

Cloud Computing for e-Science with CARMEN

Paul Watson, Phillip Lord, Frank Gibson, Panayiotis Periorellis, Georgios Pitsilis

School of Computing Science, Newcastle University, Newcastle-upon-Tyne, UK
Paul.Watson@newcastle.ac.uk

Abstract. The CARMEN e-science project (www.carmen.org.uk) is designing a system to allow neuroscientists to share, integrate and analyse data. Globally, over 100,000 neuroscientists are working on the problem of understanding how the brain works. This is a major challenge that could revolutionise biology, medicine and computer science. Solving it requires investigating how the brain encodes, transmits and processes information. In this paper we describe the CARMEN system. This is a generic e-science platform "in the cloud" which enables data sharing, integration and analysis supported by metadata. An expandable range of services are provided to extract added value from the data. CARMEN is accessed over the Web by neuroinformaticians, who are populating it with content in the form of both data and services. We describe the design of CARMEN and show how it is being used to support neuroscience.

1. Introduction

This paper describes how the CARMEN project is using cloud computing to address challenging requirements from the key scientific domain of neuroscience. Understanding how the brain works is perhaps the major challenge remaining in science, and progress in this area could revolutionise several scientific areas, in particular biology, medicine and computer science. It would help us to understand how the genome controls brain development, how to design better drugs, and how to design computer systems that can carry out tasks such as image recognition which are largely beyond existing artificial computational systems.

Globally, over 100,000 neuroscientists are working on the problem of understanding how the brain encodes, transmits and processes information. The primary material for their research are various types of experimental data including molecular (genomic, proteomic and small molecule), neurophysiological (time-series activity), anatomical (spatial) and behavioural. As techniques and instruments improve, the quantities of data being collected are increasing. For example, single electrode recording (at around 3MB/min) is giving way to multi-electrode recording with currently tens, and soon hundreds of concurrent signals being collected. This leads to order of magnitude increases in data collection rate; this is itself likely to be superseded by optical imaging techniques that will increase this by a further factor of 10.

Unfortunately, although data is at the heart of modern neuroscience, and is expensive to collect, it is rarely shared. This is mainly because each instrument manufacturer has their own data format, and so it is unlikely that the analysis tools built by one lab can work on data from another lab. Further, typically each lab describes the

experiments and the data they produce in their own informal metadata format. Therefore, even if another lab could read the data, it is unlikely that they would be able to locate data of interest, or understand the context in which it had been collected.

The overall result of this situation is that there are only limited interactions between research centres with complementary expertise, and a severe shortage of analysis tools that can be applied across neuronal systems.

The CARMEN project was set up in 2006 to address these problems. Its aim is to enable the sharing and collaborative exploitation of both data and analysis code so that neuroscience can get much more value from the data that it is collecting.

This paper describes the design of the CARMEN system. This is a generic e-Science platform which enables data sharing, integration and analysis supported by metadata. An expandable range of services is provided to extract value from the data. CARMEN has adopted a “Cloud Computing” approach in which functionality is accessed over the Web by neuroscientists, who are populating it with content in the form of both data and services. In the rest of this paper we describe the design of CARMEN and give an example showing how it is being used to support neuroscientists.

2. CARMEN Architecture

Cloud Computing is of increasing interest in the computing industry. It is concerned with building systems in which users interact with applications remotely over the internet (typically through a web browser). This approach has several advantages for both application providers and users. It prevents the application writer from having to buy and manage their own hardware; instead they can use highly scalable resources in the cloud to meet their needs. Due to the typical commercial “pay-as-you-go” payment regimes, they are only charged for resources as they need them and do not have to worry about over-provisioning (which wastes money on underused hardware) nor under-provisioning (which can result in disastrously poor performance for users). For users, having services delivered over the web removes the need to deploy, manage and maintain software on their own resources. With the growth of the mobile internet, it also opens up the possibility of being able to interact with a service from many locations - at work, at home, and while travelling. From the point of view of the resource providers, it allows them to exploit centralised data storage and computation in large data centres which, due to economies of scale, reduces costs and energy consumption.

Cloud computing has a somewhat different emphasis from Grid computing, which has largely focused on integrating heterogeneous resources, often across multiple organisations, where no one organisation has sufficient resources to meet the requirements of particularly challenging applications: “The grid integrates services across distributed, heterogeneous, dynamic 'virtual organizations' formed from the disparate resources within a single enterprise and/or from external resource sharing and service provider relationships in both e-business and e-science.” [1]

Of course, it would be possible to combine resources from more than one Cloud, in which case grid techniques would be of interest, but this is not a current focus.

There are limits to what can be achieved with Cloud Computing; highly interactive tasks requiring graphically rich interfaces may not work well as web applications. As will be seen, CARMEN utilises one such application - the Signal Data Explorer [2]—that is deployed on the users’ desktop, and so the project is taking the liberal approach of using web-based services where possible, but supporting desktop services where necessary.

The Cloud computing approach was attractive for meeting the CARMEN requirements largely because of the significant amount of data that will be stored and analysed by scientists. Current estimates put this in excess of 100TB by 2010 for the 20 neuroscientists involved in the project, though if video capture of neuronal activity continues to supersede electrode-based recording this may be a serious underestimate. Where there are huge amounts of data to be processed, it is more efficient to move the computation to the data rather than the other way around [3]. This requires having computational resources closely coupled to the servers holding the data. Cloud computing offers the chance to do this if the cloud is internally engineered with fast networking between the storage and compute servers.

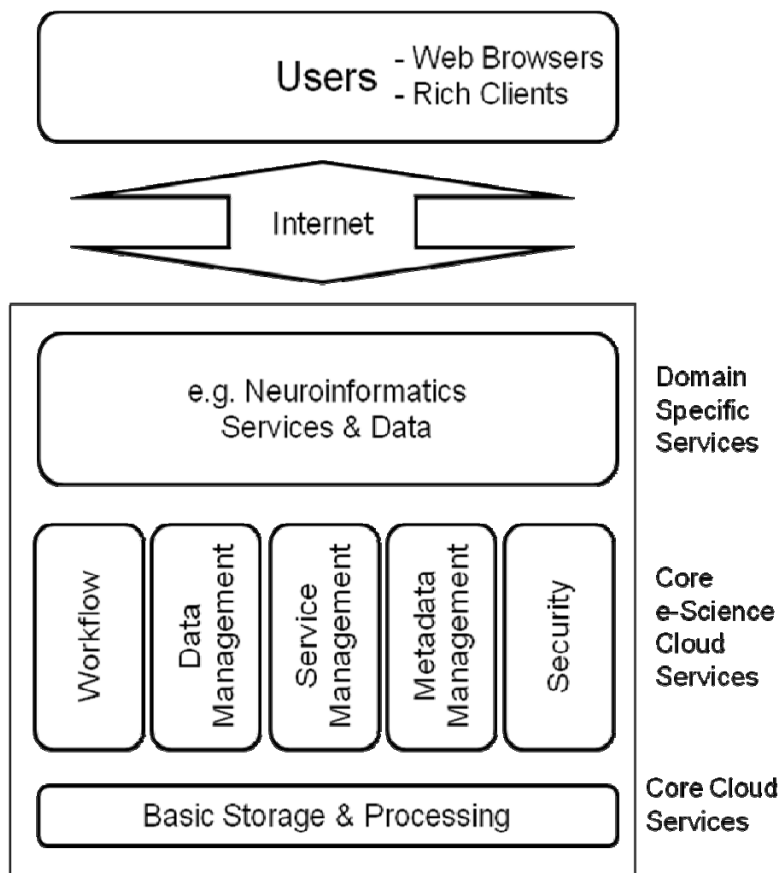


Figure 1. An e-Science Cloud

The basic aim of CARMEN is therefore to provide a cloud (which we name a CAIRN) that neuroscientists interact with through a web-based portal. We have conducted a detailed requirements capture from the scientists in the project. From this, it is clear that the main abilities required by the neuroscientists are:

- to upload experimental data to the CAIRN
- to search for data that meets some criteria (e.g. all data captured under particular experimental conditions)
- to share data in a controlled (user-defined) way with collaborators.
- to analyse data. It is not possible to define a closed set of services that will meet all the analysis needs of all scientists (indeed, new algorithms are being investigated all the time). Consequently, there needs to be a way for scientists to add new services.

Existing cloud computing offerings focus on providing low-level compute and data storage services (e.g. Amazon S3 and EC2). It would be possible to build applications to support these neuroscience requirements directly on this low-level platform, but for CARMEN we chose instead to deploy a set of generic e-science services, and then build domain specific neuroinformatics services and content on top of these (Figure 1).

The selection and design of these services was made based on our experiences in a variety of e-science projects carried out since 2001, targeting a wide range of disciplines from bioinformatics, through transport, to artistic performance. Figure 2 shows another view of the collection of e-Science services in the CARMEN Cloud (which we named a CAIRN). These services are now described in turn.

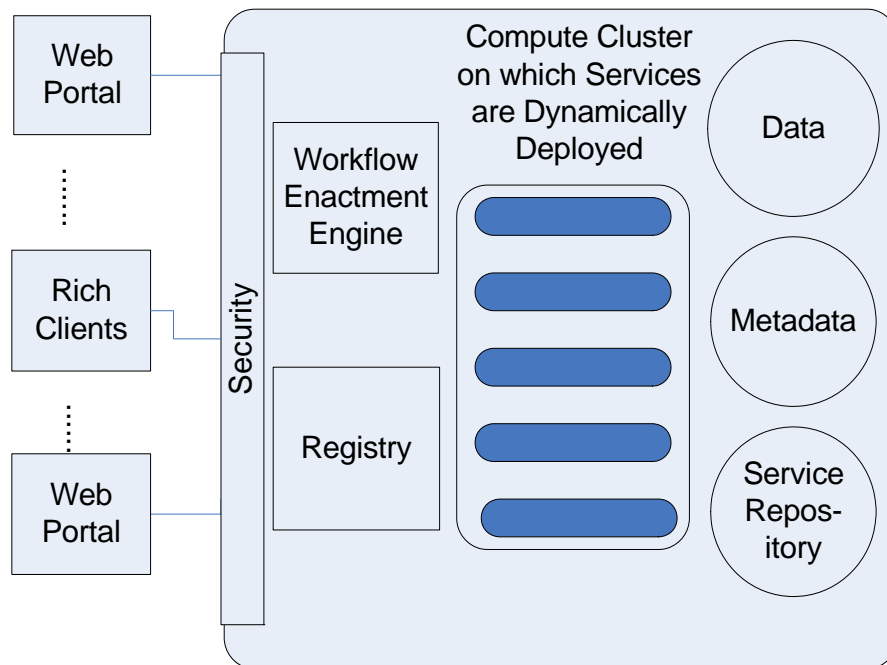


Figure 2. The CARMEN CAIRN

2.1 Data

In [4] Bowker argued that there are three stages in the “Standard Scientific Model”:

1. Data is Collected
2. Data is analysed and papers are published on the results
3. Data is gradually lost

In many scientific disciplines, this occurs because data is kept on individual scientists’ machines and is not subject to long-term curation. Often, once the papers are written the data is considered to be of lower value, with the time and cost implications for maintenance falling on the individual; as a result it is often lost. This has several undesirable consequences. Papers often draw conclusions from data that is not available to others to examine and analyse themselves - reproducibility is a cornerstone of science, but is impossible if the data is not available. Further, in areas such as neuroscience, data that may be expensive to collect cannot be re-used.

The CARMEN approach to addressing this is to provide ways for users to store, analyse and share data in the CARMEN CAIRN, rather than on their own computers. The CAIRN provides storage for file-based data and structured data. In CARMEN, the primary data is typically sampled voltage signal data collected, for example, from Multi-Electrode Array recording. Experimenters then upload them into the CAIRN where they are stored in a filestore. Due to the large volume of data that will be produced by the neuroscience experiments, there is an initial requirement to hold in the region of 100TB of data. We use a Storage Resource Broker (SRB [5]) for this due to its flexibility and scalability. Whilst the primary data is held in file storage, the derived data is stored in a database. This allows researchers to exploit the powerful functionality offered by RDBMS, especially rich querying to select data of interest.

2.2 Metadata

Metadata is essential for a system such as CARMEN which will hold thousands of data collections; without this, it will be hard to discover or understand the stored data. Therefore, when new data is uploaded, users must specify descriptions of the experimental context and conditions through a forms-based interface. The description of a particular experiment is first defined as a “Minimum Information about a Neuroscience Investigation” (MINI) checklist document, analogous to the MIAPE documents for proteomics [6]. The information defined in the MINI documents are then structured using the existing FuGE [7] standard which is a data model that represents components of experimental activity.

The SyMBA[8](<http://symba.sourceforge.net>) package under development in the CISBAN project, (<http://www.cisban.ac.uk/>) is a database implementation of the FuGE schema and is being incorporated within the CARMEN CAIRN to upload, store query and retrieve metadata. When a user uploads data they are presented with a forms-based interface for annotating metadata to data and services.

It is important to have a scheme to uniquely identify data and an associated mechanism to find the metadata associated with an identifier. There is no one dominant standard in this area, but we have chosen to adopt LSIDs (Life Science Identifiers – [9]). They offer a location-independent identifier and an associated protocol that allows both data and metadata to be accessed. This implements the Data Registry service shown in Figure 2.

2.3 Managing Analysis Services

Once users have uploaded new data into the CAIRN, or used the registry to locate existing data that is of interest, they will want to analyze it. For example, in neuroscience, electrophysiological data may first undergo spike sorting to ascribe the data to specific neurons; next statistical analysis may be applied to work out rates of spike firing; finally, graphs may be generated to visualize the results.

The section on data referred to Bowker's work on gradual loss of scientific data [4]. We argued that this had undesirable consequences, such as loss of re-use and reproducibility, which we are addressing in CARMEN by storing data in the CAIRN rather than on individual scientists' machines. However, exactly the same argument can be made regarding the programs that are used to analyse data. We can give an equivalent three stages for programs:

1. Programs are written to analyse data
2. Papers are published on the results of the analysis tools.
3. The programs are gradually lost

These problems occur because programs are deployed on individual scientists' machines and are not subject to long-term maintenance. As with loss of data, this has undesirable consequences. Papers often draw conclusions from data using programs that are not available to others to examine and use themselves – again, reproducibility is impossible if the programs are not available. Further, programs may embody great expertise on the part of the authors, and be developed over many years, but cannot be re-used. Some e-Science projects such as ^{my}Grid have attempted to address this by encouraging authors to “publish” their programs as services (e.g. Web Services) which can be executed remotely by users. This proved largely successful except that it still relies on program owners maintaining the software and the systems on which they run. As a result, scientists would sometimes discover that services that they had come to depend on would suddenly disappear.

CARMEN is addressing this by providing ways for users to store and run programs in the CARMEN CAIRN, rather than on their own computers. The CAIRN will be a repository for the long-term storage and curation of analysis programs as well as data. Programs are packaged by their authors as WS-I conformant Web Services (to give very high levels of interoperability and longevity) so that there is a common way of communicating with, and managing, them. Authors upload their services in a deployable form into the CAIRN where they stored, and metadata about them is entered into a service registry. This ensures that services are preserved so that computations can be re-run, and services re-used.

There is another compelling reason to run the analysis services in the CAIRN. As discussed, neurosciences data sets can be TBs in size. Therefore, it would often not be practical to export the required data out of the CAIRN to a client in for processing - transfer times could be very high, and many scientists would not have the local resources to manage such large datasets. Instead, having programs run in the CAIRN means that data only has to be transferred within the CAIRN. As the CARMEN CAIRN is realized by a cluster with a high performance internal network, this can be achieved at a high data rate.

The Dynasoar [10] dynamic service deployment infrastructure is used to deploy the services on demand from the repository onto the available compute resources when they are invoked. It achieves this by dividing the handling of the messages sent

to a service between two components – a *Web Service Provider* and a *Host Provider* – and defining a well defined interface through which they interact.

- The *Web Service Provider* accepts the incoming SOAP message sent to the endpoint and forwards it to a Host Provider, along with a pointer to the service repository from which a deployable version of the service can be retrieved.
- The *Host Provider's* role is to control the computational resources in the CAIRN. It accepts the SOAP message from the Web Service Provider (along with any associated information) and is responsible for processing it and returning a response to the client.

When the message reaches the Host Provider, there are two possibilities, depending on whether or not the service is already deployed on the node on which the message is to be processed. If the service is already deployed on the node then the Host Provider simply routes the message to the service for processing. This case is shown in Figure 3: a request for a service (s5) is sent by the Consumer to the endpoint at the Web Service Provider which passes it on to a Host Provider. The Host Provider already has the service s5 deployed (on nodes 1 and 2 in the diagram) and so, based on the current loading it chooses to route the request to node 2 for processing. Note that the Web Service provider is not aware of the internal structure of the Host Provider - e.g. the nodes on which the service is deployed nor the node to which the message is sent; this is managed entirely by the Host Provider.

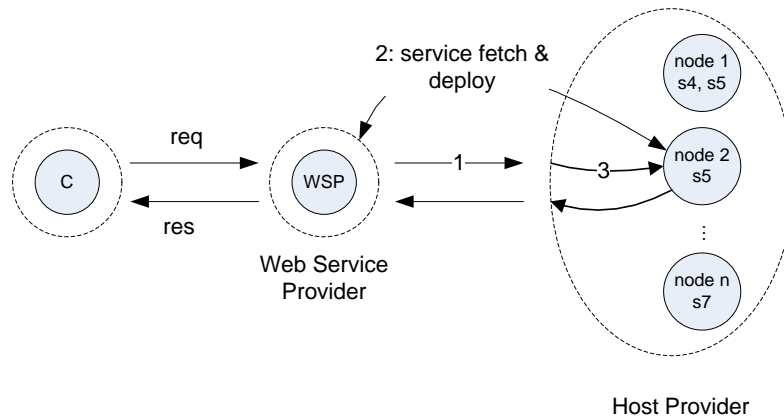


Figure 3. A request is routed to an existing deployment of the service

Figure 4 shows an example of dynamic service deployment. A request for a service (say s8) is sent by the client to the endpoint at the Web Service Provider which, as before, passes it on to a Host Provider (step 1 in the Figure). As s8 is not deployed on any of the nodes it controls, based on loading information it chooses one node (node 2 in this case), fetches the service code from the Web Service Provider and installs the service on that node (step 2). It then routes the request to it for processing (step 3). The response is then routed back to the consumer.

Once a service is installed on a node it remains, ready to process future messages until the Host Provider decides to reclaim it. This has the potential to be much more

efficient than job-based scheduling systems in which each job execution requires the program to be moved and installed.

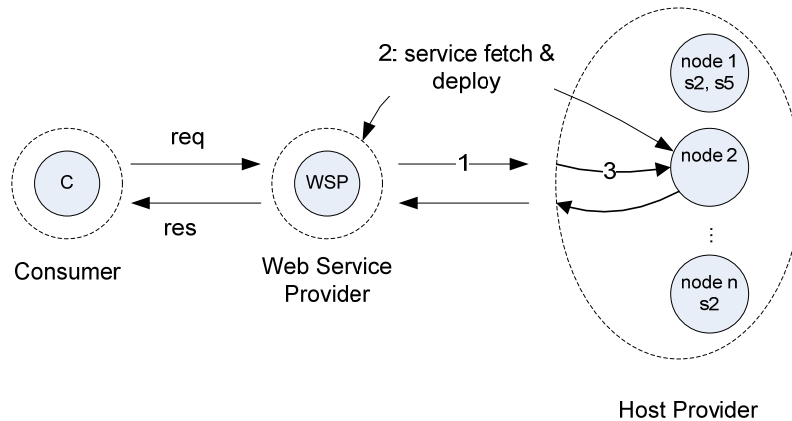


Figure 4. A service is dynamically deployed to process a request

To meet increasing demands for a service from clients, the Host Provider can choose to deploy services on multiple nodes and load-balance requests across them.

Dynasoar is agnostic as to the form of the service to be deployed or its internal structure – so long as the Host Provider has a deployer for that type of service. CARMEN currently deploys services written in a variety of languages (including MatLab, Java, C++, R) as well as services encapsulated in a VMware Virtual Machine. The latter case allows support for services that require complex environments (e.g. specific OS versions, scientific libraries etc.). VMs are often large (5GB is not unusual) and so are costly to move from the repository to the node on which they are to be deployed. However, once deployed, the service can remain in place servicing requests, so reducing the average deployment cost per service call.

2.4 Workflow Enactment

Projects such as ^{my}Grid [11] have shown the power of workflow as a way to visualise and enact processes, such as complex data analyses, that involve data being processed by multiple services. This also encourages programmers to provide components that can be composed with others to create applications, rather than monolithic applications that offer fewer opportunities for re-use. The CARMEN CAIRN contains a workflow enactment engine to allow users to submit workflows to be enacted close to the data on which they operate. As well as removing the need to install the enactment engine on the client, it removes the need to transfer data in and out of the CAIRN to an external enactor: for many workflows that act on large amounts of experimental data, this could be prohibitively expensive. For CARMEN we have adopted the widely used OMII myGrid Taverna/Freefluo [12] workflow engine and adapted it to run in the CAIRN, rather than on the user's desktop.

In CARMEN, workflows are treated the same as any other digital artifact. The user is free to run the Taverna Workbench tool in her own workstation and compose

workflows as they would normally do as standalone user. They can then upload the workflows into the system so that they can be enacted and, if desired, shared with other CARMEN users.

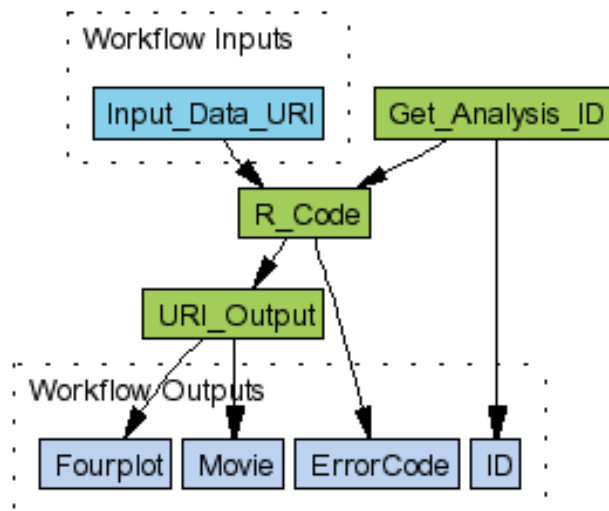


Figure 5. Workflow to Analyse Spike Train Data

Figure 5 shows a simple Taverna workflow that takes in the identifier of a spike train data set and analyses it using an analysis service written in R. This results in the creation of two new data files that are stored in CARMEN – a graph (*Fourplot*) showing the results of the analysis, and a movie that visualizes the movement of activity across a part of the brain. A sample *Fourplot* output is shown in Figure 6.

2.5 Security

Many scientists, including neuroscientists, would not use CARMEN if all data was made publicly available as soon as it was uploaded into the CAIRN. Instead, they want the ability to be able to control access to data and services. For example, researchers may wish their data to be accessible only to themselves until they have done some preliminary analysis to determine that it is a good recording. Then they may make it available to their collaborators until the point at which they have completed and published their analyses. After that they are likely to be keen to make it publicly available, typically alongside a traditional journal publication. The CARMEN security infrastructure therefore deals with the authentication and authorization of users, and will be configured so as to enforce the data owner's requirements. It utilises the Gold [13] security infrastructure, which uses XACML [14] assertions for role and task based access control. To remove the need for users to manipulate XACML directly, data owners can specify their security preferences through a simple GUI from the CARMEN portal.

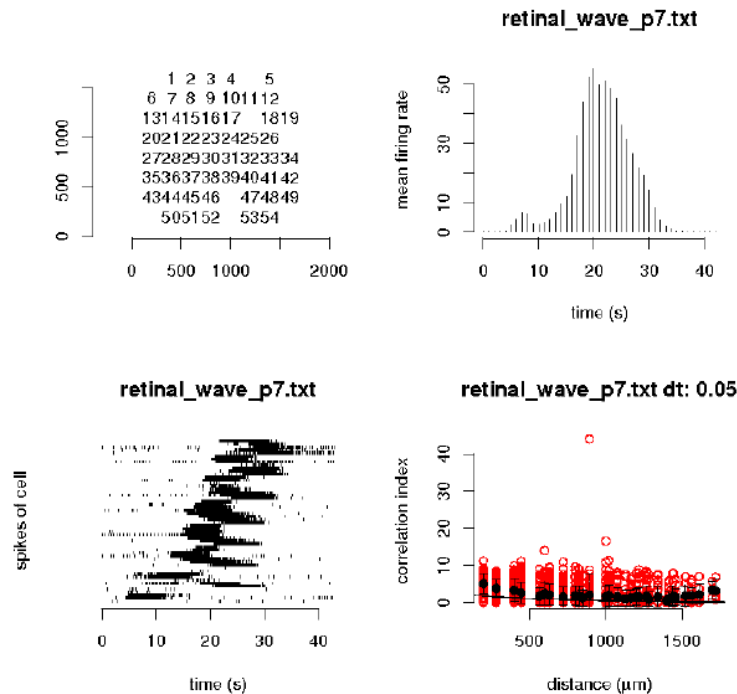


Figure 6. The "Fourplot" graphs produced by running the workflow

3. Conclusions

The CARMEN CAIRN has been designed as a general Cloud Computing solution for e-science applications. We believe that today's commercial Cloud computing offerings (e.g. Amazon) provide relatively low-level services for storing data and performing computation. There is therefore a danger that all e-science applications designed for Cloud computing will build everything from the ground up in a bespoke manner on top of these low-level services. Based on experience with a wide variety of e-science applications over the past 8 years, we have identified a core set of e-science services that we believe can form the basis for domain-specific e-science Cloud applications. In the CARMEN project we have built a prototype system containing these services, and using them to store and process electrophysiological neuroscience data.

This paper has focused on the CARMEN Cloud architecture and the main cloud e-science services. As stated in the introduction, whilst the approach of running everything from a browser is very attractive, not all client tools fit well into this model. In the case of CARMEN, we have identified the need for an interactive tool to allow scientists to visualize and search their data. The Signal Data Explorer [2] is being utilised and enhanced for this, running as a desktop application in order to achieve the required level of interactivity. This is backed up by high performance searching algo-

rithms that will run in the CAIRN, allowing a user to perform searches across huge amounts of data for specific patterns (e.g. types of spike). Future integration with the metadata search services will allow users to home in on patterns that occur in particular experimental conditions.

CARMEN runs until 2010 and during this time we will enhance the system, learning from the experiences of users who will be uploading thousands of datasets and tens of services in order to analyse Terabytes of neuroscience data.

4. Acknowledgements

CARMEN is a large, multi-site collaboration between neuroscientists and computer scientists. The work described in this paper has benefited from discussions involving all members of the extended team but we would particularly like to thank Colin Ingram, Leslie Smith and Alastair Knowles for their work on creating the project. The design of the Computing infrastructure supporting CARMEN is a collaboration between Newcastle and York, and we would like to thank our York colleagues: Jim Austin, Tom Jackson, Martyn Fletcher, Mark Jessop and Bojian Liang. The workflow example is built on the work of Evelyne Sernagor, Stephen Eglen and Christopher Adams. We would also like to thank Anil Wipat and Allyson Lister of CISBAN for SyMBA discussions. Finally, we would like to acknowledge the EPSRC for funding the project.

5. Bibliography

1. Foster, I., et al., *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. 2002, Global Grid Forum.
2. Signal Data Explorer, *Signal Data Explorer*: <http://www.cybula.com/flyers/SignalData.pdf>.
3. Watson, P. *Databases in Grid Applications: Locality and Distribution*. in *British National Conference on Databases*. 2005. Sunderland, UK: Springer-Verlag.
4. Bowker, G.C., *The New Knowledge Economy and Science and Technology Policy*. 2005.
5. Storage Resource Broker, <http://www.sdsc.edu/srb/>.
6. Taylor, C.F., et al., *The minimum information about a proteomics experiment (MIAPE)*. *Nature Biotechnology*, 2007. **25**(8): p. 887-893.
7. Jones, A.R., et al., *The Functional Genomics Experiment model (FuGE): an extensible framework for standards in functional genomics*. *Nature Biotechnology*, 2007. **25**(10): p. 1127-33.
8. Lister, A.L., et al., *Implementing the FuGE Object Model: a Systems Biology Data Portal and Integrator*, in *Newcastle University, Computing Science Technical Report*. 2007, Newcastle University: Newcastle-upon-Tyne.

9. i3c, *I3C Life Sciences Identifiers (LSIDs)*:
<http://www.i3c.org/wgr/ta/resources/lsid/docs/index.htm>.
10. Watson, P., et al. *Dynamically Deploying Web Services on a Grid using Dynasoar*. in *Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC 2006)*. 2006. Korea: IEEE Computer Society.
11. myGrid Project, <http://www.mygrid.org.uk/>.
12. myGrid Project, *Taverna Workflow* <http://taverna.sourceforge.net>.
13. Gold Project, <http://www.goldproject.ac.uk/>.
14. OASIS eXtensible Access Control Markup Language (XACML) TC, *XACML: eXtensible Access Control Markup Language, OASIS Standard Version 2.0*. 2005.