ERCIM "ALAIN BENSOUSSAN"
FELLOWSHIP PROGRAMME

ERCIM
European Research Consortium
for Informatics and Mathematics

Scientific Report

First name / Family name	Gautier Berthou
Nationality	French
Name of the *Host Organisation*	SICS
First Name / family name of the *Scientific Coordinator*	Jim Dowling
Period of the fellowship	01/04/2014 to 01/04/2015

I – SCIENTIFIC ACTIVITY DURING YOUR FELLOWSHIP

The main research activities followed during my ERCIM "Alain Bensoussan" fellowship programme integrated themselves in the End-to-End Clouds research project. The End-to-End Clouds research project aims at addressing issues in the intersection between Big Data platforms, multi-clouds and network. My work in this project focussed on the Hadoop Open Platform-as-a-Service (Hops), a scalable file system and computation framework for storing and analysing Big Data.

Hops is based on Hadoop, the de-facto standard framework for Big Data, and aims at solving Hadoop limitations in regards to scalability and customisability. In fact, even though Hadoop was designed to handle a large quantity of data and store those on a large number of nodes, two main components of the Hadoop framework are bottlenecks and single points of failure: the *name node* and the *resource manager*.

The NameMode is part the Hadoop File System (HDFS). When a file is stored in HDFS it is divided in blocks that are stored in the DataNodes present in the cluster. The metadata of this file (the file name, the owner of the file, the data nodes containing the blocks, etc.) is then stored in the name node. The scalability issue arises from the fact that all the metadata has to be stored in the heap of a single name node. Even if a modern machine can be equipped with large amount of memory, companies such as Yahoo and Spotify have reached the limit of the heap size. When the heap size is too big, the java garbage collector is slow and can freeze the cluster for tens of minutes. Furthermore all the operation on files must go through the NameNode. This puts a lot of load on the name node and limits the performance of the file system.

The resource manager is part of Hadoop YARN. Hadoop Yarn is the scheduler of Hadoop; it distributes computation tasks to nodes in the cluster and supports multiple data processing engines, such as interactive SQL, real-time streaming and batch processing. The resource manager is the node that keeps track of the resources present in the system, handles computation requests and schedules them. In the present version of Hadoop, this node is a single point of failure: if this node fails, all the jobs currently running must be restarted. This may result in the loss of hours of computation. Moreover, this node must handle heartbeats from all the nodes present in the cluster at the same time as it must take scheduling decisions. This puts the resource manager node under heavy load and results in a limit in the number of nodes that can be present in the system.

In order to remove the scalability issues caused by the name node, Hops stores the metadata in MySQL-cluster, a highly available distributed, in-memory database. It then runs several stateless name nodes that operate in parallel using the metadata stored in the database. This allows it to scale in terms of metadata operation throughput. In fact, the database can store up to several terabytes of metadata, this allows a HOPS cluster to store an order of magnitude more files than Hadoop. Moreover, having several name nodes operating in parallel allows HOPS to handle several read and write operations simultaneously while Hadoop can only run one operation at a time.

My work on this part of the project consisted in designing a leader election protocol that uses the database as distributed shared memory. HOPS need a leader election framework to run certain kinds of operation that need to be run by only one node at a time. Using the database as distributed shared memory for leader election reduces the number of specialized services that must be present in the system and reduces the amount of

maintenance work that the system administrator must handle.
This work has been accepted for publication at the International Conference on Distributed Applications and Interoperable Systems.

In order to reduce the scalability issues of the YARN resource tracker, HOPS distributes the tasks handled by this node. Instead of having one node handling the client requests, the scheduling and all the heartbeats sent by all the node managers present in the system, as it is the case in Hadoop, HOPS distribute these tasks on several nodes that specialize in a particular set of tasks. In HOPS the tasks of the resource trackers are handled by two types of nodes: a scheduler and several resource trackers. The scheduler receives the requests from the clients and schedule future tasks to be executed on the cluster, while the resource trackers handle heartbeats sent by the node managers. This allows HOPS to handle a higher number of heartbeats per second. This results in a system that can scale better in regard to the number of nodes in the cluster and that can be more responsive by increasing the frequency of the heartbeat. That is, we are building support for interactive applications on YARN.
In order for the scheduler to know the state of the cluster when taking scheduling decisions, the resource trackers store and update the state of the system in MySQL-Cluster. The scheduler could then pull this information from the database when it needs it. But, as pulling from the database is inefficient, this solution was improved to take advantage of MySQL-Cluster NDB event API. This API streams events from the database to the scheduler each time relevant tables in the database are updated. This way the scheduler is informed of update done by the resource trackers with a very small latency. Storing the state of the system also allows HOPS to provide transparent failover when the scheduler or a resource tracker fails.
I have been leading this part of the project. Starting from Hadoop Yarn we have designed and implemented all the modifications necessary to use the database and distribute the different part of the resource manager. This work is undergoing performance evaluation and should be the subject of a publication in the months to come.

II – PUBLICATION(S) DURING YOUR FELLOWSHIP

Submitted paper (Author Notification: March 23, 2015):

- **Conference:** DAIS 15th IFIP International Conference on Distributed Applications and Interoperable Systems.
- **Tittle :** Leader Election with a NewSQL Database
- **Authors:** Salman Niazi, Mahmoud Ismail, Gautier Berthou and Jim Dowling.
- **Abstract:** Leader election protocols are a fundamental building block for replicated distributed services. They ease the design of leader-based coordination protocols that tolerate failures. In partially synchronous systems, designing a leader election algorithm, that does not permit multiple leaders while the system is unstable, is a complex task. As a result many production systems use third-party distributed coordination services, such as ZooKeeper and Chubby, to provide a reliable leader election service. However, adding a third-party service such as ZooKeeper to a distributed system incurs additional operational costs and complexity. ZooKeeper instances must be kept running on at least three machines to ensure its high availability. In this paper, we present a novel leader election protocol using NewSQL databases for partially synchronous systems, that ensures

at most one leader at any given time. The leader election protocol uses the database as distributed shared memory. Our work enables distributed systems that already use NewSQL databases to save the operational overhead of managing an additional third-party service for leader election. Our main contribution is the design, implementation and validation of a practical leader election algorithm, based on NewSQL databases, that has performance comparable to a leader election implementation using a state-of-the-art distributed coordination service, ZooKeeper.

Pending:

- "Hops Yarn: Distributing Hadoop Scheduler", Gautier Berthou, Theofilos Kakantousis, Jim Dowling and Seif Haridi.

III – ATTENDED SEMINARS, WORKHOPS, CONFERENCES

- Presented:
  - **e-Infrastructures for Massively Parallel Sequencing**. 19-20 January 2015, Uppsala.

- Attended:
  - **NGS/Hadoop Workshop: Next Generation Processing for Next Generation Sequencing**. 19-20 February 2015, Stockholm.
  - **HEPTech Academia Meets Industry on Big Data ICT1**. 30-31 March 2015, Budapest.

IV – RESEARCH EXCHANGE PROGRAMME (REP)

REP Organisation: SZTAKI
Country: Hungary
department: Informatics Laboratory
local scientific coordinator: András Benczúr
dates: 24-31 March 2015
Outcome:

- I presented my current ERCIM project to the research group on the 26th of March. The group was interested and provided useful feedbacks on possible improvements.
- Attended the *HEPTech Academia Meets Industry on Big Data ICT1* workshop.