Simulating user activities for measuring data request performance of the ECM visualization tasks

Juris Rats

***Abstract*—**The aim of the research is to assess performance of the NoSQL database Clusterpoint when processing data requests of the large user base executing Enterprise Content Management (ECM) visualization tasks. The user activity model comprising user types, task types, data interaction types, expected preparation times and execution frequencies is defined. Visualization models relevant for the defined task types are specified. The data interaction types are matched against data request types of Clusterpoint database. The methodology for creating user activity flows for defined user base is described and used to measure Clusterpoint performance when processing the data requests generated by user activity flows. The performance tests are executed for user bases up to 30'000 users and for several database cluster configurations. The test results are analyzed and assessed.

***Keywords*—**Clusterpoint, Document Database, ECM, User Activity Simulation.

I. INTRODUCTION

The number of organizations wanting to profit from the Big Data technologies rapidly increase lately. The overall Big Data market is expected to grow to $41.52 billion in 2018 according to recent IDC (International Data Corporation) report [1].

The optimal choice of IT architecture and software components is of great importance for organizations. A wrong choice can lead to consequences and inefficiencies such as information loss, higher maintenance and redesign costs, stop of operational activities, and so on [2].

The ECM systems may profit from usage of the Big Data technologies as well because they allow to implement advanced communication and information sharing strategies like e-Government hexagon [3] which allows communication between different parties, including e.g. citizens commenting the government procedures.

In a recent study [4] we showed that NoSQL database Clusterpoint is a viable candidate for a backend of an ECM solution. Our current research aims to assess the suitability of NoSQL document database technology to handle the visualization tasks relevant for ECM when dealing with large data amounts (Big Data).

The research activity is co-financed by European Regional Development Fund, contract nr. L-KC-11-0003, project "Information and Communication Technology Competence Center", research No. 1.12 "The research of Advanced Visualization methods for the analysis of business related linked Big data".

We arrive there through these steps:

- We define activity model of the ECM user that includes definitions of user classes, relevant interactive visualization scenarios and their usage frequencies;
- We develop the set of atomic data interactions and match them to suitable NoSQL document database data requesting methods;
- We assess the performance of the NoSQL database when processing the identified data request sequences.

We demonstrate in our study that the visualization process can be organized in a sequence of tasks (controlled by user) so that each visualization step needs to process comparatively small subset of data even on very large databases. This is possible because typical data request of the visualization task returns aggregated data.

When dealing with Big Data even requests returning small amounts of data may take unacceptably long. The database indexing system is of great importance in this respect. The Clusterpoint inverted indexes allows for fast processing of aggregate values and this was one of arguments behind our decision to select Clusterpoint as a backend for the visualization platform.

The performance tests show that Clusterpoint database processes a typical workload of the 10'000 employee organization with a less then second response time on a single commodity server. For larger user bases the clusters with multiple nodes may be used.

Chapter 2 outlines the related research in ECM area.

Chapter 3 describes the user activity model.

Chapter 4 outlines the ECM specific visualization scenarios.

Chapter 5 briefly describes the application architecture used.

Chapter 6 defines the test database and matching of visualization specific data interactions to Clusterpoint data retrieval methods.

Chapter 7 presents the measurement methodology used.

Chapter 8 presents results of the performance tests.

Conclusion outlines the overall results of the research and suggests the direction for the future work.

II. ENTERPRISE CONTENT MANAGEMENT SYSTEMS

Enterprise content management (ECM) "comprises the strategies, processes, methods, systems, and technologies that are necessary for capturing, creating, managing, using, publishing, storing, preserving, and disposing content within and between organizations" [5].

ECM covers a wide area of functionality [6][7][8]. Fig. 1 [9] depicts the traditional application areas:

- DM (Document Management);
- Collab (Collaboration of supporting systems, group-ware);
- WCM (Web Content Management, including portals);
- RM (Records Management);
- WF/BPM (Workflow and Business Process Management).

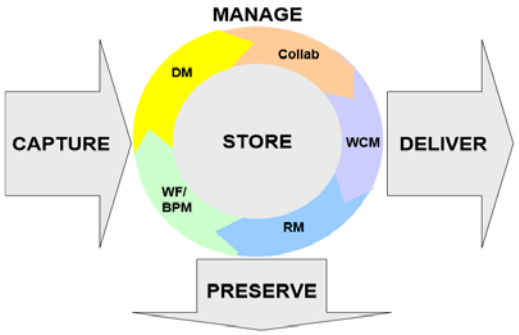

Fig. 1. Enterprise content management

ECM systems traditionally are using SQL databases as a persistence platform. Lately this has started to change and a number of NoSQL solutions are available for ECM market (e.g. Alfresco [10], NemakiWare server [11] for CouchDB). The ECM market is a comparatively new area for NoSQL solutions though. NoSQL Document databases appear to be a suitable platform for implementation of the persistence layer of ECM systems because they are meant to be schema-less, easily replicable and scalable [12]. Schema-less means that NoSQL Document databases are convenient for the semi-structured data of organizations. NoSQL databases store full entity data in one place (instead of scattering it in several related tables) that allows decreasing a number of read operations to get the data and enables for easy replication and splitting (sharding) of database entities between the database nodes.

III. USER ACTIVITY MODEL

We focus there mainly on a document management part of the functionality. Interacting (i.e. surveying) with the involved parties (users, solution providers, help desk) we established a set of ECM user classes in document management area. Four user classes have been identified as having specific requirements for the visualization functionality – Manager, Office Manager, IT Officer, Other. Surveying the users of all said classes the visualization scenarios and estimates for the usage frequencies of the scenarios were determined. The frequencies we have used later in our user activity model to determine the domain specific activity flow. The estimates of the relative count of the users inside groups were get interviewing the users and help desk.

The user activity model was developed according to the recommendations of the international standard ISO/IEC 14756 [13] which states that performance measurements must be performed by automated process that simulates activities of real users. The users according to the standard are divided into groups where users inside group have statistically similar activity profiles. User activities are represented as task chains and the user activity model is configured by setting statistical parameters of the activities and the tasks for the user groups. The statistical parameters include user activity frequencies, mean times and standard errors for task preparation times. The user activity model is defined as follows:

- The system in question is used in parallel by users of n types, the number users of type i equals to $N_{user}(i)$, the total number of users is N_{tot};
- The user activity on the system in question is a chain of tasks, where a task chain belongs to one of task chain types (total of u task chain types); the users of a user type i perform tasks of the type l with statistically determined relative frequency $q(i,l)$ (Table 1); the task chain consists of tasks that are executed one after other in the defined order;
- The tasks belong to the task types (total of w task types); user type i and task type j has statistically determined mean preparation time $h(i,j)$ and standard error $s(i,j)$.

The sample user activity model is shown on Table 1. This model is employed in our performance tests by substituting N_{tot} with the user amount in question and changing $N_{user}(1)$ to $N_{user}(4)$ accordingly.

Table 1. Sample User Activity Model parameters

Parameter	Value	Comments
n	4	Number of user types: 1 – other employee, 2 – office manager, 3 – IT officer, 4 – manager
w	19	Number of task types
u	20	Number of task chain types
$N_{user}(1)$	66	Number of other employees
$N_{user}(2)$	5	Number of office managers
$N_{user}(3)$	4	Number of IT officers
$N_{user}(4)$	25	Number of managers
N_{tot}	100	Total number of users

Thus for user base N_{tot}=5'000 we have $N_{user}(1)$=3'300, $N_{user}(2)$=250, $N_{user}(3)$=200 and $N_{user}(4)$=1'250.

IV.ECM SPECIFIC VISUALIZATION SCENARIOS

The visualization model is determined by data visualization type (chart type – e.g. bar chart, pie chart, bubble chart), characteristics of data involved (types, volumes) and set of allowed user interactions (setting data filters, drilling down on a specific data area etc.). The samples of the visualization scenarios selected are *Late Tasks*, *Documents with Tasks*, *Links* and *Audit Data*.

Scenario *Late Tasks* aims to assist user in analyzing number of late tasks and total number of tasks in respect to employees, organizational units and time periods. The data to be visualized are:

- Number of tasks in progress U_i

- Number of late tasks U_k
- Number of in time tasks U_l
- The relative amount of late tasks $U_{kp} = \frac{U_k}{U_k + U_l + U_i}$

A volume of data processed when user works with given visualization models is determined by interactions selected by user of a specific user class. We call the sequence of the user interactions on a visualization model the visualization scenario (or scenario for simplicity). The scenario consists of several steps each representing single data interaction. The initial step of the scenario provides the data to show when the user selects the given visualization model. The rest are determined by the user interactions when working with the visualization.

V. CLUSTERPOINT BASED APPLICATION ARCHITECTURE

Clusterpoint DBMS (Database Management System) [14] is a document-oriented shared nothing NoSQL database that uses sharding (database horizontal partitioning) to split the data and computational load over multiple cluster nodes. Shards are distributed across nodes in a shared-nothing architecture. The same Clusterpoint software image is installed on each node so the cluster operates in multi-master mode i.e. all nodes are equal and can handle all the operations.

A Clusterpoint database comprises one or more stores. Each store contains a repository and its associated index. The repository contains all data objects while the index contains information for all of the fields that might be used for search [14].

Each Clusterpoint store is a collection of XML documents. The XML document may contain the document text as well as metadata (Title, Author, Release Date, Rate etc.) fields [14].

We use for the visualization platform the 4 layer architecture proposed in our recent study [4]:

- User interface layer (client software on PC, smartphone, tablet).
- Business function or middleware layer (ECM business functionality).
- NoSQL Document-oriented DBMS layer (implements data clustering, replication, indexing, search, retrieval).
- Infrastructure layer (operating system, visualization, network and data transfer, event journaling etc.).

The main advantage of Clusterpoint solution for the visualization specific tasks is its indexing system that is based on usage of inverted indexes [14]. Clusterpoint index on, say, document author field, has an entry for each author in database, and contains author name and list of links to all documents of this author. This allows for fast document search against a particular author. The similar index is created for other data types as well, including date and numeric values.

Clusterpoint uses indexes to calculate aggregates. Facets and data range indexes are important in our case. Facets are used to index hierarchic taxonomies (like hierarchic structure of an organization, consisting of divisions, departments and employees). When facets are used Clusterpoint knows that employee data may be aggregated to create department data etc. Date range index is allows Clusterpoint to calculate aggregates by hours, days, weeks etc.

Let's illustrate the mechanism of inverse indexes on the sample facet index on document author. Index entry in this case is created for every author. Index entry indicates the document author and contains links to all documents of the given author as well as the count of such documents (see Figure 2). If the user is interested, say, in how many documents are created by staff members of the given department, the system selects all index entries of the department of interest and returns the document counts indicated in selected index entries. No access to documents is needed. The count of document created by each department of the given division are processed the same way: the system selects all index entries of the division and returns the department aggregates of the document counts indicated in selected index entries.

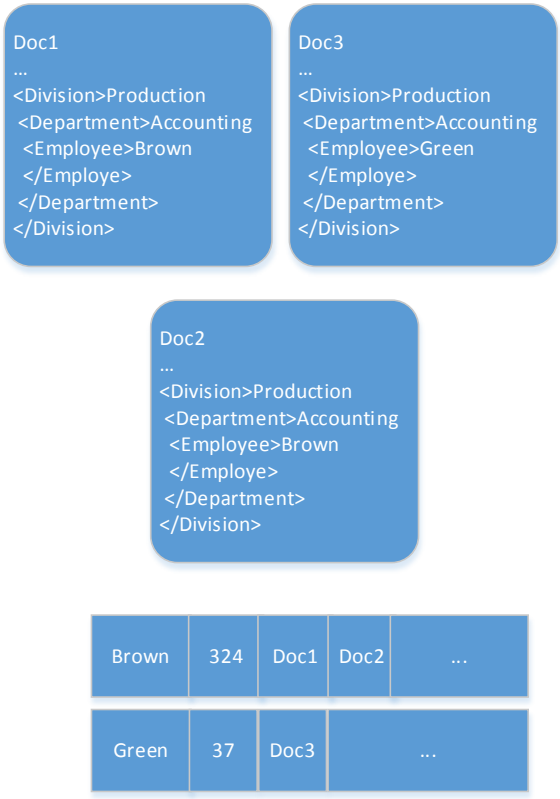

Figure 2. Clusterpoint inverse index

Usage of inverted indexes allows for the faster data retrieval because number of the index entries is smaller than that of the documents indexed (one entry may be linked to several or even large amount of documents). The inverted indexes in concert with facets and data range indexes enables fast processing of the aggregates - data requests typical for visualization tasks (e.g. – retrieving month totals or totals by department).

Apart from the technical architecture we take into consideration the business viewpoint as well.

1. We are interested in visualization system serving multiple organizations. This would allow managing content both of interest inside of organization and interchanged between several organizations. This way the content exchanged between organizations by e-mail (or other communication means) could be made available to parties via sharing. As we discovered the hit counts of the search requests relevant for the common visualization tasks on such a model is limited and does not increase significantly when we add more organizations to the common database. The reason is that organizations have access to their own data (and data shared with other organizations).
2. We organize the data visualization as a sequence of user-system interactions in a way that allows limiting the data necessary for a single interaction. User may ask for month totals over the year or day totals over the month, but not the day totals over the year, for example. This means, in particular, that application must aggregate data before returning it to the client layer and request more data when user asks for more details.

Assessing the data of about 50 municipalities we noticed that the hit count for the visualization data requests normally does not exceed 20'000. To be on the safe side we can assume that the hit count of the visualization data requests does not exceed 50'000.

VI. MATCHING INTERACTIONS TO DATA RETRIEVAL METHODS

Our objective is to evaluate the performance of the visualization specific data requests on a very large data amounts. Unfortunately there are no available ECM specific databases of appropriate scale. It was decided therefore to use as a test database English version of the publicly available Wikipedia database. The Wikipedia is regarded as a good approximation of an ECM document database as having a plenty of documents to search text in and a number of relevant for ECM metadata fields. To improve the database metadata fields from Dbpedia database were added.

We focus in our research exclusively on a performance of data request processing. This means in particular that data requests putting similar load on the system may be regarded as similar data requests and should be regarded as interchangeable. In another words we factorize all relevant data requests in equivalence classes so that all requests of a given equivalence class put the same load on a servicing system. To do this we notice first that for visualization tasks the only data request of interest is a search request (inserts, updates, deletes etc. are not relevant). As concerns Clusterpoint database the performance load of the search requests is determined by the following parameters:

- Clusterpoint data retrieval method (e.g. full-text search, xml search, facet selection, aggregation, selection by data range)
- Hit count (count of the results matching the selection criteria before the results grouping)
- Group count (count of returned rows, i.e. the count of result rows after the grouping)

The volume of the returned row can be disregarded as it is small in case of visualization tasks (usually a number representing the count of rows in a group).

We introduce two classes for both hit count and group parameters. Hit count class M is defined as having 1 to 999 hits. Hit count class L has 1'000 or more (up to 50'000) hits. Group count class M has 1 to 49 groups while group count class L has 50 or more groups. The sample data request class is xml search with hit class L and group class M. We have thus 4 performance specific classes in total for each Clusterpoint data retrieval method.

VII. METRICS USED

Recommendations of the Standard ISO/IEC 14756 [13] are used to organize the performance assessment. We defined user classes and task sequences. Execution frequencies for task sequences were determined as well as preparation times for the individual tasks and relative amount of users in user classes. Using these parameters we developed a performance evaluation process that is executed in three separate steps

- The task execution schedule is created which determines what task to execute when;
- The schedule is executed on the test harness and the execution results are logged;
- The execution log is analyzed and the performance measures are saved.

The measurement process is showed in Figure 3. The process is adapted version from one defined by the standard. We removed two steps as not important for our research - the step for checking the correctness of the system in question (we are checking the performance not functionality) and for checking the statistical correctness of user activity schedules (we test this in advance, not inside the measurement process).

Below follows the description of the software metrics used for the performance evaluation. Software metrics are measures that are used to quantify software, software development resources, and/or the software development process [15].

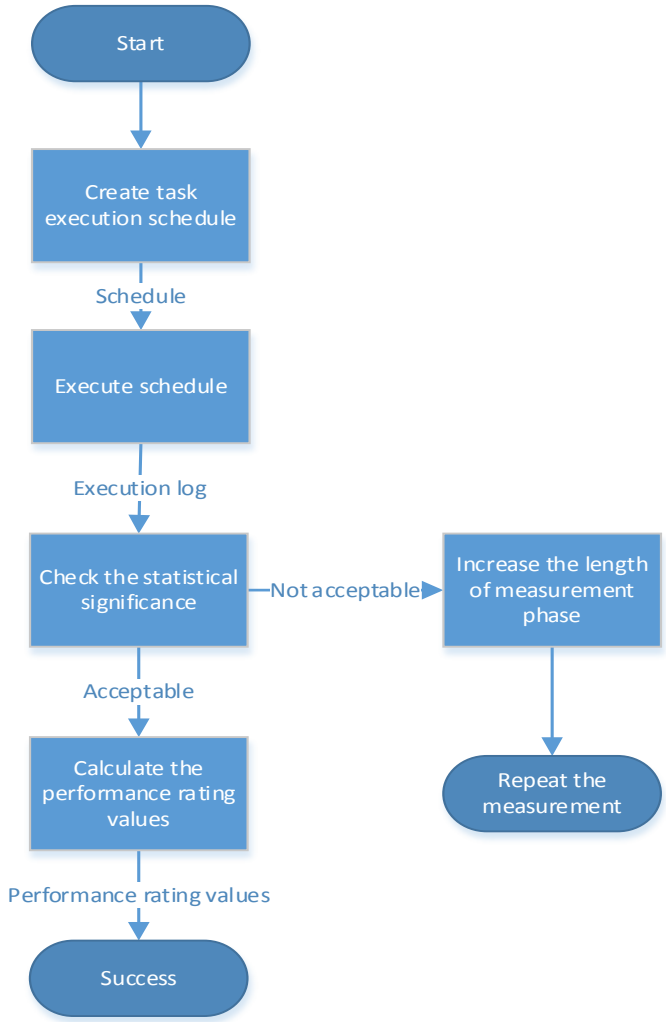

Figure 3. Performance measurement process

A. *Statistical significance*

The statistical significance of the measurements are checked by formula

$$d_j = Z_{\frac{ALPHA}{2}} * \frac{s_{ME}(j)}{\sqrt{K(j)}}$$

Here

- ZALPHA/2 is the critical value corresponding to the confidence coefficient ALPHA of the tasks mean execution time;
- sME(j) – the standard error of the execution mean time of the j-th task type;
- K(j) – a total number of tasks of type j in task chains started during the rating interval.

We use ALPHA=95% in our research. The corresponding critical value is 1.96 which results in

$$d_j = 1.96 * \frac{s_{ME}(j)}{\sqrt{K(j)}}$$

We consider the measurement results to be statistically confident if for all task types *j*

$$d_j \leq 0{,}5$$

B. *Performance measures*

Three main measures are calculated – task frequency *B(j)*, timely throughput *E(j)* and task mean execution time $T_{ME}(j)$.

The Task Frequency is calculated by the formula

$$B(j) = \frac{K(j)}{T_R}$$

Where

- K(j) – total amount of the tasks of the task chains started during the rating interval;
- TR – the length of rating interval.

The Mean Execution Time for the task type *j* is calculated according to the following formula

$$T_{ME}(j) = \frac{t_{ET}(j,1) + t_{ET}(j,2) + \cdots + t_{ET}(j,K(j))}{K(j)}$$

Where

- K(j) - total amount of the task chains started during the rating interval;
- tET(j,k) – the Mean Execution Time of the tasks of type j.

The Timely Task Frequency:

$$T_{ME}(j) = \frac{K_E(j)}{T_R}$$

Where

- KE(j) – total amount of the timely tasks of the task chains started during the rating interval;
- TR – the length of rating interval.

We use for our research timeliness function with two time classes. The formula for total amount of the timely tasks in this case is as follows

$$K_E(j) = \frac{T_1(j) + T_2(j)}{T_R}$$

Where $T_1(j)$ and $T_2(j)$ are the number of timely executed tasks according to time class 1 and 2 respectively. Finally

$$T_1(j) = \min(t_{ET}(j,1), K(j) * r_T(1))$$

$$T_2(j) = \min(t_{ET}(j,2) - T_1(j), K(j) * (1 - r_T(1)))$$

Where

- tET(j,k) is a number of tasks of type j with the execution time not exceeding the time limit of the time class k;
- K(j) is a total number of tasks executed;
- rT(1) is a minimum relative frequency of the tasks of time class 1;
- rT(2) is a minimum relative frequency of the tasks of time class 2; as we have 2 time classes rT(2) = 100%, this is taken into account in the formula for T2(j).

VIII. ASSESSING THE PERFORMANCE

Data request schedules where generated for different user bases starting from 1'000 to 30'000 users (sample schedule shows Figure 4).

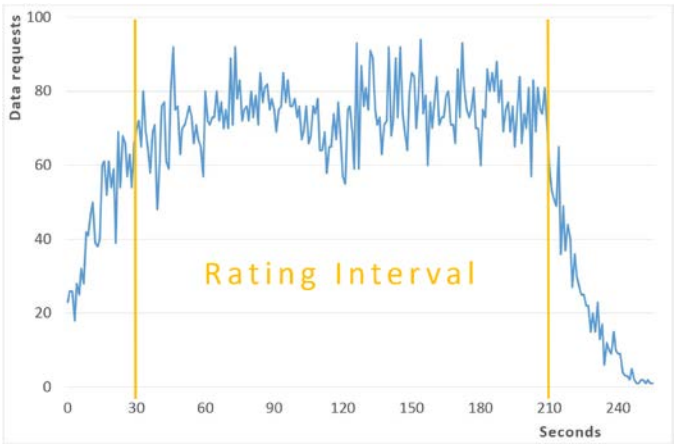

Figure 4. Data request schedule for 30'000 user base

Performance tests (amounting almost 470K executed data requests) were executed on several cluster configurations (one to four clustered nodes hosting replicas of the test database). The performance tests show that

- Visualization process can be organized as a series of steps in a way that allows very fast response times (0.5 seconds for 90% of data requests) on Clusterpoint database even for large data amounts; the primary database may be used for this (no preliminary data processing like replication to data warehouse needed);
- Clusterpoint database provides timely execution of the ECM visualization specific data requests (on commodity hardware) up to the 5'000 user base on a cluster node; Figure 5 shows that all requests processed timely (timeliness value is 100%) on a commodity server for up to 5'000 users; timeliness value drops below 100% for 20'000 users for 4 node cluster (i.e. 5'000 users on a cluster node) as shown in Figure 6;
- Clusterpoint database can be scaled out adding new database replicas with less than 12% overhead (see Figure 7).

Figure 5. Execution time and timeliness for one commodity server.

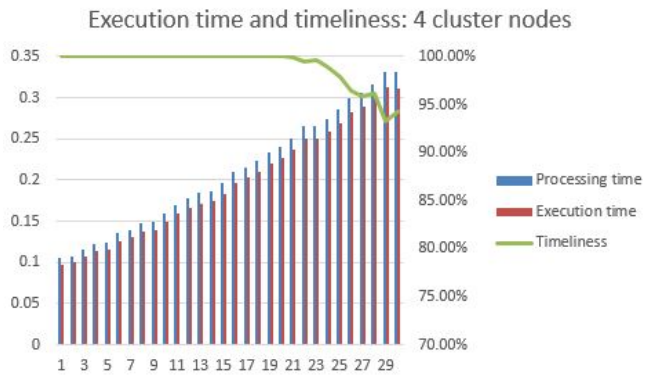

Figure 6. Execution time and timeliness for database replicas on 4 node cluster.

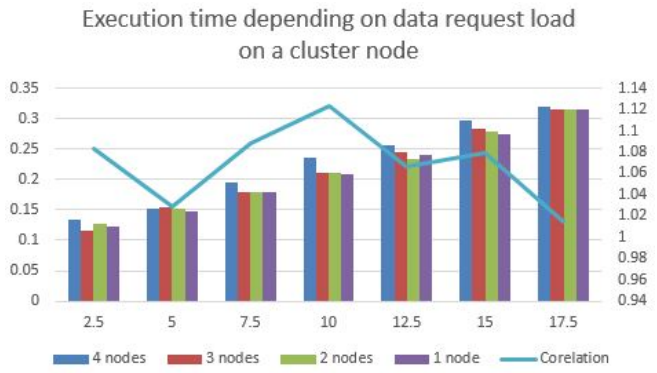

Figure 7. Execution time of the data requests depending on user amount and cluster node count

The hardware used for the testing is a virtual processing unit - Xen virtualization, CPU 4 cores 2.00GHz, RAM: 32GB, HDD: SSD 1TB.

IX. CONCLUSION

The user activity model and performance measurement metrics were defined to allow executing performance tests for simulated activity flows of an ECM user.

The results of the performance tests indicate that the architecture proposed in our recent study [4] supported by reasonable organization of the visualization process should allow to implement visualization module for ECM domain on a primary database supporting real-time processing of user requests for very large data amounts. The system on a single commodity server should be able to process effectively (in a less than second timeframe) the workload generated by 5'000 ECM users. For larger user amounts the cluster with multiple nodes should be used (one node for each new 5'000 users).

We have addressed typical visualization related data retrieval scenarios (no updates and inserts) so far. The next step of the research should deal with performance evaluation of the visualization specific data requests when executed in parallel with massive data inserts and updates.

REFERENCES

[1] A. Nadkarni and D. Vesset, "Worldwide Big Data Technology and Services 2014–2018 Forecast - 250458," *IDC*, 2014. [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=250458. [Accessed: 06-Mar-2015].

[2] F. E. Pani, G. Concas, D. Sanna, and L. Carrogu, "The FAME tool: An automated supporting tool for assessing methodology," *WSEAS Trans. Inf. Sci. Appl.*, vol. 7, pp. 1078–1089, 2010.

[3] S. Simonova and H. Kopackova, "eDocument in eGovernment," *WSEAS Transactions on Information Science and Applications*, 2010. [Online]. Available: http://www.wseas.us/e-library/transactions/information/2010/89-120.pdf. [Accessed: 09-Mar-2015].

[4] J. Rats and G. Ernestsons, "Clustering and Ranked Search for Enterprise Content Management," *Int. J. E-enterpreneursh. Innov.*, vol. 4, no. 4, pp. 20–31, 2013.

[5] K. R. Grahlmann, R. W. Helms, C. Hilhorst, S. Brinkkemper, and S. van Amerongen, "Reviewing Enterprise Content Management: a functional framework," *European Journal of Information Systems*, vol. 21. pp. 268–286, 2012.

[6] O. R. Nilsen, "Enterprise Content Management in Practice," University of Agder, Kristiansand, Norway, 2012.

[7] J. Korb and S. Strodl, "Digital preservation for enterprise content: a gap-analysis between ECM and OAIS," in *7th International Conference on Preservation of Digital Objects (iPRES2010)*, 2010, pp. 221–228.

[8] B. T. Blair, "An enterprise content management primer," *Inf. Manag. J.*, vol. 38, pp. 64–66, 2004.

[9] U. Kampffmeyer, "Enterprise Content Management ECM. White paper," Hamburg, 2006.

[10] K. R. D. Caruana, J. Newton, M. Uzquiano, M. Farman, *Professional Alfresco: Practical Solutions for Enterprise Content Management*. Wiley: Wrox professional guides, 2010, p. 514.

[11] "NemakiWare and CmisSync: A true open-source CMIS stack - TechRepublic," *Tech Republic*, 2014. [Online]. Available: http://www.techrepublic.com/article/nemakiware-and-cmissync-a-true-open-source-cmis-stack/. [Accessed: 09-Mar-2015].

[12] J. Potts, "Alfresco, NOSQL, and the Future of ECM," *ECM Architect*, 2010. [Online]. Available: http://ecmarchitect.com/archives/2010/07/07/1176.

[13] "Information technology — Measurement and rating of performance of computer-based software systems. International Standard ISO/IEC 14756." 1999.

[14] "Clusterpoint DB at a glance," 2013. [Online]. Available: http://docs.clusterpoint.com/wiki/Clusterpoint_DB_at_a_glance. [Accessed: 09-Mar-2015].

[15] L. Lazic and N. Mastorakis, "Cost effective software test metrics," *WSEAS Trans. Comput.*, vol. 7, pp. 599–619, 2008.