PROVENANCE Contract Number:
Enabling and Supporting Provenance in Grids for Complex Problems 511085

Title:	Gravitational Wave Analysis
Authors:	Omer F. Rana
Editor:	
Reviewers:	
Type:	
Version:	1.0
Date:	September 2006
Status:	Final
Class:	Public

Summary

This document provides a case study of Provenance use in the Gravitational Waves community. The document is based on work undertaken in the GridOneD project.

The *PROVENANCE* Project receives research funding from the European Community's Sixth Framework Programme

Members of the PROVENANCE Consortium

IBM United Kingdom Limited	United Kingdom
University of Southampton	United Kingdom
University of Wales, Cardiff	United Kingdom
Deutsches Zentrum für Luft- und Raumfahrt e.V.	Germany
Universitat Politecnica de Catalunya	Spain
Magyar Tudomanyos Akademia Szamitastechnikai es Automatizalasi Kutato Intezet	Hungary

The *PROVENANCE* Project receives research funding from the European Community's Sixth Framework Programme

Contents

The *PROVENANCE* Project receives research funding from the European Community's Sixth Framework Programme

Provenance Use Case:
Gravitational Wave Analysis

Omer Rana
o.f.rana@cs.cardiff.ac.uk
Cardiff School of Computer Science/Welsh eScience Centre

1 Introduction

This document describes a particular scenario developed as part of the GridOneD project and provenance questions apparent within this scenario. The GridOneD project is, in particular, focused on the analysis of Gravitational wave data from laser interferometers using the Triana workflow engine. This use case outlines provenance questions that relate to such analysis.

2 Problem Description

Galaxy Formation (GF): Galaxy and star formation using smoothed particle hydrodynamics generates large data files containing snapshots of an evolving system stored in 16 dimensions. Typically, a simplistic simulation would consist of around a million particles and may have raw data frame sizes of 60 Mbytes, with an overall data set size of the order of 6 GBytes. The dimensions describe particle positions, velocities, and masses, type of particles, and a smoothed particle hydrodynamic radius of influence. After calculation, each snapshot is entirely independent of the others allowing distribution over the Grid for independent data processing and graphic generation.
Inspiral Search (IS): Einstein's theory of General Relativity predicts the existence of "gravitational waves". We have indirect evidence for the existence of gravitational waves but no direct observation has so far been made. Such waves are generated by compact binary stars orbiting each other – until their collision. Some Laser interferometric detectors such as GEO600, LIGO and VIRGO should be able to detect the waves from the last few minutes before collision. A gravitational wave passing through the interferometer causes displacements of the mirrors and a shift in the interference pattern. The amplitude of the displacement will be extremely small. To search for an inspiral signal, the detector output is examined for signals of particular shape. This shape is called a template and it is constructed using theoretical knowledge about relativistic binary systems. It is determined by its family of parameters, the most important of which are the masses of the compact objects.

2.1 Visualisation and Steering

A user of the GF application would like to view the chronological changes in the Galaxy as an animation, and makes use of the data management components provided in Triana. Two separate user interfaces are provided to allow the

The *PROVENANCE* Project receives research funding from the European Community's Sixth Framework Programme

remote steering of the Galaxy Formation test case. One is the generic *SequenceBuffer* tool and the other is the user interface from the *ViewPointProjection* unit. Every Triana unit that implements a user interface can be viewed remotely. In this case by setting the new X, Y and Z parameters in ViewPointProjection, the user can simultaneously update these values on all nodes. Similarly, for the SequenceBuffer, the animation can be remotely started simultaneously.

The *ViewPointProjection* unit's user interface on the user's local machine is used to steer the entire process. The user can select the precise viewpoint using the given coordinates. If the user wants a different view of the data he changes these coordinates and presses the start button. Messages are then sent to all the distributed servers so that the new data slice through each time frame can be calculated and returned. Each distributed Triana service returns it's processed data in order and the frames are animated. The *ImageView* visualization unit then displays the resultant animation as a sequence of GIF files. The result is that the user can visualize the Galaxy formation in a fraction of the time compared to the time it would take if the simulation was performed on a single machine.

2.2 Data Management and Computation

The GF application consists of three main data management activities:

- File Parsing: The data files are parsed according to their format and the data loaded into data structures. Each data segment represents a distinct time frame or snap shot within the animation. The is achieved by the *DataFrameReader* unit.
- Data Set Projection: The 3D data sets are projected down onto a 2D plane from a viewpoint using a calculation – performed by the *ViewPointProjection* unit.
- Visualization: The 2D frames are run to form the animation. This is displayed using an existing Triana component, called *ImageView*.

The *ViewPointProjection* and *DataFrameReader* were built using Triana's tool builder to generate code skeletons. Code from the original application was then inserted into these skeletons. It is possible to extend the process and buffer the data for future calculations (using another generic Triana unit, called *SequenceBuffer*), based on a new view point using the Sequence unit. This simple data player component that saves us having to run the *DataFrameReader* component multiple times, which depending on data size can be expensive.

The loaded data is divided into frames, distributed amongst the various Triana services and processed to calculate whichever analysis is required, for instance this might be a simple viewing of the morphology of the Galaxy or it might be a much more numerically intensive analysis of calculating the column density using smoothed particle hydrodynamics. Distribution of the data amongst the Triana servers works as follows: the local client locates a data file,

The *PROVENANCE* Project receives research funding from the European Community's Sixth Framework Programme

reads the data from each frame and then distributes the data to a Triana server for processing.

For the IS application, the search algorithm works by correlating the data with the templates, a technique known as template matching or matched filtering. The correlation is achieved by using the fast correlation algorithm by taking the fourier transform of both the template and the data, multiplying them together, then taking the inverse Fourier transform of the result. There are typically tens of thousands of templates (in each bank), each containing different parameters defined at a certain granularity within the search space. The key factor is to be certain that the search space is fine-grained enough to catch the incoming waves.

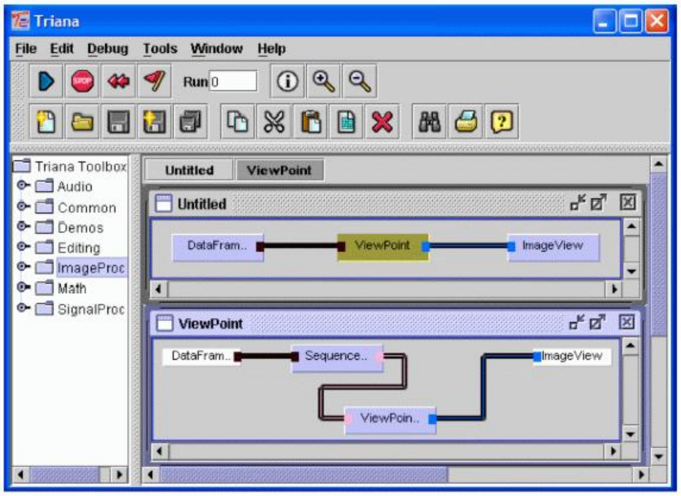

Figure 1: The Galaxy Formation Code implemented as a set of Triana units. The upper workspace containing three components is an unbuffered player: it reads data from the file every time the player is activated. The lower workspace stores the data frames in a sequence buffer for reuse.

Existing C code is currently used to calculate the bank parameters and write them to a text file. This is read, the templates are generated and the correlations are performed. For a modest search, we would need to have a computing resource capable of speeds in the range of 10 GigaFlops to keep up in real time with an on-line search. For example, the gravitational wave signal is sampled at 16kHz and sampled at 24-bit (stored in 4 bytes). However, the meaningful frequency range is up to 1 KHz and therefore a sampled representation of this contains 2,000 samples per second. The real-time data set is divided into chunks of 15 minutes in duration (i.e. 900 seconds) which results in a 7.2MB of data (4 x 900 x 2000) being processed at a time. This data is transmitted to a node and it is processed. The node initialises i.e. generates its templates (a trivial computational step) and then it performs fast correlation on the data set with each template in a library of around 10,000 templates. This process takes about 5 hours on a 2 GHz PC running a C program. Therefore, 20 PC's with fast communication abilities would need to be employed full-time to keep up with this data. To perform a search in real time we must filter each segment of data through the bank before the next segment comes in. This is a formidable task and represents an excellent test of Grid computing infrastructure.

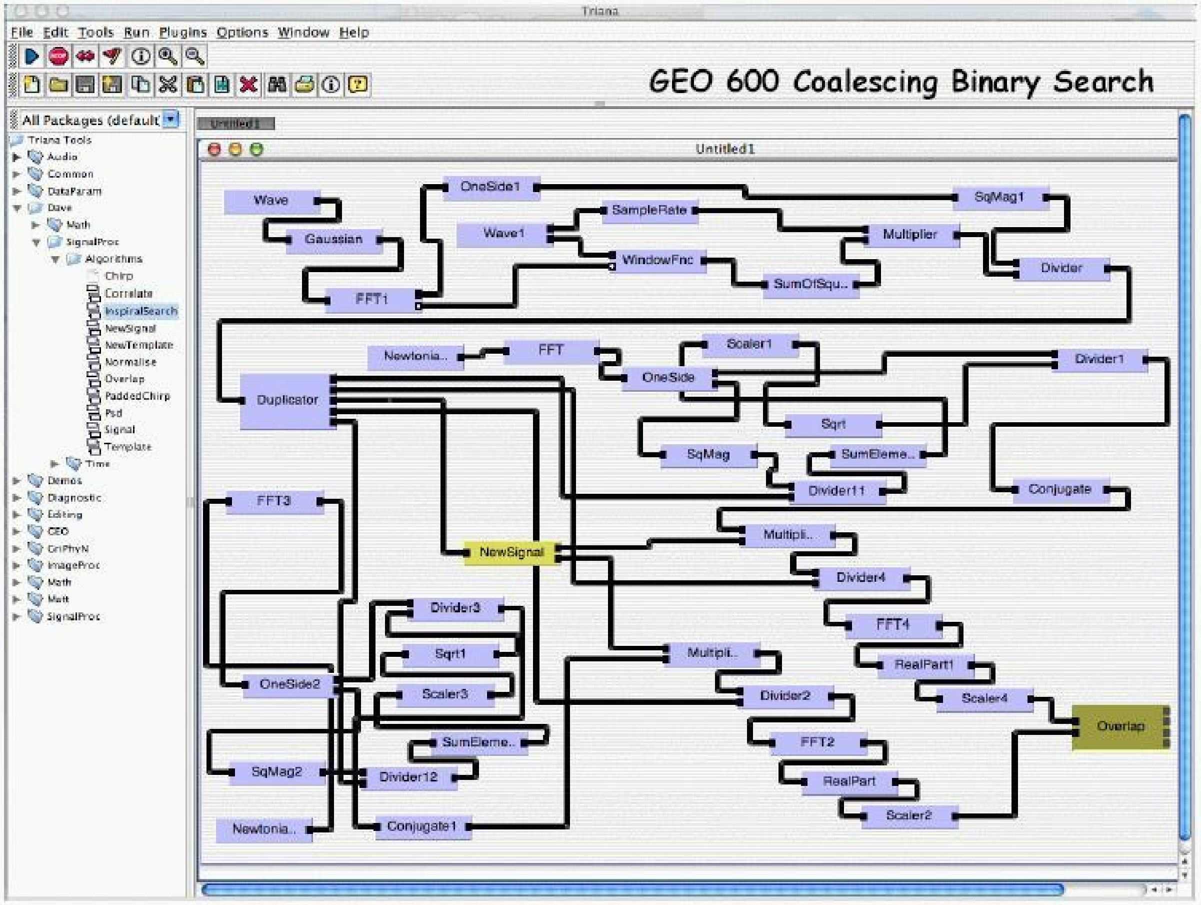

Figure 2: The full Coalescing Binary search showing most of the units displayed on one work-space.

3 Provenance Questions

The particular provenance issues that arise from these applications include:

- Steering: In this case, a user undertakes an interactive session with either a selection of pre-generated execution traces, or via direct interaction with

The *PROVENANCE* Project receives research funding from the European Community's Sixth Framework Programme

a simulation code. By changing the X,Y, and Z viewpoint parameters, it is possible to change the type of analysis being undertaken (in the GF case below). Provenance issues:

  - Dealing with issues of real-time/interactive sessions (the provenance data to be recorded here must also somehow account for time of capture of the data). The provenance question being addressed is identifying the time at which a particular data recording was made, and identifying and measuring the duration between two recordings.
  - Data format analysis – which data types can be passed to which type of service. The provenance data can be used to identify whether two services connected in a workflow have similar/identical data types.
  - Data reduction issues – if a 3D viewpoint is reduced to a 2D viewpoint for input into another component, is this still relevant and a "true" representation. The provenance data in this scenario can be used to evaluate the process involved in converting the 3D viewpoint to the 2D viewpoint, and whether this process is likely to generate inaccuracies.

- Numerical: In this case, issues related to the accuracy of data that is being generated by particular units – such a Fourier transform unit. In some of these instances, existing C/C++ codes are wrapped – and therefore likely to lead to errors in data type conversions. The provenance question being investigate relates to understanding the impact of data type translation between services connected in a workflow – and whether this type of translation is likely to lead to inaccuracies in the final result generated from the total workflow.

4 Conclusion

These use cases demonstrate various Provenance questions that would be useful to scientists involved in Gravitational Wave analysis. Relationship with the Triana workflow engine has also been outlined.

The *PROVENANCE* Project receives research funding from the European Community's Sixth Framework Programme