

Grant Agreement Number: 257528

KHRESMOI

www.khresmoi.eu

Evaluation of the ‘Early Integrated Infrastructure’

Deliverable number	<i>D6.5.2</i>
Dissemination level	<i>Public</i>
Delivery date	<i>17 October 2013</i>
Status	<i>Final</i>
Author(s)	<i>Ivan Martinez (AtoS), Miguel Angel Tinte (AtoS)</i>



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

This document describes the evaluation performed over the "Early Integrated Infrastructure", which can be considered as the result of joining the “Full Software architecture” deployed within the “Full Cloud Prototype” to achieve scalability, flexibility and elasticity requirements. Therefore, this document describes mainly the work achieved in Tasks “T6.2 Architecture design” and “T6.3 System scaling” as well as its evaluation process. This evaluation also summarizes and collects all the work performed during the project regarding cloud infrastructure deployment as well as all architecture integration work achieved.

Following this approach, this deliverable adapts previous evaluation metrics and results performed during Khresmoi infrastructure development, reported in deliverables D6.4.2 and D6.4.3, in order to offer a complete overview of the platform. To do so, this document defines different tests, some scenario-independent and some related to different Khresmoi scenarios. Then, these test results are presented and interpreted to extract useful conclusions about the assembled infrastructure.

Besides this, this document describes the last integration changes performed since the previous integrated infrastructure deliverable D6.5.1 regarding new SCA Composites and Components that have been included in Khresmoi workflows.

Finally, this document shows an overall view of Khresmoi final architecture once cloud infrastructure and integration tasks have been finished, and describes the evaluation results that will be completed with new deployment recommendations.

Table of Contents

Executive summary	2
List of abbreviations.....	5
List of figures	6
List of tables	7
1 Introduction	8
1.1 Introductory Explanation of the Deliverable	8
1.2 Purpose and Audience	8
1.2.1 Purpose.....	8
1.2.2 Audience	8
1.3 Structure of the Document	8
2 Global Evaluation Approach.....	9
2.1 Architecture Evaluation Approach	9
2.2 Cloud Infrastructure Evaluation Approach.....	9
3 Tests Scenario Definition	10
3.1. Textual Search Scenario	10
3.2. 2D Image Search Scenario.....	10
3.3. 3D Image search Scenario	10
3.4. Multilingual Search Scenario.....	10
3.5. 2D Semantic Image Search.....	10
4. Scenario Independent Results	13
4.1. Network Cohesion	14
4.2. Number of Services Involved in a Compound Service	14
4.3. Services Interdependence in the System	15
4.4. Absolute Importance of a Service.....	15
4.5. Absolute Dependence of a Service	16
4.6. Absolute Criticality of a Service	16
4.7. Overall Reliability of a Compound Service	18
5. Scenario Dependent Results and Performance Metrics.	19
5.1. Textual Search Scenario.....	19
5.1.1. Sizes of Input and Output Messages	19
5.1.2. Messages Rates	20
5.1.3. Network Load.....	20
5.1.4. CPU Usage.....	21
5.1.5. Memory Usage.....	21
5.1.6. Average Response Time	21
5.1.7. Network Bandwidth	21
5.1.8. Virtual Users	21
5.2. 2D Image Search Scenario.....	21
5.2.1. Sizes of Input and Output Messages	21

D6.5.2. Evaluation of the “Early Integrated Infrastructure”

5.2.2.	Messages Rates	21
5.2.3.	Network Load.....	22
5.2.4.	CPU Usage	22
5.2.5.	Memory Usage	22
5.2.6.	Average Response Time	22
5.2.7.	Network Bandwidth	22
5.2.8.	Virtual Users	22
5.3.	3D Image Search Scenario.....	22
5.3.1.	Sizes of Input and Output Messages	22
5.3.2.	Messages Rates	22
5.3.3.	Network Load.....	22
5.3.4.	CPU Usage	23
5.3.5.	Memory Usage	23
5.3.6.	Average Response Time	23
5.3.7.	Network Bandwidth	23
5.3.8.	Virtual Users	23
5.4.	Multilingual Textual Search Scenario.....	23
5.4.1.	Sizes of Input and Output Messages	23
5.4.2.	Messages Rates	24
5.4.3.	Network Load.....	24
5.4.4.	CPU Usage	24
5.4.5.	Memory Usage	24
5.4.6.	Average Response Time	24
5.4.7.	Network Bandwidth	24
5.4.8.	Virtual Users	24
5.5.	2D Semantic Image Search Scenario.....	25
5.5.1.	Sizes of Input and Output Messages	25
5.5.2.	Messages Rates	26
5.5.3.	Network Load.....	26
5.5.4.	CPU Usage	27
5.5.5.	Memory Usage	27
5.5.6.	Average Response Time (ART).....	28
5.5.7.	Network Bandwidth	28
5.5.8.	Virtual Users	29
6.	Conclusions	30
7.	References	31

List of abbreviations

SOA	Service Oriented Architecture
SCA	Service Component Architecture
SOMA	Service-Oriented Modelling Architecture
REST	Representational State Transfer
API	Application Programming Interface
NC	Network Cohesion
NSIC	Number of Services Involved in a Compound Service
AIS	Absolute Importance of a Service
ADS	Absolute Dependence of a Service
ACS	Absolute Criticality of a Service
DoW	Description of Work

Table 1: Abbreviations and acronyms

List of figures

Figure 1. Khresmoi Early Integrated Infrastructure System	13
Figure 2: CPU Usage Semantic Search	27
Figure 3: Memory Usage Semantic Search	27
Figure 4: ART Semantic Search.....	28
Figure 5: Network Bandwidth Semantic Search	28
Figure 6: Virtual Users Semantic Search	29

List of tables

Table 1: Abbreviations and acronyms	5
Table 2. 2D Semantic Image Search	12
Table 3. Absolute criticality of a service matrix	18

1 Introduction

1.1 Introductory Explanation of the Deliverable

The main objective of this document is to provide an overview of the “Early integrated infrastructure” seen as the joint of the Integrated Software Architecture within the Full Cloud prototype. The description of this Early Integrated Infrastructure is showed along with an evaluation chapter where the tests performed are described in order to assess the infrastructure. These tests allow us to obtain some predefined metrics to evaluate numerically some performance features.

Besides this, the development and integration of new components is also presented in this document.

1.2 Purpose and Audience

1.2.1 Purpose

The purpose of this deliverable is to describe the Early Integrated infrastructure where Full Cloud prototype and integration architecture are put to work together into a single platform. This phase should be one the final Khresmoi stages in order to obtain a final and stable platform according to predefined tasks T6.2 and T6.3 as presented in DoW.

1.2.2 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 software architecture. As such this deliverable presents a description of the Khresmoi Full Prototype software architecture, and therefore this document could be interesting for any Khresmoi member.

Because the document is public it is also intended for anybody interested in the evaluation and integration of Khresmoi.

1.3 Structure of the Document

This deliverable inherits the structure of the previous specification deliverable D6.5.1 [3] and includes new sections regarding specific Cloud Prototype evaluation. According to this, Section 0 focuses on describing the Global Evaluation Approach adopted in order to evaluate architecture and cloud solutions developed. After that, Section **Error! Reference source not found.** defines the test scenarios definition integrated so far into different workflows. Section 4 focuses on describing scenario independent results and in Section 5 the scenario dependent result along with the main performance metrics is described. Finally, Section 6 is dedicated to describe the main conclusions obtained after evaluation process.

2 Global Evaluation Approach

This document aims to evaluate the Khresmoi platform based in two main characteristics of the deployment: The architecture integration and the cloud infrastructure. The former focuses on describing the process achieved to integrate different partners’ component and functionalities, whereas the latter describes the cloud infrastructure deployed over the new hardware resources acquired in order to improve scalability and system performance.

Subsequent subchapters described more deeply the two above-mentioned characteristics of the evaluation.

2.1 Architecture Evaluation Approach

The approach adopted to evaluate the integration of the architecture consisted basically in the assessment of the set of elements (services, components, composites, workflows, etc.) that have been deployed altogether. Such combined deployment seeks to ensure a good performance of the system, incorporating all Khresmoi tools into the same cluster of machines and network.

In addition to this, the architecture evaluation is using some metrics regarding network use and service performance in order to measure the system capacity to deal with different requests to the services exposed. These metrics and their results have been presented in Chapter 4.

2.2 Cloud Infrastructure Evaluation Approach

The method adopted to evaluate the cloud infrastructure is based on the simulation of Khresmoi main use cases and workflows in order to assess some predefined metrics regarding main hardware resources, such as network bandwidth, processing, memory consumption, etc. The main goal is to check whether the so-called Full Cloud infrastructure resources are enough to deal with end-users requirements in terms of scalability, availability and reliability. In order to do so, the main workflows identified so far have been simulated and reported in Chapter 5.

Also the scenarios along with the previous metrics and results are presented in the same Chapter. Some of these metrics have been already described in Full Cloud evaluation prototype [4].

3 Tests Scenario Definition

This Chapter describes different workflow scenarios simulated in order to obtain different results and metrics to assess the main integration and scalability concepts. Some of these have been already simulated in previous evaluations to assess each infrastructure deployment, so it is referenced in each case:

3.1. Textual Search Scenario

This scenario has been already described in previous deliverable D6.4.3 [4].

3.2. 2D Image Search Scenario

This scenario has been already described in previous deliverable D6.4.3 [4].

3.3. 3D Image search Scenario

This scenario has been already described in previous deliverable D6.4.3 [4].

3.4. Multilingual Search Scenario

This scenario has been already described in previous deliverable D6.4.3 [4].

3.5. 2D Semantic Image Search

This scenario aims to describe a new workflow incorporated into Khresmoi. It focuses on a use case where the end-user performs a semantic search by using some concepts and categories predefined. Basically, the workflow is composed by the next steps:

- First, a preselected list of terms is loaded to ezDL by launching an empty query to the co-occurrence web service of ONTOText. This component allows identifying of PubMed abstracts, in which two or more concepts (anatomies, pathologies and modalities) co-occur in a specific context.
- Then, the terms are disambiguated by calling the Disambiguator web service. The user selects some of the disambiguated terms for some of the following fields: Anatomy, Pathology, Modality and Findings. These terms are given as input to the co-occurrence web service.
- The web service output contains a list of image URIs that is sent to the ParaDISE web service to retrieve the images and the corresponding information (modality, article, caption, etc.). Also, the terms in the Co-Occurrence services' response are disambiguated by the Disambiguator service.
- Finally, the response is returned to ezDL.

D6.5.2. Evaluation of the “Early Integrated Infrastructure”

Table 2 below shows different workflow steps required to cover the full scenario:

2D Semantic 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 (HES)	ParaDISE (the GNU Image-Finding Tool) is a Content Based Image Retrieval System.
	SCA-ParaDISE (AtoS)	SCA Component for Multilingual Translator
	Cooccurrence (ONTO)	Co-occurrence Search: A query searches within all fields defined for each entity. The result is a list, which can be further narrowed by using the filter facets.
	SCA Cooccurrence (AtoS)	SCA Component for Cooccurrence
	SCA Cooccurrence (AtoS)	SCA Component for Cooccurrence
Scenario	Step1 : load terms	The user performs an empty query to obtain a list of preselected terms.
	Step2 : the terms are disambiguated	Terms are disambiguated by SCA Disambiguator
	Step3 : the user selects some terms	The user selects a word as a translation of the keyword entry
	Step4: terms are passed as input to Cooccurrence	After disambiguation, some terms are selected by the user and given as input to Cooccurrence service
	Step5: Cooccurrence output is given as input to ParaDISE	Cooccurrence returns the output containing a list of images URIs and other fields
	Step 6 : ParaDISE retrieves URLs images	ParaDISE service receives list URIs and return their information
	Step 6 : response is returned to ezDL	ParaDISE response about list of URIs is returned to the user

D6.5.2. Evaluation of the “Early Integrated Infrastructure”

Main functionalities to be integrated	Functionalities used by User: searchImages() From ezDL
	Functionalities used by ezDL: List<PreselectedTerms> = Load() from SemanticManager
	Functionalities used by ezDL: List<ImageInfo>, List<CooccurrenceTerms> = FindImages(List<RadlexTerms>) from SemanticManager
	Functionalities used by Semantic Manager: List<PreselectedTerms> = EmptyQuery()
	Functionalities used by Semantic Manager: List<DisambiguatedTerms> = DisambiguateDetails() from SCA-Disambiguator
	Functionalities used by Semantic Manager: List<ImageIDs>, List<CooccurrenceTerms> = cooccurrenceCall(List<DisambiguatedTerms>) from SCA Cooccurrence
	Functionalities used by Semantic Manager: List<ImageInfo> = GetImageInfo(List<ImageIDs>) From SCA ParaDISE

Table 2. 2D Semantic Image Search

4. Scenario Independent Results

Figure 1 shows all the nodes and components involved in the Early Integrated prototype where basically it is displayed graphically the different Full Cloud nodes where the SCA Components and Composites have been deployed and the interrelations among them:

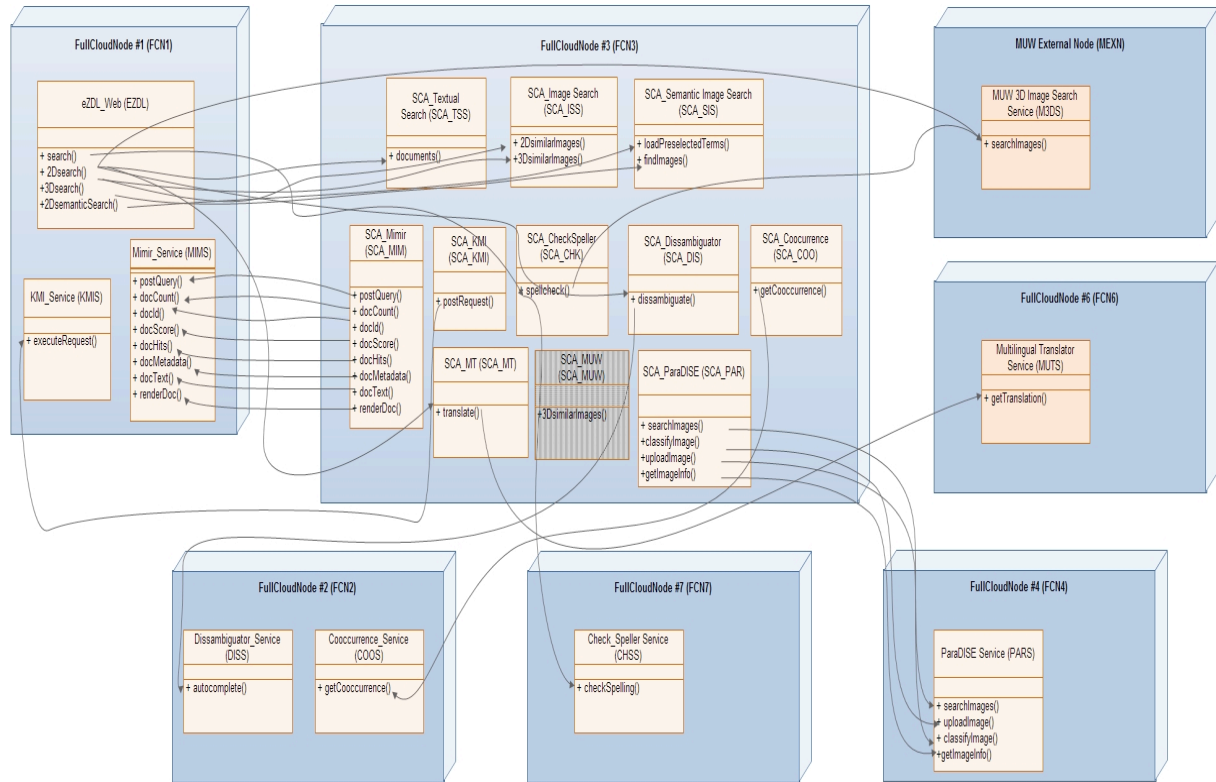


Figure 1. Khresmoi Early Integrated Infrastructure System

The next box shows the updated set A of calls between components (this set was already defined in deliverable D6.):

```
A = {< FCN1, EZDL, FCN3, SCA_TSS, SCA_TSS.documents()>,
    < FCN1, EZDL, FCN3, SCA_ISS, SCA_ISS.2DsimilarImages()>,
    < FCN1, EZDL, MEXN, M3DS, M3DS.3DsimilarImages()>,
    < FCN1, EZDL, FCN3, SCA_SIS, SCA_SIS.loadPreselectedTerms()>,
    < FCN1, EZDL, FCN3, SCA_SIS, SCA_SIS.findImages()>,
    < FCN1, EZDL, FCN3, SCA_CHK, SCA_CHK.checkSpelling()>,
    < FCN1, EZDL, FCN3, SCA_MT, SCA_MT.translate()>,
    < FCN1, EZDL, FCN3, SCA_DIS, SCA_DIS.dissambiguate ()>,
    < FCN3, SCA_MIM, FCN1, MIMS, MIMS.postQuery()>,
    < FCN3, SCA_MIM, FCN1, MIMS, MIMS.docCount()>,
```

D6.5.2. Evaluation of the “Early Integrated Infrastructure”

```
< FCN3, SCA_MIM, FCN1, MIMS, MIMS.docId()>,
< FCN3, SCA_MIM, FCN1, MIMS, MIMS.docScore()>,
< FCN3, SCA_MIM, FCN1, MIMS, MIMS.docHits()>,
< FCN3, SCA_MIM, FCN1, MIMS, MIMS.docMetadata()>,
< FCN3, SCA_MIM, FCN1, MIMS, MIMS.renderDocument()>,
< FCN3, SCA_KMI, FCN1, MIMS, KMIS.executeRequest()>,
< FCN3, SCA_CHK, FCN7, CHSS, CHSS.checkSpelling()>,
< FCN3, SCA_DIS, FCN2, DISS, DISS.autocomplete()>,
< FCN3, SCA_COO, FCN2, COOS, COOS.getCooccurrence()>,
< FCN3, SCA_MT, FCN6, MUTS, MUTS.translate()>,
< FCN3, SCA_PAR, FCN4, PARS, PARS.searchImages()>,
< FCN3, SCA_PAR, FCN4, PARS, PARS.updateImage()>,
< FCN3, SCA_PAR, FCN4, PARS, PARS.classifyImage()>,
< FCN3, SCA_PAR, FCN4, PARS, PARS.getImagesInfo()>}
```

4.1. Network Cohesion

For the calculation of this metric the definition provided in [1] is taken as reference. . This metric can be calculated as follows:

$$NC = |\pi_{InvokerNode, ServiceNode}(\sigma_{InvokerNode \neq ServiceNode}(A))| = \{ \langle FCN1, FCN3 \rangle, \langle FCN1, MEXN \rangle, \langle FCN3, FCN1 \rangle, \langle FCN3, FCN2 \rangle, \langle FCN3, FCN7 \rangle, \langle FCN3, FCN4 \rangle, \langle FCN3, FCN6 \rangle, \langle FCN3, MEXN \rangle \} = 8$$

In the current prototype evaluation, the NC has a value of 8, lower than in previous evaluation [2] although the number of nodes has also decreased due to complete integration of components. This means that system cohesion is even better than before because the average number of direct links (8) between the nodes (7) is higher than 1.

4.2. Number of Services Involved in a Compound Service

For the calculation of this metric the definition provided in [1] is taken as reference. In the Early Integrated Prototype, the composed services are SCA_TSS, SCA_ISS, SCA_SIS and ezDL and the metric values for these cases are calculated as follows:

$$\begin{aligned} NSIC[c] &= |S^R[c]| \\ NSIC[SCA_TSS] &= |S^R[SCA_TSS]| = \{SCA_KMI, SCA_MIM, KMIS, MIMS\} = 4 \\ NSIC[SCA_ISS] &= |S^R[SCA_ISS]| = \{SCA_PAR, PARS\} = 2 \\ NSIC[SCA_ISS] &= |S^R[SCA_SIS]| = \{SCA_COO, SCA_DIS, SCA_PAR, COOS, MUWS, PARS\} = 6 \\ NSIC[ezDL] &= |S^R[EZDL]| = \{SCA_TSS, SCA_ISS, SCA_SIS, SCA_KMI, SCA_MIM, SCA_COO, SCA_DIS, SCA_MT, SCA_MT, SCA_PAR, KMIS, MIMS, MUTS, COOS, DISS, PARS, M3DS\} = 17 \end{aligned}$$

Similarly to previous evaluation [3], the results for NSIC metrics show that the SCA Composites have a high value due to the complexity of the workflows implemented. However, the highest value is for

D6.5.2. Evaluation of the “Early Integrated Infrastructure”

ezDL as it is the access layer for the end-users. This value must be taken into account in order to give priority to ezDL component if system performance is affected. The main actions to avoid the consequences of this high value should be related to hardware resources assigned to FCN1 and ezDL Virtual Machine.

4.3. Services Interdependence in the System

For the calculation of this metric the definition provided in [1] is taken as reference. After the evaluation, we can conclude that there is no service interdependence in the Khresmoi Early Integrated Prototype Architecture, due to that there is not bidirectional relation between two composing services, which attest a good services design.

4.4. Absolute Importance of a Service

For the calculation of this metric the definition provided in [1] is taken as reference. It can be calculated as follows:

$$AIS[s] = |\pi_{Invoker}(\sigma_{InvokerNode=s, ServiceNode \neq n}(A))|$$

$AIS[ezDL]=0$
 $AIS[SCA_TSS]=1$
 $AIS[SCA_ISS]=1$
 $AIS[SCA_SIS]=1$
 $AIS[SCA_CHK]=1$
 $AIS[SCA_DIS]=2$
 $AIS[SCA_COO]=1$
 $AIS[SCA_KMI]=1$
 $AIS[SCA_MIM]=2$
 $AIS[SCA_PAR]=2$
 $AIS[SCA_MT]=1$
 $AIS[DISS]=1$
 $AIS[CHSS]=1$
 $AIS[KMIS]=1$
 $AIS[MIMS]=1$
 $AIS[MUWS]=1$
 $AIS[PARS]=2$
 $AIS[MUTS]=1$

Results above display how AIS is contained in a little range of 0-2. This can be interpreted as lower values for all the components, although the components with value 2 can be considered as more important than others. This numeric interpretation is based on that these components are involved in different scenarios and workflows.

4.5. Absolute Dependence of a Service

For the calculation of this metric the definition provided in [1] is taken as reference, which can be calculated as follows:

$$ADS[s] = |\pi_{Service}(\sigma_{Invoker=s, ServiceNode \neq n}(A))|.$$

$$ADS[ezDL]=8$$

$$ADS[SCA_TSS]=2$$

$$ADS[SCA_ISS]=1$$

$$ADS[SCA_SIS]=3$$

$$ADS[SCA_CHK]=1$$

$$ADS[SCA_DIS]=1$$

$$ADS[SCA_COO]=1$$

$$ADS[SCA_KMI]=1$$

$$ADS[SCA_MIM]=1$$

$$ADS[SCA_PAR]=1$$

$$ADS[SCA_MT]=1$$

$$ADS[DISS]=0$$

$$ADS[COOS]=0$$

$$ADS[CHKS]=0$$

$$ADS[KMIS]=0$$

$$ADS[MIMS]=0$$

$$ADS[MUWS]=0$$

$$ADS[PARS]=0$$

$$ADS[MTS]=0$$

As explained in deliverable [1], this metric can be defined as the opposite to the previous one, so this shows the level of dependence to other services. In this case, this value gives an idea about the autonomy of the services and possibility of being reused as stand-alone instances, etc. Therefore, autonomy and reusability are affected by this value. Similarly to previous evaluations, ADS [ezDL] shows the higher value and represents the most important dependence of the infrastructure.

4.6. Absolute Criticality of a Service

For the calculation of this metric the definition provided in [1] is taken as reference [1], which can be calculated as follows:

$$ACS[s] = AIS[s] \times ADS[s].$$

$$ACS[ezDL] = AIS[ezDL] \times ADS[ezDL]=0$$

$$ACS[SCA_TSS] = AIS[SCA_TSS] \times ADS[SCA_TSS]=2$$

$$ACS[SCA_ISS] = AIS[SCA_ISS] \times ADS[SCA_ISS]=1$$

$$ACS[SCA_SIS] = AIS[SCA_SIS] \times ADS[SCA_SIS]=3$$

D6.5.2. Evaluation of the “Early Integrated Infrastructure”

$$\begin{aligned}
 ACS[SCA_CHK] &= AIS[SCA_CHK] \times ADS[SCA_CHK]=1 \\
 ACS[SCA_DIS] &= AIS[SCA_DIS] \times ADS[SCA_DIS]=2 \\
 ACS[SCA_COO] &= AIS[SCA_COO] \times ADS[SCA_COO]=1 \\
 ACS[SCA_KMI] &= AIS[SCA_KMI] \times ADS[SCA_KMI]=1 \\
 ACS[SCA_MIM] &= AIS[SCA_MIM] \times ADS[SCA_MIM]=2 \\
 ACS[SCA_PAR] &= AIS[SCA_PAR] \times ADS[SCA_PAR]=2 \\
 ACS[SCA_MT] &= AIS[SCA_MT] \times ADS[SCA_MT]=1 \\
 ACS[DISS] &= AIS[DISS] \times ADS[DISS]=0 \\
 ACS[COOS] &= AIS[COOS] \times ADS[COOS]=0 \\
 ACS[SPES] &= AIS[CHKS] \times ADS[CHKS]=0 \\
 ACS[QMSS] &= AIS[KMIS] \times ADS[KMIS]=0 \\
 ACS[MIMS] &= AIS[MIMS] \times ADS[MIMS]=0 \\
 ACS[PARS] &= AIS[PARS] \times ADS[PARS]=0 \\
 ACS[MTS] &= AIS[MTS] \times ADS[MTS]=0
 \end{aligned}$$

The absolute critical results for different Khresmoi services and components are depicted above. The values calculated show how criticality of some components increased as a consequence of a higher number of different elements involved in some workflows Table 3 below shows a classification of different services criticality results:

Metrics	Criticality		
	Very critical (>1)	Critical(=1)	Low criticality(=0)
ACS[ezDL]			✗
ACS[SCA_TSS]	✗		
ACS[SCA_ISS]		✗	
ACS[SCA_SIS]	✗		
ACS[SCA_CHK]		✗	
ACS[SCA_DIS]	✗		
ACS[SCA_COO]		✗	
ACS[SCA_KMI]		✗	
ACS[SCA_MIM]	✗		
ACS[SCA_PAR]	✗		
ACS[SCA_MT]		✗	

D6.5.2. Evaluation of the “Early Integrated Infrastructure”

ACS[DISS]	x
ACS[COOS]	x
ACS[SPES]	x
ACS[QMSS]	x
ACS[PARS]	x
ACS[MTS]	x

Table 3. Absolute criticality of a service matrix

Summarizing the results displayed on table above, an increasing of criticality on Composites that implement different workflows is observed (for example the Semantic Image Search workflow). Also, some others Components have increased their criticality as they have augmented their internal complexity and functionalities provided. The main classification of critical services is:

- SCA_TSS, SCA_SIS, SCA_ISS, SCA_MIM, SCA_PAR with value higher than 1.
- SCA_ISS, SCA_CHK, SCA_COO, SCA_KMI, SCA_MT with value equals to 1.

4.7. Overall Reliability of a Compound Service

For the calculation of this metric the definition provided in [1] is taken as reference, which can be calculated as follows:

$$RC[c] = 1/\max(\{ACS[s] \mid s \in S^R[c]\}).$$

$$RC[ezDL] = 1/\max(\{ACS[s] \mid s \in S^R[ezDL]\}) = 1/3 = \mathbf{0.33}$$

$$RC[SCA_TSS] = 1/\max(\{ACS[s] \mid s \in S^R[SCA_TSS]\}) = 1/2 = \mathbf{0.5}$$

$$RC[SCA_ISS] = 1/\max(\{ACS[s] \mid s \in S^R[SCA_ISS]\}) = 1/2 = \mathbf{0.5}$$

$$RC[SCA_SIS] = 1/\max(\{ACS[s] \mid s \in S^R[SCA_SIS]\}) = 1/2 = \mathbf{0.5}$$

The results observed above show the same results that in previous evaluation so we can infer that reliability of the system is kept stable during this evaluation.

5. Scenario Dependent Results and Performance Metrics.

This chapter describes the metric results related to the execution of different scenario simulation. These simulations have been already performed in previous deliverables and will be indicated in those cases with appropriate references. The rest of scenarios have been simulated using the following configuration parameters: 100 users that perform 4 request each one in a time interval of 5 minutes (these parameters were also used in Deliverables [4][5]).

5.1. Textual Search Scenario.

This scenario simulation has been already assessed and appears along with its metric results in deliverable [4].

5.1.1. Sizes of Input and Output Messages

This metric is defined in deliverable [6]) and has been used in previous evaluation deliverable [5]. The main goal of this metric is to quantify the size of input and output messages that participate in each workflow. In this case, the size of messages involved in Textual Search workflow is shown below. In order to extract these values over the same query sample, some predefined queries like “diabetes” have been issued:

- AN2.spellCheck : diabetes

AIMSO [SCA_SPE.spellcheck] = 200 bytes
AOMSO [SCA_SPE.spellCheck] = 1054 bytes

- AN2.dissambiguate : diabetes

AIMSO [SCA_DIS.dissambiguate] = 243 bytes
AOMSO [SCA_DIS.dissambiguate] = 5,874 bytes

- AN1.search: diabetes

AIMSO [SCA_TSS.search] = 371 bytes
AOMSO [SCA_TSS.search] = 5,932 bytes

The results above show similar sizes of input and output except for Disambiguation that has increased considerably. This is normal, as the components, data stores and indexes are getting more complex when the project grows and evolves. Despite this, an important step to check will be how this affects the system by comparing the previous and current prototype. Current prototype, does not seems affected seriously its performance although components complexity has grown since last evaluation.

5.1.2. Messages Rates

Below the MRN (Message Rate of the Node) values obtained for Textual Search main workflow elements are presented, taking “diabetes” as example query:

- AN2.spellCheck : diabetes

$$\begin{aligned} IMRN [AN2] &= OMRAN [AN2] = \mathbf{244 \text{ msg/min}} \\ MRN [AN2] &= 244 + 244 = \mathbf{488 \text{ msg/min}} \end{aligned}$$

- AN2.dissambiguate : diabetes

$$\begin{aligned} IMRN [AN2] &= OMRAN [AN2] = \mathbf{193 \text{ msg/min}} \\ MRN [AN2] &= 193 + 193 = \mathbf{386 \text{ msg/min}} \end{aligned}$$

- AN1.search: diabetes

$$\begin{aligned} IMRN [AN1] &= OMRAN [AN1] = \mathbf{40 \text{ msg/min}} \\ MRN [AN1] &= 40 + 40 = \mathbf{80 \text{ msg/min}} \end{aligned}$$

The results obtained for this metric must be compared with the obtained ones during previous evaluation in order to give a good idea of the improvements achieved. In this context, MRN has increased for the AN2.spellCheck and AN2.dissambiguate. However, the bigger improvement is represented by AN1.search that increases from a MRN value of 12 to 80. This could be justified by the integration of the AN1 components into the same cluster and network. This feature should lower the execution time by decreasing time delays when several requests are performed for workflow completion.

5.1.3. Network Load

The importance of this metric is based on its capacity to demonstrate the viability of the use case scenario execution through the network and to identify the bandwidth required for the correct performance of the system. To do so, it is collected the total size of input and output for every component request involved in the execution:

$$\begin{aligned} ITN[AN1] &= N \times (371 + 5,932) = \mathbf{N \times 6,303 \text{ bytes/call}} \\ OTN[AN1] &= N \times (5,932 + 371) = \mathbf{N \times 6,303 \text{ bytes/call}} \\ ITN[AN2] &= N \times (200 + 1054 + 243 + 5,874) = \mathbf{N \times 7,371 \text{ bytes/call}} \\ OTN[AN2] &= N \times (5,874 + 243 + 1054 + 200) = \mathbf{N \times 7,371 \text{ bytes/call}} \end{aligned}$$

The results above show a low value of bytes exchanged through the network, what ensures a good performance of the test scenario. Despite this, the increase of the MRN Metric should alert us to watch over the scenario performance in the future.

5.1.4. CPU Usage

This metric and its results have been already presented in deliverable [4].

5.1.5. Memory Usage

This metric and its results have been already presented in deliverable [4].

5.1.6. Average Response Time

This metric and its results have been already presented in deliverable [4].

5.1.7. Network Bandwidth

This metric and its results have been already presented in deliverable [4].

5.1.8. Virtual Users

This metric and its results have been already presented in deliverable [4].

5.2. 2D Image Search Scenario.

This scenario simulation has been already assessed and appears along with its metric results in deliverable [4].

5.2.1. Sizes of Input and Output Messages

This metric has been defined and measured in the last two evaluations [1] [2]. It was expected to have a high AIMSO and AOMSO size, but since the current image search method returns results in JSON format, the values are lower compared to previous measurements.

- AN1.2DsimilarImages: example query: lactose

AIMSO [AN1.2DsimilarImages] = 233 bytes
AOMSO [AN1.2DsimilarImages] = 2023 bytes

5.2.2. Messages Rates

The comparison of the results of this metric with respect to the previous evaluation [2] shows a considerable improvement in the MRN value, passing from 482 to 662. This increase is clearly related to the Full Cloud deployment and hardware resources extension. Results are displayed below:

- AN1.2DsimilarImages: example query: lactose

IMRN [AN1] = OMRAN [AN1] = 331 msg/min
MRN [AN1] = 331 + 331 = 662 msg/min

5.2.3. Network Load

The main consequence that emerges from the low AOMSO size is a good system performance for 2D Image Search scenario. Moreover, lower values of network bandwidth usage ensure a good availability of the services as well as reduce the appearance of bottlenecks. Network load metric results are showed below:

$$ITN[ANI] = N \times (233 + 2023) = N \times 2256 \text{ bytes/call}$$
$$OTN[ANI] = N \times (2023 + 233) = N \times 2256 \text{ bytes/call}$$

5.2.4. CPU Usage

This metric and its results have been already presented in deliverable [4].

5.2.5. Memory Usage

This metric and its results have been already presented in deliverable [4].

5.2.6. Average Response Time

This metric and its results have been already presented in deliverable [4].

5.2.7. Network Bandwidth

This metric and its results have been already presented in deliverable [4].

5.2.8. Virtual Users

This metric and its results have been already presented in deliverable [4].

5.3. 3D Image Search Scenario.

This scenario simulation has been already assessed and appears along with its metric results in deliverable [4].

5.3.1. Sizes of Input and Output Messages

The current prototype has not included 3D Image Search workflow and therefore this metric has not been calculated for this evaluation.

5.3.2. Messages Rates

The current prototype has not included 3D Image Search workflow and therefore this metric has not been calculated for this evaluation.

5.3.3. Network Load

The current prototype has not included 3D Image Search workflow and therefore this metric has not been calculated for this evaluation.

5.3.4. CPU Usage

This metric and its results have been already presented in deliverable [4].

5.3.5. Memory Usage

This metric and its results have been already presented in deliverable [4].

5.3.6. Average Response Time

This metric and its results have been already presented in deliverable [4].

5.3.7. Network Bandwidth

This metric and its results have been already presented in deliverable [4].

5.3.8. Virtual Users

This metric and its results have been already presented in deliverable [4].

5.4. Multilingual Textual Search Scenario.

This scenario simulation has been already assessed and appears along with its metric results in deliverable [4].

5.4.1. Sizes of Input and Output Messages

This scenario is a special case of the Textual Search scenario with an extra call to SCA_MT to dynamically translate the input query. The result of this request is presented below along with Textual Search results:

- AN2.translator(): Zuckerkrankheit,

AIMSO [SCA_MT.translator] = 176 bytes
AOMSO [SCA_MT.translator] = 1,078 bytes

- AN1.search : diabetes

AIMSO [SCA_TSS.search] = 371 bytes
AOMSO [SCA_TSS.search] = 5,932 bytes

As showed above, the step to translate the query consumes very low sizes of AIMSO and AOMSO so we can assume that most of the consumption is associated to the SCA Textual Search component.

5.4.2. Messages Rates

MRN Metric results associated to AN2.MT step show a higher value than in previous evaluations [1][2], which confirms the improvement of the performance achieved with the Early Integrated infrastructure deployed.

- AN2.MT messages rate

$$\begin{aligned} IMRN [AN2] &= OMRAN [AN1] = 218 \text{ msg/min} \\ MRN [AN2] &= 218 + 218 = 436 \text{ msg/min} \end{aligned}$$

- AN1.search : diabetes

$$\begin{aligned} IMRN [AN1] &= OMRAN [AN1] = 40 \text{ msg/min} \\ MRN [AN1] &= 40 + 40 = 80 \text{ msg/min} \end{aligned}$$

5.4.3. Network Load

Despite the increase of the network bandwidth consumption due to the additional step to translate the query, the overall size of network requirements does not suffer a significant change. The ITN and OTN metric results are shown below:

$$\begin{aligned} ITN [AN1] &= N \times (176 + 1078 + ITN [TSS]) = N \text{ calls/min} \times 7557 \text{ bytes/call} \\ OTN [AN1] &= N \times (1078 + 176 + ITN [TSS]) = N \text{ calls/min} \times 7557 \text{ bytes/call} \end{aligned}$$

5.4.4. CPU Usage

This metric and its results have been already presented in deliverable [4].

5.4.5. Memory Usage

This metric and its results have been already presented in deliverable [4].

5.4.6. Average Response Time

This metric and its results have been already presented in deliverable [4].

5.4.7. Network Bandwidth

This metric and its results have been already presented in deliverable [4].

5.4.8. Virtual Users

This metric and its results have been already presented in deliverable [4].

5.5. 2D Semantic Image Search Scenario.

The 2D Semantic Image search scenario implements use case functionalities where an end-user search over a list of images by using some semantic concepts and categories is predefined. To do so, a semantic search request is simulated where the user searches for images related to some term list. This URL requests the 2D Semantic Search workflow:

<http://atos1-khresmoi.ms.mff.cuni.cz:8080/khresmoi-semantic-search/rest/terms/search?termList=Anatomies%3A%3Ahttp%3A%2F%2Fbioontology.org%2Fprojects%2Fontologies%2Fradlex%2FradlexOwlDIComponent%23RID1554|protein>

The main components involved in this workflow are:

- Co-occurrence: SCA Co-occurrence component is used to search for images related to some predefined concepts like: Pathologies, Anatomies and Modalities. This is an example request for SCA Co-occurrence:

http://atos2-khresmoi.ms.mff.cuni.cz:8080/khresmoi-Cooccurrence/rest/cooccurrence/search?cooccurrenceStatistics=Anatomies%2CModalities&resultType=Radiology+Reports&columns=¤tPage=1&count=10&_form=%2Fcooccurrence%2Fsearch&q=Anatomies%3A%3Ahttp%3A%2F%2Fbioontology.org%2Fprojects%2Fontologies%2Fradlex%2FradlexOwlDIComponent%23RID1385|heart%22%20-c%2060

- Disambiguator: is used to disambiguate terms chosen for searching. This URL is a request example for SCA Disambiguator:

<http://atos2-khresmoi.ms.mff.cuni.cz:8080/khresmoi-dissambiguator/rest/dissambiguator/query=LADA&language=en>

- ParaDISE: is used to retrieve information related to some images. This URL is an example of searchImages related to diabetes:

<http://atos2-khresmoi.ms.mff.cuni.cz:8080/khresmoi-ParaDISE/rest/paradise/searchImages?caption-query=diabetes>

Subsequently, some of the main metrics results related to this scenario after its execution simulation are presented in this section. These metrics are related to main performance facets as well as some specific scenario measures.

5.5.1. Sizes of Input and Output Messages

This metric has been measured for the first time in this document for 2D Semantic Image Search scenario. In summary, high values of AOMSO for the different calls that compose the workflow have been observed. These occur due to the nature of the data queried and the complexity of the workflow.

The query examples taken to simulate the workflow are more complicated than before because they are composed of different concepts, terms, etc. For instance: *search?termList=Anatomies=http://bioontology.org/projects/ontologies/radlex/radlexOwlDIComponent/23RID1554|protein*

- AN2.Cooccurrence.call:

AIMSO [SCA_Cooccurrence.cooccurrenceCall] = 203bytes
AOMSO [SCA_Cooccurrence.cooccurrenceCall] = 16,025 bytes

D6.5.2. Evaluation of the “Early Integrated Infrastructure”

- AN2.ParaDISE.getImageInfo:

AIMSO [SCA_ParaDISE.getImageInfo] = 296 bytes
AOMSO [SCA_ParaDISE.getImageInfo] = 73,354 bytes

- AN1.SIS.search :

AIMSO [SCA_2DSIS.search] = 256 bytes
AOMSO [SCA_2DSIS.search] = 10,389 bytes

As showed above, AOMSO results for SCA_ParaDISE return the highest value of the workflow. The following metrics will provide a view of the capacity of the system to deal with this amount of information.

5.5.2. Messages Rates

The metric results presented below show the Message Rates values obtained after the simulation. The main conclusion regarding these results is that despite the high AOMSO values, the system is able to deal with a good number of messages, even better than Textual Search MRN. This fact could rely on an optimal integration of component in the workflow:

- AN2.Cooccurrence

IMRN [AN2] = OMRAN [AN1] = 237 msg/min
MRN [AN2] = 237 + 237 = 474 msg/min

- AN2.ParaDISE

IMRN [AN2] = OMRAN [AN1] = 218 msg/min
MRN [AN2] = 218 + 218 = 436 msg/min

- AN1.search : diabetes

IMRN [AN1] = OMRAN [AN1] = 267 msg/min
MRN [AN1] = 267 + 267 = 534 msg/min

5.5.3. Network Load

The ITN metric results displayed below shows important network bandwidth consumption as a consequence of the results of the previous metrics. Currently, the Early Integrated Infrastructure prototype is able to deal successfully with the workflow requirements, but it must be checked in future refinements as well as the network capacity in productions environments.

$$ITN [AN1] = N \times (203 + 16,025 + 296 + 73,354 + ITN[2DSIS]) = N \text{ calls/min} \times 100524 \text{ bytes/call}$$

$$= N \times 98 \text{ Kbytes/call}$$

$$ITN [AN1] = N \times (73,354 + 296 + 16,025 + 203 + ITN[2DSIS]) = N \text{ calls/min} \times 100524 \text{ bytes/call}$$

$$= N \times 98 \text{ Kbytes/call}$$

5.5.4. CPU Usage

CPU Usage in Figure 2 below displays the graphical consumption of CPU resources during Semantic Search consumption. It is observed that this workflow requires lower percentage of processing capacity with a highest value of 22% used. Therefore, we can conclude that the Semantic Search workflow can be executed without difficulties for the users, requests and time parameters predefined.

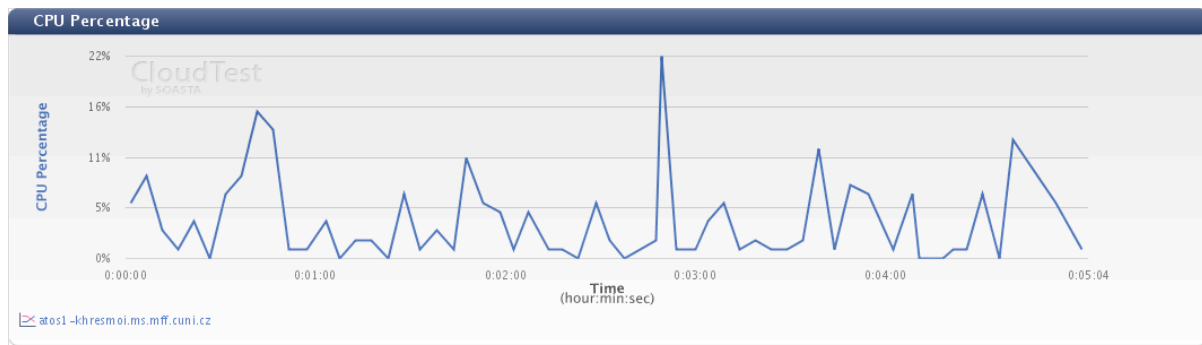


Figure 2: CPU Usage Semantic Search

5.5.5. Memory Usage

Memory Usage graphic below in Figure 3 shows a quite stable memory consumption in an interval of 420-425Mega Byte (MB). This means that the system is currently able to deal with simultaneous request to Semantic Search workflow with a very low Memory consumption. This feature ensures availability of the system and its capacity of responding is more than enough for current scenario parameters predefined.

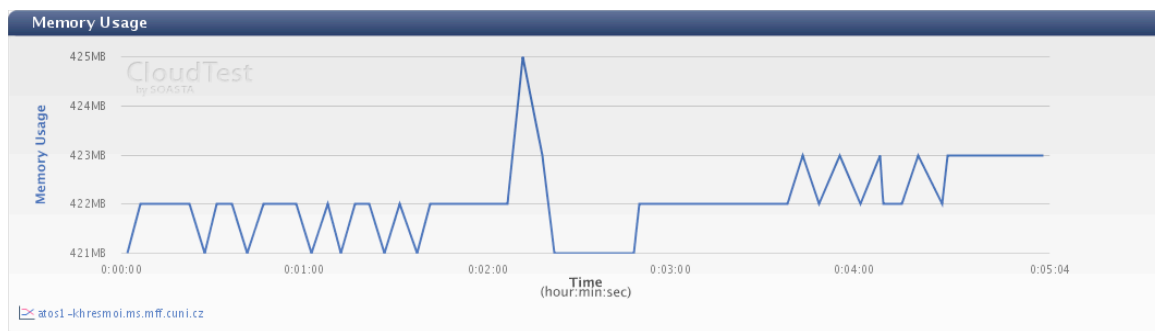


Figure 3: Memory Usage Semantic Search

5.5.6. Average Response Time (ART)

The ART metric shows low values most of the time, although Figure 4 shows some peaks in the response time that could be caused by overlapping of virtual users requests. Also, the high values of AOMSO could delay some requests due to data being transferred from and to the server.

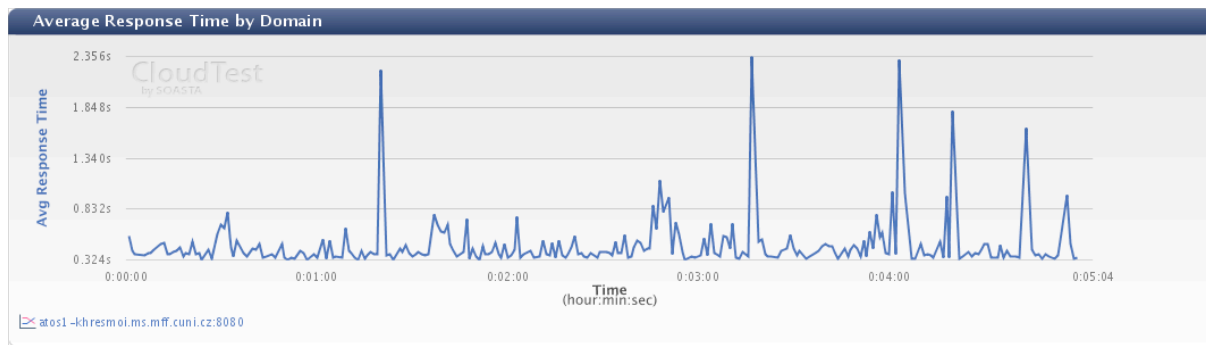


Figure 4: ART Semantic Search

5.5.7. Network Bandwidth

According to metric results in Subchapter 5.3.3, the Average Network Bandwidth consumption is about 100Kb/request. Despite this, Figure 5 display some peaks in network bandwidth usage that may be caused by overlapping user requests, and therefore requested output data are transferred simultaneously.

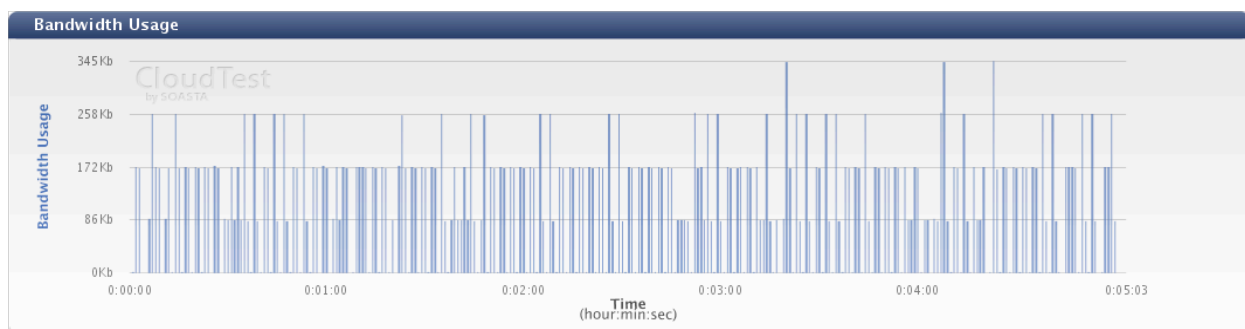


Figure 5: Network Bandwidth Semantic Search

5.5.8. Virtual Users

The virtual users diagram displayed in Figure 6 below shows an optimal execution of user requests in terms of overlap. This is due to the capacity of the system and the way it deals with several requests in a minimum time. Therefore the delays are not aggregated over the simulation. This is important because aggregated delay could produce overlapping user and even message loss in the worst cases.

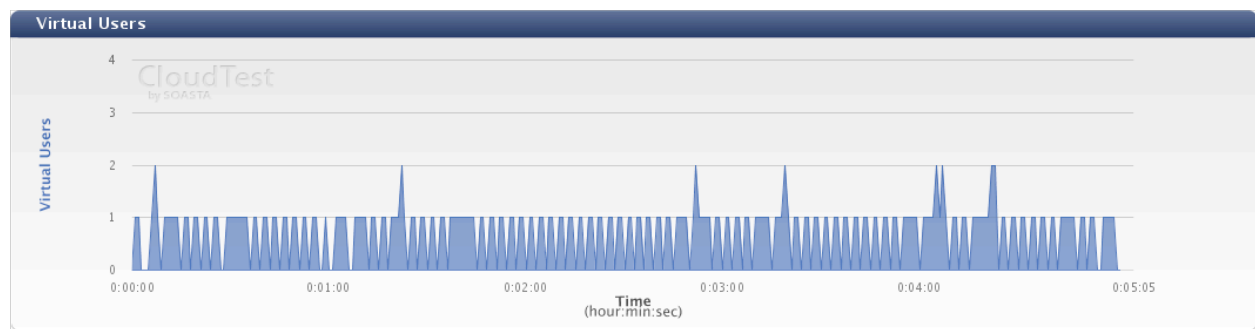


Figure 6: Virtual Users Semantic Search

6. Conclusions

This document summarizes performance of the system after joining the Full Cloud infrastructure and the integrated software architecture in order to offer an overview of the Khresmoi platform in terms of requirements and functionalities. Both aspects respond to the need of offering a scalable and available platform as well as an integrated solution where most of the Khresmoi components would be working together.

Besides this, the document assesses the current platform in order to check the scalability and the integration status. In order to perform this assessment, some predefined metrics have been used and revisited according to different criteria (such as hardware resources demanded, network consumption, etc.). Some of these metrics have been defined according to specific Khresmoi workflow scenarios that cover the main uses cases. The main evaluation outcomes can be summarized as follows:

- A better performance result of components that implement workflows has been observed, mainly due to the integration in cloud cluster.
- Also, the current prototype employs a bigger size of inputs and outputs due to larger sized databases and indexes collected during the project at this stage. This causes a lower execution time of use cases employing textual search due to the time required to access to the backend.
- Moreover, the ensemble deployment in the same network offers a lot of benefits in complex workflows, reducing time delays when calling to components hosted out of the network.

Finally, this document can be seen as the almost-final step in order to obtain a final integrated structure able to deal with all the Khresmoi system requirements in a final production environment.

7. References

- [1] 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.
- [2] Martinez Rodriguez, Ivan and Tinte Garcia, Miguel Angel, Deliverable D6.3.3 “Prototype and Evaluation of the ‘Full Software Architecture’” in KHRESMOI Project, Seventh Framework Programme.
- [3] Martinez Rodriguez, Ivan and Tinte Garcia, Miguel Angel, Deliverable 6.5.1 “Specification of the ‘Early Integrated infrastructure’” in KHRESMOI Project, Seventh Framework Programme.
- [4] Martinez Rodriguez, Ivan and Tinte Garcia, Miguel Angel, Deliverable D6.4.3 “Prototype and Evaluation of the ‘Full Cloud Prototype’” in KHRESMOI Project, Seventh Framework Programme.
- [5] Martinez Rodriguez, Ivan and Tinte Garcia, Miguel Angel, Deliverable D6.4.2 “Evaluation of the “Early Cloud Infrastructure” and specification refinement for the “Full Cloud infrastructure”” in KHRESMOI Project, Seventh Framework Programme.
- [6] 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.