Macromolecular Research, Vol. 14, No. 2, pp 140-145 (2006)

e-Science Technologies in Synchrotron Radiation Beamline - Remote Access and Automation (A Case Study for High Throughput Protein Crystallography)

Xiao Dong Wang, Michael Gleaves, David Meredith, Rob Allan*, and Colin Nave

e-Science Centre, CCLRC Daresbury Laboratory, Warrington WA4 4AD, UK

Received August 11, 2005; Revised January 18, 2006

Abstract: E-science refers to the large-scale science that will increasingly be carried out through distributed global collaborations enabled by the Internet. The Grid is a service-oriented architecture proposed to provide access to very large data collections, very large scale computing resources and remote facilities. Web services, which are server applications, enable online access to service providers. Web portal interfaces can further hide the complexity of accessing facility's services. The main use of synchrotron radiation (SR) facilities by protein crystallographers is to collect the best possible diffraction data for reasonably well defined problems. Significant effort is therefore being made throughout the world to automate SR protein crystallography facilities so scientists can achieve high throughput, even if they are not expert in all the techniques. By applying the above technologies, the e-HTPX project, a distributed computing infrastructure, was designed to help scientists remotely plan, initiate and monitor experiments for protein crystallographic structure determination. A description of both the hardware and control software is given together in this paper.

Keywords: e-science, grid, crystallography, portal, e-HTPX, web service.

Introduction

e-Science refers to the large-scale science that will increasingly be carried out through distributed global collaborations enabled by the Internet. It can easily provide access to very large data collections, very large scale computing resources and remote facilities.

The Grid is a service-oriented architecture proposed to bring all these resources together and make a reality of such a vision for e-Science. The Grid can be defined as an enabler for Virtual Organisations: 'An infrastructure that enables flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions and resources.'[1]

Web services, which are becoming adopted within Grid computing and e-Commerce, are server applications that enable clients to use remote procedure calls defined in XML over a network (typically over HTTP). They involve three major operations: publishing, finding and binding.[2]

A portal is a Web-based application that acts as a gateway between users and a range of such Web services. It provides personalisation, single sign-on, aggregation and customisation features in addition to other Grid functionality.[3,4]

The main use of synchrotron radiation (SR) facilities by protein crystallographers is to collect the best possible diffraction data for reasonably well defined problems. The process is therefore suitable for automation. Significant effort throughout the world is therefore being given to automate SR protein crystallography facilities so scientists can achieve high throughput, even if they are not expert in all the techniques.[5] Based on e-Science technology, a description of both the hardware and control software is given together in this paper.

*Corresponding Author. E-mail: r.j.allan@dl.ac.uk

Web Services and Middleware

Web services are Web-based enterprise applications that use open, XML-based standards and transport protocols to exchange data with calling clients. This can allow us to hide the details of how the service is implemented, only a URL and data types are required. As XML is used for service description and transport protocol, the language used in service implementation and on what platform the service is running become irrelevant to the client. So it is very flexible for the e-HTPX developers from different sites to develop services on different operation system platform. Web services have three major roles: service process, service description and message transformation.. The Web service architecture is shown in Figure 1.

Service processes enable clients to discover the services they want and service providers to publish the services they

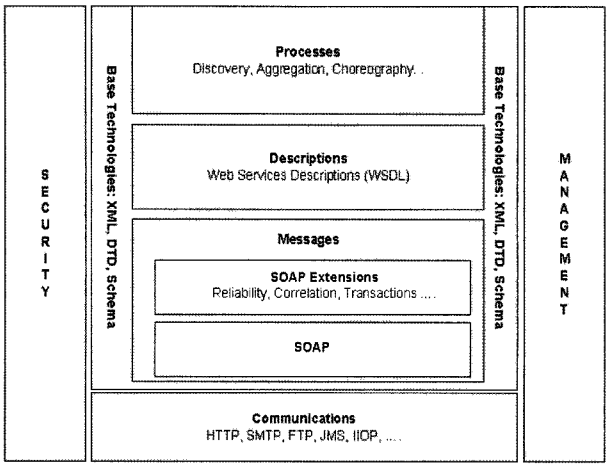

Figure 1. Web service architecture (from W3C Working Group).

provide. Usually it implements as Universal Description, Discovery and Integration (UDDI). UDDI is a Web services API for publishing and discovering the existence of Web services and a registry for managing information about Web services. UDDI defines three levels for information:

1. White Pages - used to query companies with their attributes.
2. Yellow Pages - used to query and categorize businesses by taxonomies.
3. Green Pages - used to define how to interact with the Web Services.

The XML Web Service Definition Language (WSDL) is used to describe and define the Web Services and their functions. It describes what the services can do, where they reside, and how to invoke them. The WSDL structure can be described as below:

```
<definitions>
  <types>...</types>
  <message>...</message>
  <portType>...</portType>
  <binding>...</binding>
  <service>...</service>
</definitions>
```

The <types> element contains XML Schema defining the datatypes that are to be passed to and from the Web service. The messages that will be exchanged between the client and the service can be put in the <message> element. <portType> and the sub-element <operation> will be defined in terms of where they fit in the functionality of the Web service. So a portType is analogous to a class. An operation is analogous to a method in that class. The <binding> element defines the protocol that the client will actually use to interact with the Web service. There are three protocols: SOAP, HTTP and MIME. The last element of a WSDL file is the <service> element. This element defines the actual location of the Web service that client can use to interact with the Web service.

The message transformation among Web Services, Clients and Registry is standardized in Simple Object Access Protocol (SOAP). SOAP is a XML based communication protocol. It provides a simple lightweight mechanism for exchanging structured and typed information between peers in a decentralized, distributed environment using XML. A SOAP message consists three parts: Envelope, Header and Body. The <Envelope> is the root element of a SOAP message, and it can contain two core elements. The <Header>, which contains information about the message such as security information, is optional, and then the <Body> element is mandatory and contains the XML message that is to be sent. This could be an XML message, or it could be some XML that represents a remote procedure call, or a response from a remote call.

The e-HTPX system architecture has utilised these technologies. Consequently, it has been possible to design e-HTPX using a modular architecture where separate facilities and resources can be added or removed dynamically.

Grid service is a Web service that provides a set of well-defined interfaces and that follows specific conventions. The interfaces address discovery, dynamic service creation, lifetime management, notification, and manageability. The conventions address naming and upgradeability. Different with Web service, Grid service is built around the concept of a Grid service instance. A Grid service instance potentially is transient which is created as client remote calling. So Grid service is stateful by the introducing of Grid service instance. Grid service has some middleware choices: Globus, Condor, SRB and Sun GRID Engine.

Globus is the primary middleware used in current UK e-science projects. The Globus toolkit is open source software developed by the Globus Alliance, see http://www.globus.org. The toolkit includes software for security, information infrastructure, resource management, data management, communication, fault detection, and portability. It is packaged as a set of components that can be used either independently or together to develop applications. The toolkit components are shown in Figure 2. Its core services, interfaces and protocols allow users to access remote resources as if they were located within their own machine room while simultaneously preserving local control over who can use resources and when. So the developers of different sites can deploy their unique modes of operation.

Portal

Grid portals are emerging technologies with improving specifications and enhanced support through recent standards. A portal is a Web-based application allowing users to access range of different high-level services using a browser interface. A portal framework can provide presentation

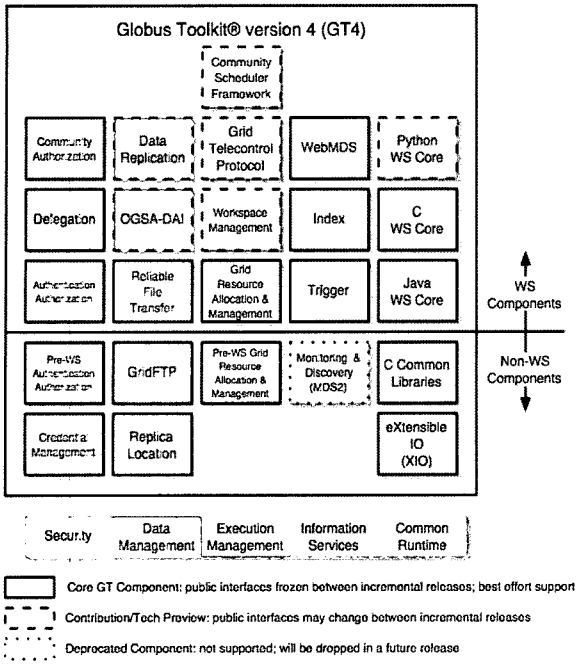

Figure 2. Globus toolkit components (from Globus Alliance).

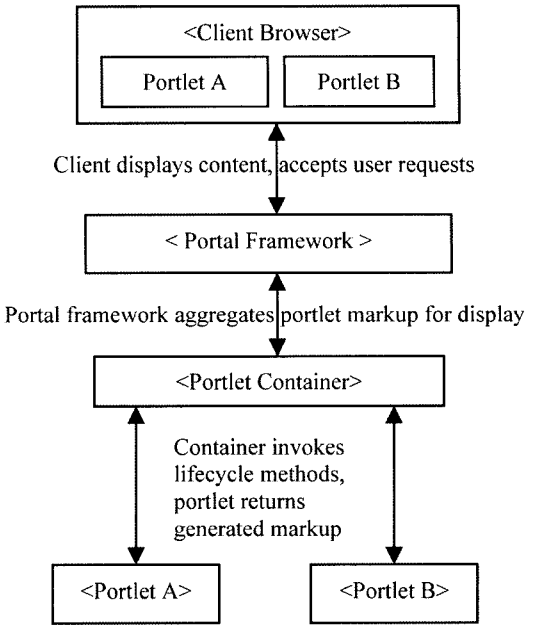

Figure 3. Portlet/portal architecture.

capabilities for Globus middleware in additional to personalisation and content management. It can customize content for users, integrate with other applications to allow access to multiple systems with a single sign-on and aggregate content from different sources. A so-called 2nd generation portal normally consists of different portlets to process consumer requests to these services and generate dynamic content from the responses. From a user's perspective, a portlet is a window in a portal that provides a specific service, for example, a calendar or news feed. From an application development perspective, a portlet is a software component written in Java, managed by a portlet container, which handles user requests and generates dynamic content. Portlets are used in portals as self-contained pluggable user interface components to the services. Portlets can be developed in different languages and managed by different portlet containers. The Portlet/Portal architecture is shown in Figure 3.

Portlet can pass information to a presentation layer of a portal system. The content generated by a portlet is also called a fragment. A fragment is a chunk of markup language (e.g., HTML or XHTML) adhering to certain rules and can be aggregated with other fragments to form a complete document. The content of a portlet is normally aggregated with the content of other portlets to form the portal page. A portlet container manages the lifecycle of portlets in a portal.

A portlet container provides a runtime environment in which portlets are instantiated, executed, and finally destroyed. Portlets rely on the overall portal infrastructure to access user profile information, participate in a presentation window, and communicate with other portlets to access remote contents, lookup credentials, and store persistent data. A portlet container manages and provides persistent storage mechanisms for portlets.

A portlet container is not a stand-alone container like a Java Servlet container; instead, it is implemented as a layer on top of the Java Servlet container and reuses the functionality provided by the Servlet container.

A Portal Framework takes responsibility for message flow from user to service and for inter-portlet communication. The messages can be stateless or stateful, but are normally stateless as software agents are context independent. The framework may either add state information to the message for multiple interactions per user, or store the state information in a persistent way removing the need for services to maintain state when invoked from different portlets. Consequently, a service failure does not result in the loss of state as the state of services is known. A service providing the software agent can even be replaced dynamically during the execution with another, equivalent service. This potentially makes recovery from partial failure relatively easy and services seen by the user can be made reliable.

Portlets are not confined to one portal framework. Based on standard Web services technology, OASIS[6] released the Web Services for Remote Portlets (WSRP) also in 2003 aiming to define a standard for interactive, user-facing Web services to make portlets hosted by different geographically distributed portal frameworks accessible in a single portal. Unlike traditional Web services however, WSRP defines a protocol which is focused on transferring the markup fragments generated by portlet producers. Although WSRP is still at an early stage as far as implementation is concerned,

it indicates the future of portlet/ portal development. Ideally, a deployed service with a portlet interface can be published and consumed in many different portals/Portal Frameworks. This remote sharing of a single portlet will greatly ease the construction of large-scale portal based systems, enabling them to be more scaleable, manageable and maintainable.

The advantages of a portlet-based architecture are that, principally each underlying function or service can be associated with a unique portlet. This makes it easy to add new services, and many different groups can then independently contribute portlets, which can be plugged into the portal. Using WSRP they can be distributed and managed remotely on many servers and the portals described by WSDL-like information. Each user can select and configure the portlets they wish to use and the selection can become part of a persistent "context".

e-HTPX Implementation

The vast amounts of data coming from the genome projects have generated a demand for new methods to obtain structural and functional information about proteins and macromolecules. This has led to a demand for high throughput techniques to determine the structure of important proteins. The e-HTPX project provides a distributed computing infrastructure designed to help structural biologists remotely plan, initiate and monitor experiments for protein crystallographic structure determination. Key services for e-HTPX have been developed by laboratories for synchrotron radiation, e-Science and protein production. The principle organisations include: CCLRC Daresbury Laboratory e-Science and SRS departments (The Council for the Central Laboratory of the Research Councils), European Synchrotron Radiation Facility (ESRF), Collaborative Computational Project Number 4 (CCP4), York University Structural Biology Department (YSBL) and Oxford University Protein Production Facility (OPPF). The services developed for the project define the 'e-HTPX Workflow'. The workflow is illustrated in Figure 4. Remote access to these services is implemented by a collection of Web and Grid services.

The initial stages of the e-HTPX workflow (target selection and protein production) are centred on project planning and initiation. This involves on-line completion and submission of requests to protein production facilities for the growth of specific protein crystals. Services have been developed to enable the remote monitoring of the progress of crystal-growth and to plan the delivery of crystals and associated safety information to a specified synchrotron.

Following crystal delivery, the user can access data collection and monitoring services. The key processes included in data collection are:

1. Matching crystals with appropriate experimental data that was previously sent to the synchrotron via the portal.
2. Automatically screening sent crystals in an unattended beamline operation.
3. Returning screening results to the scientist, with recommended diffraction plan for approval.
4. Full data collection on crystals approved by the scientist and the return of: experimental statistics, compress image file (mtz format) and raw data.[7]

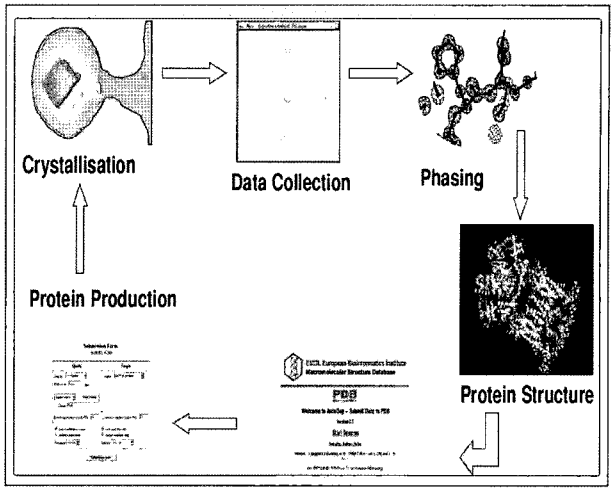

Figure 4. The e-HTPX Workflow spans all stages of protein crystallography experiments, from protein production to solution of the digital structure of the protein and deposition of the protein into a protein data bank.

In order to achieve data collection, the e-HTPX services link to the on-site database at the synchrotron (ISpyB). The data stored in ISpyB is then passed to DNA which is a system for automating data collection and processing at synchrotron beamlines.[8,9]

Using e-HTPX for the data collection stage has the following benefits:

1. For the user, they do not have to travel to the synchrotron facility in order to collect their protein crystal data.
2. For the synchrotron, the automated nature of e-HTPX allows them to schedule the analysis of protein crystals at times when scientists would not want to attend a beamline. Therefore maximising the throughput and utilisation of the synchrotron resources.

When data collection is completed, data transfer to Grid storage resources (e.g. UK National Grid Service cluster machines) is implemented using Grid middleware. Remote HPC data processing services for determining the 3-dimensional structure of the target protein are also provided and can be accessed through job submission portal pages. These stages utilize a combination of Web services and Grid middleware technologies. Following post-data collection processing, the protein structure information determined in academic projects will eventually be deposited into public databases such as the Protein Data Bank provided by the

European Bioinformatics Institute (EBI).[710]

An e-HTPX portal has also been developed for the project. As illustrated in Figure 5, the portal is a client application designed to provide a single point of access to the underlying e-HTPX Web service hierarchy. The portal provides an interface to input necessary data, and acts as an access gateway and Web service-response hub to the e-HTPX services. The main aim of the portal is to greatly simplify the coordination and remote execution of e-HTPX protein crystallography experiments. The portal can also be used to monitor data collection experiments and access beam-line data.

Two portal architectures have been designed in order to suit different research environment needs; a) the service-site portal, which is maintained by each synchrotron, offers a standard service for the majority clients and, b) the client-site portal, which can be installed at a client institution, and allows configuration and close integration of the portal functionality with a client institutional LIMS (Laboratory Information Management System). The client-site portal also places responsibility on the user for storage of their (potentially sensitive) data. For portal software development, we have adopted an object-oriented (OO) programming language, principally Java. The OO language can 'plug and play' to match the modular approach to the hardware. The portal technology is integrated as an interface to hide the complexity of Web service operations.

Significant effort within e-HTPX has also centred on the development of a comprehensive e-HTPX XML message model. As illustrated in Figure 6, an e-HTPX experiment inevitably requires numerous communications between the remote portal user and the various e-HTPX Web services hosted at the different laboratories involved in the project. The message model, defined in XML schema, has been developed to standardize the format of each communication throughout the e-HTPX workflow.[10] The message model describes a wide variety of data from diffraction plans to administration information. The main advantage provided by the XML message model is that individual service sites

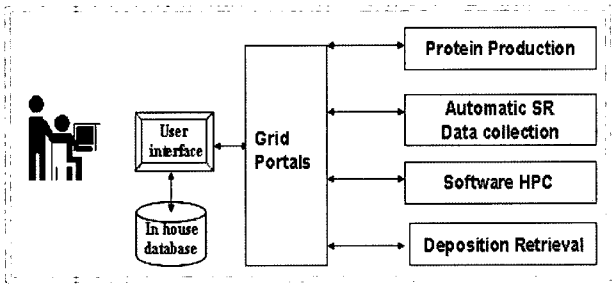

Figure 5. All e-HTPX Web services can be accessed and managed using the e-HTPX Web portal/hub. The portal provides a single point of access to the underlying distributed computing architecture and hides the user from the complexity of this computing infrastructure.

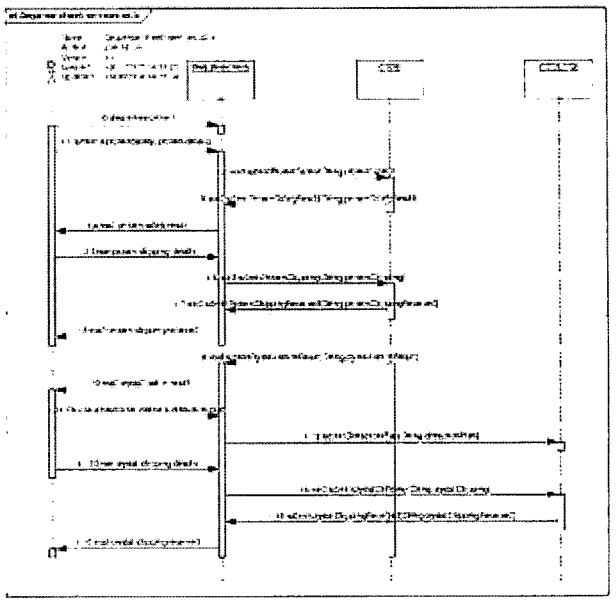

Figure 6. UML Model showing communications between the user and the various laboratories involved in the e-HTPX workflow. The data described in each communication is standardised by the e-HTPX XML message model. The portal/hub centralises the requests and responses.

are free to implement their own services according to an agreed standard and can implement their own choice of platform (platform independence).

Conclusions

E-Science concentrates on the development of new technologies designed specifically for the processing of huge quantities of globally distributed data, accessing large scale internet computing resources and coordinating remote facilities. The implementation of high throughput protein crystallography using a distributed computing infrastructure requires procedures for remote planning, initiation and monitoring of experiments for structure determination. This paper describes e-HTPX as a case study of the use of e-Science technologies in synchrotron radiation beamlines.

The e-HTPX system architecture has a modular design involving a collection of geographically distributed services. Each remote service defines their data inputs/outputs according to the e-HTPX XML message model. Consequently, each facility involved in the workflow can separately implement their site-specific services. This architectural model is very flexible for extension, administration and maintenance and clearly separates web-service code from web-service client code.

The principal e-HTPX services can be summarised as follows: services for requesting the crystallisation of proteins (including remote monitoring of crystal growth), services

for requesting the delivery of the proteins and associated data to a synchrotron radiation source (safety information, diffraction plan etc), services used for remote planning and monitoring of X-Ray diffraction experiments (data collection), services for high performance compute (HPC) services for the solution of the (digital) structure of proteins, and services for the deposition of the protein structure into the Protein Data Bank.

The e-HTPX portal has been designed to provide a single point of access to all the e-HTPX services and defines the interface between end users and the distributed resources. The portal simplifies e-HTPX experiments by hiding the underlying complexity of the distributed computing infrastructure. Uses can perform e-HTPX experiments remotely using only a web browser.

References

(1) *The UK Research Councils WebSite.* http://www.rcuk.ac.uk/escience/.
(2) X. D. Wang, *Grid & Web Services: Web Service Resource Framework,* CCLRC e-Science AHM, 10/1/2005.
(3) R. J. Allan, C. Awre, M. Baker, and A. Fish, *Portals and Portlets 2003.* Proc. NeSC Workshop 14-17/7/2003 http://www.nesc.ac.uk/technical_papers/UKeS-2004-06.pdf.
(4) X. D. Wang and R. J. Allan, *Portlet, WSRP and Applications*, GridSphere and Portlets Workshop, NESC, 3-4/3/2005.
(5) *e-HTPX Project.* http://www.e-htpx.ac.uk.
(6) *WSRP Specification 1.0 by OASIS.* http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf.
(7) D. Meredith, *e-HTPX Architecture and Remote Access Methods*, http://clyde.dl.ac.uk/e-htpx/publicDownloads.htm.
(8) http://www.esrf.fr/exp_facilities/BM14/escience/ispyb/ispyb.html.
(9) *DNA Project.* http://www.dna.ac.uk/.
(10) R. Keegan *et al. e-HTPX-HPC, Grid and Web Portal Technologies, Protein Crystallography* http://www.allhands.org.uk/2004/proceedings/papers/101.pdf.