Paper 366-2011

# Designing a Grid Computing Architecture: A Case Study of Green Computing Implementation Using SAS®

Krishnadas N, Indian Institute of Management, Kozhikode, Kerala, India

## ABSTRACT

With the growing concerns about environmental impact of computing, it has become necessary to utilize the unused computing resources. It not only ensures efficiency and reduced carbon emissions but also leads to potential cost/time savings in organizations. Intense computing actions that work on huge data, generally take hours to compute which could be reduced to few seconds using grid computing. This paper is an attempt to understand the nature of grid computing and explore the relation between work load and optimum number of nodes required for grid implementation. This would act as guidelines to design an optimal grid for an application of a specific workload. It takes up the case of a computer lab and conducts a pilot study on utilization of computing resources by the students followed by implementation of grid computing using SAS. Paper also comes up with a Green Computing Strategy plan for the computer center (grid computing and SAS being a part of it).

## INTRODUCTION

Sustainability is defined as “development that meets the needs of the present world, without compromising the ability of future generations to meet their own needs” (Brundtland, 1987). Present focus of many IT organizations/computing centers today is the triple bottom line by satisfying the sustainability of economic, social and environmental capitals simultaneously. Ecological sustainability looks upon long term survival of individual/group and deserves a higher priority than sustainable economic development (Starik and Rands, 1995). Green IT has come up as a new discipline to address the concern of sustainability when IT/Computing is involved.

Green IT is a synonym to environmentally sound Information Technology (Murugesan, 2008). It includes multiple aspects like environmental sustainability, energy efficiency economics, cost of disposal/recycling etc. Broadly, there are two sides of Green IT; one dealing with IT being the cause of environmental problem and the other using IT/IS to solve the environmental problems. It captures the technical capability including choices related to applications, data, technological configurations etc as well as human/managerial capability including experiences, competencies etc of IT personnel (Byrd & Turner, 2000).

This paper takes up a case of computing center (Lab) with more than 150 PCs. It develops a strategy for Green IT implementation and conducts a pilot study using SAS to implement /experiment Grid Computing. The strategic plan includes concepts like benchmarking, cultural aspects, regulatory measures, energy of data centers, cloud computing, videoconferencing, e-waste, duplex printing, virtualization and Green IT procurement (Each component will be described in an elaborate plan in the full paper). The pilot study on the other hand aims at increasing the utilization of the PCs which remain idle for most of the time. Another significant contribution of this study is in terms of coming up with a design of Grid computing architecture. The reduction in execution times reaches a maturity level when the number of nodes involved in grid is increased continuously. This study attempts to find a relation between workload and the optimum number of nodes required to achieve potential reduction in execution time. This would help organizations to decide on average number of nodes required to run a grid for an application of specific workload.

## GRID COMPUTING AND SAS

Grid Computing was pioneered by Dr. Ian Foster and Dr. Carl Kesselman on the “Globus Project for Grid Computing”. The basic concept lies in parallel / distributed processing where the goals are twofold. First, to reduce the overall elapsed processing time while leveraging existing hardware and second, to utilize the unused computing resources (processing power). There is an important classification of grid that classifies it into compute grid and utility grid. Compute grid generally has an application that requires intense computing actions and that could be divided into sub-tasks that could run in parallel and combined later to yield the desired result. These sub-tasks are executed in different machines in order to ensure time effectiveness and resource utilization. The visual representation of the same is shown in Figure-1. Utility grid on the other hand is a collection of hardware resources that is virtualized so that multiple users can submit their individual applications to this pool of resources and have the benefit of load balancing, maximum use of resources, and optimal turnaround time for each individual application.

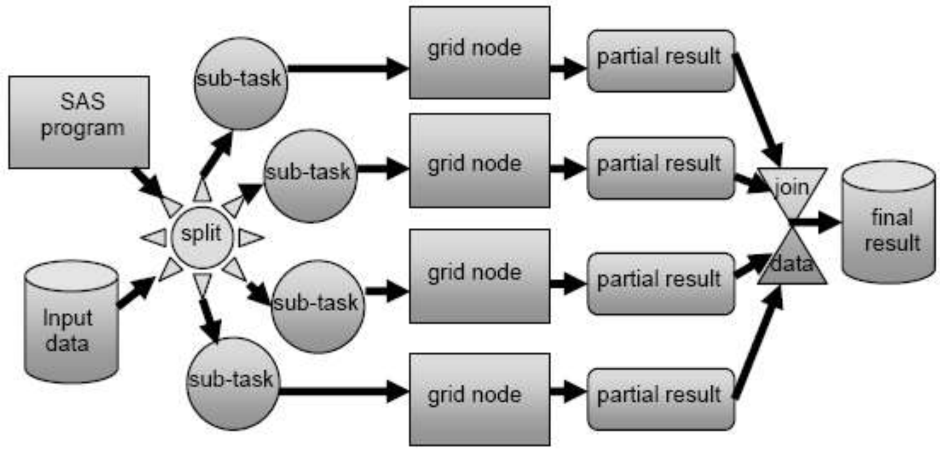



**Figure 1. SAS Grid Computing Concept**

Computing with the power of SAS Business Analytics, SAS Grid Computing provides the ability to scale business processes and accelerate decisions to create a competitive advantage. Users can submit SAS jobs to a shared pool of resources rather than an individual server to balance workload and better manage the SAS environment. Compute-intensive programs can be allocated and managed to run in pieces across grid environments, enabling IT organizations to optimize hardware. SAS makes it possible to harness the power of various computing resources in a network and use it in a single project for faster execution of tasks. Applications that generally have some tasks that repeat itself in every execution are most suitable for SAS Grid Computing. Also termed as BY GROUP processing in SAS, the concept of repetitive task distributing itself in the grid nodes is also known as embarrassingly parallel application. Such applications include statistical simulations, data analysis, searching for optimal designs etc.

SAS/CONNECT is the platform to go for grid computing implementation. A major benefit of enabling SAS grid computing with SAS/CONNECT is that it puts the capability at the intelligence architecture layer as part of the toolset. This makes it available to be leveraged by both SAS programmer customer base as well as all of the SAS solutions and vertical products developed internally by SAS. The basic implementation of the pilot study is derived from Figure-1. The grid nodes are termed as EXECUTORS while the SAS/CONNECT Server is termed as MANAGER that handles the request for running an application on grid. These terms are further explained in the upcoming sections.

The organization selected for implementation of Green IT Strategy and conducting the pilot study is an educational institute (Business School). The IT setup at this institute comprises of a multi layered architecture, the lowest layer of which is made up of personal computers and workstations. The next layer consists of File servers and Database servers. The third layer consists of Web server, FTP server, Email server and other high-end servers/computers required high resources demanding tasks. All the buildings including hostels in the campus are part of the campus LAN and all the services available on the LAN can be accessed from any node. The connectivity to Desktops is using 4 pair ECAT 5 UTP cable ensuring a dedicated 100Mbps bandwidth at desktop level. The computer centre (computer lab) acts as the main hub of the network and hosts a layer 3 backbone switch. The workgroup switches are located in the respective buildings. All the hostel rooms are also connected to the campus LAN. The summary of the infrastructure is presented in Appendix-1.

A benchmarking of this organization was conducted to set the base of developing the Green IT strategy. Benchmarking is defined as means of estimating the system performance by using multiple workloads and measuring the performance (Jones, 1975). It is used as a means to determine the best equipment to buy in various technology insertion cycles, when the standards/recommendations regarding computer hardware procurement is concerned (Letmanyi, 1984). This step remains pivotal in setting up environmental objectives/targets for an organization to reduce its environmental impact of computing.

Benchmarking can be used by this organization to measure its carbon footprints and set its carbon emission reduction goals. Direct and indirect emissions should be included in the design of metrics and models required to incorporate benchmarking and transparency. This has been one of the top priorities of organizations looking to implement Green IT in their organizations. There are various calculators available to give a rough estimate of carbon emissions. A basic version of Microsoft's Green IT calculator was used to give a start to the benchmarking process

(Advanced calculators are available for detailed study). This calculator is used only for the desktops; there are different calculators available for servers. The inputs for the same are documented in Table 1 while the output shown by the calculator is shown in Table 2.

| Desktop Energy Estimates: Inputs | |
|---|---|
| | |
| Number of PCs | 500 |
| LCD Usage | 76% |
| CRT Usage | 24% |
| Desktops | 100% |
| Notebooks | 0% |
| Windows XP | 77% |
| Windows Vista | 3% |
| Other Windows | 21% |

**Table 1. Inputs for Microsoft Desktop Energy Savings Calculator**

| Desktop Energy Estimates: Output | |
|---|---|
| | |
| Power Cost | $46,278 per year |
| Power Consumed | 473,676 kW/h |
| Carbon Emissions | 367.1 tons |
| Equivalent number of automobiles | 67.2 |
| Equivalent no of homes | 39.4 |
| Equivalent number of trees | 797.7 |

**Table 2. Inputs for Microsoft Desktop Energy Savings Calculator**

## UTILIZING THE SYSTEM RESOURCES EFFICIENTLY: A PILOT STUDY

A pilot study was conducted to test the feasibility of grid architecture at the computing center of the selected organization. The aim of this study was basically to utilize the underutilized computing resources of the systems located in computer center and other computers connected to the network (students, faculties etc). The study was divided into three phases. In the first phase, the usage of the computer systems by a typical user (student) was studied to get the information on idle times. The second stage involved the setting up of grid architecture using six executors (grid nodes), one manager (SAS/CONNECT) and users of the grid. The third phase dealt with evaluating the efficiency of the pilot study in terms of utilization of the idle system resources. This phase is also the analysis phase wherein the optimal design for grid architecture is explored by understanding the relation between workload and execution times.

Phase 1: This part of the study involved monitoring the number of times a typical user goes idle and the time duration of the same. Idle time is defined in this paper as any inactivity for more than 30 seconds. This would be an appropriate measure because the systems monitored were in the computing center of selected organization, where the usage of videos is not encouraged. A user can perform no activity for more than 30 seconds and still be active only in special cases like watching a video.

A VB application was developed and installed in all the systems being monitored. This application records the start/end time of monitoring along with the number of times a system goes idle. It also notes down the extent of time to which the system remained idle in each case. Hibernation mode and 'log-off' modes were not considered because they already have power savings mechanisms. Hence, these states cannot be attributed to underutilized computing resources (Lock state was taken into account). The details recorded in the database for a sample user is shown in Table-3.

| Start time | 1:17:38 AM |
|---|---|
| End Time | 9:47:14 AM |
| **Idle Count** | **Idle Time** |
| 1 | 30.05 |
| 2 | 64.58 |
| 3 | 47.05 |
| 4 | 94.61 |
| 5 | 187.56 |
| 6 | 23804.91 |
| 7 | 5833.7 |

**Table 3. Idle Time recording of a sample user**

The data of all the users were aggregated to gain insights on two parameters. The first being the average number of times a user goes idle per hour. Secondly, the average time for which the system remains idle each time it goes into the idle state. The summary of results is shown in Table-4 while the details of the same are presented in Appendix-2.

| | |
|---|---|
| Number of sample cases | 25 |
| Total of Idle count per hour across samples | 135 |
| Average idle count per hour | 5.4 |
| Average percentage of time the system remained idle | 57.78% |

**Table 4. Results of study, Phase-1**

Results reveal that more than fifty percentage of the time the system remains idle. This excludes the time during which the system has been logged-off. If it were to be included, the percentage of time the system remains idle would have been higher. This study reveals the acute need of using technology to utilize these idle times and contribute to the green computing. Therefore, a small application using grid setup was implemented in the computing center. The next phase of the study describes the same.

Phase 2: This part of the study involved execution of an application that required intense computing resources. This application used the SAS/CONNECT platform already described in previous sections. The concept of utilizing the system resources in a network has its deep roots in the idea of meta-computing. The growth of internet and networking has taken it to the next level framed as grid computing (Ian and Carl, 1999; Ian et al., 2001). Grid computing will play a crucial role in shaping the future of IT. SAS Grid used in this study provides a powerful toolset to leverage upon the ubiquitous presence of data, dealing with issues like reliability, heterogeneity, scheduling, resource management etc.

Executors/ Grid Nodes: They are basically the systems, the computing resources of which will be utilized by the users for running grid applications. For the pilot study, six executors were devoted to the grid. The Executor only utilizes idle CPU cycles on the machine and does not impact on the CPU usage of running programs. Each executor should be connected to the SAS/Connect manager.

The first aim was to find out the system utilization of computers in the computing centers with and without the execution of the grid application. The initial setup of architecture mentioned above was followed with two executors and one manager. The application was executed from another system which utilized the computing resources of the two executors. The executors were not touched during the data recording process and real time data on CPU utilization was captured using Performance monitoring logs in Windows. This capturing of real time data was performed before and after the execution of the grid application on a system different from executors.

After recording data on CPU Utilization, the next part aimed at testing the execution time and power usage of the application. To ensure enough variability in the data collection, the setup initially began with one executor and a workload of 200 (idiosyncratic to the application developed). The number of executors was increased one by one until all the six executors were utilized. After the completion of data recording, the no. of executors was brought back to one. Now, the same process was repeated with a workload of 400 and data was captured. The data included power usage being monitored using manager console and execution time taken by the program (revealed as output of the program). Same process was repeated for a workload of 800 as well.

## DATA COLLECTION

Firstly, the real time data on CPU utilization of the two grid nodes were looked upon. Data was recorded both prior to the grid application execution and post execution for both executors. For the first executor, the graph for CPU utilization prior to the execution of the grid program is shown in Figure-2 and that of the post execution is shown in Figure-3.

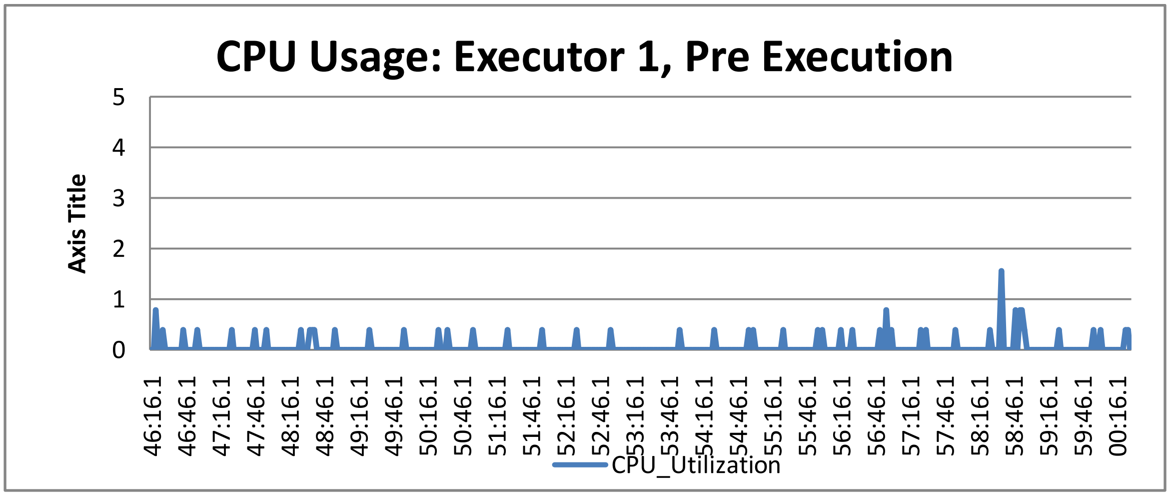



**Figure 2. CPU Utilization levels prior to execution of Grid Application**

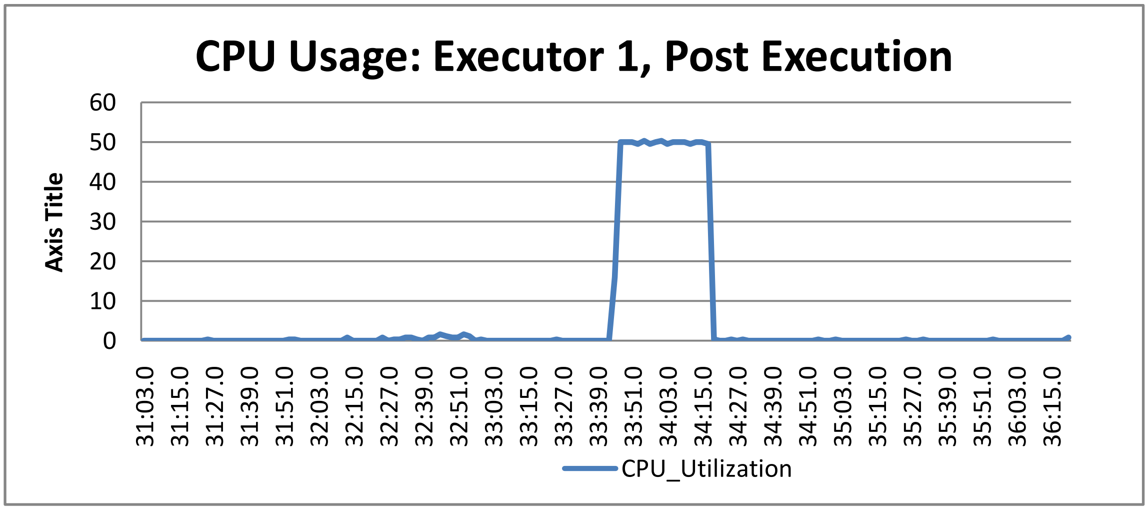



**Figure 3. CPU Utilization levels post execution of Grid Application**

The data is real time and was recorded over almost same time period for both the cases. It is evident from Figure-2 that the maximum utilization of CPU prior to the implementation of Grid application was 1.56% and it boosted up to 50.39% in Figure-3. The hump in the latter graph demonstrates the time during which the executed program rendered the service of first executor. Thus, a simple application with sample workload demonstrated considerable utilization of CPU with two executors in place. Using heavy applications on grid and with the help of multiple executors, the idle time and unused resources can be properly utilized.

Further, the data on execution times was collected for the multiple workloads (200, 400 and 800) and multiple executors (Six executors). Figure-4 reveals the execution time of the grid application across different variables. There was substantial reduction of execution times when the executors were incremented initially (50-60%). However, the reduction stabilizes as the number of executors increase to higher numbers like 4 or 5. Hence, it can be said that the marginal contribution of individual monitor to reduction in execution time exponentially decreases over the increasing number of executors. This shows that even few executors can substantially reduce the execution time of an application using grid setup. Therefore, it is an added utility along with efficient utilization of the system resources.

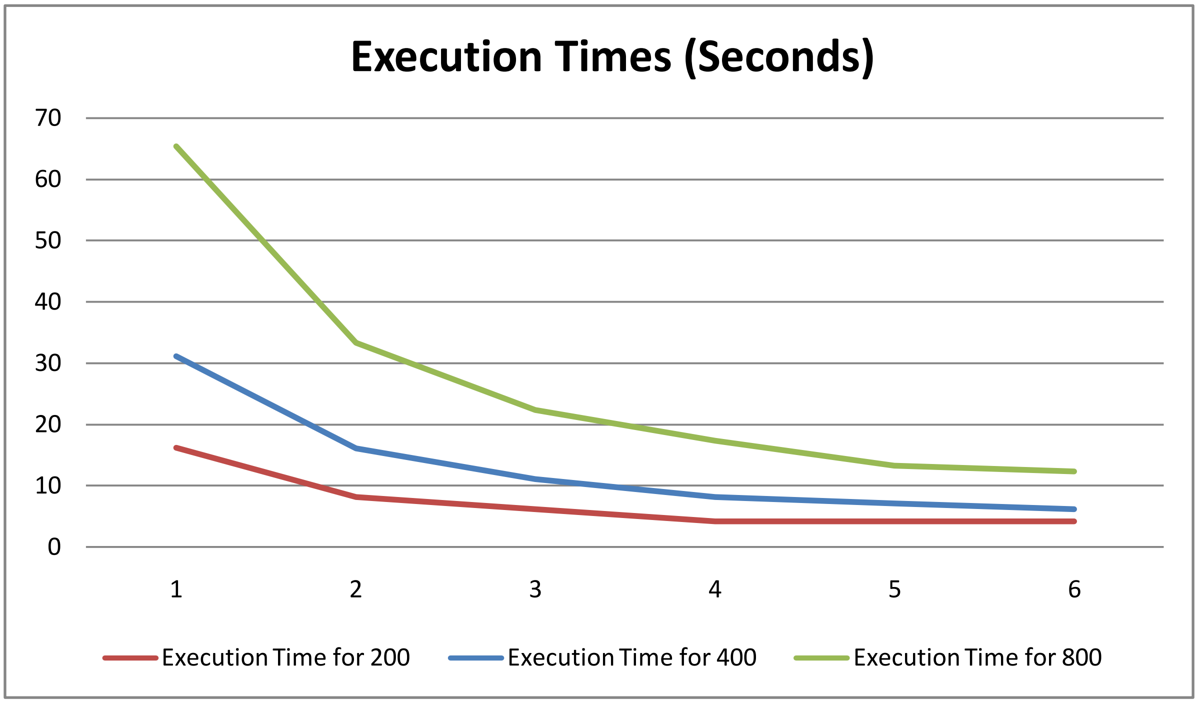



**Figure 4. Execution Time of Grid Application: 3 workloads and Six Grid Nodes**

The next analysis was conducted on the power usage in terms of GHz. Each executor adds on to the maximum power available by 2.933 GHz (all executors have the same configuration in the computing center). The total power usage before and after the execution of the program was recorded for all workloads and all executors. Roughly same amount of power was consumed for all the executions of a program keeping the load constant. When the load was doubled, the approximate power consumed also doubled but remained approximately same across the executors. Close observation of the exact numbers reveal that there is some amount of power consumed in the interaction between manager and executor, however it is almost negligible. Also, for each execution of the grid application; the maximum power available (in terms of %) and the actual power usage was recorded. Appendix-3 shows the power consumption pattern for a fixed load of 200.

The study of the pattern reveals the reduction in time during which the power usage is at its peak. Devoting fewer executors to the grid application would lead to congestion as the application utilizes the maximum power of the executors for a long period of time. In case of non-dedicated executors, this could be a problem in terms of the time taken to execute an application. Also, it could lead to disruptions in the working of executors in case it is a dedicated one. Therefore, enough executors should be available to run an application on Grid.

## CONCLUSION AND FUTURE WORK

It is clear from Figure-4 that the reduction in execution time is not significant after adding a certain number of grid nodes. It can be seen that the execution time almost reaches a saturation time after addition of five-six grid nodes. This relationship needs to be mathematically be analyzed by finding out the relation between workload and the number of grid nodes that gives effective reduction in execution time. This experiment will further be conducted by increasing the workload and increasing the number of nodes to get more data. This would provide an effective data set to conduct an analysis for finding out the optimal number of grid nodes required for a particular workload.

However, finding out the above relation alone would not suffice the need for developing an efficient design for grid computing. This is because of the power analysis. It can be clearly seen that less number of executors lead to utilization of maximum power available for a longer time when compared to a grid with more executors. Utilizing the maximum power for a longer time results in congestion and thus it is necessary to consider this factor for selecting the number of grid nodes for a particular workload. Therefore, we see a tradeoff between fewer executors for reduced time of execution and more executors for reduced peak power consumption.

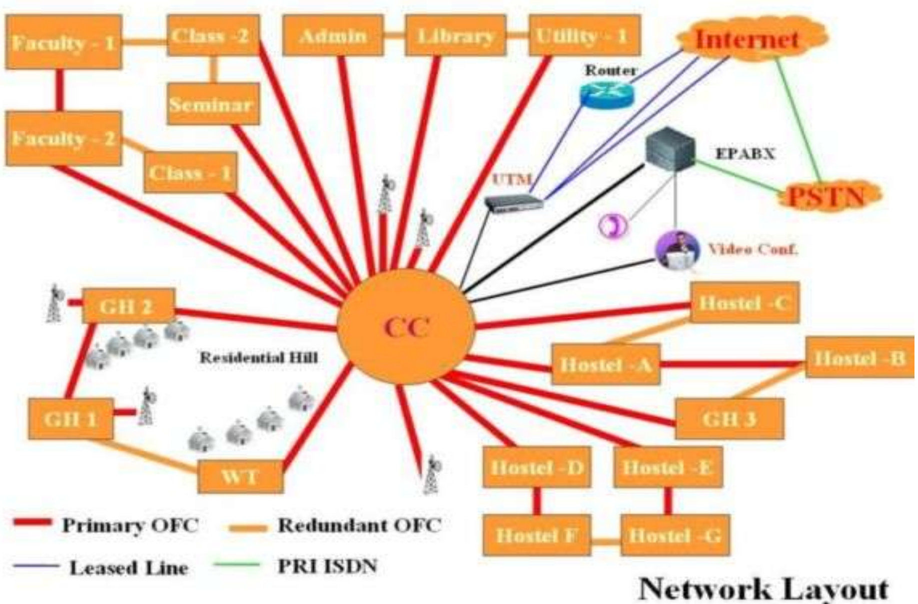
The procedure for future analysis follows a three step process. First, the issue of execution time is taken up independently and the relation between workload and number of nodes that would mature the reduction in execution time is found out. Secondly, the issue of power is taken up independently and relation between the workload and number of grid nodes is found. The last step involves taking the two issues together and finding out the trade-off between power and execution time. This would act as guidelines for any organization planning to implement an application on grid especially using SAS.

## REFERENCES


- Brundtland, G.H. (1987), Our Common Future: The Report of the World Commission on Environment and Development, Oxford University Press, Oxford.
- Murugesan, S. 2008. Harnessing Green IT: Principles and Practices. IT Professional 10, 1 (Jan. 2008), 24-33.
- Starik, M. and Rands, G.P. (1995), ‘‘Weaving an integrated web: multilevel and multisystem perspectives of ecological sustainable organizations’’, Academy of Management Review, Vol. 20, pp. 908-35.
- Byrd, T. A. & Turner, D. E. (2000). Measuring the flexibility of information technology infrastructure: Exploratory analysis of a construct. Journal of Management Information Systems,17(1), 167–208.
- Jones, R., “A survey of benchmarking: The state-of-the-art.” Benchmarking: Computer Evaluation and Measurement, N. Benwall, ed., Hemisphere Publishing Corp., Washington, DC, 1975, pp. 15–23.
- Letmanyi, H., “Assessment of techniques for evaluating computer systems for federal agency procurements.” NBS Special Publication 500-113, National Bureau of Standards, Washington, DC, March 1984.
- SAS Scalability Community at http://support.sas.com/rnd/scalability/grid
- Ian Foster and Carl Kesselman (editors), The Grid: Blueprint for a Future Computing Infrastructure, Morgan Kaufmann Publishers, USA, 1999.
- Ian Foster, Carl Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of Supercomputer Applications, 15(3), Sage Publications, 2001, USA.


**APPENDIX**

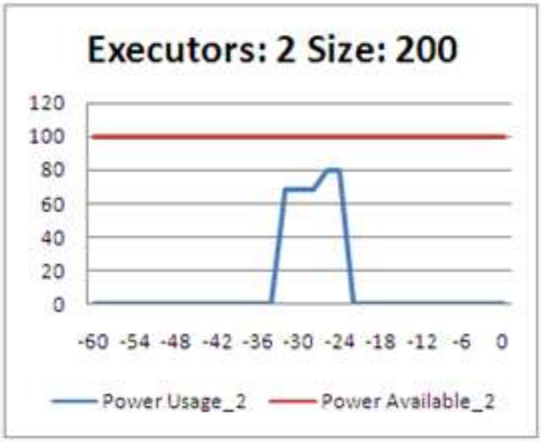
Appendix-1: Computing Infrastructure of the selected organization

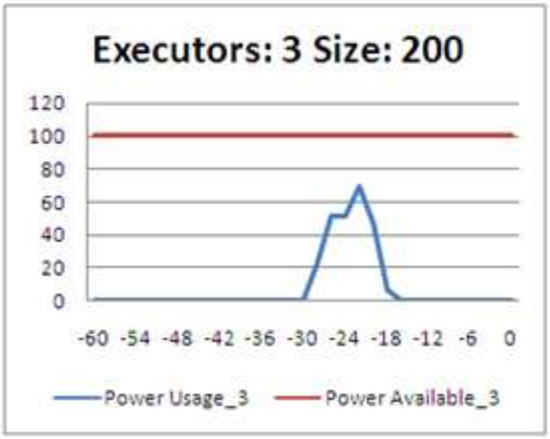


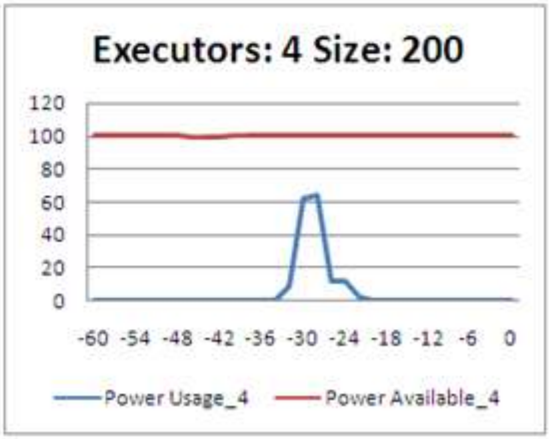
**Appendix-2: Summary of individual test cases**

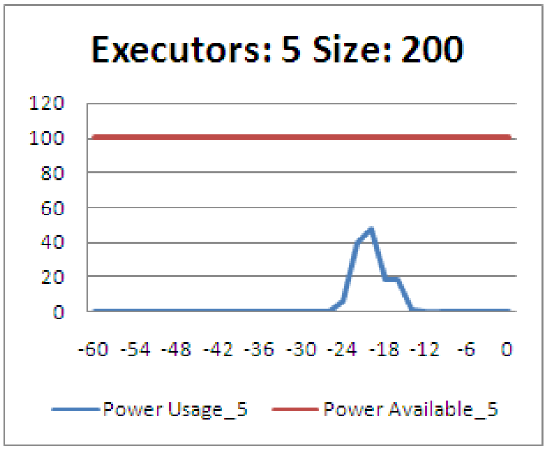
| Sample | No of times sample goes idle per hour | Idle time Percentage |
|---|---|---|
| | | |
| 1 | 1 | 92.79 |
| 2 | 6 | 46.05 |
| 3 | 7 | 70.31 |
| 4 | 6 | 72.33 |
| 5 | 1 | 71.77 |
| 6 | 6 | 50.04 |
| 7 | 7 | 34.61 |
| 8 | 5 | 50.77 |
| 9 | 5 | 74.96 |
| 10 | 10 | 43.26 |
| 11 | 5 | 73.67 |
| 12 | 10 | 57.69 |
| 13 | 5 | 65.73 |
| 14 | 5 | 48.85 |
| 15 | 7 | 34.13 |
| 16 | 5 | 44.9 |
| 17 | 5 | 61.07 |
| 18 | 10 | 75.62 |
| 19 | 1 | 49.9 |
| 20 | 4 | 59.36 |
| 21 | 4 | 56.06 |
| 22 | 7 | 48.17 |
| 23 | 4 | 70.54 |
| 24 | 8 | 36.31 |
| 25 | 1 | 55.5 |
| **Total** | **135** | **1444.38537** |
| **Average** | **5.4** | **57.77541481** |

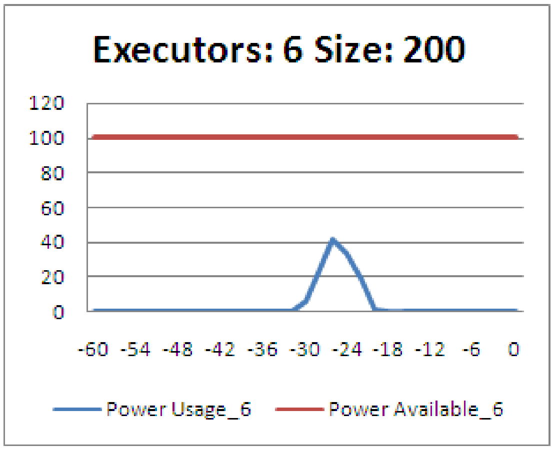
**Appendix 3: Power Consumption pattern for multiple executors, workload: 200**

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

PhD Student (FPM),
Indian Institute of Management-Kozhikode.

Room E-4, IIMK Campus,
P.O, Kunnamangalam - 673 570
Kozhikode, Kerala (India)
Email: username.krishna@gmail.com
Mob: +91 9048240816
Webpage: http://sites.google.com/site/krishnadasniimk/