EURA NOVA

Change the game with smart innovation

Master Thesis 2014 – 2015 Faculties of Science & Engineering

17/02/2014

Master Thesis proposal for the academic year 2014-2015.

TABLE OF CONTENTS

SECTION ONE

INTRODUCTION

EURA NOVA R&D

EURA NOVA is a Belgian company founded in September 2008. Our mission is simple: « Being a technologic incubator focusing on the pragmatic use of knowledge ».

Our research activities are linked to technologic directions and concrete short-term, medium-term and long-term opportunities. EURA NOVA dissociates career management from the client relationship, more in specific by basing its vision on an entrepreneurial perception of career planning.

This document presents master theses supervised by our Research and Development department. The student will work in close cooperation with the research engineers and will be invited to share his work through the in house EURA NOVA knowledge management tool.

SECTION TWO

MASTER THESIS 2014

DATA STORAGE

EVOLUTION OF STORAGE SYSTEMS AND CONVERGENCE OF RELATIONAL DATABASES AND NOSQL: STUDY, MODELING AND BENCHMARK

Context: NoSQL is a large range of technologies and techniques that has been developed by the internet community to meet their scalability requirements. They made a deliberate choice to give up the transactional aspect in return of exceptional writing performances, an efficient distribution of reads and most of all storage elasticity, and this by respecting two of the three properties of the CAP theorem also known as Brewer's theorem. But today, the situation has changed and the context of the theorem has evolved [1]. The main Web actors realize that even in the case of social applications, real time and strong consistency have become crucial elements. In addition, NoSQL has been largely adopted by the community and the need to implement consistency functionalities at application level has gained importance. This is why NoSQL databases have recently evolved towards distributed architectures allowing the execution of transactions via typical leader election mechanisms. This is described by El-Abadi as *data fusion* [2]: the evolution of NoSQL structures towards RDBMS functionalities. Examples are Google Megastore [3] or G-Store [5].

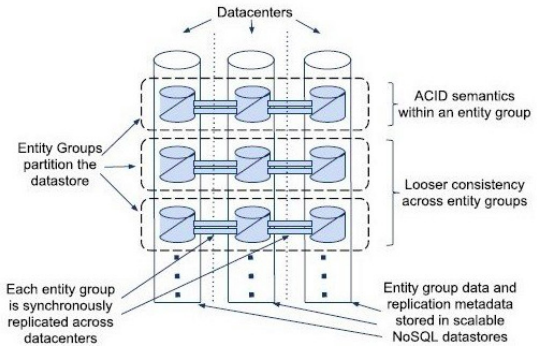


Figure 1 the Google Megastore architecture[3]

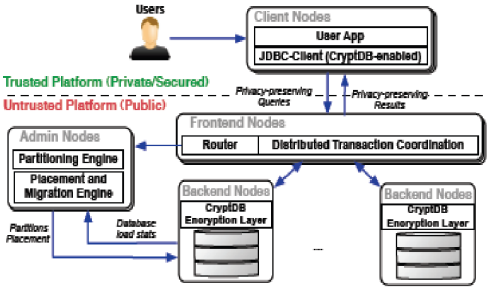


Figure 2 the MIT RelationalCloud architecture [4]

Under the pressure of NoSQL, relational databases have been gaining in performance and elasticity. The two most important evolutions are column-oriented databases and the evolution towards NoSQL functionalities. The latter has been described by El-Abadi as *data fission* [2]. Examples are Elastras, CloludSQL Server of Microsoft research or RelationalCloud of MIT.

The goal of this master thesis is to study these two major evolutions, data fission and data fusion. Besides analysing and modelling their architecture, it also aims to model their elastic behaviour, based upon research done by EURA NOVA in cooperation with the ULB and the UCL.

Contribution: The purpose of this master thesis is threefold: (1) study and model the new databases including Elastras, RelationalCloud, Spanner, Megastore and G-Store; (2) deduce their elastic nature by defining or adapting the existing models; and finally (3) validate and compare their behaviour by setting up a benchmark.

Organisation: This master thesis is organised in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

MODEL ENGINEERING

IMPLEMENTATION OF A DIAGRAM CREATION AND MANIPULATION TOOL REPRESENTING A BUSINESS MODEL

Context: Among the rich applications, there are a lot of tools allowing to create and manage all kinds of diagrams. These diagrams are often representations of an underlying business model containing a set of data that has to be represented in one way or another (process, statistics, requests, models...).

At Eclipse level, the EMF technologies make the editing of data models quite easy. Libraries such as Graphiti provide a set of graphic representations for these models. Besides visualization, they also allow the graphical creation and manipulation of these models. This graphic editing software enables the non-technical users to manipulate these models.

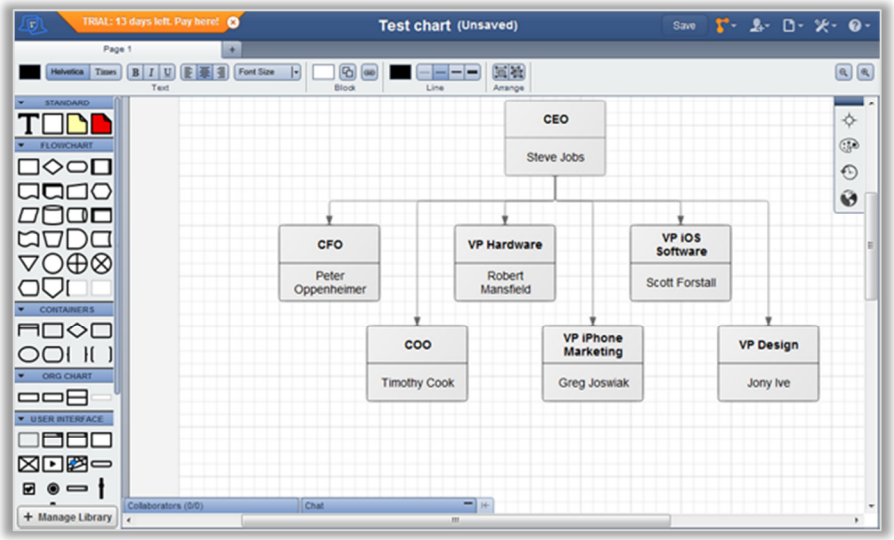


Figure 3 Most of the existing frameworks are whether owner, whether solution oriented, such as a full BPMN solution, but none of them addresses the issue of a generic modeller.

Some proprietary solutions exist but they only meet very specific needs. An equivalent tool that allows to directly edit the graphic models in a web navigator is not available. However, there are libraries enabling the visualization of data such as D3 or JIT (Javascript Infoviz Toolkit), as well as libraries able to link an underlying (potentially remotely stored)

model to its visualization (e.g.: backbone.js). But you will not find a complete tool allowing the real-time editing of models.

Such a tool would enable non-technical users to elaborate models in real time without requiring any software installation on their workstation.

Contribution: This master thesis will consist of two parts:

1. The first part will study the different data representation technologies, as well as the web development environment solutions (Orion, Cloud9, ...)

2. The second part will address the implementation of a tool allowing the visualisation of a remotely stored model as well as its real-time manipulation within a navigator, by using the most relevant technologies.

Organisation: This master thesis is organised in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

MACHINE LEARNING

DESIGN OF A LARGE SCALE IMAGE SEARCH ENGINE

Context: Image search by content provides the ability to search images through visual queries (image query), for example when you have a picture for which you would like to obtain visually similar images. A system of image search typically consists in two parts. The first part is to describe the image. This description can be:

- Textual: in this case the image is associated with a set of words (annotations) that describe it.
- Visual: it is using recognition techniques of patterns to extract important visual characteristics of images: texture, colour, shape.

The second part is to index the image descriptors (i.e. characteristic vectors) in a way that we can quickly find similar images to a given query image based on these descriptors.

In the literature, several systems of image research have been proposed [7]. However, most of these systems are not adapted to the Big Data context. Indeed, the social network users currently publish huge amounts of images, up to terabytes per day, that require new techniques of images search by content.

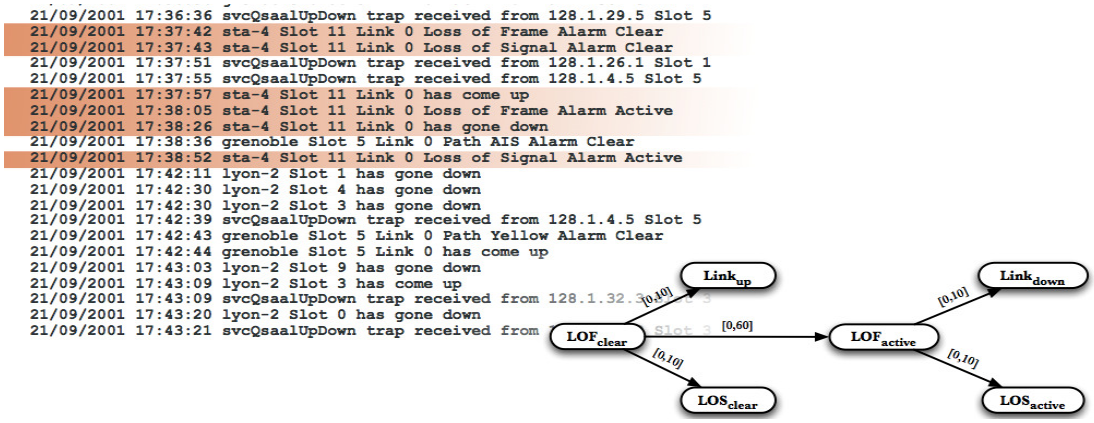
Contribution: The aim of this master thesis is:

1. To study state-of-the-art techniques of images indexation.
2. To propose a suitable solution for large image databases.
3. To optimize extraction techniques of images characteristics suited for devices with low computing power (i.e. smartphone).
4. To design a distributed and elastic architecture of the system.

Organisation: This master thesis is organised in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

CHRONICLES

CHRONICLE MINING



Context: A Chronicle [11, 12] is a representation of an interesting situation. In particular, it is composed of a set of timepoints separated by time intervals (timepoint 0 has to happen between 2 and 5 minutes before timepoint 1), and a set of events expected at some time points.

Chronicles are used to describe situations that a system should be monitored for; a chronicle engine takes the form of an event stream processor that takes as input raw observations on a system and emits as output the occurrences of interesting situations.

Before the system can detect a chronicle, the chronicle has to be defined. This is mostly done by experts, but should as much as possible be automated. As the computer will probably not be able to really grasp the concept of "interesting situation", we would rather aim for a collaborative tool between human expert and computer. For example, the computer would suggest a set of patterns, and the expert would then choose among them.

This opens the door to chronicle mining, which can be based on a number of existing data mining techniques, such as Petri-net based process mining [13], frequent itemset discovery [14] or trajectory mining [15].

Contribution: The goals of the thesis are:

1. Study the state of the art in Petri-nets based mining and assess their applicability to chronicle mining.

2. Study the state of the art in frequent itemset discovery, and assess their applicability to chronicle mining.

3. Study the state of the art in trajectory mining, and assess their applicability to chronicle mining.

4. Implement a prototype of a chronicle miner.

Organisation: This master thesis is organised in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

GRAPH PROCESSING

DISTRIBUTED GRAPH PROCESSING USING GPU ON LARGE GRAPHS

Context: The main advantage of social data is its ability to aggregate information as graph and to exploit the structural relationship. Nowadays we have collected significant graphs and we are now looking at new techniques in order to achieve efficient machine learning and graph analytics on them. However, today the distributed graph processing techniques as Pregel [1] show two main drawbacks: (1) the implementation of the learning algorithm must fit the signal and receive model and (2) the Hadoop Map Reduce [2] nature involves batch processing. This last point could represent a problem in term of processing time in certain critical scenarios.

However, new interesting techniques emerged as Pegasus[3], which proposes to use the GIM-V (Generalized Iterated Matrix-Vector multiplication) within Map reduce jobs for a set of interesting computations such as radius, random walks, page rank, centrality measures, etc.

In the other hand, the GPU computation has become a really efficient tool for performing high end computation on matrix and vectors. Recently we have started a research project, at EURA NOVA, aiming at exploiting the distributed aspects of GPU computation within the AROM[4] framework and the SNIFF project developed by EURA NOVA

The Pegasus approach would be an elegant approach for processing graph within GPU using a distributed approach. In this work we propose to study the Pegasus algorithms and to migrate them to a distributed GPU framework.

Contribution: The contributions of the thesis are the following:

1. State of the art of distributed graph processing frameworks
2. Study the SNIFF and AROM frameworks
3. Study the Pegasus approach for using the GIM-V and analyzing their usage for 5 algorithms
4. Define how they can be implemented within the SNIFF framework

5. Write a STEFFI (distributed in-memory graph DB) plugin for implementing the graph processing algorithm within the DB.

Organisation: This master thesis is organised in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

VIEWS CREATION AND MAINTENANCE ON TOP OF GRAPHS

Context: In traditional databases, views are constructed on top of tables (1) for security reasons as to restrict access, or (2) to accelerate and optimize read operations on large tables, and avoid the costly join operations. Views are of two types materialized and non-materialized views.

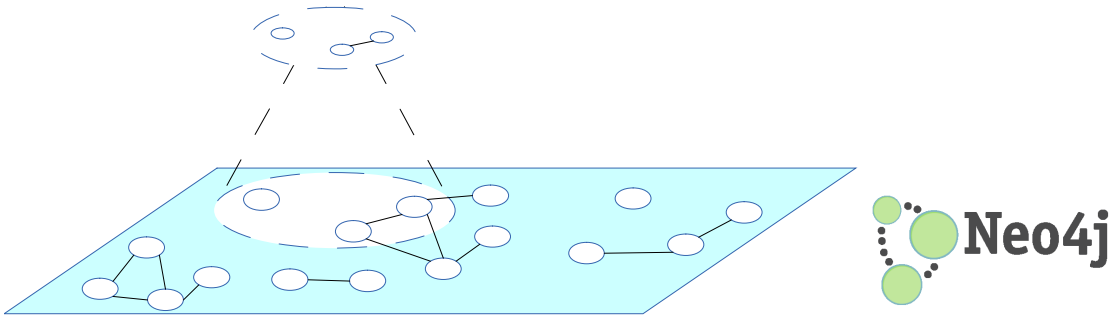
Building views on top of semi-structured and unstructured data has been investigated in the case of object-oriented, XML and RDF data [15, 16, 17].

However, views on top of native graphs remain to be investigated and implemented in current graph databases.

The goal of this master thesis is to study views on a graph databases setting, and to provide answers to the following question: How to define and build views on graph databases?

Views will enable optimized and controlled access on large graphs. The approach to propose should include answers to the following points: (1) is a view just a set of nodes, or does it also include the edges between the nodes? (2) in a materialized view, what should the ID of the copy of a node be, and how can we identify the original element? (3) how should the materialized copy of a node be linked to its original element?

An extension to this work would be by tackling the problem of incremental maintenance of views as the original data is modified.



Contribution:

1. Study the state of the art on graph DB, object-oriented DB and RDF DB, views creation and maintenance

2. Propose a graph oriented approach for building views on top of graph data

Organisation: This master thesis is organized by the ULB in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

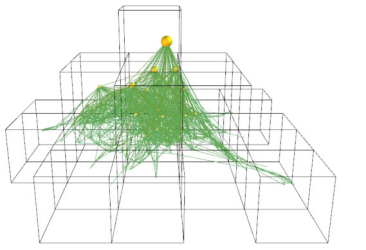
EVOLVING GRAPH CUBES: CROSS-CUBES ANALYSIS

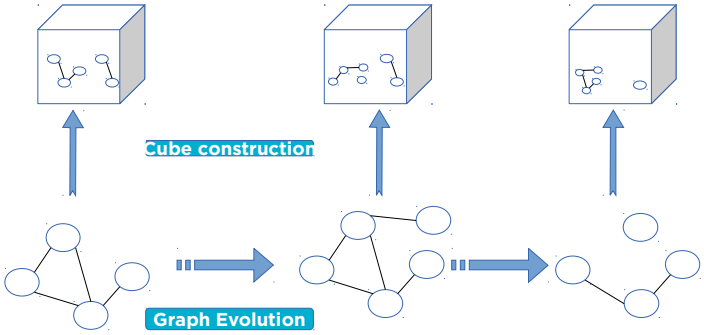
Context: In relational data warehousing systems, data is collected and aggregated from operational databases. The data stored in the data marts is then updated periodically to guarantee the consistency with the sources. OLAP Cubes are built on top of the data warehouse to provide a multi-level, multi-perspective view of the underlying data. The cubes are an important part of the analytics dashboard, and allow business analysts to perform complex querying of their data following a fact/dimension dichotomy.

As data keeps flowing, new dimensions, measures and facts could appear. Moreover, the dimensions hierarchy could be modified. To keep an up-to-date view of the underlying data while the data marts evolve, OLAP cubes need to be updated and their measures recomputed [18].

In the case of graph data, we have the same issues. The graph data warehouse stores the graph, along with its evolution trace. Graph-based OLAP Cubes are built on top of the graph data, and they contain both numeric and structural measures [19]. The graph evolution is subject to multiple factors, time-evolving is just a sub-case [20]. So we should be able to support the analysis across multiple axis such as time, temperature, pressure etc.

The querying of evolving graph cubes is a critical part of the study of the underlying dynamic graphs [21, 22]. This will help analysts discover hidden association between the actors of the network, and predict the evolution of the network.





Graph-Cubes

Spark
Lightning-Fast Cluster Computing

HDFS

Contribution: The challenges here are to:

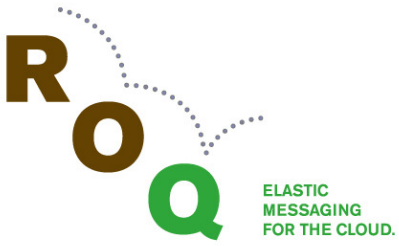
1. Study the best scenarios for the cubes update, and compare them.
2. Automate the update process, between the graph data and the cubes.

3. Model the cubes evolution over one axis (factor), and study the techniques for cross-cubes analysis.

4. Extend 3. to support OLAP analysis over multiple axis at-a-time.

Organisation: This master thesis is organized by the ULB in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

ELASTIC MESSAGING - RoQ



RoQ (pronounce /rɒkˈjuː/, as in "Rock You") is the first implementation of EQS [REFERENCE TO EQS], a new architecture designed for efficient messaging in the cloud. Traditional Message Oriented Middlewares (MOMs) are not designed to support elastic scaling. This means that in a cloud context, they may very quickly become a bottleneck in terms of performance. RoQ has been designed from day 1 to answer this problem. Its architecture is elastically scalable. This includes three properties:

1. When required, the capacity of the system will be increased automatically
2. This capacity increase has no impact on the global performance
3. When the load decreases, the system will scale down to avoid using unnecessary ressources

With its distributed architecture and its ability to be quickly deployed on a cloud, RoQ is designed to ease the usage of MOM in a cloud context.

RoQ started in 2011 as a research project within EURA NOVA and has since then evolved into an open-source project. This means that anyone can download the code, see how it works, contribute or even fork the code to build upon RoQ.

ROQ QOS MANAGEMENT AND PERSISTENCE

Context: In the current implementation of RoQ the messages being transmitted are transient, meaning that they are not persisted on a storage medium. A subscriber who misses one message cannot retrieve it later. Also there is no way to retrieve the history of the messages which have been sent. While some software designs may cope with these limitations (see http://www.imatix.com/articles:whats-wrong-with-amqp), many applications require these features to correctly work.

Contribution: The aim of the thesis is to provide the study,design and implement the following requirements to RoQ:

1. **Event persistence:** every message transiting on RoQ must be persisted within the topic (key) on which it has been sent.

2. **Delivery guarantee:** every subscriber must receive all events from the topic it listens to. The client library must be able to check whether it has received all of them and in case of missed messages, it must request them.

3. **Reconnected subscriber:** if a subscriber disconnect from the topic it must be able to either (1) receive all the events from the beginning of the topic, (2) receive all the events from the last connection or (3) receive no historical events and just starts listening new ones. For the case (2) we take as assumption that the client must keep the state of its last connection (last message sequence ID for instance).

Organisation: This master thesis is organised in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

HIGH AVAILABILITY FOR ROQ CORE COMPONENTS

Context: In the current implementation of RoQ, the core elements (exchange, queue managers ,...) are not high available, meaning that their failure can lead to the failure of the entire service. The goal of this master thesis is (1) to study the architecture patterns of high availability components in existing distributed systems (e.g. Zookeeper, Hadoop, ...) and to propose a high availabilitydesign for RoQ.

In the case a RoQ core component fails, the following use case should be considered:

1. fire the right alarms and notify the system of the crash
2. electing a new active components between the set of standbye ones
3. the newly-elected component must get the state of the previous active, re-open all connection sockets
4. all the dependent components (those which maintained a communication socket with the crashed-component) must be re-configured to connect the new elected one

Contribution: The master thesis should (1) provide an exhaustive study of the architectural patterns for high availability in distributed systems and (2) propose a design and implementation for RoQ.

Organisation: This master thesis is organised in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

IMPROVEMENT OF THE TOPIC-BASED DISTRIBUTION

Context: In the current architecture, the subscriber must connect to all exchanges and they are responsible for the filtering of the topic, which enable to move the routing intelligence to the subscriber. This Master Thesis aims at implementing all the features required to partition the subscriber space per topic, and let the subscriber only connect the exchange they need to connect.

Contribution: The goal of the Master Thesis is to propose a topic space distribution strategy for RoQ. The decrease in ports wasting can come as a side contribution (https://github.com/roq-messaging/RoQ/issues/138)

Organisation: This master thesis is organised in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

ROQ AS AN ELASTIC DISTRIBUTION PATTERN PROVIDER

Context: RoQ is currently not a message queue in its state. It provides elasticity on pub-sub message distribution pattern. The goal is to propose other message distribution patterns (see zmqguide) as service.

Contribution: The Master Thesis will study the message distribution patterns and how they can be provided elastically using the RoQ architecture, including proposing new distribution patterns on top of the existing implemented PUB-SUB pattern.

Organisation: This master thesis is organised in collaboration with EURA NOVA R&D. The student will be supervised by the EURA NOVA R&D team.

REFERENCES

[1] E. Brewer, *CAP twelve years later: How the "rules" have changed,* in IEEE Computer Journal vol. 45, 2012.

[2] D. Agrawal and al., *Database Scalability, Elasticity, and Autonomy in the Cloud*, in the Proceedings of the 16th international conference on Database systems for advanced applications - Volume Part I. 2011

[3] J. Baker and al., *Megastore: Providing Scalable, Highly Available Storage for Interactive Services*, in Proceedings of the Conference on Innovative Data system Research (CIDR) (2011), pp. 223-234. 2011

[4] C. Curino and al., Relational Cloud: a Database Service for the cloud, in CIDR 2011.

[5] S. Das and al., *G-Store: a scalable data store for transactional multi key access in the cloud*, in the Proceedings of the 1st ACM symposium on Cloud computing. 2010

[6] NL. Tran and al., *AROM: Processing Big Data With Data Flow Graphs and Functional Programming*, in the CRC workshop of the 4th IEEE International conference on Cloud Computing technology and Science. 2012

[7] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(12):1349-1380, 2000

[8] Koen E. A. van de Sande, Theo Gevers, Cees G. M. Snoek: Evaluating Color Descriptors for Object and Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9): 1582-1596 (2010)

[9] Aurélien Bellet, Amaury Habrard, Marc Sebban: Good edit similarity learning by loss minimization. Machine Learning 89(1-2): 5-35 (2012)

[10] Salim Jouili, Salvatore Tabbone: Hypergraph-based image retrieval for graph-based representation. Pattern Recognition 45(11): 4054-4068 (2012)

[11] C. Dousson. Suivi d'évolutions et reconnaissance de chroniques. Rapport LAAS. 1994. URL: ftp://ftp.laas.fr/pub/ria/theses/christophe_dousson_16_09_94.pdf

[12] C. Dousson: Monitoring with Chronicles. (Slides) http://mklab.iti.gr/events2010/sites/default/files/2010%20-%20Events%20-%20Chronicles. pdf

[13] Toon Calders: Data mining for local patterns. EBISS 2013. URL: http://cs.ulb.ac.be/conferences/ebiss2013/files/calders_ebiss2013.pdf

[14] Wil van der Aalst: Process Mining: Making Sence of Processes Hidden in Big Event Data. EBISS 2013. URL: http://cs.ulb.ac.be/conferences/ebiss2013/files/vdaalst_ebiss2013.pdf

[15] Zhuge, Yue, and Hector Garcia-Molina. "Graph structured views and their incremental maintenance." Data Engineering, 1998. Proceedings., 14th International Conference on. IEEE, 1998.

[16] Abiteboul, S., McHugh, J., Rys, M., Vassalos, V., & Wiener, J. (1998). Incremental maintenance for materialized views over semistructured data.

[17] Etcheverry, L., & Vaisman, A. A. (2012). Views over RDF Datasets: A State-of-the-Art and Open Challenges. arXiv preprint arXiv:1211.0224.

[18] Wrembel, R. (2009). A survey of managing the evolution of data warehouses. IJDWM, 5(2), 24-56.

[19] P. Zhao, X. Li, D. Xin, and J. Han, "Graph Cube: On Warehousing and OLAP Multidimensional Networks," in SIGMOD '11 Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, 2011, pp. 853–864.

[20] Fard, Arash, et al. "Towards efficient query processing on massive time-evolving graphs." Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on. IEEE, 2012.

[21] Ren, C., Lo, E., Kao, B., Zhu, X., Cheng, R.: On querying historical evolving graph sequences. Proceedings of the VLDB Endowment 4(11), 726–737 (2011)

[22] Udayan Khurana, Amol Deshpande: Efficient snapshot retrieval over historical graph data. ICDE 2013: 997-1008

[23] Aufaure, Marie-Aude, et al. "Predicting your next OLAP query based on recent analytical sessions." Proceedings of the 15th International conference on data warehousing and knowledge discovery (DaWaK 2013). 2013.

[24] Sapia, Carsten. "PROMISE: Predicting query behavior to enable predictive caching strategies for OLAP systems." Data Warehousing and Knowledge Discovery. Springer Berlin Heidelberg, 2000. 224-233.

[25] P. Zhao, X. Li, D. Xin, and J. Han, "Graph Cube: On Warehousing and OLAP Multidimensional Networks," in SIGMOD '11 Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, 2011, pp. 853–864.

[26] Benoît Denis, Amine Ghrab and Sabri Skhiri" A Distributed Approach for a Graph-Oriented Multidimensional Analysis". To appear in KMBA workshop of The IEEE International Conference on Big Data 2013, October 2013.

[27] Salim Jouili and Aldemar Reynaga, imGraph: A distributed in-memory graph database. To appear in Proceedings of the 2013 ASE/IEEE International Conference on Big Data, Washington D.C., USA, September 2013.

[28] Ghrab, A., Skhiri, S., Jouili, S., & Zimányi, E. (2013). An Analytics-Aware Conceptual Model for Evolving Graphs. In Data Warehousing and Knowledge Discovery (pp. 1-12). Springer Berlin Heidelberg.

[29] Malewicz, Grzegorz, et al. "Pregel: a system for large-scale graph processing."Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010.