Kitware Progress report

ParaViewWeb - August 2010

Sebastien Jourdain

Introduction

During the development phase of ParaViewWeb several documents have been written and published. This document provide the list of todays available resources.

Resources

- Documentation
  - Wiki: http://www.paraview.org/Wiki/ParaViewWeb
- Source repository
  - Git repository: http://paraview.org/gitweb?p=ParaViewWeb.git
- Demonstration server
  - http://paraviewweb.kitware.com
- Plublications
  - IADIS Web3D 2010: ParaViewWeb, a web framework for 3D visualization and data processing (provided in annexe)
  - Kitware Source July 2010: http://www.kitware.com/products/html/ParaViewWebBringingVisualizationToTheWeb.html

Contacts

- Sebatien Jourdain - sebastien.jourdain@kitware.com
- Utkarsh Ayachit - utkarsh.ayachit@kitware.com
- Berk Geveci - berk.geveci@kitware.com

Annexe

ParaViewWeb, a web framework for 3D visualization and data processing

Sebastien Jourdain, Utkarsh Ayachit, Berk Geveci
Kitware
28 Corporate Drive
Clifton Park NY, 12065 USA

ABSTRACT

Since the early days of the Internet, Web technologies are at the forefront of innovation with multitude of traditional desktop applications migrating to the Web and several new ones are being invented. In this paper we present a 3D visualization framework ParaViewWeb, that enables interactive large data processing and visualization over the web. To enable large data processing, ParaViewWeb uses ParaView, an open source, parallel data visualization framework to generate the visualizations on the server-side while delivering images to the client at interactive speeds. ParaViewWeb makes it easier to develop customized applications for the web that cater to a wide variety of scientist and domain experts that can use such a web-based system to analyze their datasets without leaving the familiar and omnipresent environment of a web browser. In this paper, we present the ParaViewWeb framework and its features and discuss some of the application fields that can benefit from it.

KEYWORDS

Web 3D, Scientific visualization, Data processing, Remote rendering, ParaView

1. INTRODUCTION

In recent years several organizations across the globe have been developing infrastructures that include high performance computing resources. These are often distributed across several sites. This enhanced compute power has made it possible to run large simulations, producing large datasets. The data sizes have made distance-visualization a necessity. It is no longer possible to copy datasets to your desktop for analysis. ParaView [Henderson, 2007] is an open-source, multi-platform data analysis and visualization application. ParaView was developed to analyze extremely large datasets using distributed memory computing resources. It can be run on supercomputers to analyze datasets of terascale as well as on laptops for smaller data. In this short paper, we present an exciting new approach for visualizing data over the web. We present a ParaView-based web-visualization framework that allows developers to build custom web applications for visualization. These web application can leverage ParaView's parallel data processing and rendering capabilities on the server side, while presenting easy-to-use, highly customized web-pages to the users to control and interact with their visualizations without leaving the familiar and omnipresent environment of a web-browser.

2. PARAVIEWWEB: THE WEB VISUALIZATION FRAMEWORK

The Web has been evolving rapidly since its inception. The Web has proliferated the modern way of life so much that people are now considering migrating traditional desktop applications to web e.g. document editing [Google Docs, 2010], finance management [Mint Software, 2007] and tax filing [TurboTax, 1997]. Part of the appeal of the Web is due to its ease of access. One simply types the URL in the web browser and can access the application, no matter where he is, or what device he is on. The same is true for visualization over the web. There are several frameworks that have been developed that focus on visualization eg. Protovis [Bostock, 2009], Many Eyes [Many Eyes, 2004]. However most of these are dealing with 2D visualization and use client-side rendering to generate the visualizations, which works great for smaller data sizes but is not feasible with really large datasets.

2.1 Features

To address the need for visualizing large datasets in 3D, we have developed a framework based on ParaView. We can leverage all of ParaView's visualization and data processing features such as parallel processing and rendering, volume rendering, ray-casting etc.

Furthermore, to serve interactive 3D content in real time to the client, we use server-side rendering. This approach makes it possible to render large geometries without putting any requirement on the client devices and enables the use of mobile devices such as iPhone and iPad. To manage the 3D content on the client side, the framework provides a set of components for viewing it in an interactive manner. Those components are available in Java [Sun, 1995], Flash [Adobe Flex, 2006] and JavaScript [Ecma, 1999] and can be used to rotate, zoom or pan the 3D data.

As a framework ParaViewWeb uses well proven technologies and standards such as Java Servlet, Java Server Pages (JSP), Java Messaging Service (JMS) [JSR-914, 2002], Java Persistence API (JPA) [JSR-220, 2006] on the server side to do most of the binding between the browser and the ParaView framework. We use JavaScript and JSON-RPC [JSON-RPC, 2006] for communication protocol on the client side. Being a web based application, all communication follows either HTTP or HTTPS protocols which makes it easier to deal with firewall and proxy issues.

In order to ease the integration of ParaViewWeb inside web applications, the framework provides a collection of components rather than a web infrastructure. These components can be directly used or easily integrated in an existing web portal to leverage any preexisting authentication. ParaViewWeb enables developers to create custom web application suitable to their clients with typical workflow for their specific domain.

2.2 Components

ParaViewWeb is a collection of reusable components. Those component range from server side to client side. On the server side, the visualization server (PWServer) is a ParaView-based engine that does the actual visualization either by itself or by connecting to a remote ParaView server running over a cluster with MPI. Then, the web service component named PWService manages communication between remote visualization servers (PWServer) and clients. It includes also administration web pages allowing to monitor running visualization session as well as browsing logs and informations of the previous ones. On the client side, a JavaScript library is provided for creating remote visualizations and managing them as well as several visualization components allowing users to look at 3D content in the browser interactively.

Using these components, developers can build websites or web portals with visualization and data processing capabilities. These components can be easily integrated into Rich Internet Applications (RIAs) developed using popular web designing infrastructures including qooxdoo [Qooxdoo, 2005], Dojo [Dojo, 2009], Google Web Toolkit [GWT, 2007], jQuery [jQuery, 2009], Flex [Adobe Flex, 2006], Java [Sun, 1995] and other. Figure 1 gives a schematic of the various components involved. Our implementation requires any supporting Java-based Web Application server which includes Apache Tomcat, an open source, freely available implementation.

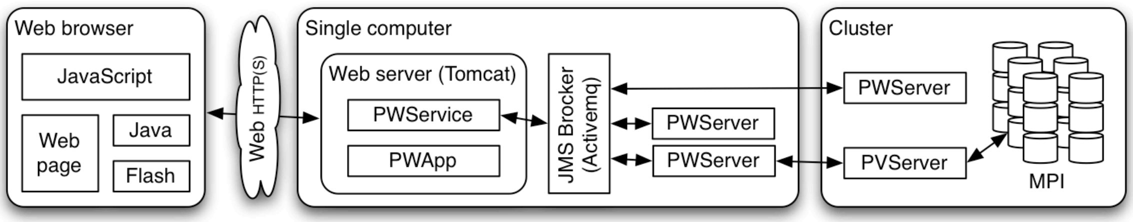


Figure 1. Schematic of the ParaViewWeb Visualization System

2.2.1 The web services

PWService is a web-application that is deployed on the web server accessible using a particular url that is determined at configuration time. Using JSON-RPC, which is a simple JSON [JSON, 1999] (JavaScript Object Notation) based protocol for remote procedure calls, clients can make calls on PWService to start new instances of PWServer, monitor running PWServer instances, and even send JSON messages to PWServer. The messages sent to PWServer range from those constructing the visualization pipeline, to those that communicate the mouse interactions.

2.2.2 The processing server

PWServer is a ParaView-based process that can be thought as a headless ParaView application that can respond to the JSON messages sent from the PWService. The communication between PWService and PWServer is done using Java Message Service (JMS) messages. The protocol is still JSON-RPC based even though the communication is over JMS. PWServer is the visualization engine. PWServer can be setup to use MPI to perform parallel data processing and rendering over a cluster or as a single process. It does all the data processing and rendering. For each separate visualization session, we create a separate PWServer process. PWService handles the management of these instances of PWServer either if they are local or working on a remote cluster and involving MPI internal communication.

2.2.3 The JavaScript library

To make it easier for client applications running in the web-browser to use PWService and communicate with PWServer, we have developed a JavaScript library. As highlighted before, this library facilitates configuration of the visualization pipeline. Figure 2 shows the JavaScript code of a static web page that embed 3D interactive visualization with a very simple visualization pipeline.

Data Processing

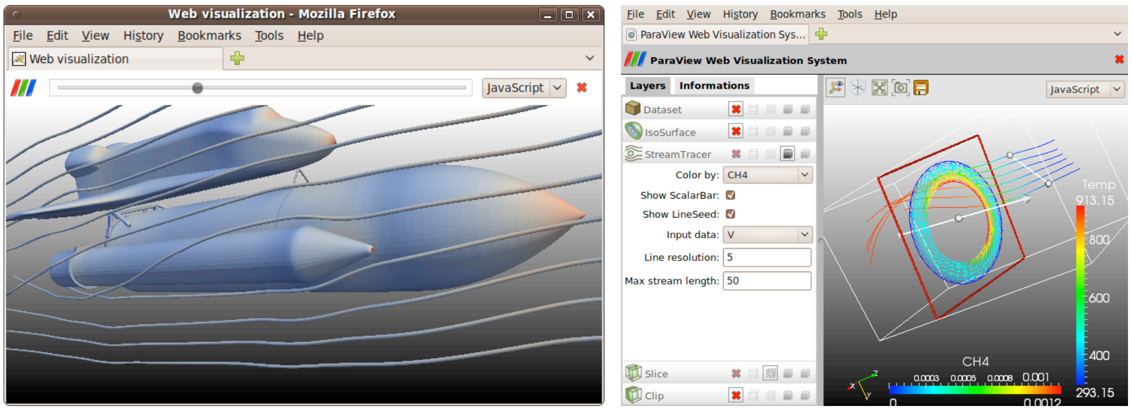
Rendering

```
var paraview = new Paraview("/PWService");                          // Create a paraview object
paraview.createSession("name", "comment");                          // Create a remote session
var view = paraview.CreateIfNeededRenderView();                     // Create a server 3D view
paraview.OpenDataFile({filename: 'bunny.ply'});                     // Load some 3D data
paraview.Show();                                                    // Show the data in the view
var renderer = new JavaScriptRenderer("myId", "/PWService"); // Create renderer
renderer.init(paraview.sessionId, view.__selfid__);                 // Initialize the renderer
renderer.setSize('200', '200');                                     // Set the renderer size
renderer.bindToElementId("container");                              // Insert renderer into HTML
renderer.start();                                                   // Start update loop
```

Figure 2. Sample JavaScript code

3. APPLICATIONS

Web visualization has several applications in real world. As mentioned earlier, it facilitates development of web portals for job submission and then result analysis using ParaViewWeb for supercomputing sites. These visualization applications can be customized to the types of the datasets that are being visualized, making them easier to use for the domain experts.



Web visualization - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Web visualization
JavaScript
ParaView Web Visualization Sys...
ParaView Web Visualization System
Layers Informations
Temp
913.15
800
600
400
293.15
CH4
0.0003 0.0005 0.0008 0.001
0 0.0012

Dataset
IsoSurface
StreamTracer
Color by: CH4
Show ScalarBar: ✓
Show LineSeed: ✓
Input data: V
Line resolution: 5
Max stream length: 50
Slice
Clip

Figure 3. Samples web applications using the ParaViewWeb framework

As a demonstration prototype, we developed a web application for post-processing where users can either pick a dataset on the server or upload their own. The supported data type are the one supported by ParaView which range from simple geometry one such as VRML and PLY to real simulation dataset using VTK, Exodus, Xdmf or other formats. Once the data is loaded on the server, it can be processed by a set of classical scientific visualization filter such as clipping plane, slicer, iso-contours and depending on the data a stream tracer which is used to see the flow based on the integration of a vector field. Moreover, interactive 3D objects can be used to move or rotate the plan used by the slicer and the clipping plane filter as well as the line seed where streaming lines starts their computation.

Web visualization also presents exciting new possibilities for modern day classrooms. Students no longer have to look at static images in a textbook. They can be study the fluid flow properties, for example, by interacting with a visualization online. Similarly with online journals, instead of merely present static images in papers, we can develop online journals and publication establishments that enable the reader to simply click on the image to start interacting with the real dataset.

Last but not least, ParaView Web can be used as a renderer engine for online 3D multimedia database, and thanks to ParaView Web user will be able to browse data and visualize them directly inside their browser. In fact some first integration steps have been done with MIDAS [Kitware, 2009] a multimedia server for storing massive collections of scientific data where user can now see their data in live 3D.

4. CONCLUSIONS

In this paper, we have presented a framework for large data visualization over the Web, ParaViewWeb. ParaViewWeb is a parallel visualization framework that is comprised of components that make it easier to develop web sites that enable interactive analysis for large datasets. It is based on ParaView, a popular open source visualization tool which uses distributed data processing and rendering. ParaViewWeb uses server-side rendering to avoid complications with delivering really large geometries to the client as well as complications with rendering 3D geometries in a web-browser in a cross-browser and cross-platform way. To provide a good responsive 3D visualization system, we had to optimize the image processing and delivery to reduce latency. Like ParaView, ParaViewWeb also will be released under a BSD license, thus anyone can extend and customize it to their specific needs.

REFERENCES

Adobe Flex, 2006. Adobe. http://www.adobe.com/products/flex
Bostock M., Heer J., 2009. Protivis: A graphical toolkit for visualization. http://vis.stanford.edu/protovis
Dojo, 2009. Dojo foundation. http://www.dojotoolkit.org/
Ecma, 1999. Standard ECMA-262, ECMAScript Language Specification, 3rd edition
Google Docs, 2010. http://docs.google.com
GWT, 2007. Google Web Toolkit. http://code.google.com/webtoolkit
Henderson A. et al, 2007. ParaView Guide. A Parallel Visualization Application. Kitware Inc.
jQuery, 2009. Software Freedom Conservancy. http://jquery.com
JSON, 1999. http://json.org
JSON-RPC, 2006. http://json-rpc.org
JSR-914, 2002. Java Message Service Specification. http://jcp.org/en/jsr/summary?id=914
JSR-220, 2006. Enterprise JavaBeans 3.0. http://jcp.org/en/jsr/summary?id=220
Many Eyes, 2004. http://manyeyes.alphaworks.ibm.com/manyeyes/
Midas, 2009. Kitware, Inc. http://www.kitware.com/products/midas.html
Mint Software, 2007. Mint, Inc. http://www.mint.com
Qooxdoo, 2005. 1&1 Internet AG. http://qooxdoo.org
Sun, 1995. Java-Oracle. http://java.sun.com
TurboTax, 1997. Intuit, Inc. http://www.turbotax.com