Security and Performance Trade-off in PerfCloud*

Valentina Casola[1], Antonio Cuomo[2], Massimiliano Rak[3], and Umberto Villano[2]

[1] Dipartimento di Informatica e Sistemistica, Università Federico II di Napoli
casolav@unina.it
[2] RCOST and Dipartimento di Ingegneria, Università del Sannio
{antonio.cuomo,villano}@unisannio.it
[3] Dipartimento di Ingegneria dell'Informazione, Seconda Università di Napoli
massimiliano.rak@unina2.it

Abstract. Both cloud and GRID are computing paradigms that manage large distributed resources, and currently there is a lot of interest in their integration. An integrated architecture cloud-GRID requires fine-grained access control and identity federation among untrusted distributed domains in the cloud. This paper deals with the trade-off between security and performance in such architectures, comparing the overhead introduced by cloud services with different security levels. The quantitative results obtained in PerfCloud, an existing cloudgrid infrastructure, are presented and discussed.

1 Introduction

According to the definition by NIST, cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released, with minimal management effort or service provider interaction [9]. On the other hand, GRID computing is basically a paradigm that aims at enabling access to high performance distributed resources in a simple and standard way. In GRIDs, users can compose complex stateful services in order to build up complex and computation-intensive tasks. GRID and clouds are at least similar, not to mention the use of many common underlying technologies. However, they are typically used for different purposes by different classes of users. In short, clouds are used by users that are prone to buy computing resources to get their results as soon as possible. On the other hand, GRID users wish to exploit the optimum set of resources that can solve their problem, overcoming the boundaries of a single enterprise. In fact, the two technologies complement gracefully each other, and currently their integration is actively investigated. The two principal integration approaches used are the

* The work described in this paper has been partly supported by MIUR-PRIN 2008 project "Cloud@Home: a New Enhanced Computing Paradigm"

M.R. Guarracino et al. (Eds.): Euro-Par 2010 Workshops, LNCS 6586, pp. 633–640, 2011.
© Springer-Verlag Berlin Heidelberg 2011

GRID-on-cloud (a cloud is exploited to build up and to manage a flexible GRID system [4]) and the *cloud-on-GRID* (the well-known and stable GRID infrastructure is exploited to build up a cloud environment).

In brief, adopting the *GRID-on-cloud* approach the GRID middleware is installed and configured on the top of a cloud system. The cloud "elasticity" makes it possible to add and to remove dynamically nodes to/from the resulting GRID infrastructure. The *cloud-on-GRID* approach is instead based on the development of typical cloud services (e.g., services to start up, to destroy, to manage virtual machines) as GRID services, offered by an existing GRID infrastructure [6,13]. Some of the most famous GRID environments are currently experimenting this solution with dedicated projects [13,10]. One of the tough tasks involved in cloud and GRID integration is to provide the integrated environment with a suitable security infrastructure. This is not trivial and might involve performance losses that could not be tolerable for all classes of users. The authors of this paper are involved in a project that aims to join the cloud-on-GRID and GRID-on-cloud approaches, by implementing a cloud on the top of a GRID, and integrating the cloud resources leased to users in the existing GRID. This solution, named *cloudgrid* [8], wishes to collect the advantages (possibly, not disadvantages) of the two computing paradigms. A previous paper [2] outlines the security requirements for a *cloudgrid* and proposes a possible solution, implemented in an existing prototype environment, PerfCloud [8]. This paper instead deals with the quantitative evaluation of the overheads introduced in the overall GRID infrastructure because of the cloud layer, pointing out the trade-off between the security level provided and the end-user GRID performance. It is worth noting that, even if the security/performance trade-off is a well known issue, very few quantitative studies are available on the topic. Besides exploring the cloud-GRID integration field, this paper is also intended to give a contribution for filling this gap. Almost surprisingly, it will be shown that the advanced authorization mechanism needed in the cloud layer turn out to have negligible impact on the GRID overall performance. The remainder of this paper is organized as follows. The next section illustrates the *cloudgrid* security requirements. In Section 3 the focus will be on the *PerfCloud* framework, showing in particular the architectural solutions used to meet security requirements. Section 4 presents a detailed analysis of the performance corresponding to each available security configuration. The paper closes with the conclusions and a discussion on our future work.

2 *Cloudgrid* Security Requirements

The integration of the GRID and cloud is of great interest, as it allows to re-use the effort spent in the last 20 years on GRID computing. In *PerfCloud*, an existing cloud-on-GRID infrastructure with provision for predictive performance evaluation [8], this problem is solved by integrating the virtual resources offered by the cloud into the underlying GRID. Given an existing computing GRID, users can gain access to virtualized resources (namely, to VCs, virtual clusters of

machines) through a cloud interface, and these virtual resources are integrated in the existing GRID and can cooperate with its component systems. Fine-grain access control mechanisms are needed to grant different levels of privilege to users that can play a different role in accessing system resources (both physical and virtual). We have analyzed the different roles and corresponding security policies for the access to both administration and user cloud services, pointing out four different roles: the **System Administrator** and the **Grid User** to respectively manage and use the GRID platform and the physical resources; the **Cloud Administrator** and the **Cloud User** to respectively supervise the cloud environment and configure/use the assigned Virtual Clusters. It is interesting to point out that the a cloud administrator is able to create a new set of virtual clusters and assign them to cloud users, *but does not have administration rights over them, which are owned by the cloud users*. Even if offering to users full rights on the virtual cluster is one of the aims of clouds, this can have a side effect on the *cloudgrid* approach: in fact, a VC administrator has full right access to the VC but he can also manage the new physical GRID site. This represents a big security issue: an user of an hosted site could access physical resources if the cloud administrator does not enforce proper security policies or if it wants to abuse of his role on the physical resources. As a consequence, it is of fundamental importance that the *cloudgrid* approach provides a powerful role-based authorization mechanism.

Data integrity, confidentiality and privacy are not secondary requirements for the proposed architecture. Proper security mechanisms must be enforced not only at application level, but also in the other layers and components of the architecture (e.g., network, transport, service communication, internal application and databases). As we will discuss later, the adoption of secure protocols and encryption techniques has proven to be very useful in such distributed architectures. Furthermore, to fully enable the cloud approach, it is desirable to grant cooperation among users and virtual resources even when they are offered by potentially untrusted domains. So, a federated approach to security and, specifically, to Identity management is required [5].

3 The Security Infrastructure in *PerfCloud*

PerfCloud, our *cloudgrid* implementation, adopts Globus Toolkit 4 (GT4) as GRID middleware, and so we customized and enriched the default Globus configurations in order to meet the security requirements described in the previous section. In particular, GT4 uses the concept of Security Descriptors as standard method for configuring the security requirements and policies of clients and services [11]. GT4 authentication founds on PKI and the adoption of X.509 digital certificates with basic path authentication. The Security Descriptor (SD) makes it possible to specify the communication protocols and some mechanisms to improve security at message level (*Secure Message*), at session and transport level (*Secure Conversation* and *SecureTransport*); they are based on the implementation of WS-Security and WS-SecureConversation specifications [1].

As for the Authorization mechanisms, by default GT4 offers only simple mechanisms: (i) the basic *GRID mapfile* that statically assign to each GRID user an existing O.S. user with its group, (ii) *Embedded PDP*, in which it is possible to define a per-container, a per-service or per-resource mechanism handled by a Local Policy Decision Point (PDP).

However, GT4 also offers a set of APIs to integrate an external PDP, as XACML [12], to support more expressive authorization policies. In a *cloudgrid* archtecture, as the one implemented in *PerfCloud*, the default security solutions offered by Globus do not meet all the security requirements we have outlined above. The first limit is related to the adoption of a basic authentication path, which limits the access only to users authenticated internally to the Virtual Organization. We enriched this solution adopting an extended path validation that supports a federated approach, as described in [3]. Moreover, we extended the authorization mechanism in order to support XACML, and forced the security descriptor to adopt secure communication channels for all the cloud-related service and resources. The main drawback of this approach is an inevitable performance penalty, as complex authorization mechanisms or secure channels may heavily increase the platform overhead. The next section will evaluate quantitatively the actual impact of these choices.

4 Overhead Evaluation

The goal of the following analysis is to evaluate the overhead introduced by the multilayered *cloudgrid* architecture by performing measurements on *PerfCloud*. We have developed a synthetic PerfCloud *NULL* service, which just sends a reply when invoked. The response time measured in several working conditions (corresponding to the different security policies discussed in the previous section) will provide insight on the delay introduced by the *PerfCloud* middleware, unrelated to the service invoked and to the actual tasks to be performed on the target environment.

In order to understand how *PerfCloud* (and the *cloudgrid* approach) affects performance, we measured the response time of the above described target service on both physical and virtual clusters. The comparison between the figures obtained will make it possible to ascertain if the use of a GRID made up of both physical and virtual resources is a reasonable solution, or if the performance penalties incurred are too high.

4.1 Performance Evaluation Methodology

A complex system as the one described, in which different factors are involved in its performance, lends itself well to the *full factorial design* of experiments [7]. The selected factors and their N possible values are shown below:

- **Resource** (N=2), can be *Physical* or *Virtual*
- **Channel (transport)** (N=2), can be *HTTP* or *HTTPS*
- **Auth** (N=3), can be *None*, *Conversation* or *Message*
- **Authz** (N=2), can be *None*, *MapFile* or *XACML*

All the tests were performed in a dedicated environment, where the only source of load is a measurement client invoking the *NULL* service. This client repeats the test 50 times. Then we evaluated the mean, the standard deviation and the confidence interval. The session was repeated three times, and we chose the session with the lowest standard deviation, discarding the remainder.

According to the well-known methodology presented in [7], the next step of the analysis was the definition of a model taking in consideration the *main effects* (the contributions that come from a single factor). Then the *interaction effects* (the contributions that come from the mutual interaction of factors) were gradually introduced, until the evaluated error (meant as the difference between the measured value and the value predicted) could be considered negligible. In the next subsection, we will describe the construction of such model. It should be explicitly pointed out that this measurement technique does not allow to test system behavior under real workloads, as it corresponds to an *ideal* condition, where only the target service uses the target resource. However, this solution has the advantage to be easily reproducible, and so to perform all the tests under exactly the same conditions.

4.2 Experimental Results and Model Evaluation

Table 1 summarizes the performance results obtained for all the possible configurations. Each cell in the table reports the mean value and the confidence interval (in square parenthesis) of the response time, in milliseconds. In order to give an interpretation of the above presented results, we derived the following prediction model that corresponds to the DOE used [7]:

$y = \mu + R_i + C_j + Auth_k + AuthZ_h + Auth - C_{jk} + e_{ijkh}$

where i=(Physical,Virtual), j=(HTTP, HTTPS), k=(None, Conversation, Message), h=(None, MapFile, XACML), and μ is the global mean (i.e., the average

Table 1. Response times for the different configurations (ms)

		Physical Cluster		
AuthZ	Transport	None	Conversation	Message
None	HTTP	106 [102.61, 109.39]	370 [358.51, 381.49]	515 [497.33, 532.67]
	HTTPS	263 [256.50, 269.50]	853 [837.60, 868.40]	612 [590.02, 633.98]
MapFile	HTTP	107 [103.86, 110.14]	367 [356.49, 377.51]	508 [490.72, 525.28]
	HTTPS	262 [255.94, 268.06]	859 [843.31, 874.69]	617 [595.12, 638.88]
XACML	HTTP	108 [104.81, 111.19]	369 [358.87, 379.13]	521 [504.50, 537.50]
	HTTPS	266 [259.67, 272.33]	866 [850.20, 881.80]	627 [604.06, 649.94]
		Virtual Cluster		
AuthZ	Transport	None	Conversation	Message
None	HTTP	85 [81.86, 88.14]	328 [317.06, 338.94]	456 [438.67, 473.33]
	HTTPS	238 [231.96, 244.04]	797 [780.52, 813.48]	556 [535.59, 576.41]
MapFile	HTTP	86 [82.48, 89.52]	322 [311.42, 332.58]	453 [434.33, 471.67]
	HTTPS	239 [232.88, 245.12]	797 [780.35, 813.65]	556 [534.52, 577.48]
XACML	HTTP	86 [82.88, 89.12]	321 [310.09, 331.91]	452 [435.36, 468.64]
	HTTPS	245 [238.58, 251.42]	799 [782.67, 815.33]	563 [538.98, 587.02]

of all values). R, C, $Auth$ and $AuthZ$ (respectively Resource, Channel trasport protocol, Authentication security protocol and Authorization mechanism) are the independent factors that we decided to take into account in the model. They appear in the equation for y, according to the factor-level under evaluation. We will consider all the possible configurations described in the previous section with $R_{Physical}$ and $R_{Virtual}$, C_{HTTP} or C_{HTTPS}, and so on. These factors are all independent, and represent the *main effects* of the prediction model. $Auth - C$ is the contribution to the model that comes from the interaction between the Authentication security protocol and the Trasport protocol, and represents the way in which such coupling affects the mean. So, we will evaluate the different combinations $Auth - C_{None,HTTP}$, $Auth - C_{Conversation,HTTP}$ and so on. They are part of the so-called *interaction effects* of the prediction model. Finally, e is the error introduced by the model, and it depends on all factors taken in consideration. The other interactions (for example, $Auth - AuthZ$ or $Auth - R$) were computed, but not included in the model, because of their negligible impact. In fact, the model that includes only $Auth - C$ *explains for* the 99.85% of the measured values, as shown in Table 2. For brevity's sake, we present below just the results of the ANOVA analysis of the obtained model (Table 2). In this table we have reported for every effect the sum of the squares of all values (second column) and the variation of y as it is *explained* by the different effect. For example, in the considered model the main effect Channel *explains* the model for the 26.99% while the factor Authentication (Auth) *explains* the model for the 59.60%. In conclusion, the correctness of the prevision model is supported by the value of the error variation, which is under 1%. Further analysis, both visual and statistical, whose details are not shown here for brevity, shows that the residuals are uniformly distributed around zero, have homogeneous variance and that there is no polarization of the results.

Table 2. ANOVA table

Component	Sum of Squares	Variation (%)
Resource (R)	927,068.06	0.91%
Channel (C)	27,565,312.50	26.99%
Authentication (Auth)	60,868,636.11	59.60%
Authorization (AuthZ)	6,211.11	0.01%
Auth-C Interaction	12,606,925.00	12.34%
Errors	149,862.50	0.15%

4.3 Performance Considerations

The above model points out clearly the fundamental sources of overhead in a fully functional *cloudgrid* solution. The first result obtained, which was not completely unexpected, is the low impact of virtualization on overhead. In practice, there is no difference (in terms of security overhead) between a service hosted on a physical cluster and on a virtual one. In fact, the ratio of variation, explained by the factor R, is 0.91%. Moreover, looking at the results in Table 1, we can note that the invocation of a service on a virtual resource performs even better

than the corresponding request on a physical one. This counterintuitive behavior is due to the fact that modern virtual engines introduce very small overhead (in our test, we used Xen 3, which declares less than 5% overhead). This can be compensated and sometimes overcome by the performance gain obtained by OS noise reduction. The virtual images, being targeted to a specific appliance, can exclude generic OS management services, which are instead needed in a general-purpose physical environment. It should noted, however, that we have implemented a service that does not stress much the system and the virtualization environment. For other kind of services, e.g., bandwidth-intensive ones, virtualization could become the bottleneck. A performance comparison between virtual and physical clusters is out of the scope of this paper.

Also the introduction of advanced authorization mechanisms as XACML, needed to deal with the different roles of the users, surprisingly, does not have a great effect on the overall system performance. The *AuthZ* factor simply does not affect the overhead, as shown by the ANOVA Table (less than 0.01% of variation explained). This can be explained by considering that even the basic GSI authorization mechanism needs to instantiate Java objects to take an authorization decision. As a result, the time required to invoke external authorization services is dwarfed by the time necessary to load the chain of GSI objects.

Unlike the previous factors, the Channel transport and the Authentication security protocol affect heavily system performance, both on virtual and physical resources. The model analysis shows that the introduced overhead depends on two factors: the adoption of security protocols at transport layer, i.e., the choice between HTTP and HTTPS, and the security protocols adopted at message layer, i.e., the cryptography applied to the XML-based SOAP message. Both factors have a great impact (with a variation on the model of about 30% and 60%, respectively). Also their interaction has a high impact (about 10%).

Considered that the measured overhead shows a minimum value of 85 ms and a maximum value of 866 ms (a value about 10 times higher), the parameters setup in the Security Descriptors should be performed with great care. From a security point of view, we can note that security at transport and security at message layer are independent of one another. They both aim at granting confidentiality and integrity of information, and they can reach the same result in different ways. The secure transport layer (HTTPS) has a lower impact on performance and, in terms of security, is considered equivalent to the message layer one. So the best solution is to disable completely the message layer security. When this is not possible, it should be avoided to enable both the security layers.

Some interesting considerations can also be done on the interaction of the two factors. Analyzing the result table (Table 1), it is possible to note that, enabling the transport layer, the performance of `Secure Conversation` (which offers a lower security level) becomes worse than the performance offered by the `Secure Message` approach. So, if for any reason both Transport and Message layer security must be enabled, the lowest overhead can be obtained by enabling the message level security protocol, which also offers the highest security level.

5 Conclusions and Future Work

In this paper we have considered the security issues linked to the use of a *cloudgrid* approach. We showed through extensive experimentation on our prototype *PerfCloud* that, in contrast with intuition, the *cloudgrid* approach does not introduce perceptible overhead, notwithstanding the presence of a virtualization layer and the need for complex authorization mechanisms. We have also proposed a trade-off analysis between performance and security.

Our plans for future research include the extension of the proposed analysis to the case of federated identities, which we outlined in another paper [3] as another necessary component in a complete security infrastructure for cloud and GRID integration.

References

1. Atkinson, B., et al.: Ws-security specification, web services security 1.0 (April 2002)
2. Casola, V., Lettiero, R., Rak, M., Villano, U.: Access control in cloud on grid: the perfcloud case study. To appear in the Proc. of SPCC 2010 (2010)
3. Casola, V., Rak, M., Villano, U.: Identity federation in cloud computing. In: Proc. of SPEDA 2010, Atlanta (August 2010)
4. Cherkasova, L., Gupta, D., Vahdat, A.: Optimizing grid site manager performance with virtual machines. In: Proc. of the 3rd USENIX Workshop on Real Large Distributed Systems (WORLDS 2006) (2006)
5. Cloud Security Alliance: Security guidance for critical areas of focus in cloud computing (2009)
6. Foster, I.T., Freeman, T., Keahey, K., Scheftner, D., Sotomayor, B., Zhang, X.: Virtual clusters for grid communities. In: CCGRID, pp. 513–520. IEEE Computer Society, Los Alamitos (2006)
7. Jain, R.: Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling. Wiley Computer Publishing John Wiley & Sons, Chichester (May 1991)
8. Mancini, E.P., Rak, M., Villano, U.: Perfcloud: Grid services for performance-oriented development of cloud computing applications. In: Reddy, S. (ed.) WETICE, pp. 201–206. IEEE Computer Society, Los Alamitos (2009)
9. Mell, P., Grance, T.: The nist definition of cloud computing (2009)
10. Purdue University: Wispy project (2009), `http://www.rcac.purdue.edu/teragrid/resources/#wispy`
11. The Globus Security Team: Globus toolkit version 4, security: Authorization framework (2005), `http://www.globus.org/toolkit/docs/4.0/security/authzframe/`
12. The OASIS technical commitee: Xacml: extensible access control markup language (2005), `http://www.oasisopen.org/committees/xacml/repository/`
13. University of Chicago: Nimbus project (2009), `http://workspace.globus.org/clouds/nimbus.html`