Research Note

cortex
aktiengesellschaft

Positioning CortexDB in Hadoop Data Architectures

Extending the Data Warehouse and beyond

An analysis by the

WOLFGANG MARTIN TEAM
powerful connections

www.wolfgang-martin-team.net

6 rue Paul Guiton
74000 Annecy
France

Contents

WOLFGANG MARTIN TEAM
powerful connections

Acknowledgement

I would like to offer my special thanks to **Mike Ferguson**, Managing Director of Intelligent Business Strategies Limited for his very constructive and valuable comments and discussions.

1. Management Summary

Up to now, data architectures for structured data consist of two parts: an architecture for operational (transactional) data and an architecture for analytical data plus the data management processes linking both worlds. Both architectures were built on top of relational database technologies. Furthermore, organizations used content management systems for managing unstructured data.

Big Data now changes this traditional world completely. Indeed, Big Data has to be managed, and this is where the Apache open source solution **Hadoop** plays a major role. Hadoop is designed to store and analyze both structured and unstructured data ("data in rest"). In addition, it can also manage data streams ("data in motion"). Also, given that new analytical workloads can run on Hadoop, this new platform needs to be added into traditional data architectures.

There is a second movement in data management pushed by Big Data: the upcoming **NoSQL** database technologies. Their success is based on a simple fact. Many of the new analytics and information processing challenges in Big Data analytics can no longer be addressed by the traditional relational database technologies.

Up to now, in traditional data architectures, the **data warehouse** is still considered to provide the single point of truth, i.e. it is the central platform for all BI systems. But Big Data challenges make Hadoop to be the new platform needed to support today's analytical workloads that go beyond those supported by a data warehouse. In addition, more and more BI vendors provide direct access to Hadoop so that traditional BI tools can now access Hadoop via SQL on Hadoop. Will the data warehouse and its relational technology become obsolete and be replaced by Hadoop?

Indeed, the market place has not yet found a definitive answer to this question. In this research note, we will discuss the issue and conclude that the data warehouse will continue to be center and single point of truth for tactical and strategic performance management, at least for the foreseeable future. But ETL processing and some analytical workloads will move increasingly to Hadoop, particularly if they involve analyzing semi-structured and unstructured data.

Furthermore, NoSQL databases are becoming an increasingly important part of this new database landscape. They complement Hadoop and offer various advantages over relational database technologies. Indeed, Hadoop complemented by NoSQL technologies offers the potential to handle Big Data analytics and to develop new innovative apps that were not possible by using relational technology.

CortexDB is one of the NoSQL database technologies that is complementary to Hadoop when sitting on top of Hadoop. In this research note, we have a closer look at CortexDB, and discuss what makes CortexDB stand out: CortexDB is a temporal, multimodal NoSQL database technology that differs from all known databases via its index structure and its content-orientation. We will demonstrate how, and we will show that networked systems, data dependencies, complex configurations, very large data volumes, and constant changes can be well addressed by the schema-less multi-model CortexDB. Indeed, requirements like these cannot be managed by Hadoop alone. In other words: CortexDB is the solution for managing large quantities of complex poly-structured data in Hadoop data architectures.

2. Big Data Challenges to Enterprise Data Management

Hadoop: Definition (from WhatIs.com [1])

Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation.

Hadoop makes it possible to run applications on systems with thousands of nodes involving thousands of terabytes. Its distributed file system facilitates rapid data transfer rates among nodes and allows the system to continue operating uninterrupted in case of a node failure. This approach lowers the risk of catastrophic system failure, even if a significant number of nodes become inoperative.

Hadoop was inspired by Google's MapReduce, a software framework in which an application is broken down into numerous small parts. Any of these parts (also called fragments or blocks) can be run on any node in the cluster. Doug Cutting, Hadoop's creator, named the framework after his child's stuffed toy elephant. The current Apache Hadoop ecosystem consists of the Hadoop kernel, MapReduce, the Hadoop distributed file system (HDFS) and a number of related projects such as Apache Hive, HBase and Zookeeper.

The Hadoop framework is used by major players including Google, Yahoo and IBM, largely for applications involving search engines and advertising. The preferred operating systems are Windows and Linux but Hadoop can also work with BSD and OS X.

As the definition of Hadoop shows, it works like an operating system for data. This is why it is going to be a standard in Big Data management. So, the good question is how Hadoop will change existing data architectures.

Up to now, data architectures for structured data consist of two parts: an architecture for operational (transactional) data and an architecture for analytical data plus the data management processes linking both worlds. Both architectures were built on top of relational database technologies. Furthermore, organizations used content management systems for managing unstructured data.

Big Data now changes this traditional world completely by bringing in five new data sources:

- The mobile internet produces new data streams of location and navigation data.
- Social media add various types of communication, mainly semi-structured and unstructured data and data streams.
- Sensors provide machine data. These are data streams with high and highest velocities.
- Other data streams consist of log data coming from all servers and providing insight in user behaviors.
- Finally, another Big Data source is clickstream data, which is already very familiar to us.

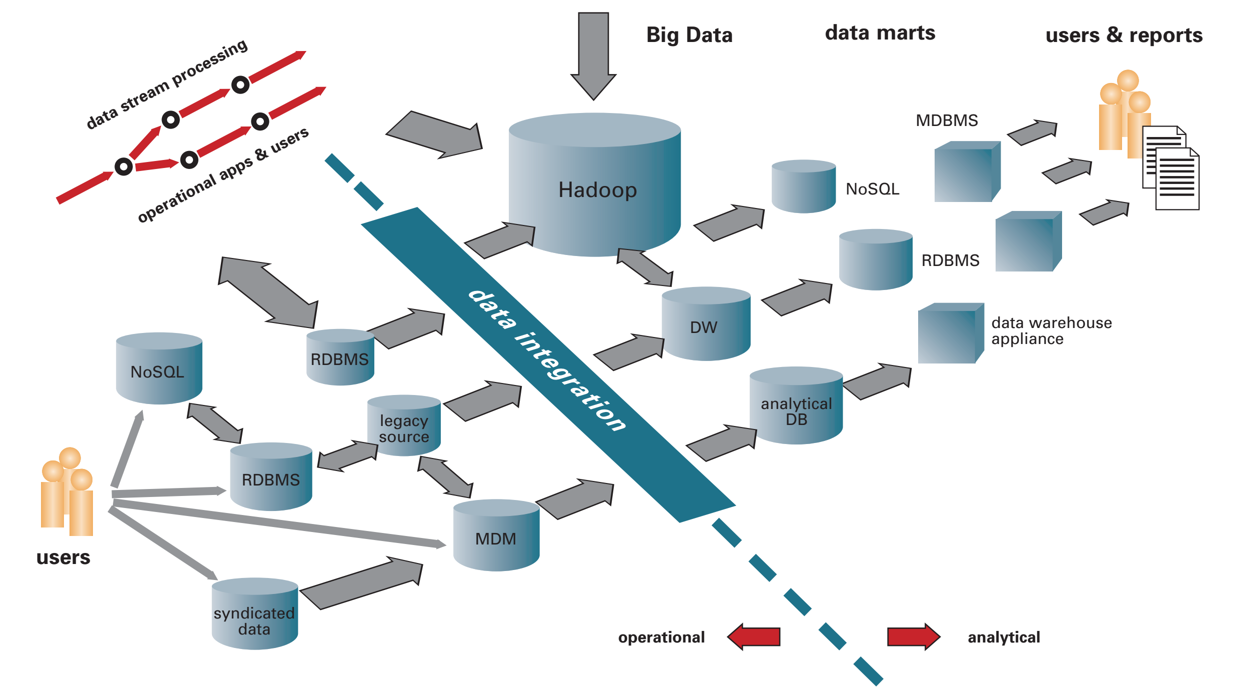
[1] see http://searchcloudcomputing.techtarget.com/definition/Hadoop

All this data now has to be managed, and this is where Hadoop plays a major role. Hadoop is designed to store and analyze both structured and unstructured data ("data in rest"). In addition, since the emergence of Hadoop 2.0, you can also analyze data streams ("data in motion") using Apache Storm running on Hadoop. Also, given that new analytical workloads can run on Hadoop, this new platform needs to be added into traditional data architectures.

But there is a second movement in data management: the upcoming NoSQL database technologies. Their success is based on a simple fact. Many of the new analytics and information processing challenges in Big Data analytics can no longer be addressed by the traditional relational database technologies. There are several reasons:

- Data volume exceeds the capabilities of relational database technology. Indeed, relational technology works best with single select inquiries, but becomes weaker with group select inquiries.
- Data arrival rates exceed the capabilities of relational database technology. Data stream processing is not really new, and alternative technologies to relational technology have been in use for a long time.
- Data variety exceeds the capabilities of relational database technology. Relational technology works well with structured data, but has its limitations when processing unstructured data.

Data Architectures: a Resume



This is not a reference architecture!

Figure 1: *Big Data is forcing data management architectures to evolve. Hadoop and NoSQL technologies complement the traditional architectures for operational and analytical data. But the entry of Hadoop and some NoSQL technologies raises questions about the future role of the Data Warehouse for all analytical workloads. Furthermore, coexistence between relational and NoSQL technologies is now required.*

Furthermore, Big Data is a combination of all five new data sources plus traditional enterprise data. Consequently, in many business cases, alternative data technologies are needed. This opens the window of opportunity for NoSQL database technologies.

Figure 1 summarizes the actual data architectures where Hadoop and NoSQL database technologies enter the enterprise. This summary data architecture is not a reference architecture – it is just the summary of what is actually happening in organizations.

3. Hadoop and the Data Warehouse

Up to now, the data warehouse is still considered to provide the single point of truth, i.e. it is the central platform for all BI systems. Hadoop is a new platform needed to support new analytical workloads that go beyond those supported by a data warehouse. In addition, more and more BI vendors provide direct access to Hadoop so that traditional BI tools can now access Hadoop via SQL on Hadoop. SQL on Hadoop is making progress. [2] Will the data warehouse and its relational technology become obsolete and be replaced by Hadoop?

Indeed, the market place has not yet found a definitive answer to this question. A very wise point of view has been stated by Steve Miller: "Big Data and the data warehouse serve different masters. DW has historically revolved on performance management, while Big Data obsesses on analytical products for data-driven business." [3] We share his point of view. In fact, BI has long been split up into performance management (PM) and analytics [4], and indeed, the requirements of performance management and analytics to data and data management are quite different. So his statement makes a lot of sense.

Unsurprisingly, his position is backed up by Data Warehouse vendor Teradata claiming "the data warehouse as a proven, high-performance environment for storing more important data and running mission-critical analytical workloads." Teradata sees coexistence, and it believes that Hadoop will be simply another player in the data management stack. [5]

Again unsurprisingly, Hortonworks, Hadoop distributor n° 2, also says that Hadoop complements the data warehouse. Well, Hortonworks has a very close partnership with Teradata and has reseller agreements with HP, Microsoft, and SAP: all three are data warehouse vendors.

Forrester Research makes a more cautious point. They say [6], "Hadoop may offer low-cost storage for data, but simply isn't fast enough to replace the EDW right now." But that may change over time. If SQL on Hadoop does make progress, then Hadoop will be winning.

Cloudera, the n° 1 Hadoop distributor, promotes the enterprise data hub (EDH) (Fig. 2). Many in the market believe that the EDH is poised to kill the EDW, but Cloudera disagrees. At the BBBT briefing, on May 9th, 2014, Alan Saldich, VP Marketing of Cloudera, called the statement about killing the EDW a myth:

[2] see http://www.b-eye-network.com/blogs/vanderlans/archives/2014/02/the_battle_of_t.php

[3] see http://www.information-management.com/blogs/big-data-vs-the-data-warehouse-10025458-1.html

[4] see "Performance Management and Analytics: BI meets Business Process Management and Big Data", Wolfgang Martin, August 2014 http://www.wolfgang-martin-team.net/BI-BPM-SOA.php

[5] see http://www.computerweekly.com/feature/Cloudera-v-Hortonworks-Hadoop-to-complement-replace-data-warehouse

[6] see same link as in foot note 3 http://www.information-management.com/blogs/big-data-vs-the-data-warehouse-10025458-1.html

"Most of our customers are building an EDH next to their EDW because they are complementary. See: Oracle BDA + Exadata." But as a matter of fact, Cloudera has already successfully replaced an ODS and a DW archive. It looks like Cloudera has taken a wait and see stance until the technology matures, and kills the EDW.

The New Way: Bringing Compute to Data

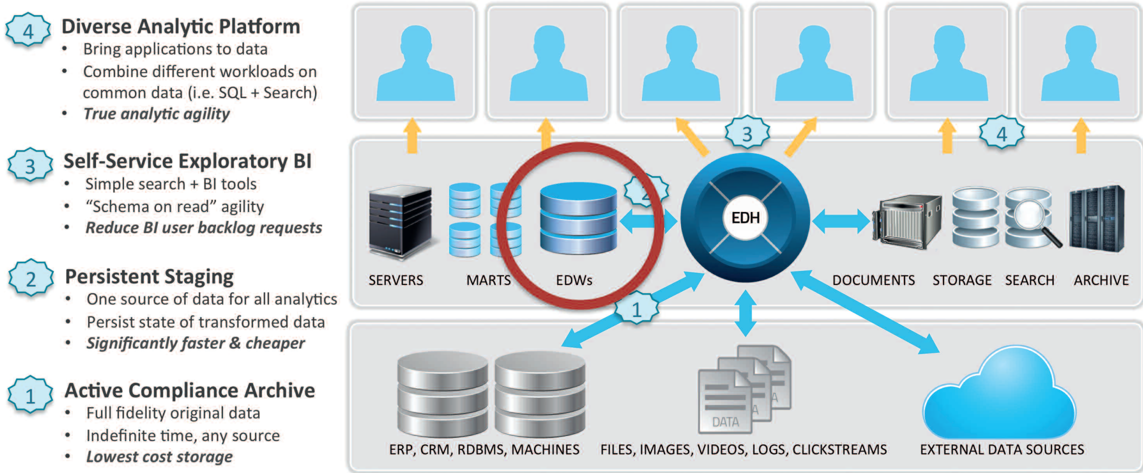


***Figure 2:** Cloudera believes that expanding data requires a new approach, a disruptive change from the traditional paradigm of "bring data to compute" to "bring compute to data". The EDH is the centerpiece of its data architecture (source: Cloudera, May 2014).*

Let us come back to Steve Miller's statement that analytics will move more and more to Hadoop, whereas the Data Warehouse will continue to support performance management (reporting, dashboards, financial controlling and planning). Performance management needs high quality master and transaction data. This is all about structured data at rest – with the exception of operational performance management. Therefore, operational performance management has mainly been processed outside the Data Warehouse. Performance management is based on a well-defined data model. Changes to the data model go with the speed of changing business models and business processes. This is still in the range of days, not hours or even faster. Furthermore, the data volumes used in performance management rarely exceed the terabytes range, whereas Hadoop can scale to handle petabytes. To summarize, relational database technology is still sufficient to meet the needs of tactical and strategic performance management, and a data warehouse offers a proven and powerful environment for managing important, business critical analytical data. For example, a data warehouse is more than adequate for compliance.

There can be exceptions to this general rule, for instance if a data warehouse does not include the level of detail that is needed for a particular performance management task. But then, the traditional solution is building a data mart with the required level of detail by using data integration and linking the necessary operational data to the corresponding data warehouse data. Data quality services can be embedded so that we end up with the necessary high quality data. If, furthermore, operational and analytical data need real-time processing for embedding analytics into business processes, then NoSQL technology is best suited to such data marts. Operational planning, disposition, and operational risk management are typical use cases. We will come back to such cases later.

Coexistence DW, Hadoop and NoSQL

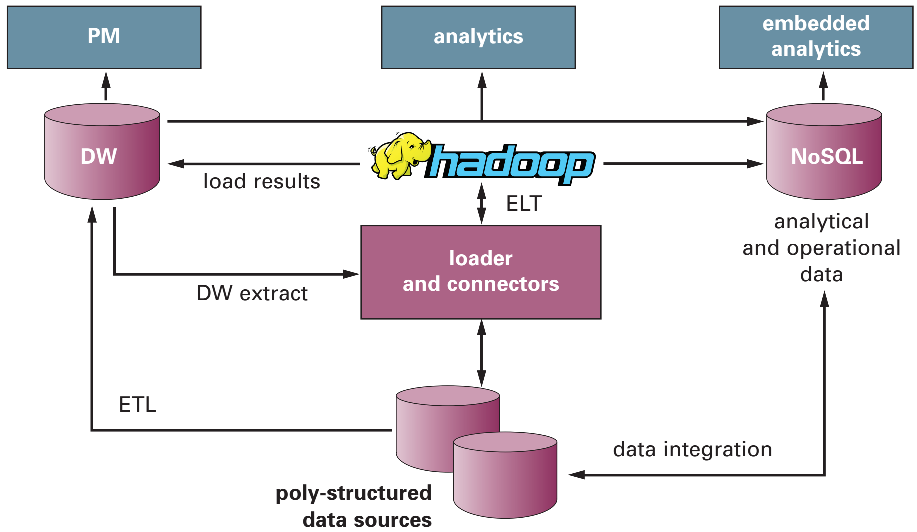


***Figure 3:** For the foreseeable future, Hadoop will not replace the Data Warehouse, but it does extend data warehouse architecture. The Data Warehouse will remain the central platform for tactical and strategic performance management as well as the source for the various data marts. However, an increasing number of new types of analytical workload will run on Hadoop and some NoSQL databases. This will particularly be the case when high performance is needed and/or data structures are highly complex. In such situations, data integration will link these data marts to corresponding operational systems, enabling operational performance management in real-time as well as embedding analytics into business processes. Analytics will increasingly run on Hadoop, and Hadoop will be connected to the Data Warehouse. Certain analytical results must be transferred to the Data Warehouse, and extracts from Hadoop will supply the Data Warehouse.*

Analytics comes with different needs and requirements. It uses structured and unstructured data. It uses data in motion. It uses high to very high volumes of data. It uses various Big Data sources and combines them with enterprise data. Furthermore, the purpose of analytics is to gain new, unknown insights into customers and markets as well as to enrich production, logistics and other business processes with embedded predictive analytics and operational BI. For such analytics, Cloudera's EDH concept makes sense. We put all relevant data from Big Data into Hadoop, and use data integration to link Hadoop data to the EDW and other enterprise data. Fig. 3 depicts the discussed scenarios.

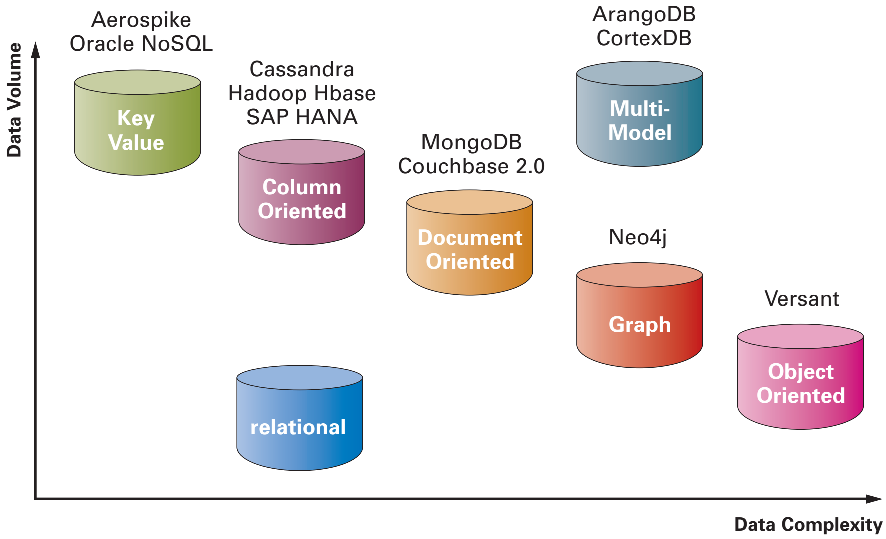
Take Away. The data warehouse will continue to be center and single point of truth for tactical and strategic performance management, at least for the foreseeable future. But ETL processing and some analytical workloads will move more and more to Hadoop, particularly if they involve analyzing semi-structured and unstructured data.

4. Hadoop and NoSQL Technologies

NoSQL data management systems enable the management and processing of poly-structured data. Thus, they complement the traditional relational DBMSs. It is therefore no longer the case that relational databases are the only option. The various methods and technologies used by NoSQL are not all new. Indeed, some of the NoSQL concepts and methods have been in use for many years. Now, Big Data is putting a new focus on them. NoSQL database technologies can be classified as shown in Figure 4.

Cloudera sees key value and document stores as well as GraphDBs as complementary to Hadoop. NoSQL key value and document databases are more suited to accelerating operational applications that require high volume write processing. Hadoop is much more suited to analytical workloads on complex data and also to batch ETL processing. It is not well suited to high performance write processing or to applications requiring single record update processing which NoSQL technologies support. Multimodal NoSQL database technologies like CortexDB also fall into the category of complementary technologies to Hadoop, because they include all the features and advantages of key value, document stores, GraphDBs and more.

NoSQL Data Management Systems



Relational DBMSs Do Not Solve All Data Problems.

Figure 4: *Classification of NoSQL database technologies and positioning according to data volume and data complexity. The mentioned products are only examples. An exhaustive list of NoSQL database technologies can be found at http://nosql-database.org/.*

But not all NoSQL technologies are seen as complementary to Hadoop. For example, Cloudera highlighted the overlap between Hadoop HBase and other NoSQL column-family oriented database technologies such as Cassandra at the BBBT briefing (May 9th, 2014).

Finally, let us look into the business cases where these NoSQL data base technologies shine and users can get clear and monetary advantages. Let us first look at just two examples [7]:

> ***Examples:*** In 90 days, MetLife built the kind of consolidated customer view it had dreamed about for nearly 10 years using a NoSQL database. Similarly, Constant Contact took three months to build a social marketing services application on NoSQL that would have required nine months to build on a conventional database and with higher levels of ongoing admin work.

If we look behind the scenes of such examples, we can derive the NoSQL advantages:

Elastic scaling. For relational database technologies, scaling was expensive because relational technology often requires specially designed hardware to scale out and cannot always scale out easily on commodity clusters. New breed NoSQL databases are designed with low-cost commodity hardware in mind and therefore have the advantage of a lower cost footprint. In addition, they can also expand transparently, taking advantage of new nodes when they are added to the cluster.

Volume handling. The volumes of data that are being stored have also increased massively. Today, the volumes of „big data" that can be handled by NoSQL systems outstrip what can be handled by the biggest relational data management systems.

Managing unstructured data. Unstructured data can be handled by relational database technologies, but if documents from different sources require flexible schema or no schema at all, or if input data needs to be kept in its original format, then NoSQL database technology comes into play. Furthermore, many NoSQL databases version each data element or each document. This enables queries for values at a specific time in history.

Fewer resources for management. Despite the many manageability improvements claimed by relational database technology vendors over the years, high-end relational systems can be maintained only with the assistance of expensive, highly trained DBAs. DBAs are intimately involved in the design, installation, and ongoing tuning of high-end relational database systems. NoSQL databases are generally designed from the ground up to require less management: automatic repair, data distribution, and simpler data models lead to lower administration and tuning requirements.

Economics. NoSQL databases typically use clusters of cheap commodity servers to manage the exploding data and transaction volumes, while relational databases tend to rely on expensive proprietary servers and storage systems. The result is that the cost per terabyte or transaction/second for NoSQL can be many times less than the cost of relational technology, allowing you to store and process more data at a much lower price point.

Flexible data models. Change management is a big headache for large production relational databases. NoSQL databases are easier to manage when data models change. For instance, NoSQL Key Value stores and document databases allow the application to store virtually any structure it wants in a data element. Even the more rigidly defined column oriented NoSQL databases, like Cassandra or Hbase, typically allow new columns to be created without too much effort. Furthermore, there are some NoSQL databases where application changes and database schema changes do not have to be managed as one complicated change unit.

[7] From Informationweek, Doug Henschen, June 2013 http://www.informationweek.com/big-data/big-data-analytics/nosql-vs-hadoop-big-data-spotlight-at-e2/d/d-id/1110260

Operational and analytical data converging. In the past, operational and analytical tasks have been strictly separated (Fig. 1). Users in charge of both tasks had to change systems. Furthermore, if the data warehouse did not provide the level of detail needed for special tasks, users typically had to invoke IT to get down to the necessary data sources. Some NoSQL database technologies can remove this barrier. Analytical and transactional systems can be merged, and analytical and operational data can be managed in one and the same database. This empowers new innovative systems that were not possible before. Typical use cases are operational planning, dispositive systems and embedded real-time analytics.

Take Away. NoSQL databases are becoming an increasingly important part of the database landscape, and can offer real benefits. They complement Hadoop and offer various advantages over relational database technologies. But relational technology will not be replaced by Hadoop and NoSQL. It will still have its place, but Hadoop complemented by NoSQL technologies offers the potential to handle Big Data analytics and to develop new innovative apps that were not possible by using relational technology.

5. CortexDB – a Complementary NoSQL Technology on top of Hadoop

As we have shown, certain NoSQL database technologies are complementary to Hadoop when sitting on top of Hadoop. They provide many advantages to users. Let us now have a closer look at one of these complementary technologies, CortexDB, and find out what makes CortexDB stand out.

CortexDB is a temporal, multimodal NoSQL database technology that provides an unlimited platform for individual enterprise web applications.

What makes CortexDB unique? It differs from all known databases via its index structure. Let us have a closer look. Different types of database management system are determined by their database schema. In the case of relational databases, data are organized in tables with columns and rows. Columns are nominated as unique identifiers and represent relationships across tables. Columns with the same domain are used to join tables. Other NoSQL databases (remembering that NoSQL = "Not only SQL") have a different database schema like key-value store, document store, etc. All these databases are different in the way they organize data. Also, in many cases, it is the application that defines the schema. The use of indices is common for all databases. Indices are flat structures of data with a reference to records (either table rows, document IDs etc.) so that they can access data in a sorted order. Hence, special indices for different types of data make up the main differences between databases, but their index structure is always flat.

CortexDB works differently. At first glance, it is a schemaless database, and one may compare it to document stores, but it is much more. It comes with a content based index structure (CorAIT [8]). This means that each item of content (value) knows in what kind of fields (keys) and in which records (Doc-ID) it exists, and every key knows what different types of values are used. So, CortexDB has a universal index of

[8] AIT = Active Index Transformation

all fields with all occurrences representing the whole database. Consequently, all data queries are based on the index structure without the need for joins, as the index structure includes the set of all possible joins for a given database schema. This also implies that a database schema is only used for output of data (data records or data documents). The memory consumption for all indices is about the same size as the size of a CSV file containing the raw data.

Other Features of CortexDB:

- CortexDB acts like multimodal NoSQL technology. It combines ALL the advantages of the various types of NoSQL databases (Key value; Document store; Graph DB; Multi value DB; Column DB). This is achieved by the index structure of CorAIT. That means that no special or user-defined index is used for fast data access, as the content is stored in schema-less mode inside CortexDB. All queries to the database run at the same speed as index databases – for all fields, combinations of fields, and even for linked information.

  - ***Example:*** When exploding a bill of materials for generic structures – such as in the automotive industry – this technology makes it possible to search recursively structured data quickly and efficiently, using any reference chains or attributes required. This enables users to determine the bill of materials for the vehicle concerned. A real-life case like this usually takes several seconds – or even minutes – but with CortexDB the result is displayed in a few milliseconds.

- In CortexDB, each value is represented by an actual value and optional data block (multi-value option). The server also offers the option of virtual fields that can be added to data records (documents), and a "Reporter" functionality that solves the problem of transactional data consistency. Last but not least, CortexDB offers a special knowledge functionality to solve the problem of data locking while editing data records (documents).

- Furthermore, Uniplex defines a database schema based on CortexDB to handle real data with defined field and record types, links and pointers as well as field based temporal data. It uses descriptions of field and record types (document types) to organize data in nested structures. The powerful integrated list functionality provides the user interface for data presentation. That includes the linking of data records in unlimited structural depth including recursively linked data. Complex database queries can be defined based on lists representing data joins. These lists, for instance, can be used as input for pivot tables and dashboard graphics. Object viewers can be configured to show all the necessary data in the nested data structures of a data object that is not directly linked to a record and should be jointly presented by multiple lists. As an example, in financial services, take a person as a data object that can now easily be linked to all his/her investments, stakes in organizations, real estate etc. across various data sources and applications. Uniplex deals with data in json structures (stacked arrays of objects) which can be accessed either by JavaScript, php, C++ or Java. An implemented role system guarantees that each user only has access to the data he or she is authorized to use.

- CortexDB is a temporal database technology. A temporal database includes valid time and transaction time. CortexDB also combines these attributes to form bi-temporal data. Temporal databases are more powerful than traditional databases that ensure only the truth at the transaction time and ignore the validity period.

  - ***Example:*** In master data management, users need a validity date ("valid from...") of information in each field in a data set. That enables them to view and track back each change in any time-related context they choose. As a result, the database knows exactly how all the information has evolved over time, including both past and future values. Furthermore, the management of event data is necessary in many use cases. Event data is data that is only valid at a given point in time, such data collected via meters (electricity, gas, water etc.). Such a feature considerably facilitates the management of redundant temporal data.

- CortexDB is a distributed database technology that runs on Linux, Windows and MacOS. The CortexDB has also been ported to Android and ARM systems like Raspberry Pi.
  - Distributed databases use master/slave synchronicity. A slave server only receives filtered data, enabling dedicated servers to be brought in for special tasks. This ensures data integrity and increases security.
- CortexDB includes a sophisticated security concept that can be activated automatically, even for in-house applications. Software developers can take advantage of this function, saving them the extra work in their own source code.
- CortexDB can be delivered on-premise or via a cloud model. Cortex AG provides server capacity based in the datacenter of a German provider, enabling customers, partners and Cortex itself to run CortexDB (and applications based on it) as cloud solutions. Large enterprises and other companies wishing to operate the database in-house are free to do so.

CortexDB Advantages: CortexDB provides unique technical features delivering various benefits:

- Flexibility to change the database schema as required by business departments and software developers – the system adapts to the processes rather than the other way round
- Extremely high database performance on standard hardware (low footprint)
- Rapid and agile application development across innovative data services without programming
- Simple modeling of complex structures
- Change requests on the fly, enabling self-service usage by business units

CortexDB Use Cases:

CortexDB is often used for hitherto unsolved database problems and for those that are very complex to solve. However, its ease of use has led to companies increasingly adopting it for “simple” tasks and new developments.

Developers who are frustrated by the limits imposed by conventional relational databases (capacity, complexity, speed, costs, amount of work involved etc.) find that CortexDB makes their everyday work significantly easier once they have learned how to use it.

As multiple tasks can be solved by the database itself and by simple configuration settings, developers only need to concentrate on their real work, not on optimizing database queries, the permissions system or other system-related tasks.

Departments with problems that can usually only be handled by the IT department find that CortexDB and applications based on it are the right self-service tools to help them work autonomously. One of the main reasons for this is that configuration settings can deal with multiple tasks and queries that previously required the assistance of an administrator (see also Fig. 5).

CortexDB is used by midsized and large companies across all industries for a wide range of different purposes. They include both small and large amounts of data with complex structures.

Some solutions based on CortexDB:

- Object dependencies in the automotive industry
- Coordination of bills of materials between construction, logistics and production
- Disposition of staff, equipment and orders in the surveillance industry
- Validation of multilingual master data at a pan-European seed production company, in order to establish the comparability of similar products between different countries
- Interactive utilization for data capture and semi-automatic data administration in the insurance industry, in order to improve data quality and maintain it long term.

Dealer Performance Management

Challenge

Data on which customers, products, services and processes contribute most to business performance

Retail needed KPIs for division management

ERP and SAP FI/CO unable to deliver analyses in the required form

No access to detailed data for operational and strategic analyses in previous systems

Solution

CortexDB integrates all data sources in one DW

Management cockpit delivers graphical KPIs

All division managers receive personalised KPIs with drill-downs to raw data and historical information

Business analytics and campaign management for CRM

Process monitoring with Email Reporter

Benefits

Customer retention with ABC analyses: customers, products, services and processes

Service effiency, optimised turnaround times and operational breakeven analysis

Financial controlling: target/actual comparisons show goal achievement and enable benchmarks in retail division

CortexDB – trancparency throughout the business

© Cortex AG, January 2014

Figure 5: *Another use case for CortexDB: it begins where the data warehouse ends (see chap. 2). CortexDB, used as database technology for a data mart, provides the necessary operational data details that are missing in the data warehouse and adds the speed and performance needed to process high volume of data at low costs. Analytical and operational data are managed in one and the same database.*

Take Away. CortexDB is a NoSQL technology complementing Hadoop when sitting on top of it. Networked systems, data dependencies, complex configurations, very large data volumes, constant changes – requirements like these cannot be managed by Hadoop alone, but need a temporal, schema-less multi-model NoSQL DB. In other words:

CortexDB is the solution for managing large quantities of complex poly-structured data in Hadoop data architectures.

6. Appendix

About Cortex AG

Cortex AG specializes in innovative NoSQL database technologies for processing large and complex data volumes. It provides an integrated family of products, suited to any type of enterprise application, that enable customers to create applications for data warehouses, business intelligence, business analytics, PDM, MES, CRM , CMS, etc. with no need for programming.

The product family is centered on CortexDB, which is a bitemporal multi-model database. It stores data in a completely new way and enables powerful database queries on any attributes as well as combinations of attributes and the relationships between them. Benefits to customers include agile software development, analytical and transactional data in the same database, change requests on the fly, self-service by business departments, and low-level hardware requirements.

Further information is available at www.cortex-ag.com

About the author

Dr. Wolfgang Martin is a leading European authority on:

- Business Intelligence, Analytics, Big Data, Performance Management
- Business Process Management, Information Management and Governance,
- Cloud Computing (SaaS, PaaS)

He focuses on technological innovations that drive business, examining their impact on organization, enterprise culture, business architecture and business processes. He is partner of iBonD (www.iBonD.net), member of the Boulder BI Brain Trust (http://boulderbibraintrust.org/), works as an advisor at the Institute of Business Intelligence at the Steinbeis Hochschule Berlin (www.i-bi.de). In 2001, Info Economist magazine declared him one of the top 10 most influential IT consultants in Europe.

Dr. Martin is a notable commentator on conference platforms and in TV appearances across Europe. A frequent contributor of articles for IT journals and trade papers, he is also the author of the Strategic Bulletins focusing on BI and Big Data (www.it-research.net) and an editor of technical literature.

Further information can be found at www.wolfgang-martin-team.net