Wadi AlShatti University Journal for Pure and Applied Science

مجلة جامعة وادي الشاطئ للعلوم البحتة والتطبيقية

Volume 2, No. 1, January-June 2024 Online ISSN: 3006-0877 المجلد 2، الاصدار 1، يناير - يونيو 2024

COMPUTER ENGINEERING

Optimal Visualization Pipeline Partitioning Technique on the Grid Environment

Ibrahim Abdullah [1,*]

[1] Higher Institute of Science and Technology Zawia, Libya

ARTICLE HISTORY

Received 13 November 2023
Revised 10 October 2023
Accepted 14 October 2023
Online 01 January 2024

KEYWORDS

Visualization pipeline;
Grid computing distributed;
visualization techniques;
Partitioning.

ABSTRACT

Remote visualization of large datasets requires the development of distributed visualization pipeline. Additionally, the computational environment should provide easy configuration and automatic selection technique for the pipeline according to the resources available. Grid computing provides the required infrastructure with limited services particularly for remote visualization. The current approaches for grid enabled visualization follow static and manual selection and mapping of the resources. This paper investigates the mechanism for predicting and partitioning the visualization pipeline on the grid environment. We propose a grid computing technique for dynamic partitioning for visualization pipeline with consideration of network bandwidth and total time delay between the visualization pipeline components.

التقنية المثلى لتقسيم انابيب العرض في البيئة الشبكية

إبراهيم عبدالله العكريمي [1]

الملخص

يتطلب التصور عن بعد لمجموعات البيانات الكبيرة تطوير خط أنابيب التصور الموزع. بالإضافة إلى ذلك، يجب أن توفر البيئة الحسابية تكوينًا سهلاً وتقنية اختيار تلقائية لخط الأنابيب وفقًا للموارد المتاحة. توفر الحوسبة الشبكية البنية التحتية المطلوبة مع خدمات محدودة خاصة للتصور عن بعد. تتبع الأساليب الحالية للتصور الممكّن للشبكة التحديد الثابت واليدوي ورسم خرائط الموارد. تبحث هذه الورقة في آلية التنبؤ وتقسيم خط التصور في بيئة الشبكة. نقترح تقنية حوسبة شبكية للتقسيم الديناميكي لخط أنابيب التصور مع مراعاة عرض النطاق الترددي للشبكة والتأخير الزمني الإجمالي بين مكونات خط أنابيب التصور.

الكلمات المفتاحية

خط أنابيب التصور
الحوسبة الشبكية الموزعة
تقنيات التصور
التقسيم.

Introduction

A remote visualization system potentially enables the end-user equipped with a simple display device and network access to visualize large volumes of scientific data stored and/or rendered at remote sites. The size of datasets is large enough and it is impractical, or impossible to transfer the entire dataset to the workstation and have good visualization. However, in the existing visualization applications specifically modular applications it is possible to distribute the visualization processes into several connected components or modules. Our observation show the possibility to divided the visualization process into sub-processes or subtasks. Every subtask assigned to a module or divided between modules. The connected visualization modules work together as pipeline. Therefore, a mechanism should be developed to tackle the delay caused from partitioning visualization pipeline and the network latency for transferring the datasets between the pipeline modules. Additionally, in the current grid competing methods the infrastructure is provided but with major drawbacks in the process of selecting appropriate resources for visualization pipeline modules. The current approaches are implementing static selection of participating nodes. The need for automatic mechanism for selection the pipeline components is obvious. For this reason the optimal formation and distribution of the visualization pipeline is needed. Additionally, the pipeline should be associated with prediction process to predict the efficiency of the pipeline components. On the other hand, the efficiency of components needs to be predicted before launching components. This paper is an attempt to describe the methodology involved in automatic selection and formation of visualization pipeline on the grid. Our particular attentions are on partitioning of the visualization pipeline. First section of this paper discusses about the limitation of current grid architectures and the benefits provided by grid to the visualization pipeline. Section 2 is a detailed description of related work that focuses almost exclusively on grid enabled visualization applications. Section 3 is a description of Visualization Pipeline partitioning problems and grid. Section 4 focuses on types of performance prediction methods. Section 5 highlights our proposed algorithm for automatic visualization pipeline selection with description of related techniques.

*Corresponding author: Ibrahim.00ly@gmail.com

Most of scientific visualization applications offer the formation of visualization pipeline. This formation is easily observed in modular visualization application such as VTK (Visualization Toolkits) [1] OpenDX and AVS. The process of formation of visualization pipeline is done by dividing and distributing the application components into a number of network connected nodes. However, several studies were carried to extend most of existing visualization application to be grid enabled. gViz project [2] is attempting to extend IRIS Explorer [3].These applications usually suffer from a number of drawbacks such as limited functionality such as ability of visualization in specific environment as in gViz and IRIS Explorer. Other grid enabled visualization such as RAVE [4] which is designed to react to changes according to the resources available on the requester machine. The RAVE internal components designed to stream images if the requester is limited resources client or to transfer geometry to client machine when it support rendering. E-Demand [5] is pure implementations of grid services to support stereoscopic visualization in a distributed environment. The idea is to allow multi stages of rendering between the collaborated resources on the grid. The SuperVise [6] in SuperVise the implementation focuses on the distributed visualization pipeline on the grid. The methodology is to select the source of data and manually select the appropriate resources. This selection process should be carried for every components or components of the visualization pipeline. However, most of the mentioned grid enabled visualization applications are well structured and designed to solve specific problem. Majority are sharing the application level workflow, built on the top of modular visualization systems to utilize the existing distributed components. Despite the fact that all of the applications fallow the existing dataflow concept presented by Haber and McNabb [7] as described in Fig.1. Almost the entire presented grid enabled applications use manual selection process of the visualization pipeline components.

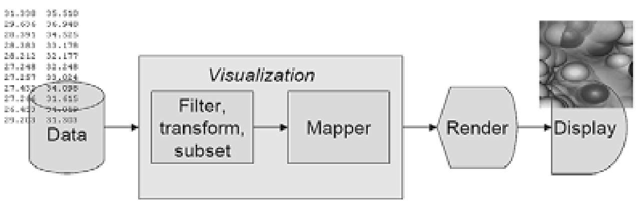


Figure 1: Haber-McNabb Visualization Pipeline

Visualization Pipeline Partitioning Problem and Grid

In this section we provide closer look at visualization pipeline portioning problem. Scientific visualization community will never have significant benefits from grid computing as long as the process of manual selection of visualization pipeline is implemented. That is because the partitioning is static, which means the partitioning never changes in response to changing application needs or environmental conditions. That is not practical with the fact that grid is unlimited number of unreliable resources. Therefore, unless we develop reliable automated methods to select appropriate visualization pipeline distributions, we stand very little chance of deriving any tangible benefit from the Grid computing infrastructure [9].

However, there is no a-priori way to select an optimal (or even tolerable) pipeline distribution at startup without first being able to accurately predict the performance of the individual components [9]. The performance visualization in general critically depends on how efficiently the visualization pipeline is mapped onto the network nodes. Based on these facts, in order to have stable and efficient remote visualization we need to overcome the predetermined partition technique of the visualization pipeline employed by existing remote visualization systems. The common scenario for this is when the server machine sends fixed stream such as raw, data, geometric primitives, or frame buffer (FB) to remote clients as in as OpenGL VizServer 3.1. This scenario is not successful. Specifically, when the server sends large datasets over wide-area connections, in this scenario the user required to have a detailed knowledge about the participating grid nodes. To map which visualization pipeline component is appropriate for which job. The other similar scenario is when we have server sends part of the visualization process to the client to finish the job as in COVISE [10].

There are two main problems with proposed techniques. First is the grid nodes are unreliable and may change at run time. Secondly, with large number of participating node that makes the manual selection process far from practical. [8] Describes the role of components based optimization for visualization pipeline. The methodology was based on analytical model of the pipeline. The overall optimization of the visualization pipeline consists of individual optimization of the pipeline components. For that reason a mechanism is built on the top of high level layers to query information about the individual distributed pipeline modules. However, [11] in their description of how the visualization pipeline can be mapped into the network nodes to specify which task is suitable for which node. Their description was given as analytical methods for mapping. The idea was based on which function should be mapped to which network nodes.

The main purpose for this analytical method is to minimize the total delay for the data between the modular components and to maximize the frame buffer between the connected nodes. As a result of above described techniques the combination of the two techniques will accurately give sufficient performance prediction functionality for our proposed visualization pipeline. That is taking into consideration the network- bandwidth, total delay, and the nature of distributed components. However, the implemented technique is not as same as described technique in [11]. This study focuses for utilized VTK visualization tool to demonstrate reconstruction of medical image using surface rendering technique [12]. However, the complexity of graphical applications has increased considerably in the past years that make them impossible to be executed on computers with limited resources remote visualization could be a solution for this problem [13]. Where they have implemented the technique for finding the optimal based on excluding the non-optimal components Fig. 2.

Performance Prediction Methods for Visualization Pipeline

In order to accurately predict the overall performance of the pipeline, we should create performance model and associate the model with each component that comprises a pipeline. The performance of a visualization pipeline varies dynamically as a function of input data, user parameters, and environmental conditions. Therefore, the performance model should continuously estimate pipeline performance of visualization. Based on that, we need to define scope of variables needed to accurately model the performance of a visualization application consisting of software components deployed on the grid. The following is the evaluation of existing performance prediction methods and their applicability to the visualization pipeline scenario. Statistical/heuristic methods the idea of this technique is to reduce the size of the performance metric space and complexity of the model by employing statistical correlations. Statistical method is implemented as follow.

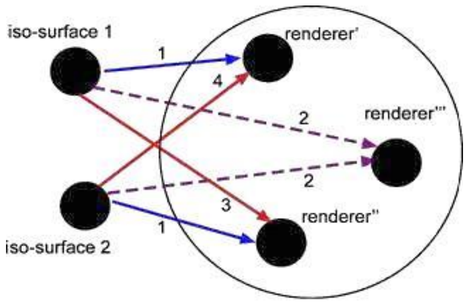


Figure 2: Visualization pipeline Techniques based on non-suboptimal selection

First an algorithm is characterized by its pattern of memory references and computational intensity. Then one uses a simplified code that derives a set of parameters that characterize a given architecture. These parameters are then fed into a statistical model that can predict the performance of the original algorithm on architecture without actually running it there. Much of this work is highly experimental and still under development [2].
Second performance method is history-based performance prediction method. An example of this method is Network Weather Service (NWS) where the predictions of component performance using historical logs of performance information. The third method is analytical technique to predict the performance. This technique is implemented using minimal set of input parameters to predict algorithm's performance. Additionally, it has been observed that in scientific visualization simple changes in the visualization parameters will result major changes in the visualized object. Therefore using large number of visualization parameters will cause difficulty in applying any of previous methods specifically analytical method. Unless we use limited number of visualization parameters and this will bring the Analytical method to the top of the list among the other methods. Therefore, in this paper we focus on applying the analytical method. Additionally, we carefully take into consideration the involved visualization parameters

Visualization Pipeline Optimization

The distribution of the visualization pipeline on the grid is shown in Fig. 3. At the display side the user will not be worried about how the visualization pipeline is configured and distributed. The user will only locate the datasets he/ she interested in. Then the application will automatically react and respond by configuring the optimal visualization pipeline and display the visualized data in the fastest time possible. The formation of optimal path pipeline is done by measure the network efficiency by known the size of selected datasets and the location of the datasets in the grid. The proposed technique for optimizing the pipeline is started after the process of selecting the source of datasets. The searching for the optimal is done based on based on two main optimization problems. First optimization problem is to minimize the total delay between the grid nodes. The second optimization problem is to maximize the frame rate between. For the selection of the first node of the pipeline which is filter Service calculation of the lowest time to deliver the data from data service to the filter service. This process will be done on the pool of nodes participating on the grid.
The underlying implementation will utilize the network latency by query the information services for networking information to formulate network latency predictions. From The resulted prediction we will formulate the nodes with lowest weight possible to be added to the pipeline.

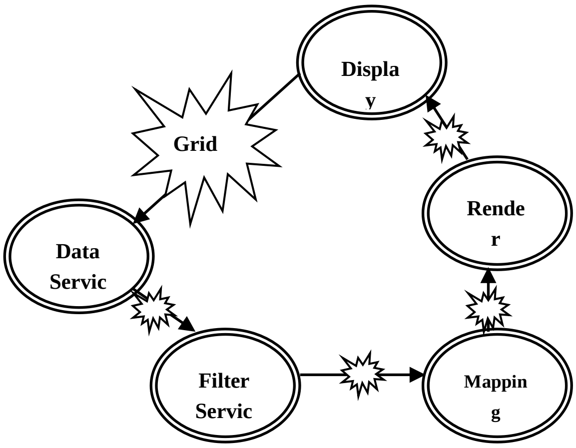
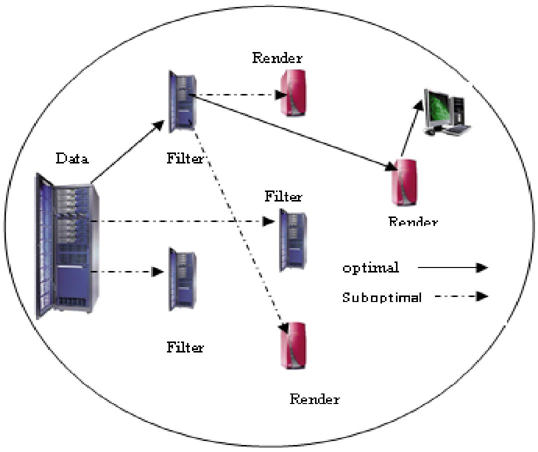


Figure 3: Distribution of the pipeline on the grid

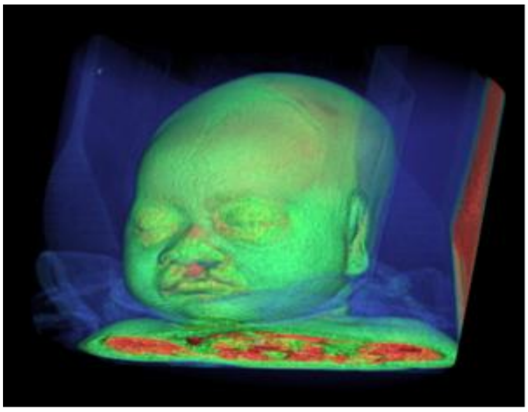
This process will also use to examine the power of nodes along the path of the configured pipeline. Thus the overall optimization of the visualization pipeline will have several optimization criteria. Figure 4 describe the optimization technique for the visualization pipeline. However, in our current implementation every module of the pipeline must contain VTK library and pipeline will be formed in response to the launched grid services at the startup process. The grid services used to query information from individual participating nodes. Additionally, all the participating machines should already been registered as a grid nodes in order to be queried by the previous node along the path of the pipeline.
However, the initial test model was done on visualizing medical CT scan and MRI datasets. Two datasets were used for our initial testing purpose. One contains a CT scan of a human data with 512 x 512 x 21 voxels. The second is MRI Brain data with 256 x 256 x 113 voxels. Figure 5 (a) shows volume rendering of with shading and (b) cross section of MRI datasets of human brain.

Conclusion

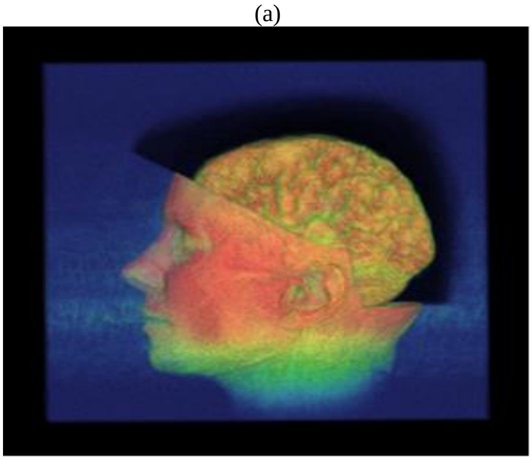
Grid-based visualization tools must be developed to support remote. Additionally, successful remote visualization requires overcoming two main problems. Low frame rate and time delay between the connected modules. We presented the analytical method for predicting the optimal visualization pipeline. Currently we have configured grid testing environment. This environment will be used to develop and deploy distributed visualization service components. Our work focuses on developing the modules and that will be distributed in the grid. The proposed architecture currently supports all data type as long as they are accepted by VTK library. Our future plan is to investigate the integration of this clients and modest resources devices. Our ultimate goal is to develop a framework for distributed visualization for medical datasets on the grid environment.



Figurer 4: Optimized Visualization pipeline Techniques based on suboptimal selection



(a)



(b)

Figure 5: Sample of visualizing medical test model (a) CT scans of Head - Size in voxels 512 x 512 x 21 and (b) MRI Brain Size in voxels 256 x 256 x 113

Funding: "This research received no external funding."

Data Availability Statement: "No data were used to support this study."

Conflicts of Interest: "The authors declare that they have no conflict of interest."

Acknowledgements: "This work was supported by management &Science University (msu) and Galactic Bridge Research & Technology (GB)".

References

[1] W. Schroeder, K. Martin and W. Lorensen, "The design and implementation of an object-oriented toolkit for 3D graphics and visualization". In Proceedings of Seventh Annual IEEE Visualization'96. pp. 93-100. IEEE. October, 1996.

[2] J. Wood , K. Brodlie., and J. Walton., "gViz–Visualization and Steering for the Grid". In Proceedings. e-Science All Hands Meeting, 2003.

[3] J. Walton, NAG's IRIS explorer. Visualization handbook, Jordan Hill Road Oxford OX2 8DR, United Kingdom, 2004.

[4] I. J. Grimstead, N.J. Avis, D.W. Walker, "RAVE: the resource-aware visualization environment ", Concurrency and Computation: Practice and Experience, vol. 21, no..4, pp.415-448, 2009.

[5] S. Charters., N. Holliman, and M. Munro, " Visualisation in e-demand: A grid service architecture for stereoscopic visualisation". In Proceedings of UK e-Science Second All Hands Meeting, September 2003.

[6] J. Osborne, H. Wright, "Supervise: Using Grid tools to support visualization". In Proceedings of the Fifth International Conference On Parallel Processing and Applied Mathematics, 2003.

[7] R. B. Haber, D.A., McNabb, "Visualization idioms: A conceptual model for scientific visualization systems". Visualization in scientific computing, vol .74, pp.93, 1990.

[8] I. Bowman, J. Shalf, K. L. Ma. E. W. Bethel, " Performance modeling for grid-based visualization", submitted to Parallel Graphics and Visualization.2004

[9] J. Toole, C., and L. Moore, "Brokering Grid Services for Distributed Visualization". In Proceedings of the Compframe Workshop, Pp 1-5, 2005.

[10] A. Martin, L. Ruth, R. Daniela, S, Jürgen A, D. Werner, W. Peter, W. Uwe. "COVISE Features" [Online]. Available: http://www.hlrs.de/organization/vis/covise/features/,HLRS, September 2000.

[11] L.K. Wee, H.Y. Chai, and E. Supriyanto, "Surface rendering of three dimensional ultrasound images using VTK". Journal of Scientific & Industrial Research, vol 70, no. 6, pp.421-426.

[12] D. Gorgan,. O. Capatana,, "Remote visualization of 3D graphical models". In 2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, pp.263-268, 2013.