PROCEEDINGS OF SCIENCE

Software for Distributed Systems - The EMI Product Portfolio

Morris Riedel[1]
Forschungszentrum Juelich
Juelich Supercomputing Centre
Juelich, Germany
E-mail: m.riedel@fz-juelich.de

Jedrzej Rybicki
Forschungszentrum Juelich
Juelich Supercomputing Centre
Juelich, Germany
E-mail: j.rybicki@fz-juelich.de

Alberto Di Meglio
European Organization for Nuclear Research (CERN)
Geneva, Switzerland
E-mail: Alberto.Di.Meglio@cern.ch

The European Middleware Initiative (EMI) brings together ARC, dCache, gLite, and UNICORE to provide a harmonised set of products and streamlined releases to the DCI community. While there are many technical solutions around, EMI is one of the key players in providing software for large-scale distributed systems that are operated around the world today. Having products and solutions in various technical areas such as compute, data, information, and security, it is interesting to understand that these products also implement many of the principles and paradigms of distributed systems. This contribution will provide an overview of the EMI product portfolio focusing on its key features and their role in distributed systems based on comparisons with known literature such as books offered by Tanenbaum.

EGI Community Forum 2012 / EMI Second Technical Conference,
Munich, Germany
26-30 March, 2012

[1] Speaker

PoS(EGICF12-EMITC2)082

1. Introduction

The European Middleware Initiative (EMI) project [7] provides an integrated set of middleware products that consists of components of ARC, dCache, gLite, and UNICORE. Its releases are specifically optimized to provide solutions for High Throughput Computing (HTC)-oriented distributed computing infrastructures (DCIs) such as the European Grid Infrastructure (EGI) [8] or infrastructures like the Partnership for Advanced Computing in Europe (PRACE) [9] driven by High Performance Computing (HPC) needs. The purpose of these infrastructures and inherently its technology providers such as EMI is to support scientific applications running on the geographically dispersed infrastructure resources. From a more general perspective those infrastructures form pan-European '*distributed systems'* facing many challenges as outlined by Tanenbaum in [1] (e.g. scalability, accessibility, availability, etc.).

The EMI project offers many solutions for problems arising in distributed systems. The majority of EMI products follow the general design approach of Service Oriented Architectures (SOAs) [1] and their implementations using Web service message exchange communication. In many of these implementations traditional Remote Procedure Calls (RPCs) [1] based on proprietary protocols of technologies (e.g. CORBA [1]) have been exchanged with the Extensible Markup Language (XML) thus creating an XML-based RPC communication method. In this sense EMI provides products that are driven by scientific needs, but that nevertheless adopts and drives open standards, best practices, and known paradigms also used in industry and commercial infrastructures.

The fundamental goal of this paper is to bridge the gap between rather theoretical design principles and practical implementation of them. Throughout this contribution, the EMI products are surveyed in the view of how they adopt general distributed system principles in the four technical areas of compute, data, information, and security. Although many products implement more than one key design principle, only one is highlighted in context of the corresponding EMI product. We refer to the EMI Web site [7] for the details of each single EMI product that offer overvier factsheets, documentation, and installation hints for each individual EMI product. This paper aims to provide thus evidence that the powerful EMI product portfolio offers a broad set of practical 'ready-to-run' solutions for a wide variety of distributed system problems.

This paper is structured as follows. Section one introduces the given problem domain generally known as distributed systems with a particular focus on DCIs. Section 2 provides insights into adopted principles in the area of 'distributed systems processing' while Section 3 gives insights into adopted paradigms in 'data in distributed systems'. Section 4 then describes how the EMI product portfolio provides scalable solutions in the 'distributed information systems' area. The complex but required 'distributed systems security' area offers insights how EMI solutions combine their strength to achieve an attribute-based authorization. Related work is briefly surveyed in Section 6 and this paper ends with some concluding remarks.

2. Distributed Systems Processing

One of the key technical areas of the EMI project deals with 'computing' that can be more in detail described as 'distributed systems processing'.

The *ARC Computing Element (CE)* is used to submit and manage a wide range of scientific applications running on computational resources available in DCIs. It is a light-weight system to execute applications across geographically distributed computing services and their underlying resources. Its architecture reveals one of the key distributed systems principles adopted within the EMI product portfolio that Tanenbaum names as **'communication'**. *'Where HTTP is the standard communication protocol for traditional Web-based distributed systems, the Simple Object Access Protocol (SOAP) forms the standard for communication with Web services'* [1]. The server-side of the ARC CE offers access through its hosting environment via the SOAP protocol using Web services message exchanges. By using the basic principle of communication, the ARC CE is generally interoperable with other services of the EMI product portfolio.

Another product of the EMI portfolio is the *CREAM CE* that is also used to submit and manage applications running on DCI resources. This product is a powerful system to execute applications across geographically distributed computing service and uses the aforementioned SOAP protocol and Web services message exchanges. In addition, the CREAM CE offers many hooks for accounting and brokering as well as data-staging functionality. The CREAM CE architecture illustrates nicely the general distributed systems **'design of a concurrent server'**. A job submission server like the CREAM CE waits for an incoming job request from a client and subsequently ensures that the request is processed and then waits for the next incoming request. *'A concurrent server does not handle the request itself, but passes it to a separate thread or another process, after which it immediately waits for the next incoming request'* [1]. The design principle of this concurrent server is adopted by the CREAM CE and other EMI products in order to ensure 'high throughput' of service requests and scalability necessary within pan-European DCIs.

The third remarkable EMI product of this area is *UNICORE* that is driven by High Performance Computing (HPC) needs being often used to submit and manage applications optimized for large-scale HPC resources. Users take advantage of its three-tier architecture that implements the functionality of a computing element but also workflows. A key benefit for resource providers is its strong support for a wide variety of available batch systems. Its design incorporates the general distributed system principle of an **'application-level gateway'** in the context of firewalls. *'...the other type of firewalls is an application-level gateway. In contrast to a packet-filtering gateway, which inspects only the header of network packets, this type of firewall actually inspects the content of an incoming or outgoing message'* [1]. UNICORE is an EMI solution that is specifically optimized for sensitive environments (e.g. HPC environments) that have less impact on site security policies. The UNICORE Gateway is only one small component of the UNICORE product that authenticates job requested, but more notably ensures that only one port is open to the public to access numerous HPC resources at a site. The

application-level functionality of this component is thus the mapping from virtual URLs to real URLs in order to route job requests to available configurable sites.

3. Data in Distributed Systems

Another technical area of the EMI project is summarized under the term 'data' meaning the 'storage and management of data within distributed systems' such as pan-European DCIs.

The *dCache Storage Element (SE)* provides functionality to store scientific data and to transparently access disk-based storage systems as well as tertiary storage (e.g. tapes) known for better cost-efficiency. It is thus used to store data in a distributed fashion without end-users being aware where their data is stored. This in turn is characteristic design in distributed systems named by Tanenbaum as **'distribution transparency'**. *'An important goal of a distributed system is to hide the fact that its processes and resources are physically distributed across multiple computers'* [1]. One of the key benefits from this principle is that data can be migrated from one data resource to another without affecting end-users.

Another complementary product of the data area is the *StoRM SE* that is used to store data and information in different underlying disk-based storage systems. It is specifically optimized for (parallel) disk-based storage systems such as the General Parallel File System (GPFS) or Lustre. The open and modular architecture of StoRM decouples it from these different underlying file systems providing end-users with a stable standard-based interface (e.g. Storage Resource Manager (SRM) standard [2]) while the underlying file system might change over time. *'An open distributed system is a system that offers services according to standard rules that describe the syntax and semantics of those services'* [1]. As a consequence, the standard-based interface in StoRM in particular and in EMI in general contributes to the fact that the EMI product portfolio provides solutions for an **'open distributed system'**.

A lightweight EMI storage solution is the *Disk Pool Manager (DPM)* that offers a simple way to create disk-based Grid storage elements and their management. It leverages all the required functionality for a Grid storage solution including support for multiple disk server nodes, different space types or multiple file replicas in disk pools. The architecture of DPM reveals another remarkable principle of the EMI distribution that is the support of a plethora of **'protocols'**. *'The collection of protocols used in a particular system is called a protocol suite or protocol stack'* [1]. As an example in the data area, the aforementioned DPM system supports many protocols for file access such as Remote File Input/Output (RFIO), XROOT, HTTP, GridFTP, NSF4.1, and SRM.

4. Distributed Information Systems

The EMI area 'information' related to several aspects that Tanenbaum introduces as 'distributed information systems' that in itself represent a whole class of distributed systems [1]. Therefore, the EMI Product portfolio provides only a fraction of the functionality still covering the crucial elements to form a geographically dispersed information ecosystem about service and resources available in large-scale DCIs. Firstly, the EMI Registry (EMIR) product provides a high robustness, scalable and performance registry using a federated model. The latter addresses a known limitation of distributed systems by preventing a centralized single entry

point. A REST interface is used to register and query the services that are in turn describes using the GLUE2 information model [3]. The EMIR product is a nice example of how EMI products achieve the design principle of **'scalability',** which is important in distributed systems like pan-European DCIs. *'...scalability is one of the most important design goals for developers of distributed systems... if more users or resources need to be supported, we are often confronted with the limitations of centralized services, data, and algorithms...'* [1]. EMIR also provides functionality for securing this information using Extensible Access Control Markup Language (XACML) [4] policies or simple Access Control Lists (ACLs).

The EMI product portfolio also offers another broadly used product in the information area known as the Berkeley Database Information Index (BDII) product. The product achieves the aforementioned scalability goal by providing a hierarchical approach with several components on different layers. A typical setup is gathering information from resource-level BDIIs and site-level BDIIs up to the top-level BDIIs that in turn reveal the resources and services available on the corresponding DCIs. Also this EMI product uses the GLUE2 information model thus contributing the overall EMI consistent information ecosystem.

5. Distributed Systems Security

One of the most important areas of EMI is 'security' that is specifically crucial for distributed systems such as pan-European DCIs that cross geographical and organizational boundaries. The EMI product portfolio adopts a wide range of different security techniques and principles (e.g. WS-Security Extensions in SOAP headers, etc.) that are best described by A. Belapurkar et al. in [5]. This paper outlines one approach as one specific example of the security area that was named by Tanenbaum as **'protection domains'** in distributed systems. *'One approach is to construct groups of users...related to having groups as protection domains, is also possible to implement protection domains as roles'* [1]. EMI follows this approach with different products that all have their unique key features as part of the approach.

One implementation approach performed by EMI requires a functionality known as Attribute Authority (AA) that is responsible for releasing signed security credentials with information beyond pure identities (roles, groups, project, etc.). The EMI product portfolio consists of two products in this category. The VOMS system is an AA server typically used to obtain signed security credentials with attributes of end-users within HTC-driven infrastructures. It is able to store identities and manage them in hierarchical groups. VOMS offers a complementary client tool in order to easily configure it as well as managing the different security attributes in context of identities (i.e. X.509 identities). The Security Assertion Markup Langauge (SAML) 2.0 [6] interface of VOMS ensures the encoding of security attributes according to a standardized format. More recently, VOMS was augmented with a REST interface. Another AA server of the EMI distribution is UVOS that is traditionally used in infrastructures driven by HPC needs. Being similar to VOMS it also stores identities and other identifiable servers and organize them in hierarchical groups if needed. UVOS is a lightweight product with a VO authentication Web component optimized for a usage within Web browsers. As UVOS also adopts the SAML 2.0 standard it is interoperable with VOMS via SOAP-based Web service message exchanges.

But both aforementioned EMI products only partly satisfies the approach of 'protection domains' as described by Tanenbaum. Another crucial element of this approach is missing that is referred to as 'attribute-based authorization'. Hence, the attributes released from AA servers need to be understood and used during authorization of end-users when they submit a request to computing or data servers in the same protection domain. The wide variety of EMI products has mechanisms in place in each product that are able to interpret the released security attributes from VOMS and UVOS in order to derive authorization decisions. In addition, the EMI product portfolio also offers one dedicated authorization system named as ARGUS that also works with EMI products and is able to interpret the released security attributes. By using the XACML standard, ARGUS enables security policy-based authorization based on the attributes extracted from security credentials. End-users with attributes such as a specific role can be granted access to Grid systems when ARGUS is used. Likewise the access can be denied when a specific project membership is required but not presented in the security attributes by end-users.

Finally, it should be noted that the security attributes and their shipping as part of transport protocols happen transparent to end-users. Although end-users may be aware of their own roles, project/group membership they are not required to understand their encoding as part of SAML attribute statements [6] or XACML-based policy elements [4].

6. Related Work

There is a wide range of other DCI-related technology in the field (e.g. desktop Grids) and also middleware in the sense of Tanenbaum emerges more and more given the easiness of creating such systems with open source tools today. One of the most known software stack that offers products for distributed systems with a particular focus on large-scale DCIs is the Globus Toolkit [10]. Many products of the gLite and ARC middleware rely on the Globus software stack using several components such as those required to build systems compliant with the Grid Security Infrastructure (GSI) [5]. Nevertheless, one of the major work items of EMI is to become independent of GSI in particular and Globus in general throughout the EMI product portfolio. Another prominent software stack that is used together with UNICORE in the Extreme Science and Engineering Discovery Environment (XSEDE) [12] in USA is the GENESIS middleware [11]. GENESIS offers components specifically optimized for HTC using a wide variety of open standard protocols from OGF and OASIS.

In contrast to Globus and GENESIS, the EMI activities on the three different middleware systems ARC, gLite, and UNICORE are specifically harmonized to work in an integrated fashion offering also a streamlined support infrastructure. Hence, EMI does not follow a toolkit approach and unifies the strength of four different middleware products (with dCache) in order to cover a broad range of distributed system areas. The EMI releases are performed following a wide variety of policies that ensure a streamlined software stack satisfying the quality requirements of large-scale DCIs today.

7. Conclusion

This paper surveyed the EMI product portfolio with the perspective on how several individual products implement principles and paradigms known from literate in the distributed

systems field. We can conclude that EMI adopts a broad range of principles such as protection domains, distribution transparency, or concurrency design paradigms. The integrated set of EMI products also complements features from different technology areas such as the principle of 'application-level gateways' and products specifically designed to meet the general distributed systems goal of scalability. The wide variety of supported protocols, and following best practices and standards in terms of communication contribute to the fact that the EMI products are very well suited to create open distributed systems satisfying the needs of large-scale DCIs of science and beyond.

Acknowledgements

This work has been partially funded by the European Commission as part of the EMI (Grant Agreement INFSO-RI-261611) project.

References

[1] A.S. Tanenbaum and M. Van Steen, *Distributed Systems*, *Principles and Paradigms*, Second Edition, 2007, Pearson Education Inc., ISBN 0-13-613553-6

[2] A. Sim and A. Shoshani, *The Storage Resource Manager Interface Specification – Version 2.2*, Open Grid Forum Document Nr. 129, 2008

[3] S. Andreozzi, S. Burke, L. Field, G. Galang, B. Konya, M. Litmaath, P. Millar, and J. Navarro, *GLUE Specification Version 2.0*, Open Grid Forum Document Nr. 147, 2009.

[4] T. Moses, *eXtensible Access Control Markup Language (XACML) – Version 2.0 Core Specification*, Organization for the Advancement of Structured Information Standards (OASIS), 2005

[5] A. Belapurkar, A. Chakrabarti, H. Ponnapalli, N. Varadarajan, S. Padmanabhuni, and S. Sundarrajan, *Distributes Systems Security – Issues, Processes, and Solutions*, John Wiley and Sond Ltd., 2009

[6] S. Cantor, J. Kemp, R. Philpott, and E. Maler, Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML), Organization for the Advancement of Structured Information Standards (OASIS), 2005

[7] EMI Project Web site, Online: http://www.eu-emi.eu (April 2012)

[8] European Grid Infrastructure (EGI), Online: http://www.egi.eu (April 2012)

[9] Partnership for Advanced Computing in Europe (PRACE), Online: http://www.prace-project.eu

[10] I. Foster, *Globus Toolkit Version 4: Software for Service-Oriented Science*, in proceedings of the Sixth IFIP Internaional Conference on Network and Parallel Computing, Beijing, China, pages 213-223, 2005

[11] M. Morgan and S. Grimshaw, GENESIS II – Standards Based Grid Computing, in proceedings of the 7th IEEE/ACM International Symposium on Cluster Computing and Grid 2007 (CCGrid 2007), Rio de Janeiro, Brazil, pages 611-618, 2007

[12] Extreme Science and Engineering Discovery Environment (XSEDE), Online: www.xsede.org