

Web-Based Management of Content Delivery Networks

George Oikonomou and Theodore Apostolopoulos

Athens University of Economics and Business,
Department of Informatics,
Athens, Greece
`{geo,tca}@aueb.gr`

Abstract. Abstract overlay networks have been considered enablers of efficient management for decentralized, large scale service deployments. A Content Delivery Network (CDN) is an example of service falling within this category. The result of our research is WebDMF, a management framework for distributed services based on the Web-Based Enterprise Management (WBEM) family of specifications. Abstract design, combined with a middleware layer of entities called “Representatives”, makes WebDMF suitable for the management of a variety of services. Details related to the management of each particular service are detached from the representative logic. This paper discusses how WebDMF can be used for the management of CDNs. A WBEM provider resides on each host participating in the service deployment and implements CDN-specific operations. WebDMF representatives decentralize, unify and coordinate those on a deployment scale. Preliminary measurements on an emulated network topology are also presented as an indication of the solution’s viability and scalability.

Key words: Content Delivery Networks, Web-Based Enterprise Management, WebDMF, Distributed Services Management

1 Introduction

The current paradigm of large scale decentralized service deployments poses new requirements in the field of network and systems management. Legacy approaches, such as the Simple Network Management Protocol (SNMP) [1] focus on single nodes and are best suited for devices. However, SNMP is considered less appropriate for applications and services. Nowadays, abstract overlay networks are considered enablers of distributed services management. Techniques used to access resources should be suitable for applications as well as devices.

WebDMF, the result of our research, is a Web-based Management Framework for Distributed Services. Its core is based on the Web-based Enterprise Management (WBEM) family of specifications [2–4]. It is not limited to monitoring but is also capable of modifying the run-time parameters of the managed service. Due to its abstract design it has wide scope and is suitable for the management of a variety of services.

WebDMF’s development consisted in the design of a CIM extension schema and the implementation of a WBEM provider enabling CIM operations on classes of this schema. The framework’s design, implementation and performance evaluation have been discussed in detail in [5]. We have particularly studied its application on the management of grids and web server load-balancing schemes.

This paper extends our previous work by discussing how WebDMF can be used for the management of Content Delivery Networks (CDNs). An open source tool (OpenCDN) was used to install a test CDN deployment and perform measurements. In this context, this paper’s main contribution is two-fold:

- We designed an extension CIM schema for OpenCDN hosts. For this schema, we implemented a WBEM provider offering management facilities for hosts participating in a CDN service deployment.
- We demonstrate how we can achieve management of the entire CDN deployment by combining WebDMF operations with the aforementioned provider.

Section 2 briefly outlines existing efforts in the field of management of content delivery networks. In order to familiarize the reader with some basic concepts, the section continues with an introduction to the WBEM standards family and the OpenCDN tool. In Sect. 3 we present an overview of WebDMF’s architectural design. In Sect. 4 we disclose the design and implementation of a WBEM provider for OpenCDN hosts. Furthermore we demonstrate how this provider can be integrated with a WebDMF network in order to perform management of the OpenCDN deployment. Section 5 presents an experiment indicative of the framework’s scalability and in Sect. 6 we discuss our conclusions.

2 Related Work

In the field of Content Delivery (or Distribution) Networks, the term ‘management’ is often used to describe a variety of different concepts. A theoretic approach focusing on *content consistency* management in caching schemes has been discussed in [6]. In [7], the authors present a “*policy-based* architecture for the control and management of content distribution networks”. An agent-based distributed management framework, using SNMP for data gathering from nodes has been presented in [8]. Finally, UPGRADE-CDN is a platform for the development of CDNs. It introduces a *monitoring* framework called “AMonitor” [9]. GLOBULE [10] is a collaborative content delivery network. This effort provides approaches and suggests solutions to large-scale, CDN-related problems. Among other topics, the authors discuss *content replication and consistency* management.

WebDMF goes past simple monitoring by providing active *configuration* management capabilities. Furthermore, in the context of this paper, the managed resource is the deployment itself. This includes content, network links as well as hosts participating in the deployment.

2.1 Web-Based Enterprise Management

Web-Based Enterprise Management (WBEM) is a set of management specifications published by the Distributed Management Task Force (DMTF). WBEM is made up of three core components.

1. Model: The “Common Information Model” (CIM) is an object-oriented, platform-independent approach for the modeling of management data [2]. 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.
2. Encoding: WBEM adopts the client-server paradigm. For the interaction between entities (clients and managed elements), it uses a set of well-defined request and response data packets. CIM elements are encoded in XML in accordance with the xmlCIM specification [3].
3. Transport: The resulting XML document is transmitted over the network as the payload of an HTTP message. This transport mechanism is called “CIM Operations over HTTP” [4].

The term CIM-XML is often used to refer to the combination of 2 and 3 above.

A WBEM server is made up of components as portrayed in Fig. 1. 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. It delegates requests to the appropriate providers and returns responses. Providers act as plugins for the CIMOM. They are responsible for the realization of management operations for a resource. Therefore, providers are implementation-specific. The repository is the part of the WBEM server that stores 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 examples of open source CIMOM implementations. There are also numerous commercial solutions.

2.2 Open Source Content Delivery Network Software

The “Open Content Delivery Network” (OpenCDN) project is an open source effort aiming to provide a platform for the deployment of content delivery services for streaming multimedia. It operates by creating an overlay network of hosts. These hosts stream content using an application layer multicast scheme. An OpenCDN deployment is made up of three components:

- Origins: Nodes that host the original multimedia content.
- Request Routing and Distribution Manager (RRDM): A centralized control entity that coordinates the streaming of content between nodes.

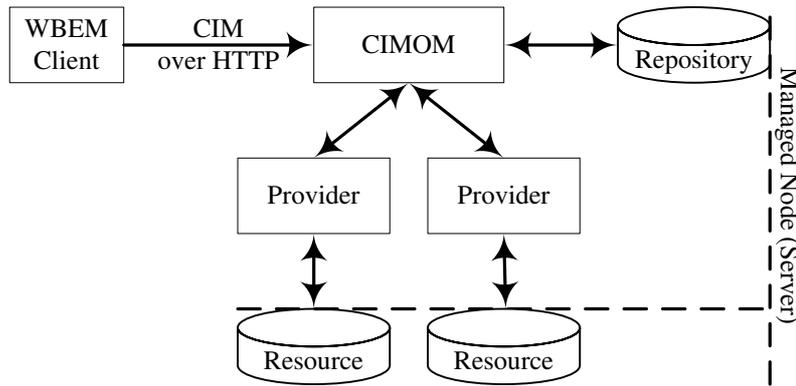


Fig. 1. WBEM instrumentation.

- Distribution Nodes (or Nodes): These are the hosts that perform stream relay from the Origin to the final recipients.

Distribution nodes are organized hierarchically. Depending on their location in the hierarchy they are classified as “First Hop”, “Transit” or “Last Hop”. OpenCDN is written in PERL and uses XML-RPC messages for the communication between distribution nodes, origins and the RRDM. Adapters with existing streaming technologies are used in order to achieve content relay. New adapters can be written in order to provide support for more technologies. Further information and technical details on OpenCDN’s design can be found in [11].

3 WebDMF: A Framework for the Management of Distributed Services

WebDMF stands for Web-based Distributed Management Framework. It treats a distributed deployment as a number of host nodes, called “Service Nodes”. They are interconnected over a network and share resources to provide services to the end user. This section summarizes the framework’s architecture and is meant as an overview. Detailed technical design has been disclosed in [5].

WebDMF’s design is based on the WBEM family of technologies. Nodes function as WBEM entities; clients, servers or both, depending on their role in the deployment. Communication between nodes is performed in accordance with CIM-XML (see Sec. 2.1). WebDMF introduces a middleware layer of entities called “Representatives”, forming a management overlay network. Representatives issue WBEM requests to perform management operations on service nodes. Once responses have been collected, they are unified in order to provide an abstract view of the deployment to the user. This resembles the “Manager of Managers” (MoM) approach. However, in MoM there is no direct communication between domain managers. In WebDMF, representatives are aware of

their peers and communicate with them. Therefore WebDMF falls into the “Distributed Management” category.

3.1 WebDMF Entities

A “Management Node” corresponds to a typical WBEM client. It is used to monitor and configure 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 content delivery network a service node is an intermediate relay or a node hosting content. A node’s role in a particular distributed deployment does not affect WebDMF’s functionality. A service node executes an instance of the (distributed) managed service and a CIM Object Manager. WebDMF requests traverse the overlay network and eventually reach CIMOMs on service nodes. This is where management operations take place.

WebDMF introduces an entity called the “Management Representative”. This entity receives requests from a WBEM client (management node) 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 is further split into building blocks, as shown in Fig. 2. 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:

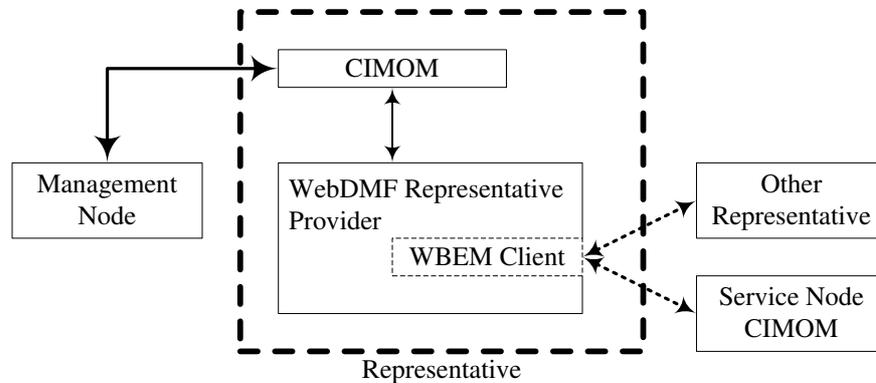


Fig. 2. The WebDMF representative. The *solid line* corresponds to the initial request and the final response whereas *dashed lines* represent intermediate request-response exchanges.

- Determines whether the request can be immediately served.

- If the node can not directly serve the request then it selects an appropriate representative and forwards it.
- If the request can be immediately served, the representative creates a list of service nodes that should be contacted and issues intermediate requests.
- Intermediate responses are processed and a final response is generated.
- Finally, a representative maintains information about the distributed service's topology.

In a WebDMF deployment, a representative is responsible for the management of a group of service nodes. The term “Domain” is used when referring to such groups. Domains are organized in a hierarchical structure. The hierarchy's root node 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 nodes geographical locations. More sophisticated clustering techniques can also be used.

3.2 WebDMF Operations and CIM Schemas

WebDMF defines two categories of management operations: i) horizontal and ii) vertical. Horizontal operations enable management of the WebDMF overlay network itself. Those functions can, for example, be used to perform topology changes. The message exchange taking place does not involve service nodes and the managed service is not affected in any way. On the other hand, vertical operations read and modify CIM instances on service nodes, thus achieving management of the target application. Some common 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 corresponds to vertical functions. Users can call WBEM methods on instances of this schema. In doing so, they can define 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, users 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. 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, the request factory does not need re-configuration when the target schema is modified.

4 Management of Content Delivery Networks

4.1 A WBEM Provider for OpenCDN Hosts

In order to achieve WBEM management of an OpenCDN host, two things are required: i) a CIM schema modeling management information and ii) a WBEM provider. That provider should be OpenCDN-specific in order to be capable of implementing operations on the managed resource. In this case, the managed resource is a process running on the host, offering the service.

As part of our work, we designed a CIM schema for OpenCDN hosts. As displayed in Fig. 3, the schema is made up of 9 classes. Seven of the classes correspond to hosts, one class represents multimedia content and finally one class represents a CIM association. Only class names and relationships are displayed in the figure. Properties and Methods are omitted for clarity reasons. Class *oCDN_Host* defines some properties and methods common to all OpenCDN host types. Similarly, class *oCDN_Node* defines attributes common to distribution nodes, regardless of relay technology. These two classes are abstract (italicized names) therefore can not have instances. Each instance of a non-abstract class corresponds to a different host (or program).

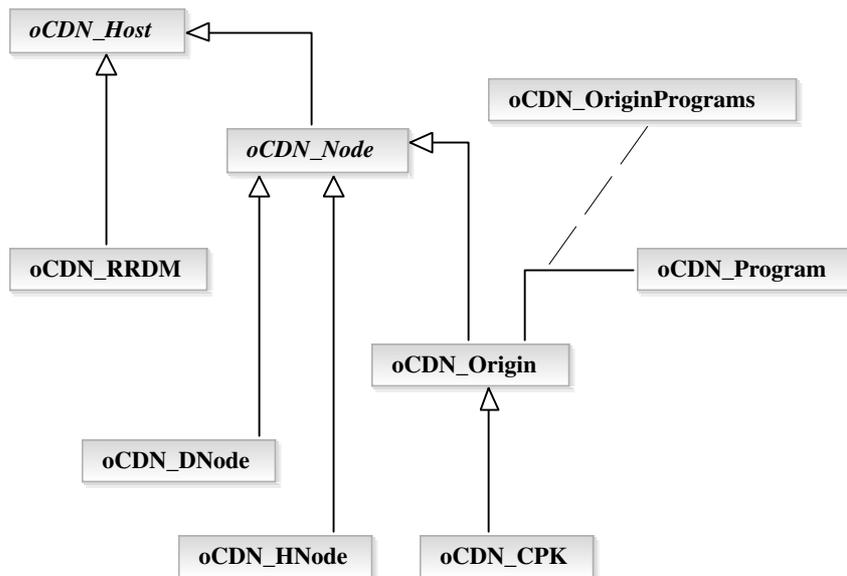


Fig. 3. CIM schema for the management of OpenCDN hosts.

Following the design of a CIM schema for OpenCDN, we implemented a provider for classes *oCDN_DNode*, *oCDN_Origin* and *oCDN_RRDM*. By calling WBEM intrinsic methods on their instances, the user can achieve management of the

respective type of host. For example, the user can restart the DNode service or modify its runtime parameters. To be more specific, the provider implements the following functions:

- Start - stop service. This is performed by calling the `CreateInstance` and `DeleteInstance` intrinsic WBEM methods respectively.
- Retrieve the names of all service instances running on a particular host. This is achieved by the `EnumerateInstanceNames` method. Only instance identifiers are returned by this method, without detailed property values.
- Read the configuration of a running service. The `GetInstance` method is used to achieve this. Furthermore, if more than one service is offered by a host (e.g. RRDM and DNode concurrently) then calling the `EnumerateInstances` method results in a response containing the parameter values for all services. Each service is represented by a separate CIM instance.
- Modify the configuration of a particular instance. This happens by calling the `ModifyInstance` WBEM method.

The figure below (Fig. 4) displays a UML activity diagram for the case of configuration modification. The user issues a `ModifyInstance` request which is received by the CIMOM. If no errors occur during processing, the following steps take place:

1. The CIMOM invokes the OpenCDN provider and delegates the request.
2. The provider checks the requested changes' validity.
3. The provider maintains information of all running services in an instance pool. Each instance corresponds to a different service. At this step the provider searches the pool for the requested instance.
4. The provider performs changes on the instance. It also modifies the configuration of the running process that offers the service and notifies the CIMOM.
5. A response is generated by the CIMOM and sent to the management station, indicating successful completion of the management operation.

4.2 Integrating an Open CDN Deployment with WebDMF

The WBEM provider discussed in the previous section handles all management requests for a single OpenCDN host. Its design is directly related to OpenCDN's implementation details and it can perform management operations directly on resources. However, an OpenCDN deployment is made up of multiple hosts. This is usually one RRDM and multiple origin and distribution nodes. Even with the OpenCDN provider installed on each of them, they would still be treated as stand alone entities in a typical WBEM-based management infrastructure. WebDMF adds an abstraction layer that unifies the hosts as parts of a single service.

In WebDMF terminology (see Sect. 3), all OpenCDN nodes are treated as "service nodes". In order to integrate the CDN deployment with a WebDMF management overlay, two steps need to be performed:

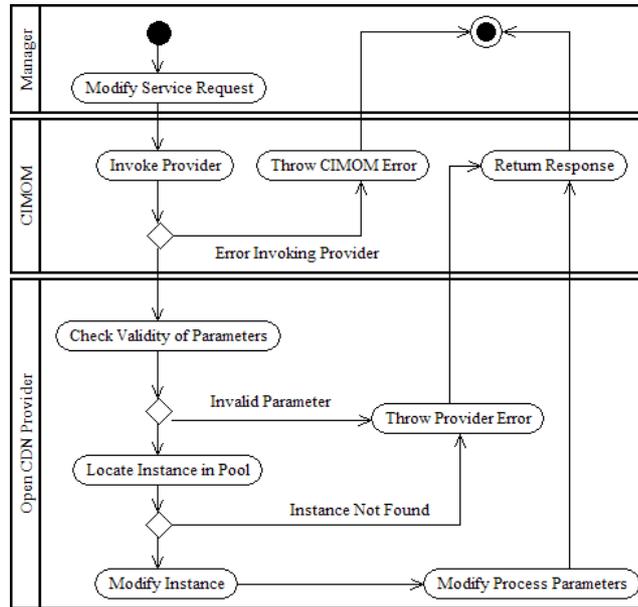


Fig. 4. UML activity diagram modeling a `ModifyInstance` operation. This operation involves three actors (the CIMOM, the OpenCDN provider and the management station). *Swimlanes* are used to group activities performed by the same actor.

- The OpenCDN network needs to be broken down into domains. This is a change in logical terms but does not cause any alterations on service nodes.
- Representatives need to be installed and assigned for the aforementioned domains. Nodes are registered on the representatives through WebDMF *horizontal* operations, thus generating a virtual topology for the deployment.

Figure 5 displays a sample OpenCDN topology. Shapes indicate a node's role in the deployment (RRDM, relay or origin). Arrows show the path that a multimedia stream would follow from the origin node towards the final viewer. This topology has been broken down into three domains, different fill textures are used to distinguish those. This domain structure adopts a two level hierarchy. The top level (root of the tree) corresponds to the entire deployment. The second level is the one displayed in the figure. This domain structure is just indicative. It would be possible to use a multi-level hierarchy with super-domains and sub-domains. For example Domains 1 and 2 could both be children of a single parent. In this example topology, 3 representatives are being used, displayed with hexagons and not linked with arrows to any other nodes. In line with the above, different fill textures indicate which domains they have been assigned to.

Once the above two steps have been completed, a user can issue management requests for the CDN by using WebDMF *vertical* facilities, as outlined in Sec. 3.2. Representatives communicate among themselves registering changes

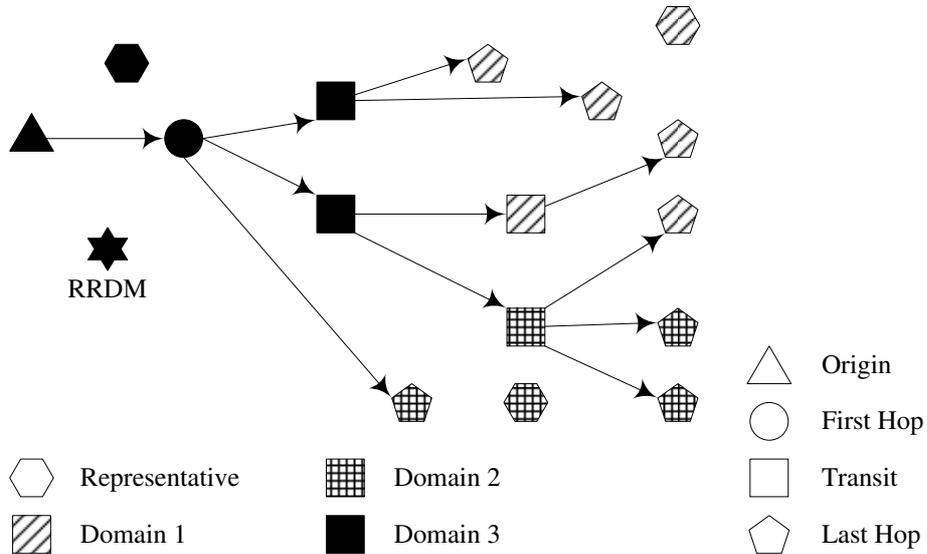


Fig. 5. A sample OpenCDN deployment broken down into three WebDMF domains.

in the CDN deployment. They also map WebDMF requests to WBEM operations for classes of the OpenCDN CIM schema on the appropriate service nodes (Sec. 4.1). Changes on nodes are performed by the OpenCDN WBEM provider that resides on each node's CIMOM.

4.3 Implementation Details

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 implementing management operations for a class of the WebDMF schemas. The OpenCDN provider is also a single shared object file implementing WBEM operations for classes `oCDN-Origin`, `oCDN-RRDM` and `oCDN-DNode`. The interface between the CIMOM and providers complies with the Common Manageability Programming Interface (CMPI) [17]. Providers themselves are written in C++. This does not break CIMOM independence, as described in [17]. The representative was developed on Linux 2.6.20 machines. We used gcc 4.1.2 and version 2.17.50 of binutils. Testing took place using version 2.7.0 of the Open Pegasus CIMOM.

5 Performance Evaluation

In order to evaluate WebDMF, we installed a testbed environment using ModelNet [12] and executed various measurements. Results were obtained from actual

code execution on an emulated network topology, they are not simulation results. In this section we present the outcome of an experiment that is indicative of WebDMF's scalability.

The topology emulated by ModelNet represents a wide-area network. It consists of 300 virtual nodes situated in 3 LANs with each LAN having its own gateway to the WAN. The 3 gateways are interconnected via a backbone network, with high bandwidth, low delay links. We also installed two WebDMF representatives (nodes R1 and R2). In our experiment, service nodes are OpenCDN origin nodes. The management operation constitutes of setting them to register with a different RRDM. This involves changing property values of the `oCDN_Origin` instances on service nodes. A client issues an initial WebDMF request to representative R1 (this is a `WBEM CreateInstance` for class `WebDMF_RequestWBEM` of the request factory). R1 forwards the request to R2. R2 sends `ModifyInstance` requests to service nodes. Responses follow the reverse path. Figure 6 portrays the emulated topology and test scenario. For clarity reasons, we omit service nodes residing in other domains.

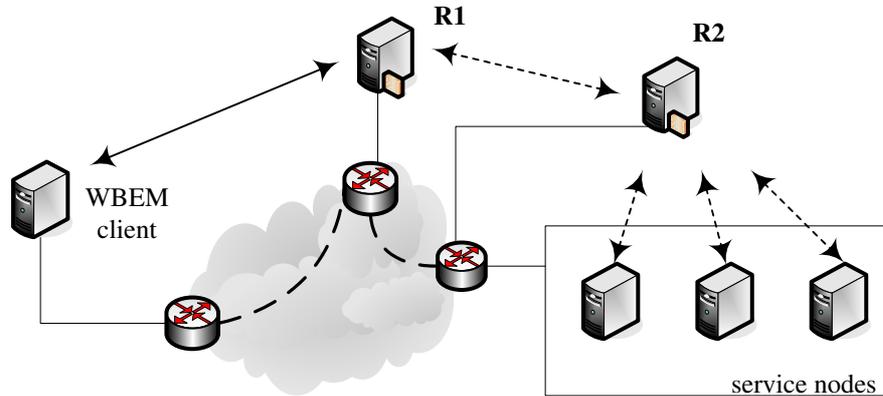


Fig. 6. Emulated topology and test scenario.

In this experiment we wish to evaluate the impact of service node count on the time needed to complete the management operation described above. We repeated the experiment by gradually increasing the number of service nodes involved in the operation. Starting with 10 service nodes, we increased the number by 10 up to 270 (27 steps). For each step we executed the management operation 100 times. For each repetition we measured elapsed time with microsecond accuracy. This resulted in a total sample of 2700 observations (27 steps x 100 repetitions). Figure 7 displays a scatter-plot of the observed values. The X axis displays the number of service nodes involved in the operation. The Y axis corresponds to the total time needed to complete the management operation.

Based on the same sample we calculated a simple linear regression line using the number of nodes as the independent variable. The resulting regression line is

Impact of service node number on total completion time.

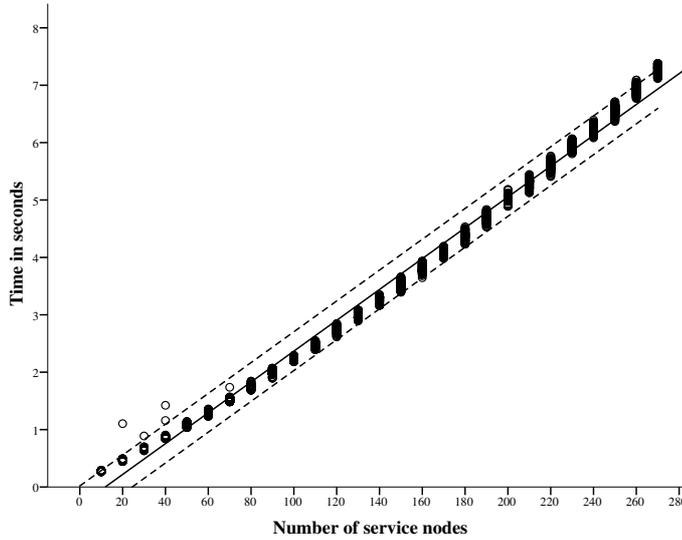


Fig. 7. Total completion time by number of service nodes. *Circles* represent observed values. The *solid line* represents a linear regression whereas the *dashed lines* display confidence intervals for the estimated values. Most of the observed values fall within the boundaries.

plotted on the same diagram. For this regression, the R^2 coefficient equals 0.993, indicating strongly linear co-relation. Furthermore, residuals are distributed normally, indicating good fit of the regression line on the sample. More analysis of the same sample and goodness-of-fit tests for this regression line on a second sample verify the validity of those findings. Those results are not disclosed here due to space limitations.

This experiment shows that under similar network load conditions, the time needed to complete a WebDMF vertical operation is linearly related to the number of service nodes that a representative will have to contact. This is an indication of the framework's scalability in dense service deployments.

6 Conclusions

In this paper we demonstrated how WebDFM can be applied in the field of Content Delivery Network management. This case study can be used to bring out some of the framework's advantages. Its open standards-based design facilitates its integration with existing WBEM infrastructures. WBEM relies on web technologies (HTTP for transport and XML for content encoding). This provides alternatives for security related problems such as firewall traversal. Existing encryption techniques, such as HTTP over SSL, can be used off the shelf, without need for modifications on the framework itself. WebDMF's design is abstract.

By detaching the details of the managed service from the representative logic, it is generic and suitable for the management of a wide variety of services.

WebDMF is resource-centric, something that may seem to be a step in the opposite direction compared to emerging, web service-based efforts [13, 14]. However, those initiatives are model-agnostic. They do not define properties and operations for the managed resources [14]. Recently, the DMTF published preliminary specifications suggesting a method for exposing CIM resources with WS-Man [15, 16]. Further study of those documents shows that WBEM and WS-Man are related technologies. By acting on the resource layer of a service-based management deployment, WebDMF is complementary to those approaches.

References

1. W. Stallings: SNMP, SNMPv2, SNMPv3, RMON 1 and 2. Addison Wesley (1999)
2. CIM Infrastructure Specification. DMTF Standard, DSP0004 (2005)
3. Representation of CIM in XML. DMTF Standard, DSP0201 (2007)
4. CIM Operations over HTTP. DMTF Standard, DSP0200 (2007)
5. Oikonomou, G., Apostolopoulos, T.: WebDMF: A Web-based Management Framework for Distributed Services. In: The 2008 International Conference of Parallel and Distributed Computing (2008)
6. Zhou, S., Katto, J., Yasuda, Y.: Supporting Consistency Management in Dynamic Content Distribution Overlays. In: The Joint International Conference on Autonomous and Autonomous Systems and International Conference on Networking and Services. IEEE Computer Society (2005)
7. Verma, D.C., Calo, S., Amiri, K.: Policy-based Management of Content Distribution Networks. IEEE Network. 16(2), 34-39 (2002)
8. Bivens, A., Gupta, R., McLean, I., Szymanski, B., White, J.: Scalability and Performance of an Agent-based Network Management Middleware, International Journal of Network Management. 14, 131-146 (2004)
9. Fortino, G., Russo, W.: Using p2p, Grid and Agent Technologies for the Development of Content Distribution Networks. Future Generation Computer Systems. 24(3), 180-190 (2008)
10. Guillaume P., van Steen, M.: Globule: A Collaborative Content Delivery Network. IEEE Communications Magazine. 44(8) 127-133 (2006)
11. OpenCDN Project, <http://labtel.ing.uniroma1.it/opencdn/>
12. Vahdat, A., Yocum, K., Walsh, K., Mahadevan, P., Kostic, D., Chase, J., Becker, D.: Scalability and Accuracy in a Large-Scale Network Emulator. In: 5th Symposium on Operating Systems Design and Implementation (OSDI) (2002)
13. Web Services for Management (WS Management). DMTF Specification, DSP0226 (2006)
14. An Introduction to WSDM. OASIS committee draft. (2006)
15. WS-CIM Mapping Specification. DMTF Preliminary Specification, DSP0230 (2006)
16. WS-Management CIM Binding Spec. DMTF Preliminary Specification, DSP0227 (2006)
17. Common Manageability Programming Interface (CMPI), The Open Group Technical Standard. (2004)