IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 2, Ver. III (Mar-Apr. 2016), PP 05-11
www.iosrjournals.org

Simulation of Parallel and Distributed Computing: A Review

Mohd Ahsan Kabir Rizvi
Department of Electrical and Computer Engineering, Sultan Qaboos University, Oman

Abstract : *Parallel and Distributed Computing has made it possible to simulate large infrastructures like Telecom networks, air traffic etc. in an easy and effective way. It not only makes the processes faster and reliable but also provides geographical distribution among the systems. Due to its great capabilities, it has led to many types of research in this field. This led to many types of computing like, grid computing, cluster computing, utility computing and cloud computing. This paper reviews some of the various advances in the modelling and simulation aspect of this field. Some modellings techniques can be PES, the system modelling, performance modelling and network modelling. The Simulation techniques and software like SimOS, SimJava and MicroGrid are also discussed and evaluated. The paper then goes deeper into its latest form, the cloud computing.*
Keywords *-Parallel and Distributed Systems (PADS), Parallel and Distributed Computing, Parallel Discrete Event Simulation (PES), Modelling & Simulation.*

I. Introduction

Parallel and Distributed Computing is a field which is being evolved for almost half a century now. During these years, it has led to intensive research and development to bring about better and better advances into the field. In the last decade, as it started finding more and more applicability in various fields, the research has just become more flourished. The applications of PADS are immense. Some of the fields where it finds application are Air traffic management, Weather forecasting, Ocean modelling, oil reservoir simulations, pollution tracking, semiconductor simulation etc.

Research in PADS has also led to the birth of many sub-fields like Grid computing, Cluster computing and Automatic computing. Each of these has their unique applicability and it's making PADS more popular among the scientific community and the general public. It is also proved by several projects that have been based on PADS. Some of them are ABC@home [2], which finds triples related the ABC conjecture [1], Bitcoin, a peer to peer electronic cash [3], Austrian Grid II [4], which developed of a topology-aware supercomputing API for the grid, Active Harmony Project, which is a framework that supports execution of computational objects [5], [6], [7] and MilkyWay@Home, which creates a highly accurate three-dimensional model of the Milky Way galaxy [8], The Earth Simulator and The Blue Gene by IBM[16].

To understand their applicability, it is essential to understand the concept of PADS. It is a topic, which requires a vast knowledge of the hardware which is to be connected in the parallel and distributed form and then, the software that is required to control the hardware in an efficient way. It is very easy to understand that at the end of any research or invention, the basic demand is the applicability, efficiency and the performance of the system. So, once the hardware and software issues are addressed, it is then required to model the system to obtain the best possible orientation and performance.

This paper concentrates more on the modelling and simulations part of the field than the hardware and software. It first introduces the topics like parallel computing, distributed computing and PADS. Then few of the Modelling techniques and the models are discussed. Special emphasis is put on PES since it's the most extensively used modelling technique. The paper then examines the various simulation techniques and the software developed and used for the purpose. The paper then gives a detailed study on the latest sub-field of the PADS, which is cloud computing.

II. Parallel and Distributed Computing

2.1 Parallel Computing

Parallel computing is a technique in which multiple processors share the same memory to execute the required computations. The processors built by companies like Intel, AMD and ARM in recent times, implements parallel computing using multiple processor cores. This makes the executions faster, reliable and efficient. Depending on the requirements, the number of processors, the orientation etc may vary. A visual representation is depicted in Fig. 1.

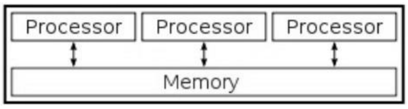

Fig. 1: Parallel Computing

2.2 Distributed Computing

This is a technique in which multiple systems of processors is connected using a network like LAN and WAN. Here, each processor consists of its own memory. The network is used to create a unified system that can break a large computation job into smaller parts and then share them among themselves for execution. This makes the simulations faster, reliable and efficient. Its other benefit is that it allows a geographical distribution among the systems. A visual representation is depicted in Fig. 2.

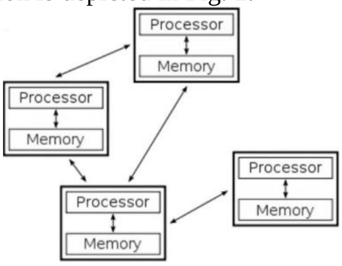

Fig. 2: Distributed Computing

2.3 Parallel and Distributed Computing

This forms a hybrid of the two types of configurations mentioned above. It forms a unique amalgam that combines the strengths and benefits of the two. Fig. 3 depicts a typical Parallel and Distributed system. Some of its advantages are:

- Capability of executing large Simulations and computations
- Higher Reliability and Fault tolerance
- Geographical Distribution
- Heterogeneous systems[27]

The classification of Parallel Computing is most widely done by the Flynn's Taxonomy [9]. The classification is based on the mode of execution of the instruction set. They are:

- SISD: Single instruction stream, Single data stream
- SIMD: Single instruction stream, Multiple data streams
- MISD: Multiple instruction streams, Single data stream
- MIMD: Multiple instruction streams, Multiple data streams

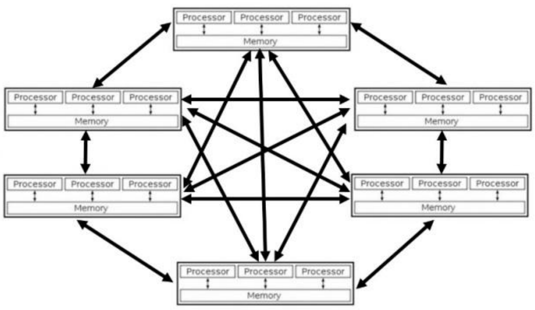

Fig. 3: Parallel & Distributed System

Based on the orientation, design and application, PADS can be classified into several types [10].

2.3.1 Cluster Computing

Here many small computers are locally interconnected to virtually form a single high power computer.

2.3.2 Grid Computing

It can be considered as an interconnection of clusters to form a geographically distributed system. It is generally used to interconnect the various branches of a single organization or the various organizations having similar interest.

2.3.3 Utility Computing

It a mode of proving access to the PADS to the smaller consumers who cannot build their own system. Utilities like hardware, software, infrastructure support etc. can be provided. Thus, it becomes a business field, where utility is provided at a cost. It generally does not involve the internet for its functioning.

2.3.4 Cloud Computing

It a form of PADS where the interconnection is achieved using the internet. Since no extra cabling and configuring costs are involved, it forms the most popular among its siblings. Due to this, it is also available to the general public and provides a great opportunity for business.

III. Modelling Parallel and Distributed Systems

Since the beginning of the research and development of parallel and distributed systems (PADS), many models and modelling techniques have been developed to understand and apply the technology in a better and efficient way. Some of the major modelsare discussed in this section.

3.1 System Models

A PADS system may contain various components like processors, memory, bus, networks, switches etc. These components communicate among one another by sharing and executing the smaller tasks so as to complete the bigger simulation as a whole. System Model is the modelling technique that helps to model the PADS system in various ways [18].

3.1.1 Hierarchical Model

Here, a hierarchical model of the overall system is created. It helps to deal with the various levels of the system in detail. Each entity is governed by another entity at a higher level. The level goes higher and higher to the top, where the central controller is placed which acts as the supervisor.

3.1.2 Layered system Model

In this type of model, the entities are built into a network of n layers. Each n entity implements functions of the nth layer and has peer protocols to communicate with other n entities. The peer protocol in each layer is a set of rules which enforce syntax, semantics and timing. This model relates the entities with strict rules between peers and layer interactions. Information hiding principle is also an important property of this model.

3.1.3 Object-based model

This model overcomes some of the flaws in the Layered system model, which have strict rules and regulations. Objects are small elements whose logic is stored in the system and they control a certain part of the systems and it may hide the hardware which it serves and controls. The objects are generally divided into 4 classes. Actorsare objects that operate on other objects. Servers are objects that come into operation only when used by other objects. Agents are objects that operate on one or more objects on behalf of another object. Passive are objects which are only made to work but cannot work on others.

3.2 Network Based Model

Depending on the requirements and the benefits, PADS can be executed in various kinds of network models. These models can be static or dynamic. Some static models are a fully connected model, star topology, ring topology, 2D mesh with wraparound etc. Dynamic models consist of switches that can instantaneously switch the connection among the systems during runtime. Fig. 4 illustrates to show a static and a dynamic model. Some Network topologies that are currently being used are Bus Based networks, crossbar networks, multistage networks, Omega networks, Hypercube networks and Tree networks.

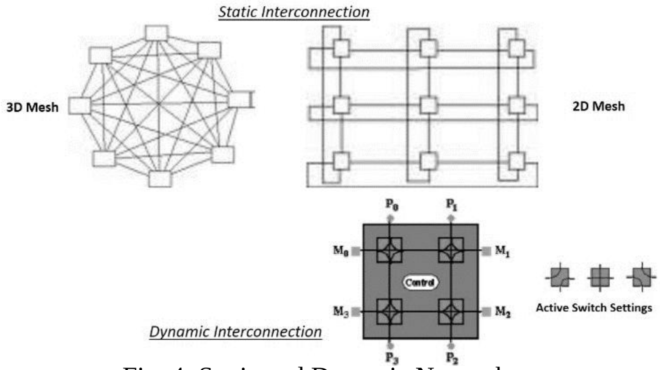

Fig. 4: Static and Dynamic Networks

3.3 Performance Modelling

This is a method of modelling using which the performance of the PADS can be monitored using the model development effort, model evaluation effort, efficiencyand accuracy of the system.Performance and Architecture Laboratory (PAL), developed at Los Alamos National Laboratory [11], [12]. To measure the performance of the PADS system, the executions in the system are represented by a mathematical model. This model depends on the problem size and the computation power of the Machine. The Parameters are program execution time, computation time, communication time and memory exchange time in the multiprocessor system [11]. The workload model depends on the overall source code of the program and the machine model depends on the communication and the computation among the nodes in the network. Performance Prophet is another tool which was developed to provide a graphical user interface which tackles the problems related to specification and modification [13].This program requires the user to graphically specify the model in Unified Modelling Language. Performance Oriented End-to-end Modelling System (POEMS) [14], developed between 1997 and 2000, is made by a composition of various evaluation tools. Once the model is created using the component model, each component of the system is evaluated using their respective evaluation tool. This technique can be used as an automatic task graph generator for High-Performance Fortran. Parallel Object-oriented Simulation Environment (POSE) [15], is used to observe the performance of programs in large systems like IBM Blue Gene [14]. It uses Charm++, which is a parallel library of C++, for the modelling purposes.

3.4 Mathematical Modelling

This modelling uses the mathematical techniques [21] to model the given PADS. Any typical system like a flight simulator or an air traffic control can be represented using differential equations. Once the equation is defined it become possible to obtain the various possible outputs based on parameters like time, situations and values of inputs. Since the systems that are being modelled are complex, these calculations become complex, so the mathematical model can be obtained using PADS as the calculations require parallel and distributed environment.

3.5 Parallel Discrete Event Simulations (PES)

This is the modelling which has become the most popular and most extensively used technique for PADS. Modelling can be done in two ways. Time stepped model and Event stepped models. Event stepped models have proved to be very efficient and reliable and thus it became a hot topic for research. In PES, the simulation is done an event based order. Fig. 5 shows the 3 different types of PES. The first one is a single processor model, the second shows the 2 processor model, where one event in one processor may lead to another event in the other processor. PES mains consist of four parameters that lead to smooth functioning and modelling of the system [17].

3.5.1 Queuing

It's seen in the figure, the events in the system are required to be ordered into a queue and the technique is called event ordering or Queuing. This is achieved by time stamping each event so that the executions can be done in an ordered.

3.5.2 Timing

Timing is required so as to understand the duration of the simulation and the actual process. It is of three types. Physical Time gives the actual time taken by an event in real life. This event may be a game of football or the building of a plane etc. When a real life event is simulated, the physical time becomes the Simulation time. The wall clock time gives the output of the hardware clock. This means that wall clock time gives the actual time taken for the whole simulation process.

3.5.3 Synchronization

Synchronization among the processors is very essentials task in the system. Synchronization is mainly of two types. In the optimistic type of synchronization, an event can be executed even if there is a chance that an event may come up at a later stage which was to be executed before the current event. Once such a violation is detected, the system rolls back to the earlier state. This system always needs to store its current state into the memory before going to the next state so that it can roll back whenever required. Though it avoids the reliance on look ahead, it may lead to overhead due to too many rollbacks. In Conservative type of approach, an event is kept on hold if there is a chance that an event of an earlier state is still required to be processed. Thus, events are executed only when it is sure that all the events with a time stamp earlier that the current state has been executed. This may lead to a Deadlock state, where every processor goes to hold state because it is waiting for the earlier event at another processor to be executed first. Its advantage over-optimistic approach is that it does not require and extra memory to keep storing its current state.

3.5.4 Execution

Execution may be the final end product of the PES. Based on the requirements of the user, execution may be of various types. As fast as possible is employed when only the end results are required by the user. The Simulation is done at one go as fast as possible and the results are given. Real Time execution is used when it is required to know each and every step of the simulation from start to the end. Scaled Real Time can be used to speed up or slow down the simulation process so as to understand the processes in detail.

Since the beginning of the PES, several algorithms and techniques have been developed for the various synchronization and execution. Many techniques have been developed to avoid the deadlock in a Conservative approach and the overhead in the optimistic approach.Next, some of the simulation software and tools used for PADS will be discussed.

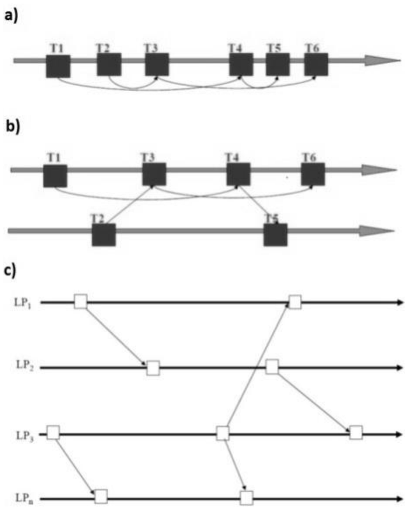

Fig. 5: Parallel and Distributed Simulation
a) Single processor PES
b) Dual processor PES
c) N-processor PES

IV. Simulating Parallel and Distributed Systems

Simulation tools help the users to obtain the final functionality and evaluate the outcome and feasibility of a given model. Some of the most used simulations will be discussed in this section.

4.1 SimOS

As the name suggests, SimOS [19] is capable of modelling and simulation a complete operating system. It can also simulate a complete computer system. It is made for studying complex systems like operating systems, computer architecture and other complex workloads. It runs on a layer between the host, which is the hardware, and software on which it runs, and the target, which is the OS and the hardware which is being simulated.

4.2 SimJava

This is a Java-based tool which is built using DEVS and it helps to simulate complex systems [20]. It is capable of providing a visual animation of the objects in the simulation and being a Java software, it also makes it easy to display these simulations or diagrams on web pages and also provide compatibility among heterogeneous systems.

4.3 MicroGrid

It is a tool based on the C programming language. It enables the user to develop a virtual grip infrastructure that can be implemented. It also allows the user to experiment on the system created and evaluate the various parameters at different levels like middleware, application and network layers. It is generally used in the Grid Computing applications.

4.4 Others

There are many other application specific tools which are used for PADS. GloMoSim [25] is a tool which is used in mobile communication systems to simulate the wireless communication protocols. Ns2 [26] is a very powerful tool that is capable of simulating a wide range of networks and also allows the flexibility in the use of coding language. Bricks [23], SimGrid [24] and GridSim [22] are two other tools, built by different organizations, to simulate the Grid computing environment. Ptolemy [27] is a tool based on Java and is generally used for embedded systems. The component-based design methodology is employed here.

V. Cloud Computing

Cloud Computing is a branch of Parallel and Distributed System where the processes are done using the internet. It forms a subscription based service which allows users to use network based storage and computer resources. Daily life examples of cloud computing are the email and social networking sites like Gmail, Hotmail and Facebook. It makes it possible to access the required resources at any time from any place. The requirements are just an internet connection and the respective device like laptop and PDA. This also becomes a cost-effective, efficient and reliable version PADS.

Cloud Computing evolved with time in three stages [28]. SPI forms the oldest form of classification, then came the IBM ontology and the Hoff's cloud model. Initially, the model was brief and simple. But as time progressed, more classifications were required to provide better flexibilities and thus the model grew larger and larger. The three general parts of the system is [31]:

- **Software as a Service (SaaS)** is the lowest form of cloud computing and requires the least skill for its use. The services like the email clients are an example.
- **Platform as a Service (PaaS)** forms the intermediate level of the cloud computing paradigm. The user is provided with a platform to create products and services. An example can be the Google AppEngine, which provides an environment to the user to build Google Apps.
- **Infrastructure as a Service (IaaS)** is the highest level of cloud service. Here the whole Infrastructure is provided to the user and this makes it most flexible but also requires the maximum skill from the users side. Resources like memory and processor in the cloud can be controlled. Amazon EC2 is an example of IaaS.

One of the most popular simulation tools is the CloudSim [29]. It is a Java-based tool built on the SimJava platform. It is an event-based simulator and the different entities in the cloud system. Other Cloud simulators are the GDCSim (Green Data Center Simulator) [32], Green Cloud Simulator [30], which is built on the ns2 platform.

VI. Conclusion

This paper has given the reader about the various modelling and simulation techniques used in PADS. As explained throughout the paper, PADS promises immense applicability and due to this, further rigorous research is inevitable. The applications and the feasibility of cloud computing have also been discussed. Also, a brief description of the simulation tools for cloud computing was provided. In the future, the author plans to evaluate and compare the modelling and simulation techniques. The knowledge will be implemented to provide innovative and effective solutions to the current cloud computing technology.

References

[1]. Elkies, Noam D. 2007. *The ABC's of number theory*. The Harvard College Mathematics Review 1(1): 57-76.

[2]. http://www.abcathome.com/ by Mathematical Institute of Leiden University. On 06/05/2015

[3]. Satoshi Nakamoto, "*Bitcoin: A Peer-to-Peer Electronic Cash System*", www.bitcoin.org

[4]. Karoly Bosa, Wolfgang Schreiner, Friedrich Priewasser, *Report on the First Feature-Complete Prototype of a Distributed Supercomputing API for the Grid. Research Institute for Symbolic Computation (RISC*), Johannes Kepler University, Linz, Austria. Technical report no. AG-D4-1-2010_1, March 2010. Austrian Grid Deliverable.

[5]. Peter Keleher and Jeffrey K. Hollingsworth, *Prediction and Adaptation in Active Harmony*, July 1998 IEEE International Symposium on High-Performance Distributed Computing (HPDC)

[6]. Peter Keleher and Jeffrey K. Hollingsworth, *Prediction and Adaptation in Active Harmony*, July 1999 Cluster Computing, 2 (1999)

[7]. Jeffrey K. Hollingsworth and Ananta Tiwari, *End-to-end Auto-tuning with Active Harmony*, 2010 in Performance Tuning in Scientific Computing, D. Bailey & S. Williams, ed.,

[8]. Nathan Cole, Travis Desell, Daniel Lombranaa Gonzalez, Fran-cisco Fernandez de Vega, Malik Magdon-Ismail, Heidi New-berg, Boleslaw Szymanski and Carlos Varela. *Evolutionary Algorithms on Volunteer Computing Platforms: The Milky-Way@Home Project*. In F. Fernandez de Vega, E. Cantu-Paz (Eds.): Parallel and Distributed Computational Intelligence, SCI 269, pp 63-90. Springer-Verlag Berlin Heidelberg. 2010

[9]. Flynn, Michael J. "Very high-speed computing systems." Proceedings of the IEEE 54.12 (1966): 1901-1909.

[10]. Indu Gandotra, Pawanesh Abrol, Pooja Gupta, Rohit Uppaland and Sandeep Singh, Cloud Computing Over Cluster, Grid Computing: a Comparative Analysis. In Journal of Grid and Distributed Computing, Volume 1, Issue 1, 2011, pp-01-04.

[11]. D. Kerbyson, H. Alme, A. Hoisie, F. Petrini, H.Wasserman, and M. Gittings. Predictive Performance and Scalability Modeling of a Large-scale Application. In ACM/IEEE conference on Supercomputing (SC 2001), Denver, CO, USA, November 2001. ACM.

[12]. D. Kerbyson, A. Hoisie, and H. Wasserman. Use of Predictive Performance Modeling During Large-Scale System Installation. Parallel Processing Letters, 15(4), December 2005.

[13]. S. Pllana and T. Fahringer. PerformanceProphet: A Performance Modeling and Prediction Tool for Parallel and Distributed Programs. In The 2005 International Conference on Parallel Processing (ICPP-05). Performance Evaluation of Networks for Parallel, Cluster and Grid Computing Systems., Oslo, Norway, June 2005. IEEE Computer Society.

[14]. V. Adve, R. Bagrodia, J. Browne, E. Deelman, A. Dube, E. Houstis, J. Rice, R. Sakellariou, D. Sundaram-Stukel, P. Teller, and M. Vernon. POEMS: End-to-End Performance Design of Large Parallel Adaptive Computational Systems. IEEE Transactions on Software Engineering, 26:1027–1048, November 2000.

[15]. T. Wilmarth, G. Zheng, E. Bohm, Y. Mehta, N. Choudhury, P. Jagadishprasad, and L. Kale. Performance Prediction using Simulation of Large-scale Interconnection Networks in POSE. In 2005 Workshop on Principles of Advanced and Distributed Simulation (PADS), Monterey, California, June 2005. IEEE Computer Society.

[16]. IBM Research: Blue Gene Project. http://www.research.ibm.com/bluegene/. On 7 May 2015

[17]. Kalyan S. Perumalla. “Parallel and Distributed Simulation: Traditional Techniques and Recent Advances” in Proceedings of the 2006 Winter Simulation Conference

[18]. Shem-Tov Levi and Ashok K. Agrawala, Fault Tolerant System Design, McGraw-Hill Series on Computer Engineering, 1994

[19]. M. Rosanblum, S. A. Herrod, E. Witchel, A. Gupta. Complete Computer System Simulation: The SimOS Approach. IEEE Parallel and Distributed Technology, 1995.

[20]. F. Howell and R. McNab. SimJava: A Discrete Event Simulation Package for Java with Applications in Computer Systems Modelling. First International Conference on Web-based Modelling and Simulation, San Diego, CA, Society for Computer Simulation, Jan 1998.

[21]. KVASNICA, Peter, Igor Kvakvasnica, and Mária Igazová. "Mathematical Model and its Parallel Simulation by Web Ser-vices." Journal of Information, Control and Management Systems 7.2 (2009).

[22]. R. Buyya and M. Murshed. GridSim: A Toolkit for the Modelling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. The Journal of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, May 2002.

[23]. Takefusa, Atsuko. "Bricks: A performance evaluation system for scheduling algorithms on the grids." JSPS Workshop on Applied Information Technology for Science (JWAITS 2001). 2001.

[24]. Casanova H. Simgrid: A toolkit for the simulation of application scheduling. Proceedings 1st IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid2001), Brisbane, Australia, May 2001. IEEE Computer Society Press: Los Alamitos, CA, 2001.

[25]. Zeng X, Bagrodia R, Gerla M. GloMoSim: A library for parallel simulation of large-scale wireless networks. Proceedings 12th Workshop on Parallel and Distributed Simulations (PADS98), Banff, Alberta, Canada, May 1998. IEEE Computer Society Press: Los Alamitos, CA, 1998; 154–161.

[26]. Fall K, Varadhan K (eds.). The NS manual. Technical Report, University of California at Berkeley, June 2003.

[27]. Buck, Joseph T., et al. "Ptolemy: A framework for simulating and prototyping heterogeneous systems." (1994).

[28]. Syed A. Ahson , Mohammed Ilyas, “Cloud Computing Software Service: Theory and Techniques”, CRC Press, Taylor & Francis Group, 2011

[29]. R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, “Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.

[30]. D. Kliazovich, P. Bouvry, and S. Khan, “Greencloud: a packet-level simulator of energy-aware cloud computing data centers,” The Journal of Supercomputing, 2010.

[31]. Fujimoto, Richard M., Asad Waqar Malik, and A. Park. "Parallel and distributed simulation in the cloud." SCS M&S Magazine 3 (2010): 1-10.

[32]. Sandeep K.S. Gupta, Rose Robin Gilbert, Ayan Banerjee, Zahra Abbasi, Tridib Mukherjee, Georgios Varsamopoulos,“ GDCSim: A Tool for Analyzing Green Data Center Design and Resource Management Techniques,” in Latin American Network Operations and Management Symposium. 2011.