Short Message Service in a Grid-Enabled Computing Environment

Fenglian Xu, Hakki Eres, and Simon Cox

School of Engineering Sciences,
University of Southampton,
Highfield, Southampton, SO17 1BJ, UK
{F.Xu, Hakki.Eres, sjc}@soton.ac.uk
http://www.geodise.org/

Abstract. Mobile computing devices such as mobile phones together with a land-based and wireless communication network infrastructure are the existing technical prerequisites for continuous access to networked services. The security of the system is a high concern in this environment, as well as its usability. This paper presents a network infrastructure for using the Short Message Service (SMS) to communicate with mobile phones via a grid-enabled service. The network system includes: a messenger server, a messenger client, Globus Servers and the Short Message Service Centre (SMSC) tied together with XML-RPC, TCP/IP and GRAM protocols for communication. A Matlab tool to use the grid-enabled SMS has been implemented and we demonstrate its use in a grid-enabled application.

1 Introduction

Pervasive computing can deliver access to information with no limits. The Geodise project [1] aims to aid engineers in the design process by making available a suite of design optimisation and search tools and Computational Fluent Dynamics (CFD) analysis packages integrated with distributed grid-enabled computing, data, and knowledge resources. Here we use the mobile phone short message service (SMS) to communicate to a service or an application in a grid-enabled environment results of, for example, progress in an Engineering Design Calculation. The SMS should provide a two-way channel to schedule messages crossing most mobile network operators immediately and it also should be robust, reusable and platform independent.

This paper demonstrates a case of how to use a mobile text message function in a grid-enabled computing environment [2] with a simple usage. The process is done remotely via Globus 2.2 [3] middleware software integrated into Matlab [4] which we use as a scripting/hosting environment. A full lifecycle of the SMS service software from design, implementation and test is presented. A usage of a tool is demonstrated to show how the SMS service is used in the entire grid-enabled environment [5]. Related work is discussed in Sect. 2. The architecture and the implementation of the message system are described in Sect. 3. An

H. Kosch, L. Böszörményi, H. Hellwagner (Eds.): Euro-Par 2003, LNCS 2790, pp. 1148–1152, 2003.

exemplar of using the SMS is depicted in section Sect. 4. Conclusions and future work are summarised in Sect. 5.

2 Related Work

Several recent efforts have been proposed to address the issue of incorporating SMS capabilities for the remote monitoring and control of automation systems. The OPC-SMS gateway [6] is a service which enables mobile users to send a SMS message to request specific information from OPC servers and to be notified by OPC servers. A similar mechanism has been used for sending notifications to a cellular phone in a grid environment [7]. IBM has focused its resource on new technologies that connect to any application, from any device over any network, using any style of interface [8]. With recent advances in mobile computing technology, it is possible to support wireless messaging services which enable mobile users to access their messages at anytime and anywhere [9]. However, they all have a complicated usages to send/access the messages and the mobile phones play roles same as the computers.

Several commercial SMS service providers allow users to send messages to mobiles via either a web application or their own applications which use APIs from the provider. Redcoal SMS [10] uses Simple Mail Transfer Protocol (SMTP) protocol so that users have to access the Internet to read and send messages. However, this service only works with Vodafone networks. Lucin SMS [11] has a web service with an engine to enable users to send text messages, ringtones, logos and retrieve a list of messages. Level 9 [12] provides a Short Message Service Centre (SMSC) which enables a two-way communication at a low cost. They also provide APIs in Perl, C/C++ and Java programming environment. Java programming is object-oriented and platform independent which meets the system requirements mentioned earlier.

3 System Architecture and Implementation

The infrastructure of the SMS system is shown in Fig. 1. The Messenger Service (MS) is the main entity which enables computer applications to send/receive messages to/from mobile phones. The MS includes a stack of services: Globus 2.2 Server, Messenger Client and Messenger Server. They provide the functions of security, sending and receiving the messages over the Internet crossing firewalls. The Globus server enables the message client to be run remotely and safely in a grid-enabled environment using the Globus grid computing toolkit. The protocol between the Globus server and the remote application is via Globus Resource Allocation Management (GRAM) and the protocol between the messenger client and the messenger server is TCP/IP. The messenger server waits for connections from clients and it then forwards the messages from the clients to the SMSC provided by Level 9. The SMSC enables communications between GSM networks and the Internet for sending and delivering text messages/binary images to current/3G mobile phones. The protocol between the messenger server

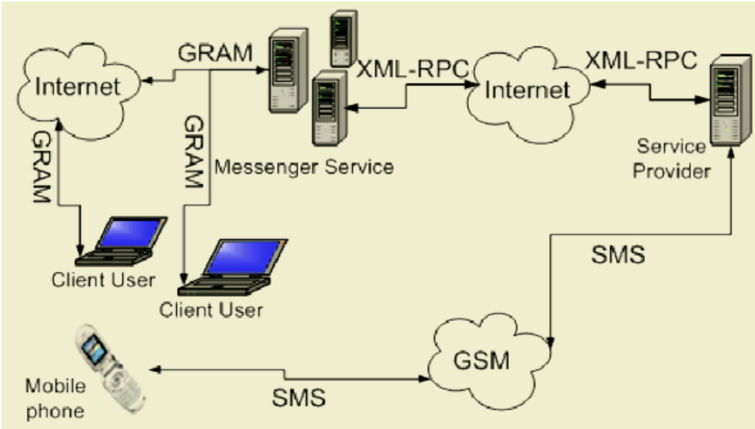


Fig. 1. System Architecture

and the SMSC software is via XML-RPC [13] over the Internet. XML-RPC is a simple, portable way to make remote procedure calls over HTTP. We have integrated the functionality into Matlab which is used as a hosting/scripting environment in Geodise.

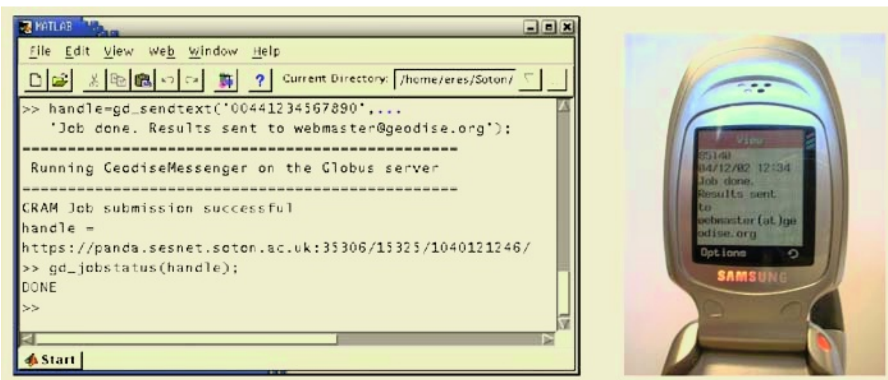
Security is an important issue in a network system due to the data transfer over the Internet and over the wireless network. A combination of authentication and authorisation is used for controlling who can use the software and what permissions are granted to the users. Globus Toolkit 2.2 Grid Security Infrastructure (GSI) [14] is used for enabling authentication and communication over an open network. Users can benefit from a single sign-on and delegation to run a job remotely via GRAM with a trusted authority, such as the Globus organisation or the UK-eScience Certificate Authority [15].

The SMS system is based on the client-server design pattern and is implemented with Java socket programming. The messenger client is responsible for requesting a connection to the messenger server, sending a short text message and closing the connection. The messenger server creates a server socket for incoming connections and listens for clients constantly for accepting a new client connection and starting a message received thread. The interface between the Matlab tool and the messenger client is via Resource Specification Language (RSL) [16] which is understood by GRAM. The message server acts as an SMSC client to connect with the SMSC and makes a remote procedure call with arguments which include the following fields: user name, password, mobile number and text message.

4 Application Exemplar

Design optimisation and search is a long and repetitive process. The process begins with a problem definition or a tool to generate a meshed geometry; a

CFD analysis is then performed from which an objective function is calculated. Design Search is the process by which the geometry is systematically modified to improve the value of the chosen objective function. This process often takes a long time (up to weeks) and may be fraught with problems caused by the compute infrastructure, meshing, or the actual CFD runs, or with the optimisation failing to make much progress in improving the design. It is therefore important to monitor the ongoing design run to ensure that it is progressing and hasn't stalled or terminated anomalously. This section demonstrates how to use the `gd_sendtext` tool to send messages to mobile users in the Matlab environment. This tool checks the security before sending a message and returns a job status to see if the message is sent successfully. The Geodise toolkit [17] provides a set of Matlab tools, such as `gd_proxyquery`, `gd_createproxy` and `gd_jobsubmit`, to perform these tasks. The job is submitted by giving an RSL string and a remote host name as arguments. The RSL string combines the remote directory, the executable program name and the arguments to execute the program. An example of using `gd_sendtext` is shown in Fig. 2. As it can be seen from Fig. 2(a), a message is sent by the tool `gd_sendtext` in the Matlab environment and the job handle can be obtained for checking the job status by using `gd_jobstatus`. Fig. 2(b) shows that the message is received from a mobile phone in a second.



(a) Using `gd_sendtext` to send a message (b) Message received from `gd_sendtext`

Fig. 2. Example of using `gd_sendtext`

5 Conclusions and Future Work

The power of this application has enabled users to be notified beyond geographic boundaries in a grid-enabled environment. The messenger service can be plugged into any grid-enabled environment and it is easy to use. We have demonstrated how the short message service has been integrated with a grid-enabled computation of design search and optimisation system to allow monitoring of a job.

The advantage of using the short message service instead of the email protocol SMTP is that it does not require an Internet connection and is fast and easy to use.

We have also developed a two way process in which a message from a phone can be used to steer or terminate calculations. Future work will include use of third generation (3G) technology to send output, e.g. graphics, from a code to a 3G mobile. We will also move to using the Open Grid Services Architecture [18] as reference implementations become available as an open standard for connecting our service to grid-enabled applications.

Acknowledgements. We are grateful to Level 9 networks for providing SMS service capability for this project and EPSRC grant GR/R67705/01.

References

1. The Geodise Project. (2002) `http://www.geodise.org/`
2. Global Grid Forum. (2002) `http://www.gridforum.org/`
3. The Globus Project. (2002) `http://www.globus.org/`
4. Matlab 6.5. (2002) `http://www.mathworks.com/`
5. Pound, G.E., Eres, M.H., Wason, J.L.and Jiao, Z., Keane, A.J., Cox, S.J.: A Grid-Enabled Problem Solving Environment (PSE) for Design Optimisation within Matlab. Proceedings of 17th IPDPS (2003)
6. Kapsalis, V., Koubias, S., Papadopoulos, G.: OPC-SMS: a Wireless Gateway to OPC-based Data Sources. Computer Standards & Interfaces **24** (2002) 437–451
7. Kranzlmueller, D.: Dewiz – Event-based Debugging on the Grid. EUROMICRO-PDP'02, IEEE (2002)
8. IBM Resource. (2002) `http://www.research.ibm.com/thinkresearch/pervasive.shtml`
9. Tan, D.H.M., Hui, S.C., Lau, C.T.: Wireless Messaging Services for Mobile Users. Journal of Network and Computer Applications **24** (2001) 151–166
10. (Redcoal Website) `http://www.redcoal.com`
11. (Lucin Website) `http://www.soapengine.com/lucin/soapenginex/smsx.asmx`
12. (Level9 Website) `http://www.level9.net:2048/RPC2`
13. (XMLRPC Website) `http://xmlrpc-c.sourceforge.net/xmlrpc-howto/xmlrpc-howto.html`
14. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A Security Architecture for Computational Grids. In: 5th ACM Conference on Computers and Communications Security, San Francisco, California (1998)
15. UK Grid-Support Centre. (2002) `http://www.grid-support.ac.uk/`
16. The Globus Resource Specification Language (RSL) v1.0. (2002) `http://www-fp.globus.org/gram/rsl_spec1.html`
17. Eres, M.H., Pound, G.E., Jiao, Z., Wason, J., Xu, F., Keane, A.J., Cox, S.J.: Implementation of a grid-enabled Problem Solving Environment in Matlab. Accepted by the Workshop on Complex Problem-Solving Environments for Grid Computing (Held in conjunction with the ICCS) (2003)
18. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: The Physiology of the Grid - an open grid services architecture for distributed systems integration. `http://www.globus.org/ogsa/` (2002)