A framework for multidimensional indexes on distributed and highly-available data stores

Cesare Cugnasco
Barcelona Supercomputing Center
cesare.cugnasco@bsc.es

Advisor: Yolanda Becerra
UPC-Department of Computer Architecture
yolandab@ac.upc.edu

Abstract-*No-relational databases are nowadays a common solution when dealing with a huge data set and massive query workload. These systems have been redesigned from scratch in order to achieve scalability and availability at the cost of providing only a reduce set of low-level functionality, thus forcing the client application to implement complex logic. As a solution, our research group developed Hecuba, a set of tools and interfaces, which aims to facilitate developers with an efficient and painless interaction with non-relational technologies.*

This paper presents a part of Hecuba related to a particular missing feature: multidimensional indexing. Our work focuses on the design of architectures and the algorithms for providing multidimensional indexing on a distributed database without compromising scalability and availability.

I. INTRODUCTION

No relational distributed databases have grown in importance in the last years, as they are the best solution to the exponential increase of the quantity of information that nowadays database system have to manage. These systems' architecture focuses on scalability and availability, so that doubling the servers doubles the performance of the overall system and at the same time, the loss of a server does not result in a system failure.

NoSQL databases have largely proved to have achieved these goals. However, they still lack many desirable functionalities and thus many applications that would profit from their potentiality, are still forced to use older and less powerful solutions.

With Hecuba [Fig. 1] our research group aims to ease the aisle of using these technologies by proving a set of tools helping the developing application on the top of distributed database.

This paper is about a particular module of Hecuba, the one in charge of create, query and management multidimensional indexes on the top of a key-value database. Multidimensional indexes are the most effective way for selecting elements from a collection when we want to limit the results to only ones that meet a particular series of constraints set by two or more characteristics (dimensions).

Research on multidimensional indexes and distributed databases have been historically divided not only by their technical background, but also by their general goal. Research on indexing algorithms focuses on how to implement new kinds of query and how to reduce the bare number of I/O requests needed to serve them. Differently, research on distributed databases, focuses on how to achieve high scalability and robust availability on systems built on multiple servers.

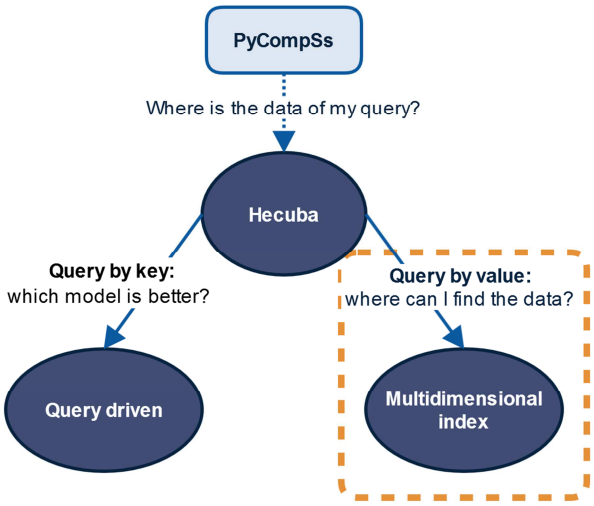

Figure 6: The architecture of Hecuba

II. CHALLENGES

What makes the meeting of these two worlds complex, is that many of the features, such as **locks**, **isolation** or strong **consistency** - that indexing algorithms need to work correctly - have been "sacrificed" in modern distributed systems, as they were huge obstacles to achieve the desired goals of scalability and availability.

New concepts, such as **eventual consistency**, **census**, **immutability** and **versioning**, have been proposed as partial replacements of those operations, but they all came with costs and limitations.

As a result, any application built on these data systems, needs careful engineering in order to decide whether to use, one technique rather than another, to decide how to deal with boundary cases and, most importantly, when and in which cases, to pay a higher performance cost.
At the same time, the research on multidimensional indexes has continued to produce new algorithms and ad-hoc optimizations to deal with particular datasets or query requirements. Unfortunately, these optimizations are usually inefficient for other applications. For example, some algorithms can work better with a lower number of dimensions while others behave best with a high number of correlated dimensions. As a result, a one-size-fits-all approach, derived from the implementation of a single algorithm, is not sufficient to deal with all possible applications.

Even though NoSQL databases are profusely used in many and complex scenarios - from transactional to streaming and batch reporting - a distributed, and highly available database system, able to deal with arbitrary dimensional data by implementing different indexing algorithms, still does not exist to our knowledge.

III. APPLICATIONS

Multidimensional indexes have plenty of applications in nowadays research and industry, even though their usage is limited by the scalability of the indexing system. Our work aims to overcome these limits thus enabling much wider and massive applications.

HPC

Molecular dynamic simulations are typical applications in High-Performance Computing (HPC) research, as long as they are used for wide cases from theoretical physics, biochemistry and biophysics. These simulations require a massive quantity of computational resources and thus they are executed in Supercomputers. Scientists use these simulations in order to validate theories and study particular physical behavior.

The typical workflow consists in: first the design of the experiment (simulation), then to reserve a time spot in a supercomputer, to execute the simulation and, once it completes, to copy the resulting trajectory files on a local computer in order to analyze the results. If instead the trajectories were directly written in a distributed multidimensional database while the simulation is running, there would be multiple benefits:

1. Researchers would be able to visualize partial results during the execution of a simulation so that they may be able to correct or to interrupt a simulation in case of incorrect behavior.
2. The database can be used as a central repository for all the simulations so that different old studies can be easily used in new research
3. The database could support complex queries such as finding similar images or trajectories across a vast database of simulation, providing a powerful research tool for researchers.

Low latency complex query on multiple attributes.

One of the most common mistakes approaching Cassandra as long as others NoSQL databases, is to try to issue a query that implies multiple conditions on the dataset. Indeed, the database instantaneously reports that it does not support this kind of query. Instead, the same request issued on RDBMSs would perform correctly, even though requiring a considerable amount of time.

As a workaround, in NoSQL databases it is common practice to create materialized views in order to filter by attributes, but still, it can help only with a limited subset of queries. Differently, a multidimensional index would be the perfect solution for implementing this kind of requests and, at the same, time achieving the desired constraints of availability and low latency.

Image and trajectory similarity research

In order to search for similar elements between images or trajectories, the existing algorithms work applying transformations on the input data so it can be described using a limited set of descriptor parameters. For example, a trajectory can be discretized using the Fourier transformation and describing the trajectory with the resulting parameters.

In such a way, a similarity query can be converted into a multidimensional range one where the boundaries are set on the descriptor parameter values.

Even though the complex part of similarity research consists in the choice of which algorithm and descriptor parameters are more effective for each specific case, the underlying system is always a multidimensional database.

Data warehousing

On Line Analytics Processing systems (OLAP), have historically used multidimensional indexes in order to support complex and reporting queries. In any way, for their architecture, they work with a two exclusive phases workflow. In the Extraction, Transformation and Loading phase (ETL), the data is read from an external source - usually from logs or transactional databases - and then it is elaborated and finally loaded into a multidimensional index. Only in the second phase, once the ETL phase concludes, it is possible to issue queries on the system. This approach forces the OLAP system to be unavailable to queries for a considerable amount of time. For many applications this can be unbearable, such as for fraud detection systems working in trade markets or anomaly detection systems for the network of large Internet Service Provider.

ACKNOWLEDGMENT

The research leading to these results has received the support of the grant SEV-2011-00067 of Severo Ochoa Program, awarded by the Spanish Government

PAST PUBLICATIONS

[1] C. Cugnasco, R. Hernandez, Y. Becerra, J. Torres and E. Ayguadé, "*Aeneas: a tool to enable applications to effectively use non-relational databases*" Procedia Computer Science ICCS 2013

[2] R. Hernandez, C. Cugnasco , Y. Becerra, J. Torres and E. Ayguadé, "*Experiences of using Cassandra for molecular dynamics simulations*" Euromicro International Conference on Distributed and Network-based Processing PDP 2015