Using a Web-based Framework to Manage Grid Deployments.

Georgios Oikonomou¹ and Theodore Apostolopoulos¹

¹Department of Informatics, Athens University of Economics and Business, Athens, Greece

Abstract - WebDMF is a Web-based Framework for the Management of Distributed services. It is based on the Web-based Enterprise Management (WBEM) standards family and introduces a middleware layer of entities called “Representatives”. Details related to the managed application are detached from the representative logic, making the framework suitable for a variety of services. WebDMF can be integrated with existing WBEM infrastructures and is complementary to web service-based management efforts. This paper describes how the framework can be used to manage grids without modifications to existing installations. It compares the proposed solution with other research initiatives. Experiments on an emulated network topology indicate its viability.

Keywords: WebDMF, Grid Management, Distributed Services Management, Web-based Enterprise Management, Common Information Model.

1 Introduction

During the past decades the scenery in computing and networking has undergone revolutionary changes. From the era of single, centralised systems we are steadily moving to an era of highly decentralised, interconnected nodes that share resources in order to provide services transparently to the end user.

Traditionally, legacy management approaches such as the Simple Network Management Protocol (SNMP) [1], targeted single nodes. The current paradigm presents new challenges and increases complexity in the area of network and systems management. There is need for solutions that view a distributed deployment as a whole, instead of as a set of isolated hosts.

The Web-based Distributed Management Framework (WebDMF) is the result of our work detailed in [2]. It is a framework for the management of distributed services and uses standard web technologies. Its core is based on the Web-based Enterprise Management (WBEM) family of specifications [3], [4], [5]. It is not limited to monitoring but is also capable of modifying the run-time parameters of the managed service. Finally, it has a wide target group. It can perform the management of a variety of distributed systems, such as distributed file systems, computer clusters and computational or data grids. However, multiprocessor, multi-core, parallel computing and similar systems are considered out of the scope of our work, even though they are very often referred to as “distributed”. The main contribution of this paper is three-fold:

• We demonstrate how a WebDMF deployment can be used for the management of a grid, without any modification to existing WBEM management infrastructures.
• We provide indications for the viability of the approach through a preliminary performance evaluation.
• We show that WebDMF is not competitive to emerging Web Service-based grid management initiatives. Instead, it is a step towards the same direction.

Section 2 summarizes some recent approaches in the field of grid management and compares our work with those efforts. In order to familiarize the reader with some basic concepts, section 3 presents a short introduction to the WBEM family of standards. In section 4 we briefly describe WebDMF’s architecture and some implementation details. In the same section we demonstrate how the framework can be used to manage grids. Finally, we discuss the relationship between WebDMF and Web Service-based management and we present some preliminary evaluation results. Section 5 presents our conclusions.

2 Related Work – Motivation

In this section we aim to outline some of the research initiatives in the field of grid management. The brief review is limited to the most recent ones.

2.1 Related Work

An important approach is the one proposed by the Open Grid Forum (OGF). OGF’s Grid Monitoring Architecture (GMA) uses an event producer – event consumer model to monitor grid resources [6]. However, as the name suggests, GMA is limited to monitoring. It lacks active management and configuration capabilities.
glite is a grid computing middleware, developed as part of the enabling grids for e-sciencE (egee) project. glite implements an “information and monitoring subsystem”, called r-gma (relational gma), which is a modification of ogf’s gma. therefore it also only serves monitoring purposes [7].

the unified grid management and data architecture (uganda) is an enterprise level workflow and grid management system [8]. it contains a grid infrastructure manager called magi. magi has many features but is limited to the management of ugan-da deployments.

mrf is a multi-layer resource reconfiguration framework for grid computing [9]. it has been implemented on a grid-enabled distributed shared memory (dsm) system called teamster-g [10].

monalisa stands for “monitoring agents using a large integrated services architecture”. it “aims to provide a distributed service architecture which is used to collect and process monitoring information” [11]. many globus deployments use monalisa to support management tasks. again, the lack of capability to modify the running parameters of the managed resource is notable.

finally, we should mention emerging service-based management initiatives, such as the web services distributed management (wsdm) [12] standard and the web services for management (ws-man) specification [13]. due to their importance, they are discussed in greater detail in section 4 of this paper.

2.2 motivation

table i compares webdmf with the solutions that we presented above. for this comparison we consider three factors:

<table>
<thead>
<tr>
<th>Name</th>
<th>Monitoring</th>
<th>Set</th>
<th>Target Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>ogf’s gma</td>
<td>y</td>
<td></td>
<td>wide</td>
</tr>
<tr>
<td>glite – r-gma</td>
<td>y</td>
<td></td>
<td>focused</td>
</tr>
<tr>
<td>ugan-da – magi</td>
<td>y y</td>
<td></td>
<td>focused</td>
</tr>
<tr>
<td>mrf – teamster-g</td>
<td>y y</td>
<td></td>
<td>focused</td>
</tr>
<tr>
<td>monalisa</td>
<td>y</td>
<td></td>
<td>wide</td>
</tr>
<tr>
<td>webdmf</td>
<td>y y</td>
<td></td>
<td>wide</td>
</tr>
</tbody>
</table>

- the ability to perform monitoring.
- whether the approach can actively modify the grid’s run-time parameters.

- whether the approach is generic or focuses on infrastructures implemented using a specific technology.

our motivation to design webdmf was to provide a framework that would be generic enough to manage grid deployments regardless of the technology used to implement their infrastructure. at the same time, it should not be limited to monitoring but also provide “set” capabilities. other advantages are:

- it is based on wbem. this is a family of open standards.
- wbem allows easy integration with web service – based management approaches.
- wbem has been considered adequate for the management of applications, as opposed to other approaches (e.g. snmp) that focus on the management of devices.
- it provides interoperability with existing wbem-based management infrastructures.

3 web-based enterprise management

web-based enterprise management (wbem) is a set of specifications published by the distributed management task force (dmtf). a large number of companies are also involved in this ongoing management initiative. this section presents a brief introduction to the wbem family of standards.

fig. 1 displays the three core wbem components. the “common information model” (cim) is a set of specifications for the modeling of management data [3]. it is an object-oriented, platform-independent model maintained by the dmtf. it includes a “core schema” with definitions that apply to all management areas. it also includes a set of “common models” that represent common management areas, such as networks, hardware, software and services. finally, the cim allows manufacturers to define technology-specific “extension schemas” that directly suit the management needs of their implementations.

for the interaction between wbem entities (clients and managed elements), wbem uses a set of well-defined
request and response data packets. CIM elements are encoded in XML in accordance with the xmlCIM specification [4]. The resulting XML document is then transmitted over a network as the payload of an HTTP message. This transport mechanism is called “CIM Operations over HTTP” [5].

WBEM follows the client-server paradigm. The WBEM client corresponds to the term “management station” used in other management architectures. A WBEM server is made up of components as portrayed in Fig. 2.

![WBEM Instrumentation](image)

The WBEM client does not have direct access to the managed resources. Instead, it sends requests to the CIM Object Manager (CIMOM), using CIM over HTTP. The CIMOM handles all communication with the client, delegates requests to the appropriate providers and returns responses.

Providers act as plugins for the CIMOM. They are responsible for the actual implementation of the management operations for a managed resource. Therefore, providers are implementation-specific. The repository is the part of the WBEM server that stores the definitions of the core, common and extension CIM schemas.

A significant number of vendors have started releasing WBEM products. The SBLIM open source project offers a suite of WBEM-related tools. Furthermore, OpenPegasus, OpenWBEM and WBEMServices are some noteworthy, open source CIMOM implementations. There are also numerous commercial solutions.

4 WebDMF: Web-based Management of Distributed Services

In this section we introduce the reader to the concept and design of the WebDMF management framework and present some implementation details. Due to length restrictions we can not provide deep technical design details. We explain how the framework can be used to manage grid deployments. The section continues with a discussion about the relationship between WebDMF and Web Service-based management. It concludes with a preliminary performance evaluation, indicating the viability of the approach.

4.1 Design

WebDMF stands for Web-based Distributed Management Framework. It treats a distributed system as a number of host nodes. They are interconnected over a network and share resources to provide services to the end user. The proposed framework’s aim is to provide management facilities for them. Through their management, we achieve the management of the entire deployment.

The architecture is based on the WBEM family of technologies. Nodes function as WBEM entities; clients, servers or both, depending on their role in the deployment. The messages exchanged between nodes are CIM-XML messages.

WebDMF’s design introduces a middleware layer of entities that we call “Management Representatives”. They act as peers and form a management overlay network. This new layer of nodes is integrated with the existing WBEM-based management infrastructure. Representatives act as intermediaries between existing WBEM clients and CIM Object Managers. In our work we use the terms “Management” and “Service” node when referring to those entities. This resembles the “Manager of Managers” (MoM) approach. However, in MoM there is no direct communication between domain managers. Representatives are aware of the existence of their peers. Therefore, WebDMF adopts the “Distributed Management” approach. By distributing management over several nodes throughout the network, we can increase reliability, robustness and performance, while network communication and computation costs decrease [14]. Fig. 3 displays the three management entities mentioned above, forming a very simple topology.

A “Management Node” is a typical WBEM client. It is used to monitor and configure the various operational parameters of the distributed service. Any existing WBEM client software can be used without modifications.

A “Service Node” is the term used when referring to any node – member of the distributed service. For instance, in the case of a data grid, the term would be used to describe a storage device. Similarly, in a computational grid, the term can describe an execution host.
previously, the role of a node in a particular grid deployment does not affect the design of our framework.

Typically, a Service Node executes an instance of the (distributed) managed service. As displayed in Fig. 4 (a), a WBEM request is received by the CIMOM on the Service Node. A provider specifically written for the service handles the execution of the management operation. The existence of such a provider is a requirement. In other words, the distributed service must be manageable through WBEM. Alternatively, a service may be manageable through SNMP, as shown in Fig. 4 (b). In such a case the node may still participate in WebDMF deployments but some functional restrictions will apply.

The framework’s introduces an entity called the “Management Representative”. This entity receives requests from a WBEM client and performs management actions on the relevant service nodes. After a series of message exchanges, it will respond to the initial request. A representative is more than a simple ‘proxy’ that receives and forwards requests. It performs a number of other operations including the following:

- Exchanges messages with other representatives regarding the state of the system as a whole.
- Keeps a record of Service Nodes that participate in the deployment.
- Redirects requests to other representatives.

Fig. 5 displays the generic case of a distributed deployment. Communication between representatives is also performed over WBEM.

The initial requests do not state explicitly which service nodes are involved in the management task. The decision about the destination of the intermediate message exchange is part of the functionality implemented in the representative. The message exchange is transparent to the management node and the end user.

In order to achieve the above functionality, a representative is further split into building blocks, as shown in Fig. 6. It can act as a WBEM server as well as a client. Initial requests are received by the CIMOM on the representative. They are delegated to the WebDMF provider module for further processing. The module performs the following functions:

- Determines whether the request can be served locally.
• If the node can not directly serve the request then it selects the appropriate representative and forwards it.
• If the request can be served locally, the representative creates a list of service nodes that should be contacted and issues intermediate requests.
• It processes intermediate responses and generates the final response.
• Finally, it maintains information about the distributed system’s topology.

In some situations, a service node does not support WBEM but is only manageable through SNMP. In this case, the representative attempts to perform the operation using SNMP methods. This is based on a set of WBEM to SNMP mapping rules. This has limitations since it is not possible to map all methods. However, even under limitations, the legacy service node can still participate in the deployment.

In a WebDMF deployment, a representative is responsible for the management of a group of service nodes. We use the term “Domain” when referring to such groups. Domains are organized in a hierarchical structure. The top level of the hierarchy (root node of the tree) corresponds to the entire deployment. The rationale behind designing the domain hierarchy of each individual deployment can be based on a variety of criteria. For example a system might be separated into domains based on the geographical location of nodes. WebDMF defines two categories of management operations: i) Horizontal (Category A) and ii) Vertical (Category B).

Horizontal Operations enable management of the WebDMF overlay network itself. Those functions can, for example, be used to perform topology changes. The message exchange that takes place does not involve Service Nodes. Therefore, the managed service is not affected in any way.

On the other hand, vertical operations read and modify the CIM schema on the Service Node, thus achieving management of the target application. Typical examples include:
• Setting new values on CIM objects of many service nodes.
• Reading operational parameters from service nodes and reporting an aggregate (e.g. sum or average).

In line with the above, we have designed two CIM Schemas for WebDMF, the core schema (“WebDMF_Core”) and the request factory. They both reside on the representatives’ repositories. The former schema models the deployment’s logical topology, as discussed earlier. It corresponds to horizontal functions.

The latter schema is represented by the class diagram in Fig 7 and corresponds to vertical functions. The users can call WBEM methods on instances of this schema. In doing so, they can define the management operations that they wish to perform on the target application. Each request towards the distributed deployment is treated as a managed resource itself. For example, a user can create a new request. They can execute it periodically and read the results. They can modify it, re-execute it and finally delete it. Each request is mapped by the representative to intermediate WBEM requests issued to service nodes.

Fig. 7. Request Factory CIM Schema.

Request factory classes are generic. They are not related in any way with the CIM schema of the managed application. This makes WebDMF appropriate for the management of a wide variety of services. Furthermore, it does not need re-configuration when the target schema is modified.

4.2 Implementation

The WebDMF representative is implemented as a single shared object library file (.so). It is comprised of a set of WBEM providers. Each one of them implements the management operations for a class of the WebDMF schemas. The interface between the CIMOM and the providers complies with the Common Manageability Programming Interface (CMPI). Providers themselves are written using C++ coding. This does not break CIMOM independence, as described in [15]. The representative was developed on Linux 2.6.20 machines. We used gcc 4.1.2 and version 2.17.50 of binutils. It has been tested with version 2.7.0 of the Open Pegasus CIMOM.

4.3 Using WebDMF to Manage Grids

In a grid environment, a service node can potentially be an execution host, a scheduler, a meta-scheduler or a
resource allocation host. The previous list is non-inclusive. The role of a node does not affect the design of our framework.

What we need is a CIM schema and the relevant providers that can implement WBEM management for the service node. Such schemas and providers do exist. For example, an architecture for flexible monitoring of various WSRF-based grid services is presented in [16]. This architecture uses a WBEM provider that communicates with WSRF hosts and gathers status data. In a WebDMF deployment, we could have many such providers across various domains. Each would reside on a service node and monitor the managed application.

4.4 WebDMF and Web Services

The grid community has been working for more than five years to transform grid computing systems into a group of web service-based building blocks. In line with this effort, management of the resulting infrastructures has also moved towards web service-based approaches. The recent OASIS Web Services Distributed Management (WSDM) [12] standard and the DMTF Web Services for Management (WS-Man) specification [13] have been considered enablers of this vision.

WebDMF adopts a resource – centric approach. This may seem to be a step in the opposite direction. It is not. The authors of this paper consider web service – based approaches to be a very necessary and extremely valuable effort. However, service – oriented management approaches are model – agnostic. They do not define the properties, operations, relationships, and events of managed resources [12]. Two important reasons why we choose WBEM for the resource layer are the following:

- WS-Management exposes CIM resources via web services, as defined in [17]. CIM is an inherent part of WBEM, as explained earlier in this paper.
- DMTF members are working on publishing a standard for the mapping between WS-Man operations and WBEM Generic Operations [18].

Furthermore, in order to implement a WS-Man operation, a Web Service endpoint needs to delegate requests to instrumentation that can operate on the managed resource. In current, open source WS-Man implementations, management requests are eventually served by a WBEM Server and the appropriate providers. WS-Man and WBEM are related and complementary to each other.

The WebDMF representative has been implemented as a WBEM provider Therefore, if the CIMOM operating on the Representative node provides WS-Man client interfaces, the WebDMF provider will operate normally.

4.5 Performance Evaluation

In this section we present a preliminary evaluation of WebDMF’s performance. Results presented here are not simulation results. They have been obtained from actual code execution and are used as an indication of the solution’s viability.

In order to perform measurements, we installed a testbed environment using ModelNet [19]. Our topology emulates a wide-area network. It consists of 250 virtual nodes situated in 3 LANs. Each LAN has its own gateway to the WAN. The 3 gateways are interconnected via a backbone network, with high bandwidth, low delay links. We have also installed two WebDMF representatives (nodes R1 and R2). This is portrayed in Fig. 8.

![Fig. 8. The emulated topology and test scenario.](image)

We assume that for this network deployment, we wish to obtain the total amount of available physical memory for the 200 nodes hosted in one of the LANs. Among those, 50 do not support basic WBEM instrumentation. They only offer SNMP-based management facilities.

In this scenario, the client will form a WBEM CreateInstance() request for class WebDMF_RequestWBEM of the request factory. It is initially sent to the WebDMF Representative (R1). The request will get forwarded to R2. R2 will collect data from the 200 service nodes as follows:

- R2 sends intermediate requests to the 150 WBEM-enabled nodes. Those requests invoke the EnumerateInstances() operation for class Linux_OperatingSystem. Responses are sent back to R2 from the service nodes.
- As stated previously, in this scenario there are 50 SNMP-enabled nodes. R2 sends SNMP-Get packets to
the packets crossing the network are of a small size (a few bytes). The total execution time includes the following:

- Communication delays during request-response exchanges. This includes TCP connection setup for all WBEM message exchanges. This does not apply to the SNMP case. SNMP uses UDP at the transport layer, therefore no connection is used.
- Processing overheads on R1 and R2. This is imposed by WebDMF’s functionality.
- Processing at the service nodes to calculate the requested value and generate a response.

The absolute value of the average completion time may seem rather high. However, in general terms, processing times are minimal compared to TCP connection setup and message exchange. With that in mind, we can see that each of the 204 request-responses completes in 30.57 milliseconds on average. This is normal. After 200 repetitions we observe low statistical dispersion (variance and standard deviation). This indicates that the measured values are not widely spread around the mean. We draw the same conclusion by estimating a 95% confidence interval for the mean. This indicates that the same experiment will complete in the same time under similar network traffic conditions.

5 Conclusions

Ideally, a management framework should support grid deployments without need for major modifications on the existing infrastructure. It should not be limited by the technology used to implement the grid and be generic in order to support future changes. In this paper we introduce WebDMF, a Web-based Distributed Management Framework and present how it can be used to manage grids. We discuss its generality and demonstrate its viability through performance evaluation. Finally, the paper presents its advantages compared to alternative approaches and shows how it is complementary to emerging Web Service-based management approaches.

6 References