

Visualisation in the Google Cloud

by Kieran Barker,¹ School of Computing, Faculty of Engineering

ABSTRACT

Providing software as a service is an emerging trend in the computing world. This paper explores the possibility of creating a visualisation application in the cloud. By using prototypes to investigate potential functionality a final application was then produced. When this application was tested for processing performance better results were recorded in the cloud than on a local PC. However, given the drawbacks to cloud development it was concluded that producing visualisation as a service in the Google cloud is not currently a feasible objective.

Keywords: Cloud computing; Google; Distributed computing; Visualisation; VRML

¹ email address for correspondence: kbarker04@googlemail.com

Visualisation in the Google Cloud

INTRODUCTION

Cloud computing is a new paradigm in the field which involves using a service provider's infrastructure to run an application and store data, rather than doing the same on your local PC. It is a new form of distributed computing which provides 'software as a service' as well as offering computational resources in an oblique way to a user. The location of the data and the location of the computer running the application is hidden from the user.

Visualisation of datasets is used to enable easier understanding of large amounts of information. It is used in fields such as the medical industry for purposes such as the visualisation of scans that are performed using Medical Resonance Imaging (MRI) or Computed Tomography (CT). From the datasets produced by these scans material of differing types, such as bone or skin, can be extracted for visualisation.

This project researches into combining these two areas to produce a visualisation application that can be run using the Google App Engine (GAE). This will entail all data storage and processing to be performed 'in the cloud' and will explore the possibility of using the cloud to provide 'visualisation as a service'. Having produced an application it was possible to assess the merits of this approach over having the application stored and run on a desktop PC.

PROJECT BACKGROUND

Research was performed to examine the general concepts and technologies used in traditional distributed computing as well as some of the milestones of cloud computing. Research was also conducted on the technologies to be used in the visualisation app.

Distributed Computing

A distributed system can be defined as a collection of independent computers that are connected through a network but that appear to the user as if they are one connected system. The aims of distributed computing are described by Tannenbaum and Van Steen (2007) as:

- Accessible resources
- Distributional transparency
- Openness
- Scalability

In a distributed system the components at networked computers communicate by passing messages to enable the sharing of resources. The transparency of this process refers to hiding from the user the fact that these resources are not local. Having openness in the system entails an agreed upon set of standards and protocols that all components use in communicating for example, when using Remote Method Invocation (RMI). A distributed system is considered scalable if it is possible to increase the number of users or resources it possesses and it still remains effective.

Cloud computing

The emergence of the commercialised cloud was pioneered by Amazon in October 2007. The Amazon Elastic Cloud Compute (EC2) offers customers the ability to create a virtual machine running on their infrastructure that is presented to the end user as if it was a dedicated server as noted by Weiss (2007). Customers can

choose customisable configurations and pricing is based only on what is used and is calculated on a per virtual machine instance basis. The real benefit of EC2 is that machine instances can be created and destroyed on demand. For example, if there is a sudden increase in visitors additional machine instances can be launched and as traffic slows down these can be terminated. In this way the application can be scaled dynamically.

While Google's most famous and popular use is as a search engine, Connar (2008) notes that their most recent developments are in putting applications traditionally installed on your home PC on-line. A number of Apps which allow the user to conduct tasks such as word processing or working on spreadsheets are now available directly through your web browser in Google Apps. In April 2008 Google announced the launch of its Google App Engine which allows applications to be developed and deployed on their infrastructure. Unlike Amazon, Google allows developers to use their cloud for free; however a restriction was applied at the time of the project that all apps had to be written in Python.

Virtual Reality Modelling Language (VRML)

Hartman and Wernecke (1998) state that VRML is used to describe and build 3D objects and scenes and can be used to create interactive simulations that include animation and motion physics in real time. Applications using VRML range from medical imaging and molecular modelling to games and advertising. To enable easy viewing of these 3D scenes over the World Wide Web, all that is required is a browser with a plug-in installed that supports viewing of VRML scenes. This allows the user to explore the scene interactively in their browser using a tool bar to manipulate the model.

DEVELOPING USING THE GOOGLE APP ENGINE

The development methodology used during the project was an agile methodology that incorporated a prototyping approach. Agile development is the name given to a collection of iterative development methods which focus on short development cycles. A prototype is a partially complete application that is built quickly to evaluate or test some aspect of the final piece of software. Three 'throw away' prototypes were produced, each being evaluated and then discarded.

To begin development using the GAE a Software Development Kit (SDK) is downloaded from the Google site. This includes a set of libraries, templates and programs to enable both running of a development environment and uploading of code to the Google site. Each prototype app included at least three files; a Python script, a HTML template and a YAML configuration file. Once the app has been created in the development environment the files are uploaded to the Google cloud using a file included in the SDK.

The first prototype allowed a user to calculate the addition of two values, X and Y, and have the result returned to them. It also allowed the user to login to the app. Having users sign into an app may be useful if attempting to provide the software as a service, in this way only registered users would be allowed access.

The aim of the second prototype was to explore the use of 2D graphical elements in the application by embedding an SVG image into the HTML code. Problems initially occurred when the SVG image was not displayed on the web page, following a post to the Google forums it was discovered that the YAML configuration file needs to declare a static directory (e.g. called media) to store images in.

The third prototype explored using 3D graphics in the form of VRML. This prototype allows a user to enter a value that is set as the dimension of a cube, which is then created in VRML and displayed to the user. Although it is possible to upload files with an app, the Google file system only allows read only access. This means that only the VRML files uploaded with the app can be used, it is not possible to create a VRML file and to then save it in the Google file system. To overcome these problems the VRML can be saved in Google's data-store in text format and, using a content type header, then displayed in the browser as VRML.

THE VISUALISATION APP

The development of the first three prototypes provided the level of understanding required to use the Google SDK to begin the development of an evolutionary prototype that would provide the functionality required to produce a visualisation from a given dataset.

The first step to producing the visualisation app was to code a set of functions that could be used to create a VRML model of a given dataset (uploaded with the app). To do this it was necessary to understand the process involved in creating a model from a 3D volumetric dataset. Having researched into the marching cubes algorithm by Lorensen and Cline (1984) a simplified version was created which, having been given a threshold to determine if a vertex is inside or outside the model, performs the following:

- Select a cell to be processed;
- Calculate the inside / outside state of each vertex in the cell to determine if the cell is included;
- Create an index that stores the x, y, z values for any cell that has an inside vertex.

Once this process is complete a VRML model is created which draws each cell in the index as a box geometry node that is translated according to the x, y, z value of the cell. The later steps of marching cubes algorithm, which look up the topological state of the cell in a case table and then calculate the contour location for each edge, are ignored by this simplified version.

The app was then improved further by allowing the user to upload their own dataset for use in the creation of the VRML model. Also, the graphical output was enhanced by implementing a more complex algorithm to produce the model. Instead of simply drawing each cell as a cube the app now stores each vertex of a cell that is inside the model and only draws the sides that can be made from these vertices. (See Appendix A for images relating to this app).

EVALUATION

The app's performance running in the cloud was measured versus the performance achieved running on a home PC. The results from this testing process (see Appendix B) showed that deploying the app in the cloud provided a performance boost. Based on experience gained during the project, conclusions on the advantages and disadvantages of developing in the cloud were drawn as well as the suitability of providing visualisation as a service.

Advantages of using the cloud

The advantages of developing an app in the cloud are:

- It is free to host the app in Google's cloud;
- There is no requirement to purchase and maintain your own server;
- Software updates to your app are available to users as soon as deployed.

The advantages for a user of the app are:

- It is available in any geographic region that has an internet connection;
- Processing times were tested faster than on a standalone PC/server;
- The results of a visualisation can easily be displayed to colleagues simply by giving them access to the app.

Disadvantages of using the cloud

For the developer these include:

- Limitations on the size of data objects saved in the data-store;
- Inability to directly work on or save files in the file system;
- Time-outs on Google's servers when performing processing requests;
- Lack of error logging being returned when errors occur on Google's servers;
- No valuable feedback for the users of your application during errors;
- Lack of multi-language support (at time of development).

These disadvantages reduce the effectiveness of developing a visualisation app using the GAE. The limitations with data storage are particularly relevant when considering visualisation as a service. Without access to the file system it would prove difficult to allow users to store any created visualisation. Another major concern is that Google's servers time-out with large requests. Obviously this would not be acceptable when deploying the app as a service.

These drawbacks could be overcome when using Google's pay service that was launched in May of 2009, which will also contain support for Java. However, by spending money on running the app a lot of the advantages listed are negated. For example, the fact that it is free and also that a developer does not need their own server. Depending on the expected number of users for the app it may in fact be cheaper in the long run to purchase a server than to use Google's infrastructure.

CONCLUSIONS ON CLOUD DEVELOPMENT

The future use of the cloud may result in a situation where users only own a thin client, a lightweight laptop with a small hard drive and a browsing application. All of a user's data and the applications they use could be located in the cloud, entertaining the idea of software as a service that you connect to and use as you require.

However, in the case of using the GAE to provide visualisation as a service it is the opinion of the author that this is not currently a useful objective, given the drawbacks discussed. If the developer has money available to deploy the app then this situation may alter, given Google's pay service. In this case pros and cons would have to be considered in terms of the cost of deploying on Google's infrastructure versus purchasing a standalone server.

REFERENCES

Connar, N. (2008). *Google Apps: The missing manual*. (O'Reilly Media Inc.) Introduction xiii – xxviii.

Hartman, J. Wernecke, J. (1998). *The VRML 2.0 Handbook*. 5th ed. Addison-Wesley Books. USA.

Lorensen, W. Cline, H. (1987). *Marching Cubes: A high resolution 3D surface construction algorithm*. In: *Computer Graphics*, Vol. 21, No. 4

Tanenbaum A.S, Van Steen M. (2007). *Distributed Systems Principles and Paradigms*. 2nd ed. Pearson Prentice Hall. Chapter 1 pg 1-16,

Weiss, A. (2007). *Computing in the Clouds*. In: *netWorker*, Vol. 11, No. 4

APPENDIX 1 - Figures

Presented below are screen shots from the visualisation app:

Figure 1. The web page as presented to the user

Figure 2. The visualisation produced when a dataset is processed

Figure 3. The visualisation using the enhanced algorithm

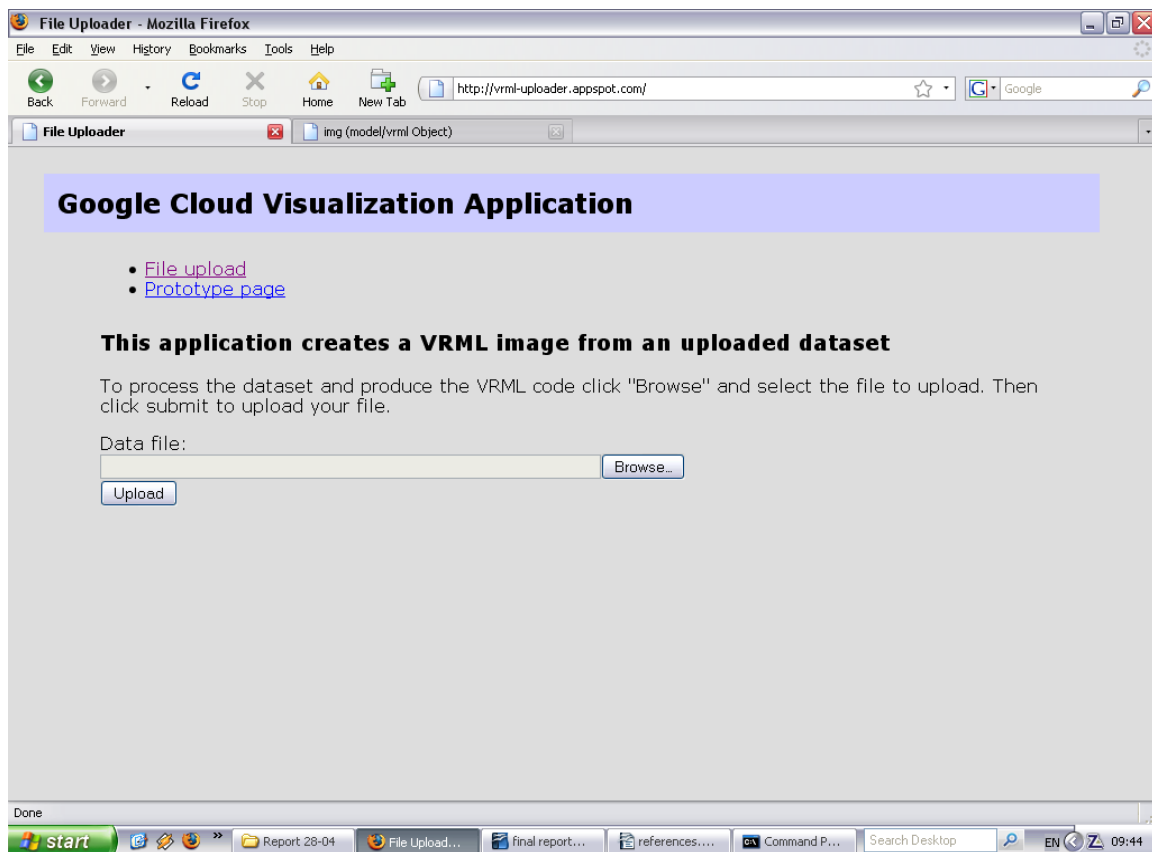


Figure 1. The web page as presented to the user

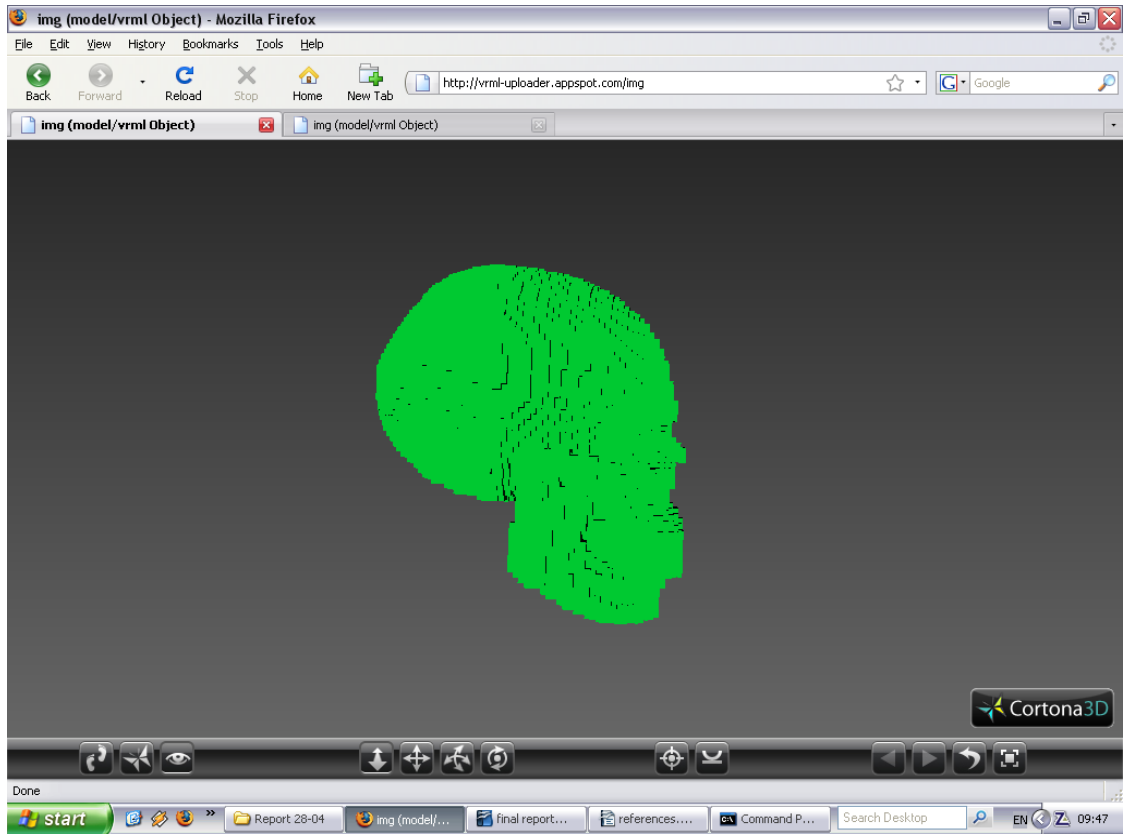


Figure 2. The visualisation produced when a dataset is processed

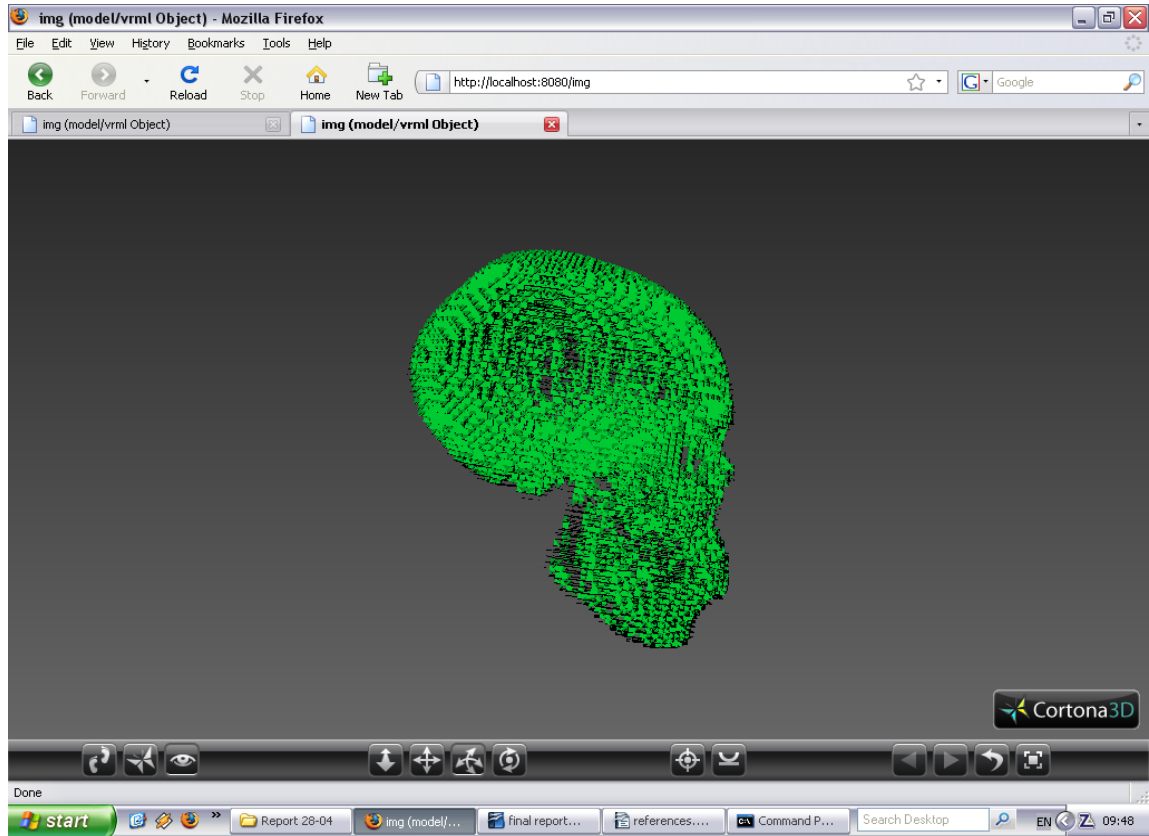


Figure 3. The visualisation using the enhanced algorithm

APPENDIX 2 - Tables

The PC used for testing had a 2.2Ghz AMD Athlon-64 processor with 1GB of Ram.

The tests were run using data sets of varying sizes.

Dataset size	Processing Times for the Visualisation App	
	Development Environment	Google Cloud
30x30x30	3.85s	1.9s
40x40x40	6.7s	2.83s
50x50x50	10.83s	3.83s
64x64x68	14.8s	6.5s

Dataset size	Processing Times for the App Version 2	
	Development Environment	Google Cloud
30x30x30	4.2s	2.3s
40x40x40	13.7s	10.1s
50x50x50	19.76s	Error ²
64x64x68	29.3s	Error

² These errors occurred when processing large requests on the Google server, resulting in the response timing out.