



OKKAM – Enabling a Web Of Entities
Grant Agreement No. 215032

D12.3: OKKAM in Numbers V2

Document Number	Deliverable D12.3
Document Title	OKKAM in Numbers V2
Version	1.2
Status	Final
Work Package	12
Deliverable Type	Report
Contractual Date of Delivery	October 2008
Actual Date of Delivery	February 2009
Responsible Unit	L3S
Contributors	ALL
Keyword List	Performance assessment
Dissemination level	RE

Change History

Version	Date	Status	Author (Company)	Description
1.1	3.3.2009	draft	Claudia Niederée (L3S)	Transfer of the online version into a word document, minor revision as a consequence of this transfer, addition of Q4 numbers and year 1 resource overview
1.2	5.3.2009	final	Gleb Skobeltsyn, Zoltan Miklos (EPFL)	Quality Control

Executive Summary

This report summarizes the results that have been achieved in the first 13 months of the project in the action line “OKKAM in Numbers” (OIN). It is the goal of this action line to foster quantitative evaluation of project results and progress for easing project assessment by the Consortium, the project coordinator, the European commission and also by other researchers and practitioners.

Work in the action Line OIN in the second phase of the project focused on collecting target performance numbers, broadening the collection of numbers to further areas of the projects and performing experiments for collecting numbers about the current performance of the OKKAM ENS and further OKKAM component. Work in this second phase builds upon the work of the first phase of the project of setting up the process for systematic performance evaluation in the different areas of the project.

Work on the action line OKKAM in Numbers will continue in parallel to the other project activities and its results will be documented in further versions of the OKKAM in Numbers report.

Table of Contents

1. INTRODUCTION.....	6
2. NUMBERS ON ACHIEVEMENTS	7
2.1. REPOSITORY POPULATION.....	7
2.1.1. Target Numbers	7
2.1.2. Current and Past Numbers	7
2.2. ENS STORAGE AND MATCHING PERFORMANCE.....	8
2.2.1. Response time	9
2.2.1. Indexing Time	17
2.2.2. Response Quality	18
2.2.3. Query Rate	22
2.3. PERFORMANCE OF APPLICATIONS	25
2.3.1. Entity-centric Semantic Search Engine.....	25
2.3.2. Entity-centric Organizational Knowledge Management.....	25
2.3.3. Entity-centric Content Authoring Application	27
2.4. PERFORMANCE OF OKKAMIZATION.....	27
2.4.1. OKKAMization Quality.....	28
2.4.2. OKKAM empowered tool set	28
3. NUMBERS ON INVOLVEMENT	30
3.1. ESTABLISHED COMMUNICATION CHANNELS	30
3.1.1. Project meetings	30
3.1.2. Telephone Conferences.....	31
3.1.3. Use of Intranet	31
3.1.4. Use of Mailing Lists in OKKAM.....	32
3.2. NUMBERS ON OKKAM ACADEMY.....	34
3.3. INVOLVEMENT IN COLLABORATIVE SOFTWARE DEVELOPMENT	34
3.3.1. Requirements Collection.....	34
3.3.2. Collaborative Software Development	36
4. NUMBERS ON IMPACT.....	37
4.1. DISSEMINATION ACTIVITIES.....	37
4.2. USE OF PUBLIC OKKAM WEB PAGE	37
4.3. NUMBER OF PUBLICATIONS.....	38
5. NUMBERS ON TIMING	40
5.1. MILESTONES.....	40
5.2. DELIVERABLES.....	40
5.3. REVIEWS	40
6. NUMBERS ON RESOURCES	41
6.1. YEAR 1 OVERVIEW.....	42
6.1.1. Consumption of Resources.....	42
6.1.2. Splitting of Costs.....	43
6.2. NUMBERS FROM OKKAM Q1	44
6.2.1. Use of Resources per Partner.....	44
6.2.2. Use of Resource per WP	44
6.3. NUMBERS FROM OKKAM Q2.....	46
6.3.1. Use of Resource per WP	46
6.3.2. Use of Resources per partner.....	46
6.4. NUMBERS FROM OKKAM Q3.....	48
6.4.1. Use of Resource per WP	48

6.4.2.	<i>Use of Resources per partner</i>	48
6.5.	NUMBERS FROM OKKAM Q4.....	50
6.5.1.	<i>Use of Resource per WP</i>	50
6.5.1.	<i>Use of Resources per partner</i>	50

1. Introduction

It is the goal of the action line “OKKAM in Numbers” (OIN) to effectively capture and monitor the performance of the OKKAM project in a comprehensive quantitative fashion. The performance and success of an Integrated European project such as OKKAM clearly does not only depend on the innovative technology and research results produced in the project. There are various other factors that substantially influence project success. Some of these factors are project-specific, such as ensuring sustainability and the need for a high level of adoption in OKKAM, some are specific for large RTD project in IT in general, such as ensuring sufficient dissemination of results, whereas a last group of factors such as a sufficient involvement of the partners in the project are of a more generic nature for the functioning of projects.

For monitoring the performance of OKKAM, thus, a wide variety of success factors has to be considered for OIN. The performance parameters currently collected for OIN have been selected based on this understanding and are organized into the following five areas:

- **Achievements:** numbers characterizing what has already been achieved in the project in R&D
- **Involvement:** numbers related to the involvement and collaboration of the partners in the project
- **Impact:** numbers related to project visibility and potential impact
- **Resources:** numbers related to the use of resources in the project
- **Timing:** numbers related to project timing, milestones and deadlines

The individual parameters monitored in the five areas as well as the observed values and, where possible, the target performances are documented in this report. They are an important means for the project management for making decisions about priorities and corrective actions in the project.

It is foreseen that during the in later phases of the project further parameters that require monitoring might be identified or other parameters might be dropped reflecting further lessons learned in how to monitor project progress.

In general, this document is considered as a living document, which is regularly adapted to the dynamics of the OKKAM project. This can already be seen in the transition between OIN V1 and OIN V2, where the goal is to a) broaden the view of performance parameters considered and b) make the results presented in the report more digestible to foster their impact on future project decisions and work.

2. Numbers on Achievements

This area covers numbers characterizing what has already been achieved in the project in RTD. This includes contributions to the OKKAM ENS in the form of implemented components, OKKAM-empowered tools and OKKAMizers. In addition, this area also contains numbers reflecting progress in OKKAM-related research that will become part of the OKKAM infrastructure only in a mid-term to long-term perspective. A further source that contributes performance values to OIN in the area of “Achievements” are the three applications on top of OKKAM.

The numbers in this section are structured into the following areas:

- Repository Population
- ENS Storage and Matching Performance
- Performance of Applications
- Performance of OKKAMization

Numbers in this area reflect progress towards the project goals most directly. Furthermore, numbers on research results will reflect the progress beyond the state of the art in the respective areas. It has to be noted here, that the by decision of the consortium the first year of OKKAM mainly focussed on getting the OKKAM ENS V1 running rather than of covering deeper research questions. This is also reflected by the performance parameters available for OIN V2.

2.1. Repository Population

The population of the entity repository, i.e. the number of entities managed in the ENS is a crucial factor for the acceptance of the ENS. It is the goal that for the major number of the request the searched ID is already available in the repository. To which degree this can be achieved depends on type of entities and the application. However, it was decided to select a set of general purpose entity types (e.g. persons, locations, etc.) and to also create a relevant population for the considered entity types (e.g. by the import of locations from a complete collection).

2.1.1. Target Numbers

For the repository population two milestones have been defined. It is the goal to have at least 1 Million entities in the repository after year 1 (MS4) and to have at least 2 Million entities in the repository in month 26 (MS 14). The first milestone has already been achieved (see below) and the consortium is confident also to reach the second Milestone.

2.1.2. Current and Past Numbers

The current repository population (February 2009) consists of 1.034.840 Entities, made up of persons, organizations, proteins, locations, special entities, and SAP products). The numbers per entity type are shown in the graphics below. Special entities are entities that have been manually added (e.g. consortium members). The graphics also shows the repository population at an earlier point in time, namely at the time of the Pre-Review (October 2008). At this point in time the repository did not yet contain the application specific entities: Proteins and SAP products.

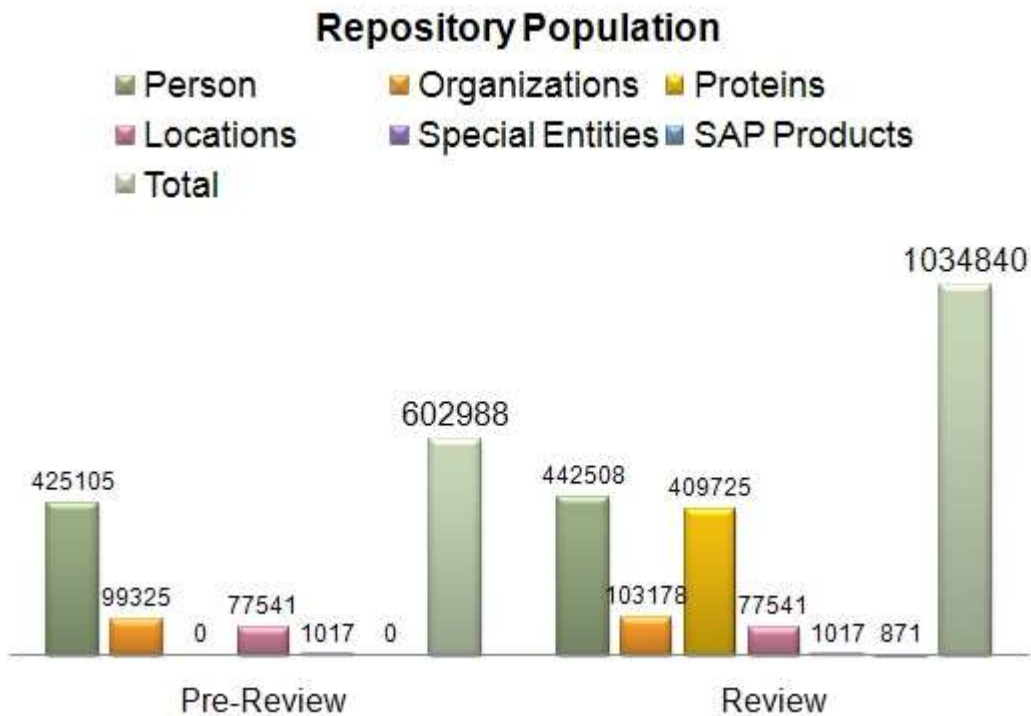


Figure 1: Split up of the Population of the Entity Repository in October 2008 (Pre-Review) and February 2009 (Review)

2.2. ENS Storage and Matching Performance

One of the most important outcomes of the OKKAM project will be the OKKAM Entity Name System (ENS). The purpose of the ENS – in a nutshell – is the management of entity identifiers and the fostering of their global re-use. It is crucial for the success and the wide acceptance of the ENS to reach a sufficient level of performance. This refers to the quality of the service delivered by the ENS as well as to its efficient and reliable service delivery. It is crucial to develop a clear understanding within the consortium about the target performance that has to be achieved by the ENS in order to make its use attractive. The knowledge about the target performance can be used to drive design and development decisions in building the ENS as well as for monitoring related success factors.

When looking into the target performance, it is also necessary to consider scalability issues, since the ENS should also be able to perform efficiently in case of a strong growth of the repository and/or the user community (with an associated growth of the request load that the ENS has to handle). This does not mean that the current setup built in the project has to be able to deal with very large numbers of entities and requests. Rather the chosen architecture and approach in building the ENS has to be designed in a way that enables scaling, i.e. - given a sufficient investment in additional hardware - the ENS can be distributed in a way foreseen in the architecture, such that the target performance stays on the expected level.

For considering and monitoring the target performance the following parameters have been identified:

- Response time of the Entity Name Server (average and maximum)
- Indexing time of the Entity Name Server

- Response quality of the Entity Name Server (measured as success@top K)
- Query Rate of the Entity Name Server (average)

In the following, target numbers for these parameters as well as the current performance numbers measured in experiments are presented and discussed. Where applicable we also present considerations and measurements with respect to scalability. Furthermore, the discussion is completed by additional considerations and comments that might be required for the assessment of the presented numbers as well as for understanding possible factors that might still require an adaptation of these numbers at a later point in the project.

2.2.1. Response time

Response time reflects the time that is required by the ENS to process an entity request. This covers the time from posing the request to receiving back the set of matching results containing one or more OKKAM ID(s).

It might be worth mentioning that for the (envisioned) scenario when a document is being OKKAMized, individual response times of each entity might be not that important as the total time spent for OKKAMization. This observation opens up interesting optimization possibilities which we plan to investigate later.

Target Performance

Based on an analysis of the current performance and considerations on the acceptance of the system, a target performance of 500 ms has been defined for the average response time of the ENS. For the maximum response we aim to not exceed the target performance number of 1 second (sub-second response time).

Past and Current Measurements (Single Server Setup)

As an important parameter, response time is measured regularly during the project to monitor the development of this parameter. For easing performance improvements, in current measurements response time is split up into the time spent in the various layers of the system. This enables to identify layers that require further performance tuning. Here we report on the measurement on the current version of the system in February 2009.

Experimental Setup

For dataset we used the 1.005.004 entities as included in OKKAM at the beginning of February 2009, and 500 requests generated using the cogito extractor by EXP from collected web content. We manually processed each request and identified their OKKAM ids. We imported these entities in an empty OKKAM Node and at various intervals we measured the processing time needed by OKKAM Match and OKKAM Store for evaluated each request. For dataset we used the 1.005.004 entities as included in OKKAM at the beginning of February 2009, and 500 requests generated using the cogito extractor by EXP from collected web content. We manually processed each request and identified their OKKAM ids.

Execution Time for Requests over Various Numbers of Entities

The following plot shows the average execution time for various numbers of entities in OKKAM. As shown, OKKAM Match is able to maintain the execution time of requests regardless of the number of entities in OKKAM. This is also expected since the number of matching candidates

returned by OKKAM store to OKKAM match is constant (for the v1.1 prototype we use top-200 entity profiles returned from OKKAM Store) and independent of the repository site. In addition, we see the time taken by OKKAM Store increases with the size of the entities. This is because OKKAM Store needs to retrieve the candidate entities from the index, and this index increases when the number of entities increases. The distributed version of OKKAM Store would handle this issue in future by adding more servers such that the index size per machines does not exceed a certain size and, thus, the overall processing time remains acceptable.

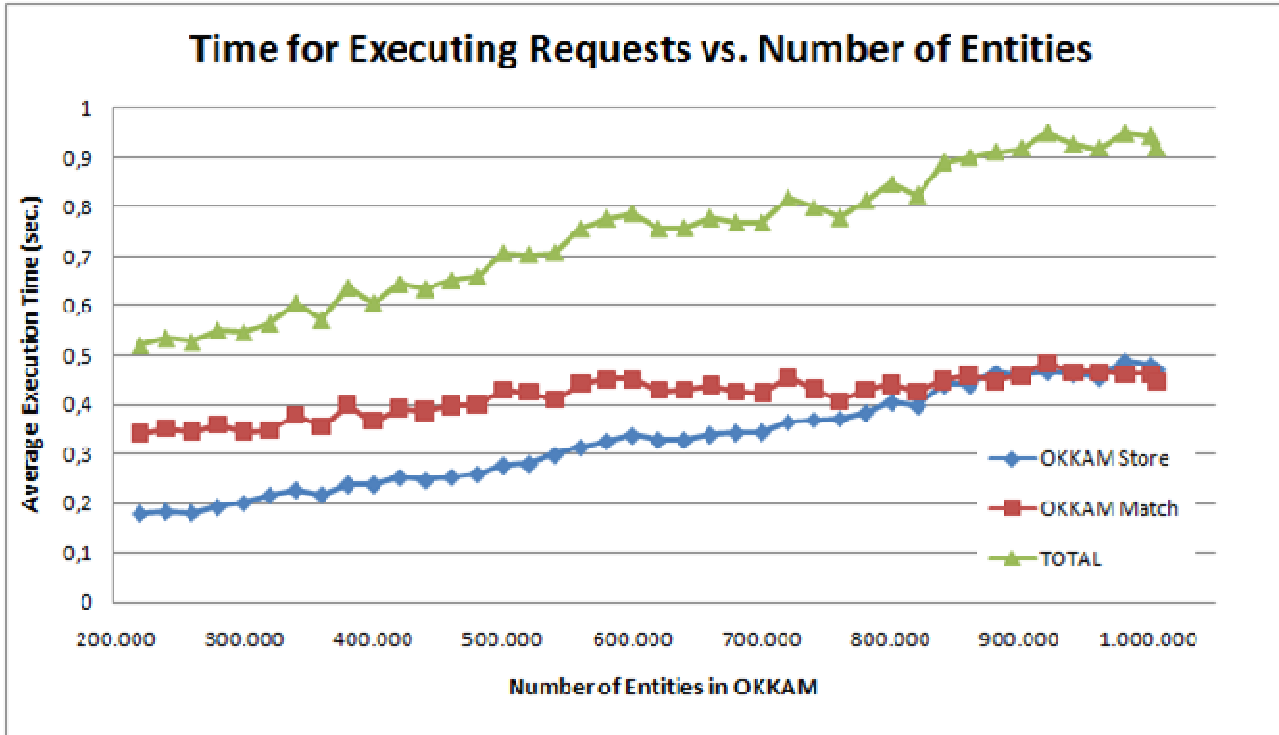


Figure 2: Development of Response Time with growing Number of Entities

Detailed Execution Time for Requests

Following the above experiment, we wanted to investigate further the time taken for OKKAM Match and Store to execute requests. For this, we collected the execution times for the last evaluation of the requests from our above experiment and grouped them into different ranges, i.e., requests executed in less than one second, requests executed between one and two seconds, etc.

The following plot illustrates the number of requests executed in different time ranges. This experiment revealed that around 70% of the requests were answered in less than one second, around 25% of the requests need several seconds, and for around 5% the time is quite high. In the upcoming weeks, we plan to further investigate this behaviour and try to identify the reasons for which some requests need so much time.

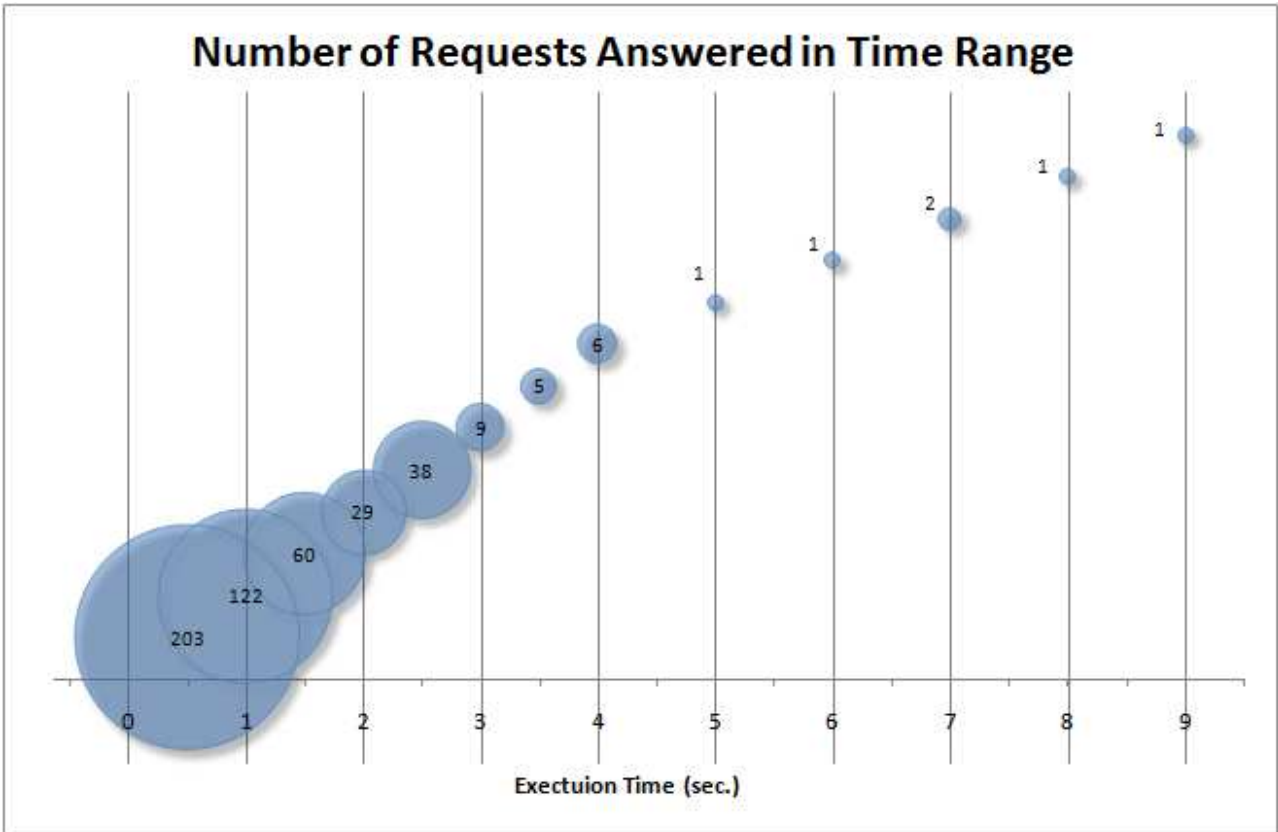


Figure 3: Analysis of the number of Requests answered in the individual Time Ranges

Split Up of Execution Time within OKKAM match

OKKAM Match performs a number of steps for evaluating requests. This includes, parsing of the given requests and converting it into an internal java object, selecting the appropriate matching module for the specific request, generating a query plan for OKKAM Store, executing this query plan, and final execute the matching algorithm.

The following plot shows the time needed by each of the OKKAM Match steps when executing 500 requests. The time is shown in nanoseconds, and logarithmic scale of ten. As shown in the plot, the time needed by the first three steps is extremely small and the major portion of the time is consumed by the Store and matching algorithm.

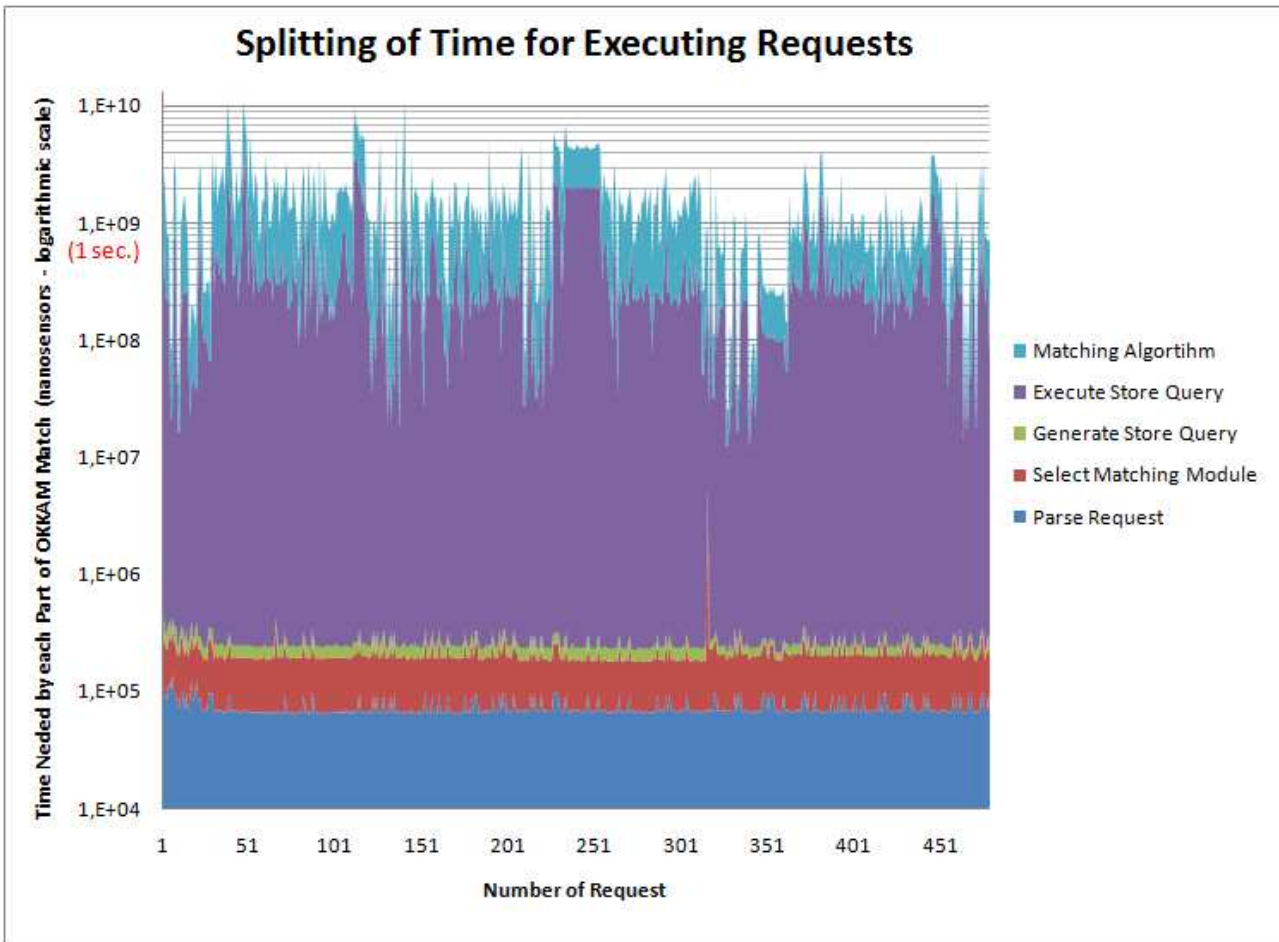


Figure 4: Splitting of the time over the different types of processing in the Matching Layer

Scalability Considerations

For ensuring scalability of the system it is important to analyse the behaviour of the response time, when the repository size is growing. Clearly, the target here is that the employed technology ensures that the response time is not seriously effected by a growth of the repository.

The component to be considered for this is the storage layer, because it is the task of the storage layer - given a request - to narrow down from all entities in the repository to a reasonable constant number of matching candidates, which are dealt with in the matching layer.

Hence, from the scalability point of view, the setup of the storage layer is similar to what Web search engines are dealing with: return top-k documents (entities) matching a query (matching request). Employing distributed inverted indexes (Apache Solr) as well as distributing the entity storage (Apache HBase) permits coping with increasing data and query volumes by adding new servers as discussed in Deliverable D6.1.

Setup

Apart from the single-machine setup used for the current prototype we have deployed an experimental distributed setup for the storage layer (aka OKKAMstore). We will refer to this setup as “distributed storage layer setup” in the rest of the document. It uses 7 computers: One server,

shown in brown (4 cores, 2.4 GHz, 8GB RAM) and six regular machines, shown in white (1 core, 3.0 GHz, 2GB RAM) deployed in a local-area network as shown below.

At the moment Hbase (responsible for the entity storage) is running on the server and the index is distributed and replicated using Apache Solr amongst regular machines. The index is divided equally amongst two Solr shards. Each Solr shard contains many replicas of the half-index it is responsible for:

The server can also execute one or more instances of the OKKAMstore API. But for measuring response time (i.e., this experiment) we only run a single instance of the OKKAMStore API. The figure below explains the setup:

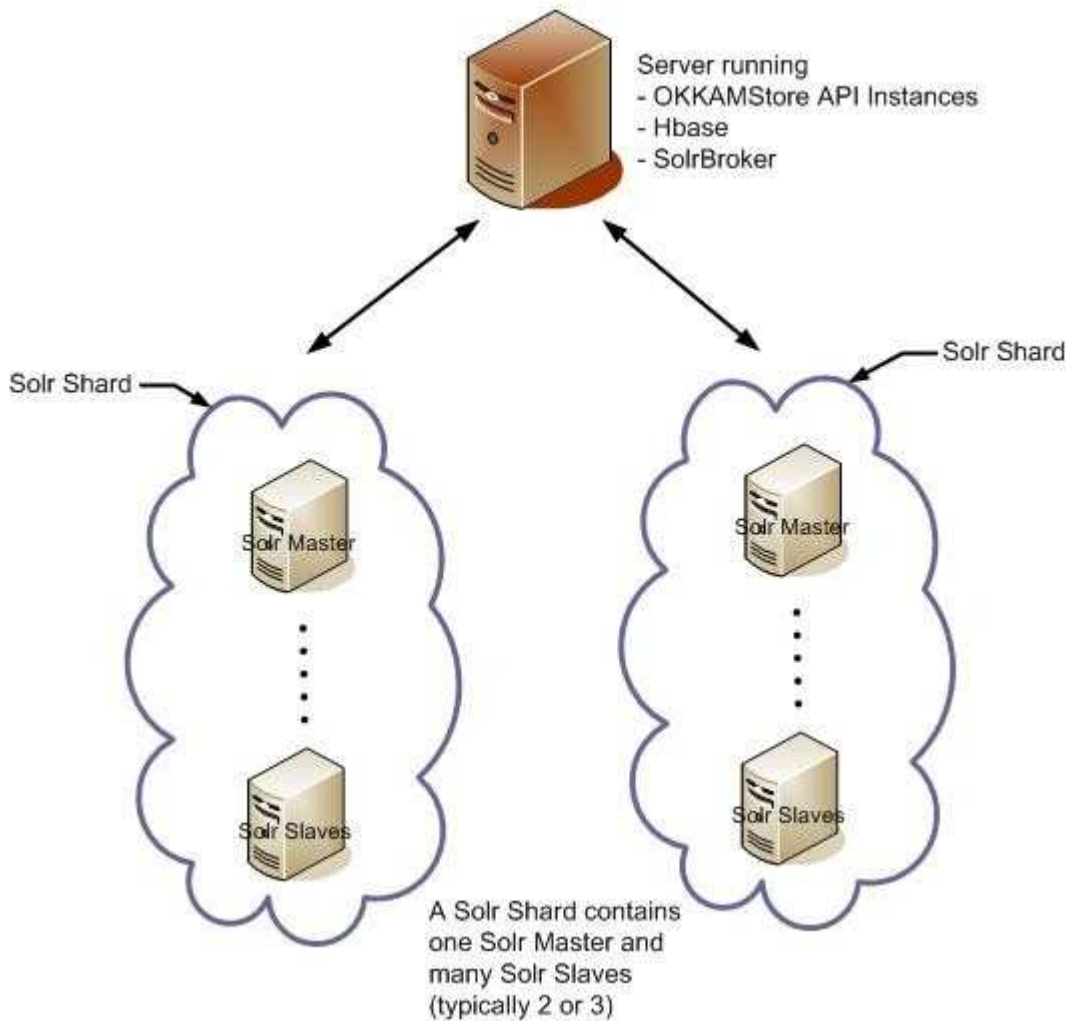


Figure 5: Configuration of the Distributed Setup for the Experiments

Experiments

We performed several experiments to verify scalability and measure query response times. We first studied the response time when all of the caches were flushed before the experiment. We repeated this experiment to obtain an average over 2 runs. Then we measured the response time in the presence of caching. Finally we studied the case when the index is in the main memory.

For all experiments, we use a fixed set of 300 queries and we run it against the same set of entities (3.3 million entities). Our initial experiments show that our system has the stable query execution time without caching, which can be improved if we allow caching or move the index to the main memory.

Experiment 1 - Setup summary: caches flushed, index on disk

To test the distributed setup we plot query response times (only for OKKAMstore) for the set of 300 queries. All caches are flushed before running the experiments (in fact all servers involved in each experiment are rebooted before the experiment is conducted). The figure below shows query processing times against a repository of 3.3M entities. The average query processing time is around 1 second including both: the index lookup and fetching 50 entities per query from the entity storage. It is also visible that once caches "warm up", the processing time becomes faster. Also recall that in this test, the entity storage (HBase) is implemented on 1 single server, we plan to investigate the distributed case in the near future.

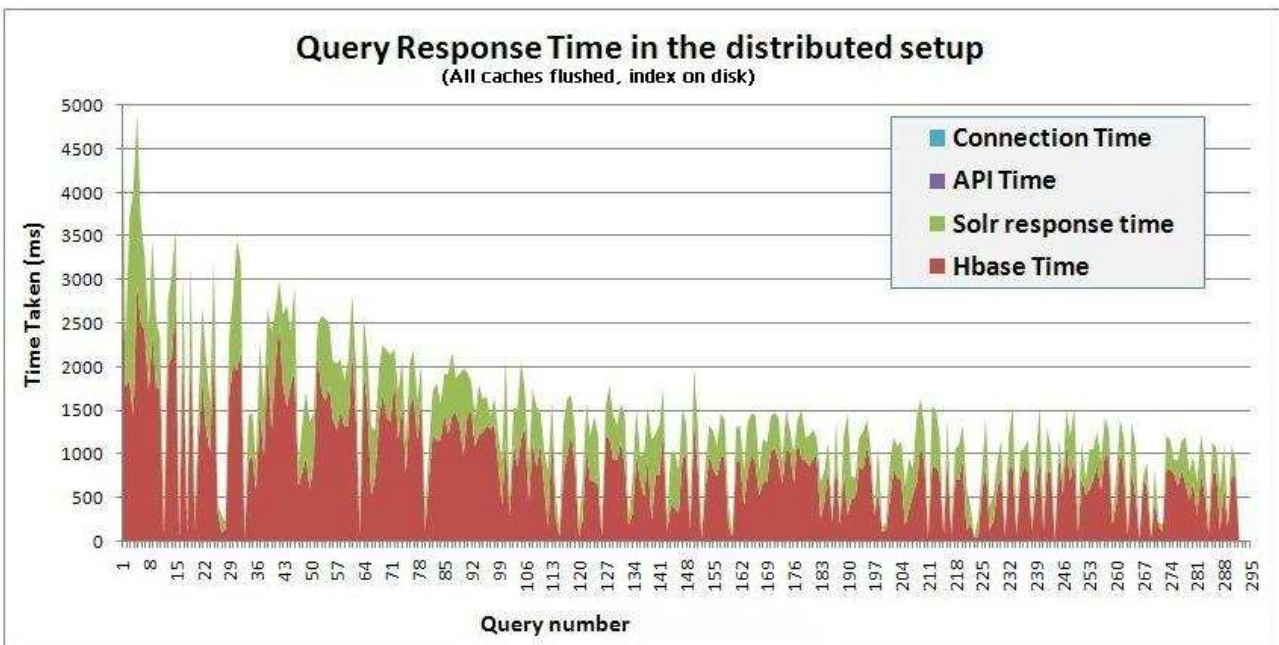


Figure 6: Results from Experiment 1

The second plot below shows query response times for the same queries and setup (caches flushed, index on disk) in 2 independent runs as well as the average query response time. It shows that the response time per query remains relatively stable. As said earlier, before each run all caches are flushed, in fact all servers involved in each experiment are rebooted before each experiment is conducted.

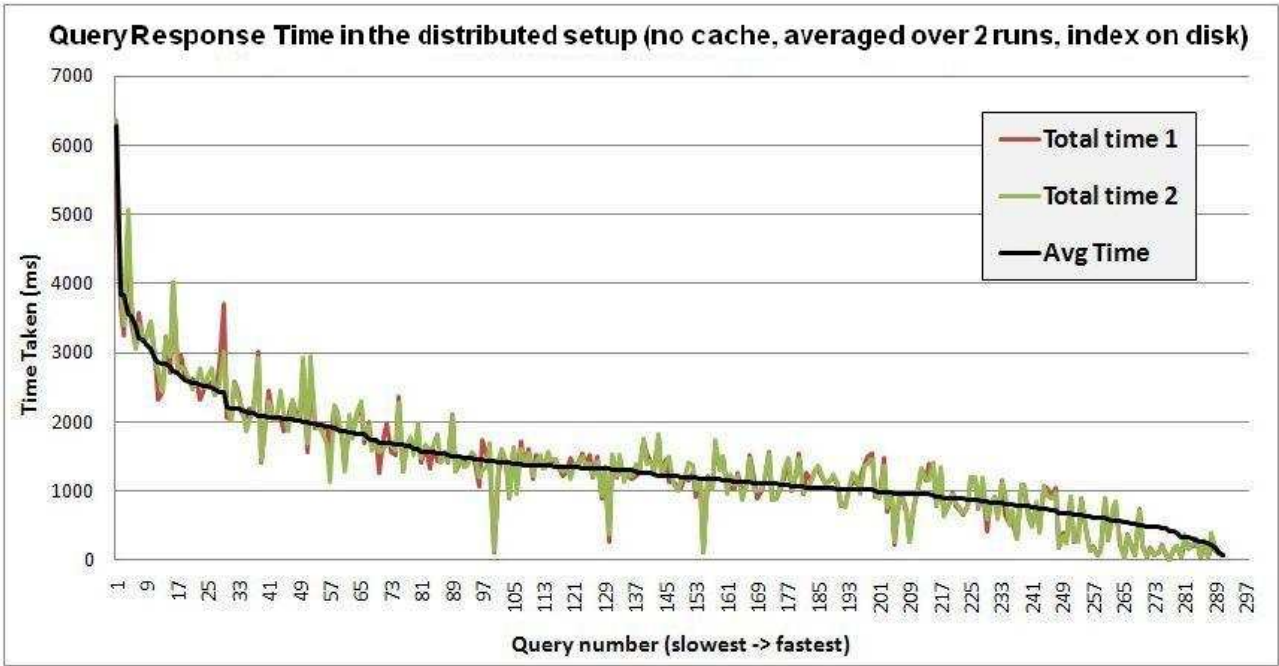


Figure 7: Results from Experiment 1 (comparison between 2 runs)

Experiment 2 - Setup summary: caching turned on (for Hbase and Solr), index on disk

We also did the following measurement: We run the same set of 300 queries two times and we record the performance at the second run. In this case due to caching at all levels of the system, we observed that the performance drastically improves, we measured an average response time of about 70ms. The figures are displayed in the figure below.

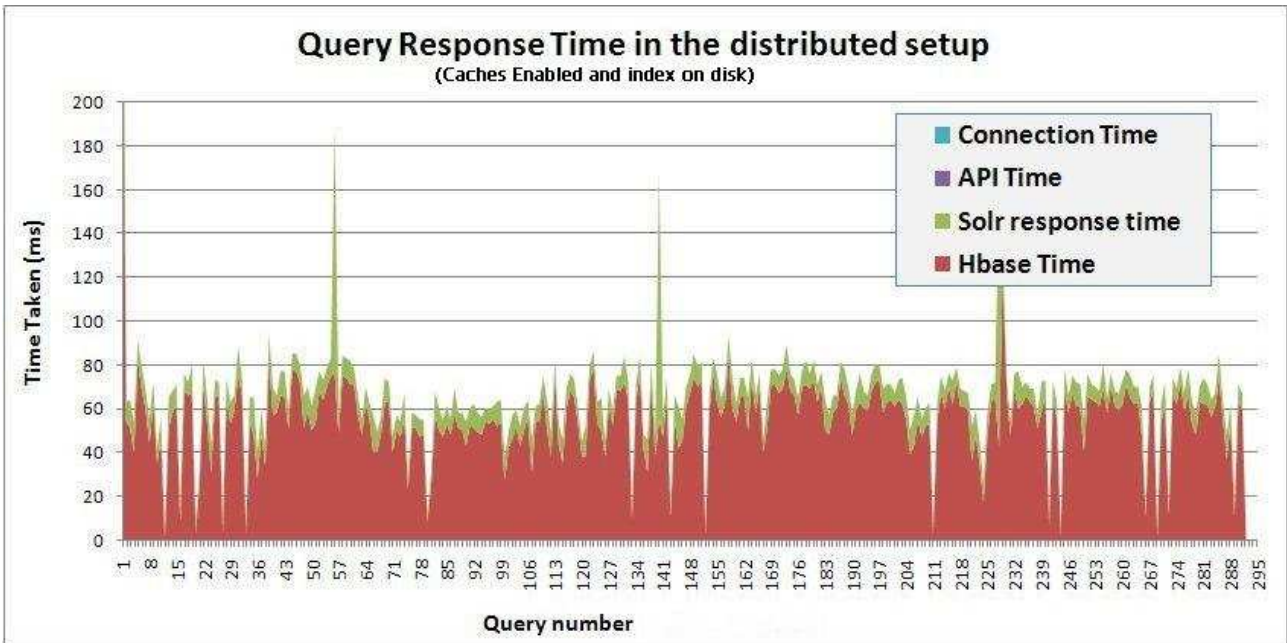


Figure 8: Results from Experiment 2

To have a more realistic estimates of the distributed system performance we would need more queries such that the caches have time to "warm up" before we start measuring the performance.

Experiment 3 - Setup Summary: only HBase cached, index in memory but not cached

We did the following experiment. First, we run our 300 queries against the data, on the setup of Experiment 1 (all the machines restarted before). Then we rebooted the Solr machines and shifted the index from disk to the main memory. We did not reboot the machines running HBase, so in this way the entity files are in the cache. Then we run our 300 queries again and measured the response time. The next figure shows the query processing time (aka response time) we measured. The queries were posed against a repository of 2 million entities.

**Query Response Time in the distributed setup
(index in-memory /hbase cached)**

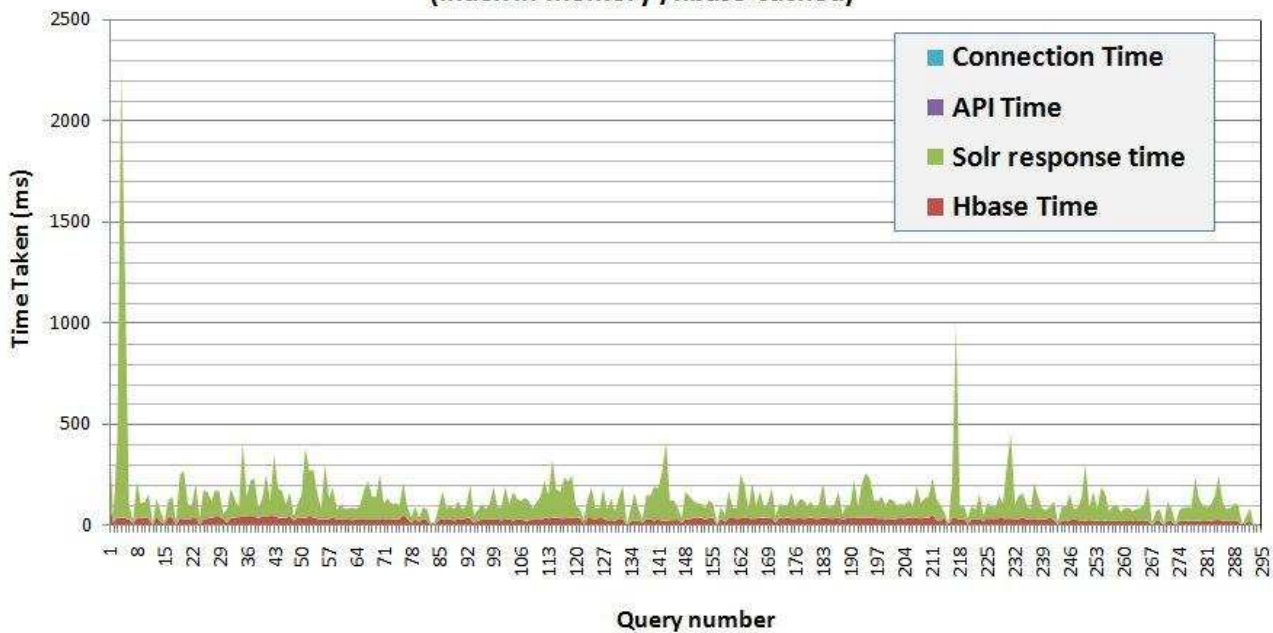


Figure 9: Results from Experiment 3

The above figure shows that we can improve the performance of the ENS even if the cache of Solr is flushed, if we move the index into the main memory. The performance improvements we get in this way are indeed significant, we get approximately 10 times better figures.

Average Response Time in Experiments 1 to 3:

We measured the average response time of the ENS. We observed the following average query response times for the various experiments performed above:

- Experiment 1: 1.3 sec
- Experiment 2: 70 ms
- Experiment 3: 140 ms

Observe that the average query response time for Experiment 3 is higher than Experiment 2 even though in Experiment 3 the entire index is hosted in main memory. The most probable reason for

this outcome is that Experiment 3 does not use any cache (generally LRU) for enhancing performance.

2.2.1. Indexing Time

Indexing or insertion time is the time to index an individual entity that has been inserted into the repository. The indexing technology employed is incremental, i.e. the index is updated and not completely re-created, when new entities are inserted into the repository. Indexing time as a performance parameter, thus, becomes most important, if larger collections of entities are imported at one time.

Target Performance

In defining the target performance for the indexing time the following aspects have been used as a basis: the current indexing and the goal of being able to insert 1 Mio of entities in a couple of hours. The target performance for the indexing time is to be able to index an entity in 20 Milliseconds. This is sufficient to index 1 Million entities in 5-6 hours.

The second target for the indexing is that the indexing time for individual entities should not grow with the size of the index or the number of entities in the repository.

Past and Current Measurements

Current measurements focus on the scalability of the indexing time in the size of the repository and are thus covered below. They also show that the insertion time target performance for the insertion of an entity as discussed above is met by the employed technology.

Scalability Considerations

The second target defined above for indexing directly refers to scalability with respect to index and repository size. Initial experiments show that the indexing time actually does not grow in size of index and number of entities. To verify this we used the distributed OKKAMstore setup as explained in Section 2.2.1 (see Figure 5) and detailed below in the context of the response time measurements.

The figure below shows the entity insertion time in the distributed setup, which includes parsing the entity XML files, indexing (Solr) and inserting entities in the entity storage (HBase). The plot shows that an entity can be added to the system in less than 10 ms which we consider very fast. Most importantly this time remains nearly constant during the whole experiment in which 3.3M entities were inserted. The legend below the plot shows the entity collections that were inserted, which also explains some fluctuations in the insertion times.

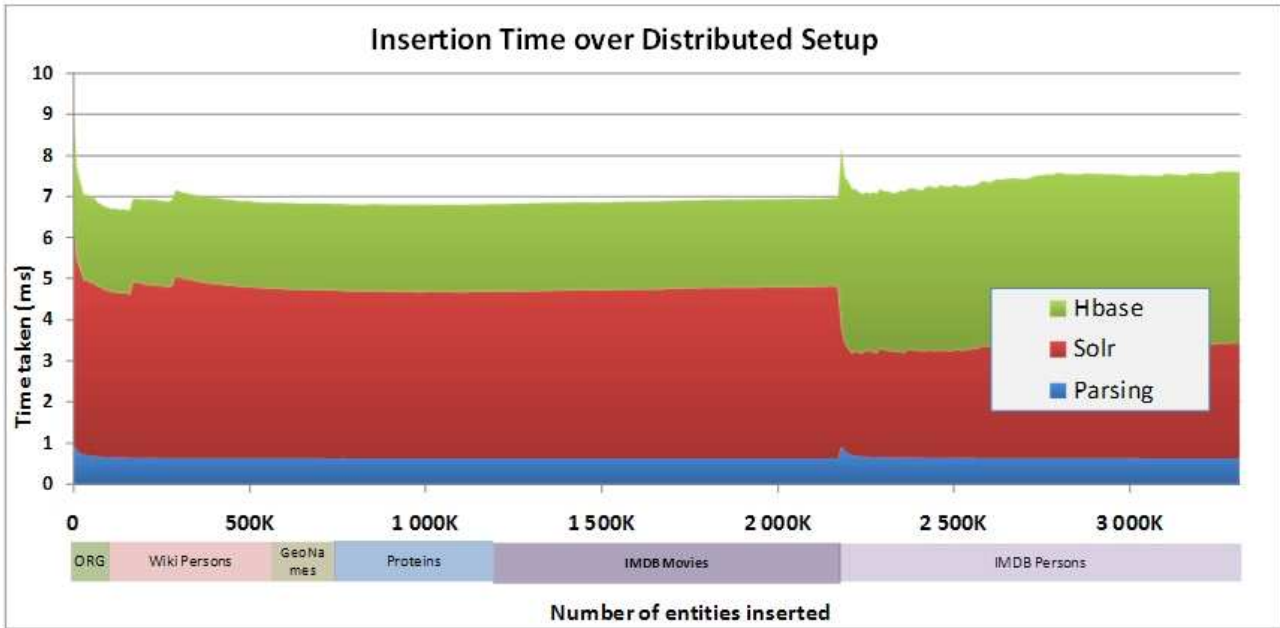


Figure 10: Development of Indexing or Insertion Time with growing number of Entities in the Repository

For the scalability experiments it was decided to use more entities than we have currently in the entity repository (namely 3M instead of 1M). For this purpose the repository was only for experimenting extended with the IMDB data set, for doing the experiment on real data not on artificially created one. The IMDB data set is not part of the public OKKAM nodes since it was so far not possible to sort out the related copyright issues.

2.2.2. Response Quality

The matching or result quality result is analysed by considering the result in the first k positions of the ranked list of results returned. A request is considered answered successfully if:

- either no results are returned and the entity intended by the request is actually not in the repository (true negatives),
- or, in case the entity intended by the request is in the repository, the ID of this entity is returned as one of the first k results in the ranked result list (true positive).

All other cases are considered unsuccessful. Based on this understanding of success, $\text{succes@top } k$ is used as a parameter for assessing the result quality for different values for k.

In more detail, the result of posing a request at OKKAM can have one of the following statuses (i.e., boolean values):

1. True_Positive is set to true when the requested entity was found.
2. True_Negative is set to true when the requested entity is not contained in OKKAM and our request for this entity correctly did not return any results.
3. False_Positive is set to true when the requested entity is not contained in OKKAM but our request falsely returned other entities

4. False_Negative is set to true when the requested entity was not found although it is contained in OKKAM.

Top-K means that instead of using of the entities returned by OKKAM, we only consider the first K entities. For example, Top-10 would mean that we consider the entity found only if it's inside the first ten entities returned, and not found when it's contained or not among the rest of the entities. We define success rate as:

$$\text{Success Rate @ top K} = \Sigma(\text{true_positives} + \text{true_negatives})/\text{total number of queries}$$

With respect to result quality, avoiding of false positives and false negatives has to be balanced in a similar way as one can favour precision or recall in tuning a search engine. A false positive in the OKKAM context would mean that an ID is returned with a high matching probability, although the entity intended in the request is not in the repository at all. The potential consequence of returning false positives is that - in an integration activity based on OKKAM IDs - information about different entities is merged into a single entity. On the other hand, false negatives mean that no ID is returned although the entity intended by the request actually is in the repository. As a consequence a new OKKAM ID is created unnecessarily (or the entity stays un-OKKAMized depending on the strategy of the respective application). As a consequence a) the quality of the repository is reduced (creation of duplicates) and the b) some integration potential might be missed. It is important to keep both the numbers of false positives and of false negatives low. However, it is understood that avoiding false positives is more important than avoiding false negatives, since the impact of false negatives can be reduced by repository purging mechanism, while the impact of too many false positives might seriously reduce the satisfaction of users of the ENS.

Target Performance

The definition of the target performance for the Matching quality is a complex task. According to current measurements the matching quality is rather high. However, there are different factors that might impact result quality:

- It is understood that due to the repository population process the repository (bulk import of existing collections) currently is relatively “clean” with respect to quality and heterogeneity; it is expected that once used on broader basis a decrease of quality along this lines can be expected.
- Evaluations are performed with a generated query set that reflects the expected query mix (currently it is expect that most queries come from automatic extractors); this is done, since so far neither major query logs from the daily use of Entity Name Server nor information on a typical query mix for such systems exist; it is expected that the real query mix will influence result quality. Of course real request logs will be used as soon as they become available – the logging infrastructure is already in place.
- Currently the repository is still relatively small. Clearly, result quality will also be affected by the repository size, since this will raise the probability of similarities between entity descriptions (see also Scalability Considerations).

Based on this consideration, it has been decided that a target performance of 95% success @ top 5 is ambitious but reachable in the OKKAM project. Due to the factors discussed above it might become necessary to correct this number in either direction, once more realistic query logs and repository population are available.

Past and Current Measurements

Various experiments have been performed to assess the quality of the developed matching technologies. The assessment of matching quality is integral part of the development and refinement of the matching technology. In the following the results of more controlled experiments for assessing the current matching quality achieved are described:

October 2008 Experiments

In September and October 2008 a first set of systematic experiments on matching quality has been performed. These experiments cover both the storage and the matching layer, since the matching results are created in a close interaction between both layers.

Experimental Setup

These first experiments have been performed with a small repository and relatively clean data. They have been based on repository population extracted from Wikipedia consisting of about 500.000 entities.

The set of queries used for evaluation has been created by extraction of entity information from randomly chosen Web pages on persons and organizations. For this purpose Expert System information extraction technology has been used to be close to the entity information to be expected from automatic extraction processes. This extracted entity information has been transformed into 5 query variants reflecting different amounts of available information (e.g. with respect to structure). Artificially, errors have been artificially introduced in some of the queries for getting “fuzzy queries” (simulating spelling and extraction mistakes). For evaluation, ground truth wit respect to matching of entities has been manually created.

Results

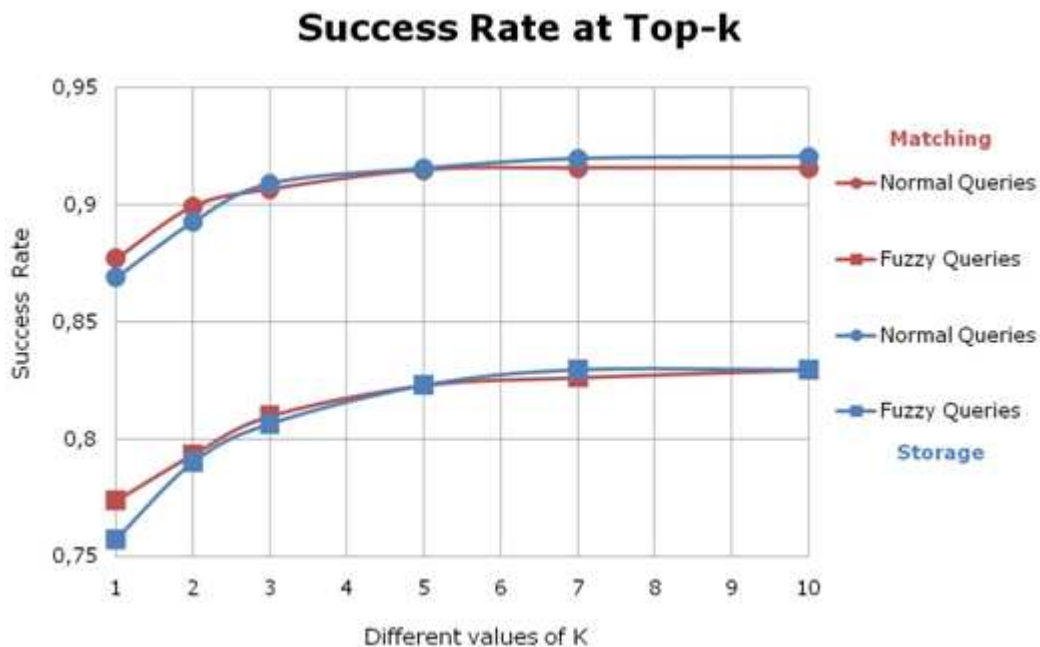


Figure 11: Response Quality in October 2008 Experiments

It has to be noted that the results are already quite good as compared to the target performance. However, as already discussed in the context of the target performance it is expected that an

increase of size and heterogeneity of the entity repository population is expected to impose additional challenges on the matching technology.

February 2009 Experiments

General note: A clear finding of the work on the ENS in the past month was that we require a strong interaction between the storage and the matching layer of the infrastructure. This includes effective query rewriting on the matching layer for the storage layer, efforts to push solutions that are successful on the matching layer on the storage layer and the joint development of optimizations with respect to quality. For this purpose the interaction of WP 2 and WP 3 has been further intensified. One effect is that WP 2 and WP3 are now jointly working on improving the success rate @ top k, which is reflected by having (in contrast to October 2008) only one graphic showing the success rate of the overall matching approach.

Experimental Setup

For dataset we used the 1.005.004 entities as included in OKKAM at the beginning of February 2009, and 500 requests generated using the cogito extractor by EXP from collected web content. We manually processed each request and identified their OKKAM ids.

We imported these entities in an empty OKKAM Node and at various intervals we evaluated all the requests. At each request execution we measured the success rate at different Top-K. The following plot shows the Success Rate for Top-2, 3, 5, 7, and 10 at various numbers of entities in OKKAM.

Results

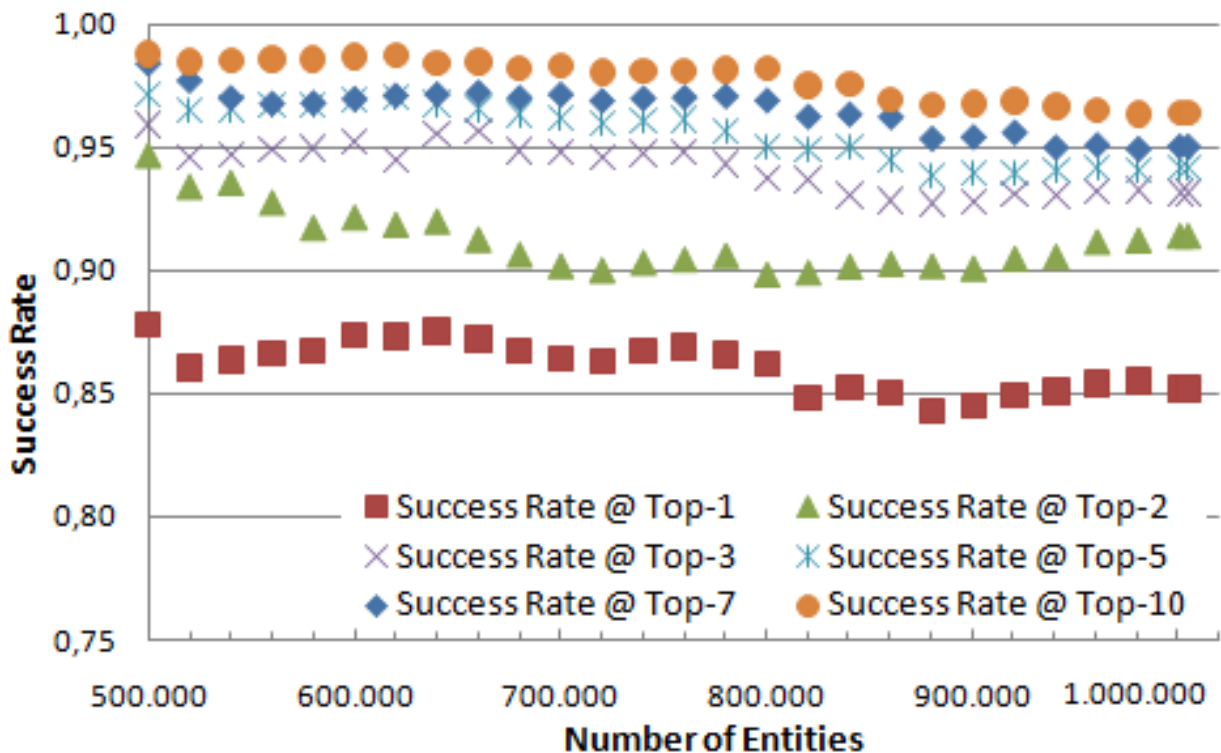


Figure 12: Result Quality measured as success rate @ top-K for various Ks and Repository sizes

As the shown by the plot, success rates for the different Ks are mainly independent of the number of entities. The following table shows the exact success rate for Top-5 and Top-10 for a population of about 1 Mio. entities.

OKKAM with ~ 1.000.000 Entities

- Success Rate @ Top-5: ~94%
- Success Rate @ Top-10: ~96%

Scalability Considerations

There clearly is a dependency between result quality and the size of the repository. Clearly disambiguation becomes more difficult, when more entities are in the repository, because the probability of random similarities is growing. Some initial experiments analysing the dependency of result if result quality from repository size have been performed. First results can be seen in the graphic from the February 2009 experiments.

2.2.3. Query Rate

The average query rate is the average number of queries that can be processed per second. It depends on the response time as well as on the system architecture, since the query rate can be increased by distributing the workload between different nodes.

Target Performance

The target performance for the query rate would depend on the current demand for the OKKAM service as well as the size of the repository. Our goal now is to ensure 10-20 queries per second with several servers and several millions of entities. The optimization task in future would be to minimize the hardware costs for a requested query rate. Unfortunately, these costs are hard to envision at the moment as, for example, search engines do not publish the number of servers they use or the size of the document collections they index.

Past and Current Measurements

In the current set up (prototype v.1.1) the query rate is directly determined by the response time, since the 1-node OKKAMstore installation does not do any distribution of workload. Thus the average query rate equals 1/average response time. Here we report initial experimental results for the preliminary version of the distributed OKKAMstore.

Scalability Considerations

We performed various experiments to determine the query rates we can achieve by the current version of the ENS. We varied the parameters of the system configuration, in particular, the number of shards, the number of replicas within a shard, and the location of the index (hard drive vs. main memory).

Recall that query processing is done in 2 steps in our architecture: 1) obtaining entity IDs of the top-k results from the index (Solr), and 2) fetching the entity profiles for the IDs found in the 1st step (HBase). For the query-rate experiments presented below we study only the 1st step (Solr) as being more important for scalability reasons and ignore the 2nd step.

The setup of the experiments is described in details in Section 2.2.1 (Figure 5).

Experiment 1: Query rate with OKKAM queries, index on disk, 2 shards

This experiment was conducted to determine performance of Solr when index is kept on a hard drive. Every Solr machine was rebooted before each experiment to make sure caching does not affect the measurements.

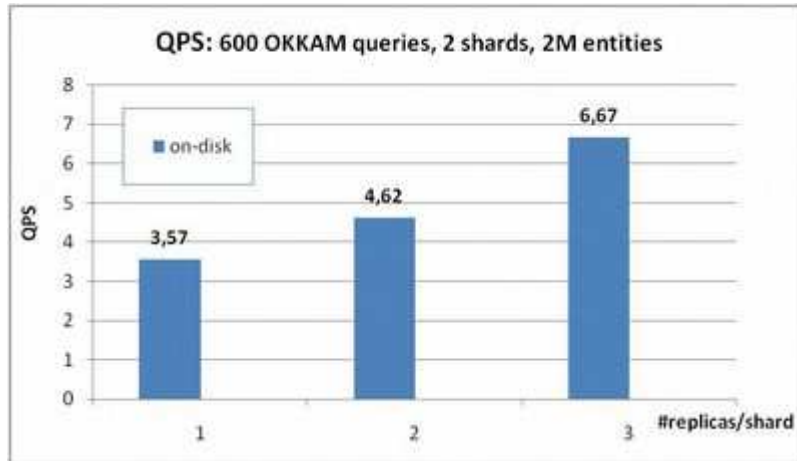


Figure 13: Results on Query Rate from Experiment 1

In the above plot we used two Solr shards (i.e., the index is split in 2 parts) with 1, 2 and 3 replicas per shard. The plot shows that we are able to support more queries per second (QPS) if we increase the number of replicas.

Experiment 2: Query rate with OKKAM queries, index in RAM, 2 shards

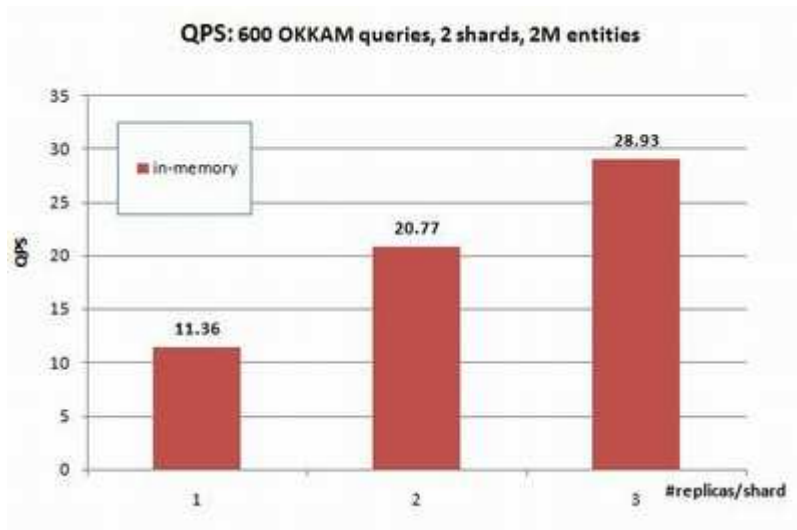


Figure 14: Results on Query Rate from Experiment 2

The figure above shows the query rate with the same setup as in the previous experiment but with one important difference: at each Solr machine index is stored on an in-memory disk partition

(RAM) instead of a hard drive. A similar trend with the QPS increase can be observed, however the absolute values are much higher than in the previous experiment.

Experiment 3: Query rate with OKKAM queries, index on disk, 1-2-3 Shards

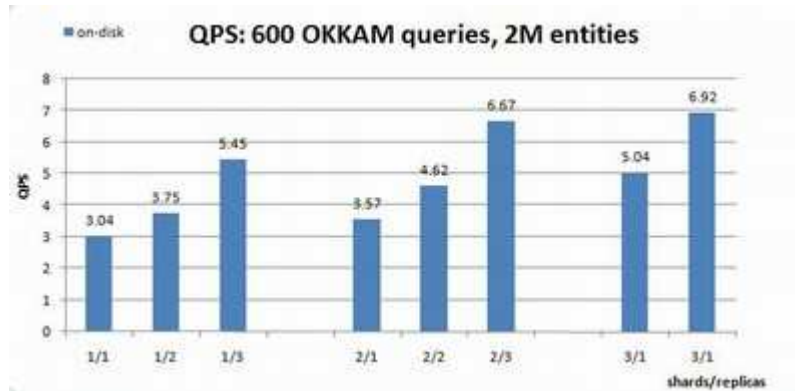


Figure 15: Results on Query Rate from Experiment 3

The above figure is an extended version of Experiment 1. It shows the query rates while varying both: the number of replicas (1, 2 or 3 replicas) and the number of shards (1,2 or 3 shards). In this experiment the index is maintained on disk. It gives a complete picture of the behaviour of the distributed index and shows that adding shards or replicas increases QPS as expected.

Experiment 4: Query rate with OKKAM queries, index in RAM, 1-2-3 Shards

Finally, the last experiment repeats Experiment 3, but with the index kept in RAM. As before, the trend remains the same but the absolute QPS values are much higher compared to corresponding on-disk experiments.

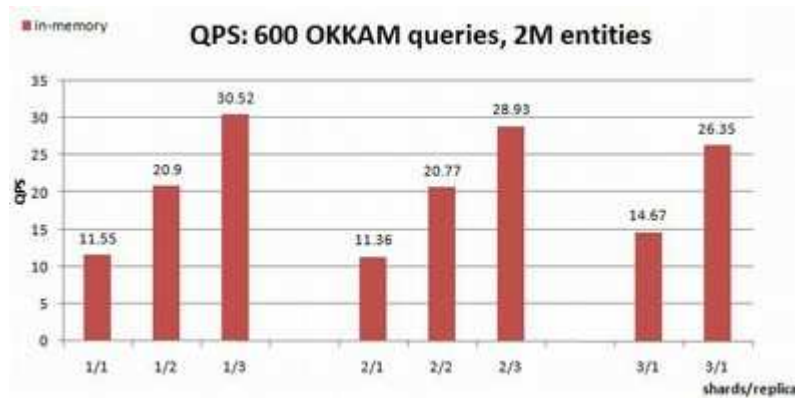


Figure 16: Results on Query Rate from Experiment 4

In general, as we increase number of replicas we get better query rates. Also, when the index is distributed between more shards query processing is faster because the index per shard gets smaller and more queries per second can be processed.

2.3. Performance of Applications

In OKKAM three applications are developed on top of the OKKAM Entity Name Server in order to showcase the benefits of the OKKAM approach as well as to create blueprints for the adoption of the entity-centric approach. The collection of the performance numbers for the three applications has been started with the following results:

- Performance Numbers for the entity-centric semantic Search engine (ECSSE)
- Performance numbers for the application in entity-centric organizational knowledge management
- Performance numbers for the content authoring application

2.3.1. Entity-centric Semantic Search Engine

So far, measurements for the performance of the entity-centric semantic Search engine (ECSSE) have focussed on the size of the indexed collection and the performance and usage of the search engine:

- Number of indexed triples:
 - In September 2008: 428.4 Mio Triples
 - In February 2009: 551,2 Mio. Triples
- Number of unique URIs addressed (resources, entities):
 - In September 2008: 51.3 Mio unique URIs
 - In February 2009: 62 Mio unique URIs
- Unique OKKAM Ids in the collection:
 - In September 2008: 41
- Response time: < 0.1 s
- average number of queries sent to ECSSE by day: ~20,000

Currently, an OKKAMization process for the information collection of ECSSE is underway, which will produce further performance numbers for assessing ECSSE (e.g. number of OKKAMized entities, quality of OKKAMization process, etc.).

2.3.2. Entity-centric Organizational Knowledge Management

This application showcases the use and benefits of the entity-centric approach in the area of organizational knowledge management. The entity-centric approach is applied in some of the knowledge management applications of SAP. This includes the analysis of SAP Forums and the creation of intelligent entity-centric applications on top.

The following performance numbers are currently considered for this application in the OIN activity:

- Numbers on the analysis of the discussion threads taken from SAP forums (SDN)
- Quality of the analysis results

This performance monitoring considers the actual and potential number of discussion threads taken from SAP forums (SDN) on which information extraction and entity recognition are to be performed as well as the quality of the respective extraction. The threads in the discussions forums are a rich source of knowledge and entity-centric information. A description of the targeted applications can be found in D1.2.

Target Performance:

The definition of a target performance within this application is slit into two perspectives:

- **business:** time for task of expert answering queries from community can be reduced from average 3 weeks to 3 seconds by application build on top of OKKAM
- **technical:** analyse 300.000 threads with high quality results

Past and Current Measurements

In September 2008 the following numbers have been measured for this performance parameter:

- 60.000 forum posts and their titles + SAP Terminology
- 36.951 direct component matches (~ in every 3rd post)
- 6.886.145 matches of related terms (57,4 matches per forum entry)
- one term points at max. to 459 components
- 6.762.484 ambiguous matches (in average 56.4 matches per forum entry)

In February 2009 the following numbers have been measured for this performance parameter. A new data set on thread level has been used for these experiments. It contains about 26.000 threads (12.7 GB database). It consists of 90.000 sentences and 450.000 noun groups. The following elements have been extracted:

- ~ 18 mil. direct matches with SAP terminology
- ~ 15. mil. entities SAP terminology identified indirectly
- ~ 42.000 links to external and internal resources identified
- ~ 3500 references to sap notes detected
- ~ 1500 Java Exceptions extracted
- ~ 305 ABAP Errors detected

For extraction quality, the following results have been achieved by the OKKAMIZER for SAP Components. In the table below they are compared with the qualitative results of applying pure Dictionary Matching:

	OKKAMizer	Dictionary Mapping
Precision	89,4%	11%
Recall	84%	2%
F-Measure:	86,6%	3,4%

The future improvements will involve the usage of a larger dataset of forum posts. The number of spotted components is rather low and can be increased by applying relationships from terms to components. Problematic is the ambiguity of those. Applying measurements for confidence and informativeness of matches will increase precision. Besides this, future investigation will be on enhancement of data sources applicable to posts to spot more entity-types.

2.3.3. Entity-centric Content Authoring Application

The applications for entity-centric content authoring are currently closely related to the efforts in OKKAMizations and the OKKAM empowered tools, more specifically the OKKAM Word Plugin. Thus the performance here heavily depends on the OKKAMization performance, which in turn as already discussed in the context of the OKKAMization performance, depends on the performance of entity recognition and of matching. Current numbers for the Performance of matching can be found at the section on Response Quality.

Use Case Potential

Elsevier: ScienceDirect's full-text collection (<http://www.info.sciencedirect.com/content/>) covers more than 2,500 journals and almost nine million full-text articles

ANSA General news includes more than 450 - 500 daily news about worldwide politics, economy, arts/shows and sports + 40 years of 1000 News a day

OKKAMization Performance Requirements

For aligning the existing performance of OKKAMization with the requirements of the entity-centric content authoring the following target numbers have been collected for the envisioned applications.

Entity Detection

- Target precision: >85% of entities in articles and news
- Target recall: >80% of entities in articles and news

Entity Matching

- Target precision: >80% of entities in articles and news
- Target recall: >90% of entities in articles and news

2.4. Performance of OKKAMization

OKKAMization is the process of analyzing existing (structured and unstructured) content and content under creation, identifying entity references in this content (entity recognition) and equipping the entity references with the associated OKKAM IDs. For assessing the work in the area of OKKAMization two aspects are currently considered in the OIN activity:

- Quality of the OKKAMization process
- Creation of a first set of functional OKKAM empowered tools (numbers of tools and coverage)

2.4.1. OKKAMization Quality

Accessing the ENS matching functionality is part of the OKKAMization process as well as analysing the respective type of content for entity references (see process description above). The quality of the OKKAMization of the different OKKAM-empowered tools and OKKAMizers, thus, depends on the quality of the entity recognition, which is most challenging for unstructured content. Furthermore, it depends on the quality of entity matching for the type of entity requests created by the respective OKKAMizers.

The employed Named entity recognition technology developed by Expert Systems, which is in the core of entity recognition of OKKAMization support for unstructured sources, has been tested and evaluated during the first year of OKKAM. This has been done on different corpora, coming from different sources such as Wikipedia, ANSA news, web sources and other Expert System-owned corpora. The corpora languages are Italian and English.

The results of the performance analysis for this technology in the context of the OKKAMization process are summarized in the following table. Similar to an information retrieval scenario, extraction quality is measured in terms of precision and recall. Precision is defined as the number of relevant entities recognized divided by the total number of entities recognized. Recall is defined as the number of relevant entities recognized divided by the total number of existing relevant entities.

Entity Type	Precision	Recall
Person	95.41%	86.00%
Organization	81.48%	74.68%
Place	75.93%	87.25%

In summary good extraction quality has been achieved for the core entities persons, organizations and places across different types of analyzed sources.

In addition, on a defined English corpus of documents, a comparison of different extraction technologies between Expert System technology and other open source tools has been performed. This analysis shows that the Expert System entity extraction system gets better results with respect to the the total number of entities recognized. For example, Expert System technology extracted 12% more entities compared to the frequently-cited OpenCalais technology.

Numbers for the quality of matching performance can be found in the section on the performance of the ENS infrastructure.

2.4.2. OKKAM empowered tool set

The first set of OKKAM empowered has been created on schedule and consists of 6 (7¹) OKKAM empowered tools:

¹ The Neon Toolkit is not yet officially a member of the OKKAM empowered tool set, but this will be the case in the near future, when it has been completely tested.

- Protégé
- foat-O-matic 2.0
- Firefox Plug In
- Internet Explorer Plug In
- Microsoft Word Plug In
- Microsoft Outlook Plug In
- (Neon Toolkit Plug In)

This can be considered as a quite impressive set of Plug ins.

3. Numbers on Involvement

This area covers numbers related to the involvement and collaboration of the partners in the project. Clearly a sufficient involvement of partners into the project and smooth collaboration between the partners is crucial for project progress and for reaching synergies between the partners. Involvement is not restricted to the number of persons working in the project. It also includes sharing the vision of the project, readiness for knowledge sharing and collaboration, openness for the ideas of other partners and adequate communication. Not all of these aspects can be easily and directly measured. However, we collected a number of parameters, which we think are good indicators for the involvement of the consortium members into the OKKAM project.

In more detail the following numbers for involvement are currently considered:

- Established Communication Channels
- Project meetings and Telcos
- Use of Intranet
- Use of Mailing Lists in OKKAM
- Numbers on OKKAM Academy
- Collaborative Software Development

In summary, involvement of the project partner into the project is good to very good. A friendly and excited mood about the project has already been established in the Kickoff meeting, which played an excellent role in establishing the identity of the project and of the consortium as the OKKAM project team. Considerable effort has been invested by the OKKAM management to keep this spirit up and to transform it into concrete and engaged involvement into project work. This also included regular (self-) assessment by the project management and the definition of corrective actions such as the OKKAM academy.

3.1. Established Communication Channels

Within OKKAM a number of collaboration and communication channels have been established driven by the needs of the project. These channels are an important means to coordinate the work between different partners – an important building block in transforming partner involvement into overall project progress. The collaboration and communication channels include the setup of a Wiki-style content management system as an collaboratively edited Intranet, the creation of mailing lists for relevant sub-groups within the project (e.g. WPs), as well as the establishment of an infrastructure for telephone conferences and virtual meetings.

Monitoring communication and collaboration (and taking corrective actions, where required) helps for identifying communication gaps. This monitoring is also important for keeping the risk low for misunderstandings, conflicts and double work due to a lack of understanding of other partner's work.

3.1.1. Project meetings

In the first year of a European project, personal information exchange plays a central role for creating an improved shared understanding of the project and for establishing the social

relationships that are required to subsequently collaborate remotely. Clearly, face-to-face meetings are important for achieving these goals. However, they are clearly also quite expensive and have to be used with care, in order to ensure effective use of resources.

In more detail, the following numbers have been collected for monitoring the meeting practice in the OKKAM project.

- Until October 2008, 44 meetings haven been held in OKKAM.
- 29 of these have been organized as virtual meetings, 15 as physical face-to-face (f2f) meetings.
- Of the 15 f2f meetings, 8 meetings have been cross WP meetings
- An average of 7.8 participants took part in the f2f meetings
- An average of 3.9 organization where involved into the f2f meetings

In addition, there were two large meetings that involved persons from all members of the consortium: The Kickoff meeting in January 2008 and a plenary meeting in August 2008.

Assessing these numbers a good mix between f2f and virtual meetings has been established. Virtual meetings actually proved very useful, if very specific issues had to be resolved in a meeting (e.g. integration sessions). It is not the goal of the OKKAM management to further reduce the number of f2f meetings. They play an irreplaceable role in discussing ideas, in fostering collaboration and project involvement and in resolving misunderstandings. Good use of the f2f meetings has been made of the meetings by heavily using them to resolve cross-WP issues.

3.1.2. Telephone Conferences

Until October 2008 Research and development issues where discussed in about 64 telcos. In addition, there were regular status and management telcos.

3.1.3. Use of Intranet

For information exchange, collaborative development of ideas and concept and for communication a content management system with a Wiki-like front end has been established in the project. The system is used as an intranet, but also has flexible options to dynamically publish content to the public OKKAM Web pages as well as to make it visible to the European Commission.

Within the project the Intranet was well adopted and actively used. This surely is partly due to the acquaintance of many of the partners with the use of Wiki system as Intranet technology. However, it is also a sign of involvement of the partners into the OKKAM project that many pages, events, news and other types of content was created and that a stable growth of the number of pages can be observed, since the Intranet has been established.

In more detail, until October 2008 the following items have been added to the Intranet:

- 54 announcements of external and internal events
- 41 news items
- 55 folders for structuring the information

The following graphic reflects the stable growth of content in the Intranet from May to September 2008. It shows the size of the backup for the Intranet.

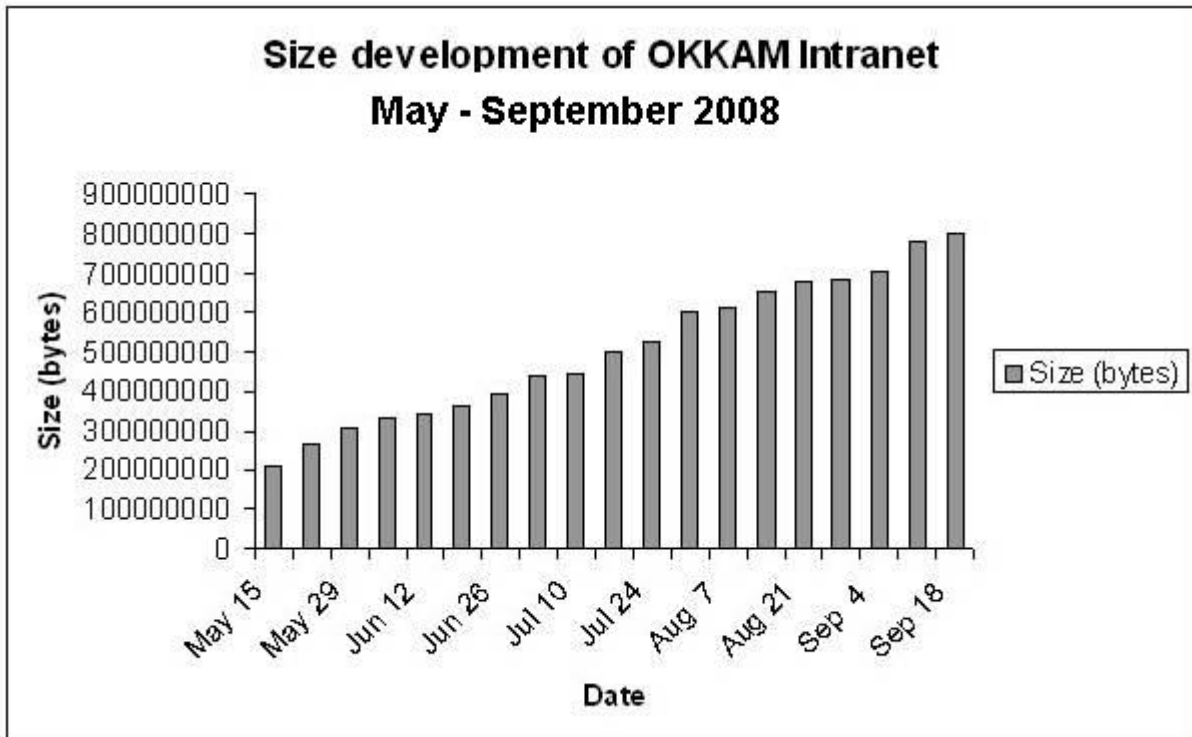


Figure 17: Development of Intranet Size (reflected by Backup size)

3.1.4. Use of Mailing Lists in OKKAM

An initial set of mailing lists has been established in the first quarter of the project based on an analysis of the project structuring (e.g. mailing lists for each WP, PCA, and organization). Later further mailing lists have been added according to the needs identified in the project (such as okkam-all). Overall, 43 mailing lists have been created. The average number of members in the mailing lists is 9.8. The figure below shows the distribution of mailing list members in more detail.

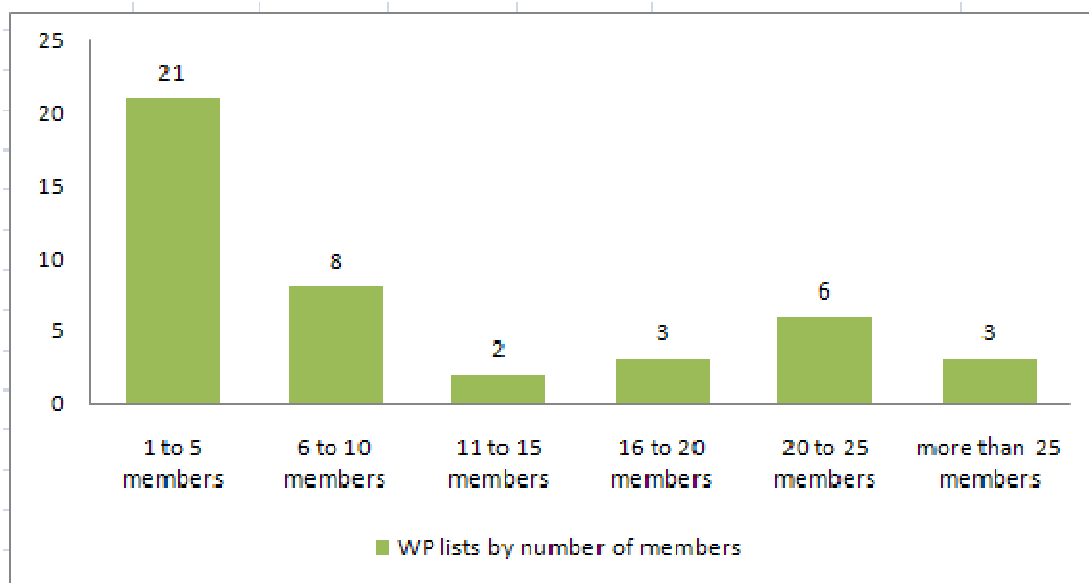


Figure 18: Number of Mailing List Members

Only about 15 of the established mailing lists are frequently used. This reflects the communication needs and practices in the project as well as the way the work is organized e.g. a strong organization of the work along WP structures, which did lead to a good use of the WP mailing lists.

The following figures show the use of the mailing lists in the OKKAM project in the time between mid March (when they were set up) until October 2008. From the established mailing lists the WP mailing lists for WP1 to WP7 have been selected for monitoring. All of these technical WPs are related to the creation of the OKKAM infrastructure, which has been a focus of the first phase of the project. Furthermore, these WPs also have a high need of interaction within the individual WP as well between the WP to ensure alignment and preparation of integration for the components developed in the individual WPs.

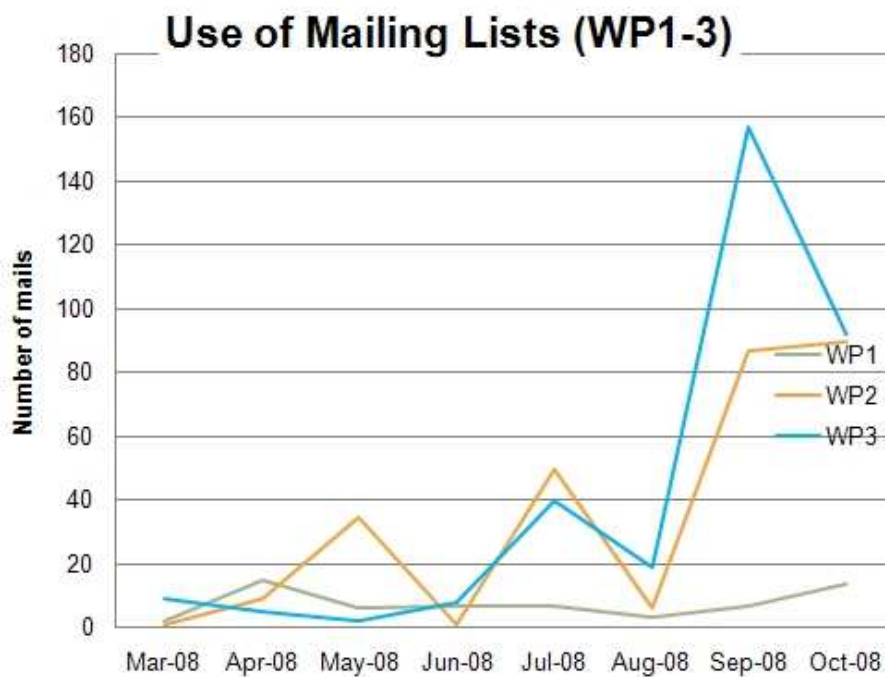


Figure 19: Use of Mailing Lists for WP1, WP2 and WP3 (March – October 2008)

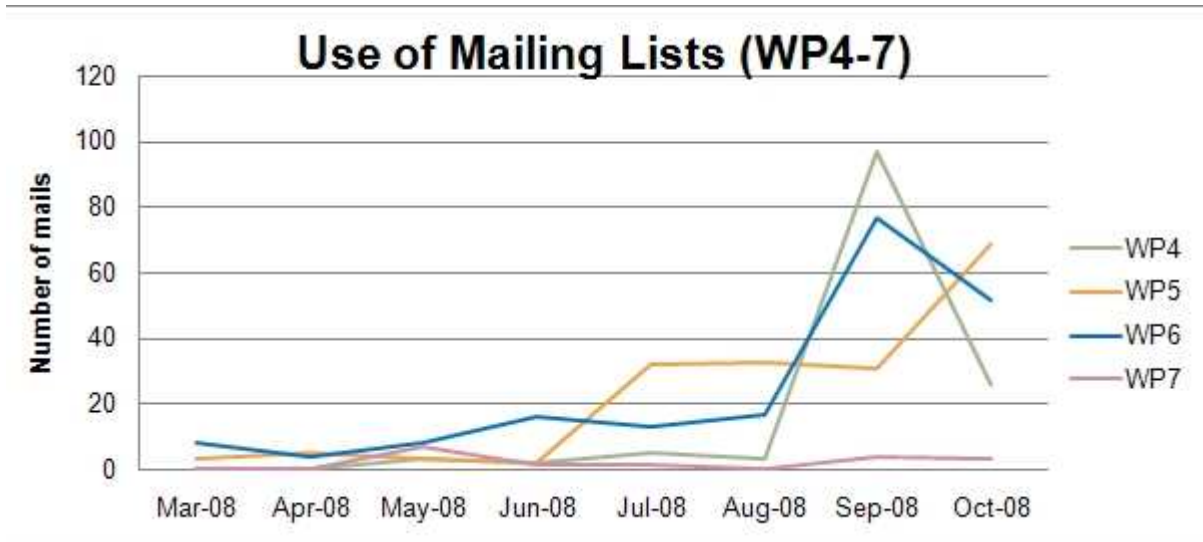


Figure 20: Use of Mailing Lists for WP4, WP5, WP6 and WP7 (March – October 2008)

In general, an increase of mailing list use can be seen. Additionally, there are clear peaks in the mailing lists activities at special events e.g. when the main work on the integration of ENS V1 was done in September.

For the OKKAM management (including the respective WP leaders) an analysis of the mailing list activities can help in identifying communication gaps. Longer breaks with a very low number of mails might point to a lack of communication in the WP and should be addressed. However, it also has to be considered that other communication channels might be used alternatively.

3.2. Numbers on OKKAM Academy

So far 3 OKKAM Academies have been organized for project internal knowledge exchange with an average of about 10 participants.

3.3. Involvement in Collaborative Software Development

Involvement in collaborative software development is split up into involvement into the iterative requirement analysis process as well as into the software development itself.

3.3.1. Requirements Collection

An initial set of requirements for the ENS and the applications has been collected in the first month of the project. The process has been based on the identification of major use cases (see D1.1 for details). An important outcome of this activity is a set of requirements, open issues within individual WPs as well as open interaction issues between WPs. The OKKAM team has collected:

- 88 major requirements
- 70 open issues
- 43 open interaction issues

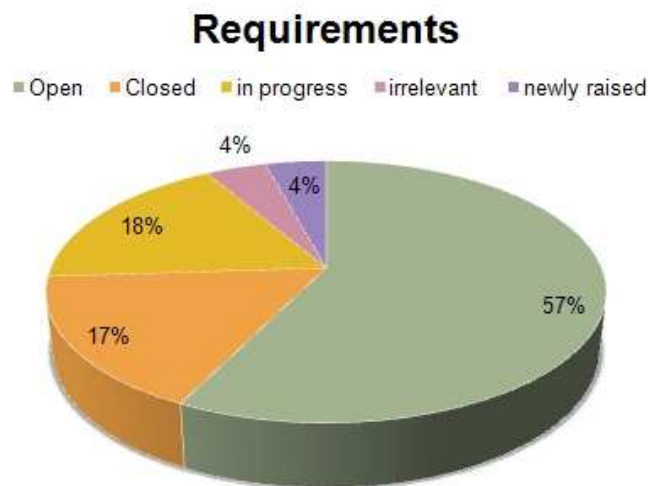
The requirements are documented in the project deliverable D1.1.

Following an iterative approach, the requirements have been revisited in month ten of the project for checking the status of the identified issue and for documenting new issues that have been identified in the course of the project. This process will be regularly repeated and the results are documented in a requirements working document in the OKKAM Wiki as well as from a quantitative point of view in the OIN report. The quantitative results of the first iteration of requirement analysis are reported here.

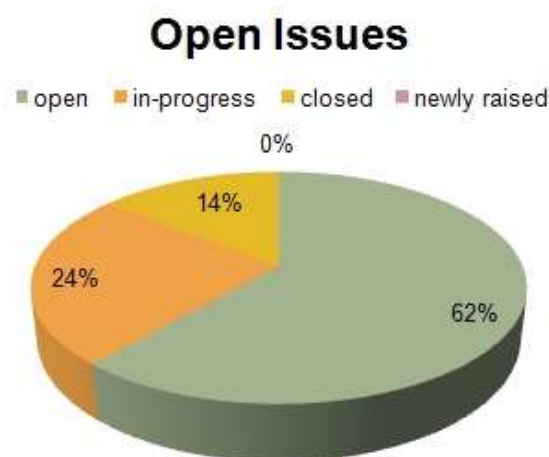
In month 10 of the project the requirements and issues have been revisited, in order to verify them against the current status of the project. This has been done as part of an agreed upon regular requirement revisiting process. The results of this process is documented in an OKKAM Wiki page, which acts as an D1.1 follow up working point.

From the quantitative side the requirements revisiting process in month 10 produced the following results:

- 92 requirements of which:

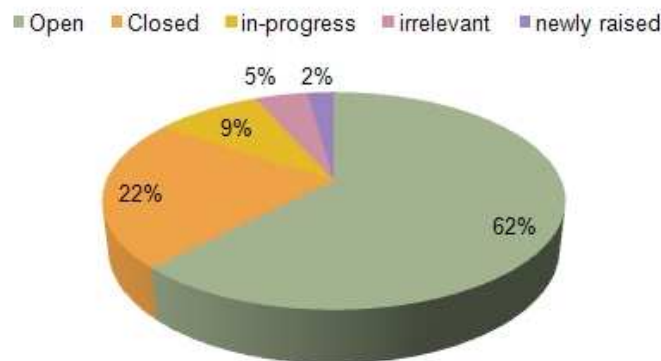


- 70 open issues of which:



- 44 interaction issues of which:

Interaction Issues



3.3.2. Collaborative Software Development

For coordinating the collaborative software development a gForge environment has been set up and is used by the OKKAM distributed development team. The environment is actively used for managing releases, for exchanging and consolidating code versions and for managing the reporting and fixing of bugs (ticketing system). The following numbers reflect the level of usage of the gForge platform:

- In October 2008: 36 active developers in 17 active gForge OKKAM projects
- In February 2009: 45 active developers in 24 active gForge OKKAM projects

As a further indicator for involvement and activities the amount of code for ENS produced in the most active projects is used (lines of code, numbers of methods, numbers of packages, and numbers of classes). The following table reflects the status of code development in October 2008:

Component	Lines of code	Number of packages	Number of classes	Number of methods
OKKAM Match	6715	14	70	432
OKKAM Store	640	8	13	49
OKKAM Core	2349	16	56	243
OKKAM Lifecycle	431	2	8	45
Total	10135	40	147	769

4. Numbers on Impact

For achieving its impact it is very important for the OKKAM project to achieve good visibility for the OKKAM project. The consortium and especially the coordinator have been very active in this area. Relevant numbers for assessing the impact-related activities are numbers on OKKAM related dissemination activities, the use of the OKKAM public Web site and the number of scientific publications on OKKAM related topics.

4.1. Dissemination Activities

Up to October 2008, OKKAM has been involved in 15 number of dissemination events (including ESWC2008 sponsorships). With these events an estimated audience of 500-600 persons has been reached. Furthermore, 5 News articles have already reported on OKKAM.

Further dissemination activities are planned, especially the Web 3.0 Academy on February 18, 2009. [Update: This has meanwhile been successfully organized]

4.2. Use of Public OKKAM Web Page

The following numbers from the statistics on the public part of the Web Page reflect the use the public Web page:

- Status October 2008 (covering Feb - Oct 2008)::
 - 4.737 absolute unique visitors
 - 29.258 page views
 - 9.099 visits (average 38,72 per day)
 - 798 visits to the page of OKKAM publications
 - 663 visits to the page of deliverables
 - 88 countries
 - 48,49 % returning visitors
- Status December 2008 (covering Feb - Dec 2008):
 - 7.133 absolute unique visitors
 - 39.847 page views
 - 13.299 visits (average 41.05 per day)
 - 1070 visits to the page of OKKAM publications
 - 953 visits to the page of deliverables
 - 542 visits to the TestTubes
 - 100 countries
 - 46,36% returning visitors

4.3. Number of Publications

List of OKKAM-related papers published to date.

1. José Júlio Alferes, Ricardo Amador, Philipp Kärger, and Daniel Olmedilla. Towards reactive semantic web policies: Advanced agent control for the semantic web. In *Poster and Demo Session of the 7th International Semantic Web Conference (ISWC 2008), Karlsruhe, Germany*. CEUR Workshop Proceedings, October 2008.
2. Barbara Bazzanella, Paolo Bouquet, and Heiko Stoermer. A Cognitive Contribution to Entity Representation and Matching. Technical Report DISI-09-004, Ingegneria e Scienza dell'Informazione, University of Trento., 2009. <http://eprints.biblio.unitn.it/archive/00001540/>.
3. Barbara Bazzanella, Junaid A. Chaudhry, Themis Palpanas, and Heiko Stoermer. Towards a General Entity Representation Model. In *Proceedings of the 5th Workshop on Semantic Web Applications and Perspectives (SWAP2008)*, Rome, Italy, December 2008. online http://ceur-ws.org/Vol-426/swap2008_submission_57.pdf
4. Barbara Bazzanella, Junaid Ahsenali Chaudhry, Themis Palpanas, and Heiko Stoermer. Towards a general entity representation model. In *SWAP*, 2008.
5. Paolo Bouquet and Heiko Stoermer. OKKAM: Enabling an Entity Name System for the Semantic Web. In *Proceedings of the I'ESA2008 Workshop on Semantic Interoperability*, 2008.
6. Paolo Bouquet, Heiko Stoermer, Claudia Niederee, and Antonio Mana. Entity Name System: The Backbone of an Open and Scalable Web of Data. In *Proceedings of the IEEE International Conference on Semantic Computing, ICSC 2008*, number CSS-ICSC 2008-4-28-25 in CSS-ICSC, pages 554-561. IEEE Computer Society, August 2008.
7. Paolo Bouquet, Heiko Stoermer, and Barbara Bazzanella. An Entity Name System (ENS) for the Semantic Web. In *The Semantic Web: Research and Applications. Proceedings of ESWC2008.*, volume Volume 5021/2008 of *Lecture Notes in Computer Science*, pages 258-272. Springer Berlin / Heidelberg, June 2008.
8. Paolo Bouquet, Heiko Stoermer, Daniele Cordioli, and Giovanni Tummarello. An Entity Name System for Linking Semantic Web Data. In *Proceedings of the Linked Data on the Web Workshop*, number 369 in CEUR Workshop Proceedings. CEUR, April 2008. <http://ceur-ws.org/Vol-369/paper23.pdf>
9. Junaid Chaudhry, Themis Palpanas, Periklis Andritsos, and Antonio Mana. Entity lifecycle management for okkam. In *IRSW, Tenerife, Spain*, June 2008.
10. Philippe Cudré-Mauroux, Parisa Haghani, Michael Jost, Karl Aberer, and Hermann de Meer. idMesh: Graph-Based Disambiguation of Linked Data Engines. In *WWW'09: Proceedings of the 18th International World Wide Web conference*, Madrid, Spain, 2009.
11. Gianluca Demartini, Julien Gaugaz, and Wolfgang Nejdl. A vector space model for ranking entities and its application to expert search. In *ECIR*, 2009.
12. Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. A model for ranking entities and its application to wikipedia. In *LA-WEB*, pages 29-38, 2008.
13. Gianluca Demartini and Claudia Niederee. Finding experts on the semantic desktop. In *Personal Identification and Collaborations: Knowledge Mediation and Extraction (PICKME 2008) Workshop at ISWC 2008*, 2008.

14. Gianluca Demartini. Comparing people in the enterprise. In *ICEIS (2)*, pages 455-458, 2008.
15. Gianluca Demartini. How many experts? - a new task for enterprise search evaluation. In *Workshop on Novel Methodologies for Evaluation in Information Retrieval at the 30th European Conference on IR Research*, pages 39-43, 2008.
16. Elena Demidova, Xuan Zhou, Gideon Zenz, and Wolfgang Nejdl. Suits: Faceted user interface for constructing structured queries from keywords. In *DASFAA'09: Proceedings of the 14th International Conference on Database Systems for Advanced Applications (Demo track) Systems for Advanced Applications*, pages 772-775, 2009.
17. Julien Gaugaz and Gianluca Demartini. Entity identifiers for lineage preservation. In *IRSW*, 2008.
18. Philipp Kärger. Advanced semantic web policies: Evolution reactivities, and priorities. In *7th International Semantic Web Conference, Karlsruhe, Germany*, Lecture Notes in Computer Science. Springer, 2008.
19. Sheila Kinsella, Adriana Budura, Gleb Skobeltsyn, Sebastian Michel, John Breslin, and Karl Aberer. From Web 1.0 to Web 2.0 and Back – How did your Grandma Use to Tag? In *Proceedings of the 10th International Workshop on Web Information and Data Management (WIDM'08) collocated with CIKM'2008*, 2008.
20. Themis Palpanas, Junaid Chaudhry, Periklis Andritsos, and Yannis Velegrakis. Entity data management in okkam. In *SWAE, Turin, Italy*, September 2008.
21. Ferry Irawan Tantonono, Nishad Manerikar, and Themis Palpanas. Efficiently discovering recent frequent items in data streams. In *SSDBM, Hong Kong, China*, June 2008.
22. Yannis Velegrakis. On the Importance of Updates in Information Integration and Data Exchange Systems (Keynote talk). In *(VLDB collocated) International Workshop on Databases, Integration Systems and P2P Computing (DBISP2P)*, August 2008.

5. Numbers on Timing

The monitoring of the project progress in terms of timing of intermediate results is crucial for ensuring the overall timeliness of OKKAM results as well as for being able to identify the need for corrective actions in case of delays.

In the OKKAM work plan (as documented in the DoW) a number of different checkpoints have been defined, which support monitoring of timeliness. These are the milestones defined for the project, the deliverables (equipped with deadlines) and the three project Reviews (plus the Pre-Review), which also impose a temporal structure to the project.

5.1. Milestones

Of the 19 milestones defined in the OKKAM project, 6 have been due in the first year of the project:

- MS1 - Deployment strategy (Month 4)
- MS2 - Tool Selection for Tool Suite V1 (Month 6)
- MS3 - Basic OKKAMization and entity import (Month 8)
- MS4 - Basic Entity storage infrastructure (Month 10)
- MS5 - Basic Entity Matching Methods (Month 10)
- MS6 - Entity Population "First Million" (Month 12)

All of these 6 milestones have been achieved in time.

5.2. Deliverables

Within the first year of the project 16 of the overall 40 deliverables have been due. These 16 deliverables include reports, application mock-ups as well as the first version of the ENS infrastructure. Most of the deliverables have been delivered in time. Some of the deliverables have been delayed mostly for a) integrating further ideas that have been raised during the pre-review or b) incorporating ideas on possible quality improvement that have been raised as part of the established quality assessment process.

5.3. Reviews

The Pre-review of the project has been held in month 10 of the project and resulted into rich input for the project. At the time of preparation of this document, the OKKAM team is preparing for the 1st OKKAM Review.

6. Numbers on Resources

A careful monitoring of the resources spent in the project is a crucial task for the OKKAM management and administration. It is required to ensure effective use of money from public sources, a demand-oriented, budget-aware, and goal-driven spending of the resources by all partners and all work packages taking into account forecasts for the complete lifetime of the OKKAM project. It is also necessary to detect deviations from the planning and to discuss such deviations with the respective partners and/or WP leaders.

An overview of the used resources is included into the OIN report, since this is the counterpart of the quantitative and qualitative assessment of the achievements of the project. The presentation is structured in three-monthly periods following the quarterly reporting pattern of the OKKAM project. This OIN report covers the first three quarters of the project. Furthermore, a summary accumulating over the full three first quarters of the project is included.

Use of resources is shown from two perspectives: by partners (reflecting use of resources across the consortium) and by work packages (reflecting the main structuring of work in the project). In each case the resources used are compared to the planned use of resources. Planning numbers per Partner are based on PM involvement of the respective partner according to the DoW divided by the time of partner involvement in the project. Planning numbers per WP are based on PMs assigned to the Work package in the DoW divided by the duration of the WP within the project.

Overall, use of resources is in line with the planning. However, there are some deviations mainly due to the following strategic decisions that have been taken at the start and in the early phase of the project for improving its performance and impact:

- Focus on getting the infrastructure running
- Earlier start of some WPs in the project

Full detail of the use of resources per partner and work package can be found in the Six-Monthly report(s) of the OKKAM project and in the Periodic Progress Report.

6.1. Year 1 Overview

This section gives an overview of the consumed resources in year 1. Details can be found in the section on the individual project quarters.

6.1.1. Consumption of Resources

The graphic below (Figure 21) shows the consumption of resource in terms of budget in the first year of the OKKAM project. Since OKKAM is a 30 month project, the first twelve month constitute about 40% of the runtime of the project. In spite of high activity in the first year of the project - as it can be seen by the consumption of PMs as shown in Figure 22-, just 37% of the overall budget has been spent in the first year of the project. The consortium thus is confident that the goal of the project can be achieved within the limits of the assigned budget.

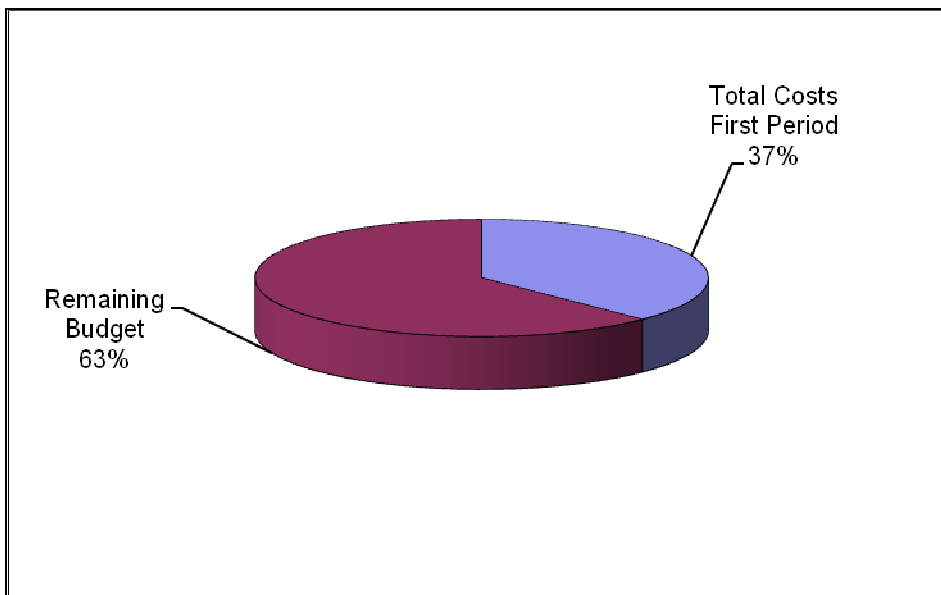


Figure 21: Consumption of Budget in Year 1 as part of the overall available Budget

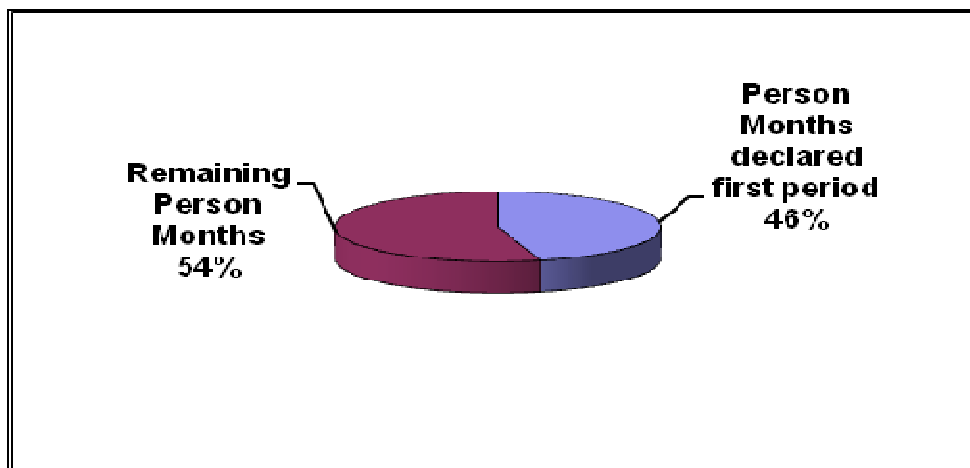


Figure 22: Consumption of PMs on Year 1 as part of the overall planned PMs

6.1.2. Splitting of Costs

The following table shows the breakdown of costs occurred in the first year into different cost categories.

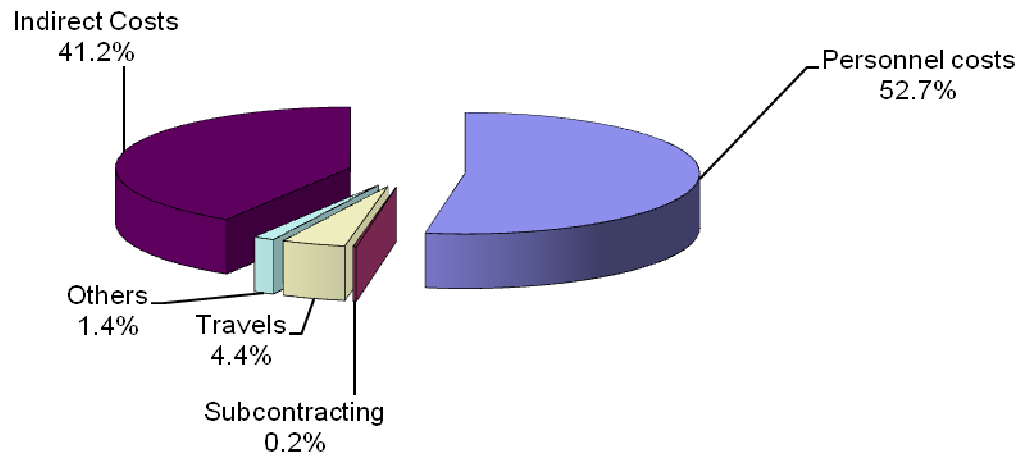


Figure 23: Splitting of the Costs occurred in Year 1

6.2. Numbers from OKKAM Q1

6.2.1. Use of Resources per Partner

The first quarter of the project shows typical start up pattern for some partners of the project (see graphics below), which is implied by establishment of planning and work environment within the first three months of the project. However, it has to be said the project came into action quite quickly and seamlessly, triggered by a very engaging Kick off meeting and partly by previous experience of part of the consortium in working together.

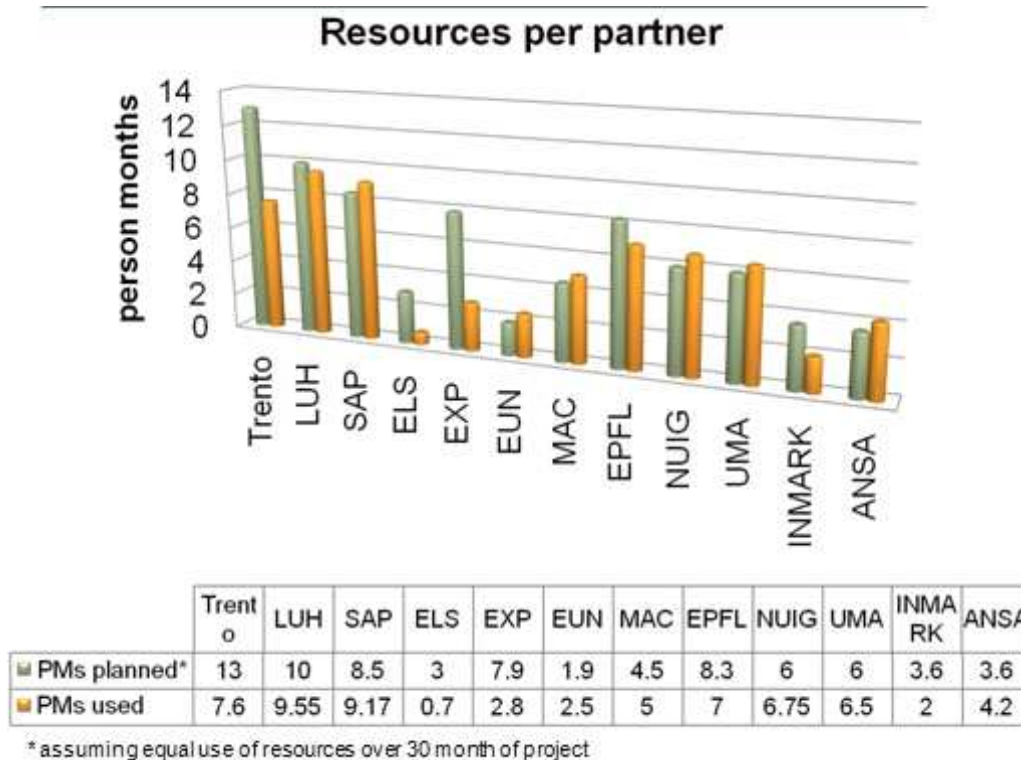
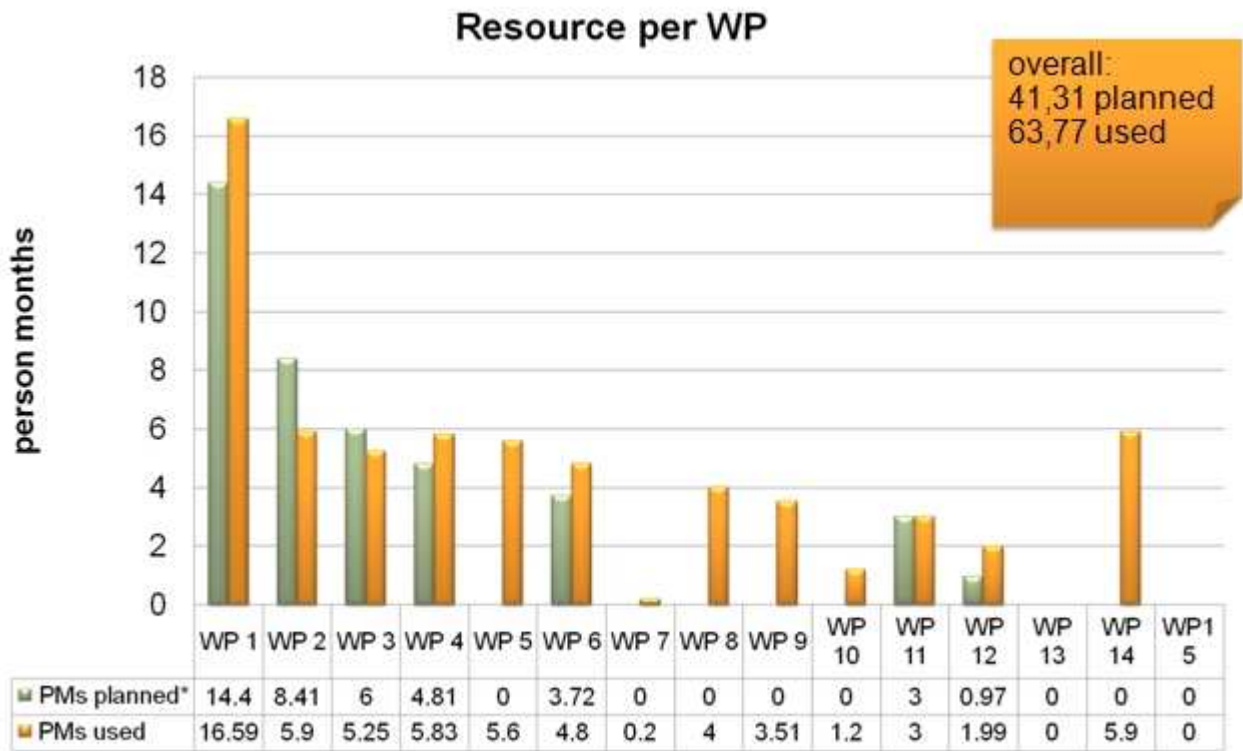


Figure 24: Use of resources per partner in Q1 vs. Planned use of resources

Some of the further deviation from the planning are also triggered by the later start of the work package (according to planning in DoW) where the respective partners are involved (This leads to a difference between planned and used resources, since we assume equal use of resources over the complete project runtime in all cases.). This holds, for example, for Elsevier, where the WP of main involvement – WP 10 – only starts in month 4.

6.2.2. Use of Resource per WP

The diagram below shows the use of resources (in person months) in Q1 per WP of OKKAM. It clearly shows the earlier start of (initial) work the application WPs (WP8, WP9, WP10) as explained in the introduction of this section. In addition, it also shows the earlier start of WP5 (in comparison with plans in DoW), which is implied by the target to get an integrated first version of the ENS running as early as possible. This goal made an early start of WP5 – which is the WP responsible for coordinating the integration – necessary.



* assuming equal use of resources over runtime of WP, note that some WPs were started earlier than planned

Figure 25: Use of resources per WP in Q1 vs. Planned use of resources

In WP 7, WP 13, and WP 15 there were - as planned – no or no major activities in the first quarter of the project. All of these WPs start later in the project. There were some unplanned use of resources in WP14 is due to the fact that we have used some important dissemination and community building opportunities from the beginning of the project.

6.3. Numbers from OKKAM Q2

6.3.1. Use of Resource per WP

For this quarter there are more small-to-medium deviations as compared to the planning. In WP1 less resources were spent in Q2 than planned, compensating for the fact that more PMs have been spent in Q1. This means there was an early start with WP1 with intensive work in WP1 already achieving major parts of the goals of WP1. A similar situation can be seen for the application WPs - WP 8, WP9 and WP10 – though to different degrees.

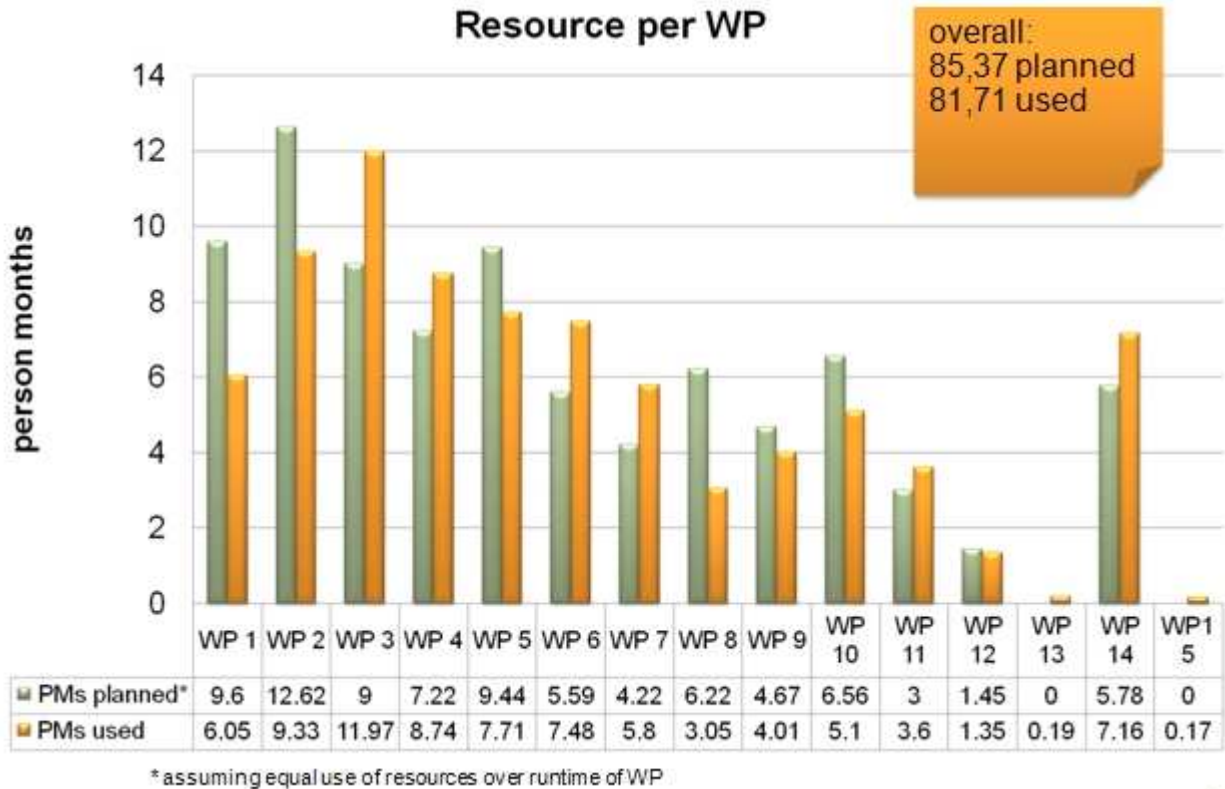
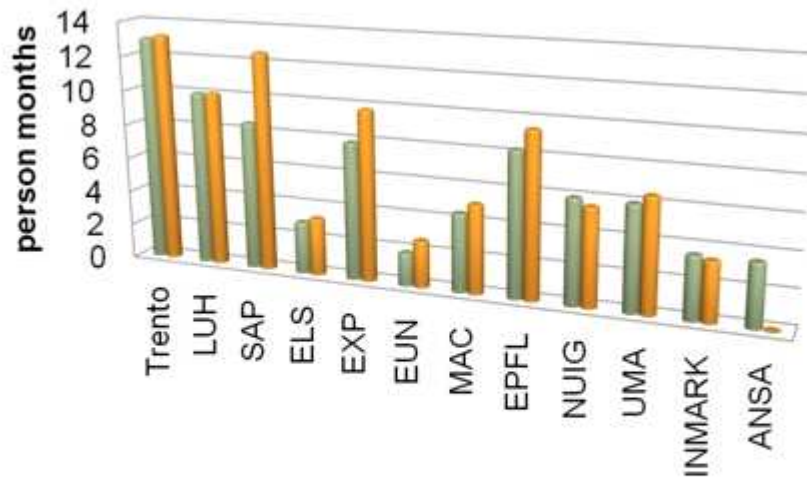


Figure 26: Use of Resources per WP in Q2 vs. Planned Use of Resources

6.3.2. Use of Resources per partner

Resource spending by partner has been much more balanced with respect to planning in the second quarter of the project. As can be seen in the diagram most partners spend resources as planned with minor over- and under-spending compared to planning. Largest deviations to planning are with SAP and ANSA. SAP engaged very heavily in the creation of the application for organizational entity centric knowledge management, in order to get an early prototype started.

Resources per partner



	Trento	LUH	SAP	ELS	EXP	EUN	MAC	EPFL	NUIG	UMA	INMARK	ANSA
PMs planned*	13	10	8.5	3	7.9	1.9	4.5	8.3	6	6	3.6	3.6
PMs used	13.2	10.05	12.57	3.3	9.8	2.7	5.1	9.46	5.6	6.5	3.4	0

* assuming equal use of resources over 30 month of project

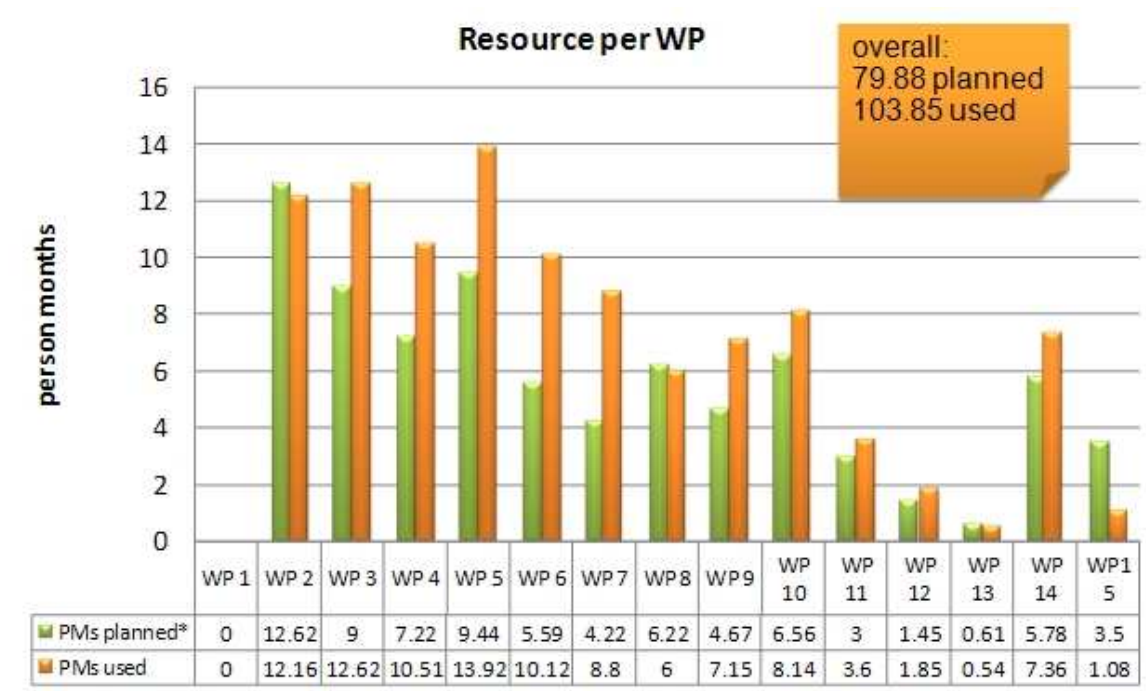
Figure 27: Use of Resources per Partner in Q2 vs. Planned use of resources

Still there have been no major activities in WP13 and WP15, which are planned to be started later in the project.

6.4. Numbers from OKKAM Q3

6.4.1. Use of Resource per WP

The following graphics shows the use of resources in Q3 of the project, i.e. July 2008 - September 2008 per WP. The resources are considered in terms of person months spent in project activities. The actual numbers of resource used are compared with planned resource usage. For the planning numbers, an equal distribution of PMs assigned to the respective WP (as documented in the DoW) over the active time of the respective WP is assumed.



* assuming equal use of resources over runtime of WP

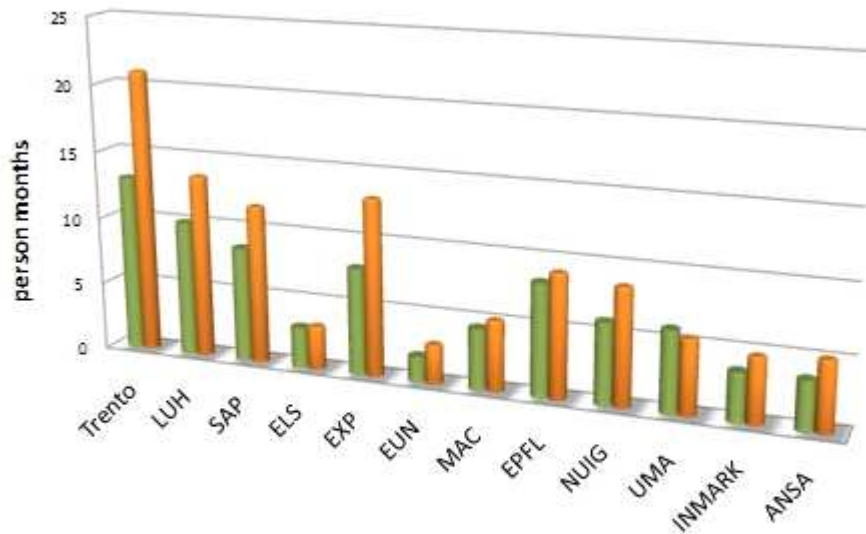
Figure 28: Use of Resources per WP in Q3 vs. Planned Use of Resources

An analysis of the numbers of resources used (as compared to the planned resources) shows high activity in the WPs related to the creation of the first running of the ENS (WP2-WP7). A higher investment of resources has been made here to get the first version of the ENS running as soon as possible, since the consortium has decided that this is a good strategy to get the OKKAM idea established. Furthermore, a higher than planned effort can be seen in WP 14, which deals with Dissemination activities. Here early dissemination activities have been started by the respective partners under the lead of the leader of this WP to start preparing the future of the OKKAM ENS.

6.4.2. Use of Resources per partner

The following graphics shows the use of resources in Q3 of the project, i.e. July 2008 - September 2008 per partner. The resources are considered in terms of person months spent in project activities. The actual numbers of resource used are compared with planned resource usage. For the planning numbers and equal distribution of maximum available PMs (as documented in the DoW) over the 30 months duration of the project is assumed.

Resources per partner



	Trento	LUH	SAP	ELS	EXP	EUN	MAC	EPFL	NUIG	UMA	INMARK	ANSA
PMs planned*	13	10	8.5	3	7.9	1.9	4.5	8.3	6	6	3.6	3.6
PMs used	21.00	13.55	11.71	3.20	13.10	2.90	5.20	9.12	8.62	5.50	4.90	5.15

* assuming equal use of resources over 30 month of project

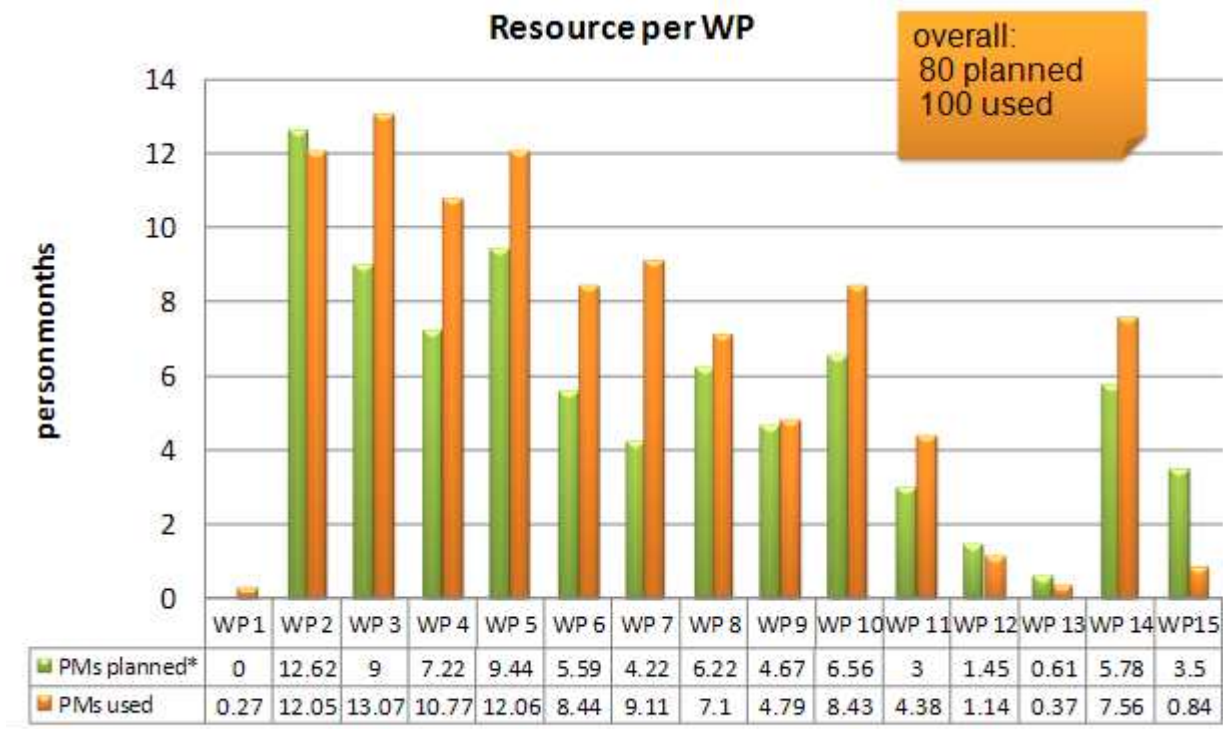
Figure 29: Use of Resources per Partner in Q3 vs. Planned use of Resources

The comparison of the used PMs with the planned PMs shows high activity of the partners in Q3. Especially the partners involved in the development of the ENS and of the applications show a high activity level. This is due to the fact that the plan to complete a first version of the ENS and of the applications for the Pre-Review. This was before the original schedule of these versions and required some additional investment of resources in this phase of the project.

6.5. Numbers from OKKAM Q4

6.5.1. Use of Resource per WP

The following graphics shows the use of resources in Q4 of the project, i.e. October 2008 – December 2008 per WP. The resources are considered in terms of person months spent in project activities. The actual numbers of resource used are compared with planned resource usage. For the planning numbers, an equal distribution of PMs assigned to the respective WP (as documented in the DoW) over the active time of the respective WP is assumed.

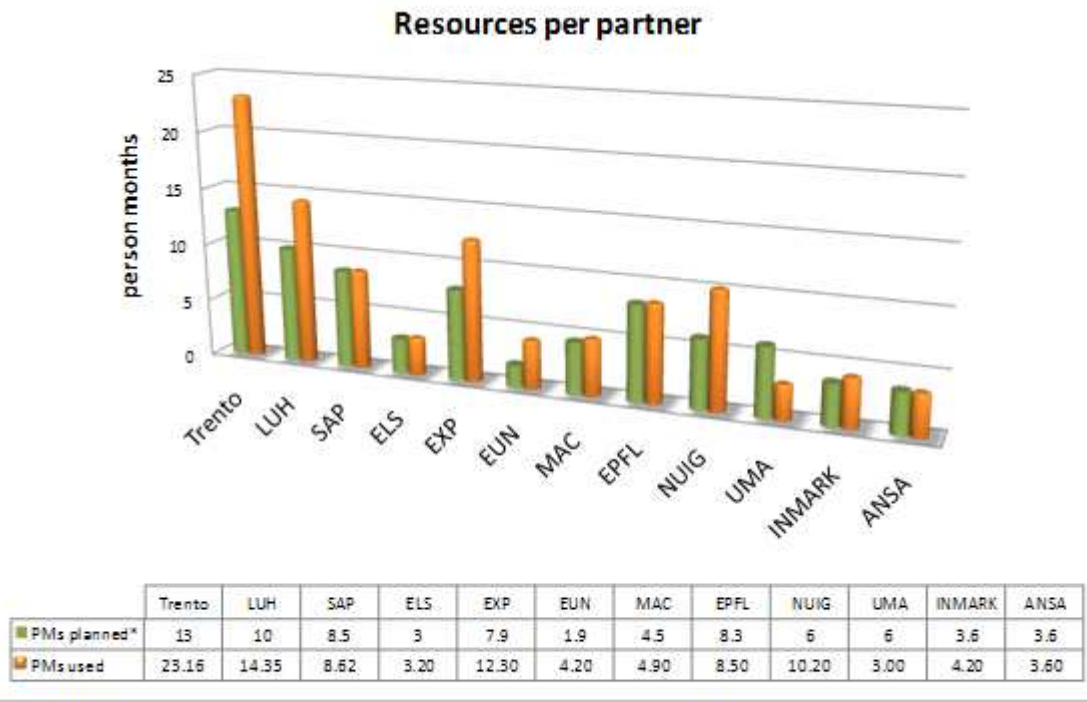


* assuming equal use of resources over runtime of WP

Figure 30: Use of Resources per WP in Q4 vs. Planned Use of Resources

6.5.1. Use of Resources per partner

The following graphics shows the use of resources in Q4 of the project, i.e. October 2008 – December 2008 per partner. The resources are considered in terms of person months spent in project activities. The actual numbers of resource used are compared with planned resource usage. For the planning numbers and equal distribution of maximum available PMs (as documented in the DoW) over the 30 months duration of the project is assumed.



* assuming equal use of resources over 30 month of project

Figure 31: Use of Resources per Partner in Q4 vs. Planned use of Resources