

EURV
NOVA

Change the game with smart innovation

Master Thesis & Internship Proposals

Faculties of Science and Engineering

EURV NOVA
Research & Development
Academic year 2014-2015
3rd March 2014

CONTENTS

Introduction	3
Master Thesis 2014	4
Data Storage	4
Evolution of Storage Systems and Convergence of Relational Databases and NoSQL	4
Chronicles	5
Chronicle Mining	6
Mining User Profiles on a Web Portal with Multidimensional Trajectories De- fined by Chronicles (in collaboration with Yahoo!).	7
Graph Processing	8
Master Data Management with Graphs	8
Design and Implementation of a Graph Database Management Framework	9
Elastic Messaging - RoQ	10
RoQ QoS Management and Persistence	11
High Availability for RoQ Core Components	12
Improvement of the Topic-Based Distribution	13
RoQ as an Elastic Distribution Pattern Provider	14
References	15

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 shortterm, mediumterm 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.

Unless stated otherwise, all the master theses and internships presented in this document are organised by EURA NOVA R&D in collaboration with the ULB. The student will be supervised by the EURA NOVA R&D team.

MASTER THESIS 2014

DATA STORAGE

EVOLUTION OF STORAGE SYSTEMS AND CONVERGENCE OF RELATIONAL DATABASES AND NOSQL

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. 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 [AEADE11]: the evolution of NoSQL structures towards RDBMS functionalities. Examples are Google Megastore [BBC⁺11] or G-Store [DAEA10].

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 [AEADE11]. Examples are Elastras, CloudeSQL 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.

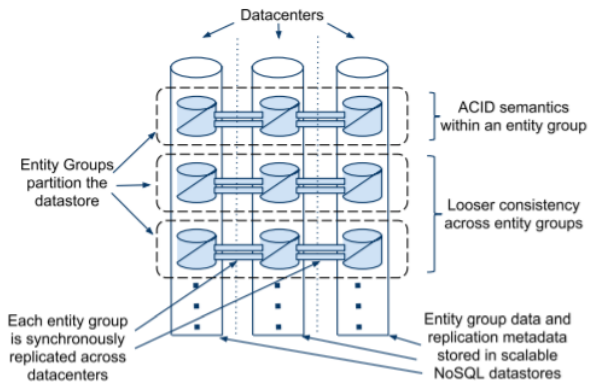


Figure 1: The Google Megastore Architecture

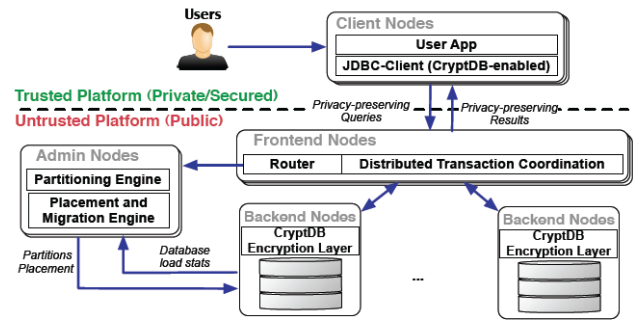


Figure 2: The MIT Relational Cloud Architecture

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.

CHRONICLES

```

21/09/2001 17:36:36 svcQsaalUpDown trap received from 128.1.29.5 Slot 5
21/09/2001 17:37:42 sta-4 Slot 11 Link 0 Loss of Frame Alarm Clear
21/09/2001 17:37:43 sta-4 Slot 11 Link 0 Loss of Signal Alarm Clear
21/09/2001 17:37:51 svcQsaalUpDown trap received from 128.1.26.1 Slot 1
21/09/2001 17:37:55 svcQsaalUpDown trap received from 128.1.4.5 Slot 5
21/09/2001 17:37:57 sta-4 Slot 11 Link 0 has come up
21/09/2001 17:38:05 sta-4 Slot 11 Link 0 Loss of Frame Alarm Active
21/09/2001 17:38:26 sta-4 Slot 11 Link 0 has gone down
21/09/2001 17:38:36 grenoble Slot 5 Link 0 Path AIS Alarm Clear
21/09/2001 17:38:52 sta-4 Slot 11 Link 0 Loss of Signal Alarm Active
21/09/2001 17:42:11 lyon-2 Slot 1 has gone down
21/09/2001 17:42:30 lyon-2 Slot 4 has gone down
21/09/2001 17:42:30 lyon-2 Slot 3 has gone down
21/09/2001 17:42:39 svcQsaalUpDown trap received from 128.1.4.5 Slot 5
21/09/2001 17:42:43 grenoble Slot 5 Link 0 Path Yellow Alarm Clear
21/09/2001 17:42:44 grenoble Slot 5 Link 0 has come up
21/09/2001 17:43:03 lyon-2 Slot 9 has gone down
21/09/2001 17:43:09 lyon-2 Slot 3 has come up
21/09/2001 17:43:09 svcQsaalUpDown trap received from 128.1.32.3 Slot 3
21/09/2001 17:43:20 lyon-2 Slot 0 has gone down
21/09/2001 17:43:21 svcQsaalUpDown trap received from

```

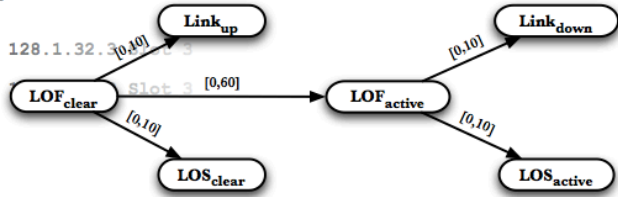


Figure 3: Illustration of a chronicle, as defined in [Dou94].

A Chronicle [Dou94] 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.

CHRONICLE MINING

Context: 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, frequent itemset discovery or trajectory mining [ZGM97]...

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.

MINING USER PROFILES ON A WEB PORTAL WITH MULTIDIMENSIONAL TRAJECTORIES DEFINED BY CHRONICLES (IN COLLABORATION WITH YAHOO!).

Context: Through the analysis of the navigation behaviour of a user's visit on an internet portal such as Yahoo!, it is possible at the same time to model her profile. Indeed, we can define the path taken by a user through her journey on the portal as a spatio-temporal trajectory in a multidimensional space. It is then possible to apply mining techniques on these trajectories (such as clustering, pattern extraction, etc...) in order for instance to predict the next move of the user and recommend relevant pages and associated information.

On the other hand, Chronicle [Dou94] is an existing paradigm aimed at formalizing interesting situations. In particular, a Chronicle 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.

Business Opportunity: User modeling is a common practice used to customize and adapt a service to meet user's specific needs. On the internet it is extensively implemented by services and shopping websites for example in order to tailor the displayed ads and recommend products to users. More precise recommendations leads to increased revenues and a provides a better experience for the user.

Contribution: The goal of this master thesis is to model the user's trajectories on a web portal by using the chronicle paradigm. The student will first review the existing litterature on trajectory mining (not only geographical). The next goal will then be to study and extend EURA NOVA's current work on Chronicles to take into account the definition and mining of trajectories.

GRAPH PROCESSING

MASTER DATA MANAGEMENT WITH GRAPHS

Context: The graph model is generic and flexible. It puts equal focus on data and network topology. It is naturally suitable for modeling domains having complex structural relationship such as biological, social and information networks. Extracting the complex information hidden in these various networks is made possible thanks to the solid mathematical background and algorithms developed on graphs.

On the other side Master data provides a single unified view of an organization 's critical data. This data is either scattered across the IT infrastructure, or added through mergence or acquisitions of organizations. Master Data Management solutions are then used in industry as middle-wares to integrate multiple data management systems and provide a unified view atop. They provide the mapping and matching between critical entities shared across the organization's infrastructure.

Business Opportunity: Current data models such as XML, relational ,RDF and object-oriented could be generalized using the generic structure of graphs. Designing an MDM using graphs enables (1) the integration of heterogenous data structures, (2) complex and rich relationships semantics such as compositions and hierarchies, and (3) graph-oriented analysis scenarios, such pattern discovery and paths and neighborhood exploration.

Contribution: The goal of this master thesis is to design an MDM framework that enables:

1. Modeling of complex relationships between master data across the organization
2. Integration and provides unified view of the organization master data
3. Caching and Querying of master data for graph analysis scenario

DESIGN AND IMPLEMENTATION OF A GRAPH DATABASE MANAGEMENT FRAMEWORK

Context: Multiple graph data models have been proposed in the literature, but most of them ignore at least one of the three components of a complete database model defined as a set of: (1) data structures, (2) integrity constraints, and (3) operators. Current graph databases, such as Neo4j and Titan are built for native graph management [AG08].

Native graph database models have the advantage of targeting the graph structure at both the modeling and querying stages. However, the data structures introduced by current graph databases are simple and oriented for operational workload. Moreover, integrity constraints are not specified. Common graph query languages, such as Gremlin and Cypher, have been proposed by these graph databases. However, no algebra has formally been defined.

At EURA NOVA, in cooperation with ULB and UPC, we have designed a novel graph database model [GSJZ13]. The model is analysis-oriented, enriched with integrity constraints to guarantee consistency, and proposes an algebra for native querying of graph structures.

Business Opportunity: Graphs are flexible data structures, used for modeling and solving diverse domains. Social networks representation, fraud detection, recommendation, routing and optimization are typical graph applications. Graph analysis offers valuable insights on the data while being more user-friendly and less error prone. Moreover, native graph databases provide better performance and reduced response time, which lowers the development and production costs.

Contribution: The goal of this master thesis project is to implement the model at the physical level, on top of current graph databases engines.

1. State-of-the-art on current commercial graph databases.
2. Design of integrity constraints checking mechanisms.
3. Implementation of the algebra algorithms.
4. Design of the graph querying DSL.

ELASTIC MESSAGING - ROQ



RoQ is the first implementation of EQS [TSZ11], 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 resources.

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 EURL 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 disconnects 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).

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 availability design 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. elect a new active components between the set of standby ones.
3. the newly-elected component must get the state of the previous actives and re-open all connection sockets.
4. all the dependent components (those which were maintaining a communication socket with the crashed-component) must be re-configured to connect the newly 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.

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>)

ROQ AS AN ELASTIC DISTRIBUTION PATTERN PROVIDER

Context: RoQ is currently not a message queue in its state. It provides elasticity on the PUB-SUB message distribution pattern. The goal here is to propose other elastic message distribution patterns (see the zmqguide) as a 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 PUB-SUB pattern.

REFERENCES

- [AEADE11] Divyakant Agrawal, Amr El Abbadi, Sudipto Das, and Aaron J. Elmore. Database scalability, elasticity, and autonomy in the cloud. In *Proceedings of the 16th International Conference on Database Systems for Advanced Applications - Volume Part I, DASFAA'11*, pages 2--15, Berlin, Heidelberg, 2011. Springer-Verlag.
- [AG08] Renzo Angles and Claudio Gutiérrez. Survey of graph database models. *ACM Comput. Surv.*, 40(1), 2008.
- [BBC+11] Jason Baker, Chris Bond, James C. Corbett, Jj Furman, Andrey Khorlin, James Larson, Jean-Michel Leon, Yawei Li, Alexander Lloyd, and Vadim Yushprakh. Megastore: Providing scalable, highly available storage for interactive services, 2011.
- [DAEA10] Sudipto Das, Divyakant Agrawal, and Amr El Abbadi. G-store: A scalable data store for transactional multi key access in the cloud. In *Proceedings of the 1st ACM Symposium on Cloud Computing, SoCC '10*, pages 163--174, New York, NY, USA, 2010. ACM.
- [Dou94] Christophe Dousson. *Suivi d'évolutions et reconnaissance de chroniques*. PhD thesis, 1994.
- [GSJZ13] Amine Ghrab, Sabri Skhiri, Salim Jouili, and Esteban Zimányi. An analytics-aware conceptual model for evolving graphs. In Ladjel Bellatreche and MukeshK. Mohania, editors, *Data Warehousing and Knowledge Discovery*, volume 8057 of *Lecture Notes in Computer Science*, pages 1--12. 2013.
- [TSZ11] Nam-Luc Tran, S. Skhiri, and E. Zimanyi. Eqs: An elastic and scalable message queue for the cloud. In *Cloud Computing Technology and Science (Cloud-Com), 2011 IEEE Third International Conference on*, pages 391--398, Nov 2011.
- [ZGM97] Y. Zhuge and H. Garcia-Molina. Graph structured views and their incremental maintenance (full version). Technical Report 1997-16, Stanford InfoLab, 1997.