Effect of Grid Job Slice Variation on Virtualized Desktop Grid Performance

Gurjit Singh Sra
Assistant Professor
Department of Computer Science & Engineering
Igce, Abhipur, Mohali (Punjab), India

Dinesh Kumar
Associate Professor
Department of Computer Science & Engineering
Daviet, Jalandhar (Punjab), India

ABSTRACT
Grid computing has come a long way in solving data and compute intensive problems. There have been significant gains in solving these problems using grid based systems by dividing these jobs into sub jobs and submitting sub jobs on various nodes in the grid system. Various job characteristics may have effect on the execution of jobs on grid systems. The performance of grid systems varies with the type of problem taken. In this paper, the grid systems are studied for variations in the size of job fragments of sub jobs. With different size of sub jobs the grid systems responds differently. As the job fragment size is increased the performance of grid system is observed to increase. The job taken is video conversion from AVI to FLV.

Keywords
Grid, Globus, Virtualized Grid, Video Encoding, AVI, FLV, Job Partitioning.

1. INTRODUCTION
The Grid computing provides CPU and data handling capabilities that far exceeds what can be attained in most research institutions and allows for parallel implementation of various existing algorithms and software. This analyses computationally intensive data and tasks that otherwise could be time consuming to perform on standalone computers. Grids enable the sharing and aggregation of wide resources. Grid computing has the design goal of solving problems that are not possible to solve on a single computer in a reasonable amount of time. Grid computing offers easy access to rarely available and limited resources such as high performance parallel computers and their resources. In computational grid [1] [2] distributed computer systems cooperate to execute user jobs either on local or on remote computer systems. It is similar to the former meta computing[3] where the focus was limited to compute resources while grid computing takes a broader approach[4][5]. Grid computing provides the user with access to locally unavailable resource types from a pool of distributed resources and expects that this will result in a reduction of the average job response time. This also results in better utilization of the grid computers and the job-throughput is likely to improve due to load-balancing effects between the participating systems. Grid computing also offers a mechanism for various computing and data resources dispersed in the world to form a virtual organization, and the resulting single system virtual image enables users to easily access to every type of resources [6]. The concept of grid computing is realized through emerging standardization of sharing resources and the availability of higher network bandwidth [7].

Grid computing is sometimes taken as a specialization of traditional parallel processing. However, several factors make grid computing a more promising trend. Grid computing is also confused with cluster computing, however a key difference is that a cluster is a single set of nodes at one location, whereas a Grid is composed of many clusters and other kinds of resources including computers, supercomputers, data sources, storage systems and devices that are geographically distributed and managed by different companies, universities and research institutes, to which users other than the owners can be granted access. Sites grant access to groups of users that form virtual organizations [8].

Grid computing is currently being applied in various types of applications, and can be implemented in different ways. Grids computing offer a way to solve Grand Challenge problems such as protein folding, financial modeling, earthquake simulation, and climate/weather modeling [9]. Various other project and application areas of Grid computing include CERN Large Hadrons Collider[10], Distributed aircraft engine diagnostics, Worldwide telescope, Biomedical informatics research network[11][12]. Other project areas based on grid computing are POEM @Home, Climateprediction.net, SZTAKI Desktop Grid, MilkyWay@Home, ZetaGrid[9][10]. Nevertheless, the building, administration, and operation of a global grid environment require new technology, which involves grid architectures, software protocols and middleware [13] [14].

In grid computing the job is partitioned and the sub jobs are distributed to geographically distributed computers all over the world connected to grid system. Earlier, self-scheduling schemes partition the size of loop iterations according to a formula instead of node performance, so additional slave nodes may not get good performance. Intuitively, we may want to partition problem sizes according to CPU clock speed. However, the CPU clock is not the only factor which affects node performance. Many other factors also have dramatic influences in this respect, such as the amount of memory available, the cost of memory accesses, and the communication medium between nodes, etc [15].

2. RELATED WORK
A large number of computational problems have been successfully implemented and solved using grid computing [14]. This is due to efficient and effective utilization of the computing resources in grid systems. Meanwhile, the complex problems such as scientific, engineering and business need to utilize the huge resources that are available in vast heterogeneous computational environment. That is why grid computing is considered as an ultimate solution to solve these problems [16] [10]. Extensive usage of grid systems is done to carry out compute and storage intensive tasks, so there is a need to evaluate the performance of grids with respect to different parameters like processing time, resource utilization, memory statistics, network statistics etc., so that user of the grid can make easy decisions regarding the configuration of the

environment they have to build or use for the execution of their grid jobs [17]. Other applications areas such as medical applications are built upon grid infrastructure. The privacy and security of sensitive data in such cases can be implemented using trusted data storage [11]. Further the type of scheduling strategies adopted plays a vital role in the performance of grid system. Various scheduling and dispatching rules such as First Come First Served (FCFS), Earliest Due Date (EDD), Earliest Release Date (ERD), and an Ant Colony Optimization (ACO) can be applied. The performance of FCFS is better than others as far as total scheduling time is concerned [16].

Digital video is most common way of communicating in the digital world. Digital video can exist in many formats, using any number of settings. When a digital video file does not meet the specifications, or the file type is bad for the intended use, it must be converted to the proper format, using video encoder hardware or software. Video compression and conversion technology has evolved from last few decades and there have been approximately 50% improvement in the compression ratios for last few years [18]. Software and hardware based mechanisms have been used for implementing video encoding process in parallel especially in real time scenario. Hardware based parallel video encoder[19] is implemented on bus network which makes use of Divisible load theory (DLT) paradigm for strip-wise load partitioning, balancing, load distribution. Strategies are developed to make use of the data parallelism already present in the video encoding process. In this, both the size of the job slices or load partitions and all the other resulting overheads caused in the process of parallel video encoding can be dually considered. The processing time of video encoding process is significantly reduced as a result of implementation of parallel video encoding systems [19]. This shows that video streams can be encoded in parallel using hardware based or software based systems to increase the overall efficiency of video encoding systems.

3. PROPOSED WORK

Video encoding is also attempted using conversion from AVI to FLV format on grid systems [20]. The problem is computation intensive as it requires large number of compute cycles to complete. Large amount of video data is transferred to computational nodes resulting in utilization of network resources. AVI file format is the most widely used format for playing videos on desktops and televisions. With the wide spread use of Flash plug in, most of the video data is viewed on the web in the form of FLV format. AVI file format results in large video size which is not efficient enough to be transferred on web especially with low bandwidth connections. Grid systems are used for execution of such jobs as the video file can be easily be split into number of fragments and each fragment is independently and simultaneously converted into FLV format on the distributed system. Each encoded fragment is then recombined to create full video in FLV format. The video conversion has been successfully been implemented on globus grid systems and the performance of grid system is measured as in [20]. The job fragment size used in [20] is 2 minutes. The grid system may perform differently with different job fragment size submitted to the grid nodes. The objective of current work is to study the effect of variations in the size of video fragment size as submitted on various grid nodes on the overall grid performance. The metrics used for evaluating the performance of grid system is job execution time.

4. METHODOLOGY

To study the effect of job size variation on grid systems, the same virtual desktop based grid environment is used as in [20]. All the grid nodes have virtualized desktops using VMware with CentOS operating system and connected through a central hub. Node B runs the globus container and Node A is the client for submitting the job to grid system which is composed of all nodes with Node B as head node. All other nodes form a torque cluster. Node B runs pbs_server and all other nodes run pbs_mom as indicated in the Table I. The grid system and its software components are as shown in Table I.

The grid job is submitted by script which partitions the video into various fragments of equal size. For each fragment job submission file is created dynamically by the job script. The job submission file is in rsl format which indicates the command to be executed on the grid compute node to convert the AVI into FLV format.

The job submission file is submitted to globusrun service of Globus middleware, which translates the job request into format as required by the local resource manager such as torque.

Table I. Software Components on Grid Nodes [20]

Software components Nodes	IP Addresses	Globus Container Services	Globus run-ws	GridFtp Sever	NFS Server	NFS Client	Torque server, Scheduler	Torque Client	Mencoder
Node A	192.168.31.39		Yes	Yes					Yes
Node B	192.168.31.40	Yes	Yes	Yes	Yes		Yes		Yes
Node C	192.168.31.41					Yes		Yes	Yes
Node D	192.168.31.42					Yes		Yes	Yes
Node F	192.168.31.43					Yes		Yes	Yes
Node G	192.168.31.44					Yes		Yes	Yes
Node H	192.168.31.45					Yes		Yes	Yes
Node I	192.168.31.46					Yes		Yes	Yes
Node J	192.168.31.47					Yes		Yes	Yes

5. RESULT ANALYSIS

To analyze the effect of variation in number of compute nodes on the execution time of Grid Job execution, the same job script is submitted to the grid system. The number of nodes is added to the grid system by executing *pbs_mom* on the execution nodes. When the node is added into the cluster, the job script is submitted by the client node and the readings are noted. Due to variations in the reading an average of two readings is taken for each number of execution nodes in grid system. The grid job is executed for 60 minute video and 84 minute video.

For studying the effect of change in job size on the performance of the grid in terms of job completion time, the slice size is varied as 2 minute, 4 minute and 6 minute video. The experiments are conducted for with varying slice size for 60 minute and 84 minute video. From the performance chart it is clear that as the number of nodes is added into the grid system, the execution time to execute the grid job keeps decreasing and performance is increased. The performance becomes saturated with addition of 8 nodes and further addition of nodes has very less effect on the performance of grid system.

Table II. Execution time for different slice size for 60 minute video

Number of Execution nodes	Execution Time (in sec)		
	Fragment size= 2 min	Fragment size= 4 min	Fragment size= 6 min
1	558	253	163
2	311	170	129
3	309	136	97
4	288	154	87
5	256	125	86
6	277	116	98
7	252	184	98
8	242	114	97

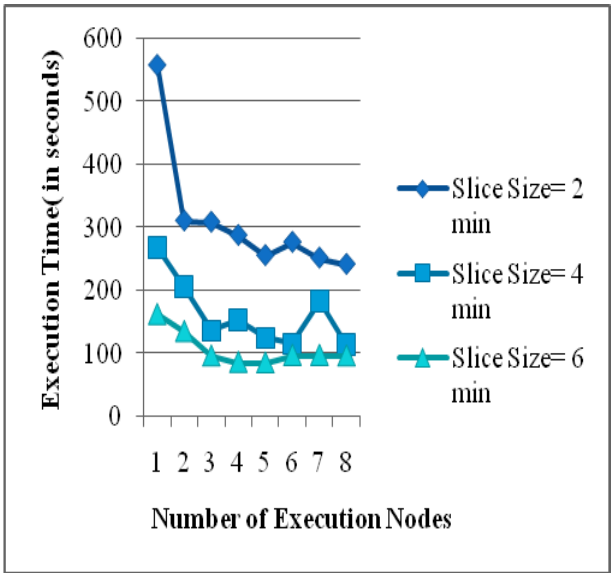

Fig. 1 Performance Comparison for Different Slice Size for 60 minute video

Table III. Execution time for different slice size for 84 minute video

Number of Execution nodes	Execution Time (in sec)		
	Fragment size=2 min	Fragment size=4 min	Fragment size=6 min
1	719	357	221
2	596	383	165
3	410	215	142
4	414	199	118
5	399	180	117
6	365	171	109
7	357	174	129
8	342	161	117

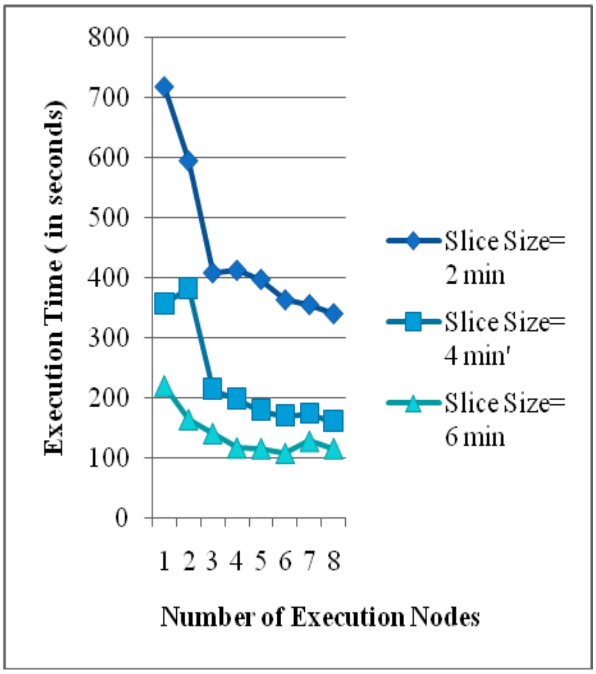

Fig. 2 Performance Comparison for Different Slice Size for 84 minute video

In both cases it is observed that as the slice size is increased the execution time is decreased substantially. In 6 minute slice size, for the same number of nodes, it is observed that the execution time first decreases but after 5 execution nodes the execution time increases and becomes saturated with no further performance improvement. This is due to the fact that time is also required to slice the grid job. After 5 nodes the time required to slice the job becomes the same as the time required to submit the job onto grid and get back and assemble the results. So no further improvement in results is expected in this grid job. There can be further improvement for any other type of job.

6. CONCLUSION

Grid systems are more suitable for the execution of jobs that can be partitioned and each fragment can be executed on separate execution node than running the same job on single or centralized system. The performance of grid system is highly dependent on how the job is submitted on to the grid. The job fragment size used has significant effect on the performance of grid. With the increase in fragment size the execution time for completing the job is decreased. The performance improvements need to be studied on faster networks with high bandwidth. Further the time required to slice the job into sub jobs may also have significant effect on the how the grid system depends on fragment size variations.

7. REFERENCES

[1] European GRID Forum, http://www.egrid.org, October 2001.

[2] The grid forum, http://www.gridforum.org, October 2001.

[3] Smarr, L., and Catlett, C. E., 1992, Metacomputing, Communications of the ACM, 35(6):44–52, June 1992.

[4] Livny, M. and Raman, R., 1999, High-throughput resource management. In I. Foster and C. Kesselman, editors, The Grid Blueprint for a New Computing Infrastructure, pages 311–337. Morgan Kaufmann.

[5] Foster, I., Kesselman, C. and Tuecke, S., 2001, The anatomy of the grid: Enabling scalable virtual organizations, International Journal of High Performance Computing Applications, Volume 15, pp. 200-222.

[6] What Is Grid Computing, http://www-1.ibm.com/grid/about grid/what is.shtml/

[7] Cheng, K.-W., Yang, C.-T., Lai, C.-L., Chang, S.-C., 2004, A Parallel Loop Self-Scheduling on Grid Computing Environments. In: Proceedings of the 2004 IEEE International Symposium on Parallel Architectures, Algorithms and Networks, KH, China, pp. 409–414.

[8] Baker, M., Buyya, R. and Laforenza, D., 2002, Grids and Grid technologies for wide-area distributed computing. Software Practice & Experience 32, 1437-1466.

[9] http://www.grid.org/

[10] Grid computing - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/ Grid_computing

[11] Guido J., Noordende V., 2008, 'A Trusted Data Storage Infrastructure for Grid-based Medical Applications", Eighth IEEE International Symposium on Cluster Computing and the Grid, pp 627-632.

[12] www.cloudbus.org/reports/EOSJ Article TR05.pdf

[13] Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications and High Performance Computing 15(3), 200–222 (2001)

[14] Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: HPDC-10. Proceedings of the 10th IEEE International Symposium on High-Performance Distributed Computing, pp. 181–194 (August 2001)

[15] Wen-Chung Shih, Chao-Tung Yang and Shian-Shyong Tseng, 2008, Performance-Based Loop Scheduling on Grid Environments, Springer-Verlag Berlin Heidelberg, Lecture Notes in Computer Science, Volume 4759/2008, pp. 238-245

[16] Lorpunmanee S., et.al. , 2007, 'An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment", Pwaset Vol. 23 Issn 1307-6884.

[17] Sidhu J. and Singh S., 2010, 'Design, Implementation and Evaluation of Grid Environment for DV to MPEG4 Video Conversion", International Journal of Computer Science & Communication Vol. 1, No. 2, pp. 151-155

[18] Fernando P., 2009, 'Video Compression: Discussing the Next Steps", 978-1-4244-4291-1/09, IEEE, pp 1582-1583.

[19] Ping Li, Veeravalli B., and Ashraf A.K., 2005, 'Design and Implementation of Parallel Video Encoding Strategies Using Divisible Load Analysis", IEEE Transactions On Circuits And Systems For Video Technology, Vol. 15, No. 9, pp 1098-1112.

[20] Gurjit Singh, Dinesh Kumar, 2012, 'Performance Evaluation of Globus Grid Environment Using AVI to Flash Video Encoding'', International Journal of Computer Science and Technology, Volume No. 3, Issue No. 2, ISSN: 0975-9646, pp.3360-3364.