

A Scalable Registry Service for jGMA

Mark Baker and Matthew Grove

Distributed Systems Group, University of Portsmouth, UK
[mark.baker@computer.org, matthew.grove@port.ac.uk]

Abstract

jGMA is a pure Java reference implementation of the GGF's Grid Monitoring Architecture, which identifies the core characteristics required for a scalable grid-based monitoring infrastructure. In the GMA, producers or consumers can publish their existence in a directory service (registry). This paper focuses on the development of jGMA's Virtual Registry service which resides within the Mediator component. The Virtual Registry will provide consumers and producers with a scalable and robust mechanism to publish and discover each other over the wide area.

1. Introduction

jGMA [1] is a pure Java reference implementation of the GGF's GMA [2], which represents the characteristics required for a scalable grid-based monitoring infrastructure. In this model, producers or consumers that accept connections publish their existence in a directory service (registry). Producers and consumers can then both use the registry to locate parties, which will act as a source or destination for the data they are interested in.

In this paper we briefly introduce the jGMA architecture and outline the requirements of the Virtual Registry (VR) component. We then introduce an initial implementation of the VR service using Internet Relay Chat (IRC). Finally we conclude the paper and outline our future work.

Our motivation for developing jGMA, is described elsewhere [3], but can be summarised by saying some existing systems were embedded into larger software packages, such as the MDS in the Globus Toolkit, and Network Weather Service, and were deemed difficult to breakdown into a standalone version needed for our purpose. R-GMA and pyGMA, two standalone GMA implementations were considered, but both were less than ideal for our purpose; the drawbacks are discussed elsewhere.

2. The jGMA Architecture

Our reference implementation of GMA has number of idealised features, including complying with the GMA specification, capable of scaling from LAN to WAN proportions, and across thousands of sites with millions of

producers/consumers, supporting both non-blocking and blocking events, capable of taking advantage security components such as TLS and/or the GSI. In addition, we wanted the actual implementation to have a small and well-defined API, a minimal number of installation dependencies, be easy to install and configure, have a minimal impact on its hosts, good performance and be capable of working through firewalls.

jGMA consists of three entities:

- Mediators that permit producers and consumers to discover each other and allow remote communications,
- Consumers,
- Producers.

In jGMA, producers or consumers publish their existence in a directory service (registry). In turn, producers and consumers can use the registry to locate other parties, which act as a source or destination for the data they are interested in. Currently jGMA uses TCP Sockets for LAN communications and HTTP over the WAN. The Mediator allows wide-area connectivity for nodes that do not have direct access to the Internet; it acts effectively as a gateway into a localised jGMA installation. Figure 1 shows the registry components within the jGMA architecture.

3. The jGMA Virtual Registry

A simple volatile data store was initially implemented in jGMA; this only stores information about clients that are connected directly to the Mediator. The new Virtual Registry (VR) component within the Mediator needs to provide discovery, naming, and querying services for jGMA clients. VRs within Mediators require a mechanism to discover an

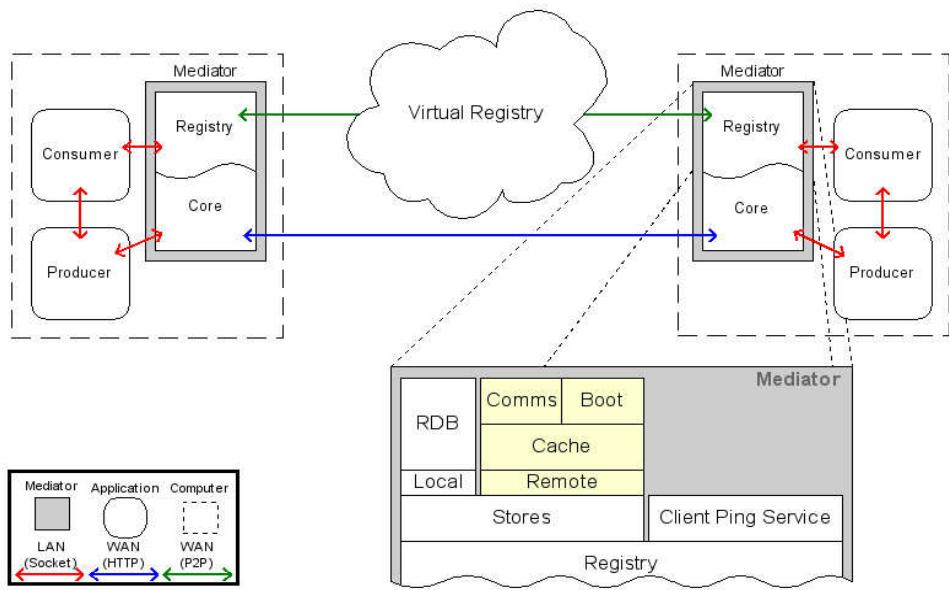


Figure 1: The jGMA architecture - highlighting the virtual registry components

efficiently communicate in order to form a virtualised registry.

Virtual Registry requirements:

- Be scalable,
- Store sufficient information to be GMA compliant,
- Be secure, and prevent unauthorised access to the data,
- Need minimal configuration,
- Ideally have no single point of failure,
- Be robust and tolerant of network failure,
- Efficient query routing between individual registries.

The VR architecture provides two pluggable layers; one allows the use of different implementations of local data storage, for example using text files, a relational database such as MySQL or an XML database such as Xindice. The second layer allows different protocols to be used for communications between components in the VR; we are exploring the use of Peer-to-Peer (P2P) technologies to provide the distributed database.

3.1 Virtual Registry Design

The VR provides three core services: discovery, querying, and caching. It should be noted that not all P2P systems provide all three of these services, so some generic jGMA services are needed to provide the missing functionality.

Implementing these services as separate modules makes it possible to try different implementations of a service without having to redesign the whole system. A feature of this approach is interoperability with other systems, should a standard protocol be adopted for monitoring information on the Grid, it would be possible to connect jGMA without redesign or significant refactoring.

1. Discovery: This is the process by which the registry component within a Mediator locates other registries to form the VR. If this mechanism relies on centralised services, the greater the likelihood of a failure in the discovery process. For example, one solution is to contact a well-known server to get a list of existing VRs; obviously this is not a scalable or fault tolerant approach. We require a more dynamic discovery process, which does not rely on so called hardwired addresses to work. We are currently assessing the suitability of existing P2P communications protocols for providing Mediator discovery and inter-VR communications (Section 3.2).

2. Distributed Queries: Once the VR has been joined, queries can be despatched and routed through the VR infrastructure. The VR network topology must be self-healing, by this we mean that should one or more of the registries fail; it should have a minimal affect on the overall VR. The VR network topology should aim to reduce the number of hops required to traverse the

network in order to minimise the time required to propagate a query. The design of the VR infrastructure will allow us to experiment with different network topologies and P2P protocols, such as Gnutella [4].

3. Caching: Using a cached registry query reduces the number of queries sent over VR. A cache hit will allow the VR to respond immediately to the query rather than having to wait for a remote query to complete. Caching also provides some fault tolerance during transient faults with either the WAN connectivity or remote registries. Issues of data consistency will have to be dealt with at this level too.

3.2 VR IRC Service

The IRC VR service Figure 2 is an attempt at meeting the criteria of the jGMA VR by building a system which uses the mature Internet Relay Chat protocol. IRC was chosen as it provides a set of services closely matching the needs of jGMA's VR. IRC networks have servers connected via a graph topology, which attempt to efficiently route messages and provide fault tolerance; some IRC networks manage thousands of users and can provide fault tolerant boot-strapping via DNS.

Using combinations of existing IRC features can provide a range of security levels:

- Leveraging functionality built into the IRC protocol for example locking channels, and private messages,
- Using the features of the IRC daemons such as encrypted routing for IRC messages,
- Running your own IRC daemons rather than using a public network.

Each Mediator has an "IRC Bot" which connects to a chosen IRC network. The steps the Bot goes through to join and query other jGMA Mediators are:

1. The Bot joins a channel and announces itself to other Bots,
2. The Bot publishes search queries to the IRC channel,

3. Other Bots receive the queries and pass them onto the IRC registry backend,
4. The Bots reply to the queries via a private message over IRC.

All messages are routed via the IRC network.

Currently IRC channels are used to partition the VR, we need to explore methods of reducing the impact of a 'flood query' on the network such as using IRC channels to create a virtual topology (an overlay network).

4. Summary and Conclusions

In this paper we have described the requirements for a distributed virtual registry for jGMA. We have outlined the design for a pluggable registry framework for jGMA, which will allow us to explore how best to leverage existing P2P technologies to create a scalable, robust Virtual Registry. We have introduced an embryonic implementation of the VR using the IRC network. We currently support SQL as a query language and LDIF [5] as a response mark up. The VR has a layer of abstraction, which translates queries and responses into an intermediate format internally; this allows us to add support for different query languages and response mark ups, such as XPATH and RDF, to permit interoperability with other GMA implementations in the future.

In order to demonstrate the capabilities of the jGMA framework we intend to develop a library for online distributed gaming, which has become increasingly popular with the widespread uptake of broadband. Games publishers have each tried to provide an infrastructure to support their online games. We believe there is an opportunity to develop standard services, based on a jGMA, to support these games.

An early binary version of jGMA [6] is currently available for developers interested in investigating and further enhancing its capabilities. jGMA has been extensively benchmarked throughout its development, the latest results are presented in [7].

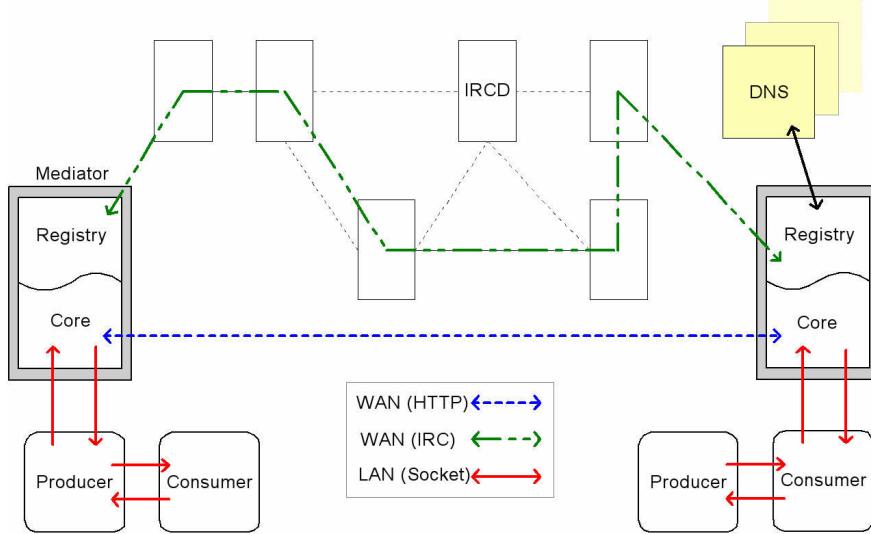


Figure 2: The jGMA IRC Virtual Registry Service

References

- [1] jGMA,
<http://dsg.port.ac.uk/projects/jGMA/>
- [2] GMA,
<http://www.ggf.org/documents/GFD/GFD-I.7.pdf>
- [3] jGMA: A lightweight implementation of the Grid Monitoring Architecture, M.A. Baker and Matthew Grove, UK e-Science All Hands Meeting, September 01-03 2004
- [4] Gnutella, <http://www.gnutella.com/>
- [5] RFC 2849 - The LDAP Data Interchange Format (LDIF) - Technical Specification, Gordon Good, <http://www.faqs.org/rfcs/rfc2849.html>, June 2000
- [6] jGMA Download,
<http://dsg.port.ac.uk/projects/jGMA/software/>
- [7] M.A. Baker, M. Grove and R. Lakhoo, A Preliminary Performance Evaluation of jGMA With the NaradaBrokering Framework, submitted to Grid 2005 - 6th IEEE/ACM International Workshop on Grid Computing, http://dsg.port.ac.uk/~mjeg/jGMA/jgma_pprint_sc-grid2005.pdf, June 2005