Visualization of the Distributed Data of Huge Volume. Assembly, Filtration, Sorting

Dmitri Manakov, Alexey Mukhachev, Alexey. Shinkevich
Institute of Mathematics and Mechanics,
Ural Branch of the
Russian Academy of Sciences,
Ural State University, Yekaterinburg, Russia
manakov@imm.uran.ru

Abstract

In this paper some possible approaches to realization of huge volumes of the data representation are considered. A filtration in association with interactiveness, problem-oriented metafile generation and parallel execution of data filtration algorithms is in our opinion the most effective approach to reducing of huge volume of distributed data. One of the purpose of this work is a development of the demonstration tests for assembly, filtration, sorting of the distributed data of huge volume with the subsequent visualization.

Keywords: On-line Visualization, Filtration.

1. INTRODUCTION

The development of parallel computing puts before computer graphics number of new problems, some of that did not appear earlier. Among them there are the following:

1. Organization the interactive graphics in the parallel programs. It is important to note, that the organization only on-line visualization is insufficiently. It is necessary to have a high-grade interactiveness [1]. That is we need in the realization of messages exchanges between users and parallel processes in the both sides as with input, and with output of various type data, including texts, numbers and graphics.

2. Visualization of huge volume of data.
The huge data files may be regenerated as a result of solving such mathematical physics tasks as gas and hydro dynamics. These tasks may generate terabytes files which can't be neither visualize nor even if send to workstation for acceptable time. Thus, to solve the problem of reduction the transmitted file volume we need in algorithms of file compression or its filtration directly on parallel processors.

3. Parallel Software Visualization.
On our opinion the tasks of parallel software visualization above all include the visual environments for parallel program developments and proper visual parallel languages. Also we need in means for visual debugging of parallel programs including systems of performance debugging and performance tuning.

4. Parallel algorithms of image (raster) processing.

In this paper we consider the problems of huge data file visualization and more detail some approaches to parallel filtering of these files.

2. THE GENERAL APPROACHES TO VISUALIZATION OF HUGE VOLUMES OF DATA

The main task of visualization of huge volumes of data is the reduction of sizes of the transmitted files. There are two main approaches to this problem decision: the use of interactiveness and also compression or filtration of the data. The foreign literature contains some interesting descriptions of decisions basing on this approach (for instance [2]), but as usual only questions connected with raster image generation are considered there. On our opinion the transmission of raster files in some instance is not only effectless but also is unacceptable. It is more expedient to transfer a specially selected mathematical data and then to visualize them.

2.1 Interactiveness

It is known, that in some cases, for example during the decision of interpolation tasks, there can be some of the intermediate data, which are necessary for calculations but are insignificant at the final analysis of results.

Interactive means using in the parallel computing processes makes possible user determine what data are necessary for visualization and how these data will be represented: as the text or spreadsheet, as some forms of graphics or (if the data are not necessary) to exclude them from total volume of visualized data. It is clear, that for real tasks, when the total computing time is measured by hours and days, and the volumes of the data, which are operated with the program reach terabyte sizes, we can't consider interactiveness in naive understanding of this word (as a dialogue mode of human-computer interaction). However, for the above-stated reasons, injection of interactiveness elements to parallel computing also is useful. The possible decision of this problem is the realization of scenarios of interaction with the parallel program. For example if there is no response from the user for a long time, the interactive mode are aborted, and the prepared data are not visualized, but are written to files. This problem requires more detailed consideration that leaves for frameworks of the given paper.

2.2 Assembly, filtration, and sorting of the of huge volume of distributed data

The second main approach (or more exactly the whole class of the approaches) is the assembly, filtration and sorting of the data generated by parallel computing. Within the framework of the

given approach it is possible to show some actions which are carried consistently on each processor of the parallel computer:
The generation of the mathematical data as a result of performance of the given algorithms;
The application of the filter for processing of the given data (Here it is necessary to note, that the filter is applied to the data generated as a result of computations just on this processor and stored in the local memory of this processor);
The transmission of the filtered data to the workstation (probably remote) on which will be realized the final visualization.

It is necessary to note, that there is a correspondence between the filter and model of visualization, but it is not one to one. Thus to one filter there can correspond some models of visualization, and vice versa.

3. FILTRATION AND COMPRESSION

As it was mentioned above one of the possible approach to processing of huge data files is the preliminary filtration and compression of data before their subsequent visualization. There are two alternative variants of data compression [3] [4], namely: data compression with loss or without loss of quality. The major drawback of such approaches is that the compression without loss of quality can't reduce the transmitted file sizes on adequate orders. And the compression with loss of quality may be the reason of the data distortion that frequently is inadmissible. In this connection the filtration of the data with an opportunity of changes of initial filter parameters (depending on a concrete task) is most preferable. Let's notice, that in this case it is possible to look the transfusion of classes on which the approaches to file volume reducing were subdivided. So the compression may be considered as some kind of filter.

It is possible to recognize such filters as decimation, multidimensional projection, section by a plane, and also filters focused on a concrete task and on a concrete model of visualization. Therefore it is possible to define two basic tasks: the general description of filter applications or technology of a filtration, and also development of filters for concrete tasks. More detail we shall consider a filtration on rather simple, but not trivial example - the section by plane.

4. MATHEMATICAL MODEL OF THE FILTER "SECTION BY A PLANE"

In three-dimensional space the grid with function value in nods is defined. It is supposed to use section by the defined plane to reduce the volume of data. As result volume of the data is reduced to dimension. Basing on resulting data a visualization system may generate 2-D surfaces and its isolines (Figure.1). Sections by a plane can be realized as for a grid defined by a irregular step on the axes Ox, Oy and Oz, and for an irregular grid. Let's consider briefly the section by a plane $Ax + By + Cz + D = 0$ for a grid with a irregular step on various axes. Let's consider a general case, when $A*B*C\neq0$. Let's go on z-axes with the defined step, that is we shall fix z. Let's consider movement from M0 to, Mn, where (M0, Mn) - intersection of a plane $Ax + By + Cz + D = 0$ and plane $z = \text{Const}$. We calculate points Mi under the formula Mi = (M0, Mn) intersection y =Const or X = Const.
Now we consider a general case for an arbitrary grid, when $A*B*C\neq0$. We shall choose only those points, which distance up to the given plane is more than zero and less e, that is $0 <D< E$.

In both cases we find a projection of a point to a plane, transforming coordinates so that distance between two points on a plane $Ax + By + Cz + D = 0$ and in space were identical.

It is necessary to note, that the application of the filter depends on a type of a grid. In the first case the algorithm of movement on a direction, in the second exhausting search are applied. As a matter of fact the same results turn out, but the first algorithm in N/4 of time is faster, where N dimension on one of axes.

It is important for the parallel programs to supply the data distribution on processes. Frequently the geometrical distribution is used. In our test example this distribution was defined as a some sort of line of parallelepiped. The general scheme of filter application the can be described in the following kind: the task of data distribution on processors, cycle including describing of the secant plane, parallel application of the filter dependent on a type of a grid, data assembly and data transfer for the visualization. When parameters of function of filter construction are transmitted to weaken dependence of the filter from concrete data the indexes on a function are used. Generally these parameters are determinate by users. This function returns coordinates of a point and value in this point up to the number of the point, it determines the action with the received results: for example, to print out values, to save data into files, to send data onto visualization. Thus, the parallel program was realized with using of MPI library, farm topology and oriented on geometrical data distribution. The filtration occurs independently on each processor that reduces number of exchanges. At the description of the general scheme of a filtration it is desirable to consider and other questions.

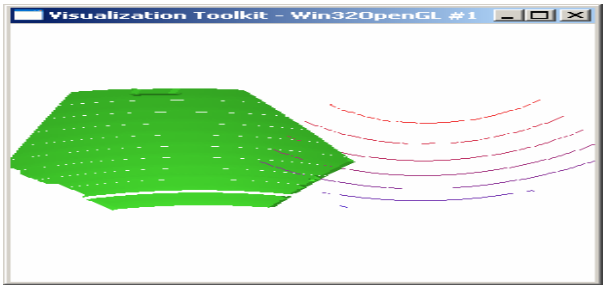

Figure 1: The filter "section by a plane".

5. TECHNOLOGY OF A FILTRATION FOR DISTRIBUTED DATA OF HUGE VOLUME

In this section we consider some questions connected with technology of a filtration for distributed data of huge volume. Three units (5.1-5.3) discussed below are based on our experience of realizations, and the last (5.4) consists of the next tasks of the research and development of filtration.

5.1 Efficiency of filters parallelization

Analyzing results of measuring of parallel program time characteristics in view of features of the parallel computer structure, it is possible to make a conclusion, that in this case the productivity essentially depends on communication channels, and the acceleration effect is achieved only when the number a grid nods is enough big or when labour-consuming algorithms of a filtration are used (for example, methods using exhaustive search). Despite of it, the fast algorithms of a filtration are more preferable, because the main goal is the appreciable reduction of the transmitted data, and the result at use of the fast filter is achieved for smaller time.

5.2 Realization of the parallel program on the basis of sequential, with use of library DVM (Distributed Virtual Machine)

As the model of parallelism used in tasks of data assembly and data filtration, is data parallel model, the logical step is to apply DVM system [5], which has the large opportunities for realization just this model. However during using of DVM system it was found out, that the DVM compiler imposes on the program as a whole and on used data some additional restrictions, which were not taken into account when the initial program code was developed. The overcoming of these restrictions also has caused some difficulties. For realization of parallelism in DVM to system we have distributed files x [], y [], z [] on processors. It has appeared enough, and additionally it was necessary only to place the references to the removed points in several cycles. Thus, we were convinced that DVM-system is suitable as well as possible for realization of similar tasks. And in this case the realization of immediate parallelism was not time consuming.

5.3 Comparison of using shadow sides and sorting

Depending on the concrete filter the choice of a point satisfying to a filtration condition can depend on the next points. Thus, there can be a situation, when the points laying on borders of areas of the data, filtered by each separate processor can differ from points laying on borders of areas, adjacent with given one. During the visualization of such data the visually appreciable breaks on borders of areas may appear (Fig.1). As result the undesirable artifacts of visualization may frequently appear also. To avoid the similar phenomena, it is possible to use so-called shadow sides. In this case points of adjacent areas involve to the choice of the point satisfying to the filtration condition. Thus the boundary point is considered as internal one. Or we may apply alternative variant - to sort additionally data on borders. The term "shadow sides " is widely used in DVM system and it has there the similar sense. Naturally, it is easier to realize shadow sides, applying just DVM system, here there is a special program means for this purpose. However, using of shadow sides increases loading by the communication channel between processors of the parallel computer. In this situation full filter performance may be adversely affected.

5.4 Unsolved problems

Using of distributed parallel file system (for example PVFS [6]) can be well-taken, because the data can be recorded parallelly, reducing full execution time of the filter, and visualization in any case executes sequentially.

To organize parallel interactiveness it is desirable to apply three-linked client-server model with the generation of problem-oriented metafiles. The analysis and account of concrete applied problems allow us to develop server of the graphic applications using filtration.

It is necessary to consider the descriptions of their own filters for each areas or processors and also to consider the applications of several filters consistently.

Realization of filters basing at the paradigm of object-oriented programming. To realize the filter we used such features of C syntax that allow to transfer the reference to the function as parameter of the function realizing the filter. However to decide a similar task it is possible to use another methods. For example, in frameworks of object-oriented programming paradigm it is possible to encapsulate numbers, coordinates of points and functions returning their values. Also the polymorphism allow the program to choose dynamically what filters to use depending on the signature of a concrete call. However, in the case of object-oriented techniques for the development and the realization of filters we may lost advantages of DVM system, which for today supports only simple data types.

There are some other unsolved tasks, among them: The evaluation of the maximal dimension dependent on number of processors; The preparation of the demonstration tests for technology of the filtration and more complex filters and, The decision of real applied problems.

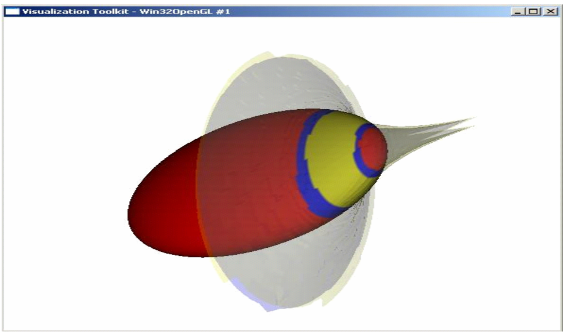

Figure 2: Filter realization for a body in curvilinear grid.

6. FILTER REALIZATION FOR A BODY IN CURVILINEAR GRID

In the prototype example the curvilinear grid was chosen. This grid is generated on the parallel computer. Then the various criteria for the data filtration are used and as result isosurfaces are formed. The filter data to a workstation are transmitted. As a model object in this example the ellipsoid was chosen and isosurfaces are constructed around it. Physical meaning of the isosurfaces may be for example the areas with the same pressure or others interesting phenomenon appeared in real problems. The grid is computed in the such manner: the ellipsoid is stretched out (with some step) into all directions. Everyone new ellipsoid we shall name as a layer. It is possible to consider (in this example) that the points having identical numbers in different layers are in relation to a surface of a body on the same straight line. The directing vector of this straight line is normal about a surface of a body.

The grid in the example is regular. Accordingly (for simplification of algorithm of the filter and program as a whole) the following scheme of a filtration was used: A chosen point on a surface of a body initially is checked, whether it satisfies to the condition. If it belongs to the isosurface it is added, if it is not a point with the same number from a above layer is checked again. The process proceeds as long as either all layers will be sorted out or the point satisfying to a condition will be found. It is possible to define this algorithm as a movement into the direction.

All layers are broken onto eight parts for simplification of triangulation algorithm. During the filtration on the parallel computer the interval is underlined for each processor. The nods of the grid computed on the given processor lie between borders of this interval.

Thus each processor filters the data in eight parts of a layer within the limits of the specified interval. The internal border of an interval belongs to two processors, that is the concept of shadow sides is used. Each eighth part of layer is broken on levels used for triangulation. If for the triangle a pair of next points was chosen from one level, the third point should be from the subsequent or previous level. The resulted data are saved as a problem-oriented metafile.

In our prototype realization the values of function in nods were computed with logarithmic (Figure 2) and elliptic (Figure 3) distributions. In a reality these values are the results of solving, for example, the system of the differential equations. In this case the filtration procedure and the concrete filter have been oriented on a real problem, that, probably, will change the parallel realization.

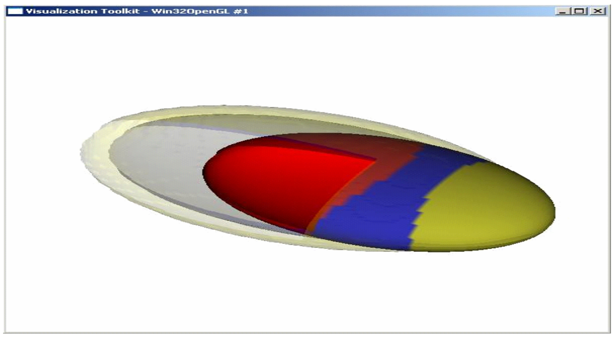

Figure 3: Filter realization for a body in curvilinear grid.

7. CONCLUSION

In this paper some possible approaches to realization of huge volumes of the data representation are considered. At this takes place the various methods of parallel computing realization are used. The filters described above, collected data directly from the parallel processor memory and transferred the information to a workstation for the subsequent visualization.

This approach has allowed to reduce the volume of transmitted data much more, if other approaches to reduction of the transmitted data (for example compression) were applied. A filtration in association with interactiveness, problem-oriented metafile generation and parallel execution of data filtration algorithms is in our opinion the most effective approach to reducing of huge volume of distributed data.

One of the purpose of this work is a development of the demonstration tests for assembly, filtration, sorting of the distributed data of huge volume with the subsequent visualization.

7. REFERENCES

[1] D. Manakov, M. Shagubakov. Adaptive Builder for Interactive Tasks in Mass-Parallel Machines //Proceedings of Graphicon 2002, Nijniy Novgorod, 2002, pp. 405-408 (In Russian)

[2] Philip D. Heermann. Production Visualisation for the ASCI One TeraFLOPS Machine //Proceedings of the 9th Annual IEEE Conference on Visualization (VIS-98), Oct 18-23 1998, ACM Press, New York, 1998, pp 459-482.

[3] Sukov S.A. Yakobovskii M.V. Processing of the three-dimensional not structured grids on multiprocessor systems with the distributed memory //Fundamental physical and mathematical problems and modeling technic-technological systems / Proceedings,STANKIN, 2003, release 6, 8pp. (In Russian)

[4] Abalakin I.V.,Boldyrev S.N. Jokhova A.V.Parallel algorithm of account gas dynamics currents on irregular grids //Fundamental physical and mathematical problems and modeling technic-technological systems / Proceedings, STANKIN, 2000, release 3, pp. 41-45. (In Russian)

[5] N. A. Konovalov, V. A. Krukov, Yu. L. Sazanov. C-DVM -A Language for the Development of Portable Parallel Programs //Programming and Computer Software, v 25, 1-1999, pp. 46-55.

[6] P. H. Carns, W. B. Ligon III, R. B. Ross, R. Thakur PVFS: A Parallel File System For Linux Clusters //Proceedings of the 4th Annual Linux Showcase and Conference, Atlanta, GA, October 2000, pp. 317-327

About the author

Dmitri Manakov is a lead programmer at Institute of Mathematics and Mechanics,
Yekaterinburg, Russia.
His e-mail is manakov@imm.uran.ru

Alexey Mukhachev is a student at Ural State University,
Yekaterinburg, Russia

Alexey Shinkevich is a student at Ural State University,
Yekaterinburg, Russia