Extending the MIAS-Equator Toolkit to Intermittently Available Network Environments

Alastair Hampshire[1], Chris Greenhalgh[1]

[1] School of Computer Science and IT, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK
{axh, cmg}@cs.nott.ac.uk

Abstract. The present service oriented approach to grid computing is designed to support synchronous operations on always connected services and cannot easily be used to support asynchronous, intermittently connected services on the grid. Middleware, such as the globus toolkit assume the availability of a permanent connection to a grid service. However, many applications require the integration of services and clients which are mobile, and consequently only intermittently connected to the fixed Grid infrastructure. The Grid Based Medical Devices for Everyday Health project has produced a toolkit designed to expose a wearable medical sensing devices as OGSI grid services, to explore the extent to which the grid can support remote patient monitoring; intermittent network availability was found to be a critical problem for grid based mobile sensing applications.

This paper describes a framework for supporting both mobile grid clients and services in an intermittently connected network environment. A prototype implementation has been developed and used to support the operation of the wearable medical device described above. Initial tests show the framework successfully supporting invocations in the face of intermittent network non-availability.

1 Introduction

Grid Computing symbolises a vision in which users share and access compute and storage resources, such as databases, processor servers and mobile devices, which are distributed across large geographical areas and multiple organisational boundaries. One approach to grid computing utilises a *service-oriented architecture* in which clients interact with resources through entities known as grid services. The Open Grid Services Infrastructure (OGSI) [OGSI] defines a standard set of facilities for interacting with these grid services.

There are many applications which would benefit from using service-oriented grid techniques to support mobile/remote devices. The Eqautor eScience program has used grid services to facilitate the interaction with a remote environmental monitoring device deployed in the Antarctic [Antarctic]. The Grid Based Medical Devices for Everyday Health project has produced a toolkit designed to expose sensing devices as grid services; this toolkit has been used to explore the extent to which the grid can be used to support wearable mobile sensing devices for health monitoring purposes [Medical]. These projects have: demonstrated a clear need for OGSI to support mobile applications; identified that OGSI is designed to support synchronous, always connected services and cannot easily be used to facilitate the interaction with asynchronous, intermittently connected grid services and resources.

This workshop paper describes a framework which can be used to support interaction with OGSI grid clients and services on intermittently available devices. The paper also describes a test scenario in which the framework has been used to extend the toolkit described above to support the interactions with a wearable medical sensing device. The work combines techniques from mobile aware protocols and middleware to specifically address Grid requirements and idioms.

2 Related Work

2.1 The Open Grid Services Infrastructure

OGSI specifies a standard set of facilities required for the creation and management of grid applications, through the interaction with Grid services. The OGSI specification extends the Web services specification [WSA] to provide the functionality required to build large scale wide area distributed computing systems.

The OGSI idiom uses grid factory services to create grid service instances. A client (or clients) then interacts with the grid service instance, which is able to maintain state data for the duration of the client(s) interaction(s). OGSI defines the use of a *Grid Service Handle* (GSH) and a *Grid Service Reference* (GSR). A GSH is a persistent handle to the service, but does not contain protocol or location information. The GSR is a transient network pointer with an associated lifetime, which can be used to locate and invoke the grid service. The GSH can be resolved to a GSR using a *Handle Resolver Service*. OGSI also provides the following functionality:

- Service Data (SD): Standard mechanism for accessing and updating service state.
- Notifications: A mechanism to inform interested parties about changes to SD.
- Lifetime management: Mechanisms for controlling the lifetime of transient grid service instances.

2.2 Mobile Aware Protocols/ Middleware

There is a wealth of research into mobile aware protocols and middleware to support communication over unreliable wireless networks. The work described in this workshop paper applies a number of the techniques from this field to enhance OGSI's support for mobility of services and clients. Whilst a complete review of work in this field is beyond the scope of this paper, a brief overview of some of the key work is given below.

Several approaches, such as I-TCP [I-TCP] and M-TCP [M-TCP] propose alterations to TCP to support communication over wireless networks whilst maintaining interoperability with existing TCP implementations. This is achieved by splitting an end-to-end TCP connection into wired and wireless network hops, connected by some transport layer intermediary. This allows specialist support for mobile networks to be provided over the wireless hop, whilst maintaining normal operation over the fixed network. Rover Queued RPC [Q-RPC] supports mobility through non-blocking RPC calls. ALICE [ALICE] provides mobility support for object oriented middleware, again by allowing mobile hosts to connect via an intermediary, the mobile gateway. ALICE also provides support for disconnected operation through caching and application feedback, such as notification of network disconnection.

2.3 Grid Based Medical Devices for Everyday Health

Equator e-Science program in collaboration with the Medical Imaging and Analysis (MIAS) IRC have explored the extent to which the grid can be used to support interactions with a wearable medical sensing devices for everyday health monitoring. Attached to the wearable medical device are several sensor taking readings such as heart rate, blood oxygen saturation (SPO2), geographical location (GPS), etc. A wireless communication technology, e.g. Wavelan, is used to communicate sensor readings to interested parties on a fixed network.

The project has produced a standard WSDL interface and associated software toolkit (the *MIAS-Equator* toolkit) which provides a generic mechanism for exposing mobile/ remote sensing devices as Grid services. The toolkit has adopted a proxy based approach: a *Device Proxy Service* (DPS) represents the sensing device; a *Sensor Proxy Service* (SPS) represents an individual sensors attached to the sensing device. Each SPS contains SD to store the value of the latest measurement ID (an incremental counter) and a queue of measurement ID/ value pairs. Every X^{th} (e.g. 10^{th}) new measurement the SPS delivers a notification containing the current measurement ID to any interested parties. DPSs and SPSs are deployed on intermediary machines available on the fixed network. The mobile sensing device runs a lightweight soap client using gsoap [gsoap] to make calls to the appropriate sensor proxy when a new sensor reading has been taken.

The project identified a number of limitations associated with the deployment of mobile/ remote sensing applications on the grid.

- *Overhead incurred using OGSI*

The overhead incurred by using OGSI compliant grid middleware (i.e. specific service creation and handle resolution steps) to support intermittently available clients and service initially seems large. However, these overheads become less significant for long lived applications such the wearable medical device which may be deployed for days, weeks or months. In many circumstances the benefits gained from using OGSI as a standard interface technology outweigh the additional overhead incurred.

- *Deployment of heavyweight OGSI grid services on lightweight devices*

Mobile devices are typically more limited in capability than their fixed counterparts, making deployment of large distributed applications difficult. For this reason the Grid Based Medical Devices for Everyday Health project chose to deploy the grid service proxy on a fixed network machine. However, the Java Community Process has defined a Java 2 Platform Micro Edition (J2ME) profile option for Web Service access on small devices [J2ME], and IBM's Web Services Toolkit for Mobile Devices (WSTKMD) [WSTKMD] supports both this and the deployment of web services on small mobile devices such as PDAs and Mobile phones.

Deploying grid services on the mobile device allows the device to be self-contained and requires no special purpose proxy support from the fixed network, allowing the device to roam from network to network more easily.

- *Intermittent connectivity exhibited by wireless networks*

OGSI's inability to cope with intermittent network non-availability was identified as a critical problem to the support of mobile/ remote applications on the grid. There is no explicit management of network disconnection in the MIAS-Equator toolkit. Network availability between the device and proxy is assumed. Loss of network connectivity will cause failure to deliver new readings and therefore loss of data. The toolkit can be seen to support continued operation in the face of disconnections because the proxy service is deployed on a permanently available fixed network machine. However, the data returned by the proxy to the client may not be the latest sensor readings. Further, the client has no mechanism for ascertaining if the readings collected are the latest readings.

The rest of this paper describes a framework which extends OGSI to support the deployment of clients and/ or services on devices which are subject to intermittent network connectivity. The paper also describes a test scenario in which the MIAS-Equator toolkit has been modified to make use of the framework.

3 Intermittent Availability Support Framework

3.1 Exploring the problem

Using the default OGSI communication of synchronous RPC via SOAP over HTTP over TCP, a period of disconnection between client and service in the order of minutes will cause the underlying TCP connection to timeout. This will cause the HTTP operation and therefore SOAP RPC to fail. If the disconnection occurs whilst the request is being transferred from client to service, the operation will not be performed. If the disconnection occurs whilst the operation is being processed by the service or whilst the response is being returned to the client, the operation will have been performed. When an invocation fails, the client is unable to ascertain if the operation was successfully performed; therefore simple retransmission of an invocation may cause duplicate updates to occur.

3.2 Requirements

The primary goal of this framework is to allow grid service invocations to succeed in the face of intermittent network non-availability, where disconnections may last minutes, hours or longer. The framework also aims to support interaction with existing unmodified grid clients and service by maintaining standard grid service invocation semantics. The framework should also:

- Require minimum modification of existing grid clients and service to make use of the framework
- Be scalable to large numbers of clients and services
- Exhibit a minimal performance overhead.

3.3 Approach

The approach taken in this workshop paper introduces an application layer (SOAP) intermediary at the wired-wireless network boundary, splitting the end-to-end invocation into wired and wireless hops. This avoids lengthy end-to-end retries of invocations that fail over the final intermittently available network hop, which is particularly important due to the wide area nature of grid computing. Further, splitting the invocation allows wireless specific protocols to be used over the wireless hop to, for example, perform reliable delivery or optimise wireless network usage. The latter is particularly important if the network usage is charged per Kb, e.g. GPRS. For example Mobile-IP could be used to support roaming of the client or service whilst I-TCP could be used to improve performance over the wireless hop.

Using the standard grid approach, a GSH is resolved to a GSR by contacting the *home handle resolver service* which typically resides in the grid service's service container. Any attempt to perform handle resolution for a temporarily unavailable grid service will therefore fail. In the worst case, the client will interpret this as indication that the service no longer exists. Therefore, this framework introduces and third party handle resolver service which is permanently available on a fixed network machine.

In OGSI, a client determines how to communicate with a grid service by examining the appropriate GSR. When using this framework, a service publishes multiple GSRs: the normal GSR and one or more Mobile-GSRs (M-GSR). Information inside the M-GSR is mapped into the header of the SOAP RPC request message and used to direct the invocation to the appropriate SOAP intermediary (SI). The SI may then be responsible for routing the invocation to the appropriate grid service end point. Alternatively, the M-GSR may contain source routing information which informs the SI how to appropriately forward the invocation. The M-GSR may optionally contain per hop protocol information to specify or suggest which transport protocols should be used per hop, e.g. HTTP/I-TCP for wireless hops and HTTP/TCP for wired hops. Further, expected service availability patterns could be encoded into the M-GSR to, e.g. aid the selection of reliable messaging retry periods. Figure 1 shows the system architecture for a framework supported invocation from a fixed client to a wirelessly connect service; the M-GSR is depicted with embedded source routing and per hop protocol information.

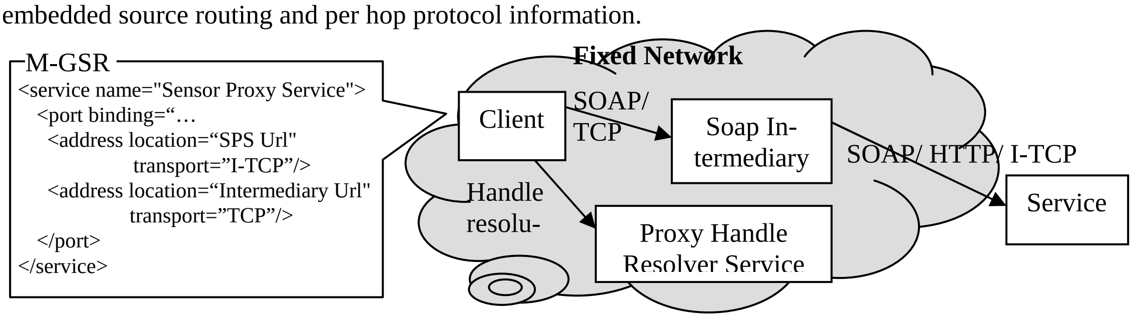


Figure 1, Fixed network grid client invoking a wirelessly connected grid service

4 Testing

4.1 Setup

The MIAS-Equator toolkit has been modified to utilise the framework described. The framework functionality is simply inherited from a super class which extends and replaces the current grid service implementation. A test scenario has been constructed in which a client application collects measurements and alters the sample rate, in real-time, of a wearable medical device; the device is worn by a 'patient' as they carry out a normal days work. The DPS and SPSs are deployed on the wearable device. Three wireless access points (APs) are deployed in locations commonly used by the patient; gaps in the wireless network coverage are present. The locations used by the patient are: (1) office (covered by AP1); (2) laboratory (covered by AP 2 & 3); (3) coffee shop (no coverage); (4) corridors (coverage is incomplete, variable and signal strength is often weak). The patient's movements are recorded by logging when the patient arrives at or leaves locations 1, 2 or 3. Wireless network signal strength for APs 1, 2 & 3 is recorded. The number of readings measured by the device and collected by the client and the number of sample rate changes initiated by the client and received by the device is recorded. The test is performed using both the framework supported and non-framework supported MIAS-Equator toolkit, for comparison. Figure 2 shows how the test scenario has been configured. For the purpose of the framework supported test, reliability is provided using SOAP layer retries.

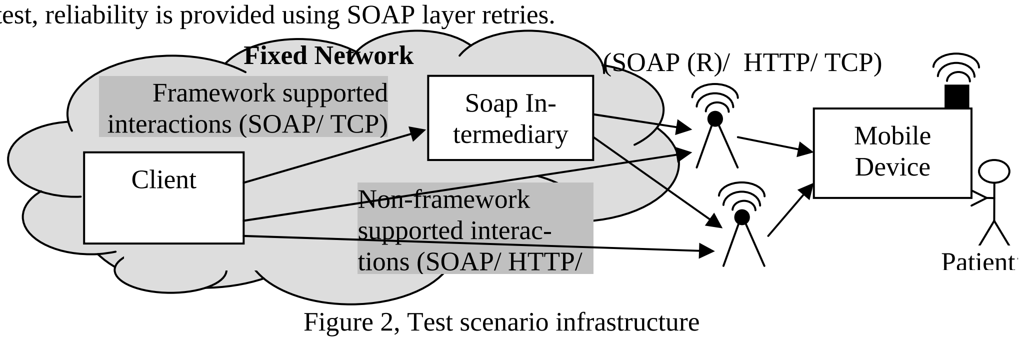


Figure 2, Test scenario infrastructure

4.2 Results

For both tests, the patient left their office and walked to the coffee shop where they purchased a coffee and croissant; walked from the coffee shop to the lab, where they ate their croissant and began drink their coffee whilst talking to a colleague; left the lab began walking back to their office, pausing in the corridor to talk to another colleague and drink more coffee; arrived back in their office.

Table 1 shows, for both framework and non-framework test: the number of readings measured by the device and collected by the client; the number of sample rate updates initiated by the client and received by the device.

	Framework supported		Non-framework supported	
	Client	Device	Client	Device
Readings (measured/ collected)	763	763	581	581
Sample rate updates (initiated/ received)	30	30	46	27

Table 1, results of framework and non-framework supported tests

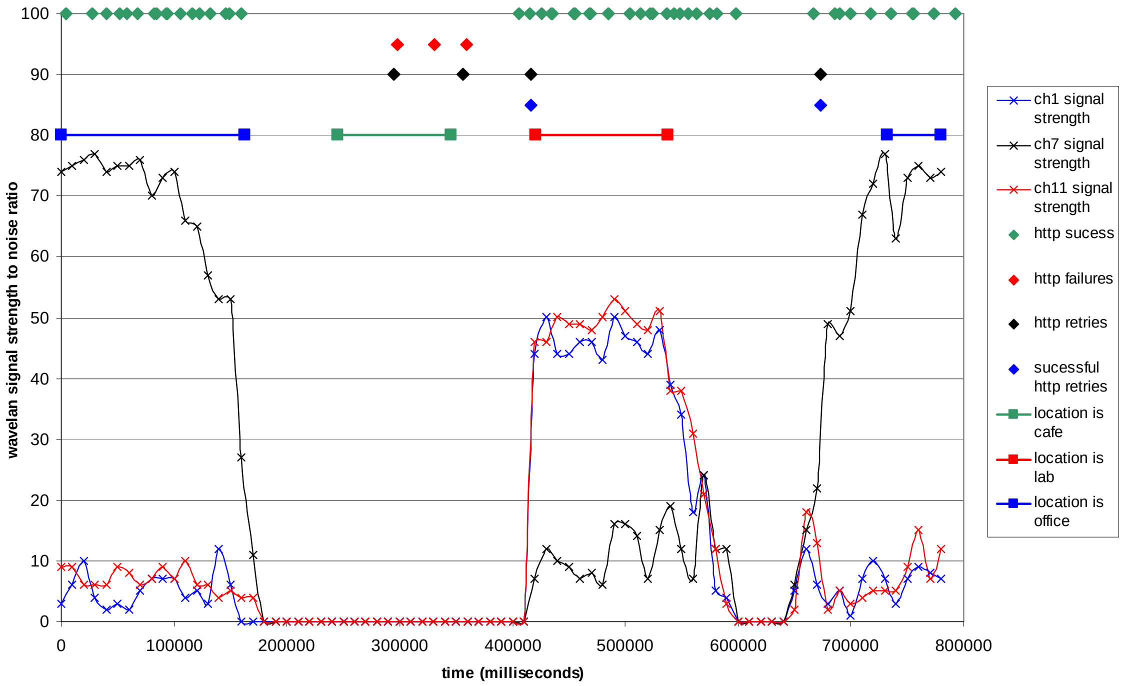
Table 2 shows, for both framework and non-framework supported tests, the number of http success, failures, retries and successful retries.

	Frame-	Non-framework

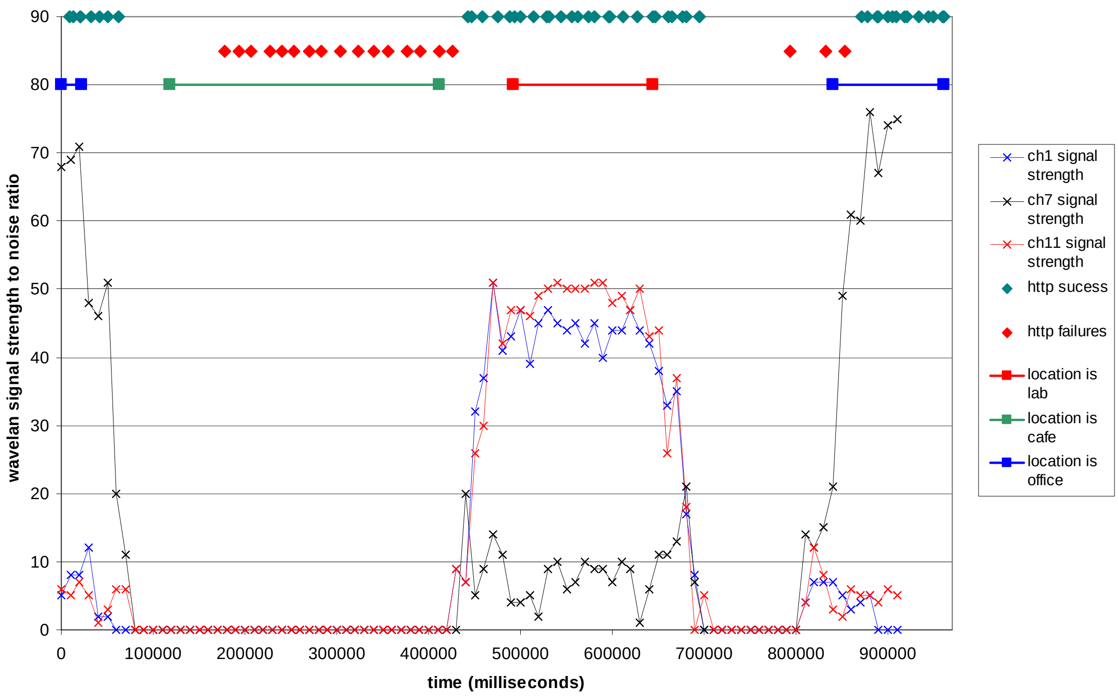
	work	
Successful http invocations	52	49
Failed http invocations	3	19
Http retries	4	0
Successful http retries	2	0

Table 2, http invocation statistics for framework and non-framework supported tests

Graphs 1 and 2 show wavelan signal strength to noise ratio against time (in milliseconds) for the three wireless APs used in the test. The graphs also shows the time of entry to and departure from each location; between locations, the patient is using the corridors. The time of successful http invocations, http failures, http retries and successful http retries are shown. Graph 1 shows the framework supported test. Graph 2 shows the non-framework supported test.



Graph 1, results of framework supported test



Graph 2, results of non-framework supported test

The results show that for both framework supported and non-framework supported tests, all measurements were successfully collected by the client application. This is because measurement collection involves a simple query operation; data lost in failed invocations is collected in subsequent invocation. However, the results shows that whilst all framework supported update sample rate invocations were successfully received by the SPS, only 59% of non-framework supported update sample rate invocations were received by the SPS. The graphs show framework supported invocations recovering from temporary network non-availability.

5 Conclusion

This workshop paper has shown that current grid technologies do not provide adequate support for intermittently connected clients and services. An application which benefits from the use of grid technologies to support its operation (the wearable medical device) has been described. It has been shown that the current grid infrastructure used by the wearable medical device does not support the intermittently available nature of wireless networks. This paper has described a framework which extends OGSI to support clients and services which are deployed on mobile and therefore intermittently available devices. Initial tests have shown the framework to successfully support the operation of the wearable medical device despite intermittent network non-availability. The work in this paper has extended the grid user community to an additional application domain and has demonstrated the viability of utilising grid technologies to support mobile applications.

References

[Medical] Carl Barratt, et al. Extending the Grid to Support Remote Medical Monitoring. UK e-Science All Hands Meeting 2003.

[Antarctic] Steve Benford, et al. e-Science from the Antarctic to the Grid. UK e-Science All Hands Meeting 2003.

[OGSI] S. Tuecke, et. al.. Open Grid Services Infrastructure (OGSI) Version 1.0. Global Grid Forum Draft Recommendation, 27 June 2003.

[WSA] David Booth, et. al. Web Services Architecture. W3C Working Group Note, 11 Feb 2004.

[I-TCP] Ajay Bakre, B. R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. Proceedings of the 15th International Conference on Distributed Computing Systems (ICDCS'95), Page: 136, Year of Publication: 1995, ISSN:1063-6927

[M-TCP] Kevin Brown, Suresh Singh, M-TCP: TCP for Mobile Cellular Networks, 29 July 1997.

[Q-RPC] Anthony D. Joseph, et. al. Rover: A Toolkit for Mobile Information Access. Proceedings of the fifteenth ACM symposium on Operating systems principles, Copper Mountain, Colorado, United States, Pages: 156 – 171, Year of Publication: 1995, ISBN:0-89791-715-4

[ALICE] Mads Haahr, Raymond Cunningham, Vinny Cahill. Towards a Generic Architecture for Mobile Object-Oriented Applications. Proceedings of the 2000 IEEE Workshop on Service Portability and Virtual Customer Environments, pp 91-96, December 2000.

[J2ME] Java Community Process, JSR-000172 J2ME™ Web Services Specification, 2003, http://www.jcp.org/aboutJava/communityprocess/review/jsr172/ (verified 4 Nov 2004)

[WSTKMD] Web Services Toolkit for Mobile Devices, http://www.alphaworks.ibm.com/tech/wstkmd, (verified 2 Oct 2004).