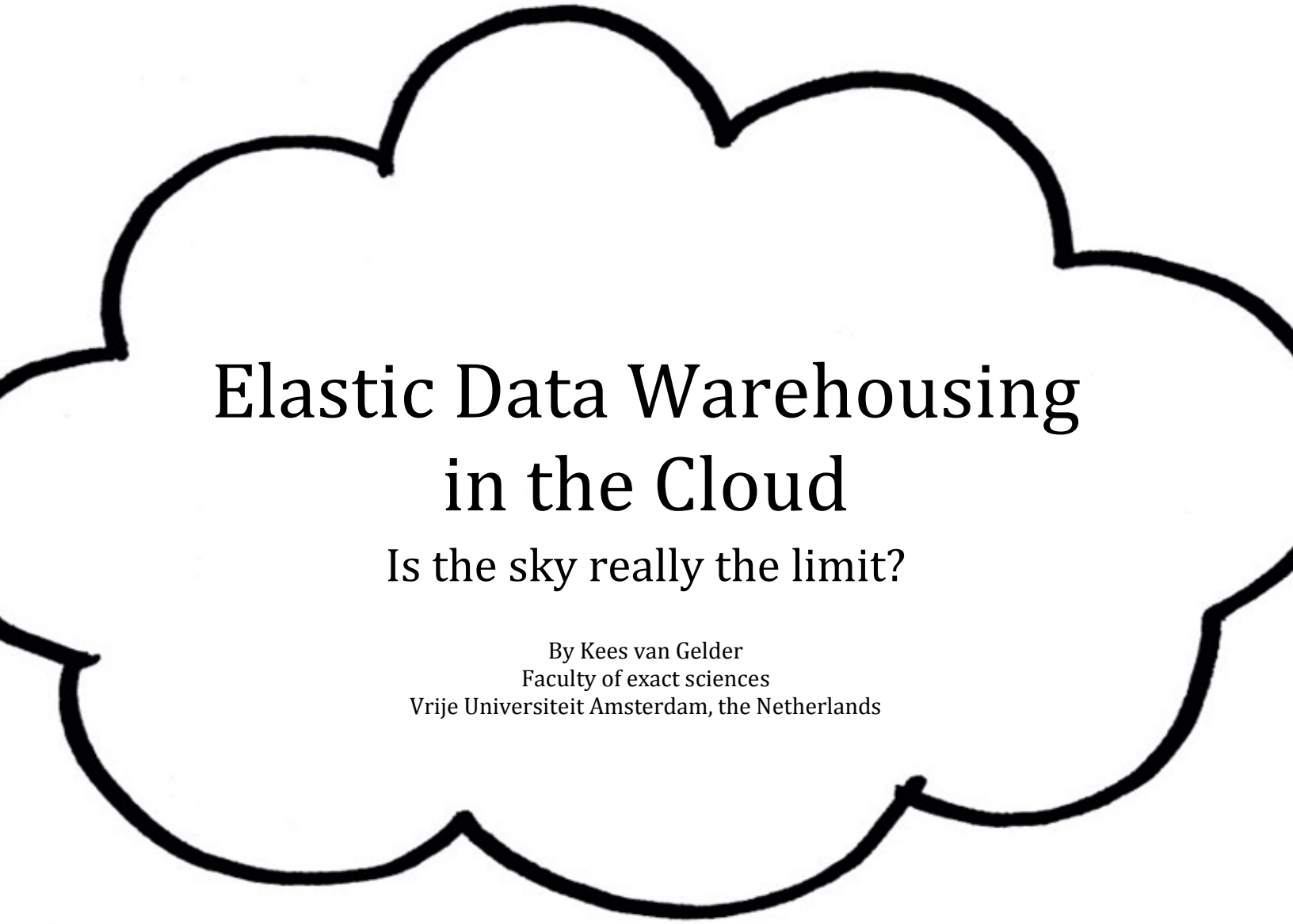

Elastic Data Warehousing in the Cloud

Is the sky really the limit?

By Kees van Gelder
Faculty of exact sciences
Vrije Universiteit Amsterdam, the Netherlands

Index

Abstract

Software-as-a-service, typically on cloud infrastructures, offers significant benefits such as shorter development time and risk, reduced operational cost and a dynamic pay-as-you-go deployment model that provides elasticity in the face of hard-to-predict workload patterns. In this project, we try to answer the question if business intelligence applications, and in particular data warehousing environments, can benefit from this trend. For this purpose, we first provide a functional overview of state-of-the-art data warehousing techniques as well as cloud infrastructures, and a review of existing efforts to combine the two. Throughout, we focus on a major technical challenge for the underlying cloud database architectures: how do current cloud infrastructures, which tend to be restricted in terms of inter-node storage and network bandwidth, support data warehouse *elasticity*.

1. Introduction

Organizations rely more and more on business intelligence (BI); the gathering of valuable information within organizations. Technologies able to support business intelligence include data warehousing systems. In a data warehouse, data from the entire organization can be brought together in one place. The data in a data warehouse can be used directly by means of a query language, or indirectly by means of data intelligence tools (e.g. data mining) to inform decision makers within an organization, or to operate without any human intervention whatsoever as part of a automated decision making instance (for example to steer customer interactions on a website). Data warehouses may support OLAP (on-line analytical processing) tools, allowing the decision maker to navigate the data in the data warehouse. Data warehousing systems can be used for example to compute detailed statistics or detect underlying trends within the data.

Because of the BI advantages that a data warehousing system can bring to organizations there is a lot of demand for these kinds of systems. However, there are a number of factors making developing and maintaining a data warehouse system a painful process:

- Setting up a data warehouse can take a long time.
- Over-provisioning (over-estimating the needs of the system to meet a certain service level at peak workloads) can lead to high costs.
- Organizations may lack the expertise needed to set up and maintain a data warehouse.
- System crashes, downtime or system overload can have numerous consequences for an organization.

There is a potential solution for these issues: cloud computing.

Cloud computing refers to computational resources in terms of software or computational power made available through a computer network (e.g. the internet). There are companies trying to utilize cloud computing by offering software as service (SaaS) to customers. This allows customers to access the application online via the internet. An example of SaaS is Google Apps from Google. Other examples include SkyInsight [41] and GoodData [42], or the many providers of hosted email applications. Instead of software, computational power can also be provided as a service on the cloud, which can then be used as a platform for customers to run their own applications (PaaS, or Platform-as-a-Service). Examples of these services include the Azure services from Microsoft [37], services from Rackspace [36], and Amazon with their S3 and EC2 services [1]. Organizations that use these services to host their application experience several benefits, including:

- The expertise of building and maintaining the systems hosting these applications is no longer needed within the organization itself.

- Over-provisioning can be avoided because the organization itself does no longer have to be responsible for estimating the needed computational power for an application; in the cloud the right amount of resources can be provided to an application without interference of the customer. PaaS services often allow the customer to pay only for the computational power that is actually used (pay-per-use), allowing the customer to save money.

There are several challenges when deploying data warehouses into the cloud:

- Importing the data needed for the data warehouse into the cloud for storage can be a challenge, because when using the cloud, a customer is dependent on the internet and the infrastructure of the cloud provider. This may be both a performance and cost issue.
- Getting large amounts of data from cloud storage to virtual nodes provided by the cloud provider for computing can be a performance issue.
- Getting the data warehouse to perform as desired can be challenging because cloud providers tend to offer low-end nodes (e.g. virtual machines) for computations, whereas local data warehousing systems tend to be well-provisioned in terms of CPU, memory and disk bandwidth.
- Applications running in the cloud experience WAN latency.
- Loss of control can lead to issues involving security and trust.
- There is a need for comprehensive cloud-based system administration and data lifecycle management.

The aforementioned pay-per-use model has the potential to handle peaks and valleys in the use of a data warehouse in a financially efficient way. A data warehouse might for example not be used much at night, or the different parts of a data warehouse of an international organization may be in heavy or low use during certain times of the day. An elastic data warehousing system in the cloud would automatically increase or decrease the number of nodes used, allowing one to save money. The potential for elasticity is important to us, because it provides a good reason (financial advantages) for organizations to consider data warehousing in the cloud.

In the next section we will take a look at how conventional data warehousing systems work. We will also provide an overview of what the large providers of data warehousing products currently have to offer. After this, we will shift our focus towards the cloud. We will look at what the cloud environment looks like, and we will analyze several products currently offered by large cloud providers. Thereafter we will address important issues in deploying data warehousing in order to analyze the potential for data warehousing in the cloud. We also address functional requirements that data warehousing systems will largely have to comply with in order to make data warehousing systems more cloud-friendly. This paper is meant to answer the following fundamental questions:

- Is it feasible to deploy elastic data warehouses on to the cloud?
- What would an elastic data warehouse on the cloud look like?

Contributions of this paper:

- Identifying the important properties of the cloud.
- Identifying the important properties of data warehousing.
- Providing an overview of current-date state of the art data warehouse soft- and hardware.
- Providing an overview of large cloud providers and their characteristics.
- Reviewing the possibilities and impossibilities of data warehousing in the current-date cloud.

2. The basics of data warehousing

Before we can look at the possibilities for data warehousing in the cloud, we first survey the field of data warehousing in general. Understanding the general principles in data warehouse system design will help us understand what is needed for a data warehouse system to work effectively, and this will help us understand whether or not data warehouse systems are deployable in the cloud as well.

2.1 What a data warehouse is and means

In the past few decades, virtually all organizations of any size have started to make use of any kind of database system. Organizations have also started to deploy data warehousing systems. In a data warehouse all data from an organization can be brought together in one place. Information that can be derived from this data includes information about for example abnormalities in credit card behavior, customer behavior, healthcare analytics and so on [2]. In general data warehouse systems are used to compute detailed statistics or detect underlying trends within data.

Data warehouse systems require a different kind of database than conventional database systems. While originally database systems were used to perform mostly small transactions (OLTP or on-line transaction processing), data warehousing systems are also used for complex analytics involving huge amounts of data (OLAP or on-line analytical processing) [2]. Most of the time, systems for OLTP and OLAP will have to co-exist in order to comply with all the needs from a business perspective [2].

2.2 The general architecture of a data warehouse

There are many commercial and non-commercial solutions to data warehousing. There are a number of components that are always needed in order for a data warehouse to work as desired. These components can be seen in the following figure and are also described in [2,20]:

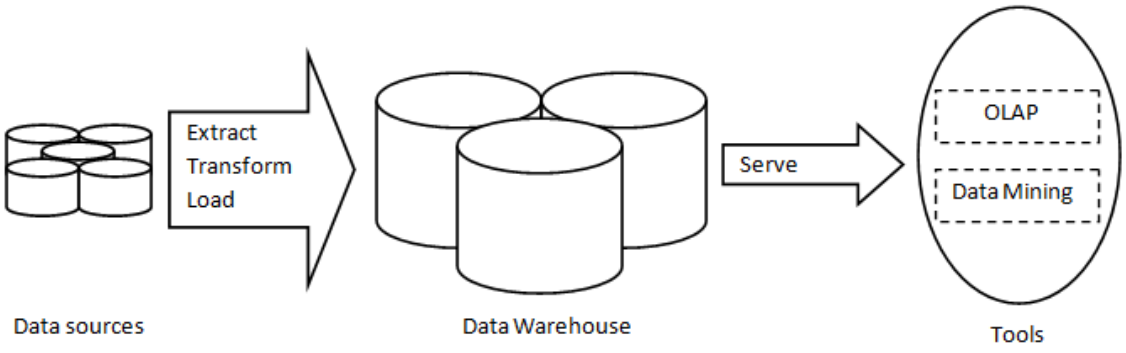

Figure 1: the general structure of a data warehouse.

The information in a data warehouse comes from the data sources, which are typically OLTP systems. The data warehouse component is the place where all data from an organization or business can come together. It can be useful to clean data before it enters the data warehouse component. This means that anomalies like inconsistencies and duplicates will have to be removed from the data in order for the data warehouse system to operate as desired. In order for the data warehouse to be able to reason with data coming from potentially many different sources it can be useful if data is presented in a specific uniform format, meaning that data will usually have to be transformed before it enters the data warehouse. It is also possible to perform

less or no cleaning or transformation whatsoever, but this can lead to difficulties in querying the data [21].

Once the process of pre-processing the data is completed the data can be loaded into the data warehouse. In the past it was still ok for a data warehouse to be offline for some time (at night or during the weekends for example) in order to able to load the data into the data warehouse, but this is not a viable option anymore. More and more data has to be available 24/7. Ensuring performance of queries towards the data warehouse while data is being loaded at the same time is a challenge in current data warehouse environments.

In addition to one data warehouse where all data comes together, an organization may also choose to use data marts which carry only part of the data warehouse. Data marts are more specialized and therefore easier to deploy, but on the long run they can lead to integration problems [2,20] as data in the organization gets replicated and data consistency might be affected. A data mart is typically set up by a single department or division within an organization for a single purpose. It is often a solution chosen to quickly implement a needed system, without affecting or having to change the existing data warehouse. We believe the question of moving towards the cloud is just as relevant for data marts as for data warehouses. It may even be more relevant because the fundamental requirements of a data mart are simpler and human resources and expertise are likely to be managed on the department level instead of for the central organization.

Once the data is loaded into the data warehouse, it is ready for usage by OLAP and data mining tools. The OLAP and data mining tools themselves might not be our key interest throughout this paper (we are mostly interested in the data warehouse component) but it is still useful to get a general idea of these tools because the inner workings of the data warehouse depend also on these tools.

In OLAP tools, data is usually presented in the form of a data hypercube. An example of a three dimensional data cube is presented in figure 2.

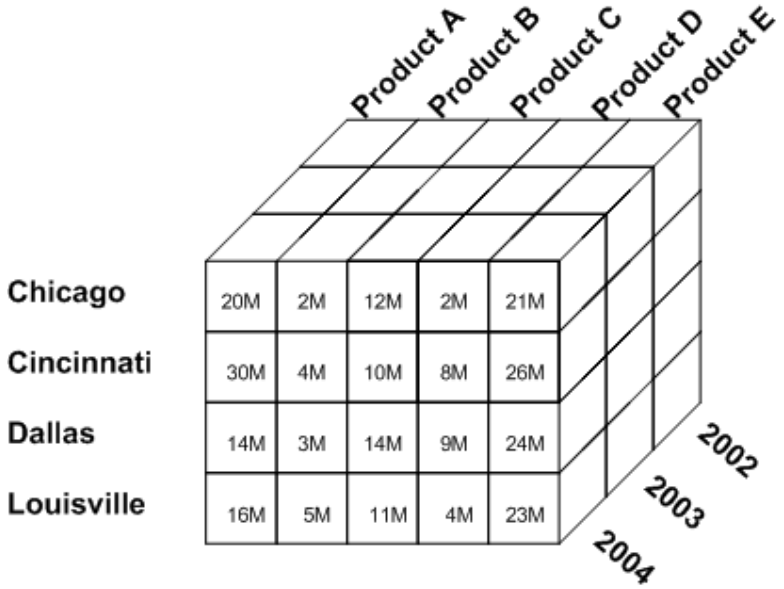

Figure 2: a data cube representing the dimensions city, product and year [22].

Every one of the three dimensions in figure 2 represents a data dimension; one represents region (cities), one represents products and one represents the time (years). This data cube can be used to detect for example how much revenue product A had in Chicago in 2002. Dimensions often have multiple granularities; time could for example be represented in years, quarters,

months, weeks or days. Typical operations of the OLAP tools are drill down and rollup. In drill down we take one or more dimensions and make it more detailed. We could for example split the years up into months. We could also do the opposite, namely generalize, which is called rollup. We could for example look at an entire country or an entire product range instead of specific cities and products.

2.3 How a data warehouse works

We will now look in some more depth to how the data warehouse components itself works. Please note that the reason we were most interested in this component is that deploying just the data warehouse component into the cloud might be especially interesting for organizations that already maintain a data warehouse. This would allow these companies to largely maintain the way they operate without having to introduce changes to their existing applications.

Two main approaches can be distinguished in the storing of data within a data warehouse [2,20]. One of these approaches is the MOLAP approach (Multidimensional OLAP). In the MOLAP approach data is stored directly into multidimensional data cubes. A multidimensional storage engine is used in order to store these data cubes. The advantage of this approach is that the data can be used directly by OLAP tools without much need for transformation (because data in OLAP tools is usually represented in data cubes). The disadvantage is that it is harder to integrate; the data from the original data sources needs to transformed significantly in order to fit into the data warehouse. The MOLAP approach also has the disadvantage of not scaling well [23] and being harder to integrate with SQL [20].

The other approach in storing data is the ROLAP approach (Relational OLAP). The idea is that the data warehouse can be implemented using conventional SQL techniques. In order to achieve this 'star schemas' or 'snowflake schemas' are used. ROLAP has the advantage of being much easier to integrate with conventional SQL database sources [20] and it is a scalable approach [2]. In a star schema there is a single fact table. In this table the base data cube is stored. It contains pointers to all available dimensions. In the snowflake schema the dimension hierarchy is explicitly represented by making dimensions 'higher' in the hierarchy point to dimensions that are 'lower' in the hierarchy. Examples of a star- and snowflake schema's are shown in figure 3 and 4.

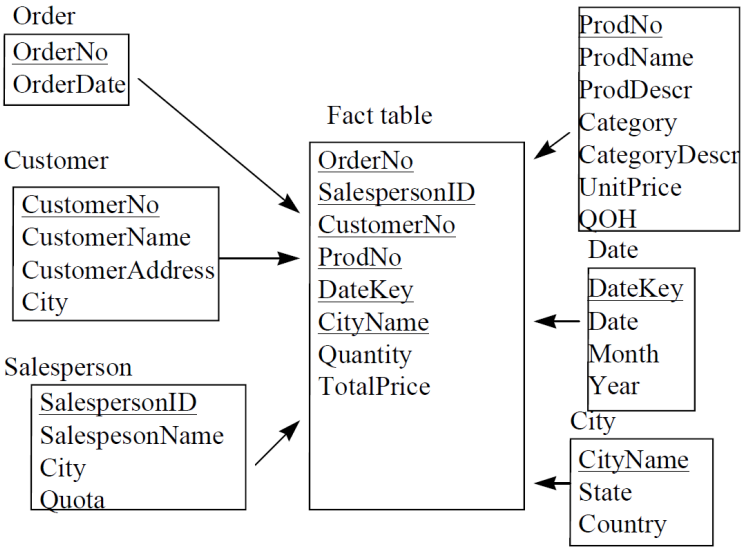

Figure 3: a star schema [2].

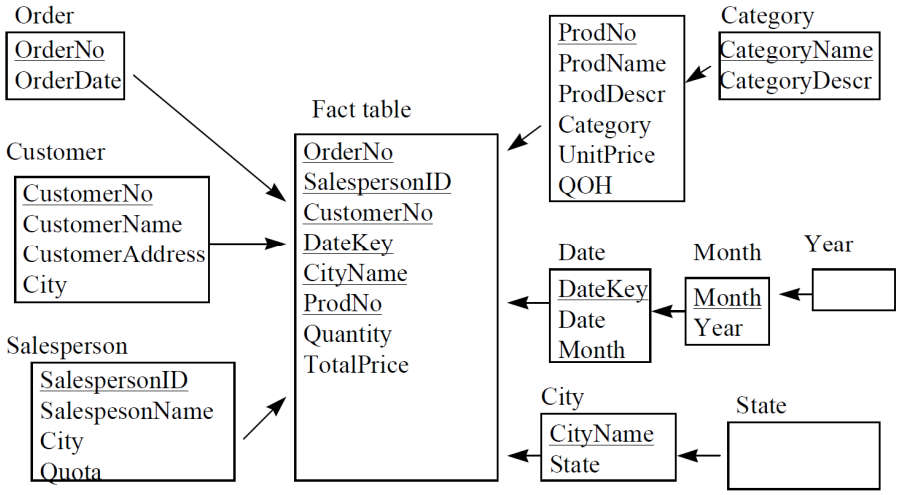

Figure 4: a snowflake schema [2].

In a data warehouse system, where many terabytes of data might have to be processed, query performance can be an important issue. In order to speed up queries, data warehouse systems often use materialized views [2,20]. A materialized view is basically the result of a completed query on the data stored explicitly. If a new query comes, requiring the same operation that was used to create the materialized view, the database system can refer to the materialized view in order to get the right information quickly. The user might or might not have to be aware of materialized views. Determining what views to materialize and how to maintain them is an important field of research and many solutions have been proposed [24,25,26].

In order to achieve good performance out of a data warehouse system, parallelism can be applied. Two approaches can be distinguished. One is to use multiple cores for executing a query. The other approach is using partitioning [27,28,29] to achieve parallelism by using multiple machines. A shared-nothing architecture can be especially interesting for this approach [27]. In a shared-nothing architecture no data is shared across partitions. The latter approach is considered more scalable since many machines can be used to achieve parallelism.

Data warehouse systems generally have capabilities to monitor and administer the data warehouse [2]. Workload management can be part of these capabilities. Workload management is important for keeping track of and analyzing queries being executed. Workload management can ensure that work is fairly distributed across for example different database partitions. It can also help by for example determining what views need to be materialized. Workload management is an important factor for *elastic* data warehousing in the cloud, because there is a need for detecting changes in the workload in real-time in order to be able to act accordingly.

2.3 The shape of the current data warehouse market

In this subsection we will take a look at a number of the biggest commercial providers of data warehousing soft- and hardware. We will focus at the following big suppliers of data warehousing products: IBM [30], Oracle [31] and Teradata [32]. Understanding what the market is offering helps us in understanding what data warehouse systems currently look like, and this will help us figure out if the cloud can be used as a suitable environment for data warehousing.

Comparing the different suppliers of data warehousing products quality-wise is hard; they all claim to be the fastest and cheapest one. Instead, we look at what kinds of functionality are offered by the different companies. Even comparing functionality can be difficult because the companies we focus at do not share implementation details very often.

IBM owns many data warehousing technologies. These include DB2, Informix, solidDB and pureScale. We focus on their flagship product: DB2 [33]. DB2 uses a shared nothing architecture. This means that the different database partitions do not share any information, making the data warehouse modular and scalable. DB2 uses a cost-based optimizer for queries, meaning that the system will try to find out what possible execution of a query will give the best performance from a financial perspective. DB2 uses partitioning techniques based on hashes or on specific natural categories (time, location, etc). It also provides Multi Dimensional Clustering (MDC) [40] as a table storage method. Materialized views [38] are another important feature offered to speed up data warehouse workloads. IBM uses a b-tree for indexing . IBM uses data compression techniques to reduce the I/O volume needed for data warehousing, claiming the reduced I/O costs outweighs the extra decompression costs. DB2 also uses 'shared scan' techniques [39] to be able to share certain results between data scans if they are looking for the same data. This does not only reduce work, but also prevents thrashing (large amounts of computer resources are used to do a minimal amount of work). DB2 supports changing the physical design of the data warehouse without having to take the system offline. IBM relaxes the dependency model in DB2, making sure that views do not have to be redefined after the definitions of underlying tables change. DB2 uses a workload management system that they compare to the autopilot of an airplane. The rules in the workload manager would be fully customizable. Monitoring capabilities are also provided.

IBM also offers a range of fully integrated data warehousing systems, ranging anywhere from systems with a total of 8 cores, 64GB memory and 14TB of storage to systems with a total of 320 cores, 2560GB memory and 560TB of storage spread across multiple modules.

Oracle, like IBM, offers a variety of different database products. Their most prominent product regarding data warehousing right now is their 'Database 11g' product range [34]. It comes with several options, including OLAP capabilities. These OLAP tools use data cubes. Ways to materialize views are also provided. SQL can be used to query the data cubes because queries are automatically rewritten in order to access the data cubes. Just like IBM, the 11g database has capabilities for compressing almost everything in the data warehouse. We notice that IBM offers more all-inclusive solutions while Oracle offers a lot of capabilities optionally. Oracle also offers capabilities for partitioning data. Data can be partitioned based on hashing, range and more. Also, a combination of partitioning categories can be used to partition more specifically. The partitioning is meant to improve performance, manageability and availability. Oracle also offers clustering capabilities, meaning that one database can run across multiple servers; a private cloud can be created. This appears to be very useful for elasticity, because nodes can be added and removed to or from the private cloud.

Oracle also offers hardware appliances, including their top of the range 'Exadata Database Machine X2-8', that is supposedly suited for data warehousing as well as OLTP. It comes with 14 'Exadata Storage Servers'. These have capacities up to 24TB. The appliance comes with a total of 128 cores and 2TB of memory for database processing, as well as 168 cores for storage processing. The system contains 2 'industry standard' database servers as well.

Teradata's 'Database 13.10' [35] has properties comparable to the ones from the aforementioned data warehousing products from IBM and Oracle. It works with data compression, workload management tools, queries are carefully planned in order to improve performance and Teradata tries to parallelize as much as possible. Teradata uses special networking hardware (BYNET). Their architecture is a shared nothing architecture. OLAP tools are provided to the users.

Teradata offers a range of hardware appliances for data warehousing. Each node in their data warehousing products has two six core processors and there can be up to 9 nodes in a cabinet (up to 6 cabinets). Storage starts at 5.8TB, up to 343TB in 6 fully populated cabinets.

There is a clear trend within the data warehousing systems analyzed above; data warehouses are increasingly complex systems that require state of the art computing technologies in order to provide data compression, partitioning, OLAP tools and parallelizability while being capable of sustaining data loads and updates at even shorter intervals.

3. Cloud computing: principles and practice

In this section we survey the field of cloud computing in general, which will help us determine whether or not data warehousing systems can be made suitable for the cloud environment. In this section we will also look at the different providers of computational resources in the cloud in terms of both the performance and cost-based characteristics of their services.

3.1 What is cloud computing?

Cloud computing refers to computational resources in terms of software or computational power made available through a computer network (e.g. the internet). In the cloud either software can be provided as a service to customers (SaaS or Software-as-a-Service) or computational resources can be provides for a customer to use the cloud as a platform (PaaS or Platform-as-a-Service). Cloud computing is sometimes seen as the potential fifth utility besides electricity, water, gas and telephony [18]. The cloud can be seen as a virtual environment because consumers do not have to be aware of the internals of the cloud environment. Because we are interested in running a data warehousing system by using the cloud as a platform we are mostly interested in PaaS services throughout this paper.

There are also SaaS approaches to data warehousing in the cloud, like GoodData [42] and SkyInsight [41]. A consequence of this approach is that companies need to switch their applications (OLAP tools) as well as the data warehouse to a new purely web-based environment. Our discussion of data warehousing in the cloud from a PaaS perspective is also relevant for SaaS data warehousing approaches because SaaS providers must typically deal with PaaS data warehousing issues themselves as well.

A provider offering PaaS services can offer computing, networking or storage capabilities to the customer via a computer network (e.g. the internet). This is generally applied by using a pay-per-use model; a customer only pays for what he/she actually uses. Nodes in the form of VM's (virtual machines) can be offered by providers to customers, allowing a provider to share and distribute physical computing power across multiple customers. VM's can be completely separated and multiple VM's can run on a single physical computer [18]. Computing, networking or storage capabilities are provided, without the consumer having to be aware of the underlying architecture.

3.2 Why use cloud computing?

There are several reasons that make the cloud computing a contender for traditional computing techniques. Some these reasons are listed below:

- Infinite scalability. With cloud computing there is an illusion of infinite computing resources [17]. If a customer desires more resources, he/she can rent these resources and more capabilities will become available to the customer almost instantly.
- Speed of deployment. Offering of full-fledged services by cloud providers can reduce deployment time compared to in-house deployment.
- Elasticity. In cloud computing a pay-per-use payment model is generally applied [17], meaning that you only pay for the resources you actually use. This model ensures that startup costs and costs due to over-provisioning are avoided, without the risk of missing service levels due to under provisioning.
- Reliability. Theoretically, a cloud provider can achieve high reliability. This can be achieved not only by making backups, but also by having for example multiple data centers (allowing for handling for example power outages). However, there have already been significant service outages in cloud computing [17], making this debatable.

- Reduced costs. Thanks to economies of scale (things tend to get cheaper when scale increases) at the cloud provider, costs can be reduced, potentially allowing customers to reduce costs as well. Elasticity allows for reduced costs when usage is low, as the customer uses less resources and therefore pays less. Because certain expertise can be centralized at the cloud provider the customer does no longer have to have this expertise in-house, allowing the customer to potentially save money.

3.3 Why not use cloud computing?

Besides advantages, there are also some potential disadvantage and bottlenecks to keep in mind when discussing cloud computing. The potential disadvantages listed below underline why it is challenging to deploy data warehousing systems in the cloud.

- Communication with the cloud happens by means of a WAN link. When dealing with huge amounts of data (i.e. terabytes) this might become a bottleneck [17]. Data transfer over a WAN link is generally not very fast compared to data transfer in local systems. A WAN link can also lead to latency issues.
- Performance in the cloud can be unpredictable [17]. When a large number of customers are active in the cloud at the same time, this may decrease performance. Especially when sharing I/O to write to traditional disk [17]. This potential disadvantage does not have to be present; it depends on the architecture of the cloud by the cloud provider and careful design of the cloud might go a long way in eliminating these issues.
- Loss of control [17]. When an organization starts to use the cloud as a platform, it loses some of the control it previously had. One can for example no longer increase performance by buying better hardware, nor can one fix downtime themselves anymore. Loss of control also leads to significant security and thrust issues.
- High costs. Costs can both be an argument for and against using the cloud. When many terabytes of data are involved, data transfer to the cloud can become expensive. For example: Amazon charges roughly $100 to transfer 1TB of data towards their cloud [17]. It is even claimed that physically shipping data on disk to a different location is the cheapest way to send large quantities of data [19] (some cloud providers allow customers to do this). Exploiting economies of scale can be difficult for large organizations once dependent on cloud provider prices.

3.4 Analyzing the market of cloud computing

In this section we will focus at the following big cloud providers that provide a platform as a service (PaaS): Amazon [1], Rackspace [36] and Microsoft [37]. Understanding what the cloud environment looks like in the market helps us understanding whether or not this is a viable platform for data warehousing. We compare mostly based on the following ingredients: computing power, storage, VM's, elastic capability and costs.

Amazon's flagship product for computing in the cloud is the Amazon Elastic Compute Cloud (EC2). Amazon offer a range of different instances for computing in the cloud. Each instance uses 'EC2 compute units'. Amazon claims that these units have the computing power equal to a 1.0 to 1.2 GHz 2007 Xeon processor. Their most extreme instances (the 'cluster compute' and 'cluster GPU' instances) have up to 23 GB of memory, 33.5 EC2 compute units and 1690 GB of storage (a 10 Gigabit Ethernet connection is offered). Instances start at 613 MB of memory with up to 2 EC2 compute units (with no storage at the instance). For separate storage Amazon currently offers two products: Amazon Elastic Block Store (EBS) and Amazon Simple Storage Service (S3). The amount of available storage is virtually unlimited. As the name suggests S3 is supposed to be simple and the feature set is 'minimal'. It can be seen as a standard data storage server. EBS allows for storing data blocks from 1 GB up to 1 TB and the blocks can be mounted as devices by EC2 instances. Amazon uses 'Amazon Machine Images' (AMI's) for virtualization. A range of

different Linux and Windows Server operating systems can be used for these AMI's. Amazon has support for following database systems: IBM's DB2, IBM Informix Dynamic Server, Microsoft SQL Server Standard, MySQL Enterprise, Oracle Database 11g and more. Amazon offers 'Auto Scaling' to achieve elasticity. Instances can be added or removed based on rules the customer defines (startup times of nodes are not clearly specified). Amazon also offers 'Elastic Load Balancing', used to automatically distribute workloads over multiple instances.

Windows Azure is Microsoft's platform for cloud computing. There are a number of different computing instances available with varying computing capabilities from small instances with a 1 GHz processor, 768 MB of memory and 20 GB of storage up to instances with 8 1.6 GHz processors, 14 GB of memory and 2TB of storage. Microsoft also offers a range of separate storage products. These products include a product to store binary objects, a product to store tables and a product to store virtual hard drives (VHD's). There is no specific limitation to the amount of storage used. In terms of virtualization Microsoft only supports the deployment of a custom Windows Server 2008 R2 image. Microsoft offers little functionality regarding elasticity. Microsoft also offers SQL Azure for databases in the cloud. SQL Azure databases are limited to a maximum of 50 GB. In terms of elasticity SQL Azure databases scale automatically but only up to 50 GB per database.

For cloud computing Rackspace also offers a variety of different computing instances. Instances range from 256MB of memory and 10 GB of storage up to almost 16 GB of memory and 620 GB of storage. Rackspace does not share exact details about CPU power but they do offer up to 4 virtual CPU's per instance with a service called 'CPU Bursting' that should supposedly make the virtual CPU's run faster when more computing power is available. For storage Rackspace offers the 'Cloud Files' service. It is a basic storage system that can be seen as a standard storage server. In terms of VM's Rackspace offers support for a range of different Linux distributions as well as Windows images to use as operating systems. A SQL Server 2008 database can be added to a Windows image. Elasticity is provided in the form of 'In-Place Resizing'. By using the API the customer can manage the number of instances used. The system is taken offline to reallocate resources. According to Rackspace this process takes 'just a few minutes'. Rackspace also offers technology to enable load balancing within the cloud.

A summary of the characteristics and properties offered by the aforementioned cloud providers can be found in table 1, with the addition of pricing information.

	Amazon	*Rackspace*	*Microsoft Windows Azure*	*Microsoft SQL Azure*
Computing capabilities for instances	Up to roughly 33.5 1GHz processors, 23 GB of memory and 1690 GB of storage	Up to 4 virtual CPU's with variable speeds, 16 GB of memory and 620 GB of storage	Up to 8 1.5 GHz processors, 14 GB of memory and 2 TB of storage	-
Storage	S3 for basic storage and EBS for mountable block storage. Unlimited capability	Basic storage system. Unlimited capability	Storage capabilities for simple binary objects, table storage and VHD storage. Unlimited capability	Database storage. Limited to max. 50 GB
Payment model	Pay-per-use. Subscriptions and bidding for resources available	Pay-per-use	Pay-per-use. Subscriptions available	Partial pay-per-use. Subscriptions available.
VM's	Yes, both Linux and Windows operating systems	Yes, both Linux and Windows operating systems with support for SQL Server 2008	Yes, only Windows Server 2008 R2 operating system	No
Elasticity	Automated managing of number of instances, automated elastic load balancing	Managing the number of instances via API. Load balancing technology	No functionality.	Limited (up to 50GB) automatic scalability
Computing prices	From $0.020 ('micro' instance with Linux) up to $2.100 per hour ('Cluster GPU' instance).	From $0.015 (most basic instance with Linux) up to $ 1.080 per hour (most high-end instance with Windows)	From $0.050 (smallest instance) up to $0.960 per hour (largest instance).	-
Data storage prices	S3: from $0.037 up to $0.140 per GB per month depending on amount stored & amount of redundancy. EBS: $0.100 per GB	$0.150 per GB per month	$0.150 per GB per month	From $9.990 (1 GB 'Web Edition') up to $499.950 (50 GB 'Business Edition')
Data transfer prices	In: $0.100 per GB. Out: from $0.000 up to $0.080 per GB depending on amount stored	In: $0.080 per GB. Out: $0.180 per GB	In & out: $0.150 per GB	In: $0.100 per GB. Out: $0.150 per GB
Other costs	S3: $0.010 per 1.000 PUT, COPY, POST, or LIST transactions. $0.01 per 10.000 GET transactions. EBS: $0.100 per 1 million I/O transactions. Cloudwatch (includes 'Auto Scaling'): from $0.000 up to $3.50 per instance per month for monitoring, $0.50 per custom metric per month, $0.10 per alarm per month and $0.01 per 1.000 transactions	Load balancing: $0.015 per hour and $0.015 per 100 concurrent connections	Storage or data transfer: $0.010 per 10.000 transaction	Bus: $3.990 per connection or from $9.950 (5 connections) up to $995.000 (500 connections). Cache: from $45.000 (128 MB) up to $325.000 (4 GB). Access control: $1.990 per 100.000 transactions

Table 1: The properties of cloud services provided by Amazon, Microsoft and Rackspace. Prices are in USD and may vary per region

4. Putting it together: Elastic Data Warehousing in the Cloud

Now that the fields of both data warehousing and cloud computing have been surveyed, we will speculate about the combination to see whether or not there are promising possibilities for data warehousing in the cloud. We will do so by analyzing the following aforementioned arguments against moving towards the cloud: the slow speed of moving data towards the cloud, poor performance in the cloud, loss of control and costs issues. We will also specifically analyze the current possibilities for enabling elasticity. After this, we will present a list of functional requirements for data warehousing systems in the cloud, and we will discuss the issues involved in complying with these requirements.

4.1 Discussion & Vision

Moving data towards the cloud can be both expensive [17] and slow (because a WAN link is usually involved). Solving the speed issue can be difficult because one depends on the internet. However, WAN speeds might improve over time, for example because of the deployment of optic fibers. Amazon also offers a service allowing the customer to physically ship their data on disks towards Amazon. This can be both cheaper and faster, depending on provider charges and the speed of the WAN link available to the customer [19]. Because WAN connectivity, cloud application may also suffer from latency issues. However, given the nature of data warehousing applications, we believe that these issues do not tend to be significant. We do not see a query result becoming visible after for example 2.2 seconds instead of 2.0 seconds as a potential bottleneck.

For a data warehousing system complex analytics involving large amounts of data can be involved. If data analytics do not perform well in the cloud this might be a reason to argue against moving towards the cloud. We have analyzed the data warehousing appliances offered by some big suppliers of data warehousing software and hardware, and we have looked at the capacity of nodes in the cloud. From this we can derive that nodes tend in the cloud tend to be significantly lower-end than data warehousing appliances offered by big companies. We believe that the solution to this potential bottleneck lies in distributing both workload and data across multiple nodes in the cloud. By using multiple nodes and applying both parallelization and partitioning techniques as seen in current-date state of the art data warehousing soft- and hardware one could potentially build data warehousing systems that are highly parallel and distributed on the cloud. In fact, attempts to introduce general architectures for OLTP databases utilizing these possibilities in the cloud have already been presented [3,10]. A challenge that requires attention is querying data spread across multiple nodes, for nodes might be too small to contain a large partition. Mechanisms to provide load balancing in order to achieve better performance are already provided by some big cloud providers, although the cloud providers do not have specific insights in how specific applications operate in the cloud. We have not seen any load balancing being offered specifically relevant for data warehousing. With cloud nodes providing up to 23GB of memory, 2TB of in-node storage and speeds equal to roughly 33.5 1GHz processors, we believe that these nodes themselves might provide enough capabilities for small data warehouses or, more importantly, for data marts. If a data mart can be deployed in just one node it might become more interesting for organizations to move their data marts towards the cloud, because for example performance issues regarding the usage of multiple nodes can be avoided.

When an organization decides to move its data warehousing system (or data marts) towards the cloud, this organization will lose some of the control it previously had. This can lead to security and trust issues. While being difficult to solve, it is our believe that security and trust might not always be a bottleneck. With data warehousing, sensitive data can for example often be left out of the analysis [4], reducing security risk. Another trend to deal with security is to store

encrypted data, though this leads to technical issues [16] because encrypted data tends to require different forms of analysis. Political issues might also be involved regarding security. In some countries (e.g. China) there are limitations in terms of what you may store in a foreign country. This can be solved by building data centers in these countries. Instead of feeling less secure, organizations might also feel more secure when moving towards the cloud because the cloud provider might have the capabilities and expertise to deal with security issues, while the organization moving towards the cloud might not have these capabilities in-house. This can typically be the case for smaller organization unable to afford these capabilities. However, for both small and large organizations, security and trust will always be something having to be considered.

In terms of costs we believe that the pay-per-use model is an interesting model for many organizations because it provides reduced startup costs and a way to financially cope with variations in system usage (elasticity). Data transfer can still be expensive though (around $100 per TB). When dependent on cloud providers, large organizations may encounter potential other cost issues because economies of scale cannot be exploited at the organization itself anymore. It is our belief that the pricing issues that still exists will resolve themselves as the market of cloud computing grows more competitive. Amazon for example has reduced pricing a number of times already.

Elasticity can lead to saving money, because less capabilities will have to be rented at the cloud provider if usage of the system is low. In order to use elasticity, a cloud provider needs to offer the ability to rent or let go extra resources. Some of the big cloud providers offer this possibility. Rackspace for instance allows the customer to manage the number of nodes rented by using an API. If a customer uses this service, his/her system will be taken offline momentarily in order to reallocate resources. Amazon offers the ability to manage the number of nodes rented automatically, by using customer-defined rules. This service has potential because customers do not have to manage resources manually in order to be able to save money. Elastic services are a significant (financial) argument in favor of moving data warehousing systems towards the cloud. Not all cloud providers offer capabilities for elasticity yet, but is our belief that in order to be competitive a lot of providers will do so in the future.

4.2 Functional requirements

In order to adapt and become more suited for the cloud environment data warehousing systems would have to comply with the functional requirements listed below. While data warehousing systems in general have many more functional requirements, we only state the most important ones specifically relevant to moving towards the cloud. This list of requirements is largely based on [43], as well as our discussion in section 4.1.

- High performance.
    - Data needs to move from persistent cloud storage towards compute nodes or between compute nodes in a fast way. Moving significant amounts of data (up to terabytes) can be needed when new nodes become active (in order to deal with a changing workloads for example) as well as during query processing. This is a challenge because storage and network bandwidth in the cloud are generally not fast compared to traditional data warehousing systems. A possible (partial) solution is to use compression to reduce bandwidth usage while paying the price of higher CPU usage.
    - The capabilities of virtual machines will have to be exploited by data warehousing systems in the cloud, resulting in low-level technical issues [5]. While traditional data warehousing systems generally do not make use of virtual machines, data warehousing systems in the cloud will likely have to because VM's are generally the platform offered by cloud providers. Virtual machine optimizations are also discussed in [15].

    - Data warehousing systems in the cloud will have to make good use of heterogeneous resources in the cloud, because data warehousing systems in the cloud can no longer depend on for example a stable and homogeneous cluster environment. Performance in the cloud is much less predictable than it is in a local cluster environment, because varying hardware (different from traditional data warehousing systems), concurrent users and irregular communication speeds influence performance. Communication speeds can be irregular because sometimes nodes can be close to each other from a networking perspective (e.g. on the same network switch), while sometimes they can be far away from each other. This issue can potentially be (partially) resolved by placing data warehouse nodes in the cloud as close (from a networking perspective) to each other as possible.
- Flexibility (e.g. elasticity).
    - In order to achieve elasticity data warehousing systems in the cloud will have to be able to scale up and down automatically once workloads, amounts of users or data volumes increase or decrease. Partitioning can be used in order to distribute data across different nodes in the cloud, comparable to the situation in traditional (cluster) data warehousing systems. Each node can receive a number of these partitions (or 'shards'). In a traditional (cluster) data warehousing system, partitioning in and replication of shards is static and performed by the database administrator. In a cloud a much more dynamic system is required. If scaling in the cloud is required, new nodes can be brought on-line to deal with this, resulting in a possible need to re-partition, re-replicate and/or re-distribute data based on changing usage patterns. As far as we are aware, dynamic algorithms used to determine how to re-partition, re-replicate and/or re-distribute data in the cloud including the criteria used for this do not exist yet and remain an open research question.
    - Load balancing is required in order to be able to fairly spread workloads over nodes in the cloud environment. This potentially brings some of the same aforementioned issues regarding the re-partitioning, re-replication and/or re-distributing of data based on changing usage patterns.
    - Data warehousing systems in the cloud will have to support 'user-driven scaling'. This means that the system will have to support higher performance at a higher price, as well as lower performance at a lower price, depending on user preferences. This brings economic issues into the equation, like what can be charged for a specific level of performance. Issues involving performance guarantees also play a role; a specific level of performance can be hard to guarantee because, like we discussed before, performance in the cloud can be unpredictable compared to traditional systems.
    - In order to achieve flexibility data warehousing systems in the cloud must also make good use of heterogeneous resources. For example query planning will have to be aware of the heterogeneous environment in order to react to for example slow nodes, which may mean that workloads may have to be distributed differently or new nodes may have to be brought on-line in order to compensate.
- Multi-tenancy. Data warehousing systems in the cloud will have to be able to deal with multiple different users potentially using the same database server.
    - Data warehousing systems in the cloud will have to make sure that database schema's are strictly separated. Different tenants must not be able to see each other's data even though data may be placed at the same physical machine or even in the same table.
    - Data warehousing systems in the cloud should be able to meter used storage, time, memory, the number of queries and network communication by different tenants. A SaaS data warehouse product can then for example charge customers based on these measures. Measurements can help in deciding how to achieve

better performance because measurements can help in detecting for example changes in usage patterns.

- Infrastructure.
  - In terms of infrastructure data warehousing systems in the cloud will need API's that allow local system entrance without going via the main server continuously. This can help in reducing latency for example, because if a data warehouse server is on the other side of the world, latency can potentially become an issue (although, like we discussed, it will not likely be a bottleneck). Having to communicate over large geographical distances is a new issue in data warehousing applied in the cloud, because traditional data warehousing systems are generally situated at or close to the owning organizations.
  - Data needs to be efficiently imported and exported towards and from the cloud. To achieve this, compression needs to be used whenever possible, because this reduces the amount of bandwidth needed to transfer data. Making importing and exporting data resilient to failure might also help, because this can for example reduce the amount of re-starts of importing/exporting needed. Bandwidth might possibly always be an issue, because, like we discussed before, a WAN link is involved.
  - It must be easy and fully automated to deploy a data warehouse in the cloud. Ease of deployment can be an important argument for organizations deciding whether to use traditional or cloud based data warehousing systems.
  - Recovery from failures (back-up recovery) should be automated in cloud based data warehousing systems. To achieve this back-ups should be easy to reach, which can possibly be achieved by residing back-ups outside the cloud or at different data centers. Incremental backups are of the essence.
- Privacy. Data warehousing systems in the cloud must be able to encrypt data locally in order to ensure privacy. The possibility to operate on encrypted data would be a valuable addition. This does lead to previously discussed technical issues [16] because encrypted data tends to require different forms of analysis.
- Security. Data warehousing systems in the cloud must be secure so that the data warehouse including all forms of communication to the data warehouse are accessible only to the original customer. Security can lead to many issues discussed before.
- Monitoring. Monitoring capabilities have to be available for customers and administrators in order to analyze the systems operations. Potential bottlenecks (e.g. 'killer-queries') must also be detected and cancelled when required. Monitoring capabilities are important in achieving for example elasticity and load balancing; changing usage patterns need to be detected in order to be able to decide whether or not scaling up/down is required, or whether or not loads need to be rebalanced.
- Availability & reliability. Data warehousing systems in the cloud must be available at all times to the customer and data warehousing systems in the cloud must be highly reliable. Having a separate replication server could help, because it possibly allows dealing with system failures, like for example power-outages, automatically.

Besides the fact that mostly data warehousing systems will have to adapt in order to enable them to operate effectively inside the cloud, there are also requirements that cloud providers will still have to comply with in order for organizations to gain interest in moving their data warehousing system towards the cloud. These requirements include:

- High performance. The cloud provider must offer performance guarantees, and high bandwidth needs to be offered. Bandwidth will possibly always be an issue, because the cloud depends on WAN connectivity. In order to achieve high performance cloud providers can use several approaches. For example Amazon offers VM's to operate on while Microsoft's SQL Azure offers a specific database service without the use of VM's (databases operate directly on physical machines).

- Scalability. The cloud environment will have to be scalable in order to comply with the needs of ever-growing data warehouses. Currently this requirement is generally offered by all major cloud providers, although SQL Azure databases are still limited to a maximum of 50GB.
- Flexibility (e.g. elasticity). The cloud environment will have to support both scaling up and down (by for example letting customers rent more or less nodes) when users require this.
- Infrastructure. It must be easy to deploy data warehousing on the cloud and recovery from crashes must be automated (back-ups). The infrastructure must be well maintained and administered.
- Security. Data in the cloud must be well protected by the cloud provider.
- Reliably. The cloud must provide a reliable environment in terms of performance as well as availability. The customer must be able to rely on storage guarantees including storage availability, persistence and performance.
- Service. The customer must experience a certain quality level of service offered by the cloud provider. The cloud provider must live up to service levels advertised.

Many of the above requirements, like flexibility and scalability, are starting to be offered by cloud providers. Performance and security remain large issues within cloud computing. Performance and security guarantees are often vague or barely offered at all. However, it is our belief that in the future more and more cloud providers will start to comply with these requirements.

Currently there are still a lot of issues like the ones discussed above in need of being solved by suppliers of data warehousing products, in order to make data warehousing systems more cloud-friendly. This is for a large part due to the fact that the cloud environment is significantly different from a traditional (cluster) environment. Also cloud providers can help, by making the cloud environment more attractive for data warehousing systems.

5. Related work

Throughout this section we will look at existing efforts to combine data warehousing and cloud computing including other relevant efforts, allowing us to see whether or not positive results have already been achieved within this field, or what the vision of others regarding data warehousing or database systems in general moving towards the cloud is. Most work has been done in the field of moving OLTP databases towards the cloud. As far as we are aware there are no significant attempts of moving large scale data warehousing systems into the cloud.

D.J. Adabi explores the limitations and opportunities of moving data management into the cloud in [4]. Conclusions that come forward from this paper are that current database systems are not yet particularly suited for moving into the cloud. However, the author argues that decision support systems (which includes data warehousing systems) are the most likely database systems to take advantage of the cloud. Reasons to support this claim are:

- A shared nothing architecture works well for analytical data managements as well as for the cloud
- ACID (atomicity, consistency, isolation, durability) properties are important for relational databases and while these properties are hard to guarantee in the cloud, they are typically not needed for data warehouses
- Sensitive data in terms of security can often be left out of the analysis, making security less of an issue.

In [8], M. Brantner et al. attempt to build a database system on top of Amazon's S3, while focusing on OLTP. Though the paper does not focus on data warehousing, it is interesting and relevant to see whether a simple database system can be build on S3. M. Brantner et al. do not achieve the ACID properties, but these are typically not important for data warehousing [4]. Some promising results are presented. The paper is limited in the sense that it does not explore the possibilities of Amazon's EC2 service in detail.

In [9], D. Lomet et al. propose an architecture for making transactional databases (OLTP) more suited for deployment in the cloud. It is proposed that the database system be split up into 2 types of components: transaction components (TC's) and data components (DC's), supposedly making the database more suited for operating inside the cloud due to for example flexibility. The data components do not know anything about the transactions. Multiple TC's can be linked to multiple DC's, allowing a database to be partitioned across multiple DC's, and making the architecture interesting for data warehousing from a parallelizability perspective.

S. Das et al. propose a different architecture in [3]. The architecture, called ElasTraS (Elastic TranSactional relational database), is meant to be deployable on the cloud while being fault tolerant and self managing. The architecture consists out of the following components:

- Distibuted Fault-tolerant Storage (DFS) for the distributed storing of data.
- Owning Transaction Managers (OTM's) that can exclusively 'own' multiple partitions in the DFS.
- TM Master responsible for assigning partitions to OTM's as well as monitoring OTM's including load balancing, elastic scaling and recovering from failures.
- Metadata Manager (MM) maintaining the mapping of partitions to their owners.

While S. Das et al. focus at transactional databases, we believe some of their architectural principles are also applicable to data warehousing. The distributed nature of the architecture makes the architecture potentially suited for parallel workloads.

In [10] A. Aboulnaga et al. describe some of the challenges involved in deploying databases onto the cloud. Challenges empathized are placement of VM's across physical machines, the partitioning of resources across VM's and dealing with dynamic workloads. These challenges, if

not solved, could impact customer willingness to move their data warehousing system towards the cloud. A solution to the partitioning of CPU capacity is provided.

A way to provide adaptive workload execution in the cloud (elasticity) is provided in [11]. More about workload management can be found in [12], where it is argued that query interactions should be taken into account when managing workloads. I/O performance for database systems is analyzed in [13]. In [14], machine learning techniques are proposed to predict query execution times, which can be important regarding workload management in for example distributed and parallel database systems. A technique using VM's to separate database instances is introduced in [15], which is relevant for example for multi-tenant hosting in the cloud (more customers operating on a single physical machine).

Distributing databases can be important while using multiple nodes in the cloud in order to run a data warehousing system. Leading papers in distributed database systems include [6] and [7].

6. Conclusion

Throughout this paper we have surveyed both the field of data warehousing and the field of cloud computing. We have discussed what the possibilities and impossibilities are for combining the two, in order to potentially provide elastic data warehousing that is both cost-effective and efficient via the cloud. It is our belief that data warehousing systems in the cloud have great potential, due to potential for elasticity, scalability, deployment time, reliability and reduced costs (due to e.g. elasticity). However, we believe that the current products in cloud computing are not yet instantly capable of performing large-scale data warehousing due to current-date in-cloud performance, data transfer speed and pricing issues. In order for a data warehousing system to be able to utilize the capabilities of the cloud it will have to be both highly parallel and distributed, while complying with many functional requirements discussed in this paper. Furthermore, we believe that in the short term data marts may have more potential in the cloud compared to data warehousing systems because they tend to be smaller due to their specialized nature allowing them fit in less or just a single node. This makes distributing and parallelizing less of an issue. Security issues will likely always be involved in the decision of moving a data warehousing systems or data marts into the cloud.

Further research in combining data warehousing and cloud computing is needed. Our work regarding data warehousing in the cloud is largely hypothetical and for example benchmarking the cloud may lead to new insights in the possibilities and impossibilities of deploying data warehousing systems into the current-date cloud environment. Support by Amazon for IBM's DB2 and Oracle's database 11g is promising, and testing these recently added capabilities may lead to more insights in the applicability of these systems in the cloud. The introduction of a general architecture capable of utilizing the potential of the cloud for data warehousing may have significant value as well.

Acknowledgements

Special thanks go to Peter Boncz for both his extensive feedback and invested time throughout this project. Without his support this project would not have been possible.

Thanks to Marcin Zukowski for his valuable contribution in terms of functional requirements.

References

[1] Amazon web services
http://aws.amazon.com/

[2] S. Chaudhuri, U. Dayal: An Overview of Data Warehousing and OLAP Technology. In ACM Sigmod record, 1997

[3] S. Das, S. Agarwal, D. Agrawal, A.E. Abbadi: ElasTraS: An Elastic, Scalable, and Self Managing Transactional Database for the Cloud. In USENIX HotCloud, 2009

[4] D. Abadi: Data Management in the Cloud: Limitations and Opportunities. In Data Engineering, 2009

[5] P.M. Chen, B.D. Noble: When Virtual Is Better Than Real. In: Proceedings of the 3rd conference on Virtual Machine Research And Technology Symposium, 2004

[6] J.B. Rothnie, P.A. Bernstein, S. Fox, N. Goodman, M. Hammer, T.A. Landers, C. Reeve, D.W. Shipman, E. Wong: Introduction to a System for Distributed Databases (SDD-1) In ACM Transactions on Database Systems (TODS), 1980

[7] M.T. Özsu: Distributed database systems. Prentice Hall, Englewood Cliffs, 1991

[8] M. Brantner, D. Florescu, D. Graf, D. Kossmann, T. Kraska: Building a Database on S3. In SIGMOD '08 Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008

[9] D. Lomet, A. Fekete, G. Weikum, M. Zwilling: Unbundling Transaction Services in the Cloud. In CIDR Perspectives, 2009

[10] A. Aboulnaga, K. Salem, A.A. Soror, U.F. Minhas, P. Kokosielis, S. Kamath: Deploying Database Appliances in the Cloud. IEEE Data Eng. Bull., 2009

[11] N.W. Paton, M.A.T. de Aragão, K. Lee, A.A.A. Fernandes, R. Sakellariou: Optimizing Utility in Cloud Computing through Autonomic Workload Execution. IEEE Data Eng. Bull, 2009

[12] M. Ahmad, A. Aboulnaga, S. Babu: Query Interactions in Database Workloads. In DBTest '09 Proceedings of the Second International Workshop on Testing Database Systems, 2009

[13] W.W. Hsu, A.J. Smith, H.C. Young: I/O Reference Behavior of Production Database Workloads and the TPC Benchmarks - An Analysis at the Logical Level. In ACM Trans. Database Syst., 2001

[14] A. Ganapathi, H.Kuno, U.Dayal, J.L. Wiener, A. Fox, M. Jordan, D. Patterson: Predicting Multiple Metrics for Queries: Better Decisions Enabled by Machine Learning. In Proceedings of the 2009 IEEE International Conference on Data Engineering, 2009

[15] A.A. Soror, U.F. Minhas, A. Aboulnaga, K. Salem, P. Kokosielis, S. Kamath: Automatic Virtual Machine Configuration for Database Workloads. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008

[16] N. Ahituv, Y. Lapid, S. Neumann: Processing encrypted data. ACM, 1987

[17] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Pattersom, A. Rabkin, I. Stoica, M. Zaharia: Above the Clouds: A Berkeley View of Cloud Computing, 2009

[18] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. In: Future Generation Computer Systems, Volume 25, Issue 6, 2009

[19] J. Gray, D. Patterson: A conversation with Jim Gray. In: ACM Queue, 2003

[20] J.Wijsen: Een kort overzicht van data warehousing en OLAP. 2006

[21] E.Rahm, H.H.Do: Data Cleaning: Problems and Current Approaches. IEEE Data Engineering Bulletin, 2000

[22] Oracle webpage about OLAP
http://download.oracle.com/docs/html/B13915_04/i_olap_chapter.htm

[23] S. Chaudhuri, U. Dayal: Database Technology for Decision Support Systems, 2001

[24] J. Goldstein, P. Larson: Optimizing Queries Using Materialized Views: A Practical, Scalable Solution. In: ACM SIGMOD Record, 2001

[25] A. Gupta, I.S. Mumick: Maintenance of Materialized Views: Problems, Techniques, and Applications. In: Data Engineering Bulletin, 1995
[26] C. Zhuang, X. Yao, J. Yang: An Evolutionary Approach to Materialized Views Selection in a Data Warehouse Environment. In: IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2001
[27] D.J. De Witt, J. Gray: Parallel database systems: The future of high performance database processing. CACM, 1992
[28] M. Mehta, D.J. De Witt: Data placement in shared-nothing parallel database systems. In: The VLDB Journal, 1997
[29] L. Bellatreche, K. Boukhalfa: An Evolutionary Approach to Schema Partitioning Selection in a Data Warehouse. DaWaK 2005
[30] IBM (general US website) http://www.ibm.com/us/en/
[31] Oracle (general US website) http://www.oracle.com/us/index.html
[32] Teradata (general website) http://www.teradata.com/
[33] IBM DB2 http://www-01.ibm.com/software/data/db2/
[34] Oracle Database 11g http://www.oracle.com/us/products/database/index.html
[35] Teradata Database 13.10 http://www.teradata.com/products-and-services/database/teradata-13/
[36] Rackspace http://www.rackspace.com/
[37] Microsoft Azure http://www.microsoft.com/windowsazure/
[38] D.C. Zilio, C. Zuzarte, S. Lightstone, W. Ma, G.M. Lohman, R.J. Cochrane, H. Pirahesh, L. Colby, J. Gryz, E. Alton, D. Liang, G. Valentin: Recommending materialized views and indexes with IBM DB2 design advisor. ICAC, 2004
[39] C.A. Lang, B. Bhattacharjee, T. Malkemus, K. Wong: Increasing Buffer-Locality for Multiple Index Based Scans through Intelligent Placement and Index Scan Speed Control. Proceedings of the 33rd international conference on Very large data bases, 2007
[40] S. Padmanabhan, B. Bhattacharjee, T. Malkemus, L. Cranston, M. Huras: Multi-Dimensional Clustering: a new data layout scheme in DB2. Proceedings of the ACM SIGMOD international conference on Management of data, 2003
[41] SkyInsight web page http://www.ingres.com/products/cloud/skyinsight
[42] GoodData general website http://www.gooddata.com/
[43] M. Zukowski et al.: Aspects of a DBMS service in a cloud. 2011