Reshaping Smart Businesses with Cloud Database Solutions

Bogdan NEDELCU, Andreea Maria IONESCU, Ana Maria IONESCU,
Alexandru George VASILE
University of Economic Studies, Bucharest, Romania
bogdannedelcu@hotmail.com, andreea_maria_ionescu@yahoo.com,
anamaria.ionescu27@gmail.com, alexandru.vasile26@gmail.com

The aim of this article is to show the importance of Big Data and its growing influence on companies. We can also see how much are the companies willing to invest in big data and how much are they currently gaining from their big data. In this big data era, there is a fiercely competition between the companies and the technologies they use when building their strategies. There are almost no boundaries when it comes to the possibilities and facilities some databases can offer. However, the most challenging part lays in the development of efficient solutions - where and when to take the right decision, which cloud service is the most accurate being given a certain scenario, what database is suitable for the business taking in consideration the data types. These are just a few aspects which will be dealt with in the following chapters as well as exemplifications of the most accurate cloud services (e.g. NoSQL databases) used by business leaders nowadays.

Keywords: *Smart business, cloud database, cloud solutions, NoSQL databases, weaknesses, key-value databases, Riak, columnar databases, Hbase, document-oriented databases, MongoDB, graph databases, Neo4j, cloud services*

1 Introduction

It is becoming increasingly clear that the cloud technology brings a different outlook to the digital world, making it easier to understand, powerful and more efficient. Not only is the everyday consumer affected by these continuously shifts in technology, but also the businesses around the world. What the business leaders should bear in mind is that their organizational strategies must keep up with the new trends in technologies, and, therefore make use of the cloud in order to better manage their resources, improve internal performance among with the necessary IT-related knowledge that helps accomplish success.

The companies should start building their strategies by using the latest trends in technology considering the high applicability of the applications within cloud as well as the benefits which could arise when using cloud integrated solutions. As a matter of fact, cloud computing sometimes could be the most effective answer to common problems that the businesses usually encounter such as the reduction of the costs, the implementation of new services and applications. Companies can add computational capacity faster using the cloud than they usually can using in-house stuff. Nowadays, the business models tend to gain access to services instead of have ownership of products. For example, Microsoft Office 365 and Google Apps provide applications that are available over the Internet (instead of via traditional software packages that must be purchased and installed). [1]

Despite the fact that the cost of implementing cloud setups has reduced and the performance improved significantly, the cloud reliability has left a question mark over, especially because most of the enterprises have concerns about placing sensitive data on a third-party cloud. The evolution of the cloud and how far has this trend gone in terms of time and space are also some subjects

which will be tackled within the following pages. “The Long Nimbus” article released by the Economist magazine about the impact of cloud computing on company organization structures states that ‘‘Businesses are becoming more like the technology itself: more adaptable, more interwoven and more specialized. These developments may not be new, but cloud computing will speed them up.” [2]

The Riak, Apache HBase, MongoDB, and Neo4J cloud databases services are just a few examples that will be presented in this article. It is particularly important when creating a business strategy to understand the capabilities and constraints of each type of the databases in order to choose the right one for the specific job. The focus should be more on whether a particular database is suitable when considering a business problem space, the usage patterns as well as the available resources.

Overall, one of the major benefits that the companies can gain by using the cloud technology comes not from cost savings for IT resources on a per-use basis, but from the revenue they earn by becoming more flexible and responsive when it comes to customers’ changing needs. This would further enable businesses efficiently deliver their new products and services as well as expand successfully into new markets.

“While enterprise IT use will continue to grow, the largest source of economic impact through 2025 will likely come from enabling the delivery of services and applications to Internet users. We estimate the total potential economic impact for cloud technology across sized applications could be $1.7 trillion to $6.2 trillion in 2025.”3 Taking into consideration this huge potential for the global economy and the fact that the cloud technology would definitely reshape the world through its fast changing development, the smart corporate business should come up with strategies using the tools of the cloud services that would boost both their productivity and performance.

2. The Cloud Concept

The cloud is not simply the latest fashionable term for the Internet. Though the Internet is a necessary foundation for the cloud, the cloud is something more than the Internet. The cloud is where you go to use technology when you need it, for as long as you need it, and not a minute more. You do not install anything on your desktop, and you do not pay for the technology when you are not using it.

NIST (National Institute of Standards and Technology) defines cloud computing as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” [4]

Nevertheless, confusion remains about exactly what it is and when it's useful, causing Oracle's CEO Larry Ellison to vent his frustration: "The interesting thing about cloud computing is that we've redefined cloud computing to include everything that we already do.... I don't understand what we would do differently in the light of cloud computing other than change the wording of some of our ads." [5]

Cloud computing is a computing platform that resides in a large data center and is able to dynamically provide servers with the ability to address a wide range of needs, from scientific research to e-commerce. The provision of computing resources as if it were a utility such as electricity, while potentially revolutionary as a computing service, presents many major problems of information policy, including issues of privacy, security, reliability, access, and regulation. This article explores the nature and potential of cloud computing, the policy issues raised, and research questions related to cloud computing and policy. Ultimately, the policy issues raised by cloud computing

are examined as a part of larger issues of public policy attempting to respond to rapid technological evolution [6].

2.1 Short History

When we think of Cloud Computing, we think of situations, products and ideas that started in twentieth century. Historically, behind all of these modern concepts it is another story.

It all begun with a gradual evolution that started in the 1950s with mainframe computing. Multiple users could access a central computer for dumb terminals, whose only function was to provide access to the mainframe. Because of the costs to buy and maintain mainframe computers, it was not practical for the organizations to buy and maintain one for every employee. The best solution to save money, in this complicated piece of technology was to share access to a single resource.

Around 1970, the virtual machine (VMs) was created. Using virtualization software like VMware, it became possible to execute one or more operating systems simultaneously. This kind of operating system took the 1950s shared access mainframe to the next level, permitting multiple distinct computing environments to reside on one physical environment. Virtualization came to drive the technology, and was an important catalyst in the communication and information evolution.

Telecommunications companies only offered single dedicated point–to-point data connections. The newly offered virtualized private network connections had the same service quality as their dedicated services at a reduced cost. Instead of building out physical infrastructure to allow for more users to have their own connections, telecommunications companies were now able to provide users with shared access to the same physical infrastructure. Nowadays, SoftLayer is one of the largest global providers of cloud computing infrastructure which was founded in 2005, but was acquired in 2013 by IBM to form IBM Cloud Services Division. In 2011 the company reported hosting more than 81,000 servers for more than 26,000 customers. IBM already has platforms in its portfolio that include private, public and hybrid cloud solutions. The purchase of SoftLayer guarantees an even more comprehensive infrastructure as a service (IaaS) solution. While many companies look to maintain some applications in data centers, many others are moving to public clouds.

In the end, the story is not finished here. The evolution of cloud computing has only begun and lead us to a widespread area. Even if companies are creating their own internal cloud called “private” or others are moving to clouds from external services known as “public”, this process of “moving” is the most profound evolution and produce significant changes in the way they run.

2.2 Cloud Services

There are three models of cloud services:

- SaaS (Software as a Service)

It refers to the capability of the clients to access the provider’s applications which are running on a cloud infrastructure. The applications are available from various client devices, through a simple client interface, like a web browser, or an interface of the program. Examples: Email services provided by big companies like Microsoft (Hotmail), Google (Gmail), or Yahoo! (Yahoo Mail).

- PaaS (Platform as a Service)

It refers to the ability of customers to install their applications (created or acquired) on the cloud infrastructure using programming languages, libraries, services and tools provided by the supplier.

Allows access to information about new software, given low cost and preset distribution channels to attract more

efficient the customer, which proves that the cloud is a way to improve your business strategy.

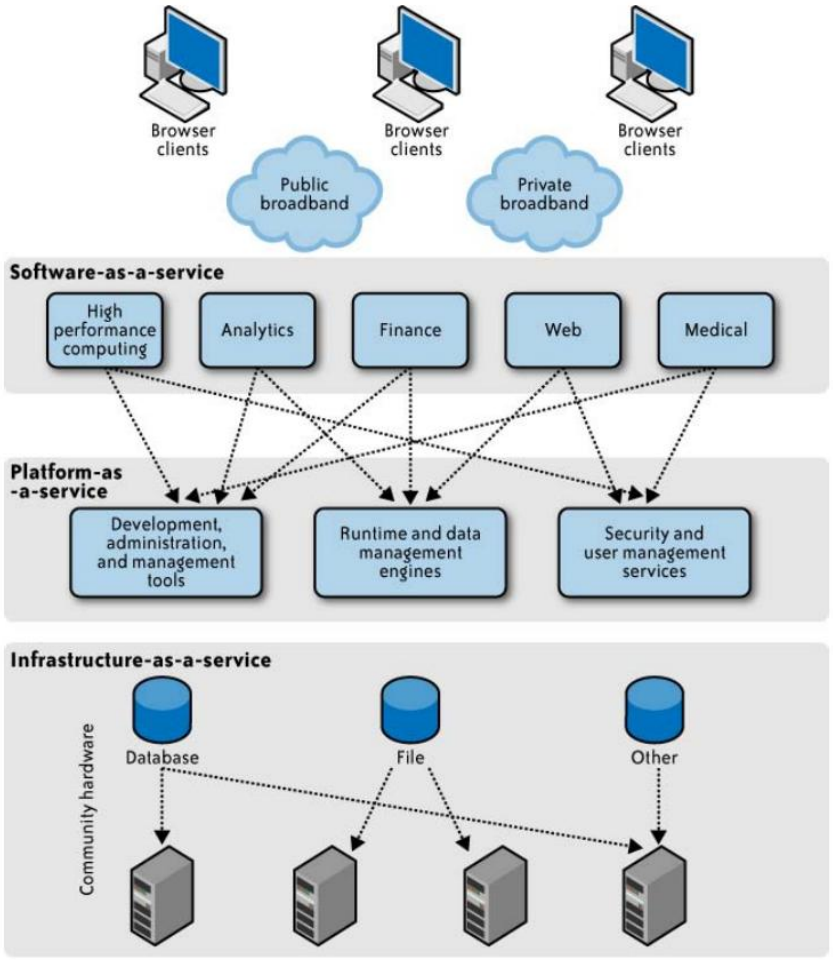


Fig 1. Models of cloud services [8]

- IaaS (Infrastructure as a Service)

It refers to the customers' capability to use processing power, media storage, networking and other basic computing resources from the provider to install and run operating systems, applications and other software on a cloud infrastructure.

The basic technique is virtualization (the ability to provide a consistent view on a set of resources), namely: virtualization of servers, equipment storage or networks.

IaaS services provided are: interface for resource management and interface for system monitoring.

Cloud computing is not so much a technology as it is the combination of many preexisting technologies. These technologies have matured at different rates and in different contexts, and were not designed as a coherent whole; however, they have come together to create a technical ecosystem for cloud computing. New advances in processors, virtualization technology, disk storage, broadband Internet connection, and fast, inexpensive servers have combined to make the cloud a more compelling solution.

When it comes to the usage of databases as a cloud service, they are more than columns and rows. It is not compulsory to know the profile of a business and the field it operates, but the most important aspect is the way the collected data is stored. Even though it might be a mailing list or an extensive product inventory, the type of database you chose might be beneficial or harm the way data is structured.

Although the businesses today focus their competitive advantage on how fast they

react when making a decision, sometimes this solution does not guarantee the proper answer to the complex business issues they were dealing with. In this fiercely and challenging economic environment, collecting more and more data and accessing it as fast as possible is mandatory especially because the transformation of data assets into innovative strategies can most of the times maximize the productivity of resources, and therefore lead to sustainable growth.

Most companies today have plenty of data. Creating intelligence and gleaning real insight and value from this data is what continues to elude organizations. Despite years of talk about scorecards and metrics, gut feelings and experience are often still the guides for making important, sometimes critical decisions. [7] As James Taylor, the CEO of Decision Management Solutions, states "organizations need to be much more focused on directing analysts towards business problems. Find the decisions that are going to make a difference to business results..." [8]

In the past, companies have been using relational databases to store their structured data. Today, even though they had a huge impact on the world of databases and unlocked data for many applications, relational databases lack in characteristics necessary to cope with the fast-changing transaction of data in the big data era.

NoSQL databases are the answer that solves many of these problems because they make you see the database world in a new light. In 1998 Carlo Strozzi first mentioned the NoSQL to name his lightweight, open-source relational database which did not convey with the standard SQL interface. However, the term became popular in the late 2009 when people realized how beneficial they were. Since then, the NoSQL movement has continuously developed. Although they have no schema, the NoSQL databases are fast and adapt easily to the businesses leaders' needs. For example, the NoSQL can work with the non-relational distributed and unstructured data which is the type of data most of the companies typically collects. [1]

The NoSQL databases are particularly known for scalability, agility and flexibility. There are different types of NoSQL databases and each focuses on different applications: Key-value stores, Wide-column (Columnar) stores, Document database, and Graph store.

The Key-value (KV) stores the pairs of keys to values in approximately the same way that a map (or hashtable) would in any popular programming language. They might have different functionality: some KV implementations might provide a means of iterating the keys, while others can allow complex value types such as hashes or lists. For example, a filesystem can be considered a key-value store, if you think of the file path as the key and the file contents as the value. Although databases of this type can be incredibly performant, they would lack the capacity to manage complex query and aggregation needs.[2]

The columnar (column-oriented) databases took their name from a particular feature in their design – data from a given column (in the two dimensional table sense) is stored together. On the other hand, a row-oriented database (like an RDBMS) is used to keep information about a row together. Although it is hard to see the difference, the impact of this design decision has a more deep meaning than it seems. An inexpensive feature of the column-oriented databases is that adding columns is done a row-by-row basis. It is not compulsory for a row to have a set of columns, the tables would still remain sparse without paying the cost of storage for null values. With regard to the structure, the columnar type of databases is about midway between relational and key-value types.

As its name sounds, the Document-oriented databases store documents. Basically, a document resembles a hash, it has a unique ID filed and its values can be

any of a variety of types, including more hashes. Documents can permit a high degree of flexibility because they have the possibility to form nested structures. Moreover, there are few restrictions on incoming data imposed by the system. The basic condition it has to meet is to be expressible as a document. Depending on what type the document database is, it can have different approaches regarding the indexing, ad hoc querying, replication, and consistency. Only by understanding these differences and the impact of them on the documents can lead to a wise decision and, thus, contribute to the creation of a feasible business strategy.

Less commonly used, the Graph databases are usually considered the best at coping with the highly interconnected data. Their structure is made up from two elements: nodes and the relationship between the nodes. Both elements have a variety of properties – key-value pairs – that store data. Of course, the real strength of graph databases lays in the capacity to travel the nodes by following relationships.

"A variety of NoSQL databases are available, each intended to focus on a particular data storage and access strategy. While a typical NoSQL database might not be as comprehensive as a relational database, its focus on a well-defined range of tasks enables it to be highly optimized for those tasks. The key to success is to understand the features of different NoSQL databases, and then use these features to implement a repository that matches the specific requirements of your applications."[3]

3. The Cloud Computing impact on the business

3.1 Cloud Movement

Cloud computing can be considered primarily as a cost-saving technology that's used here and there on cost-cutting projects and for quick fixes to provide point solutions to specific operational problems. On the other hand, cloud computing can be understood in the context of an overall business strategy based on agility and responsiveness. Cloud computing certainly provides cost savings in some situations, but cost savings is not the most important benefit. The real value of cloud computing is the way in which it can be used to support an overall strategy designed to create agility for the business.

The spread of cloud computing is the best example of "creative destruction". The phenomenon was popularized by the economist Joseph Schumpeter who described it as the "process of industrial mutation that incessantly revolutionizes the economic structure from within, incessantly destroying the old one, incessantly creating a new one."

Why move to the cloud? There are plenty of good reasons, but mainly it makes good business sense: cloud computing lets you focus on what's important, your business. This is called efficiency. This field can be used for almost all types of applications and it is clear that it saves its users money.

First of all, the hardware is fully utilized. Cloud computing brings natural economies of scale. The practicalities of cloud computing mean a high utilization and smoothing of the inevitable peaks and troughs in workloads. Sharing sever infrastructure with other organizations, allows the cloud-computing provider to optimize the hardware needs of its data centers, which means lower costs for business.

Secondly, when you run your own data center, your servers won't be fully. Idle servers waste energy, so a cloud service provider can charge you less for energy used than you're spending in your own data center. In conclusion, power costs are lower. When you run your own servers, you're looking at up-front capital costs. But in the world of cloud-computing, financing that capital investment is someone else's problem. Sure, if you run the servers yourself, the accounting wizards do their amortization magic which makes it appear that the cost gets spread

over a server's life. But that money still has to come from somewhere, so it's capital that otherwise can't be invested in the business—be it actual money or a line of credit. Moving to the cloud will save you money, not just for your cloud security needs, but for many other types of data center workloads.

To sum up, Bernard Golden, CEO of HyperStratus, draws an insightful analogy between the early adoption of the Internet by business and the growing business use of cloud computing:

"At a certain point in time, the technology vendor community, especially startups, just caught fire about the Internet. They were convinced that, once experienced, no one could avoid adopting their work lives to the Internet. At that same point in time, mainstream IT looked at the Internet with a skeptical eye, focusing on its shortcomings. At that time, I heard statements like ''nobody is going to let their data cross insecure public networks'' and ''Nobody is going to put real business functionality out on the Web.'' Of course, the indisputable benefits of the Internet overwhelmed the dubious responses. As we look back now, the chaos and cynicism is hard to remember, but believe me, it was there—and strong. But those attitudes didn't stand a chance against easy access to information, and I think it's unlikely that a jaundiced view of cloud computing is going to prevail, either." [4]

3.2. A Cybernetic Economy

The size of a company can be measure by the number of contractual relations it creates and by the number managed internally versus externally. But because of the expansion of the wireless Internet, mobile computing and business application services delivered over the Internet, it is becoming easier and less expensive to manage external contractual relationships and transactions. The original organization structure of twentieth century companies was modified and optimized for outside-in communications. This new change is described by the co-chair of the President's Information Technology Advisory Committee:

"Since we can now use technology, the Internet and open standards to begin to automate, standardize and integrate business processes, those transaction costs described by Ronald Coase are dropping precipitously. Consequently, the whole nature of the firm, and what it means to run an efficient business, is going through very extensive changes. These are not easy changes. Not only is there a great deal of innovation required to automate and integrate business processes, but perhaps more important, there are even greater changes in culture required to transform Industrial Age business models to something more appropriate to our Internet era." [5]

One of the companies that changed the old structure is Cisco Systems. In 2002, the organization hit hard in the collapse of the dot-com bubble when their stock went from around $77 a share to around $11. [5] This step was a wake-up call and the company took it as an opportunity. Cisco could learn some lessons and changed its structure, the traditional pyramid-shape corporate hierarchy where all the decisions were made by a small group of senior executives, with a network organization structure which is an efficiency one. Now, the decisions are made by people who have the authority to figure out what is happening and are powered by Internet-based collaborative technologies like blogs, wikis and social media tools.

Cisco's CEO John Chambers makes the case that Cisco's new business model is ''the best possible model for how a large, global business can operate: as a distributed idea engine where leadership emerges organically, unfettered by central command.''

Jeremy Rifkin is a senior lecturer at the Wharton School's Executive Education Program and has spent 10 years as an advisor to the European Union. He is president of the Foundation on Economic

Trends and author of several bestselling books on the impact of scientific and technological changes on the economy, the workforce, and the environment. One of the bestselling books is The Empathic Civilization. The author considers the latest phase of communication and energy regimes as bringing people together, but also creating environmental problems. He calls it an industrial revolution and also the business models are "cybernetic, not linear". Nowadays, business is about accessing the services instead of owning the products. For example, customers buy membership that provides them access to information, instead of purchasing them.

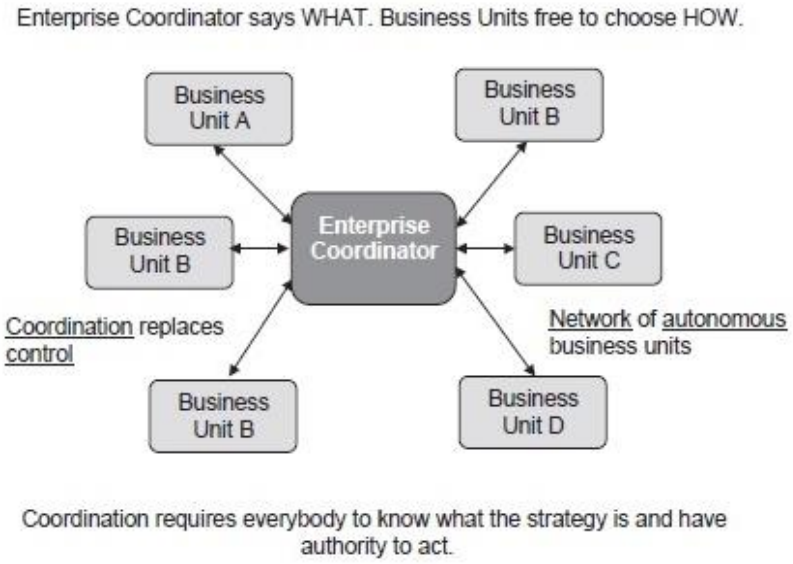


Fig 2. New Organization Structure [5]

The science of cybernetics describes the control and communication processes that work best for network organizations. So familiarity with some basic principles of cybernetics is helpful in exploring how responsive network organizations operate.

4. Famous NoSQL databases

4.1. Key-value Database - Riak

The key/value databases do not concentrate on structure of the data, but on the ability to store and retrieve that data. As a consequences, queries are more efficient and quicker to implement, lending themselves fast scalable applications that are supposed to read and write a wide and variety of data in this big data era.

This key/value type database allow clients to read and write values using a key as follows:

- Get(key), returns the value associated with the provided key.
- Put(key, value), associates the value with the key.
- Multi-get(key1, key2,.., keyN), returns the list of values associated with the list of keys.
- Delete(key), removes the entry for the key from the data store. [6]

Riak is a distributed key-value where values can be anything – from plain text to text, JSON, or XML to images or video clips - all accessible through a simple HTTP interface. It does not matter what type of data you have, the Riak database can store it.

Riak is also fault-tolerant. This means that servers can fluctuate at any moment without giving any notifications or reasons why they have failed. Your cluster continues humming along as servers are

added, removed, or (ideally not) crash. One of the big advantages is that with Riak you do not have to worry about your cluster because, even though, a node failed, is not an emergency and you do not need to solve the problem right away.
However, this flexibility also has disadvantages. Riak lacks robust support for ad hoc queries, and key-value stores, by design, have trouble linking values together (in other words, they have no foreign keys).
Riak is a great choice for datacenters like Amazon that must serve many requests with low latency. If every millisecond spent waiting is a potential customer loss, Riak is hard to beat. It's easy to manage, easy to set up, and can grow with your needs. If you've ever used Amazon Web Services, like SimpleDB or S3, you may notice some similarities in form and function. This is no coincidence. Riak is inspired by Amazon's Dynamo paper. [7]
Riak allows us to control reads and writes into the cluster by altering three values: N, W, and R. N is the number of nodes a write ultimately replicates to, in other words, the number of copies in the cluster. W is the number of nodes that must be successfully written to before a successful response. If W is less than N, a write will be considered successful even while Riak is still copying the value. Finally, R is the number of nodes required to read a value successfully. If R is greater than the number of copies available, the request will fail. [8]
There is a great difference between the relational databases and Riak. The absence of transactions, of the SQL and of the schema leads to difficulties in understanding and usage of the Riak database. Although there are keys, the procedure of linking between buckets is not like a table join. However, some problems can better be solved using Riak because it has the ability to adapt to increase demands of the servers in terms of performance rather than increase in size for larger single servers which, thus, helps to solve the unique scalability problems of the Web. What is more, Riak transmits data bi-directionally when considering the HTTP structure, allowing maximum flexibility for any framework or web-enabled system.

Riak's Strengths:
One of Riak's strengths refers to the fact that it removes the possibility of failure and supports maximum uptime and grow (or shrink) to meet changing demands. It does not matter if your data is complex or not, Riak can store both simple data and allow you to introduce sophisticated information if needed. There are many client libraries for Riak, including Java, Python, Perl, Erlang, Ruby, PHP, .NET, and many others. [9] If you are in the case in which you need more speed than HTTP can handle, communicating via Protobuf [10], might be a better solution because it is a more efficient binary encoding and transport protocol.

Riak's Weaknesses:
There are several drawback when it comes to Riak. There are features things which this type of databases cannot support, for example simple queryability, complex data structures, a rigid schema or the possibility to scale horizontally with your servers.19
One of the major disadvantage about Riak is it the fact that the querying framework remained the same - easy and robust ad hoc. Although, the MapReduce provides strong functionality, there should have been more built-in URL-based or other PUT query actions. Finally, if Erlang is not your favorite programming language, there are few limitations when using JavaScript, such as the unavailability of post-commit or the execution of MapReduce is slow.

4.2. Column-Oriented Database – Hbase

HBase is a column-oriented database management system that runs on top of HDFS. It is perfect for sparse data sets, which are common in many big data use cases. HBase does not support SQL, which

is a structured query language. HBase applications are written in Java much like a typical MapReduce application. HBase does support writing applications in Avro, REST, and Thrift.
An HBase system contains a set of tables. Each table contains rows and columns, much like a traditional database. Each table must have an element defined as a Primary Key, and all access attempts to HBase tables must use this Primary Key. An HBase column represents an attribute of an object. In fact, HBase allows for many attributes to be grouped together into what are known as column families, such that the elements of a column family are all stored together. This is different from a row-oriented relational database, where all the columns of a given row are stored together. With HBase you must predefine the table schema and specify the column families. [12]

Key Features of HBase:

- Scale-out Architecture - add servers to increase capacity
- Automatic Sharding - Transparently and efficiently scale out your data across machines in the cluster
- Full Consistency - Guard against node failures or simultaneous writes to the same record
- Active-active Replication - Stream data across locations for disaster recovery and data protection
- High Availability - Multiple master nodes ensure continuous access to data
- Full-text, Faceted Search - Give non-technical users and your applications a familiar yet powerful, interactive search experience [13]
- Security - Secure table and column family-level access via Kerberos
- SQL Access - Query data interactively with Cloudera Impala and for batch processing with Apache Hive

HBase's Strengths:
Noteworthy features of HBase include a robust scale-out architecture and built-in versioning and compression capabilities. For example, keeping the version history of wiki pages is a crucial feature for policing and maintenance. By choosing HBase, we don't have to implement page history—we get it for free.
Talking about performance, HBase is meant to scale out. If you work with large amounts of data, measured in many gigabytes or terabytes, HBase may be for you.

HBase's Weaknesses:
Although HBase is designed to scale out, it doesn't scale down. The Hbase community seems to agree that five nodes is the minimum number you'll want to use. Because it's designed to be big, it can also be harder to manage. Solving small problems isn't what HBase is about.
HBase doesn't offer any sorting or indexing possibilities except the row keys. Rows are kept in sorted order by their row keys, but no such sorting is done on any other field, such as column names and values. So, if you want to find rows by something other than their key, you need to scan the table or maintain your own index. Another missing concept is datatypes. There is no distinction between, say, an integer value, a string, and a date. They're all bytes to HBase, so it's up to your application to interpret the bytes.[14]

4.3. Document-Oriented Database - MongoDB

MongoDB is in many ways like a power drill. Your ability to complete a task is framed largely by the components you choose to use (from drill bits of varying size to sander adapters). MongoDB's strength lies in versatility, power, ease of use, and ability to handle jobs both large and small. Although it's a much newer invention than the hammer, it is

increasingly a tool builders reach for quite often.
First publicly released in 2009, MongoDB is a rising star in the NoSQL world.
It was designed as a scalable database—the name Mongo comes from "humongous"—with performance and easy data access as core design goals. It is a document database, which allows data to persist in a nested state, and importantly, it can query that nested data in an ad hoc fashion. It enforces no schema (similar to Riak but unlike Postgres), so documents can optionally contain fields or types that no other document in the collection contains.23
MongoDB Features: focuses on flexibility, power, speed, and ease of use:
Flexibility - MongoDB stores data in JSON documents (which we serialize to BSON). JSON provides a rich data model that seamlessly maps to native programming language types.
Power - MongoDB provides a lot of the features such as secondary indexes, dynamic queries, sorting, rich updates, upserts (update if document exists, insert if it doesn't), and easy aggregation.
Speed/Scaling - By keeping related data together in documents, queries can be much faster than in a relational database where related data is separated into multiple tables and then needs to be joined later. MongoDB also makes it easy to scale out your database. Autosharding allows you to scale your cluster linearly by adding more machines. It is possible to increase capacity without any downtime, which is very important on the web when load can increase suddenly and bringing down the website for extended maintenance can cost your business large amounts of revenue.
Ease of use - MongoDB works hard to be very easy to install, configure, maintain, and use. To this end, MongoDB provides few configuration options, and instead tries to automatically do the "right thing" whenever possible. This means that MongoDB works right out of the box, and you can dive right into developing your application, instead of spending a lot of time fine-tuning obscure database configurations. [24]

Mongo's Strengths
Mongo's primary strength lies in its ability to handle huge amounts of data (and huge amounts of requests) by replication and horizontal scaling. But it also has an added benefit of a very flexible data model.
Finally, MongoDB was built to be easy to use. You may have noticed the similarity between Mongo commands and SQL database concepts (minus the server-side joins). This is not by accident and is one reason Mongo is gaining so much mind share from former object-relational model (ORM) users. It's different enough to scratch a lot of developer itches but not so different it becomes a wholly different and scary monster.

Mongo's Weaknesses
How Mongo encourages denormalization of schemas (by not having any) might be a bit too much for some guys to take. It can be dangerous to insert any old value of any type into any collection. A single type can cause hours of headache if you don't think to look at field names and collection names as a possible culprit. Mongo's flexibility is generally not important if your data model is already fairly mature and locked down.
Because Mongo is focused on large datasets, it works best in large clusters, which can require some effort to design and manage. Unlike Riak, where adding new nodes is transparent and relatively painless for operations, setting up a Mongo cluster requires a little more forethought. [15]

4.4. Graph Database – Neo4J

Neo4j is a new type of NoSQL datastore called a graph database. As the name implies, the data stored look like a graph (in the mathematical sense). You can refer

at the Neo4J database using the term "whiteboard friendly," which means that it does not matter what you draw, there might be boxes or lines, the drawings can be stored in Neo4J. This type of database concentrates more on the relationships between values rather than on the commonalities among sets of values (such as collections of documents or tables of row). [16] As a matter of fact the data can be stored in a natural and straightforward way. Being small in size, it is possible to embed the Neo4j into approximately any application. Nonetheless it has the capability to store tens of billions of nodes and as many edges. And with its cluster support with master-slave replication across many servers, it can handle most any sized problem you can throw at it.

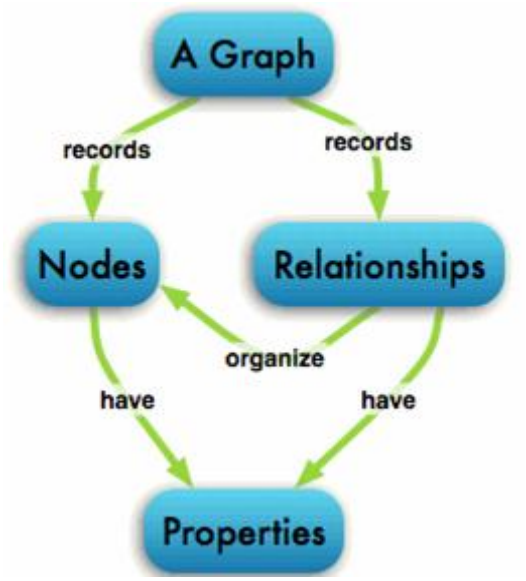


Fig 3. The structure of a graph database [15]

Neo4j's Strengths

Neo4j is one of the finest examples of open source graph databases. Graph databases might be considered the answer for unstructured data, in many ways even more so than document datastores. Even though Neo4j has no type and no schema Neo4j, the constraints it puts on how data is related are essential. At the moment, Neo4j has the ability to support 34.4 billion nodes and the same number as many relationships, which is sufficient for most uses (For example, Neo4j could hold more than 42 nodes for each of Facebook's 800 million users in a single graph [17]).
Beyond ease of use, Neo4j is fast. In spite of join operations in relational databases or map-reduce operations in other databases, the benefit is that graph traversals are constant time. The majority of the databases usually join the values in bulk and filter the desired results. The graph databases, however, act just like the data is only a node step away and it is not compulsory to know how large a graph becomes, moving from node A to node B is always a one-step if they share a relationship.

Neo4j's Weaknesses

There are also a few drawbacks for Neo4j. One of them is the fact that edges in Neo4j cannot direct a vertex back on itself. There is also a problem with nomenclature because it is called node rather than vertex, and relationship rather than edge, thus, adding more complexity when communicating. HA can only replicate a full graph to other servers, even though it is excellent at replication. Finally, if you are in search of a business-friendly open source license (like MIT), Neo4j may not suitable for your company.

Table 1. Comparison between Riak, HBase and Mongo DB

Databases name / Features	Riak	HBase	Mongo DB
Data Model	*Riak stores key/value pairs in a higher level namespace called a bucket.	*HBase stores data in a pre-defined column family format; *Data in HBase is sorted, sparse, and physically grouped by column family	*MongoDB's data format is BSON (binary equivalent to JSON) stored as documents (self-contained records with no intrinsic relationships). Documents in MongoDB may store any of the defined BSON types and are grouped in collections.
Storage Model	*Riak has a modular, extensible local storage system which features pluggable backend stores designed to fit a variety of use cases. The default Riak backend store is Bitcask.	*Hadoop Distributed File System (HDFS) is the storage system used by HBase. Data is stored in MemStores and StoreFiles, where data is streamed to disk (implemented via HFiles, a format based on BigTable's SSTable). Implementations generally use the native JVM-managed I/O file stream.	*MongoDB's default storage system is the Memory-Mapped Storage Engine. It uses memory mapped files for all disk I/O. It is the responsibility of the OS to manage flushing data to disk and paging data in and out.
Query Types and Query-ability	*There are currently four ways to query Riak. ▪ Primary key operations (GET, PUT, DELETE, UPDATE) ▪ MapReduce ▪ Secondary Indexes ▪ Riak Search ▪ Comparing MapReduce, Search, and Secondary Indexes	*HBase has two query options: looking up values by getting/scanning through ordered keys (optionally filtering out values or using a secondary index), or by using Hadoop to perform MapReduce.	*MongoDB has a query interface that has some similarities to relational databases, including secondary indexes that can be derived from the stored documents. MongoDB also has a facilities for performing MapReduce queries and ad-hoc queries on documents. Hadoop support is available, too.

Concurrency	*In Riak, any node in the cluster can coordinate a read/write operation for any other node. Riak stresses availability for writes and reads, and puts the burden of resolution on the client at read time.	*HBase guarantees write atomicity and locks per row. HBase has also recently added multi-action and multi-row local transactions (though you cannot mix read/write actions).	*MongoDB exhibits strong consistency. Eventually consistent reads can be accomplished via secondaries. A MongoDB cluster (with auto-sharding and replication) has a master server at a given point in time for each shard.
Replication	*Riak's replication system is heavily influenced by the Dynamo Paper and Dr. Eric Brewer's CAP Theorem. Riak uses consistent hashing to replicate and distribute N copies of each value around a Riak cluster composed of any number of physical machines. Under the hood, Riak uses virtual nodes to handle the distribution and dynamic rebalancing of data, thus decoupling the data distribution from physical assets. *The Riak APIs expose tunable consistency and availability parameters that let you select which level of configuration is best for your use case. Replication is configurable at the bucket level when first storing data in Riak. Subsequent reads and writes to that data can have request-level parameters.	*HBase supports in-cluster and between-cluster replication. In-cluster replication is handled by HDFS and replicates underlying data files according to Hadoop's settings. Between-cluster replicates by an eventually consistent master/slave push, or more recently added (experimental) master/master and cyclic (where each node plays the role of master and slave) replication.	*MongoDB relies on locks for consistency. As of version 2.2, MongoDB has a DB Level Lock for all operations.

Scaling Out and In	*Riak allows you to elastically grow and shrink your cluster while evenly balancing the load on each machine. No node in Riak is special or has any particular role. In other words, all nodes are masterless. When you add a physical machine to Riak, the cluster is made aware of its membership via gossiping of ring state. Once it's a member of the ring, it's assigned an equal percentage of the partitions and subsequently takes ownership of the data belonging to those partitions. The process for removing a machine is the inverse of this. Riak also ships with a comprehensive suite of command line tools to help make node operations simple and straightforward.	*HBase shards by way or regions that automatically split and redistribute growing data. A crash on a region requires crash recovery. HBase can be made to scale in with some intervention on the part of the developer or DBA.	*Mongo relies on sharding for scaling out. This involves designating a certain server to hold certain chunks of the data as the data set grows. *To scale in, MongoDB has support for removing shards from your database.
Multi-Datacenter Replication and Awareness	*Riak features two distinct types of replication. Users can replicate to any number of nodes in one cluster (which is usually contained within one datacenter over a LAN) using the Apache 2.0 licensed database. Riak Enterprise, Basho's commercial extension to Riak, is required for Multi-Datacenter deployments (meaning the ability to run active Riak clusters in N datacenters).	*HBase shards by way of regions, that themselves may be replicated across multiple datacenters.	*MongoDB can be configured to run in multiple datacenters via various options.

Graphical Monitoring/Admin Console	*Riak ships with Riak Control, an open source graphical console for monitoring and managing Riak clusters	*HBase has a few community supported graphical tools, and a command-line admin console.	*MongoDB does not ship with a graphical monitoring/admin console. However, several community projects have developed graphical monitoring/admin programs. *10Gen offers a hosted monitoring service. [18]

5. Conclusions

Even if not many people know about cloud computing, it became popular in the latest years. Also, famous companies like IBM adopted or created their own cloud. The major advantage is probably the effects on costs. You can save money, time, even work from home. But there are also some disadvantages like security. If your cloud architecture it is weak you can lose valuable data and the company may lose money.

"If you think you've seen this movie before, you are right. Cloud computing is based on the time-sharing model we leveraged years ago before we could afford our own computers. The idea is to share computing power among many companies and people, thereby reducing the cost of that computing power to those who leverage it. The value of time share and the core value of cloud computing are pretty much the same, only the resources these days are much better and more cost effective." [21]

When creating a business strategy, the main reasons why you should choose NoSQL databases is for better performance, scalability, &flexibility. In this modern era, owning a business sometimes could put you in the position to get in touch with clients and provide them with several applications.

These apps developed in-house not only strengthen the relationship with the customer, but also have the role to protect data and keep the company gain access easily and better management the incoming data.(for example, the Adventure Works Shopping application [22])There are some megatrends that had a huge impact on these applications' needs. One of them is the fact that the number of users that applications must support is growing continuously. Moreover, elevated users' expectations for how applications should perform increase proportionally with the number of users. Secondly, the shift in applications should also consider the increase in the volume and the variety of data available. [23]

In conclusion, the main reasons why the usage of NoSQL technology is increasing among companies and enterprises are because it offers data management capabilities that meet the needs of modern applications, including:

- Better application development productivity through a more flexible data model.
- The ability to scale out dynamically and cost effectively to support more users and big data.
- Improved performance that satisfies user expectations for highly responsive applications and allows more complex processing of data.

NoSQL is increasingly seen as a viable alternative to relational databases, and should be considered especially for interactive web and mobile applications.

References

[1] "How the Next Generation of Databases Could Solve Your Problems", htttps://datafloq.com/read/generation-database-solve-problems/139

[2] "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", Eric Redmond, Jim R. Wilsonreleased in 2012 by Pragmatic Programmers, LLC

[3] "Data Access for Highly-Scalable Solutions: Using SQL, NoSQL, and Polyglot Persistence", Douglas McMurtry,Andrew Oakley, John Sharp, Mani Subramanian,, Hanz Zhang, Microsoft 2013

[4] Bernard Golden, ''McKinsey Cloud Computing Report Conclusions Don't Add Up,'' CIO.com (April 27, 2009), www.cio.com/article/490770/ McKinsey Cloud Computing Report Conclusions Don t Add Up.

[5] Irving Wladawsky Berger, ''The 'Outside In' Enterprise,'' October 10, 2005, blog post http://irvingwb.typepad.com/blog/2005/10/the outsidein e.html.

[6] 3 Pillar Global, Exploring the Different Types of NoSQL Databases Part II http://www.3pillarglobal.com/insights/exploring-the-different-types-of-nosql-databases

[7] "Dynamo: Amazon's Highly Available Key-value Store",http://allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf

[8] "Dynamo: Amazon's Highly Available Key-value Store",http://allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf

[9] Basho, Riak Distributed Database http://basho.com/riak/

[10] Protocol Buffer http://code.google.com/p/protobuf/

[11] "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", Eric Redmond, Jim R. Wilson

[12] IBM, Hadoop http://www-01.ibm.com/software/data/infosphere/hadoop/hbase/

[13] Cloudera, CDH http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/hbase.html

[14] "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", Eric Redmond, Jim R. Wilson

[15] "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", Eric Redmond, Jim R. Wilson

[16] "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", Eric Redmond, Jim R. Wilson

[17] "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", Eric Redmond, Jim R. Wilson

[18] Basho, Riak Comparisons http://docs.basho.com/riak/1.3.1/references/appendices/comparisons/

[19] Joe McKendrick, 10 Quotes on Cloud Computing That Really Say it All, Forbes Contribuitors, March 2013 http://www.forbes.com/sites/joemckendrick/2013/03/24/10-quotes-on-cloud-computing-that-really-say-it-all/

[20] Douglas McMurtry, Andrew Oakley, John Sharp, Mani Subramanian, Hanz Zhang, "Data Access for Highly-Scalable Solutions: Using SQL, NoSQL, and Polyglot Persistence" released in 2013 by Microsoft

[21] David Linthicum**,** *Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide*

[22] Douglas McMurtry, Andrew Oakley, John Sharp, Mani Subramanian, Hanz Zhang, "Data Access for Highly-Scalable Solutions: Using SQL, NoSQL, and Polyglot Persistence" released in 2013 by Microsoft

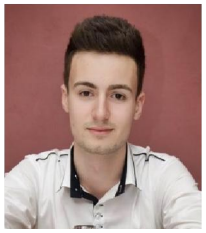
[23] CouchBase, Why NoSQL? http://www.couchbase.com/nosql-resources/what-is-no-sql

[24] MongoDB, Introduction to MongoDB http://www.mongodb.org/about/introduction

Bogdan NEDELCU graduated Computer Science at Politehnica University of Bucharest in 2011. In 2013, he graduated the master program "Engineering and Business Management Systems" at Politehnica University of Bucharest. At present he is studying for the doctor's degree at the Academy of Economic Studies from Bucharest.

Andreea Maria IONESCU is currently pursuing an undergraduate degree at the Bucharest University of Economic Studies. The field of study is Cybernetics, Statistics and Economic Informatics.

Ana-Maria IONESCU is studying Informatics and Economics at Academy of Economic Studies from Bucharest since 2013. She is volunteer at SiSC, a NGO in faculty.



Alexandru George VASILE studies at Academy of Economic Studies from Bucharest, Faculty of Cybernetics, Statistics and Economic Informatics since 2013.