White paper

open source solutions for big data management

Your business technologists. Powering progress

Atos

Open Source Solutions for Big Data Management

Big Data Management is becoming a key issue in the IT world. To deal with it, the Open Source world is providing numerous solutions, often powerful, but often immature as well. In this paper, we focus on two major technologies: NoSQL solutions and MapReduce frameworks. We provide a general description of the principles behind these technologies, an overview of some of the available solutions and we discuss their capabilities and their limits.

About the Authors

Julien Carme, R&D engineer, Atos Worldline (julien.carme@atos.net).

Francisco José Ruiz Jimenez, Technical Architect Manager, Atos Spain (fjose.ruiz@atos.net).

Thanks to Antoine Fressancourt (antoine.fressancourt@atos.net), R&D engineer, Atos Worldline, for his support.

Contents

Introduction

Until the end of the 1990s, computer performance increased rapidly enough to meet growing data storage and processing needs.

However since then, several major changes in the IT world have dramatically increased this rate of growth. The first was the expansion of the Internet that allows global access to huge amounts of information and led to a need for efficient tools, typically search engines, to process it. The second was the rise of Web 2.0, which offers everyone the ability to generate and share their own data (i.e. blogs, YouTube, Facebook, etc.). Next will probably be the development of the Internet of Things, and Context-Aware Computing (CAC), which will bring data storage requirements and associated information processes to an even higher level.

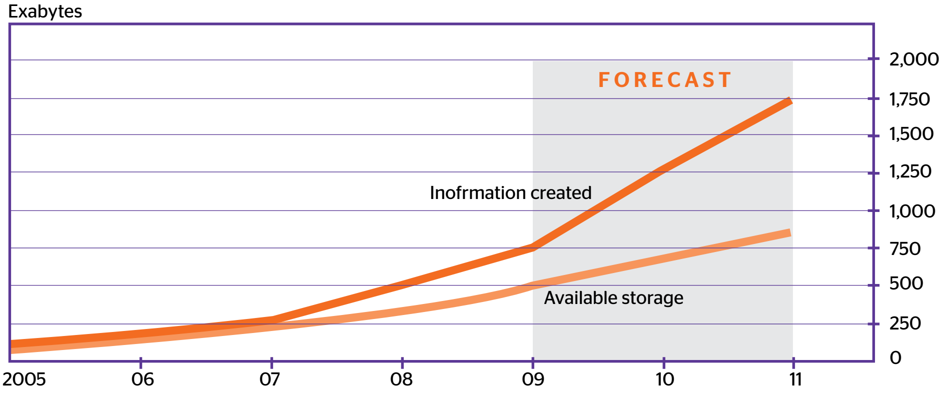
Computer capabilities have not increased fast enough to meet these new requirements. When data is counted in terabytes or petabytes, traditional data and computing models can no longer cope.

The first companies to face this issue were the big Internet players: Providers of search engines and web portals, and major e-commerce players. Their answer was to develop solutions that feature horizontal scalability; that are able to scale information to arbitrarily large clusters of computers and enable performance to grow linearly with the size of these clusters.

At the beginning, these solutions were specifically developed and designed in house to run internally. Then, in order to share efforts, some open sourced their solutions and started working collaboratively on them. As a result, Open Source solutions are at the forefront of massive data management.

When data is counted in terabytes or petabytes, traditional data and computing models can no longer cope.

Overload
Global information created and available storage



Source: IDC

Storing Data

The limits of Relational Database Management Systems

Relational database management systems (RDBMSs) have been the main and almost sole means of data storage for almost 40 years. They are based on a firm theoretical background, and implementations, including Open Source implementations like MySQL and PostgreSQL, have reached high levels of maturity and performance.

However, RDBMSs are not horizontally scalable. They have not been designed to run on clusters of servers on which to perform masses of processes in parallel. Since their limits, in terms of performance, have been reached, alternative solutions have to be found.

Ways to distribute relational databases do exist (i.e. Oracle RAC, Sybase ASE Cluster Edition), but they come at a price in terms of features, data availability, performance, and administration cost, and do not seem to satisfy the most major Internet players. The approach most of these companies have chosen is to give up the relational model and use solutions that have been designed from scratch with horizontal scalability in mind. This is the NoSQL approach.

NoSQL: Definition

NoSQL is a movement which was originally defined in the negative: NoSQL solutions are data stores which are not based on the relational model.

However, NoSQL solutions share one common goal, the ability to handle very large quantities of data and distribute it to a large number of data servers. In order for data storage distribution to be manageable, NoSQL solutions feature, to some extent, **elasticity** and **fault tolerance**.

Elasticity is the ability to add and remove new data nodes as desired without interrupting service. This is necessary for large clusters of servers running services, typically web services, which must run 24/7. Elasticity raises non-straightforward technical problems: Each time a new data node is added, some part of the stored data should be transferred to it dynamically in order to balance the load between data nodes, with limited impact on performance.

Fault tolerance is the ability to make data safe and as available as possible, even when hardware fails (network or data node). This is achieved through internal data replication mechanisms. On large clusters, hardware failures are unavoidable, so all NoSQL solutions feature some level of fault tolerance, as discussed in the next section.

The way NoSQL databases store data and the way the client accesses it is not unique; it depends on the data model, and therefore on the chosen NoSQL solution. More details are provided below.

Google WebTable: A Typical NoSQL Usage Example

The Google WebTable is the primary use for and original goal of Google BigTable, a NoSQL solution developed by Google (which is not Open Source, but has been cloned by Open Source solutions, like HBase).

It is a great example because it shows very clearly how scalable NoSQL solutions can be.

The Google WebTable stores, in one single table, a complete copy of the part of the Internet which is indexed by Google, with the complete graph of links between pages. Google crawls around the Internet in order to update this table and process it for its pagerank algorithm and to build answers to user queries.

	Content	Link: www.site2.com	Link: www.site3.com	...
www.site1.com/ firstpage	<html><head> ...(page content) </body></html>	Site1	Link to site1	
www.site1.com/ examplepage	<html><head> ...(page content) </body></html>			
www.site2.com	<html><head> ...(page content) </body></html>			
...				

The Google WebTable. Each line stores a webpage. All webpage contents are stored in a single column (labeled 'Content'). Another column (labeled 'Link: X') contains the links between pages. The cell ('Y', 'Link:X') contains, if it exists, the text of link from the website 'X' to the page 'Y'. The WebTable contains a complete copy of the Internet, billions of rows, millions of columns, hundreds of terabytes of data.

Is NoSQL Mature for Business Usage?

Open Source relational databases are high-performance tools that have reached a high level of maturity, and for which Atos has a large number of in-house experts. These databases include a lot of features that Atos' customers require and that NoSQL solutions do not currently provide without additional client code. Typical examples include rollback capabilities, backup, and right management. In addition, consistency, as explained later in this paper, is not an issue.

NoSQL solutions are comparatively recent, have evolved quickly, but are generally considered to still be immature. They are used in production for large-scale projects; however feedback tends to show that their usage is still considered experimental. Using NoSQL solutions requires specific expertise so, often, companies using them bring in the experts who developed the solution for help.

In most situations, using NoSQL solutions instead of RDBMSs does not make sense in cases where the limits of the RDBMS have not been reached.

Although, given the current exponential growth of data storage requirements, these limits are increasingly likely to be reached in the future. Unless RDBMS evolves quickly to include more advanced data distribution features, NoSQL solutions will become more and more important.

Choosing a NoSQL Solution: A Brief Guide

A wide range of NoSQL solutions currently exists and choosing the right one for a task can be difficult. There is a clear lack of trustable information, the field is evolving very quickly, and most observers are often users or even contributors, and therefore do not have adequate distance from the subject.

The guidelines provided in this paper are based partially on a synthesis of existing information from NoSQL users and partially from an evaluation of NoSQL solutions that Atos is currently undertaking in order to offer customers the best option to meet their requirements.

Overview of Investigated Solutions

The most well-known and recognized NoSQL solutions are being considered:

- HBase - Clone of Google BigTable, adapted to massive data storage based on Hadoop, developed by Powerset (acquired by Microsoft), and used by Yahoo and Microsoft.
- Cassandra - Similar to HBase, developed by Facebook.
- Hypertable - Clone of HBase claiming higher performance, supported by Baidu.
- Voldemort - Pure key-value store, oriented toward low latency, high performance, adapted to medium-sized clusters, developed by LinkedIn.
- Membase - Based on Memcache, developed by Zynga.
- MongoDB - Designed for storing semi-structured documents, developed by 10gen.
- CouchDB - Another document-oriented data store, similar to MongoDB.

The Data Model

NoSQL solutions are not all based on the same data model. Understanding the differences between existing NoSQL data models and what each of these data models is used for is important in selecting the right NoSQL solution.

Existing Data Models

Four NoSQL data models exist:

Key-value stores use the most basic NoSQL data model. Key-value stores are distributed, persistent hashtables. Data is stored in small chunks (bytes or megabytes) and linked to a key. Data can be retrieved using this key. Among the NoSQL solutions investigated by Atos, Voldemort and Membase are key-value stores.

Column-oriented data stores are similar to key-value stores, except that they use bi-dimensional keys (rows and columns) with data stored in columns. As a result, the data stores are optimized for accessing blocks of consecutive elements in the same columns. The ancestor of column-oriented data stores is Google BigTable, which is used by Google to store a complete copy of the Internet. The most well-known Open Source NoSQL solutions are based on this data model: Cassandra, HBase, and Hypertables.

Document-oriented data stores are also based on key-value stores, but the data associated with each key is a semi-structured document (typically JSON or XML). Advanced queries based on specific properties of these semi-structured documents are available. MongoDB is the most widely used document-oriented database.

Graph-oriented databases are designed to store and query graph-structured data. The main graph-oriented database is Neo4j.

Choosing the Right Data Model

For a given application, depending on the data that has to be stored and the way it needs to be accessed, some data models are more appropriate than others. For example, storing and querying graph data is easier with a graph-oriented data store. It also offers better performance. Document-oriented databases are more suitable for storing large sets of small semi-structured documents. Accessing big blocks of consecutive data can be carried out more efficiently using a column-oriented database.

In practice, NoSQL databases are often used for basic key-value storage: One key is linked to a small chunk of data, and this key is then used to write, read, or update the data. In this case, pure key-value, column-oriented and document-oriented data models work.

Scale

Even though all NoSQL solutions are scalable, not all NoSQL solutions are used at the same scale.

HBase, Cassandra, and Hypertable are used by major players with clusters of several hundred data nodes. At this size, clusters remain manageable because cluster administration does not require any task to be carried out individually on each server. Using a solution that can handle this is a reasonable approach in cases where terabytes or petabytes with millions of users need to be managed.

Voldemort, MongoDB, Riak, and Membase are different. Their administration requires more work and individual server management, which does not seem compatible with clusters of several hundred machines. Publically available feedback on the usage of these solutions involves clusters of tens of servers at most. These solutions are appropriate where a reasonable amount of data needs to be managed, typically less than a terabyte, and there is a preference for a NoSQL solution because the existing database cannot cope.

The Consistency Model

The consistency model is a technical feature that has to be taken into account in the choice of a NoSQL solution, depending on the requirements of the application.

For theoretical reasons, distributed data storage has to make a trade-off between consistency, the ability to avoid conflict during concurrent data access, and availability, the ability to always provide an answer even when hardware fails.

Some NoSQL solutions advertise **high availability** with reduced consistency as a constraint. These solutions are suitable in applications in which availability is crucial but conflicts are not a big issue (typically when data is not massively updated) or when it can be handled easily. Typical examples of use cases are website databases that should always be up and there are usually few data updates. Examples of solutions featuring high availability and no strong consistency are Voldemort and MongoDB.

Some other solutions have a stronger consistency model. HBase features a strong consistency model inside each rack (subset of a cluster comprised of servers located in the same place, therefore featuring low latency), but no strong consistency between racks.

The Cassandra approach is to make consistency customizable by the client. For each query, strong consistency, high availability, or an intermediate consistency model can be implemented.

The question of the consistency model and the various ways to solve consistency issues are major topics of debate within the NoSQL community. It is interesting to note that Facebook has recently decided to use HBase instead of Cassandra (even though Cassandra was developed by Facebook) for its new messaging system, because it believes that the HBase consistency model is more suited to this specific task.

Support and Trends

Open Source NoSQL solutions are often associated with a company which is its main user, main contributor, and often its original creator. This is the case of Cassandra with Facebook, HBase with Microsoft (since it has acquired Powerset), Hadoop with Yahoo, Voldemort with LinkedIn, Membase with Zynga, and HyperTable with Baidu.

Some other NoSQL solutions are more community-based; they are therefore not associated with a single big player. This is the case with MongoDB, which is used by a large number of Internet startups, but whose main developer is 10gen.

Having a NoSQL solution supported by major players can be considered good for future corresponding solutions. However, the field is still evolving rapidly and even big players may switch to alternative solutions if they are technically superior. For example, Facebook's current move from Cassandra to HBase means it is now putting more development effort into HBase than Cassandra.

In terms of long-term support and evolution, the current most reasonable choice seems to be HBase. Although community-based solutions like MongoDB, used by a very large number of small companies, are also likely to be well supported in the future.

Data Processing Requirements

As explained in the next section, the usual way to process distributed data uses the MapReduce algorithm. Both Cassandra and Eclipse provide a good interface for the Hadoop implementation of MapReduce. MongoDB also provides its own MapReduce scheme. Other NoSQL systems do not natively include MapReduce capabilities.

Performance

All NoSQL solutions claims to reach high levels of performance and high levels of scalability, although the few benchmarks available do not seem reliable, in particular because they are often contradictory.

Atos' ongoing evaluation of NoSQL solutions includes measuring the performance of a cluster of five quad-core servers on which each of the cited NoSQL solutions has been installed. Performance is measured by the number of queries per second that the system can handle in various scenarios: Read only, read-write, updates, read while load balancing, etc.

Preliminary results show that:

- Voldemort, which advertises very high performance and availability, but at a price in terms of data consistency, actually performs much better than other solutions.
- HBase, Cassandra, Membase, and MongoDB achieve similar results.
- CouchDB does not scale properly and performs much worse than other solutions.

However, it should be made clear that raw performance on a small cluster only provides partial information about real performance on a system: No indication about a solution's ability to scale on larger clusters is given.

Open Source Solutions Summary

	Cassandra	HBase	Voldemort	Membase	MongoDB	CouchDB
Type	Column-oriented	Column-oriented	Key-value	Key-value	Document-oriented	Document-oriented
Maurity	***	***	***	*	***	**
Support	****	****	***	*	***	***
Typical use	Massive amount of data	Massive amount of data	Heavy load	Heavy load	Rich querying, Heavy load	Rich querying
Typical cluster size	Up to hundreds of servers	Up to hundreds of servers	Up to tens of servers	Up to tens of servers	Up to tens of servers	Some servers
Specific multi-site management	Yes	Yes	No	No	Yes	No
Map.reduce processing support	Yes	Yes	No	No	Yes	Yes
Performance	***	***	*****	***	***	*

Processing Data

With the ability to store data on large clusters of servers came the need for tools that would be able to process that data. In some cases, processing is part of the primary purpose, typically in the case of a search engine where answering requests requires the efficient processing of all data collected. In other cases, processing is used for data analysis, in order to extract relevant and valuable information from the stored data.

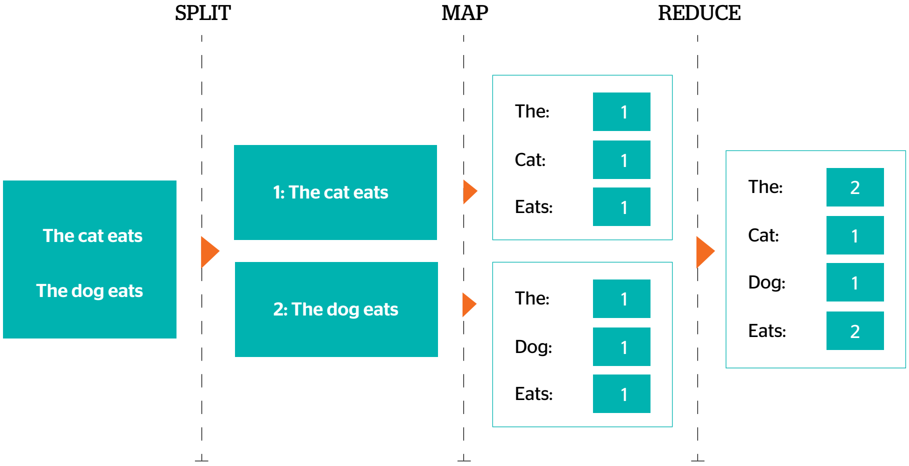
The Map Reduce Framework

MapReduce is a framework which provides the necessary tools to distribute processing tasks to an arbitrarily large cluster of computers. It was developed by Google for internal use. Google published its principles, which have since been implemented in several Open Source tools, in particular in Apache Hadoop.

Principle

In the MapReduce Framework, every data processing task is divided into two steps: the Map step and the Reduce step. In the Map step, the task is divided into independent subtasks which are executed in parallel by Map processes, running on distinct servers. In the Reduce step, results of the Map processes are recursively merged by Reduce processes, which also run in parallel on distinct servers, until the final result is computed.

For example, assume one wants to count the number of occurrences of some given strings in a NoSQL database. A split process divides the problem into autocontent data using specific criteria. In this case, the data are sentences. Each Map process counts the number of occurrences of each of the strings in a single data node. Each Reduce process adds up the outputs of Map processes for a given string.



Data Storage and Data Processing

The MapReduce framework is designed to reduce data transfer as much as possible. Map processes run, where possible, on the servers on which the corresponding data is stored, and Reduce processes are placed as close as possible to the Map processes that provide the input.

In order to create this optimal environment, which is crucial for the efficiency of the system, the MapReduce framework has to be tightly linked to the data storage management system. In practice, this means that the data processing system and the data storage system must be part of the same system (this is typically the case with Hadoop and HBase) or that the data storage system has been specifically designed to interface with a given data processing system (for example, Cassandra with Hadoop).

Development Requirements

Developing in the MapReduce framework means developers have to think in new ways. In the previous example, expressing a task as a MapReduce process is pretty straightforward, but for more complex tasks and for tasks that cannot obviously be carried out in parallel, it can be challenging. Often, a task has to be divided into a sequence of consecutive subtasks which should themselves be efficiently divided into Map and Reduce processes. In addition, running a MapReduce process requires optimization: The MapReduce framework user has, for example, to configure a balance between resources allocated to Map processes and to Reduce processes.

Usage of this kind of distributed-computing framework requires specifically trained people. There have been, however, attempts to build an abstraction layer over the MapReduce algorithm to hide its complexity. Hadoop includes some hints, which are given below.

Hadoop and the Hadoop Ecosystem

Hadoop, and its ecosystem, is the main distributed-computing framework. It is an Open Source project supported by the Apache Foundation. Its main contributor is Yahoo, but most big players use it, including Microsoft, EBay, Google, Facebook, Amazon, Rackspace, and many others.

Hadoop includes a file system; Hadoop Distributed File System (HDFS), which allows file storage on large clusters of data nodes in a transparent way. It is a file storage system which can handle a limited number of (potentially big) files (not more than a number of millions before memory issues appear), and therefore cannot be considered to be a NoSQL solution. However, it is suitable for many data storage tasks, as long as the data can be stored sequentially. In particular, it is perfectly suitable for data stream storage, for example, for log storage.

Hadoop includes a MapReduce implementation in Java.

Hadoop comes with a set of tools, called the Hadoop ecosystem, which include:

- Pig and Hive - two high-level languages built over Hadoop MapReduce that hide its complexity. These tools allow users to write basic distributed data processing tasks quickly without any knowledge of distributed computing. It is estimated than 30 percent of Hadoop jobs run at Yahoo are actually Pig jobs.
- ZooKeeper, an administration tool for large Hadoop clusters.
- HBase, the Hadoop NoSQL solution built on top of HDFS, already mentioned above.

What is Hadoop used for?

Hadoop is usually used by companies to run batch computations, in order to analyze stored data. Typical applications are:

- Data mining – To get relevant information about users' behavior, anomalies in a system, suspicious network access, etc. where terabytes of logs are stored.
- Image processing – When basic image processing services, like resizing and conversion, or more advanced processing services, like facial recognition, are provided to website users and millions of images have to be processed.
- Web crawling – When an application is being run, typically a specialized search engine, that needs to process large parts of the Internet, a Hadoop cluster can help.

It should be noted that the MapReduce algorithm in general, and Hadoop in particular, have been written with batch computation in mind, and are therefore not suitable for real-time computation. When real-time results are required, answers to queries should be based on results already computed and stored in a NoSQL database.

Recently, a new generation of distributed-computing frameworks designed for real-time computation has been introduced. These include Google Caffeine and Yahoo S4, the latter has, interestingly, been open sourced by Yahoo. Yahoo S4 is presented as an Open Source distributed-stream processing platform. It is still an immature tool (still in alpha version), but is already seen by many as the future of Open Source distributed computing.

Search Engines

The role of a search engine is to process massive amounts of data and index it in such a way that it can be efficiently accessed by a large number of users simultaneously. It is therefore not surprising that, as stated above, search engines are major users of distributed-computing and storage technologies, and search engine providers were first to introduce them.

The most important Open Source project related to search engines is Lucene, which was originally implemented by Doug Cutting and has been supported by the Apache foundation since 2001. Lucene has been deployed on many websites or within enterprises as a search engine. It has also been the reference for other projects, such as Solr and ElasticSearch. Solr, also an Apache Foundation project, adds some key functionality to Lucene: Full-text search, faceted search, dynamic clustering, REST-like and JSON APIs, database integration or distributed search, none of which are provided as ore functionalities by Lucene, which just implements primary search functions without any other abstraction or integration layer.

Lucene/Sorl and NoSQL are similar in a number of ways. From a technical point of view, the non-structured storage facility included in Lucene's indexes is similar to NoSQL models (key-value or document-oriented). Indeed, there are a number of initiatives concerned with storing Lucene indexes using NoSQL storage (i.e. CouchDB). Search-engine technologies are evolving to become more distributed systems, implementing some of the key NoSQL characteristics, such as massive parallel processing, fault tolerance, distributed processes, and scalability. Solr architecture allows these mechanisms and facilitates new Open Source initiatives based on Hadoop because Zookeeper provides infrastructure to the SolrCould project, which will work as a massive distributed search engine.

Lucene/Sorl has been widely accepted as the most important search-engine solution, not only for medium-sized websites, but also for some larger ones, like Wikipedia or LinkedIn, which use Lucene for member search functionality.

Open Source Big Data Management at Atos

Atos is developing and investigating Big Data Management as key technology for future customer requirements. Both internal and global customers will require more and more data stores and the associated processes to manage huge amounts of information. Some projects and initiatives are focused on HTTS (massive transaction management) and Atos Research & Innovation, which provides high-value and technical solutions to global key customers.

Atos

Projects focused on distributed computing using some Open Source solutions have been initiated:

BUSCAMEDIA Project

This project is a Spanish-funded R&D project the main objective of which is to provide a semantic tool to index, access, and locate media content. In order to perform the automatic categorization and annotation of media content, the technologies being applied are based on Google's MapReduce concept, using Hadoop to implement this functionality. http://www.cenitbuscamedia.es/

BEinGRID Project

The main objective of the 'Business Experiments in Grid' (BEinGRID) project is to foster the adoption of the so-called next-generation grid technologies by carrying out 25 business experiments and creating a toolset repository of grid middleware upper layers. Of these 25 experiments, one is notable; TAF-GRIDs, whose main objective was to detect Telecom Fraud where customers' usage records were exchanged, captured, stored, and analyzed among different telecom operators. The technologies used to perform the experiment were Globus Toolkit, for computational grid jobs and, OGSA-DAI, for data management. http://www.beingrid.eu

Atos Worldline (HTTS)

Atos Worldline provides e-commerce solutions to several major e-sellers, in particular to the majority of French mass retail sellers. One of the main features of this kind of customer is the massive amount of users that their e-commerce websites have to handle simultaneously.

For its next e-commerce solution, Massilia, Atos Worldline is investigating the use of NoSQL solutions for some data storage tasks as NoSQL solutions are considered to store product-related data, which is an extremely heavy load, efficiently. The Atos Worldline R&D team is currently evaluating NoSQL solutions in this context.

Atos Worldgrid

In a project for FP7, Open Node, Atos Worldgrid is working on the definition of a NoSQL storage mechanism for SmartGrid. In the context of Open Architecture for Secondary Nodes of the Electricity SmartGrid, a repository will be implemented using an Open Source Big Data solution. The repository will store a huge amount of data including all the measurements gathered from the secondary nodes. Cassandra has been chosen as the best solution for this implementation. Companies like IBD, EDF, EDP and ITE are also involved in this project (http://www.opennode.eu/)

Customers will require more and more data stores and the associated processes to manage huge amounts of information.

Conclusion

Until recently, only a couple of big Internet players were concerned by the issue, but things are changing fast and now more companies have to handle massive amounts of data.

In addition, Atos believes that emerging fields, like Internet of Things and Context-Aware Computing, which necessarily involve the ability to manage lots of data, will become key fields in the near future.

NoSQL and distributed-computing technologies provide immature yet powerful solutions to Big Data challenges. Understanding their capabilities and their limits are key.

Interestingly, most of these solutions are Open Source. This usually means that users do not depend on a single vendor, but that a higher level of expertise is required. This is a good opportunity for Atos, which can provide this expertise and help customers get the most out of these solutions.

About Atos

Atos is an international information technology services company with annual 2010 pro forma revenues of EUR 8.6 billion and 74,000 employees in 42 countries at the end of September 2011. Serving a global client base, it delivers hi-tech transactional services, consulting and technology services, systems integration and managed services. With its deep technology expertise and industry knowledge, it works with clients across the following market sectors: Manufacturing, Retail, Services; Public, Health & Transport; Financial Services; Telecoms, Media & Technology; Energy & Utilities.

Atos is focused on business technology that powers progress and helps organizations to create their firm of the future. It is the Worldwide Information Technology Partner for the Olympic Games and is quoted on the Paris Eurolist Market. Atos operates under the brands Atos, Atos Consulting and Technology Services, Atos Worldline and Atos Worldgrid.