A Web Services based system for data grid

Irene Pompili[1], Claudio Zunino[1], Andrea Sanna[1], and Giacomo Piccinelli[2]

[1] Department of Computer Science
Politecnico di Torino
C.so Duca Abruzzi, Torino I-10129, Italy
irenella@tin.it,
{claudio.zunino,andrea.sanna}@polito.it

[2] Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK
g.piccinelli@cs.ucl.ac.uk

Abstract. In this paper an architecture for a data brokerage service will be proposed. The brokerage service is a part of the system that is being implemented within the European Grid for Solar Observations (EGSO) to provide a high-performance infrastructure for solar applications. A broker interacts with providers and consumers in order to build a profile of both parties. In particular, the broker interacts with providers in order to gather information on the data potentially available to consumers, and with the consumers in order to identify the set of providers that are most likely to satisfy specific data needs.

Introduction

Nowadays, Web Services are especially known as a way to improve business systems; in this paper, they will be exploited for the implementation of data grid services. The main aim of the proposed brokerage architecture is to collect information from providers and allow users to search data over a grid. For this purpose, the broker receives from providers a meta-catalogue that contains coarse granularity information. Brokers also act as access points for EGSO and allow data searches in the grid.

Brokers offer Web Services interfaces to consumers (i.c. the users) and data providers. In particular, brokers supply a mechanism to allow consumers to perform data searches, select the providers that can satisfy a specific request and forward the query. Finally, brokers collect query results and send them back to the consumer.

The content of this paper is organized as follows: in Section Background a brief background on Web Services is presented; while Section Framework describes in detail the proposed architecture. Conclusive remarks and future work are presented in Section Conclusion.

Background

Over the past few years, applications have interacted using ad hoc approaches. At the present moment, Web Services[1][2] are emerging as a framework for application-application interaction, based on existing Web protocols and based on open XML standards.

Web Services are essentially rely upon three technologies: Web Services Description Language (WSDL)[3]; Universal Description, Discovery and Integration (UDDI)[4]; and Simple Object Access Protocol (SOAP)[5].

WSDL is a specific XML format that can be used to describe Web Services interfaces. A WSDL specification provides a description of the service and the specific protocol that users have to follow to access the service itself.

On the contrary, UDDI is an industry-standard centralized directory service that can be used to advertise and locate Web services. UDDI allows users to search for Web services using various search criteria, including company name, category, and type of Web service.

Finally, SOAP is a protocol for exchanging XML data and provides the basic mechanism for Web Services communication. It can uses a textual format, as opposed to binary formats such as in CORBA[6] or Java RMI[7].

Various examples of Web services-based architectures can be found in literature. For instance, in [8] a Web Services-based system was proposed to integrate ad-hoc mobile applications with the Bluetooth and Wi-Fi technologies.

Web Services have also been used in [9] for the implementation of a biomedical portal. The proposed architecture consists of a grid portal for the management of biomedical images in a distributed environment.

The framework

The core of the proposed architecture is constituted by the broker, which offers a set of Web Services to both consumers and providers. The main broker interfaces are the connection, the provider data update and the data search interface. Core component of the proposed system is a two layer search engine. Each provider has a catalogue and periodically sends updates to a broker. The broker receives them and generates a version of this information which is inserted in this in the local database. This summarized catalogue (referred to as meta-catalogue) is obtained by using a set of parameters (e-y time, wavelength, spatial coordinate positions), in order to collapse sets of rows in the original catalogue to a limited number of rows in the meta-catalogue. For instance, if in the original provider catalogue ten rows are used to describe data related to a specific day, and the granularity of the time parameter is set to one day, then, the meta-catalogue will contain just one row. Therefore, the meta-catalogue allows the broker to immediately discard providers that certainly need not to be searched for a resource given, but cannot tell the broker if a provider actually posses a given resource. For instance, if a consumer searches data related to a specific hour, and the time parameter is used to generate the meta-catalogue with granularity equal to one day, then the broker by querying the meta-catalogue, can immediately

identify which providers can potentially satisfy the query; however, the selected providers will still need to be directly interrogated. In order to perform searches efficiently the meta-catalogue on the broker has to be constantly updated.
Moreover, after receiving an update from a provider, brokers have to propagate this information to other brokers in the network. To this aim, the following procedure is used:

Procedure propagate_updates	
B1→B2 (all neighboring brokers, duplicates are discarded)	updates in XML format + time_stamp

The time stamp is used in order to allow brokers to select only new records.
In Fig. 1 a search session is shown. The network is composed by the broker, a consumer and three providers. The session starts when a consumer submits a query. The search session consists of the following phases:

1) The consumer submits a query to the broker.
2) The broker search engine queries the local database to obtain the list of potential providers that can answer the consumer's query. The local database contains the summarized version of catalogue (i.e. the meta-catalogue).

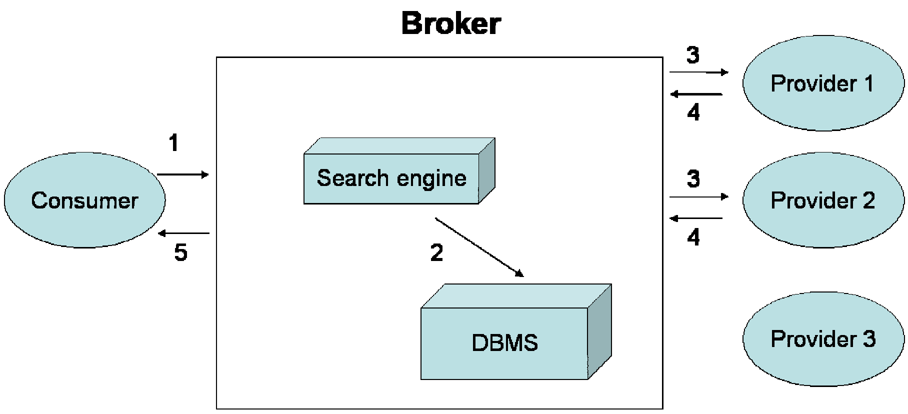

Fig. 2. A typical search session.

3) The broker forwards the query to all the providers that may satisfy the query (in this case the Provider 3 is excluded).
4) Each provider sends back results to the broker.
5) The broker collects and sends to the consumer the results obtained by each provider.

It has to be noticed that each consumer query is managed by a specific thread; in this way, the consumer, is not blocked. Only after that all results have been collected, the broker sends them to the consumer.
Finally, the consumer directly contacts the provider to retrieve the data.

The interface of the query procedure is the following:

Procedure query	
C→B	Query
B→C	The broker returns a query_ID
B→C	The broker sends to the consumer, in asynchronous mode, query results in XML

Conclusion and future work

In this paper a brokerage architecture dedicated to information retrieval and metadata management is proposed. Metadata mainly consist of catalogues of solar data that are produced and maintained by various providers. A local database in the broker allows a faster search.

The proposed architecture can be possibly improved by using a distributed version of the broker meta-catalogue. In the case of a distributed meta-catalogue, the propagate_updates procedure still needs to be used since the information has to be replicated on a certain number of brokers selected by an ad-hoc algorithm to ensure fault-tolerance. Moreover, a new procedure to propagate consumer queries to other brokers has to be included. In fact, a broker is no longer able to immediately identify the providers that can possess requested data if the meta-catalogue is distributed among brokers. A broker will thus propagate queries to neighbouring brokers to receive information about the meta-catalogue. The broker that starts will receive results from the other brokers in XML and merge them in order to select a set of providers to be interrogated, as in the current architecture.

References

[1] Curbera, F., Duftler, M., Khalaf, R., Mukhi, N., Nagy, W., Weerawarana, S. “Unraveling the Web Services Web - An Introduction to SOAP, WSDL, and UDDI”, IEEE Internet Computing, Vol.6 Issue 2, March-April 2002, pp.86-93.

[2] Roy, J. Ramanujan, A. “Understanding Web services”, IT Professional,Vol. 3 Issue 6, Nov/Dec 2001, pp. 69-73.

[3] http://www.w3.org/TR/wsdl.html

[4] http://www.uddi.org/

[5] http://www.w3.org/TR/SOAP/

[6] http://www.corba.org/

[7] http://java.sun.com/products/jdk/rmi/

[8] Steele, R., “A web services-based system for ad-hoc mobile application integration”, Proceedings. ITCC 2003. International Conference on , April 28-30, 2003, Page(s): 248-252

[9] Aloisio, G.; Blasi, E.; Cafaro, M.; Fiore, S.; Lezzi, D.; Mirto, M., “Web services for biomedical imaging portal”, Proceedings. ITCC 2003. International Conference on , April 28-30, 2003 Page(s): 432 -436