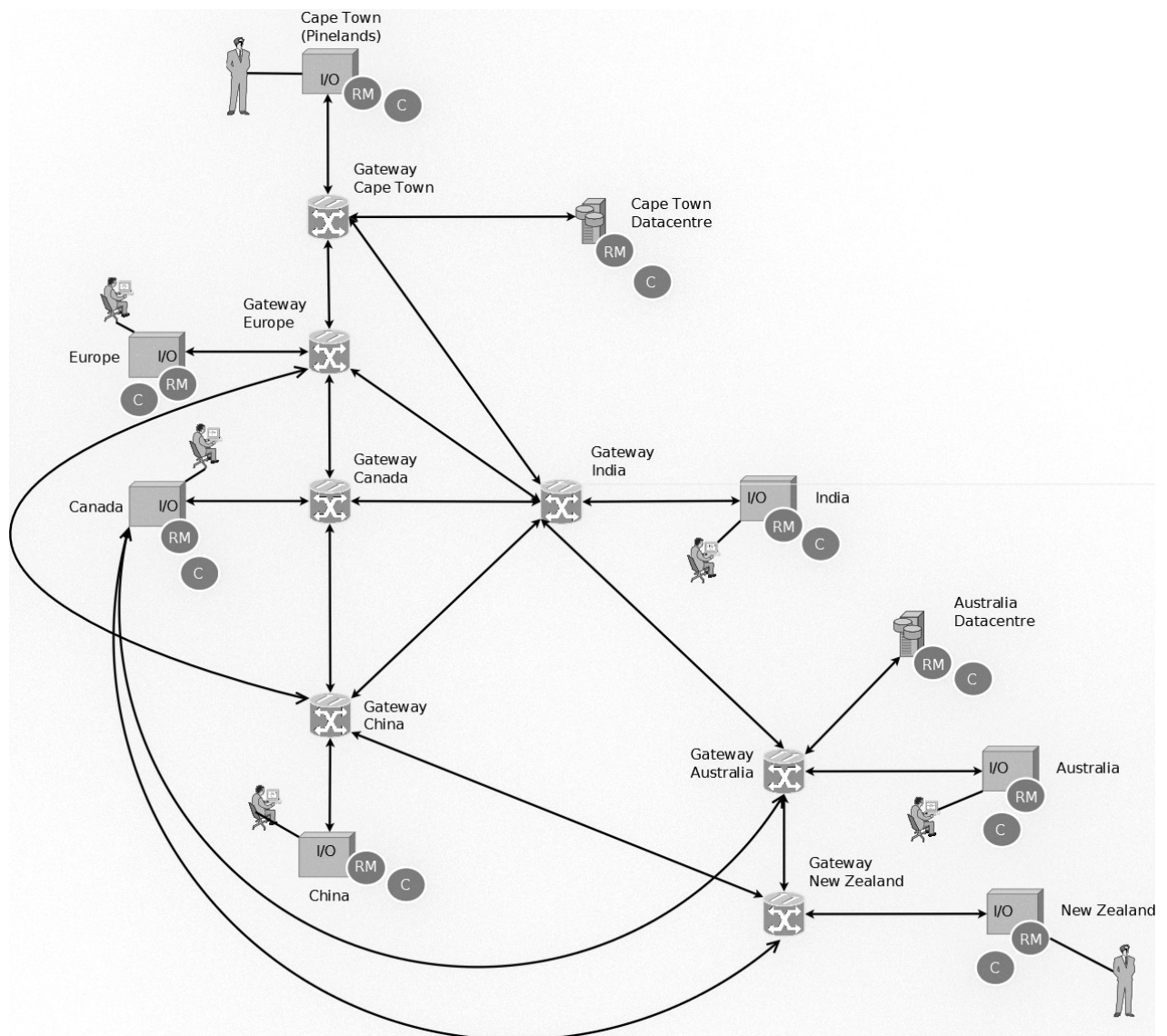
the publication "The Global SRC network for the Square Kilometer Array (SKA1 and SKA2) for the period 2017–2023".

The following are some of the characteristics of the SKA data grid simulation:

- Grid users occupy the regional science facilities, they request science-ready data from Cape Town and the New Zealand regional catalogues.
- Catalogues at Cape Town and New Zealand are the only stores of the master file.

Figure 5.

The simulated SKA Wide Area Network



Note. RM = Replica Manager; C = Replica Catalogue; I/O = Input & Output buffered ports. NetIO port communication between RM and its GridResource. Buffered I/O packet communication between linked nodes of the WAN. Not shown are the precursor telescopes and processing facilities.

- Requests for astronomy data products from collaborating regional centres receive replicated copies of the master file.
- GridSim has the notion of a global replica catalogue (RC) serving as a common store of replicated data in the grid. If a grid resource does not have a local regional catalogue the file is registered with the global RC.
- Users from one region may request data from another region, provided the data exists in the RC of the data grid resource. The “get” task retrieves the file from the specified grid resource without keeping a copy; the “replicate” task retrieves the file but stores a copy in the data grid’s regional RC. Other data grid tasks include “getattribute”, “delete”, and the catalogued files are modifiable.
- Only the network packets of the datagram, transferred between the file storage location and the user’s data grid resource, are applicable for the energy model.

3.2 The Invalidation Experiments

Invalidation experiments test the implementation of algorithms in the software. This work uses invalidation experiments to confirm whether a problem exists in the GridSim toolkit that could be a problem for performing network simulations. Here we describe the work to confirm packet loss and the shortest path routing protocol.

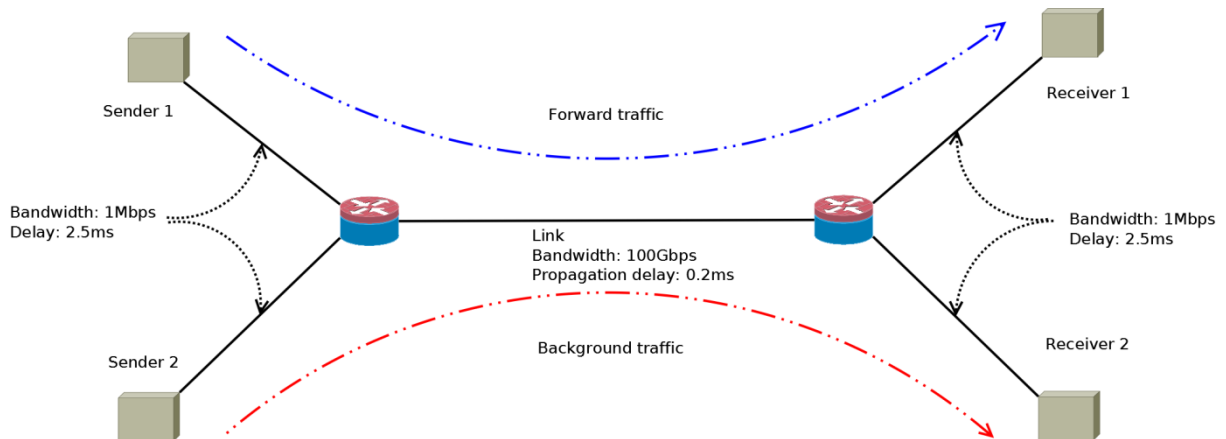
3.2.1 Packet loss

The small MTU-sized packets arising from fragmentation contribute to the accuracy of a network simulation. In this invalidation experiment, packets sent by the Output port of the sender do not all arrive at the receiver’s Input port in the dumbbell network topology for the experiment (Figure 6). The number of packets is limited to two in-flight packets, without background traffic; the size of the packet for the ping is large enough to create two

datagrams, one MTU-sized packet (an empty packet) and the original packet with an updated size which is the remainder i.e. $\text{size} < \text{MTU}$. The empty packets in GridSim simulate the overall transmission and propagation delays for the flow.

Figure 6.

The dumbbell network



The experiment shows that the simulation fails to process the empty packet when it is transferred from the Future Event List (FEL) to the event buffer (evbuf) of the receiving entity. Although, when the original event is transferred from the FEL to the Deferred Event List (DEL), the entity state changes from WAITING to RUNNABLE. The SimJava synchronization mechanism restarts all entities in a RUNNABLE state, i.e. ends the wait state. The active entity then takes an event from the DEL, which overwrites the contents of the evbuf previously updated with the empty packet event values.

An event is lost when,

- 1) there is more than one event in the FEL for the receiver,
- and
- 2) the first and next FEL events have matching event execution times

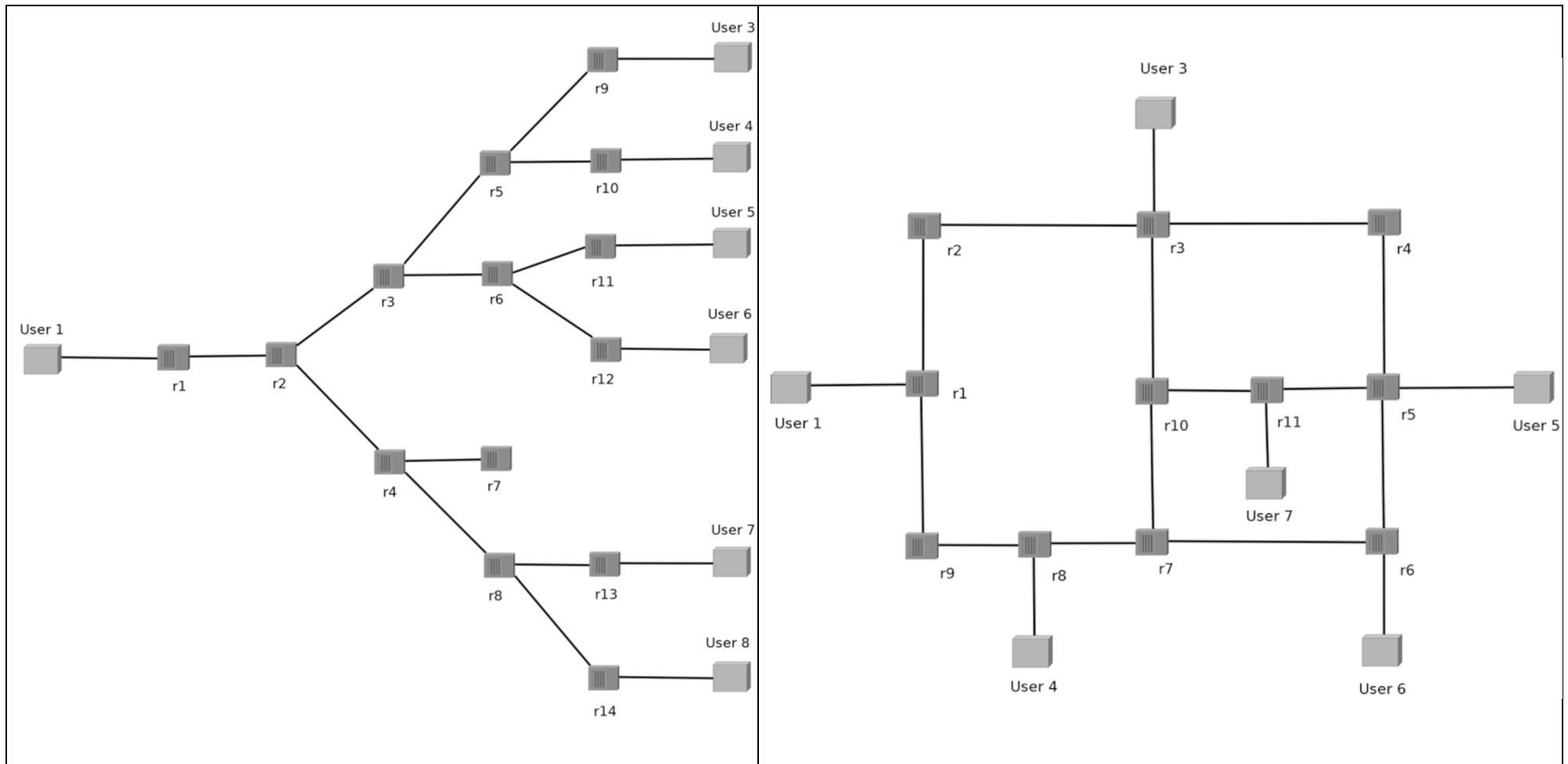
The proposed solution is to add a condition to SimJava's `sim_entity.get_next()` method which tests the evbuf for content. When true, the content of the evbuf is processed before the buffer will accept the next event from the DEL.

3.2.2 GridSim, shortest path invalidated

In GridSim, network transfers follow the shortest path stored in the routing tables from sender to receiver. The network topology of the SKA WAN is a tree type topology, i.e. each node can have more than one route to other nodes in the network. The invalidation experiment uses the tree network topology (Figure 7a) and the graph network topology (Figure 7b). The same configuration settings are used for both topologies; Bandwidth 10 Mbps, Propagation delay 0.5s, Scheduler baudrate 10 Mbps, MTU size 1500 bytes. The simulation of a ping request is then repeated 500 times for both topologies with the output of the ping request written to a flat file for analysis.

Figure 7.

Network topologies for the invalidation of shortest path first in GridSim



(a) Tree network for the routing problem

(b) Graph network for the routing problem

3.2.3 The proposed design for ECOFEN in GridSim

The number of ports in a GridSim resource cannot be more than one Input and one Output port. The problem is allegedly caused by a bug in GridSim's SimJava library, "*This is also due to SimJava's definition of ports, i.e. they are unidirectional (1-way) communication links*" (Sulistio, 2008). However, the simulation of multiple ports is a prerequisite for the implementation of the ECOFEN energy model prototype in GridSim. ECOFEN uses both Adaptive Link Rate (ALR) and on/off algorithms to manage the energy consumed by network equipment like gateways, Network Interface Cards (NICs), routers etc.

GridSim implements some elements of the User Datagram Protocol (UDP) for datagrams of a user-defined size (Casanova, 2007). In 2008, Caminero et al. added buffers and buffer management policies to GridSim for the future implementation of the Transmission Control Protocol (TCP). However, the current version of the GridSim toolkit does not simulate TCP nor TCP-based GridFTP (Sulistio, 2007), both of which are used by the SKA collaboration platform.

3.2.3.1 ECOFEN implementation criteria. The SKA WAN simulation topology, in Figure 5, has a total of 5 regional customers. The data centre user in Cape Town and the data centre user in New Zealand perform catalogue related tasks. The implementation of ECOFEN requires: a) the energy consumption values e.g. ALR energy, min-max energy (in watts), and chassis energy (in watts), of equipment are defined by, the type of equipment, the brand and model; b) the values of energy consumption are the measurements taken from readings of the equipment or provided by the manufacturer; c) router ports are full power i.e., when there is cross-traffic (multiple flows) the port energy consumption is the equipment power divided by the number of flows and d) the energy of a port in a NIC uses the Adjustable Link Rate (ALR) algorithm. For example, Table 4 lists the energy profiles of ports in a NIC. Accordingly, the ALR for a link of bandwidth of 20Mbps is 100Mbps ($10\text{Mbps} < 20 < 100\text{Mbps}$) and the energy consumption for the ports is 0.18W.

Table 4.

The energy profile of a NIC port

NIC (brand, model, version, etc.)	Baudrate in Mbps	Power in Watt
Chassis power (constant)	-	140
Port	10	0.12
	100	0.18
	1000	0.87
	10000	1.9

Note. The port can support any of the transmission rates. The ports of the NIC use the same energy profile table (Orgerie, 2015).

3.2.3.2 ECOFEN in GridSim. The prototype for the SKA simulation configures multiple NIC ports for the classes which define GridResource and GridUser objects, respectively. Network linkages attached to a router are treated as if they are a particular active port of the router. The configuration files, parameter.txt, users.txt, network.txt (Figure 8) and resources.txt (Figure 9) are updated with an appropriate equipment identifier. The equipment identifiers are defined in a new configuration file, equipment.txt (Figure 10). The various equipment might include hardware such as routers, NICs and other equipment and their respective energy metrics. A new file loader class, Equipment.java, parses the equipment.txt configuration file which caches its contents in a static HashMap. Shown in Figure 10 are energy attributes for the equipment to be used by the ECOFEN simulation prototype. Other equipment types e.g., bridges, switches, repeaters, hubs and firewalls, may be included later to extend the current list of equipment and energy attributes. The configuration files resource.txt, network.txt and users.txt must be updated to include the unique equipment identifier. In addition, their respective file loaders, shown in Figure 11, are also updated to lookup configured equipment from the cached list. There is no scripting language, like Jython or a similar language, employed in GridSim to facilitate the configuration of parameters into objects for code injection (for example, Jython's execfile command) into Java's JVM at runtime.

3.2.3.3 Code updates. The design concepts for the implementation of multiple ports in the network extension are shown in Figure 12. The GridSim class extends GridSimCore which extends SimJava's Sim_entity class. The GridResource class extends the GridSimCore class unlike some other entities, like the Users class which extend the

GridSim class. To implement multiple ports in the GridSim toolkit the GridSimCore is extended with attributes and methods to setup the multiple ports which in turn get their settings from lines of source code or from the resource file loader, "ResourceFileLoader". The Router class needs to be port aware for the implementation of ECOFEN model because it extends from SimJava's Sim_entity class. The number of ports is set by the configuration parameters for the router. The number of ports cannot be less than the number in the router's linkTable; when the number of linkages is greater than the excess is treated as idle ports.

Figure 8.

Example of a network configuration file, network.txt

```
# The Network topology.
# N routers
# N RouterName(s)
# RouterName sender, RouterName receiver, baud rate (Bytes/s), propDelay (sec), mtu (bytes), [optional link_{min,max}_power], log
2
router1
router2
router1 router2 12500000000 1.0 9216 true
```

```
# The Network topology.
# N routers
# N RouterName(s) EquipmentIdentifier
# RouterName sender, RouterName receiver, baud rate (Bytes/s), propDelay (sec), mtu (bytes), [optional link_{min,max}_power], log
2
router1 EquipTypeRaw
router2 EquipTypeRaw
router1 router2 12500000000 1.0 9216 true
```

Note. The framed section is the revised configuration, showing the changes in red.

Figure 9.

Example of the updated grid resource configuration file, resource.txt

```
#the description of resources in a grid
#the format of the file:
#resource_name storage_size(in GB) bandwidth(in MByte/s) link_{min,max}_pwr EquipmentIdentifier router_name list_of_files
#
Res_0 99999 12500000000 {1,1} NIC_0 router1 fileM file1
Res_1 99999 12500000000 {1,1} NIC_0 router3 fileH file2 fileG
Res_2 99999 12500000000 {1,1} NIC_2 router4 fileF fileK fileD
```

Note. In the SKA data grid, the “EquipmentIdentifier” refers to the NIC of the data grid resource (regional catalogue) tasked with processing all tasks to do with the data grid file (representing astronomical data) using a NIC port.

Figure 10.

Example of the *new equipment.txt configuration file*

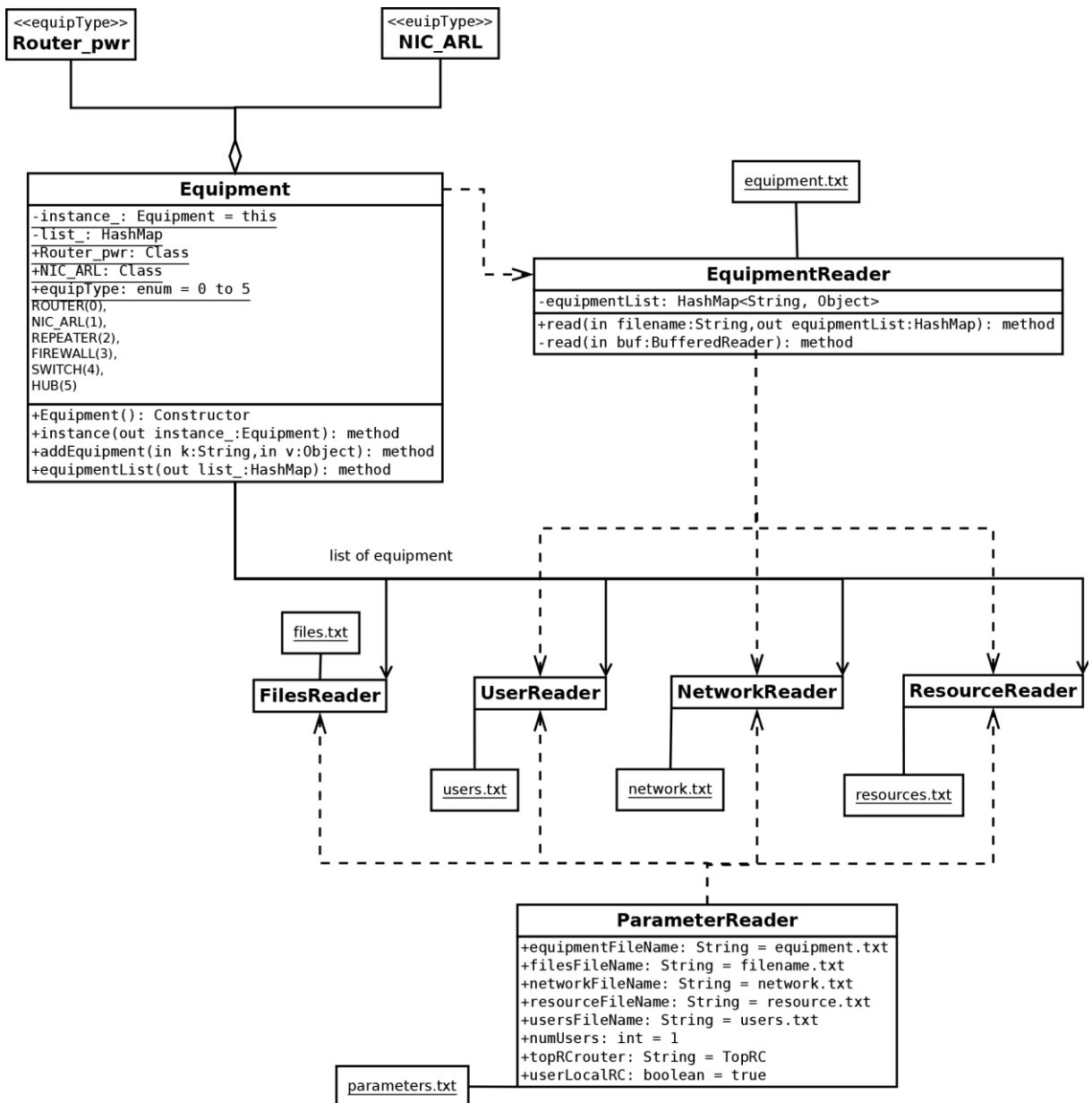
```
# Equipment Energy attributes
#
# For routers in the topology
# Router_EquipmentIdentifier Brand, Model, num_ports, {idle_power:work_power}
#
# For Adaptive Link Rate (ALR)
# NIC_EquipmentIdentifier Brand, Model, num_ports, chassis_power, list_of_{baudrate:unit;power}
#
#Routers
EquipmentRaw_1 Brand_Z Model_A 9 {0.1:0.8}
EquipmentRaw_2 Brand_Z Model_B 9 {0.1:0.8}
#NIC_ALR
NIC_0 Brand_X Model_0 9 146 {10:Mbps;0.12} {100:Mbps;0.18} {1:Gbps;0.87} {10:Gbps;0.99} {100:Gbps;1.2} {120:Gbps;1.5} {150:Gbps;1.9}
NIC_1 Brand_X Model_1 9 146 {10:Mbps;0.12} {100:Mbps;0.18} {1:Gbps;0.87} {10:Gbps;0.99} {100:Gbps;1.2} {120:Gbps;1.5} {150:Gbps;1.9}
NIC_2 Brand_Y Model_0 9 146 {10:Mbps;0.12} {100:Mbps;0.18} {1:Gbps;0.87} {10:Gbps;0.99} {100:Gbps;1.2} {120:Gbps;1.5} {150:Gbps;1.9}
```

Note. The file loader handles various categories of equipment according to the predefined tag preceding the configuration definitions e.g. #Routers, #NIC_ALR[†]

[†]ALR = Adaptive Link Rate

Figure 11.

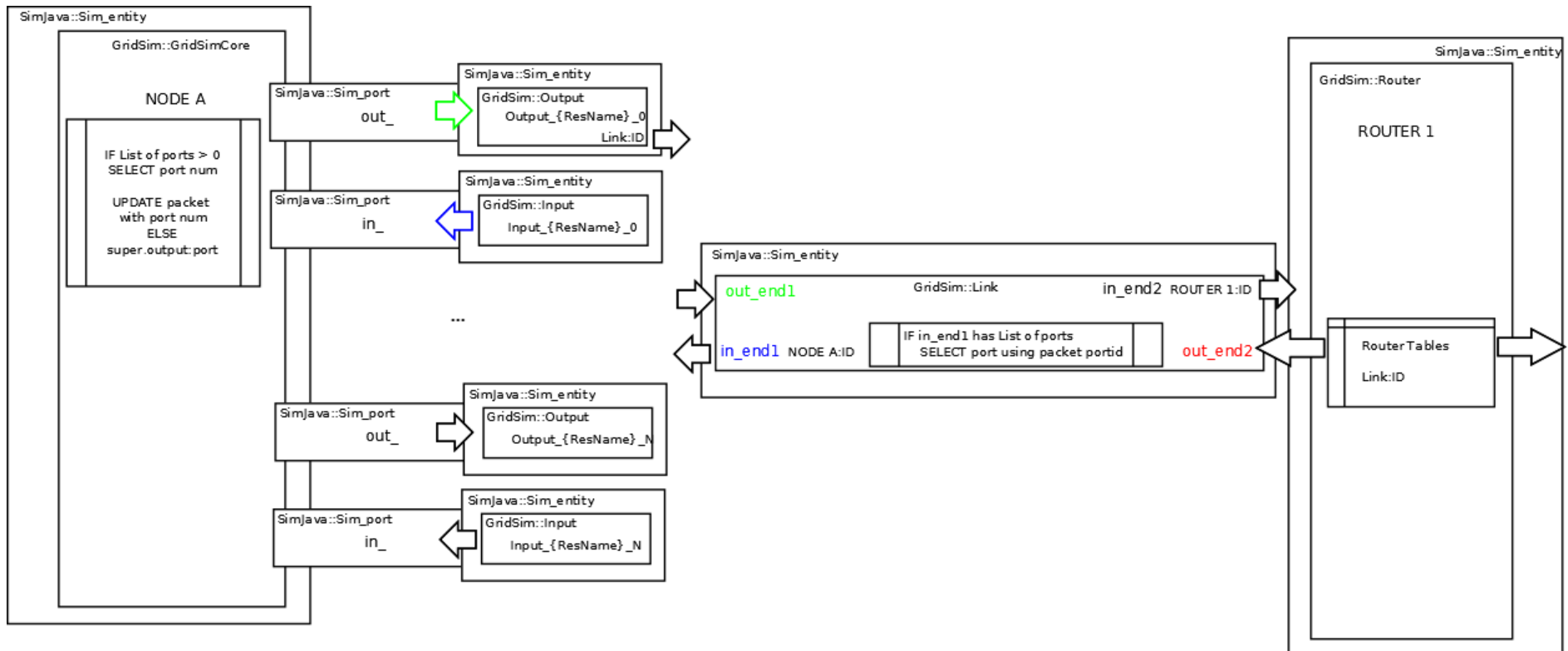
Association of the equipment file loader and other GridSim file loaders



Note. The list of equipment is a static Java Map of equipment identifier and equipment object, it is used solely for configuring the various resources.

Figure 12.

Network extension object associations for multiple ports

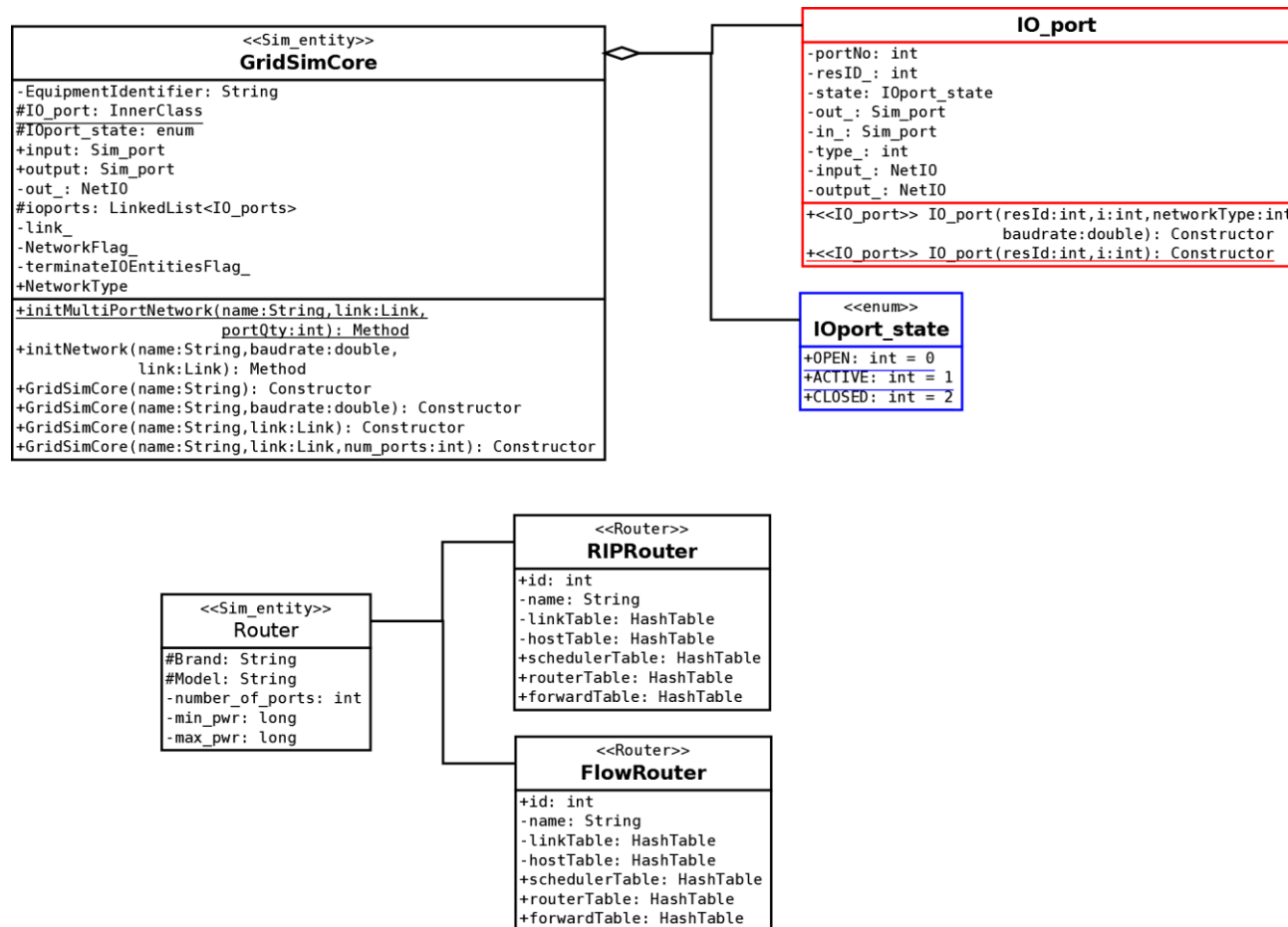


Note. n.d.

The port configuration for the GridSimCore class is updated to include a list of ports of the type of IO_port which is an inner class in GridSimCore, see Figure 13. The IO_port is assigned a state, defined by the enum IOport_state with the values OPEN, ACTIVE and CLOSED. A novel initMultiPortNetwork method, creates N number of IO_port objects inserting them to the ioports list. The “out_” and “in_” Sim_port attributes associate the GridSim buffered Output and Input objects with the SimJava API layer. The GridSimCore object defaults to the single input/output Sim_port configuration when the list of ioports is empty i.e. the entity is not configured with multiple ports. The various entities for the simulation receive their respective energy attributes when the simulation starts.

For multi-port equipment, the ports are loaded from a list of IO_ports with the state of the port set to “open”. The port’s corresponding receiver port is also in an “open” state. A receiver without multiple ports does not open a port; the default port is used. The state of the “out_” port, switches to active when packets begin their transfer from the port to the link. The state of the “in_” port changes to the active state when the first packets arrive at the port. The applicable ports close when the all the packets have passed from the sender to the receiver. In GridSim, the receiver returns a datagram to the sender to acknowledge the success or failure of the task, sometimes including additional information or data. If configured, the return packet will use the port with a matching port number or the next available port.

Figure 13.

Multiple port class in GridSim

Note. IOport_state and IO_port are inner classes available to any child of the GridSimCore class, including the GridUser, GridResource and GridSim classes.

3.3 Summary

This Section described the configuration of the SKA Wide Area Network simulation. Multiple port support is a requirement for energy model engines like ECOFEN. However, the literature claims multiple port support in GridSim is not possible because of a bug in the SimJava library. A design for the future implementation of Multiple ports in GridSim's network extension is proposed. The design also describes the implementation for communication energy metrics associated with various types of equipment as would be needed to model the power consumption of communication over the SKA WAN. In addition, two invalidation exercises are employed to show a scenario where unexpected packet loss occurs and to demonstrate a problem with the GridSim's implementation of the shortest path first RIP, respectively. The invalidation exercises used a dumbbell network topology. A solution for the packet loss problem in GridSim is given, but no solution for the shortest path routing problem has been found. Vigilance is, therefore, necessary when performing complex network simulations.

The implementation of the ECOFEN model in GridSim is planned for a future date after the failure of an initial attempt.

4 Results

Source code inspection as a strategy to find problems in GridSim's network extension yielded a few results. The CTADetector automated tool found no problematic Check-Then-Act idioms in GridSim's programming. Other problems in the programming of GridSim can cause unexpected simulation failures, they include (i) Incorrect ordering of events in SimJava's future event queue. This is resolved by restoring the LinkedList Evqueue or defining a Comparator for the new FutureEvqueue class; (ii) Adding entities dynamically to a simulation. To resolve failures, ensure the SimJava Semaphore objects are updated; Incorrect entries are inserted into the routing tables; Fragmentation of packets does not model the additive impact of header size of new in-flight packets.

Invalidation of GridSim's RIP implementation found that for data sent from user1 to user6 there was no variation in the results of the tree-topology network (Figure 7a).

500 Data sent successfully. Duration of simulation is 828.8544

500 Path taken: router_1 >> router_2 >> router_4 >> router_7 >> router_13

Initially, there was no variation either in the results of the graph topology network (Figure 7b). Both simulations completed in 828.8544 seconds with the shortest path of 4 hops.

500 Data sent successfully. Duration of simulation is 828.8544

500 Path taken: router_1 >> router_9 >> router_8 >> router_7 >> router_6

However, an unexpected result in the graph topology simulation was observed when the data was sent from sender user6 to receiver user7. Of the 500 simulations performed, 35 pings followed a longer path of 7 hops instead of the shorter path of 6 hops.

465 Data sent successfully. Duration of simulation is 815.265

465 Path taken: router_1 >> router_2 >> router_3 >> router_4 >> router_5>>router_11

35 Data sent successfully. Duration of simulation is 815.6756

*35 Path taken: router_1 >> router_9 >> router_8 >> router_7 >> router_6 >> router_5
>>router_11*

Inspection of the routing tables found invalid hop count values in the forwarding table which causes the incorrect path and scheduler to be selected. The maximum hop count for the RIP is 15 hops, in the GridSim toolkit the maximum number of hops can be updated when necessary. There is no routing problems mentioned in the available literature. The tree-topology network was unaffected by the routing problem.

The packet loss experiment shows a failure of the simulation to process the empty packet transferred from the Future Event List (FEL) to the event buffer (evbuf) of the receiving entity. The empty packet is lost when the original event is transferred from the FEL to the Deferred Event List (DEL) and the entity state changes from WAITING to RUNNABLE. The SimJava synchronization mechanism restarts all entities in a RUNNABLE state, i.e. ends the wait state. The active entity then takes an event from the DEL, which overwrites the contents of the evbuf previously updated by the empty packet.

An event is lost when,

- 1) there is more than one event in the FEL for the receiver,
and,
- 2) the first and next FEL events have matching event execution times

A quick solution is to add a condition to SimJava's `sim_entity.get_next()` method to test the evbuf for content. When true, process the evbuf contents prior to the buffer accepting the next event from the DEL.

Implementation of the ECOFEN model is dependent on the simulation of multiple network equipment ports in GridSim. From the analysis of source code, a design for the future implementation of the ECOFEN model in GridSim was proposed. For the future implementation of the model in GridSim some of the problems with the network extension might need to be resolved, if they are not to be an encumbrance for the simulation. Some performance problems in GridSim might not have a solution, such as

synchronising the shared resources in large simulation and selection of RIP shortest path first. Simulation in GridSim is process based which for packet-level simulation of large data grid networks such as SKA will perform poorly at scale. The flow-level network simulation promises better performance but does not have the accuracy of the packet-level network simulation. There are also limitations with respect to the representation of large data sizes in GridSim, which need to be addressed using mechanisms that can simulate the scale of the real-world system. Improving the accuracy of flow-level simulation in GridSim for the ECOFEN model is necessary to ensure confidence in the simulation output. The alternative is to use another simulation tool, for example SimGrid, to implement the ECOFEN model. Improvement and validation of the SimGrid simulator appears to be a feature throughout its lifespan. The implementation of the proposed design of the ECOFEN extension in GridSim is marked for future implementation after the failure of a first implementation attempt.

Problems with the GridSim's network extension are listed in Table 5.

Table 5.

A summary of some problems affecting the GridSim toolkit

- The packet level simulation is not scalable.
- RTT does not change with a change in link MTU size.
- The flow level simulates bandwidth sharing but assigns incorrect bandwidths to its flows.
- Packets in a tree topology network do not always follow the shortest path.
- The first packet in the FEL is lost with each iteration.
- Network simulation is coarse-grained.
- UDP like network.
- Routers do not simulate ports.
- Resources have one Input/Output port.
- Flow level values are incomparable to Packet level values.

5 Conclusion and Future Work

The GridSim toolkit extends the SimJava discrete event library to simulate a grid network in computational clusters and distributed networks. Available literature confirms that GridSim has some unresolved problems. The ones that are important for this work include (a) not routing messages correctly in a graph network topology; (b) incorrect band-width sharing in a flow-level network; (c) not modelling the additive effect of fragment headers on data volumes; (d) losing the first packet from the future queue with every tick of SimJava's simulated clock. If not resolved network latency will not be calculated correctly which in turn will determine the power needed for data transfers. The performance of process-based simulators, like GridSim, are known to deteriorate when there has been an increase in simulation complexity or there are more messages in-flight. To this end, attempting to simulate the largest collaborative big data challenge known to science, will have to use the flow-level network in GridSim and a mechanism to scale the real-world values used for its configuration.

The claimed lack of support for multiple ports in GridSim suggests that the toolkit is not suitable for the implementation of the ECOFEN model. The design of a solution to simulate multiple ports in GridSim is proposed here in text, UML and Object diagram. The contents of GridSim configuration files are flat text, despite there being better alternatives for representing configuration parameters such as the Python language. The proposed design will not be good for GridSim performance because every port in the multi-port configuration is a Thread. Increasing the number of Threads in GridSim for front-loading increases the chance of Synchronization exception in GridSim. Although, synchronization errors should be mitigated to some degree by the fixes proposed in this work. The first, resolved the ordering of messages in SimJava's event queue. The second, updates the Semaphore objects when new resources are created dynamically by the simulation.

The implementation of the ECOFEN model in GridSim and the SKA WAN is for future work. Future work must validate the accuracy of simulation. For instance, do events sent in parallel in a GridSim pseudo-parallel simulation reach the destination at the same time? This is an open question that requires further experimentation.

Despite some of the benefits GridSim offers the modeller, there are alternative

simulation tools that are better suited to more detailed network simulation, such as SimGrid. The accuracy of GridSim simulation was a concern raised in the literature. In this work the discrepancy in RTT between a packet-level and flow-level network was noted. If the simulation of the SKA-WAN is to continue using the GridSim toolkit then improving the accuracy of simulation warrants further investigation and improvement.

References

- Achyryja, D. P. & Kauser, A. P. (2016). A survey on big data analytics: Challenges, open research issues and tools. *International Journal of Advanced Computer Science and Applications*, 7(2)
- Adineh, H. (2012, July 28). *Exception in thread "AWT-EventQueue-0" java.lang.OutOfMemoryError: Unable to create new native thread -> ERROR [GridSim-users].Sourceforge.* <https://sourceforge.net/p/gridsim/mailman/message/29613325/>.
- Aikema, D. (2017). *Data transport for Meerkat and SKA regional centres.* Unpublished manuscript.
- Bąk, S., Krystek, M., Kurowski, K., Oleksiak, A., Piątek, W. & Wağlarz, J. (2011). GSSIM – A tool for distributed computing experiments. *Scientific Programming*, 19, 231-251. <https://doi.org/10.3233/SPR-2011-0332>
- Bambrik, I. (2020). A survey on cloud computing simulation and modeling. *SN Computer Science*, 1(5), 249. <https://doi.org/10.1007/s42979-020-00273-1>
- Bedwal, T., Tayal, R. & Batra, A. (2014). Grid computing and GridSim toolkit: An overview. *International Journal of Computer Applications*, 17-20.
- Bell, W. H., Cameron, D. G. & Capozza, L. (2003). Optorsim - a grid simulator for studying dynamic data replication strategies. *International Journal of High-Performance Computing Applications*, 17(4), 403-416.
- Bonaventure, O. (2011). *Computer networking.* Saylor Foundation.
- Broberg, J. (2008, February 18). *[GridSim-commits] SF.net SVN: GridSim: [113] branches/gridsim4.0-branch2/source/gridsim/net/flow/FlowRouter.java.* Sourceforge. <https://sourceforge.net/p/gridsim/mailman/message/18620803>.
- Broberg, J. & Buyya, R. (2018). Flow networking in grid simulations. *IEEE/ACM International*, <https://doi.org/10.1201/9781420067682-c15>
- Buyya, R. (2015). DCA-services: A distributed and collaborative architecture for conducting experiments in service-oriented systems. *International Journal of Services Computing*, 3, 14-28.
- Buyya, R., Abramson, D. & Giddy, J. (2000-05). Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid. Paper presented at the *High-Performance Computing in the Asia-Pacific Region, International Conference on (2000)*, Beijing, China.
- Buyya, R., de Assunção, M. D., Garg, S. K. & Javadi, B. (2002). *GridSim: A grid simulation toolkit for resource modelling and application scheduling for parallel and distributed computing.* <http://www.cloudbus.org/gridsim/>.

- Buyya, R., de Assunção, M. D., Garg, S. K. & Javadi, B. (2009). *Roadmap*. Unpublished manuscript. <https://cloudbus.org/gridsim>
- Buyya, R. & Murshed, M. (2002). GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurrency and Computation: Practice and Experience*, 14, 1175-1220. <https://doi.org/10.1002/cpe.710>
- Buyya, R., Ranjan, R. & Calheiros, R. N. (2009). Modelling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities. Paper presented at the *7th High Performance Computing and Simulation Conference*,
- Buyya, R. & Sulistio, A. (2008, April 13-16). Service and utility oriented distributed computing systems: Challenges and opportunities for modelling and simulation communities. Paper presented at the *Simulation Symposium, 2008. ANSS 2008. 41st Annual*, <https://doi.org/10.1109/ANSS-41.2008.35>
- Cameron, D., Carvajal-Schiano, R., Millar, A., Nicholson, C., Stockinger, K. & Zini, F. (2010). OptorSim: A simulation tool for scheduling and replica optimisation in data grids. *Computing in High Energy and Nuclear Physics*,
- Caminero, A., Sulistio, A., Caminero, B., Carri, C. & Buyya, R. (2008). Simulation of buffer management policies in networks for grids. Paper presented at the *Proceedings - Simulation Symposium*, <https://doi.org/10.1109/ANSS-41.2008.6>
- Casanova, H. & Fujiwara, K. (2007, October 22). Speed and accuracy of network simulation in the SimGrid framework. Paper presented at the *Proceedings of the First International Workshop on Network Simulation Tools (NSTools)*, Nantes, France, 1-10.
- Casanova, H., Schnorr, L., Velho, P. & Legrand, A. (2011). *Flow-level network models have we reached the limits*. (No. 7821). HAL. <https://hal.inria.fr/hal-00646896>
- Casanova, H., Velho, P., Legrand, A. & Schnorr, L. (2013). On the validity of flow-level TCP network models for grid and cloud simulations. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 23(4), 1-26. <https://doi.org/10.1145/2517448>
- Casanova, N., Velho, P., Schnorr, L. M. & Legrand, A. (2013, October). A fast and accurate network simulation. Paper presented at the *NII Seminar Series*, <https://www.nii.ac.jp/userimg/lectures/20131010/Lecture5.pdf>
- Chelladurai, S. R. (2016, November). *GridSim: A flexible simulator for grid integration study*. Core. <https://core.ac.uk/download/pdf/84872859.pdf>
- Chen, K. (2015). OMNeT++. (pp. 81-127). <https://doi.org/10.1002/9781119006190.ch6>

- Chen, W., Wang, D. & Zheng, W. (2005). A single thread discrete event simulation toolkit for java: STSimJ. Paper presented at the *Computational Science - ICCS 20*, 1131-1137.
- CLOUDS lab. (2010, November 26). *Readme*. Buyya.
<http://www.buyya.com/gridsim/doc/readme.txt>
- Cunha, G. (2008, June 11). *Re: [GridSim-users] how create machines during the simulation*. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/19698099>.
- Curiel, M., Alvarez, G. & Flores, L. (2007). Evaluating tools for performance modelling of grid applications. *Frontiers of high-performance computing and networking – ISPA 2006 workshops* (pp. 854-863). Springer Berlin Heidelberg.
https://doi.org/10.1007/11942634_87
- Dakkak, O., Arif, S. & Nor, S. A. (2015). A critical analysis of simulators in grid. *Jurnal Teknologi*, 77(4), 111-117.
- de Assunção, M. D. (2008a). *How the clock of the gridsim is increased? [GridSim-users]*. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/20924695>.
- de Assunção, M. D. (2008b). *Re: [GridSim-users] many simulations with only a java process*. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/21212365>.
- de Assunção, M. D. (2009a). *GridSim_ A grid simulation toolkit _ re_ [GridSim-developers] [GridSim-users] strange error*. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/21576216/>.
- de Assunção, M. D. (2009b). *Re: [GridSim-users] delaying the shutdown of grid entities re: [GridSim-users] delaying the shutdown of grid entities*
- de Assunção, M. D. (2009c). *Source code for simjava? [GridSim-users]*.
<http://www.inf.ed.ac.uk/research/isdd/admin/package?download=62>.
- Deeve, N. V., Vijesh, J. C. & Narmatha, K. (2015). Study on benefits of green computing. *International Journal of Current Research*, 7(4), 14442-14445.
<http://www.journalcra.com>
- de Frutos, H. B. (2009). *[GridSim-users] problems with communications [GridSim-users] problems with communications*. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/23882279>.

- Dumitrescu, C. & Foster, I. (2005). *GangSim: A simulator for grid scheduling studies*.
<https://doi.org/10.1109/CCGRID.2005.1558689>
- Eng, M. (2016, July 18). *GridSim event timing problem [GridSim-users]*. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/35226458/>
- Ernst, M. & BNL/ATLAS. (2015). The changing face of networks and implications for future HEP computing models. Paper presented at the *CHEP*, Okinawa.
- Giancarlo (2009, Feb 12). Strange error [GridSim-users]. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/21562721>
- Gomes, I., Morgado, P., Gomes, T. & Moreira, R. (2009). *An overview on the static code analysis approach in software development*. <https://pdfs.semanticscholar.org>.
- Grace, R. K., Surya, S. & Priya, S. (2012). A survey on grid simulators. *International Journal of Computer Science and Information Technology & Security*, 2(6)
- Gupta, D. & Rani, R. (2019). A study of big data evolution and research challenges. *Journal of Information Science*, 45(3), 322-340.
<https://doi.org/10.1177/0165551518789880>
- Harshadkumar, P. & Shah, V. (2014). Scheduling in Grid Computing Environment. *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, 315-324. <https://doi.org/10.1109/ACCT.2014.32>
- Hedrick, C. (1988). *Rfc1058*. Unpublished manuscript. Retrieved Dec 1, 2021,
<http://www.ietf.org/rfc/rfc1058.txt>
- HL-LHC network needs and data transfer challenges (n.d.). Indico Cern.
https://indico.cern.ch/event/983436/contributions/4226012/attachments/2213578/3746893/HL-LHC_network_challenges-V2.pdf, accessed: 2025-03-20.
- Hojiyev, S. (2012, November 3). *How to manage time in GridSim [GridSim-users]*. Sourceforge. <https://sourceforge.net/p/gridsim/mailman/message/30051010>
- Huyn, M. (2019, November 4). The square kilometer array computing - from SDP to SRC. Paper presented at the *24th International Conference on Computing in High Energy Nuclear Physics*,
<https://indico.cern.ch/event/773049/contributions/3581362/attachments/19377>
<https://doi.org/10/3211780/SKA>
- Ilyas, B. & Elkhailifa, I. (2016). Static code analysis: A systematic literature review and an industrial survey.
- Issariyakul, T. & Hossain, E. (2012). Introduction to network simulator 2 (NS2). In T. Issariyakul, & E. Hossain (Eds.), *Introduction to network simulator NS2* (pp. 21-40). Springer US. https://doi.org/10.1007/978-1-4614-1406-3_2

- Jones, D. L., Wagstaff, K., Thompson, D. R., D'Addario, L., Navarro, R., Mattmann, C., Majid, W., Lazio, J., Preston, R. & Rebbapragada, U. (2012). Big data challenges for large radio arrays. Paper presented at the 1-6.
<https://doi.org/10.1109/AERO.2012.6187090>
- Klusáček, D. (2011). *Event-based optimization of schedules for grid jobs* (Doctor of Philosophy).
- Klusáček, D. (2014). *Re: [GridSim-users]. Running several simulations in a row (gridsim-users digest, vol. 261, issue 1)*. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/31877913>.
- Klusáček, D. & Hana, R. (2010). Alea 2 – job scheduling simulator. Paper presented at the *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010)*,
- Lazio, J. (2008). The square kilometre array. *The Evolution of Galaxies through the Neutral Hydrogen Window (AIP Conference Proceedings Volume 1035)*, 1035, 303-309. <https://doi.org/10.1063/1.2973606>
- Lee, S., Bhattacharjee, B. & Banerjee, S. (May 25, 2005). Efficient geographic routing in multi-hop wireless networks. Paper presented at the *MobiHoc 05: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 230-241. <https://doi.org/10.1145/1062689.1062720>
- Lin, Y. & Dig, D. (2013). CHECK-THEN-ACT misuse of Java concurrent collections. Paper presented at the *Proceedings of the 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, 164-173.
<https://doi.org/10.1109/ICST.2013.41>. <http://mir.cs.illinois.edu/~yulin2/CTADetector/>
- Lin, Y. & Dig, D. (2015). A study and toolkit of CHECK-THEN-ACT idioms of java concurrent collections. *Software Testing, Verification and Reliability*, <https://doi.org/2510.1002/stvr.1567>.
- McKee, S. (2017). Networking: The view from HEP. *Journal of Physics. Conference Series*, 898(2), <https://doi.org/10.1088/1742-6596/898/2/022001>
- Medina, A., Lakhina, A., Matta, I. & Byers, J. (2001). BRITE: An approach to universal topology generation. Paper presented at the 346-353.
<https://doi.org/10.1109/MASCOT.2001.948886>
- Mehta, H. M., Chandwani, M. & Kanungo, P. (2010). Performance evaluation of grid simulators using profilers. Paper presented at the *Computer and Automation Engineering (ICCAE), 2010 the 2nd International Conference On*, Singapore.
<https://doi.org/10.1109/ICCAE.2010.5451385>
- Newman, R. & Tsen, J. (2011). *Memo 134 cloud computing and the square kilometer array*. Unpublished manuscript.

- Orgerie, A., Lefevre, L., Guerin-Lassous, I., & Pacheco, D. (2015). *ECOFEN: An end-to-end energy cost model and simulator for evaluating power consumption in large-scale networks*. IEEE. <https://doi.org/10.1109/wowmom.2011.5986203>
- Ostermann, S., Plankensteiner, K., & Prodan, R. (2011). Using a new event-based simulation framework for investigating resource provisioning in clouds. *Scientific Programming*, 19(2-3), 161-178. <https://doi.org/10.3233/SPR-2011-0321>
- Patel, N. & Patel, H. (2016). A comprehensive assessment and comparative analysis of simulations tools for cloud computing. *International Journal of Engineering and Computer Science*, 5(11), 18972-18978.
- Patibandla, A., Guptha, M. A., & Nagma, M. (2020). *Fiber optical communications*. Unpublished manuscript. https://mrcet.com/downloads/digital_notes/ECE/III Year/FIBER OPTICAL COMMUNICATIONS.pdf
- Quinson, M. (2012). Flow network model accuracy issue [GridSim-users]. Sourceforge. <https://sourceforge.net/p/gridsim/mailman/message/29118705>
- Raynard, L. & Rowland, V. (2016). Assembly integration and verification consortium. *SKA SA News* https://www.sarao.ac.za/ska_newsletter_2016_01
- Riley, G. F. (2003). The Georgia tech network simulator. Paper presented at the *Proceedings of the ACM SIGCOMM 2003 Workshops*, Karlsruhe, Germany.
- Shirazi, J. (2020, January 29). *Java performance tuning (2020, January 29), synchronization performance tips*. *FasterJ.com*. <http://www.javaperformancetuning.com/tips/synchronization.shtml>.
- Simatos, C. (2002). *Making simjava count*. (2002). University of Edinburgh: University of Edinburgh School of Informatics. <http://www.icsa.inf.ed.ac.uk/research/groups/hase/simjava/>
- SKA Media. (2014, March). *Fact sheet Meerkat*. <https://www.sarao.ac.za>. <https://www.sarao.ac.za/wp-content/uploads/2016/07/meerkat-fact-sheet-2016.pdf>
- Spekkens, K., Chiang, C., Kothes, R., Rosolowsky, E., Rupen, M., Safi-Harb, S., Sievers, J., Stairs, I., van der Marel, N., Abraham, B., Alexandroff, R., Bartel, N., Baum, S., Bietenholz, M., Boley, A., Bond, D., Brown, J., Brown, T., Davis, G., . . . Wulf, D. (2019). Canada and the SKA from 2020 - 2030.10.5281/ZENODO.3825168
- Squier. (2010, August 12). Info about measures [GridSim-users]. Sourceforge. <https://sourceforge.net/p/gridsim/mailman/message/25945295/>
- Srivastava, S., Banicescu, I., Ciorba, F. M., & Nagel, W. E. (July 2011). Enhancing the functionality of a GridSim-based scheduler for effective use with large-scale scientific applications. Paper presented at the *Proceedings of the 2011 10th*

International Symposium on Parallel and Distributed Computing, 86-93.
<https://doi.org/10.1109/ISPDC.2011.22>, <https://ieeexplore-ieee.org/document/6108259>.

- Stancu, S. N., & Martelli, E. (2015). *LHCOPN and LHCONE: Status and future evolution*. (No.664). Bristol: IOP Publishing. 10.1088/1742-6596/664/5/052025
 Retrieved from CrossRef <http://cds.cern.ch/record/2017201>
- Sulistio, A. (2008, April 18). *Re: [GridSim-users] help: How to simulate parallel file transfer*. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/19144321>.
- Sulistio, A., Poduval, G., Buyya, R., & Tham, C. K. (2005). Constructing a grid simulation with differentiated network service using gridsim. Paper presented at the <http://scholarbank.nus.edu.sg/handle/10635/69701>
- Sztajnberg, A. (2012, March 27). *Timing question [GridSim-users]*. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/29046420/>.
- Taheri, J., Zomaya, A. Y., & Khan, S. (2012). *Grid simulation tools for job scheduling and data file manipulation*. (1st ed.). John Wiley and Sons.
- Teng, F. (2009, June 3). *[GridSim-users] source code for simjava?* Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/22642924>.
- The SimGrid Team (2023), *SimGrid 3.35*. <https://simgrid.org>. Retrieved Mar 20, 2024, from <https://simgrid.org/doc/latest/Plugins.html#plugin-link-energy>
- Tunbridge, J. K. (2017). Personal correspondence with Prof. R. Buyya. University of Cape Town mail server. *Correspondence with Prof. Buyya 2017*
- Varga, A. (2021). *OmNet++Simulation Manual*.
<https://doc.omnetpp.org/omnetpp/SimulationManual.pdf>
- Walter, R., & Samborski, R. (2007). *Simulating the grid*
- Wicenec, A. (2015, April). From antennas to multi-dimensional data cubes: The SKA data path. Paper presented at the 5645-5649.
<https://doi.org/10.1109/ICASSP.2015.7179052>,
<https://ieeexplore.ieee.org/document/7179052>
- Ye, H. (2008, September 5). Simulation fails (OOM) with 1000 resources [GridSim-users]. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/20230101>
- Ye, H. (2009, September 5). *Re: [GridSim-developers] [GridSim-users] simulation fails (OOM) with 1000 resources*. Sourceforge.
<https://sourceforge.net/p/gridsim/mailman/message/20230101/>.

Annexures

Annexure A

GridSim 5.2 Roadmap

Short term goals:

- + Make lists typed to be compliant with Java 1.5+
- + Check which packages are in use
- + Remove the dependency on entities that wait for particular events. For example, the advance reservation class waits for the response of a given reservation request and ignores other events. This creates a dependency on multi-threading for entities. That becomes a bottleneck.
- + Not every entity should be a thread. Work with the idea of thread-pool.

For version 6.0:

- + Complete new design for the simulation core. Remove dependency on SimJava

Annexure B

GridSim 5.2 Release Note

CLOUDS Lab to Release GridSim Toolkit 5.2 November 2010

The Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia has released the next version of Grid simulation software, the GridSim Toolkit 5.2.

This new version of GridSim brings a new package (i.e. gridsim.fta) that provides:

- * A failure generator based on the Failure Trace Archive (FTA) format.

The provided packages read the tabbed version of the failure traces and generate a list of events that show the availability/unavailability pattern for each machine in a grid resource. For more details, see the FTA web site (<http://fta.inria.fr>) and the following publication:

Derrick Kondo, Bahman Javadi, Alex Iosup, Dick Epema, The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems, In the 10th IEEE/ACM CCGrid, Melbourne, Australia, May 2010.

- * one allocation policy for scheduling of a bag of tasks on cluster systems in the presence of resource failures.

- * An implementation of the parallel jobs workload model of DAS-2 for modelling the workload of clusters. Details about the workload model are available in the following publication:

Hui Li, David Groep, and Lex Wolters, "Workload Characteristics of a Multi-cluster Supercomputer", LNCS 2005, Volume 3277/2005, 176-193.

The version 5.0 of GridSim brings a new package (i.e. gridsim.parallel) that provides:

- * A graphical user interface for debugging resource allocation policies.
- * Several allocation policies for scheduling parallel jobs on clusters and supercomputers. The policies include aggressive backfilling, conservative backfilling, advance reservation with conservative backfilling, selective backfilling and aggressive backfilling with multiple resource partitions. The allocation policies implemented are based on the following research papers: Ahuva W. Mu'alem and Dror G. Feitelson, Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM SP2 with Backfilling. IEEE Transactions on Parallel and Distributed Systems, 12:(6), pp. 529-543, 2001.

Barry G. Lawson and Evgenia Smirni, Multiple-Queue Backfilling Scheduling with Priorities and Reservations for Parallel Systems, 2002 Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), pp. 72-87, 2002.

Srividya Srinivasan, Rajkumar Kettimuthu, Vijay Subramani, and Ponnuswamy Sadayappan. Selective Reservation Strategies for Backfill Job Scheduling. In JSSPP 2002, LNCS 2537, Springer-Verlag Berlin Heidelberg, pp. 55-71, 2002.

- * An implementation of the workload model by Lublin and Feitelson for modelling the workload of clusters and supercomputers. Details about the workload model are available in the following publication:

Uri Lublin and Dror G. Feitelson, The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Jobs. J. Parallel & Distributed Computing. 63(11), pp. 1105-1122, Nov 2003.

- * A data structure to facilitate the scheduling of jobs and advance reservations. This data structure is described as an appendix on Marcos Assuncao's PhD thesis.

- * Logging features using Java logging API.

In addition, other improvements have made GridSim more compatible with Java 5 and 6. Some bugs have been fixed and some changes in SimJava were required to enable a simulation to be paused and resumed.

All components developed as part of the GridSim Toolkit are released as "open source" under the GPL license to encourage innovation and pass full freedom to our users.

The early version of our GridSim toolkit has been used/downloaded by several academic and commercial organizations around the world including:

University of Southern California (USA), California Institute of Technology (USA), Argonne National Labs (USA), University of Manchester (UK), CERN, Universidad de Santiago de Compostela (Spain), Indian Institute of Technology, Tsinghua University (China), Sun Microsystems, IBM Research, Unisys, HP, Northrop Grumman Information Technology, British Telecom and EMC Corp.

The contributors to the GridSim software (from early to new versions) are:

- * Rajkumar Buyya, CLOUDS Lab @ The University of Melbourne.
- * Manzur Murshed, GSCIT @ Monash University, Australia.
- * Anthony Sulistio, CLOUDS Lab @ The University of Melbourne.
- * Gokul Poduval and Chen-Khong Tham, Dept. of Electrical & Computer Engineering @ National University of Singapore.
- * Marcos Dias de Assuncao, INRIA Lyon @ Ecole Normale Supérieure de Lyon
- * Uros Cibej and Borut Robic, Faculty of Computer and Information Service, the University of Ljubljana, Slovenia.
- * Agustin Caminero, Department of Computing Systems, Universidad de Castilla La Mancha (UCLM), Spain.
- * James Broberg, CLOUDS Lab @ The University of Melbourne.
- * Saurabh Garg, CLOUDS Lab @ The University of Melbourne.

* Bahman Javadi, CLOUDS Lab @ The University of Melbourne.

To download the GridSim software, please visit the CLOUDS Lab web site at
<http://www.cloudbus.org/gridsim/>

Join the GridSim mailing lists at <http://sourceforge.net/projects/gridsim>