Experimental Validation as Support in the Migration from SQL Databases to NoSQL Databases

Abraham Gomez
Département de génie logiciel et des TI
École de Technologie Supérieure (ÉTS)
Montreal, Quebec, Canadá
abraham-segundo.gomez.1@ens.etsmtl.ca

Rafik Ouanouki
Département de génie logiciel et des TI
École de Technologie Supérieure (ÉTS)
Montreal, Quebec, Canadá
rafik.ouanouki.1@ens.etsmtl.ca

Anderson Ravanello
Département de génie logiciel et des TI
École de Technologie Supérieure (ÉTS)
Montreal, Quebec, Canadá
ravanello@gmail.com

Alain April
Département de génie logiciel et des TI
École de Technologie Supérieure (ÉTS)
Montreal, Quebec, Canadá
alain.april@etsmtl.ca

Alain Abran
Département de génie logiciel et des TI
École de Technologie Supérieure (ÉTS)
Montreal, Quebec, Canadá
alain.abran@etsmtl.ca

***Abstract*—NoSQL databases, also known Not only SQL databases, is a new type of databases that provides structures other than the tabular relations used in relational databases, for storage and retrieval data. This new databases are now a valuable asset to design complex real-time applications that use Big Data in cloud environments (NoSQL cloud databases). Today, the migration process from relational databases to NoSQL databases is unclear and mainly based on heuristics approaches, such as the developers' experience or intuitive judgments. This paper which forms part of a more extensive research project regarding how the design and use of a guidelines set could improve the migration process. The results present an experiment designed to obtain a baseline that allows an effective comparison between two migration processes: the first one, without the use of any guidelines and based on the traditional heuristic approach and the second one, with the guidelines. The experiment reports that the use of such guidelines improves the migration process.**

Keywords-Column oriented databases; NoSQL databases; distributed databases; software experimentation; cloud computing.

I. INTRODUCTION

Cloud Computing (CC) is rapidly becoming an integral part of our daily life. The applications developed for this new way of computing has brought new challenges for software engineering because of the large amount of data that is collected by the CC software applications. In fact, these applications accumulate and analyze a lot of information on a daily basis, which has led to create a new research area called "*big data*".

Until a few years, companies had used massively the relational database technology (RDBMS) to deal with this bigdata. However, Abadi [2] states in his research that accessing petabytes of data efficiently using RDBMS, in the cloud, is very challenging, and solutions like sharding, creates many other problems. At this point, the NoSQL databases emerge as a solution to these challenges.

This generates the problem about how to migrate the data from RDBMS to NoSQL environment. Unfortunately, little work has been done to explore the migration from RDBMS to NoSQL. As a matter of fact, there have only been preliminary researches focusing on certain minimum elements such as tables or types of relationships. One first approach was reported by Chonxing [3], who proposed some migration rules for a conversion to HBase. However, more experimentation is needed, to show that a one-size-fits-all approach is not possible and, more importantly, not all applications are good candidates for this migration, according to Stonebraker in [4] [6].

Despite the above situation, there are a lot of enterprises that offer the migration service from RDBMS to NoSQL, but they conduct this migration using a heuristic approach that implies a lot of experience in NoSQL environments. On the contrary, the use of a standardized formal way could help those that are NoSQL neophytes, but have experience in RDBMS.

This paper forms part of a more extensive research project regarding how the design and use of a guideline set could improve the migration process from RDBMS to NoSQL. At the end of the project, the idea is to offer something that really improves the process; so, it is important to have some baseline that allows, at least, a

 ISBN: 978-1-61208-388-9

comparison. The work presented here focuses on the design of an experiment to establish a baseline that allows a valid comparison between the migration process of a database from RDBMS to NoSQL. This paper reports on the design of the experiment, its application and its results. In a near future paper, a second group of RDBMS experts will conduct the same migration, but using guidelines developed by our research team.

The paper is organized as follows. Section II presents the problem statement. The research objectives are presented in Section III, followed by the Section IV with the related and previous work of the project. Section V presents the experimental design and Section VI reports the case study results. Section VII presents the conclusion, and finally, the future work is presented in Section VIII.

II. PROBLEM STATEMENT

Although, it is still very popular to use the relational database model for CC applications, when the data deployed in the database servers (in the cloud) grow beyond 1TB, this technology starts to show its limits, e.g., in their work Stonebraker [4][6][8] state the volume of data stored is related with problems in response time in the research field of "big data".

Also, the large increase in the number of users connected to cloud applications can cause other problems, such as transactional difficulties, storing space management and the non-compliance of ACID properties (ACID is an acronym for Atomicity, Consistency, Isolation, and Durability. Complying with all these properties guarantee that database transactions are processed reliably). That is, the administration of these systems becomes more and more complex, as reported by Abadi [2].

When a big data application, using a relational database technology, reaches its limit and the solutions, such as, sharding, are failing to solve the issues its time to think about NoSQL technologies. This solution provides new levels of economies of scale, agility, and flexibility compared with traditional IT environments based on the relational database model.

The problem that the whole project wants to address is as follows:

Since the industry uses mainly relational databases and they are likely to migrate some of their large scale existing applications to a NoSQL model, there is a research need to improve this process by identifying a set of guidelines to help database specialists in this first time migration from RDBMS to NoSQL database. Our experiment will be focusing on an HBase migration, which is a popular column-oriented NoSQL database developed as part of Apache Hadoop project.

III. RESEARCH OBJECTIVES

First of all, it is necessary to make a distinction between the research objective and the paper's scope. The research objectives is the design and use a set of guidelines as a way of improving the migration process of databases from RDBMS to NoSQL databases, focusing on HBase. In this article, the concept "improve" is going to be used in the sense to bring into a more desirable or excellent condition the current migration process from SQL database applications to NoSQL database applications. Fig. 1 shows, graphically, the difference between the two things (the paper scope is inside the red dotted border). The idea will be use the same relational database application and follow two experimental tracks: the first, without the use of the guidelines, and the other one with the use of the guidelines. At the end of the project, two HBase databases will be obtained and compared (see Fig. 1).

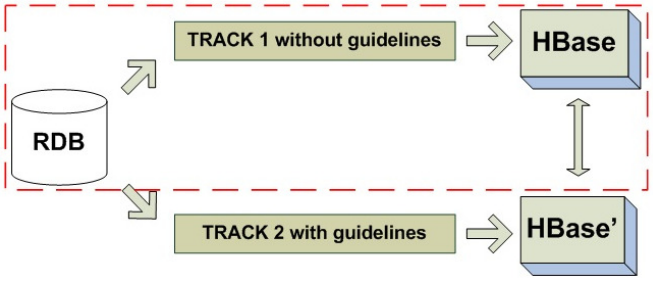

Figure 1. This paper scope versus the overall research objective.

In order to conduct a comparison between the applications and resulting databases HBase and HBase', some preliminary experiments must be conducted. The scope of this paper is to create a baseline that allows a valid comparison between the migration process of a database from RDBMS database to NoSQL database (HBase), without the use of guidelines (heuristic approach) and with the use of guidelines (the proposed solution)

IV. RELATED WORK

Few researchers have addressed the problem of migration of a software application from RDBMS environment to NoSQL. Indeed, there is very little literature on how doing it, at least in a standard way; this fact is supported in the references, because there is no recent research work on this field, the closest is around 2011, and the others references are significantly older than that.

The options currently available to accomplish this migration are mostly based on a heuristic approach. It means, based on the developers' experience, educated guess, intuitive judgment, or common sense. This approach does not guarantee that an optimal solution will be found; but, if it is properly done, it can provide a satisfactory solution.

Salmen has proposed, in his initial attempt focused on identifying some of the core activities that are common to every migration process, to draw some general conclusions about how start this migration process (e.g., the DDI methodology proposed by Salmen [13]), where DDI stands for Denormalization, Duplication, and Intelligent keys.

Denormalization is the process used to optimize the read performance of a database by adding redundant data or by regrouping data. Data duplication can be defined as the occurrence of redundant data within a system by various means. An intelligent key is a database key which depends wholly on one or more other columns in the same table. An intelligent key might be identified for implementation convenience, when there is no good candidate key.

 ISBN: 978-1-61208-388-9

Another contribution to this research field was published by Chongxin who developed some rules to help in the migration from relational databases to a specific cloud computing database, which is HBase [3]. However, Chongxin explored a reduced set of ideas that he called "rules" (three to be precise) and these rules do not cover the entire characteristic that implies a relational database application today. Besides, Chongxin establishes their rules in a consecutive order, it means, in the first stage, one must apply the rule number 1; then, the rule number 2, and finally, the rule number three. This way of working reduces drastically the results of Chongxin's method because the solution offered only covered the relational aspect "relationships" (including the migration of "one-to-one", "one-to-many" and "many-to-many" relationships), the author does not offer a way to deal with other relational aspects like tables, fields, store procedures or triggers. Furthermore, the rules were designed to be applied in a pipeline way (one after the other) and not as the user needs it, indeed, the Chongxin's work does not offer any method to deal with the case if one user tries to apply the rule number two, without applying the rule number 1 previously.

Notwithstanding the above, the Chongxin's work offers an initial approach to the problem of migration a RDBMS to NoSQL. In fact, separate the relationships into its different types and create a rule for every type is an excellent first solutions' approach.

The last work was suggested by Singh in [14][15]. In their work some general guidelines were proposed, but the problem is the guidelines were developed using the methodology of use cases that follow a heuristic approach and reduce the possibilities to replicate the work or adapted in general ways to applied in other contexts.

It is clear from the above that there is a twin problem: first and foremost, a new perspective to address the migration problem is needed. But, along with this new solution, a way to measure the impact of the new solution is also necessary. This document is part of an entire research whose goal offers some guidance for converting an RDBMS to NoSQL database based on a guideline set. The goal of the paper is to offer an experimental baseline that allows the comparison results between two migration's processes, one using the guidelines and another one without the use of the guidelines.

V. EXPERIMENTAL DESIGN

As stated earlier, in Section III, this experiment is the first part of the entire research project to address the migration problem from relational to NoSQL databases. Only the results of the experimentation of track 1 will be presented and used as a baseline for the future comparison with the track 2 experimentation results (see Fig. 1). This experimental design was based on Jedlitschka's work [16].

A. *Hypotheses*

The null hypothesis is:

H_0: "there is no real improvement in the migration process with the use of guidelines; if there is any advantage, it is only coincidental, and the best option is to use a heuristic approach based on the developer's experience".

On the other hand, the alternative hypothesis is:

H_1: "there is a significant improvement in the migration process with the use of guidelines; this is not coincidental and the better option to achieve this process is to use the guidelines".

B. *Case study participants and data collection procedure*

This subsection follows the structure of Easterbrooke [19], Marcos [20], and Zelkowitz [1]; moreover, the experiment was designed using the point of view of a typical developer. Taking this point of view, the participants were asked to state their experience level and they were classified according to 1) their academic background 2) working field; 3) number of years of work experience with relational database, and 4) the number of years of work experience with any NoSQL database.

The word "*experience*" was related to the domains of programmer, relational database programmer or relational database administrator.

Moreover, the classification was summarized according to different options. The academic background had the options Graduate with PhD, Graduate with Master, Graduate, and Undergraduate Student. The working field had the options Industry, Academic, and Research Center.

The number of years of work experience with relational database environment had the options of No Experience, Low Experience (less than a year), Middle Experience (2 to 5 Years), and Advanced Experience (more than 5 Years).

The number of years of work experience related to any NoSQL database had the same options as above.

The goal was obtain a classification for the participants according their experience that would allow us to know the combinations (pair) "*relational-NoSQL*" experience that needs the solution and where it can be most useful. Fig. 2, for instance, highlights the pair Low-Medium, meaning a "*low*" experience in relational database environment and "*medium*" experience in NoSQL database.

Eighteen individuals participated in the experiment: twelve participants belong to the industrial sector and four participants were graduate students at the École de Technologie Superieure (ÉTS). All participants were provided with a clear and well established knowledge about the purpose of the experiment.

Relational Database	No-SQL Database
Low	Low
Medium	Medium
High	High

Figure 2. Classification for the participants according their experience.

The material used in the execution was:

- The document including the call for participants (date, time, place and activities that took place in the workshop), which was an invitation sent by email

and telephone calls two months before the workshop.

- The participant's instructions.
- The synthetic relational schema (Blue document). This was the schema that they must have migrated to NoSQL. The schema was based on the research of Singh [14] and it was composed by seven tables, four large tables (City, Department, Doctor and Hospital) and three junction tables (DoctorDeparment, HospitalCity and HospitalDepartment,). The tables City, Department, Doctor and Hospital remain the classical "Id, Name" structure, with "Id" as primary key. Table "Doctor" contains "Id, Name, Age, Sex and BorIn". The latter field is the Id of the city where the doctor was born. The junction tables allow expressing the "many-to-many" relationship indicated by each junction table's title (see Fig. 3). It is important to note, the participants were offered the opportunity to choose between several sub-schemas from the main schema. For instance, one participant could choose only migrate the sub-schema composed by the entities Hospital – HospitalDepartment – Deparment or the sub-schema Doctor – DoctorDepartment – Deparment or the participant could select the entire schema (see Fig. 3).
- The NoSQL solution (Green document). This was an empty sheet, where the participant could draw the new schema resulting from their knowledge.
- The participants training document (White document). It was a document that summarizes the training part explained at the beginning of the experiment, including the relational database and Not-SQL explanations.
- The drafts documents (Yellow documents). It means sheets to draw any thing the participant could use as support.

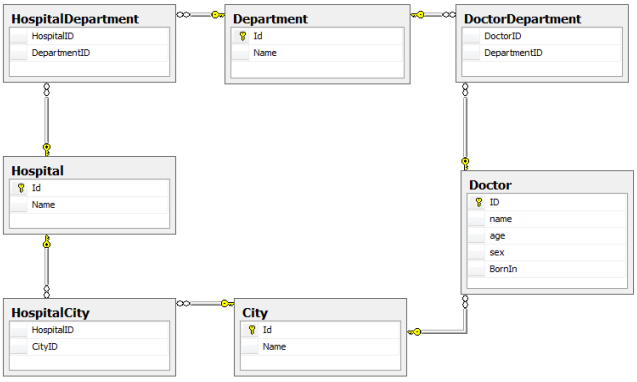

Figure 3. Relational schema given to the participants.

- The final survey form, which was applied to the participants, after the experiment. Besides, it was one of the two measurement instruments used in the experiment. The second one was the schema designed in the green document. The survey was designed following the research work devised by Kasunic [21] and Lethbridge [22]. It was composed of nine questions, with the first four were totally oriented to "experience classification", as explained earlier. The fifth question was related with the migration process and the opinion about the first step to begin it. The sixth question was related with the effort needed to achieve the process without the guidelines. This question was rated from 1 to 5, where 1 indicates that the process was easy to achieve without effort, a value of 3 indicates that it was required a maximum effort to achieve it and a value of 5 means that no matter how comprehensive the effort, it was not possible to achieve it. The seventh question was designed to evaluate their level of confusion during the process, e.g., no idea where to start or what the next step was. The questions were rated from 1 to 5, where: always confused, very often confused, sometimes confused, rarely confused, and never confused. The eighth question is a matrix for evaluating the percentage that covers the designed solution with regards to the relational aspects mentioned earlier (Table, Constraint, PK, and FK). Finally, the ninth question, the participant's opinion to know if he/she thinks that to receive some guidelines could improve their process. This question was rated 1 to 5 with the levels: strongly agree, agree, undecided, disagree, and strongly disagree.

VI. CASE STUDY RESULTS

As previously mentioned, there are no experiments and data that support conclusions or decisions in the domain of migration from RDBMS to NoSQL databases. Generally speaking, all the migrations have been conducted using a heuristic approach, e.g., the developers experience or the developer's educated guesses or their common sense.

The goal of this paper is using the results obtained as a baseline for comparisons in future stages of the entire research. The experiment process consisted of two well established parts, first at all an explanation of all the technological context, it means, a tutorial about the RDBMS and the NOSQL technology, a duration of 30 minutes was scheduled. After, all the participants received the documentation stated in the section above. Subsequently the participants conduct the experiment, eventually filling the green sheet (the NoSQL schema resulting from the migration). Finally they expressed their opinions filling a survey.

Table I indicates the different educational level of the participants. Generally speaking, it reports a low level of interest from undergraduate students to participate in this kind of studies. Besides, in the participants is found an 89% of graduate that shows an interest to conduct the experiment. (50% graduates with master plus 39% of graduates).

It can be observed in Table III that a great number of participants have the experience in RDBMS field; 45% have more than 5 years of experience and this result together with Table II's result (83% of participants in industry sector) give a lot of value to the results of this experimentation.

 ISBN: 978-1-61208-388-9

TABLE I. EDUCATIONAL LEVEL OF THE PARTICIPANTS

Educational Level	
Classification	***Response in percentage***
Graduate with Phd	5%
Graduate with Master	50%
Graduate	39%
Undergraduate	6%

Table II shows a great participation from industry sector (83%).

TABLE II. WORK AREA OF THE PARTICIPANTS

Area of work	
Classification	***Response in percentage***
Industry	83%
Academic	17%
Research Center	0%

In contrast, Table III also shows that 94% of the participants have no knowledge about NoSQL databases technology. The results shown by Table III strongly indicate that a set of guidelines could be an invaluable tool for the RDBMS experts in migration process.

TABLE III. TABLE LEVEL OF EXPERIENCE IN DB

Type of DB	**Experience in years**			
	No Exp	***Low Exp (< 1 Year)***	***Middle Exp (2-5 Years)***	***Advanced Exp (>5 Years)***
RDBMS	22%	11%	22%	45%
NoSQL	94%	0%	6%	0%

With regards to the first thing to do at the beginning of migration process, Fig. 4 provides the different paths presented in the participants. Considering Table II (83% in industry sector) and Table III (45% with more than 5 years of experience), there was a large proportion of 61% (resulting from 33% plus 28%) of the participants that chosen to begin with the RDBMS "tables" element (see Fig. 4). This leads to think that start by the tables could be a good guess.

The difficulty during the whole process is reported by Fig. 5. As can be seen, the initial perception that the procedure is difficult was unchanged (near 78% resulting from 39% plus 39%). This notion was reinforced considering also Table III data, about experience in NoSQL databases. So, the participants think the process demands a considerable amount of effort, because the NoSQL databases are totally new for them.

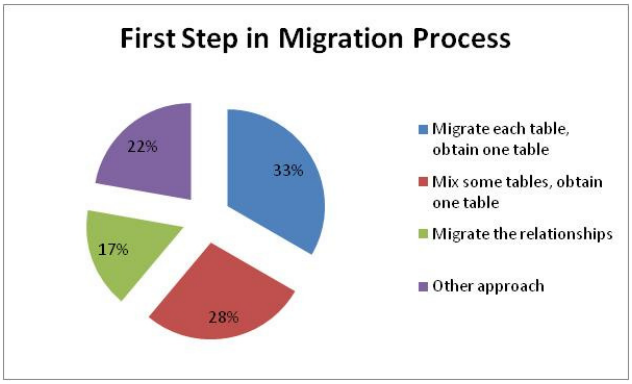

Figure 4. First step in the migration process.

The above argument is further reinforced by Fig. 6; it demonstrates that the majority of the participants (44%) had felt sometimes confused, i.e., without knowing how to go about it.

Fig. 7 provides the opinion of the participants in case that a set of guidelines it had been provided. 28% strongly agreed about their usefulness and 44% are agreeing with the relevance of this kind of tool in the migration process.

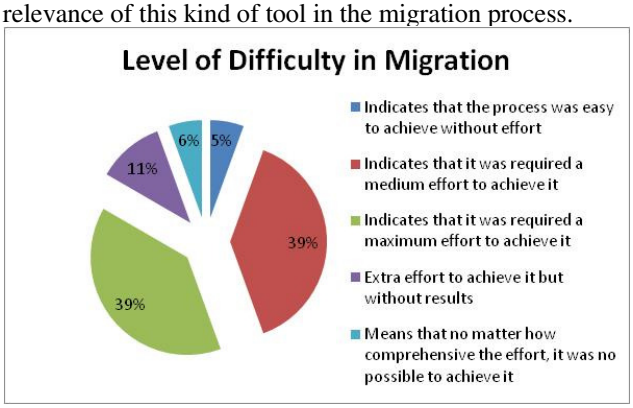

Figure 5. Level of difficulty in the migration process.

In the matters of the different database aspects, only five were studied in the experiment: tables, constraints, Primary Keys (PK), Foreign Keys (FK) and others (including all the aspects not specified in a clearly way such as fields, types of relationships, views, indexes, procedures and triggers).

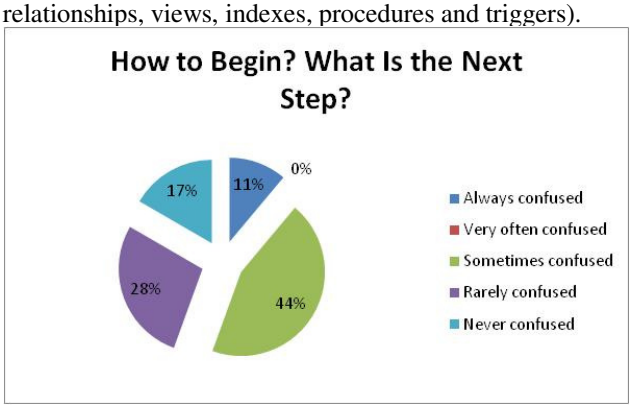

Figure 6. How to begin the process?.

Table IV reports the above information displaying the relational aspect covered against the percentage of coverage in terms of 0%, 25%, 50%, 75% and 100%. For instance, it is reveals that 50% of the participants think that their solution cover the relational aspect "tables" in a 100%. In contrast, 22% think that their solution covered this aspect in a 0%.

Moreover, Table IV presents that 28% of the participants think that their solution covers 100% the relational aspect "constraints". On the other hand, 39% of the participants think that their solution covered this aspect in a 0%.

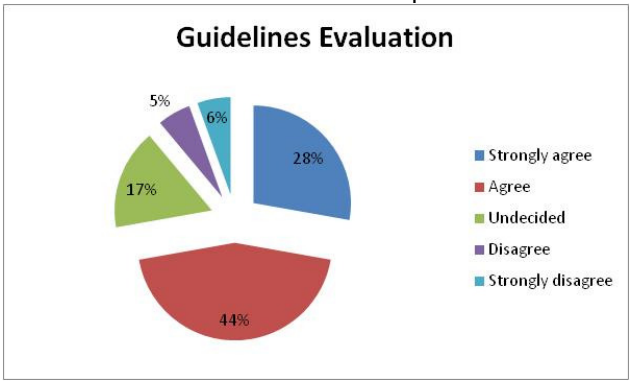

Figure 7. Guidelines evaluation.

TABLE IV. TABLE LEVEL OF COVERAGE IN DIFFERENT DB ASPECTS

Relational aspect covered	Percentage of coverage				
	0%	*25%*	*50%*	*75%*	*100%*
Table	22%	0%	6%	22%	50%
Constraint	39%	11%	17%	5%	28%
PK	29%	0%	18%	12%	41%
FK	28%	0%	11%	22%	39%
Others	94%	0%	0%	0%	6%

Furthermore, Table IV reports that 41% of the participants think that their solution covers 100% the relational aspect "primary keys". However, 29% think that their solution covered this aspect in a 0%.

Table IV shows that 39% of the participants think their solution covers 100% the relational aspect "foreign keys". Conversely, 28% think that their solution covered this aspect in a 0%.

Other relational database improvement aspect like fields, store procedures or triggers were put together in the relational aspect "others" and it reveals that 94% of the participants show no interest in these aspects. See Table IV, last row.

VII. CONCLUSION

The problem of migrate a relational schema from RDBMS to NoSQL environment was studied. We have developed a complete experiment that will allow a comparison with futures solutions approaches. This comparison will use the data obtained as a baseline. Besides, the data obtained in the experiment show the need for a formal and standard way to conduct this kind of migration. Despite the fact that today the migration is possible using heuristic techniques based on developer's experience, we have demonstrated that not all the staff with expertise in RDBMS has the enough expertise in NoSQL to carry out this kind of migration.

A surprising outcome of the experiment was that nearly all participants try to migrate the relational aspect "table" first, but they do not pay attention to other relational aspects like "relationships" or "fields". To explain this empirical finding, we investigated the background of the participants and we strongly believe, this result, was mainly caused by their large RDBMS experience.

Because this is the first step of the project, we cannot make a hypothesis testing, and much less, accept or reject the alternative hypothesis.

On the other hand, it is possible to show some feedback based on comments received during the workshop. Any information about guidelines was given to participants. It is reasonable to assume that those without familiarity in database have experienced more difficulties than others with some years of working with them.

The comments about the training session were positive in general. Despite the experiment trainer's effort, it can be observed that during the first half hour of the experiment there was a considerably spent of time consulting the reference documentation, especially those participants without the requested experience. According to the feedback of some PhD students, the first obstacle was to figure out what could be the first step to start the process. We expect to complete the all the research objectives in summer 2015.

VIII. FUTURE WORK

In near future, a more in-depth analysis will be presented. Besides, we will finish the guidelines set. The idea is refine the set with the advice of experts NoSQL users. Once the process of refinement is done, we will apply the same experiment with the same synthetic relational schema. At the end, the comparison explained in Section III will be conducted.

ACKNOWLEDGMENT

The authors gratefully acknowledge the ideas and support of David Lauzon and Khaled Almakadmeh from ÉTS software engineering department, and Jean-Marc Spaggiari, from Cloudera. Some of their ideas and perspectives were used in the experiment section. Special thanks also to Luis Bautista for his ideas about some solutions approaches. Besides, the authors want to express their gratitude for the guidance and invaluable technical assistance received from Patrice Dion, responsible for the technological platform of the software engineering laboratories in the ÉTS. Their help was invaluable to achieve this paper.

REFERENCES

[1] M. V. Zelkowitz, D. R. Wallace, and D. W. Binkley, "Experimental validation of new software technology", in Lecture notes on empirical software engineering. World Scientific Publishing Co., Inc. 2003. pp. 229-263.

 ISBN: 978-1-61208-388-9

[2] D. Abadi, “Data management in the cloud: limitations and opportunities. IEEE Data Engineering Bulletin, vol 32(1). pp 3-12. 2009.

[3] L. Chongxin, “Transforming relational database into HBase: a case study”. in Software Engineering and Service Sciences (ICSESS), IEEE International Conference on. 2010. pp 683-687.

[4] M. Stonebraker, S. Madden, D. Abadi, S. Harizopoulos, N. Hachem, and P. Helland, “The end of an architectural era: (it's time for a complete rewrite)”, in Proceedings of the 33rd international conference on very large data bases, VLDB Endowment: Vienna, Austria. 2007. pp. 1150-1160.

[5] M. Stonebraker, D. Abadi, D. DeWitt, S. Madden, E. Paulson, and A. Pavlo, “MapReduce and parallel DBMSs: friends or foes?” Commun. ACM, vol. 53(1). pp. 64-71. 2010.

[6] M. Stonebraker, “Technical perspective: One size fits all: an idea whose time has come and gone”. Commun. ACM. Vol. 51(12). pp. 76-76. 2008.

[7] J. Gantz and D. Reinsel, “Extracting Value from Chaos”, in EMC Corporation, E. Corporation, Editor. pp. 12-24. 2011.

[8] M. Stonebraker, “The case for shared nothing”. A quarterly bulletin of the IEEE computer society technical committee on database engineering, vol. 9(1). pp. 4-9. 1986.

[9] H. Boral and D. J. DeWitt, “A methodology for database system performance evaluation”. SIGMOD Rec.,vol. 14(2): pp. 176-185. 1984.

[10] J. Varia, “Migrating your existing applications to the AWS cloud”, Amazon Web Services. 2010.

[11] G. Demarest and R. Wang, “Oracle cloud computing”, Oracle, Editor, Oracle Redwood Shores, CA. 2010.

[12] H. Goto, Y. Hasegawa, and M. Tanaka, “Efficient scheduling focusing on the duality of MPL representatives,” Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.

[13] D. Salmen, T. Malyuta, R. Fettersand, and A. Norbert, “Cloud data structure diagramming techniques and design patterns”. 2009.

[14] P. Singh, “Schema guidelines & case studies. 2010.

[15] T. Singh and P. S. Sandhu, “Cloud computing databases: latest trends and architectural concepts”. Proceedings of World Academy of Science, Engineering and Technology. vol. 73(Compendex). 2011. pp. 1042-1045.

[16] A. Jedlitschka and D. Pfahl. “Reporting guidelines for controlled experiments in software engineering”. in Proceedings of the 4th International Symposium on Empirical Software Engineering (ISESE 2005). IEEE Computer Society. 2005. Pp. 95-104.

[17] N. Juristo and A. M. Moreno, “Basics of software engineering experimentation”. Springer Publishing Company, Incorporated. 2010.

[18] C. Wohlin, M. Höst, and K. Henningsson, “Empirical research methods in software engineering”, in Empirical Methods and Studies in Software Engineering, R. Conradi and A. Wang, Editors. Springer Berlin Heidelberg. 2003. pp. 7-23.

[19] S. Easterbrook, J. Singer, M. A. Storey, and D. Damian, “Selecting empirical methods for software engineering research guide to advanced empirical software engineering”, in Guide to Advanced Empirical Software Engineering, F. Shull, J. Singer, and D. Sjøberg, Editors, Springer London. pp. 285-311. 2008.

[20] E. Marcos, “Software engineering research versus software development”. SIGSOFT Softw. Eng. Notes, vol. 30(4). 2005. pp. 1-7.

[21] M. Kasunic, “Designing an effective survey”. Carnegie Mellon Software Engineering Institut. 2005.

[22] T. C. Lethbridge, “A survey of the relevance of computer science and software engineering education”, in Proceedings of the 11th Conference on Software Engineering Education and Training. IEEE Computer Society. 1998. pp. 56-66.

 ISBN: 978-1-61208-388-9