Grant Agreement Number: 257528

KHRESMOI

www.khresmoi.eu

Evaluation of the 'Early Cloud Infrastructure' and specification refinement for the 'Full Cloud Infrastructure'

Deliverable number	*D6.4.2*
Dissemination level	*Public*
Delivery date	*9 Nov 2012*
Status	*Final*
Author(s)	*Iván Martínez Rodríguez, Miguel Ángel Tinte García*

This project is supported by the European Commission under the Information and Communication Technologies (ICT) Theme of the 7th Framework Programme for Research and Technological Development.

Executive summary

Deliverable D6.4.2 of the KHRESMOI project describes evaluation of the "Early Cloud Infrastructure" and specification refinement for the "Full Cloud Infrastructure". Based on the evaluation results of the "Early Cloud infrastructure" deployment, a full refinement of the infrastructure specification is delivered to satisfy the large-scale requirements and improve performances.

A metric is always used to measure and understand the behaviour of software or a program. To evaluate the quality of cloud, there is a need to assess and analyse its design and implementation using appropriate metrics.

Unlike testing conventional web-based software, testing clouds and cloud-based software has several unique testing quality assurance objectives, requirements, and distinct features.

Besides this, our system presents other specific features that must be taken into account during metrics definition. Concretely, we are not evaluating just a SOC system but we are evaluating also SCA Integration Services and KHRESMOI Back-End Services performance. Therefore, the test type's definition will have to consider all these premises to achieve an optimal evaluation.

Besides this, the report includes:

- A description of Evaluation Approach defined for testing the KHREMSOI Cloud Infrastructure based on five steps.
- A list of Metrics to assess and analyse the Cloud design and implementation.
- A set of Facets based on the initial requirements specified for Cloud Computing environment which are: Scalability, Elasticity and Availability.
- Finally, a definition of the four main Scenarios identified in the project which have been refined during second year of the by means of the requirements analysis and specification activities within WP8 and WP9.

The report presents the results of the evaluation applied on the first implementation of the "Early Cloud Prototype". The objective consists of enriching the first Cloud Infrastructure specification and providing more specific recommendations for the following developments, specifically with regard to requirements for large-scale performance.

To conclude we can highlight that two bottlenecks have been identified in the current Cloud Infrastructure by means of the analysis results, which are:

- CPU Usage, and
- Average Response Time

To prevent that these bottlenecks appear in the system, the approach of "Full Cloud Infrastructure" proposed in [1] should be performed taking into account the Specification Refinement provided in Section 11. Specification Refinement shows the new requirements in terms of HD and SW resources that will be needed for deploying the KHRESMOI Full Cloud

Table of Contents

1 List of abbreviations

ART	Average Response Time
CPU	Central Processing Unit
DoW	Description of Work
QMS	Query Mapping Service
SCA	Service Component Architecture
SOA	Service Oriented Architecture
SOC	Service Oriented Cloud
URI	Uniform Resource Identifier
URL	Universal Resource Locator
VM	Virtual Machine

Table 1. Abbreviations and acronyms.

2 List of figures

3 List of tables

4 Introduction

4.1 Introductory Explanation of the Deliverable

The goal of Task 6.3 "System scaling", in which the Cloud Evaluation task is contextualized, is to scale the system which was previously specified and implemented by means of the KHRESMOI SOA.

Scaling the system depends on the components, processes involved, but primarily on the architecture design principles and their application in practice through the system integration. The success of the project depends heavily on the scale the resulting system is capable of covering. The task of scaling up the system will focus on iterative scale-up cycles involving evaluation and improvement of the key characteristics of the system with formal progress criteria.

In [1], we described the approach followed for the definition and deployment of the Early Cloud Prototype giving solutions satisfying the system-scaling requirement. The work reported in this document describes the evaluation process of the "Early Cloud Infrastructure". Analysing the results and the end of the evaluation process led us to update and refine the initial cloud requirements in order to provide an updated specification for the "Full Cloud Infrastructure".

4.2 Purpose and Audience

4.2.1 Purpose

The purpose of this deliverable is to enrich the "Early Cloud infrastructure" specification [1] and provide a full refinement of the infrastructure specification to satisfy the large-scale requirements and improve performances.

4.2.2 Audience

This deliverable is relevant to all technical work packages in KHRESMOI (WP1-WP9). The target audience includes component providers, users, and any person inside or outside of the KHRESMOI project interested in learning about the internal processing of the KHRESMOI Cloud Infrastructure. As such this deliverable presents the results of the evaluation applied on the first implementation of the "Early Cloud Prototype".

4.3 Structure of the Document

This deliverable is organized as follows: Section 5 describes the Evaluation Approach followed by the evaluation of the "Early Cloud Infrastructure". After the evaluation approach, Section 6 describes a set of key metrics needed for an empirical evaluation of the KHRESMOI cloud. Section 7 takes into account several facets aligned with initial KHRESMOI cloud requirements which are: scalability, elasticity and availability. Section 8 provides a description of the main test scenarios that have been defined according to initial WP8 and WP9 requirements. Section 9 presents the results report collecting the values returned for each one of the cloud metrics defined in Section 6. Section 10 analyses how results affect to cloud facets, and finally in Section 11, the conclusions of the deliverable are presented.

5 Evaluation Approach

According to Wikipedia[1], "cloud testing is a form of software testing in which Web applications that leverage Cloud computing environments ("cloud") seek to simulate real-world user traffic as a means of load testing and stress testing web sites". Trying to align KHRESMOI cloud evaluation with this principle we propose an evaluation approach based on five steps as shown in Figure 1. The first step in the evaluation approach followed is the definition of a set of metrics and facets described in detail in Sections 6 and 7 respectively.

The second step is the definition of a list of test scenarios on which to calculate the metrics defined (Section 8). The next step is the evaluation and selection of one tool to performance cloud test scenarios described in the previous step. We based this decision on the comparison view about cloud testing products, solutions and services provided in [3]. We decided to use SOASTA Cloud Lite Software[2] as the tool for supporting the cloud test execution and monitoring.

This software allows us to make several types of test over the KHRESMOI cloud platform and therefore to extract some useful metrics that will be taken into account for the next cloud developments and evaluation. Besides this, tests will be defined to closely resemble a real scenario with real KHRESMOI end users. The next information provided by the project DoW has guided us to configure these scenarios: *"Representative groups of end users are available for sizable evaluations, accessed through a medical search engine with 11000 queries per day, a professional association of 2700 medical doctors, and two radiology departments with 175 radiologists."*

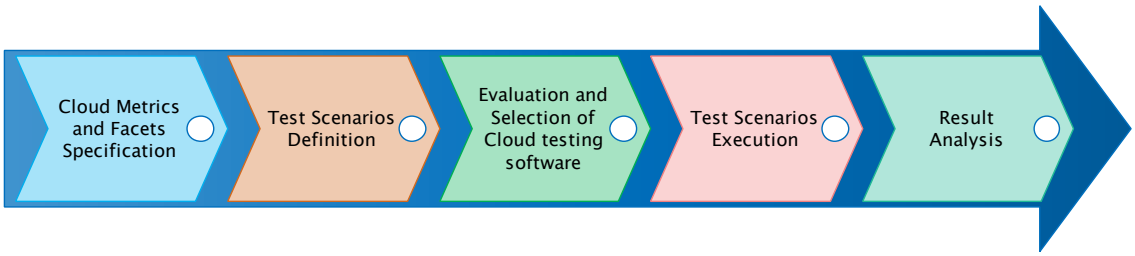

Figure 1. Evaluation Approach.

In this prototype phase, this information is worth to be used as an example in order to assess the system. Therefore we can extract the next information about an approximated amount of users and queries from previous paragraph:

- The number of users per unit of time is:
  - 2875 users / day
  - 119,7 users / hour ~ 120 users / hour
  - 2 users / min
- The number of queries per user is:
  - 11000 queries / day divided by 2875 users / day
  - 3,82 queries / user ~ 4 queries / user

[1] http://en.wikipedia.org/wiki/Cloud_testing

[2] http://www.soasta.com/products/cloudtest-lite/

Hence, we can use this information to define the next parameters in order to assess the system in a concurrent using scenario. We need to take an estimated time of execution and concrete how many users can be accessing to the system at the same time:

- **5 minutes interval**
- **60 virtual users following a ramp type time function (we can guess as the worst case than half of the users per hour meet during this interval time)**
- **4 queries / user**

These are the parameters that will be used to test the cloud platform performance and to extract some valuable metrics for next cloud iterations.

Final steps regarding to execution of Tests scenarios and Results analysis are described in depth in Sections 9 and 10.

6 Cloud Metrics

To evaluate the quality of the cloud, there is a need to assess and analyse its design and implementation using appropriate metrics. Due to this and taking into account the metrics that the SOASTA Cloud Test is able to calculate we can see in Figure 2 each one of the metrics taken into account for the "Early Cloud Prototype" evaluation.

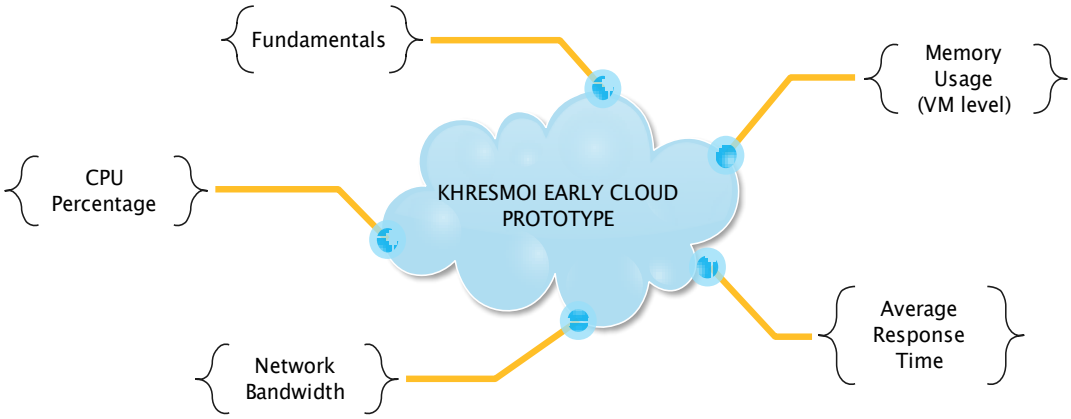

Figure 2. KHRESMOI Cloud Metrics.

We present in the next subsections each one of the metrics to be considered.

6.1 Fundamentals

This metric provides the fundamental test information, including lapsed time, number of messages sent, and error count.

6.2 CPU Usage

The percentage of CPU usage indicates how much of the processor's capacity is currently in use by the system. When the CPU usage reaches 100% there is no more spare capacity to use for running other programs. When the percentage of CPU usage begins to max out at 100% additional action may need to be taken, as for instance on demand VM provisioning based on performance models defined for the VM overloaded.

6.3 Memory Usage

Memory Usage can be measured at the host level, the VM level, and as granted memory. We consider only memory usage at VM level. Memory is allocated to the VM only when a final user uses it. Once allocated, the used memory segment is considered "granted." Most systems cannot (or will not) reclaim memory, and there is little point in trying, because most guests use any extra for buffers and caches and never relinquish what has been "granted." Therefore, we can know a VM's physical memory usage by looking at granted memory.

6.4 Average Response Time

In the current cloud based system, we will consider that the system response time is the interval between the receipt of the end of transmission of an inquiry message and the beginning of the transmission of a response message to the station originating the inquiry.

6.5 Bandwidth Usage

In our context we can define bandwidth usage as total bytes sent plus received for each period of the test. Only messages that are successfully sent are included in the final value.

7 Cloud Facets

Considering the initial requirements specified for Cloud Computing environment, the solution deployed for supporting the "Early Cloud Prototype" provides a set of important characteristics from which we can highlight the following facets (see Figure 3):

- Scalability: the capability to trigger many tasks together, and dispatch them on the cloud,
- Elasticity: the capability to increase the task resources according user needs,
- Availability on demand: the capability to always provide resources to the users.

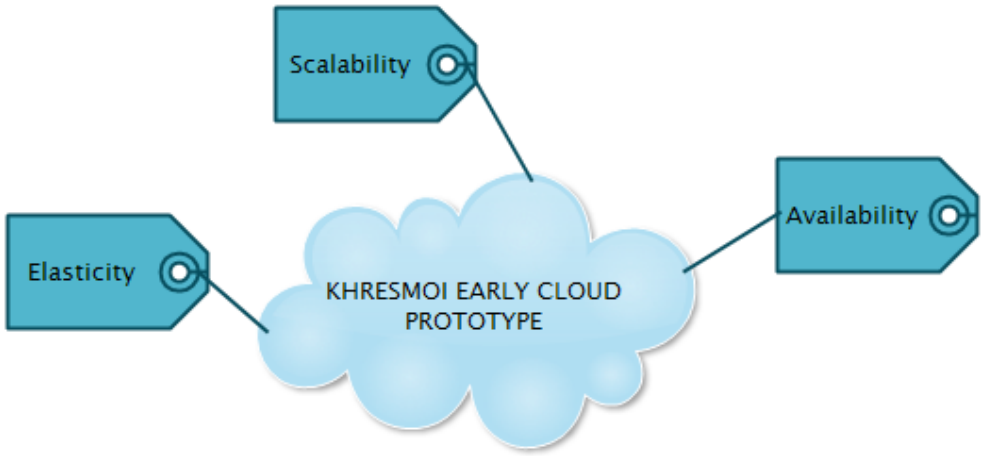

Figure 3. KHRESMOI Cloud Facets.

Next we provide a description of each one of these facets considered in the evaluation process.

7.1 Scalability

This facet is a desirable property of a system, which indicates its ability to either handle growing amounts of work in a graceful manner or its ability to improve throughput when additional resources (typically hardware) are added. A system, whose performance improves after adding hardware, proportionally to the capacity added, is said to be a scalable system.

7.2 Elasticity

This facet is one of the major factors for the success of the cloud as an IT infrastructure. We take as reference the definition provided in [5] according to which "Elasticity, i.e. the ability to deal with load variations by adding more resources during high load or consolidating the tenants to fewer nodes when the load decreases, all in a live system without service disruption, is therefore critical for these systems. Even though elasticity is often associated with the scale of the system, a subtle difference exists between elasticity and scalability when used to express a system's behaviour".

7.3 Availability

Cloud Services should be available for a maximum time. As it is mentioned in [5] “the on demand, elastic, scalable, and customizable nature of the cloud must be considered when deploying cloud architectures. Many different clients might be accessing the same back-end applications, and many providers that are providing the cloud services have the expectation that only their application will be properly delivered to users. In cloud computing it is essentially required to gather the information instantly without making a user to wait and the gathered information should be related to each other”.

8 Test Scenarios Definition

The main test scenarios presented in this document remain the same as in the previous deliverable [2] related to Architecture Evaluation. They have been refined after two years of project based on the requirements analysis and specification activities within WP8 and WP9. Therefore, the same test scenarios structure will be adopted in order to fulfil the objectives of the evaluation process. Despite this, some minor changes have been done to these scenarios, in order to update the workflows to changes done during the last year and to update some components that have been replaced by some partners during this time. In these cases, the test scenarios workflows have been updated to new component functionalities.

8.1 Textual Search Scenario

This scenario represents one of the main workflows in the KHRESMOI project and uses most of the components created by all partners. From the previous Textual Search workflow, we have changed the Spell Checking service which is no longer Wrapin and also added some new functionalities are shown in Table 2. The Textual Search workflow contains the following components, steps and functionalities:

Prototype 1	**Textual search**	
Components	ezDL (UDE)	ezDL is a multi-agent search system for heterogeneous data sources and a tool-set for building search user interfaces to support complex tasks
	Speller (HON)	HON's medical multilingual spell checking service
	Multilingual Translator (CUNI)	CUNI Multilingual translator service
	Disambiguator (ONTO)	ONTO Disambiguation service
	QMS (USFD)	Given a string of words and concept URIs, the service allows you to generate a Mimir query
	Mimir (USFD)	Mímir search Web Service
	TextManager (AtoS)	This component manages the complete Textual Search Workflow that integrates the other components and its iterations
Scenario	Step1 : search(keywords)	The user introduces a list of keywords through the UI in order to obtain a proper answer. In this version, the system allows also to add some constraints in order to adjust the search.
	Step2 : improve query	The user obtains some improvements to the query keywords performed by the system: spelling correction, possible language translation and disambiguation.

	Step3 : perform search of final query	The query is divided for different topics and the search is performed
	Step4 : return list of Documents	The user receives a list of documents with hits found as the answer of the search
	Step5 : view document	The user can see and translate any document of the list returned from the search
Main functionalities to be integrated	Functionalities used by User: from ezDL	search(List<Keywords>)
	Functionalities used by User: from ezDL	viewDoc(docURI)
	Functionalities used by ezDL: from Speller	List<Suggestion> :getSpelling(keywords,lang)
	Functionalities used by ezDL: from MT	translateQuery(text,docType,sourceLang,targetLang,profile)
	Functionalities used by ezDL: from MT	translateDoc(docURI)
	Functionalities used by ezDL: from Disambiguation	List<Label> :getDissambiguation(keywords)
	Functionalities used by ezDL: from TextManager	searchByText(userProfile,keywords)
	Functionalities used by TextManager: from QMS	List<Query> :queryMapping(id,keyword)
	Functionalities used by TextManager: from Mimir	postQuery(index_id,queryString) hitCount(index_id,query_id) hits(index_id,query_id,start_index,count) docMetadata(index_id,query_id,doc_id) docText(index_id,query_id,rank,term_pos,length)

Table 2. Textual Search Scenario Definition.

8.2 2D Image Search Scenario

The Image Search workflow in KHRESMOI can be divided into two different sub-workflows: 2D Image Search and 3D Image Search.

For 2D Image Search we can observe in Table 3**Error! Reference source not found.** the list of components, steps and functionalities. The main change with respect to the last version of the scenario is that the previous image retrieval service (GIFT) has been replaced by ParaDISE:

Prototype 1	**2D Image search**	
Components	ezDL (UDE)	ezDL is a multi-agent search system for heterogeneous data sources and a tool-set for building search user interfaces to support complex tasks
	ParaDISE (HEVS)	HEVS content-based image retrieval service
	SCA-ParaDISE (AtoS)	SCA Component for ParaDISE
	Repository	Image repository
	ImageManager (AtoS)	This component manages the complete Image Search Workflow that integrates the other components and its iterations
Scenario	Step1 : search(images)	The user introduces a list of images through the UI in order to obtain a proper answer
	Step2 : the user obtains a list of images	The user obtains a list of images after the search processing ranked by a predefined score
Main functionalities to be integrated	Functionalities used by User: from ezDL	search(List<Image>)
	Functionalities used by ezDL: from ImageManager	get2DSimilarImages(List<ImageScore>,List<ImageCollection>, userProfile)
	Functionalities used by ImageManager: from SCA ParaDISE	searchBySimilarity(captionQuery, relevantImages, irrelevantImages)

Table 3. 2D Image Search Scenario Definition.

8.3 3D Image Search Scenario

The 3D Image Search workflow continues being one of the main important scenarios for KHRESMOI end-users. In this version, the image retrieval component MUW has been updated with new methods in order to improve the functionalities required. In Table 4 we can see the current workflow in terms of components, steps and functionalities:

Prototype 1	**3D Image search**	
Components	ezDL (UDE)	ezDL is a multi-agent search system for heterogeneous data sources and a tool-set for building search user interfaces to support complex tasks
	MUW (MUW)	MUW 3D image retrieval web service
	SCA-MUW (AtoS)	SCA Component for MUW
	ImageManager (AtoS)	This component manages the complete Image Search Workflow that integrates the other components and its iterations
Scenario	Step1 : search(images)	The user introduces a list of images through the UI in order to obtain a proper answer
	Step2 : the user obtains a list of thumbnails from images	The user obtains a list of thumbnails from images similar to search performed
	Step3 : the user selects an image	The user selects an image thumbnail in order to obtain the full image
Main functionalities to be integrated	Functionalities used by User: from ezDL	search(Image,Text)
	Functionalities used by ezDL: from ImageManager	3DsimilarImages(image,text)
	Functionalities used by ImageManager from SCA MUW	getImage(String id)

Table 4. 3D Image Search Scenario Definition.

8.4 Multilingual Textual Search Scenario

This scenario supports the dynamic translation of query text when it is introduced into the ezDL interface. This allows the user to find out terms in different languages that can improve the original query. The Multilingual Textual Search will be considered as a special type of Textual Search workflow where the query text performed can be translated during the search. This workflow, shown in Table 5, is composed by the next components, steps and functionalities:

Prototype 1	**Multilingual query translation**	
Components	ezDL (UDE)	ezDL is a multi-agent search system for heterogeneous data sources and a tool-set for building search user interfaces to support complex tasks
	Multilingual Translator (CUNI)	CUNI Multilingual translator service
	SCA-MT (AtoS)	SCA Component for Multilingual Translator
Scenario	Step1 : translate(keyword)	The user introduces keyword through the UI in order to obtain a proper answer
	Step2 : the user obtains dynamically a list of possible translations	The user obtains a list of translations in a different language for the word introduced
	Step3 : the user selects a word	The user selects a word as a translation of the keyword entry
	Step4: Textual Search workflow	The rest of the scenario is the same as Textual Search workflow
Main functionalities to be integrated	Functionalities used by User: from ezDL	search(Image, Text)
	Functionalities used by ezDL: from SCA-MT	Translate(keyword)
	Functionalities used by SCA-MT: from Multilingual Translator	getTranslation(action, sourceLang, targetLang, text)

Table 5. Multilingual Textual Search Scenario Definition.

9 Results Report

This chapter aims to assess cloud evaluation metrics and facets defined in the previous sections. This assessment consists basically in testing different scenarios separately simulating real work scenarios based on main use cases workflows. Mainly, these tests will assess the capacity of the system to deal with many concurrent users and requests with reference to some metrics chosen in chapter 6. The metrics that will be held for these scenarios are those already defined in chapter 6:

- Fundamentals,
- CPU Usage (shows some periodically peaks consumption due to monitor service),
- Memory Usage,
- Average Response Time,
- Network Bandwidth
- Virtual Users.

These metrics aim to extract some useful values that will help us to evaluate the cloud system in three main facets which have been defined in Section 7 as critical for the system performance: scalability, elasticity and availability.

The following subsections show the results for different scenarios defined previously as well as the diagrams obtained from SOASTA Cloud Lite test software.

9.1 Textual Search Scenario Results

In this scenario, we assess the performance of the Textual Search scenario with parameters defined in chapter 8.1. The Textual Search workflow can be represented by one query example as a GET request to this URL:

http://khresmoicloud2.es.atos.net:8080/khresmoi-textual-search/rest/documents?query=diabetes

This query returns by default 20 documents with the text queried and is taken as a request from each virtual user in order to simulate concurrent access to the system. Also, the test includes an increasing load of concurrent users to know how the system responds to this type of request in order to extract some valuable conclusions about the system performance.

9.1.1 Fundamentals

"Fundamental metrics" represent the main statistics of the test composition as Elapsed Time, Messages and Actions requested, Status, Name of the results composition, etc. Moreover, there are two calculated metrics that may be useful for us:

- Average Response Time (ART) that shows the average response time per query as well as the maximum time and minimum per each query
- Effective Throughput, which aims to extract a value that represents how effective the system is at dealing with the requests. This value represents how many messages the system is able to respond in a specific time interval. In our case, it relies on the number of messages per second during the execution time of the test.

In this case, the fundamentals metrics for the textual search composition (Figure 4) shows how the total amount of Messages/Actions planned have been requested. Besides this, we can observe how Average response time value is not very high in a concurrent scenario with 60 virtual users in only 5 min time, which represents a scenario of high concurrence for our system. In any case, it should be improved in next iterations of cloud deployment, mainly in the next Full Cloud Prototype.

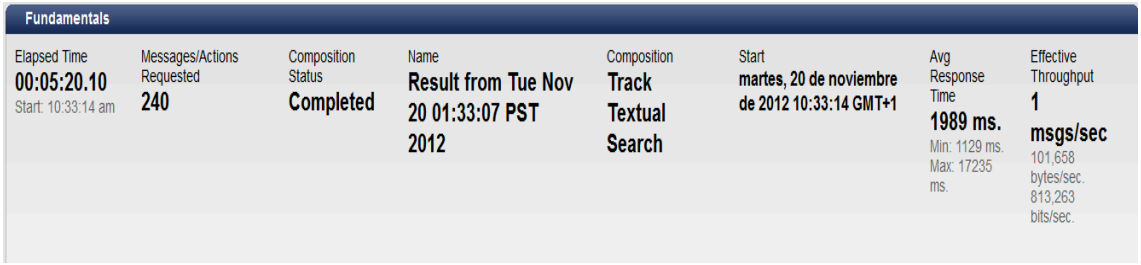

Figure 4. Textual Search fundamentals metrics.

9.1.2 CPU Usage

CPU Usage shows the percentage of CPU resources used during test composition. In this case, we can realize how CPU demand is quite high for server in charge of dealing with requests. Moreover, the peaks that can be observed can be justified because during some moments different requests overlap at the same time. We can see in Figure 5 the values obtained during the test composition execution to calculate the CPU usage:

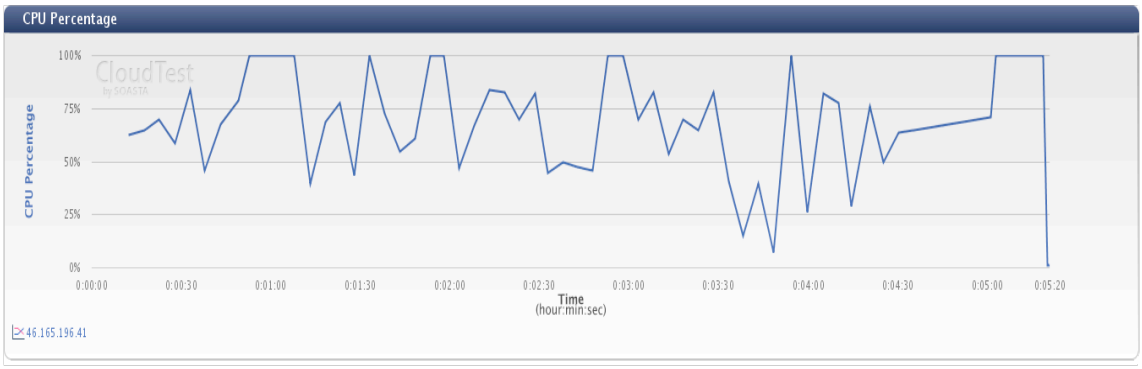

Figure 5. Textual Search CPU Usage diagram.

The main implications of these results are the probably next cloud deployments will require increase CPU resources in order to improve performance. The main performance problems in case of system overload by requests could appear as slow capacity to response and eventually in extreme cases by losing messages.

9.1.3 Memory Usage

Memory Usage or RAM Usage also shows how another different hardware system resource is consumed during test composition execution. In this case, we focus on memory usage in Giga bytes units so we can see in Figure 6 a very stable consumption about 1.4 GB.

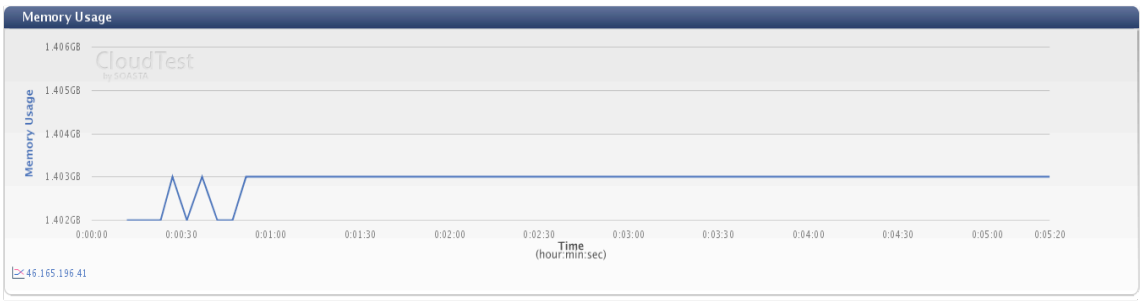

Figure 6. Textual Search Memory usage diagram.

We can conclude after this diagram that Memory consumption dedicated is enough currently for this scenario so it doesn't show peaks in its consumption. Anyway, RAM Memory for servers is one of the most important resources for dealing properly with requests and it is very important to check in next cloud deployments that user requirements are successfully satisfied.

9.1.4 Average Response Time

This metric show the ART for queries requested to the system. In this case, we can appreciate in Figure 7 how the system increases the response time in some parts of the execution because the number of requests is higher at this moment. Also we can correlate this metric with Virtual Users diagram because we can appreciate how concurrent users meet at the same time about 4:40 and ART directly increases at the same moment.

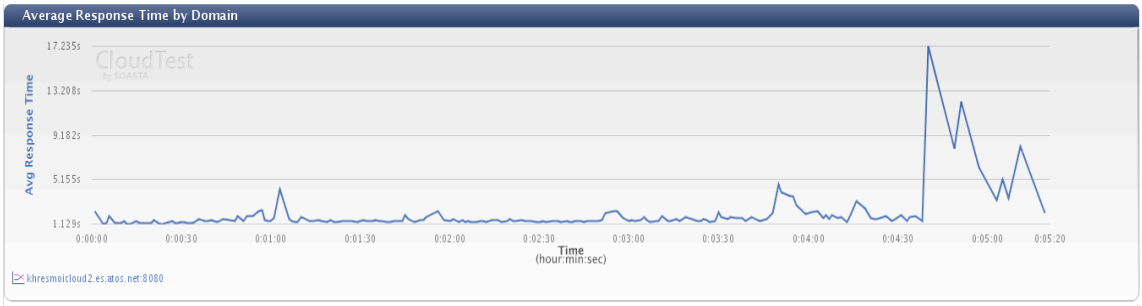

Figure 7. Textual Search Average Response Time diagram.

In next iterations of Cloud Deployment, we will need to ensure a good performance in terms of ART when several users meet using the KHRESMOI system. Mainly, these problems can be avoided by improving the hardware resources to the system for Full Cloud Prototype.

9.1.5 Network Bandwidth

Figure 8 shows the network bandwidth consumed during test composition execution. The diagram shows that bandwidth consume oscillates between 1.1 Mb and 2.2 Mb except at the end of time that it increases up to 4.3 Mb. This is due to there is a concurrent use of network bandwidth by several users and therefore the consume increases. It is important to have into account these cases for next iterations and try to ensure enough network bandwidth for all traffic may be generated. This could be implemented by increasing bandwidth resources or dedicating most existing resources to actives messages or actions.

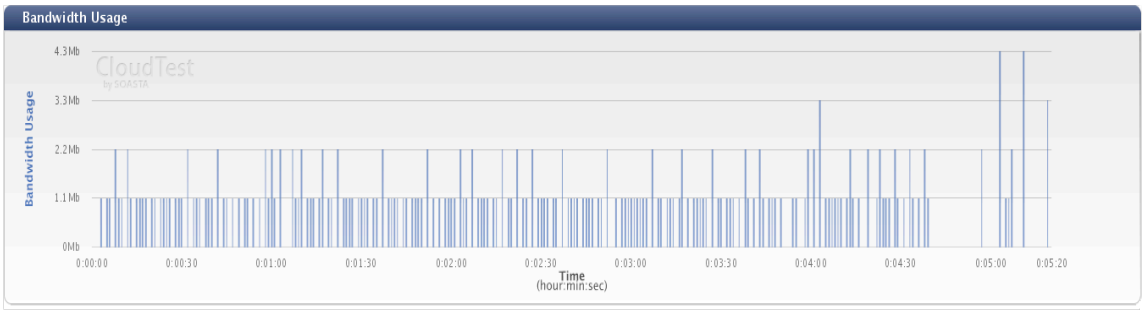

Figure 8. Textual Search Network bandwidth diagram.

9.1.6 Virtual Users

The virtual Users metric represents an approach to user access to the cloud platform. In Figure 9, we can see how users execute queries mostly individually but sometimes they overlap. This happens quite frequently but at the end is when more active users meet at the same time. This fact can be directly correlated to the network bandwidth usage diagram and observe how bandwidth consume increases considerably when users concur at the end.

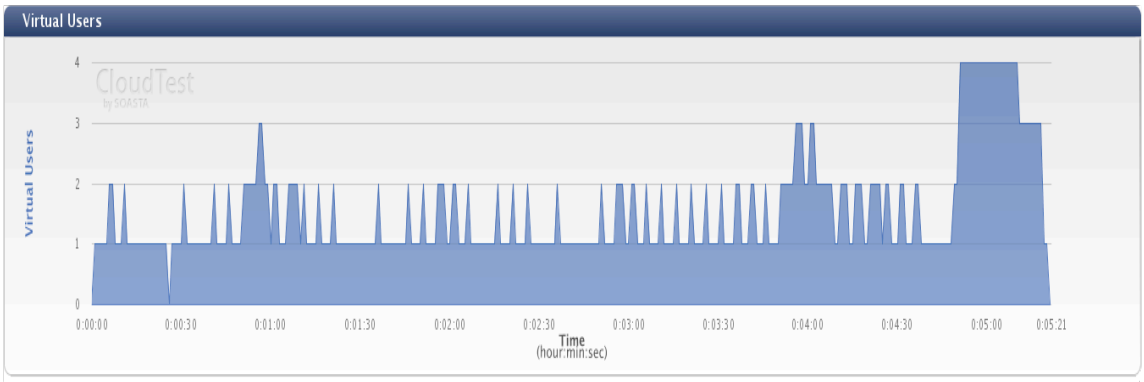

Figure 9. Textual Search Virtual Users diagram.

9.2 2D Image Search Scenario Results

In this scenario, we assess the performance of 2D Image Search scenario with default parameters defined at the beginning of the section. 2D Image Search workflow can be represented by query example as a GET request to this URL:

http://khresmoicloud2.es.atos.net:8080/khresmoi-image-search/rest/images/2D/similarImages?captionQuery=lactose&relevantImages=1475-2859-2-2-1

This query is taken as a request from each user in order to simulate concurrent access to the system. Also, the test includes an increasing load of users to know how the system responds to this type of requests in order to extract some valuable conclusions about the system performance. In this case, the 2D Image search workflow returns an image unlike previous scenario. Therefore, these metrics are worth to be compared to previous case in order to know how the system works with different types of data returned to the requests.

9.2.1 Fundamentals

Exactly as for textual search, "Fundamentals metrics" represent the main values of the test composition (Elapsed Time, Messages and action requested, status, Composition name, etc.). In Figure 10, we can observe a lower value for ART which means that requests and responses are delivered with a suitable velocity. Also, we can notice how Effective Throughput appears as a value of 1 which means that the system is able to deliver with at least one message per second, which implies an acceptable performance at this moment.

Figure 10. 2D Image Search Fundamentals metrics

9.2.2 CPU Usage

In Figure 11 we can identify some peaks in usage of this hardware resource that show how increase resource consumption as it nears the end of the test. This can be justified by the periodic monitor service which is in charge of extracting these metrics. Apart of this, we can realise how the CPU hardware does not show much consumption percentage, show we can assume currently 2D Image search workflow CPU requirements are currently covered.

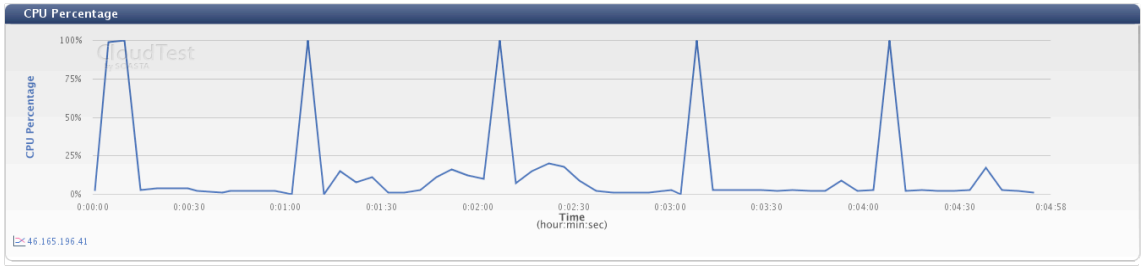

Figure 11. 2D Image Search CPU Usage diagram.

9.2.3 Memory Usage

Figure 12 shows how the memory usage keeps a stable performance during test life cycle. Similarly, we can observe how the amount of memory is very close to 1.4 GB during whole execution. This stable consumption can be considered as a good signal since the Memory Usage value can be affordable by the system.

Figure 12. 2D Image Search Memory Usage diagram.

Hence, currently we can conclude KHRESMOI Cloud Memory does not require extra resources for 2D Image search.

9.2.4 Average Response Time

The average response time's metric shows lower values that in Textual search response times so we can conclude that 2D Image search workflow is faster in this case. This is explained because of the different data types returned in each case. In Figure 13, we can appreciate how some average response times are even lower than 1 second in some parts of the execution.

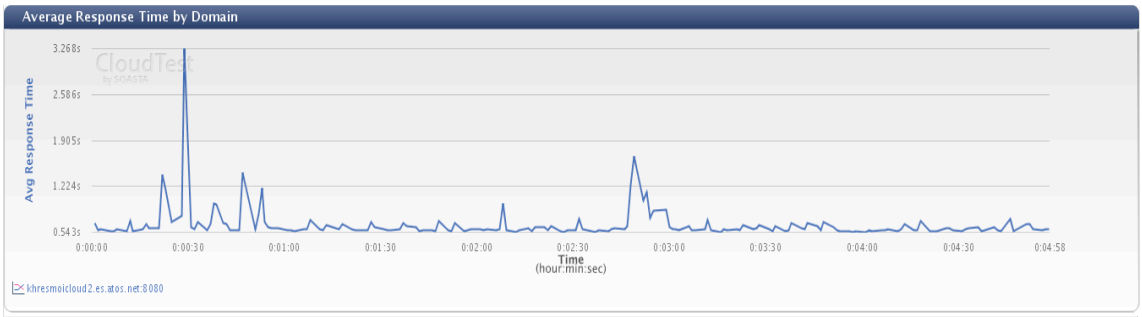

Figure 13. 2D Image Search Response Time diagram.

Also, we notice one peak around 00:30 when ART delays up to three seconds. This malfunction can be justified because of some users overlapping. In next Full Cloud Prototype, it would be necessary to ensure a good working just by adding more hardware resources to the system.

9.2.5 Network Bandwidth

As we can see in Figure 14, network bandwidth consumption in this case is not very high and can be easily correlated with the number of requests performed by the virtual users of the test composition. Also, we can realize how ART value for 25-35'' can be related with the higher network bandwidth consumed for the same interval.

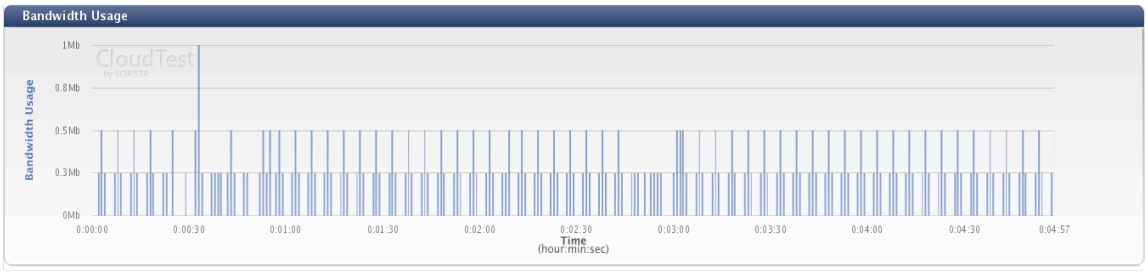

Figure 14. 2D Image Search Network Bandwidth diagram.

9.2.6 Virtual Users

In Figure 15, we can see how virtual users appear during test composition execution. This diagram also indicates that network bandwidth consumption is directly proportional to the existence of virtual users requesting to the system. Besides this, it is quite feasible correlate this diagram with Average response time and realise how virtual users overlapping at the beginning is affecting to response time directly.

Besides this, it is important to highlight that in this case there is not a curve of overlapped users because time execution is not long enough to provoke this situation. Anyway, this possibility will be taken into account for next Cloud deployments.

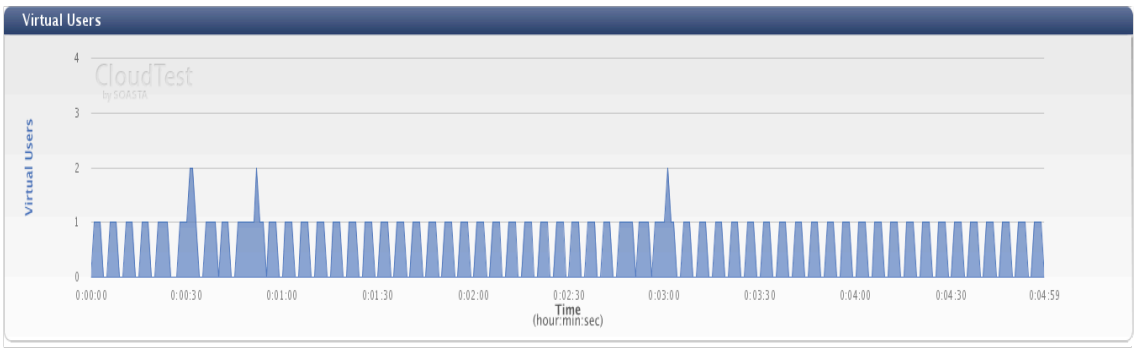

Figure 15. 2D Image Search Virtual Users diagram.

9.3 3D Image Search Scenario Results

Similarly to the previous scenario, on this one we assess the performance of the 3D Image Search workflow with default parameters defined at the beginning of the section. 3D Image Search workflow can be represented by a query example as a POST request to this URL, that simulates the search from some user for similar 3D anatomy images to the image given as input:

- http://khresmoicloud2.es.atos.net:8080/khresmoi-image-search/rest//images/3D/similarImages/anatomy
- Body request: "queryImageID":"ID_RA10001172134000_3_1"

This query is taken as a request from each user in order to simulate concurrent access to the system. Also, the test includes an increasing load of users to know how the system responds to this type of request in order to extract some valuable conclusions about the system performance. In this case, 3D Image search workflow returns a set of similar images IDs so now the system is working with JSON text instead of images and this should improve its performance.

9.3.1 Fundamentals

Figure 16 represents the fundamentals metrics for this scenario. Values obtained show an excellent Average Response Time as well as a faster and complete execution. The effective throughput metric shows a value of 1, so our system has been able to deal with at least one message per second.

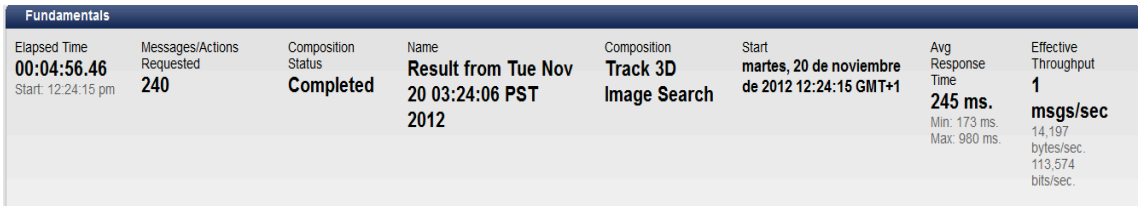

Figure 16. 3D Image Search Fundamentals metrics

9.3.2 CPU Usage

Figure 17 shows CPU usage which captures a small activity in terms of CPU requirements that can be justified by low complexity associated with the 3D image search workflow. The data type returned in this case (JSON response) contributes to provide this good value for CPU usage. As well as before, the peaks consumption can be due to monitor service.

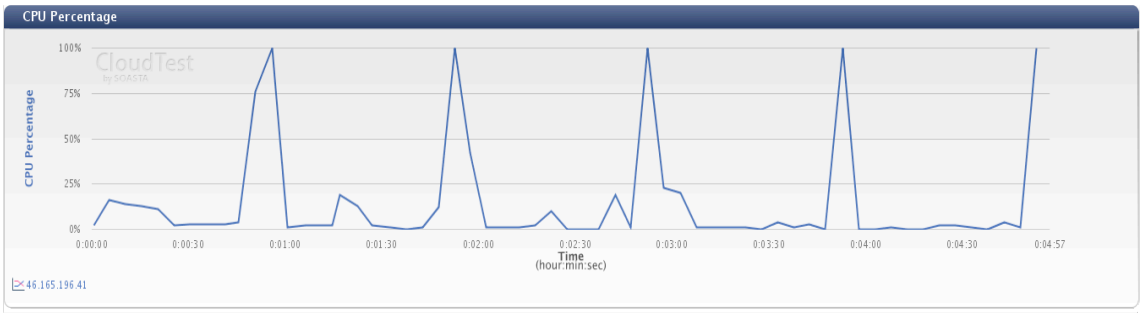

Figure 17. 3D Image Search CPU Usage diagram

9.3.3 Memory Usage

In **Error! Reference source not found.** we can see how Memory usage is quite stable, even though it is possible to appreciate how it is bigger at the beginning of the execution and starts to decrease after that. Anyway, the total amount of Memory consumed for the system during this test is always around 1,403 and 1,404 GB.

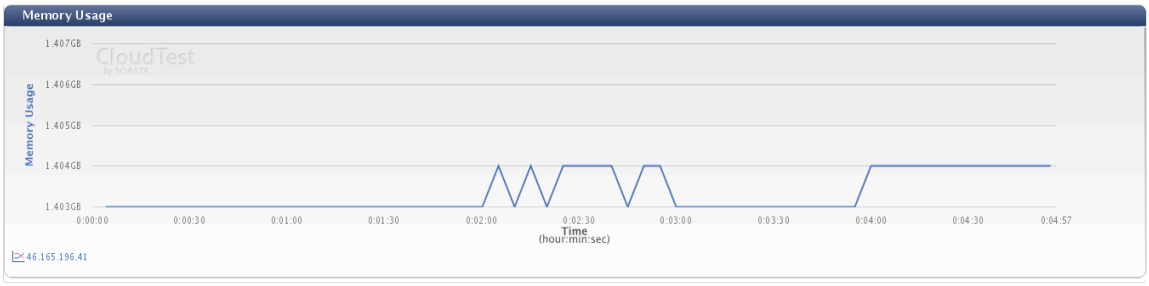

Figure 18. 3D Image Search Memory Usage diagram.

9.3.4 Average Response Time

Figure 19 shows ART obtained which is quite balanced during the whole execution except for a moment around 00:30. This can be explained by several reasons like an instant network traffic bottleneck or some process overlapping. In either case, the response time is always below one second, which in our Early Cloud prototype current status can be considered as an optimal value. Currently, any response time below one second can be defined as a good performance.

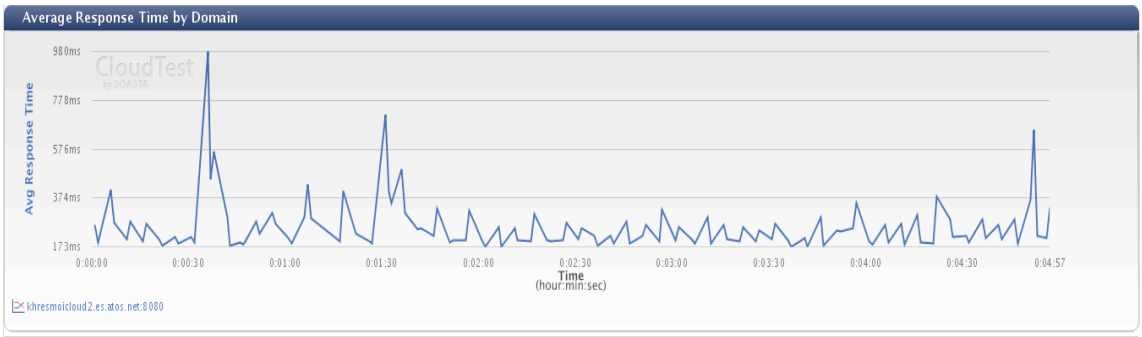

Figure 19. 3D Image Search Average Response Time diagram

9.3.5 Network Bandwidth

Figure 20 represents the amount of network traffic produced during test. It is easy appreciated how test are performed very fast because the time interval of bandwidth use is very small. Moreover, the requests size is very changing, which can be cause because of some request overlapping. Also, we can appreciate around minute 10 a different bandwidth usage which can be related to the higher response time seen in the previous diagram.

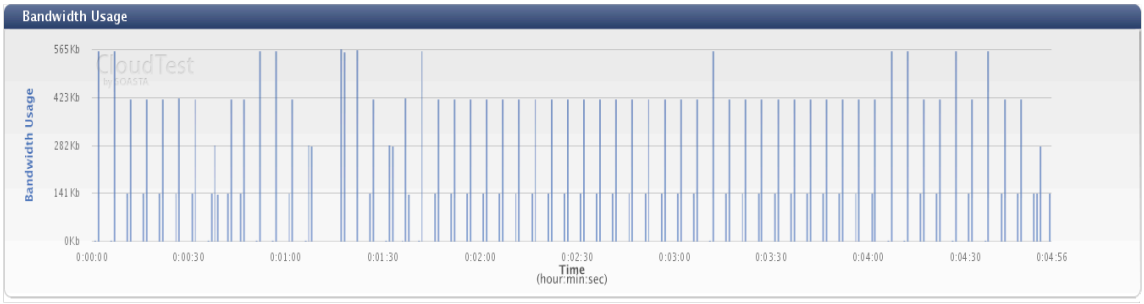

Figure 20. 3D Image Search Network Bandwidth diagram.

Then, we must check that network bandwidth is wide enough for next iterations and evaluations to ensure a good performance of the system.

9.3.6 Virtual Users

Figure 21 shows the distribution of virtual users during 3D Image Search test composition. Low concurrency value for whole user executing time was obtained; only two concurrent users appear around 1:20. This can also be explained because of the data type exchanged during the execution because JSON does not require much network traffic and therefore Virtual Users request are rapidly responded. This diagram can be directly correlated to network bandwidth consumed so it can be appreciated also a very fast execution in the network bandwidth timeline.

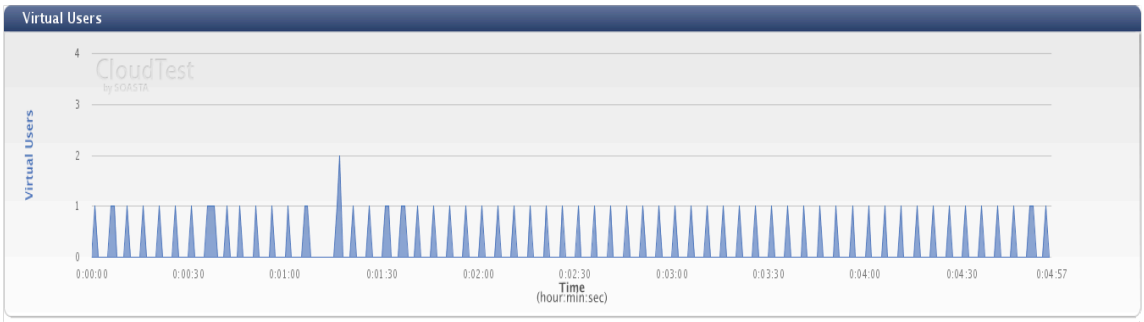

Figure 21. 3D Image Search Virtual Users diagram.

9.4 Multilingual Textual Search Scenario Results

This scenario simulates the KHRESMOI Use Case where the user is doing a query in a different language and some keyword is dynamically translated to English in order to run that query over several datasets. Basically, there are two main steps in this scenario:

- First, the user enters the query and the system should be able to translate some keyword. This action is represented by GET query below to the URL:

  http://khresmoicloud3.es.atos.net:8080/khresmoi-MT/rest/MT/translator/get/simple?action=%22translate%22&sourceLang=de&targetLang=en&text=Zuckerkrankheit

  In this query, we take as an example a user entering "Zuckerkrankheit" as query, which means diabetes. This URL calls instantly to the SCA Multilingual Translator service which is in charge of translating from "sourceLang" to "targetLang" the parameter "text".

- After that, the workflow continues as a typical Textual Search workflow where the query input is the result of the previous translation:

  http://khresmoicloud2.es.atos.net:8080/khresmoi-textual-search/rest/documents?query=diabetes

9.4.1 Fundamentals

Fundamentals metrics in Figure 22 show for this scenario an acceptable performance for the test in terms of response times and execution time. This test includes two different calls from each user, which should complicate workflow. Due to ART is calculated as an average between Multilingual Translation request and Textual Search workflow by CloudTest, in diagram below we have included a box where Total ART is calculated manually. We can realise how value is even lower that for Textual Search scenario in chapter 9.1.1. This is not the most normal result but it could happen for the same test scenario in case that some possible request overlapping of users and requests may delay the performance of the system.

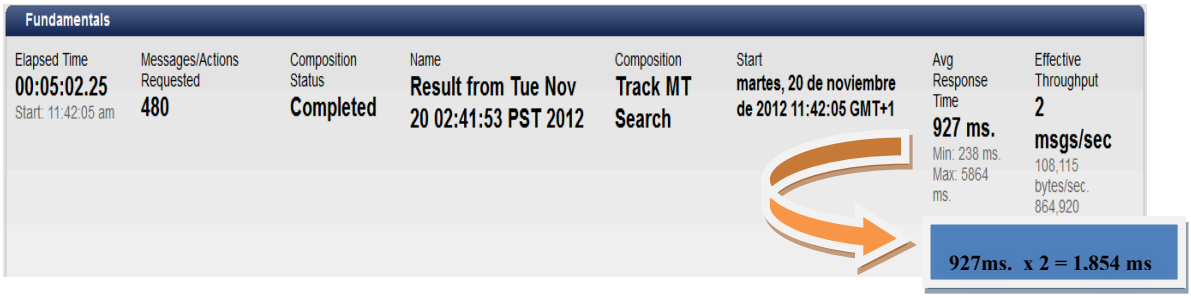

Figure 22. Multilingual Textual Search Fundamentals metrics.

9.4.2 CPU Usage

CPU Usage represented in Figure 23 offers us a whole vision of the system for this complex test. The CPU Usage shown is very inconstant which can be caused because of the different types of requests and also due to any possible overlapping of users. Although CPU consumption reaches 100 % several times during execution we must say that average % consumed is not very high. Anyway, next iterations and evaluations should be used to determine whether new hardware resources may be necessary as well as additional changes.

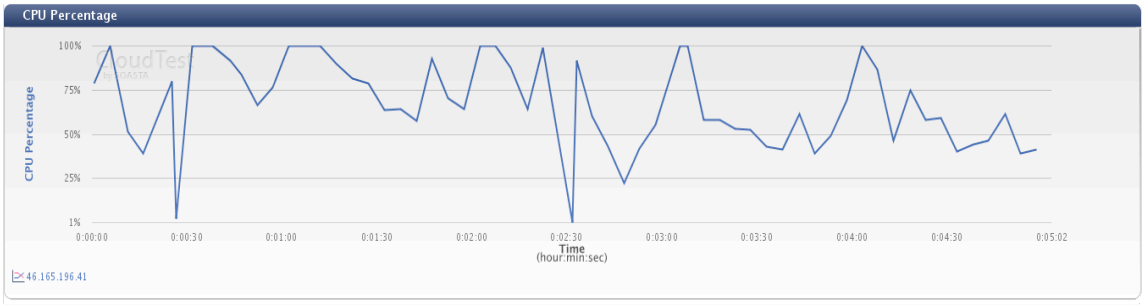

Figure 23. Multilingual Textual Search CPU Usage diagram

9.4.3 Memory Usage

Figure 24 shows a similar behaviour to other scenarios previously performed for memory usage. We can realize how memory consumption keeps quite stable during execution so we can conclude that the system seems to be able to deal with these types of queries.

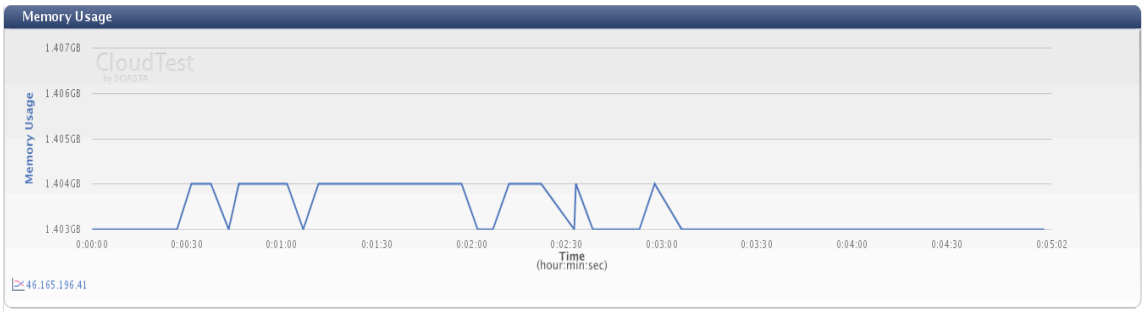

Figure 24. Multilingual Textual Search Memory Usage diagram

Hence, during this first evaluation phase for Early Cloud Prototype the Memory consumption seems not to be a critical point so far.

9.4.4 Average Response Time

Figure 25 is significant helpful as we can appreciate different response times for each method (or http call). Therefore, we can realize how the machine translation service (green line) is quite fast whereas the Textual Search service (blue line) takes much more time to execute.

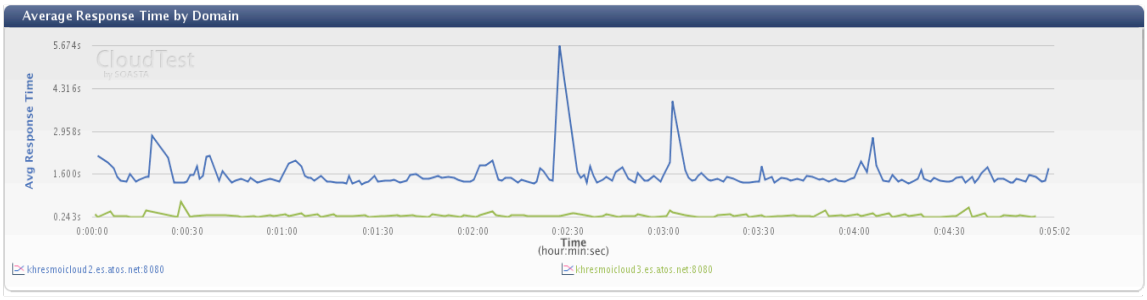

Figure 25. Multilingual Textual Search Average Response Time diagram

9.4.5 Network Bandwidth

In this case, we can appreciate in Figure 26 how queries are composed for two different calls so time required for each user is clearly divided. In this division, we can appreciate how some lines consume some traffic network during more time. Also, the fact of having a constant consumption of 1.1M b could indicate that the system is dedicating all the bandwidth to every user request.

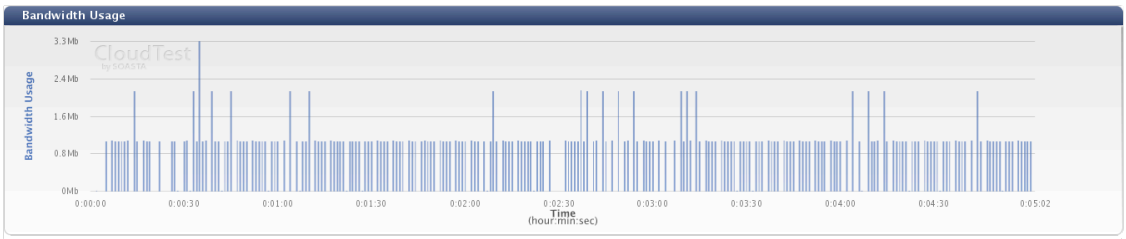

Figure 26. Multilingual Textual Search Network Bandwidth diagram

9.4.6 Virtual Users diagram

Error! Reference source not found. show that during this test we can see frequently different concurrent users at the same. In this case, this fact can be also explained because of more requests are required to complete the task. Anyway the system seems to be very reliable to deal with several requests from different users over long time execution.

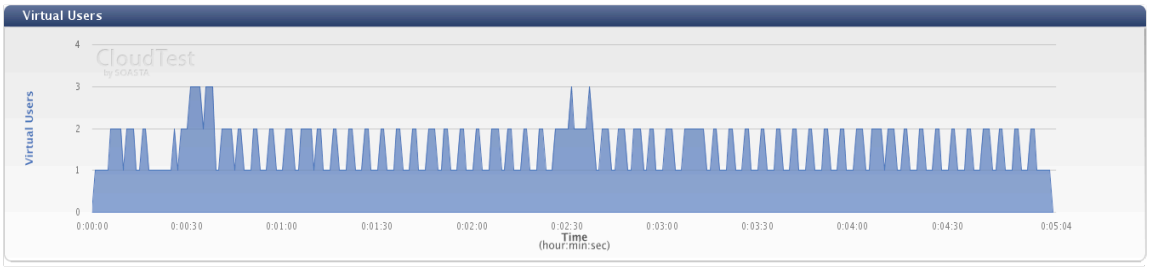

Figure 27. Multilingual Textual Search Virtual Users diagram

10 Facet-Linked Result Analysis

This chapter focuses on the important goal of relating results obtained in the previous chapter with the KHRESMOI Cloud Facets we have chosen as more critical for the system assessment, which are: scalability, elasticity and availability. To achieve this, we make use of radar charts that allow us to compare results from the four main test scenarios described in Section 8. Based on this, we extract valuable conclusions about how the Cloud facets defined in Section 7 are affected by the values represented in each one of the radar charts. Radar charts show values corresponding to the metrics listed in Section 6 taking as data sources the test scenarios.

Therefore, the following subsections are divided within the different metrics previously calculated in order to relate results to scenarios and facets we want to assess. We take as reference the approach described in [4].

10.1 CPU Usage

The CPU Usage is one of the most critical resources in the KHRESMOI Cloud Infrastructure. We have to assure a reasonable value for this metric in our Cloud in order to ensure a good performance as well as a good capability to deal with several requests. In Figure 28, we can observe how average percentage consumed is significantly high in the scenarios related to Textual Search (with and without multilingual support), taking into that we have performed the searches using the default value for the expected results which is 20 documents. A substantial increase in the number of expected results (above 50) imply the occurrence of a bottleneck in CPU consumption. Opposite to that, in Section 9 in the CPU Usage graphics associated to each one of the four test scenarios, we can see how in Image Search scenarios (2D and 3D) only some punctual peaks appear which do not take too much time and therefore do not affect in a critical way to the system.

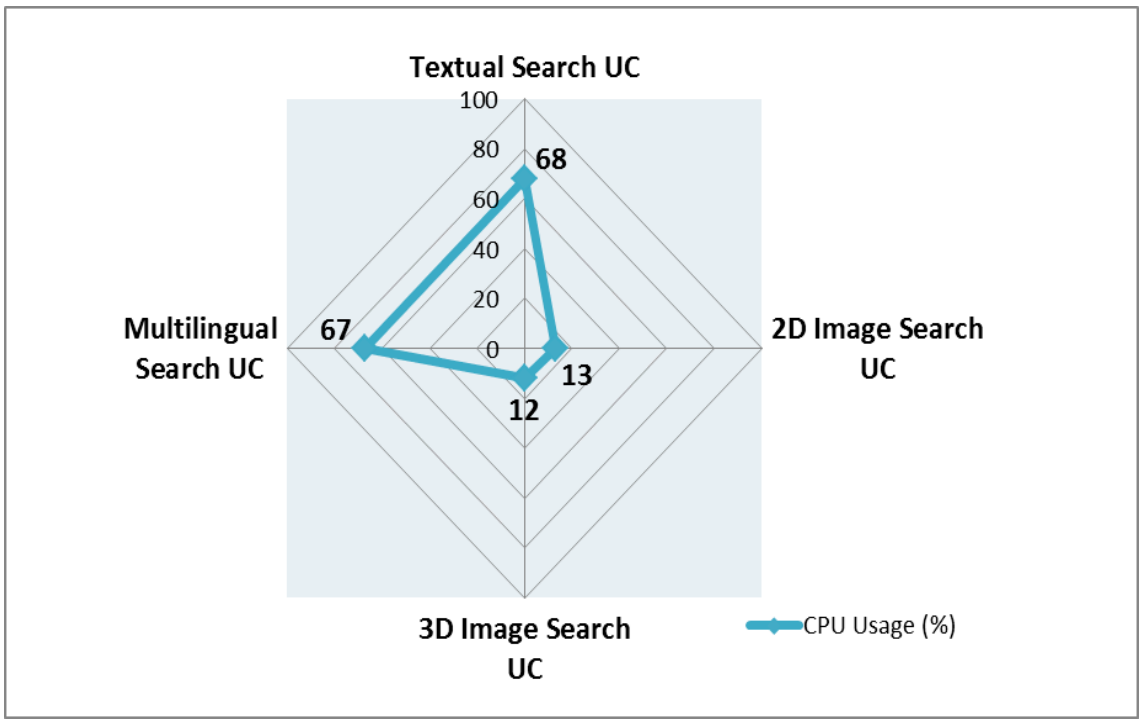

Figure 28. CPU Usage Radar Chart.

Regarding the main facets related to CPU Usage, we can highlight the following conclusions:

- Scalability: the diagram shows how the free CPU percentage is wide enough for Image Search to scale the system in case the demand increases over time. For Textual Search and Multilingual Search we can realise how system consumes almost 70% of resources which must be carefully monitor for next evaluations. A bottleneck could affect to the scalability in the future if the results expected in the scenarios adressed with Textual Search increase the number of expected results.
- Elasticity: according to radar chart results we can realise that the system shows good elasticity because it has been able to adapt dynamically to increasing demand . Similarly to scalability, the possible occurrence of a bottleneck in the CPU usage could cause a collapse of the system on this resource and thus make the system ceases to be elastic.
- Availability: Overall CPU percentage used is not as low as desired so system will not be able to ensure its availability as well as a good performance.

10.2 Memory Usage

The Memory Usage in the test scenarios performed shows good results in terms of overall memory consumption. In Figure 29**Error! Reference source not found.**, we can realize how the average amount of Memory resources required during test executions is quite similar. This can be explained because the SCA Components and Composites deployed may have been initialized with limited assigned size RAM Memory corresponding to the expected maximum load. In the current Memory Usage chart, memory was limited to 3 GB maximum per Virtual Machine although current consumption is around 1,5 GB per scenario. Hence, this size could be updated in a new deployment in case new requirements for extra Memory are identified.

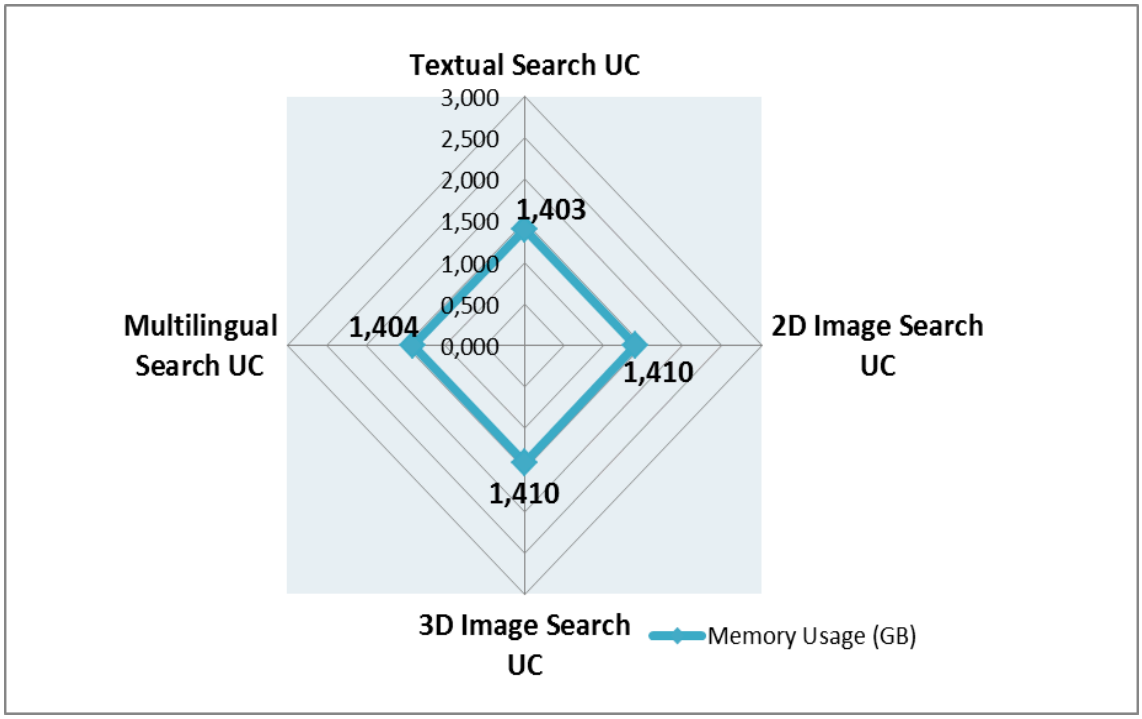

Figure 29. Memory Usage Radar Chart.

Regarding the main Cloud facets, we can extract some brief conclusions:

- Scalability: understanding scalability as statically memory allocation, the system is able to ensure a good value for this facet due to a maximun usage percentage of around 50% with respect to the total memory available.
- Elasticity: in this case, the system behaviour was linear in terms of memory consumption. For this reason we can conclude that elasticity is not a decisive facet from the memory usage point of view in our system.
- Availability: it is currently ensured in a first test approach, for about 60 users in 5 minutes time. In next iterations with new requirements maybe extra memory resources could be needed.

10.3 Average Response Time

The Average Response Time (ART) results aim to show how the system is able to deal with end user requests, providing an acceptable performance in terms of response times for the totality of the KHRESMOI scenarios considered. In Figure 30**Error! Reference source not found.**, we can observe the average response times and we can realize how these values are lower or close to 2 sec. For Textual Search and Multilingual Search we can define their results as high ART values although it has to be considered that concurrent user access to the system has been tested. In any case, improving ART values will be one of the main goals for future evaluations.

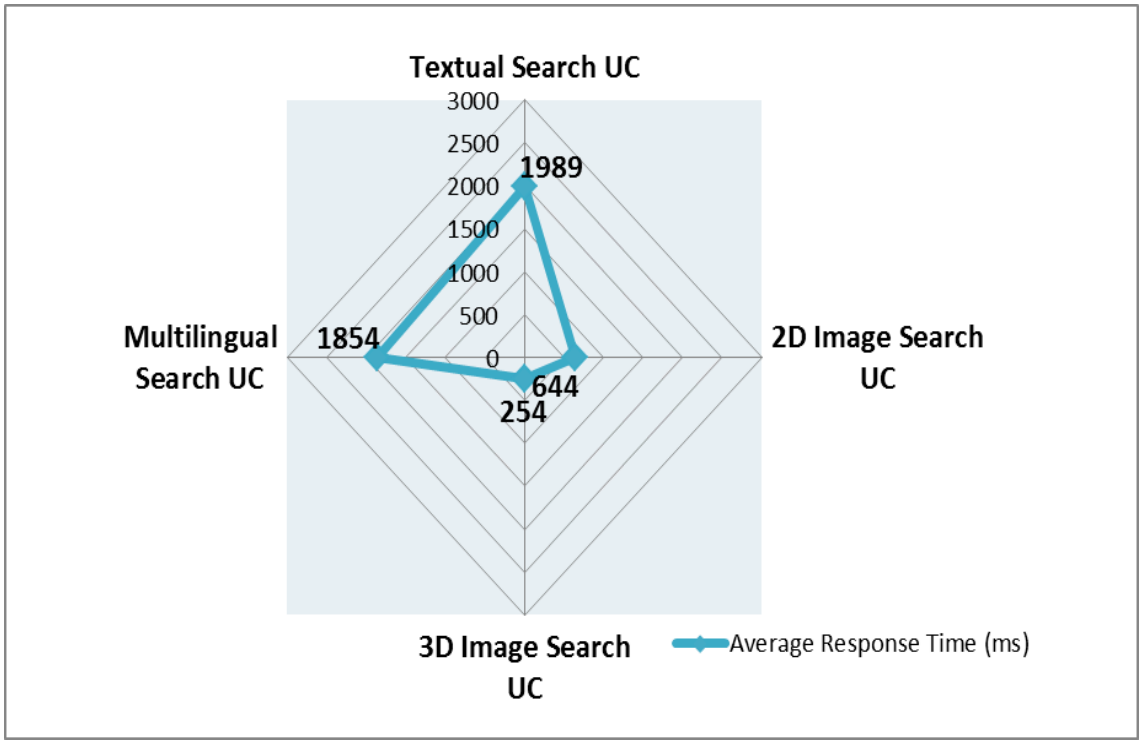

Figure 30 Average Response Time Radar Chart

The main cloud facets defined can be affected by ART obtained in the next terms:

- Scalability: the values obtained for ART closed to 2 seconds not ensure that the Cloud can scale variably and is able to meet the growing amount of demands. A bottleneck

could affect to the scalability in the future if the results expected in the scenarios adressed with Textual Search increase the number of expected results.

- Elasticity: ART can affect elasticity so higher response times may require new changes in Cloud resources (mainly providing new CPUs and extra Memory) in order to improve possible non-acceptable results. In this sense, we can appreciate how the current Cloud infrastructure has been able to elastically adapt to the needs of the system. But for the Textual Search related scenarios, we could consider the possibility of dedicating extra resources or perform new tests to find a more accurate cause of the problem in order to avoid the bottleneck occurrence in ART when the results expected be increased in future scenarios.
- Availability: ART results ensure a complete availability of the system in terms of a 100% of response times before time out. This time out by default, set to 30 seconds, can be modified in next iterations depending on system needs although a good performance should work always in values not higher than 2 seconds.

10.4 Network Bandwidth

The Network bandwidth results show how average consumption is not very high so it should not be a problem for our system. This can be easily explained because the system tends to be not busy during long periods so the average network traffic does not usually reach higher values.

Hence, we can realize in **Error! Reference source not found.** how network bandwidth in the situations simulated is not a problem except in a possible traffic peak scenario where concurrent requests could meet at the same time. Specifically these situations could appear while transferring images is performed in **2D and 3D image visualization scenarios not considered** in the early cloud evaluation process.

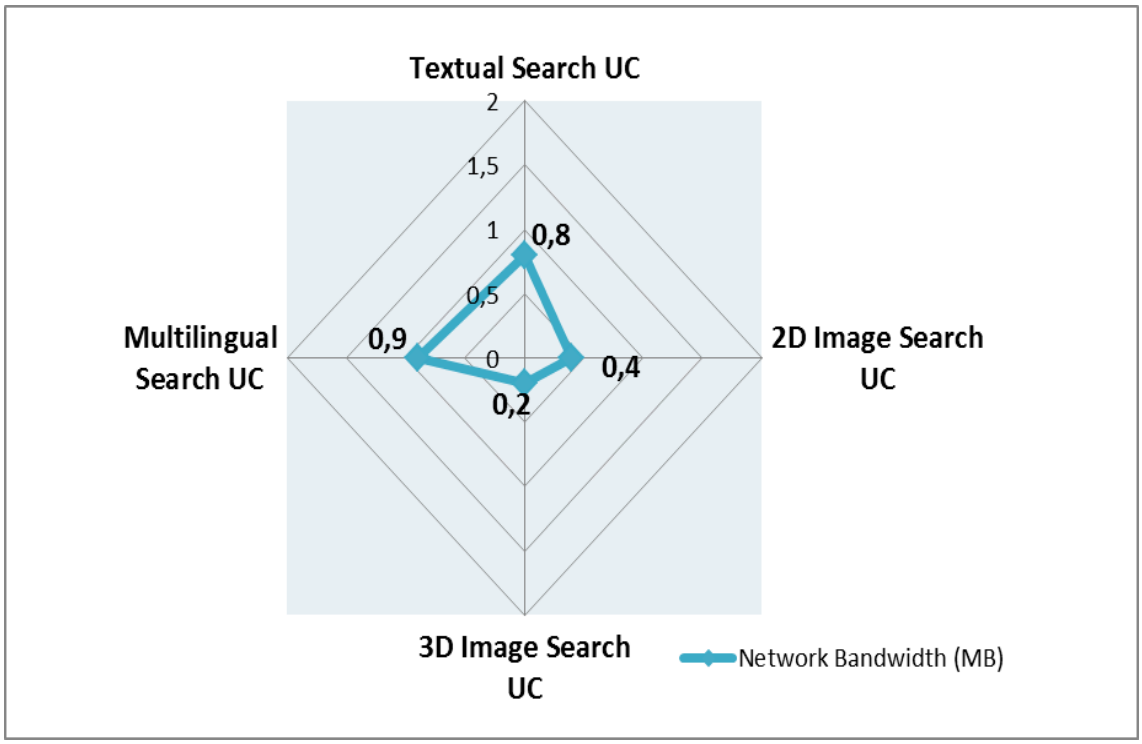

Figure 31 Network Bandwidth Radar Chart

Next we consider possible effects of Network Bandwidth on system performance related to the main facets:

- Scalability: scalability could be affected whenever network traffic grows in an unexpected way. This scenario should be planned early in order to avoid future problems in next iterations.
- Elasticity: Whether a possible increasing of traffic could affect scalability, elasticity should be able to handle these possible dynamic requirements. Therefore, both dynamic and static changes should be taken into account in the future.
- Availability: availability relies on how the system is capable to respond to requests as long as they are requested and consuming bandwidth of the system. In our test cases, we can appreciate a good availability result, but for next iterations possible problems such as network bottlenecks should be taken into account to be avoided.

11 Specification Refinement

In order to solve the bottlenecks identified after result analysis in Section 10 related to CPU Usage and ART in Textual Search and Multilingual Textual Search Scenario, we conclude that a new version of the KHREMOI Cloud Infrastructure will be required during year 3 of the project to get the "Full Cloud Infrastructure" defined in [1].

Table 6. shows the new requirements in terms of HD and SW resources that will be needed for deploying the KHRESMOI Full Cloud.

Partner	Component	VMs	RAM	HDD	Extra storage	Bandwidth	Hypervisor	Monitoring	Software requirements
UNI-DUE	ezDL	1	4GB	2-10GB	Possible	No special configuration required	Xen	CPU load, free HDD, thread count, DB connections, IO load, nw load	Sun Java 1.7 and Mysql 5.X. Debian Linux preferred
HES	PARADISE	2 min.	30GB	1-5TB	Possible	No Special configuration required	Vmware (but opened to change)	Seems like they like to monitor some parameters	To be defined. They will install some custom libraries. If they join the cloud we'll need to store custom images
ONTO TEXT	BigOWLIM Disambiguator	2	15GB min. 30GB max.	600-700GB	Highly probable	No special configuration required	Vmware and LXC (but opened to operate on already built vm)	Webserver	Sun Java, Servlet container and Khresmoi integration system. GUI (Xwindows) and they prefer to use Ubuntu rather than Debian
USFD	GATE/Mimir	1		~1.5TB	Possible	No special configuration required		Not required	Sun Java, Tomcat
CUNI	MOSES	1-2	8GB and 2 CPUs	tend of GB	Possible	No special configuration required	KVM or Virtualbox	Not required	Debian preferred, Python, PERL, GCC
MUW	MUW 3D	more than 4x3.0GHZ CPUs	74GB	1TB+ and fast file system (SAN)	Easy if SAN	No special configuration required	No preferences	Standard logs	WebApp Server and Matlab components
HON	Spell Checker	1	8GB	30GB		No special configuration required	No preferences	Standard logs	Tomcat, Java 1.7 update 1

ATOS	SCA Components	8	4GB per VM	50GB	Possible	No special configuration required	Xen	Standard logs	Tomcat, Java 1.7, Apache Tuscany
ATOS	SCA Composites	2	4GB per VM	50GB	Possible	No special configuration required	Xen	Standard logs	Tomcat, Java 1.7 , Apache Tuscany
ATOS	Cloud Manager	1	4GB per VM	200 GB	Possible	No special configuration required		Application Manager or Nagios required.	Open Nebula, CentOS (Linux distribution)

Table 6. Full Cloud Infrastructure requirements.

12 Conclusion

In this deliverable, we present a summary of the efforts provided in task T6.3.4 related to the "Early Cloud Prototype Evaluation".

After describing the Evaluation Approach defined for testing the KHRESMOI Cloud Infrastructure together with a set of metrics and facets included in the evaluation approach, test cases have been carried out based on the four main scenarios considered in the project.

Regarding the metrics considered for our evaluation approach we can conclude that:

- CPU Usage is one of the most critical resources in the KHRESMOI Cloud Infrastructure. We have to assure a reasonable value for this metric in our Cloud in order to ensure a good performance as well as a good capability to deal with a future increase of several requests or virtual users concurrently in the system. Current values of this metric, in the Textual Search and Multilingual Search Scenarios, led us to identify that if the number of results expected by the users is increased significantly, a bottleneck in CPU Usage could appear in the Cloud infrastructure.
- Memory Usage in the test scenarios performed shows good results in terms of overall memory consumption.
- Average Response Time provides an acceptable performance in terms of response times for the totality of KHRESMOI scenarios considered. Similarly to CPU Usage, in the Textual Search and Multilingual Search Scenarios, a bottleneck in ART could appear in the Cloud infrastructure if the number of results expected by the users is increased significantly (over 100 results).
- Network Bandwidth values show us how average consumption is not very high so it should not be a problem for our system.

Analysis of the results led us to ensure that the current integration and deployment status of the KHRESMOI system could deal with several users and systems satisfying the requirements established in the evaluation approach. Despite this, new approaches should look for new requirements in order to ensure a good performance of the system.

Finally, one of the main achievements at the end of the evaluation task has been the provision of overall visibility of the Cloud infrastructure from the performance point of view to assess the impact of a future implementation change before such a change is introduced. In this way, we will be able to keep good values for main Cloud facets which are scalability, elasticity and availability in future iterations.

13 References

[1] Martinez Rodriguez, Ivan and Tinte Garcia, Miguel Angel, Deliverable D6.4.1 "State of the art, concepts and specification for the "Early Cloud infrastructure" in KHRESMOI Project, Seventh Framework Programme.

[2] Martinez Rodriguez, Ivan and Tinte Garcia, Miguel Angel, Deliverable D6.3.2 "Evaluation of the 'Early software architecture' and further specification" in KHRESMOI Project, Seventh Framework Programme.

[3] Jerry Gao, Xiaoying Bai, and Wei-Tek Tsai, "Cloud Testing- Issues, Challenges, Needs and Practice" in Software Engineering: An International Journal (SEIJ), Vol. 1, No. 1, September 2011.

[4] Gurdev Singh and Rakesh Kumar, "Availability Metrics for Cloud Vibrant Behaviour with Benchmarks Influence On Diverse Facets" in International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.1, January 2012

[5] Proko Eljona, Ninka Ilia, "Analysis and Strategy for the Performance Testing in Cloud Computing" in Global Journal of Computer Science and Technology Cloud & Distributed, Vol. 12 Issue 10 Version 1.0 July 2012