Visualization Pipeline and architectural implications on Grid Environment

Aboamama Atahar Ahmed
Universiti Teknologi Malaysia
FSKSM 81310, UTM Skudai, Johor Bahru, Malaysia
+607-55556503
Atahar74@gmail.com

Muhammad Shafie Abd Latiff,
Universiti Teknologi Malaysia
FSKSM 81310, UTM Skudai, Johor Bahru, Malaysia
+6075532006
shafie@fsksm.utm.my

Kamalrulnizam Abu Bakar
Universiti Teknologi Malaysia
FSKSM 81310, UTM Skudai, Johor Bahru, Malaysia
+607-5532382
kamarul@fsksm.utm.my

Zainul Ahmad Rajion
Universiti Sains Malaysia
SDS, USM, 16150 Kubang Kerian, Kelantan, Malaysia
+609 766 3764
zainul@kck.usm.my

ABSTRACT

The distribution of visualization pipeline components on the grid environments produces numerous implications. Some of these implications are related to the architectural design of these components in the grid services and required compatibility with these components. Additionally, even with the integrations of existing network monitoring tools to provide support to implement a good grid computing architecture there still issues specifically with the development of real time visualization applications and systems on the grid. This paper presents the architectural implications and conflicts that are related to the performance and construction of visualization pipeline on the grid. The experiments were carried out to investigate and justify these issues with the presented results from the developed grid architecture for sequential resources selections and parallel resources selection.

General Terms
Algorithms, Measurement, Performance, Design.

Keywords
Sequential Processing, Parallel Processing, Visualization Pipeline, Grid Computing

1. INTRODUCTION

Despite the advances of visualization algorithms and the presence of high speed networks the visualization of complex datasets remains computationally expensive. The performance of visualization algorithms running on powerful clusters and specialized design supercomputers are around for over 2 decades now without filling the gap of providing accurate and interactive visualization to remotely located users. These facts resulted from the direct attempts of visualization techniques on standalone existing high cost hardware. The unfortunate news is that, most of the presented solutions are in the scope of an attempting to give an optimal solution in the techniques that used to perform the rendering or surfacing or even ray casting the large datasets on one single or a group of interconnected machines [1], [2], [3], [4]. Several studies were presented in the past decade to propose and optimize the performance of the visualization algorithms and the architectural adjustments to suite the visualization requirements [5], [6]. Some of these studies proposed sequential processing techniques and algorithms to work on one single powerful super computer [7], [8], [9]. Even though some of these attempts were trying to justify and implement the parallel techniques in their implementations, but the outcome were ended up designing a cluster of expensive hardware. Some other implementation focuses on client server solution for this problem which is seems to be far from introducing cost effective solution to the visualization community.

Figure 1 Sequential Visualization pipeline Selection

2. SEQUENTIAL VISUALIZATION PIPELINE AND GRID SYSTEMS

The sequential is a sequenced execution of visualization operations distributed in the grid as independent modules. These independent modules should have already been configured on each node that is expected to participate on the pipeline before the launching of grid enabled services. The shared scenario between the implementations of sequential visualization pipeline and parallel visualization pipeline is in the resources discovery stage.
The discovery stage has no effect on the usage of what type of processing and it is done in an independent fashion. The effect of the discovery stage on the selection is on the possible number of utilized nodes in the pipeline. Due to the fact that the resources selection stage is an important for optimal utilization of the involved resources and for speeding up the process as well as gaining higher performance, therefore the selection process in the sequential pipeline for the architecture is based on the selection technique. With the existing visualization techniques and algorithms which are associated with a very limited support for sequential processing on the existing high speed networks and with the presence of high speed networks. It is possible to develop reliable architecture to support the requirements of interactive visualization on the grid environment. The introduced complications with the existing grid computing implementations are existed in the absence of considering the heterogeneity factor. This heterogeneity factor is varying from one node to another according to some other characteristics on the machine specifications. Additionally, the cost of communication is also adding extra costs to any architecture that is trying to utilize the existing networks for any network based solutions. Based on the above facts it is necessary to consider the computational differences and to what degree these differences are affecting the overall performance of the architecture on the grid.

4. VISUALIZATION PIPELINE REQUIREMENTS FOR RESOURCES SELECTIONS

The requirements for resources selection stage from visualization pipeline perspective include how the visualization pipeline components are automatically constructed to cooperate and answer the remaining architecture components requests. The design consideration of the visualization pipeline components should be made according to the overall design of architecture and not to the randomly placed components in the distributed machines. In addition to other important factors that give great influence on the performance of the architecture such as the priority of selection between the visualization components as in the visualization pipeline where it consist of several modules and stages where the task of assigning processes such as Surface Extractors to the specified data Readers should consider the distance location of the selected for Extractors and the power of each selected machine representing for extracting process. However, there are several other factors involve in visualization pipeline components selection in the grid some are completely related to the grid requirements and techniques are impeded in the visualization client. Some other factors are related to resources discovery mechanism where the network related issues are taken into account and these factors are normally related in selecting the closest powerful node other than selecting far a way limited resources node to be a part of connected visualization grid nodes such as surface extracting process. Additionally, the size and complexity of selected datasets have a great impact of the selected components and it is also related to the discovery mechanism.

5. GRID ARCHITECTURE REQUIREMENTS FOR RESOURCES SELECTIONS

The requirements for grid architecture in the selection process are more sophisticated than the visualization pipeline requirements. This is due to the heterogenous nature of the grid tools and components not only from the integrated visualization pipeline components but also with the internal components of the grid containers where the initial designs of the grid containers consider the heterogeneous aspect of the software. For that reason the process of building heteronymous system on the top of another heteronymous considered as adding extra complexity to already complex system. However, avoiding this scenario as in this particular problem is possible in only two ways which is either the
developed the grid system in a specific grid container. The other way to is embedding or integrating the grid functionality and services in the new designed software which considered as better solution. The first choice is not practical solution due to several reasons such as grid containers are always changing their services in a form of upgrading their functionalities and putting the previous releases out of support. The other contributing factors are the change operating systems which also puts the older compatible releases out maintenance support. From the above scenarios we can come to conclusion that the grid enabled systems and applications should consider integrating the grid functionalities in the system itself and not to design the system to run on a specific grid container. As for our visualization pipeline case the components of the pipeline are designed as grid components and the required services are integrated in the initial design of the pipeline. Furthermore, the requirements for grid architecture for resources selection are very important and it comes to work if the previous stages such as resources discovery process are implemented accurately. However, the main important factor in the selection process is the ability to differentiate between the available machines not only in the static specifications such as memory, computation power and storage but also in the current load of these resources. The process real time reporting the machine current load is now possible with the availability of very sensitive monitoring tools such as Ganglia [10] and NWS [11]. Although some grid containers are trying to utilize some of these tools but faced with difficulties of integrating these tools in the containers for several out of control reasons. Some of these reasons are in a form of difficulties in providing generic solutions for the grid applications developers where the grid containers developers are not able to predict the possible usage of the developed containers and services. This situation puts the grid enabled applications developers in a very hard task where they must modify their applications to suite the containers and that may affect the overall applications purpose. Additionally, there are some other requirements for real time applications that must be taken into account such as the case for real time visualization systems specifically remote real time visualization systems, where the transferred datasets are normally large in size and algorithmically complex. This problem makes the rendering of the datasets requires the availability and support of real time interaction. The used techniques for providing the real time remote rendering of large polygons on one computer screen suffers from several drawbacks as the memory constrains and this particular problem normally puts the machine in unresponsive state. The relationship between resources selection process and this particular problem is very important issue in grid environment. The conventional solution to this is providing a cluster of machines to do the rendering and stream the produced images to the clients which may be located remotely. Unfortunately this solution is not practical in grid environment where the nodes are not fixed and the users will not be able to know the existing resources before the rendering jobs are sent for processing. The selection process in this case must be resources sensitive and the response must be very in order to avoid the delay in the rendering. Additionally undesired delay will results in having some part of the datasets being rendered and the other part will be still under process in the grid pool and that also will produce an incomplete visualization. The fast response for resources selection process is completely depends on the used algorithm and technique for resources selection in addition to other dependencies for grid which are related to the selection process such as the dependent on fast resources discovery and fast datasets transfer techniques. In the context of real time visualization the optimal utilization of resources is also a contributing factor for solving several problems such real time Isosurface extraction or distributed rendering. The used algorithm for selection process should be designed with consideration of the attributes of the analyzed datasets. As our experience showed that the different sizes and complexity of datasets results in different selection possibilities although the experiment was conducted on same environment, but the results were different.

![Figure 3](image)

Figure 3 shows the two selected extractors for 51.2 MB CT scan datasets.

Figure 3 shows the number of automatically selected machine to perform the extraction as distributed visualization pipeline processes on the grid. However, for this particular datasets which is in this case 15.1 MB of CT scan of full human head read as raw skeleton consists of 121 slices of 256×256 × 256. The produced number of polygons is changing by changing the Isolevel (Isosurface surface value). Analyzing the same datasets in larger number of grid nodes will result in reduction of time needed to process the datasets. For the above figure the used number of machines was only two machines as specified in the figure mewah.fsksm.utm.my and kulai.fsksm.utm.my as Isosurface extractors in addition to reader which is implemented in the Skudai.fsksm.utm.my node. The performance after the selection process is shown which was faster than one single machine, with adding more machines to perform the processing for the same datasets and partitioning the datasets according to it is attributes the number of selected machine will be increased due to the availability of more computational power to be involved in the extraction process. Where we see the first two previously selected nodes which are mewah.fsksm.utm.my and kulai.fsksm.utm.my and adding more nodes to provide the required computational power needed to perform the extraction process. The required power in this case is not limited to CPU cycles but it includes the memory for every selected node, where the comparison is made for the both memory and computational processing.
Figure 4 shows the Performance of distributed Isosurface Extraction between the nodes

In addition to the different selection requirements the architecture should be associated with load balancing techniques between the nodes as in the case of the figure 4 the number of processed polygons for some nodes is higher than the remaining nodes that always resulted by the partitioning process of the datasets. Due to the fact that the medical datasets specifically conventional CT scans and MRI datasets are supplied as slices which is always presented in large number of numbered sequence of files. The partitioning process for this type of medical datasets is done based on number of files or number of slices, as for the implementation of our partitioning techniques involved in resources selection process. The distributed partitioning for the datasets will result in assigning the available nodes with equal number of slices for every node. However, the datasets itself have different density in it is nature where some parts of datasets are rich in details as in figure 4 and some other parts are very low in details and it is impossible fact to predict that before the processing stage. This fact is shown in figure 5 where the number of processed polygons by the extractor 3 is larger than the number of polygons processed by the remaining extractors the reason for that is due to the number of produced polygons and not related to the number of slices where the number of slices are partitioned equally between the selected nodes.

6. CONCLUSION AND FUTURE WORK

We presented architectural implication that affects the performance of the visualization pipeline on the grid. We presented the relationship between the resources selection mechanism for real time visualization pipeline and the sequential pipeline selection. The support of parallel pipeline selection is presented with the actual implementation of the prototype. We show how the resources selection of both sequential and parallel could be implemented and the effect of their implementation on the grid. The future work of this research is in the direction of implementing the parallel rendering techniques and their effects on these selections techniques.

7. ACKNOWLEDGMENTS

This research is supported by the Ministry of Science, Technology and Innovation Malaysia and collaboration with Research Management Centre, Universiti Teknologi Malaysia

8. REFERENCES


