The importance of locality in the visualization of large data sets

J.M. Brooke¹, J. Marsh¹, S.Pettifer¹, L.Sastry²
Department of Computer Science, University of Manchester,
2. CCLRC, Rutherford Appleton Laboratory

Abstract

GODIVA (Grid for Ocean Diagnostics, Interactive Visualization and Analysis) is one of the NERC e-Science projects focusing particularly on data analysis via visualization. The data analyzed in GODIVA comes from high-resolution global models of the ocean and atmosphere, thus datasets are large and there are many datasets in a time sequence. We describe the GODIVA architecture, which provides for visualization capabilities in a Web Services context where the visualization can be performed either by a Web Service or by an intelligent client which exploits the power of the new generation of programmable PC graphics cards. We utilise the concept of locality in a networking sense (bandwidth and latency) in the data and visualization flows. This architecture is directly related to the needs of the end-user scientists in the project who have to analyse complex physics in oceanography involving at least four dimensions in the data space. We also consider issues of locality and re-use in the computation and caching of derived data fields such as gradients and flow vectors, obtained from the original model fields via subsidiary calculations.

1. Introduction

1.1 Scientific Context

We address the problems of visual analysis of very large datasets, as derived in large scale atmospheric and ocean modelling. These datasets are produced by large and complex models, which aim to model the physical processes behind the flows of the atmosphere and ocean. This requires global models, which also have sufficiently fine resolution to capture physics at multiple length and time scales. Our work has particularly focussed on oceanographic data, where the OCCAM ocean model is now being deployed at 1/12 of a degree resolution in latitude and longitude with 66 levels in depth. At each point there may be multiple field values, e.g. density, salinity, temperature. Such models have traditionally used finite-difference methods of solution, which means that in order to resolve fine scales, the mesh has to be uniformly fine over the whole extent of the oceans. The models are costly to run as they require dedicated supercomputers and may need to be "spun up" over an extended period before they begin to reach a state where they can be regarded as representing the behaviour of the real ocean. They represent a resource of great importance to a substantial community, are rich in spatial and temporal information on fluid flow and are typically carefully stored in central repositories. The datasets require advanced visualization to aid understanding of the behaviour of the model.

The scientists may wish to visualize the data to bring out complex structures, such as the interface where two masses of water at different temperature are mixing and the scientific interest may lie in dynamic features such as the evolution of such interfaces and possibly the mapping of other quantities (scalar or vector) onto them. Some of these quantities may be directly present in the data but some of them may be derived from the raw data via computations on values at the grid points and their neighbours in space and time. Fluid velocity is an one such example requiring such computation, gradients are another.

1.2 The GODIVA project

GODIVA [1] (Grid for Ocean Diagnostics, Interactive Visualization and Analysis) is a NERC e-Science project, which is building a persistent infrastructure for the analysis of such datasets to enable the visual analysis to be performed locally on desktop or laptop computers access remote data servers via Web Services. The aim is to analyse the data in four dimensions (three of space and one of time) to discover behaviour and features in the model which can be checked against observation and which can also inspire the development of the models themselves. There are three specific science drivers

- 1. The relationship of small scale oceanic convection to large-scale ocean flows, which lies at the heart of many climate change issues
- 2. Heat and freshwater budget studies, and thermohaline water mass transformation analyses using assimilated
- 3. The dynamical adaptation of an ocean model to an unstructured density mesh and developing the ability to easily visualise the irregularly spaced data obtained

Each of these science drivers presents interesting challenges for a Grid organised around visual analysis using visualization techniques. In (1) the users need to shuttle between the large and small scales of the ocean. If we take the data for the whole ocean and render isosurfaces at suffient resolution to capture the small scale detail we will greate huge objects that have to be manipulated in memory and are likely to lose all interactivity even on the most powerful visual supercomputers. In (2) we have the need to map different fields within the same visualization, e.g. viewing the salinity distrubution over an isosurface based on density values since cold fresh water may have the same density as hot salt water. In (3) we require mappings from regular to irregular meshes and can take advantage of more efficient ways of storing information by adapting the number of mesh points to the amount of detail at a particular location. Examples here are regions of intense mixing or small scale flows around boundaries between large scale flows compared with larger scales within the flows.

1.3 Structure of the paper

The rest of the paper is structured as follows. In Section 2 we review some current work on the visual analysis of large data sets. In Section 3 we introduce the key concept of locality and use it to present and explain the architecture we are in the process of developing. In Section 4 we present the results we have so far, the lessons we have learnt and how we intend to progress towards the achievement of GODIVA's goals. Section 5 presents some preliminary conclusions and suggests pointers for further research.

2.Visualization of large data sets

2.1 Why this is a Grid problem

The question of scale of dataset is a natural Grid question. There are two main reasons for needing to go beyond the model of visualization of data on a single resource. One is to deal with issues in collaborative working (for example as tackled in the Gviz project). Another is to apply large scale computing resources to enable desktop users to be able to visualize very large datasets. Until recently the only way of visualizing such datasets was to go to a specialist visualization resource, such as the Reality Centres pioneered by SGI. If one wished to take the results back to the local workplace, snapshots or videos could be saved in various graphics file format and them transferred by file transfer or on storage media (tapes or CDROMS). If the user then examined these finished products and had further questions that required the ability to rotate, pan or zoom, it mean a return visit to the remote resource

Recent developments, notably SGI OpenGL Vizserver and open source counterparts such as Chromium, enable the output of graphics pipes on such specialist resources to be sent to users desktops where they can be handled as local objects to be viewed interactively. This is a most important step towards a Grid for visual analysis and has been exploited to great effect in the RealityGrid [2] project. In particular in the TeraGyroid [3] collaboration between the UK Level 2 Grid and the US TeraGrid. computations were run on a transatlantic Grid and specialist visualization facilities at Argonne, Manchester, UCL and Phoenix were used in a workshop viewed by sites across the world via the integration with the Access Grid advanced videoconferencing technology. These groundbreaking experiments have won two major awards in the field of data analysis via visual supercomputing. This shows the potential of the Grid approach.

2.2 Large Datasets in GODIVA

In GODIVA, community access to the visual analysis of such data is being tackled in several complementary ways. One is to compress the data by focussing on areas of interest using adaptive mesh technology and remapping the



Figure 1: the GODIVA architecture showing data flow. Solid lines indicate sending of data, dotted lines indicate requests for data. Hexagons represent major services, which may also be internally componentized.

field from a regular to an adapted mesh. This is a form of intelligent data compression, which means that data can be sent to local sites for visualization and analysis. Another is to use large central servers for visualization and use grid services to distribute the output from such servers, by tools such as OpenGL Vizserver from SGI, or the open source Chromium, A third way is explored in this paper, namely to directly use the graphics processing capability of advanced graphics cards and to reproduce the complexity of the data to the limits of the screen resolution of a local device. This inherently involves questions of locality and data-caching since this technique does not extract geometric objects, such as isosurfaces, but relies on the visual impression of the data and the scientists intuition to rapidly search for features of interest which can then be rendered remotely by more conventional means with key geometric objects being extracted and stored. An example in

the interface between cold fresh water and warm salted water, where the geometry can be very complex.

The original GODIVA portal supplied data visualization in well-established formats. These are the traditional 2D and 2D vector plots, 2D contours, isosurfaces and volume visualisation for a couple of scenarios. Nevertheless, it also emerged that the physics of the oceanography models often requires visualization in four or more dimensions and complex visualisation

scenarios are needed to effectively and productively explore the data. As a result what emerged as a key "will be most productive and useful to have" requirement was that far from needing a single step generation of a high quality geometric representation of a huge amount of data, the scientists need the ability to dynamically choreograph complex visualization scenarios using modular services. The services based architecture and advanced rendering on commodity platforms jointly address this requirement.

3. Locality in the architecture

3.1 Service based approach

Current design of Grid architectures is mostly based on the Open Grid Services Architecture (OGSA) approach, which builds on Web services to provide Grid functionality, although the exact best way of expressing this functionality is still a matter for debate. Here we focus on information flow in an OGSA approach by which we mean that the processing of data at various stages is wrapped as a set of interacting services, as shown in Figure 1. At D we have a local Client that directly utilises the graphics card in a local PC or laptop. The clients get data from a Visualization Service at C which can request data from the Data Service B, close to the actual data stored at A.. This adapts to the form of the Client, either by sending geometrical objects to be locally

rendered (the Vizserver approach) or else sends a part of the original dataset, which is then displayed directly by the Client utilising the memory and programmable features of the graphics card itself. The Data service at B (known as GADS [4]) is now established and is being used outside of this visualization framework. This is what we intend with the GODIVA architecture, the Data service is not tightly coupled to the visualization.

We call this first method server-side rendering. It is this approach that has been used very successfully in RealityGrid and which has been enabled by tools to send graphics objects in compressed format over wide area networks. The GODIVA architecture can exploit this method via the Visualization service at C, in essence the Client viewer at D then becomes something like a Vizserver client. However the architecture gives us more than this. Since we now have the possibility that the client can render the large datasets faster than the centralised visualization servers (see next section for details), the dialogue between C and D is much richer. The interactive exploration at client level can proceed rapidly to select features of interest. The location of these features can then be passed back to the visualization server which then extracts the geometry of the feature by large scale processing and can store this for future exploration by science teams or render it to very high fidelity for camera ready copy or video presentations. In a pure exploration phase we can even bypass the visualization service and the client at D can request data directly from the data server at B. Thus the architecture is potentially very flexible.

3.2 The client rendering

Using recent consumer graphics cards we have been developing techniques that utilise their programmable rendering facilities to visualise smooth isosurfaces using a variation on a technique called direct volume rendering. This involves rendering a stack of polygonal slices which sample the 3D dataset stored in a 3D texture. Pixels that match the desired iso-value are rendered opaque, with the rest made transparent. With enough slices, and sophisticated lighting and smoothing, the impression of a smooth 3D surface is produced. Our particular focus is on rendering high-quality smooth surfaces showing how one attribute varies across the isosurface of another by colouring the isosurface depending on the value

of the second attribute. This work has developed to the point where we are able to view isopycnal surfaces of the full 1/12 degree OCCAM dataset interactively on a desktop PC.

Figure 2 shows a snapshot of such visualization. Data fields are held in the graphics card memory enabling the colour mapping of one quantity (salinity) on isosurfaces of another quantity (temperature). This type of interactive visualization is generated at near interactive rates (currently 5-10 frames per second) without the need to invoke the remote visualization service. As such features are identified, we can pass the information back up to the visualization service which can render camera quality isosurfaces on much more expensive resources.

We note here that this method allows visualization in four dimensions with the colour map superimposed on the isosurface providing the fourth dimension.



Figure 2: a view of isosurfaces rendered by the client renderer with salinity mapped as colour on isosurfaces of temperature

3.3 The Visualization Services

RAL as partners in GODIVA have built (and continue to extend) a set of mediating services for data manipulation and visualisation data generators which can be used by other partners, especially the graphics experts at Manchester to deliver real-time visual data using their advanced rendering techniques. These services are represented at C in the dataflow architecture of Figure 1.

The GODIVA application services are Grid enabled modules that compute derived quantities on the fly and pipe such data to visualisation services, which are also implemented as services. Both application and visualisation services can take URI description to third-party data resources and transfer data to the compute resource without having to create a copy at the client. This allows the desktop users to explore larger amounts of data than what can be handled by local hardware. Figure 3 below provides a schematic of how our services communicate with Grid resource layer as well as third-party Web Services (for metadata catalogue query and data access) and how can be interfaced to any desktop client at the client side. The client side API is currently under development.

3.3.1 Isosurface Service

This service generates an isosurface from scalar value file. The isosurface is generated using parallel marching tetrahedron on a Beowulf cluster and the resulting geometry information is stored in a file. Currently the geometry information is written in our specific geometry file format in HDF5. There is a client interface library for reading and writing our geometry file.

3.3.2 Slice service

This service generates slices from a scalar value field. The slice information is written in our specific geometry file format. There is a client interface library for reading and writing this geometry file format.

3.3.3 Animation service

This service generates an MPEG movie from a time series data. The MPEG is generated in parallel. It takes the geometry information from the time series and generates images in parallel and stitches them to form a MPEG animation. Performance measurements and improvements are being made to this service at the time of writing this document.

3.3.4 HardCopy service

This service generates a rendered image in a PDF/EPS/PS format, which can be printed without loss of quality. This service then takes geometry information and writes into a lossless image quality file.



Figure 3: Grid enabled data analysis toolkit

3.3.5 Data manipulation service

Density calculation service calculates sea water density. The input to this service is a file containing sea water potential temperature and sea water salinity variable data. The calculation is done in parallel Beowulf cluster and an URI pointing to the density file is sent back to the user. Similar descriptions apply to the computation of heat flux and surface

In addition, we have also built a spectral transformation service which extracts subsamples of irregular grid data stored in Grib file format and filters and transforms that data to any other required format and coordinate system. The service can take a user provided transform function.

There are several other services, especially those for session management and housekeeping and annotation are currently under development.

4.Current results and lessons learnt

4.1 Frameworks for visualization

We have found that in the differing science driver cases described in Section 1 give rise to a variety of different method of data transmission and visualization, which need to be mapped onto the architecture described in Section 3. Data compression by mapping a uniform grid (used by the Occam model) onto an adaptive mesh by interpolation, is an effective method of data compression because it relates directly to the actual solution and not to some generic method of compressing data. This method, provided by the Imperial College partners in GODIOVA could be provided either as part of the Data Service at B in Figure 1 or else as part of the Visualization service at C. Even better could be to provide such functionality at both, since the local client at D may interact directly with either. If it is a relatively "dumb" client, such as an OpenGL Vizserver client, it must utilise the Visualization service at C, if it is the graphics card client of Section 3.2 then it can interact directly with the Data service at B especially in the interactive exploration phase.

In terms of extracting geometrical objects such as isosurfaces tessellated by polygons, we must use the Visualization service at C, which needs to be able to access a Computational Grid or large parallel machine to do the extraction of the geometry, as described in the services of Section 3.3.

Standard visualization tools, such as vtk and OpenDX, have been used by different science groups. Our experience is that a visualization architecture for large and complex data sets needed by scientific teams must take this diversity into account and build the visualization services as required. We believe that our architectural framework can do this, but more experience will very likely modify it.

4.2 Importance of derived quantities

Some of the key physical quantities such as gradients and flow vectors need to be derived from the model data by computation as the users are exploring the data. Here we have an opportunity in our architecture by exploiting the interplay between the client at D and the Visualization service at C. The service at C can calculate derived quantities and cache them if they are to be reused during the data exploration. Decisions as to where to cache the results can be made according to locality, which can be defined in network access (latency and bandwidth). It may for instance be the case that over a slow network, fields of derived quantities are downloaded and cached near to the client with requests for new parts of the field, or new derived quantities being predicted ahead of time of need according to the users exploration of the total data.

If this can be done we can explore the data in a very flexible and interactive manner particularly with the programmable graphics card client. In Figure 4 we give an example of rendering of derived quantities. The surface normal values are mapped in colour onto the isosurface providing more detail about the behaviour of the fluid than can be observed from the isosurface of the original model quantity (temperature) alone. We are just at the intitial stages of learning how to exploit such functionality and work with the science case drivers will be needed to evaluate case studies to test and further develop the architectural possibilities.



Figure 4: Isosurfaces of temperature with surface normal values (derived quantity) mapped in colour over the isosurface.

5 Conclusions

We have described an architecture for exploiting locality in the exploration of large datasets. The RAVE project [5] from the Welsh e-Science Centre is also looking at issues of locality and client capability. Our approach is characterized by its data-centricity, at every point in the architecture the key question is how is the data to be used. Visualization is part of this usage but the GADS server can be used without any need for visualization. This design allows us to visualize at different points in the data-flow and in different ways. Thus we can exploit the new capabilities of programmable graphics cards as they are now becoming available. We anticipate that in the near future, pragmatic rather than ideal considerations will apply to choice of visualization tools and methods. In part this is due to issues of locality due to the highly non-homogeneous nature of networking bandwidth and latency but also because we found that different scientific

groups use different tools, which suit their problems and methods of analysis and will not abandon this for abstract considerations for a purer architecture. In essence we provide for multiple ways of visualizing large datasets, exploiting specialist graphics supercomputers, commodity clusters and modern PC graphics cards. We consider this multiplicity of methods to be an essential strength of our approach.

Acknowledgments

This GODIVA project was funded by the National Environment Research Council, as part of the NERC e-Science programme.

References

1. www.godiva-grid.org 2. www.realitygrid.org 3. The TeraGyroid Experiment S.M. Pickles, R.J. Blake, B.M. Boghosian, J.M. Brooke, J. Chin, P.E.L. Clarke, P.V. Coveney, N. Gonzalez-Segredo, R. Haines, J. Harting, M. Harvey, M.A.S. Jones, M. Mc Keown, R.L. Pinning, A.R. Porter, K. Roy, M. Riding, GGF10 Applications RG Workshop www.zib.de/ggf/apps/meetings/ggf10.html 4. GADS, Grid Access Data Service information and references at http://www.resc.rdg.ac.uk/twiki/bin/view/Resc/ **GridAccessDataService** 5. RAVE - Resource-Aware Visualization Environment, home page www.wesc.ac.uk/projects/rave/