Managing Cloud-Based Big Data Platforms: A Reference Architecture and Cost Perspective

Leonard Heilig and Stefan Voß

Abstract The development of big data applications is closely linked to the availability of scalable and cost-effective computing capacities for storing and processing data in a distributed and parallel fashion, respectively. Cloud providers already offer a portfolio of various cloud services for supporting big data applications. Large companies like Netflix and Spotify use those cloud services to operate their big data applications. In this chapter, we propose a generic reference architecture for implementing big data applications based on state-of-the-art cloud services. The applicability and implementation of our reference architecture is demonstrated for three leading cloud providers. Given these implementations, we analyze main pricing schemes and cost factors to compare respective cloud services based on a big data streaming use case. Derived findings are essential for cloud-based big data management from a cost perspective.

Keywords Big data management · Cloud-based big data architecture · Cloud computing · Cost management · Cost factors · Cost comparison · Provider selection · Case study

1 Introduction

The cloud market for big data solutions is growing rapidly. Besides full-service cloud providers that offer a large portfolio of different infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) solutions, there are also some niche providers focusing on specific aspects of big data applications. In general, such big data applications are highly dependent on a scalable computing infrastructure, programming tools, and applications to efficiently process large data

L. Heilig (✉) · S. Voß
Institute of Information Systems (IWI), University of Hamburg, Hamburg, Germany
e-mail: leonard.heilig@uni-hamburg.de

S. Voß
e-mail: stefan.voss@uni-hamburg.de

F.P. García Márquez and B. Lev (eds.), *Big Data Management*,
DOI 10.1007/978-3-319-45498-6_2

sets and extract useful knowledge [17]. In this regard, cloud computing represents an attractive technology-delivery model as it promises the reduction of capital expenses (CapEx) and operational expenses (OpEx) [11] and further moves CapEx to OpEx, closely correlating expenses with the actual use of tools and computing resources [5]. A recent scientometric analysis of cloud computing literature further indicates that there is a huge research interest in scalable analytics and big data topics [10]. As cloud-based big data applications are usually composed of several managed cloud services, it becomes increasingly important to identify important cost factors in order to evaluate potential use cases and to make strategic decisions, for instance, concerning the choice of a cloud provider (for an extensive overview on decision-oriented cloud computing, the reader is referred to Heilig and Voß [9]). The variety of possible configurations and pricing schemes makes it difficult for consumers to estimate overall costs of cloud-based big data applications. Often, consumers appear in the form of cloud application providers, as companies like Netflix and Spotify, outsourcing operations of their services to third-party cloud infrastructures. To benefit from big data technologies and applications in companies, it is meanwhile essential to address the economic perspective and provide means to evaluate the promises cloud computing gives with regard to the use of highly scalable computing infrastructures in order to unlock competitive advantages and to maximize value from the application of big data [18]. To the best of our knowledge, a cost perspective for implementing big data applications in cloud environments has not yet been addressed in the current literature.

In this chapter, we propose a generic reference architecture for implementing big data applications in cloud environments and analyze pricing schemes and important cost factors of related cloud services. The cloud reference architecture considers state-of-the-art technologies and facilitates the main phases of big data processing including data generation, data ingestion, data storage, and data analytics. Both batch and stream processing of big data is supported. We demonstrate the applicability and implementation of the proposed architecture by specifying it for the cloud services of the, according to Gartner's magic quadrant [6], three leading cloud providers, namely *Amazon Web Service*, *Google Cloud*, and *Microsoft Azure*. Practical implementations of large companies like Netflix and Spotify verify the relevancy of the defined architectures. The individual architectures provide a basis for evaluating important cost factors. For each of the main phases of big data processing, we identify and analyze the scope and cost factors of relevant cloud services based on a case study. In cases a comparison is useful, we compare cloud services of the different cloud providers and derive important implications for decision making. Thus, the contribution of this chapter is twofold. First, the chapter provides a blueprint for implementing state-of-the-art cloud-based big data applications and gives an overview about available cloud services and solutions. Second, the main part is concerned with providing a cost perspective on cloud-based big data applications, which is essential for big data management for cloud consumers.

The chapter is structured as follows. Section 2 defines the main phases of big data processing and presents the generic reference architecture. Moreover, we describe the implementation of the reference architecture using cloud services of the three

leading cloud providers. For each big data processing phase, cloud pricing schemes and relevant cost factors of those cloud services are analyzed based on a case study focusing on streaming analytics in Sect. 3. In Sect. 4, we discuss main findings and implications. Finally, we draw conclusions and identify activities for further research.

2 Big Data Processing in Cloud Environments

The calculation of costs is highly dependent on the utilized cloud services. Major cloud service providers offer a plethora of different tools and services to address big data challenges. In this section, we define a common reference architecture for big data applications. The reference architecture corresponds to the state-of-the-art and supports main phases of big data processing from data generation to the presentation of extracted information, as depicted in Fig. 1. After briefly explaining these phases and the corresponding reference architecture, we give an overview on its implementations with cloud services of the three leading cloud service providers.

2.1 Generic Reference Architecture

The processing of big data can be divided into five dependent phases. In the first phase, data is generated in various applications and systems. This might include internal and external data in various forms and formats. Depending on the rate of occurrence and purpose of collected data, velocity requirements may differ among data sources. The second phase involves all steps to retrieve, clean, and transform the data from different sources for further processing. This may include, for instance, data verification, the extraction of relevant data records, and the removal of duplicates in order to ensure efficient data storage and exploitation [3]. Typically, the data is permanently stored in a file system or database. In some cases of streaming applications, however, value can only be achieved in the first seconds after the data is produced, making a persistent storage obsolete. Nevertheless, information and results being extracted during processing and analysis usually need to be stored and managed permanently. In the fourth phase, different methods, techniques, and systems are used to analyze and utilize the data in order to extract information relevant for

Fig. 1 Phases of big data processing

supporting business activities and decision making. The information and results of the data analytics phase need to be visualized, allocated, distributed, and presented to its users in the final phase.

To define a generic reference architecture, we reviewed several technical documentations, recommendations, and use cases provided by cloud providers (see, e.g., [2]) and interviewed experts of one of the largest cloud providers. Moreover, we analyzed practical implementations of cloud-based big data platforms of large cloud consumers, such as Netflix [12] and Spotify [15], using cloud services of different large cloud providers. Considering both batch and real-time data stream processing, we identify a common structure of systems interacting with each other to support the different phases of big data processing. To express this structure, we present a state-of-the-art generic reference architecture in Fig. 2. The arrows represent the data flows. Note that variations of this generic architecture and associated data flows are possible. In the following, we briefly explain the basic components of the reference architecture.

Data Generation: The number of data producers and the amount of data being produced is continuously increasing. This involves business data from internal systems (e.g., production data, inventory data, sales data, e-commerce platform data, etc.) and data from external third-party systems (e.g., social network data, government data, weather data, finance data, search trends, etc.) being offered through the Internet. The emergence of the internet of things (IoT), enabling physical objects to sense and act on their environment by interacting with each other, represents another big data source.

Data Ingestion: For the data ingestion, it is essential to consider velocity requirements, which mainly determine how fast the data is fed into the overall system and the processing latency between data generation and presentation. Thus, a system architecture must consist of components that support both real-time and batch processing of data. The former must be supported by systems that enable a fault-tolerant, scalable, and consistent real-time processing of *data streams* from a large number of

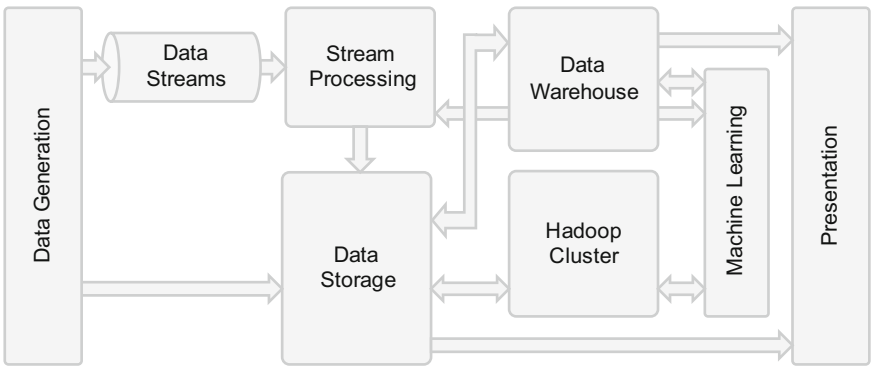


Fig. 2 Generic reference architecture

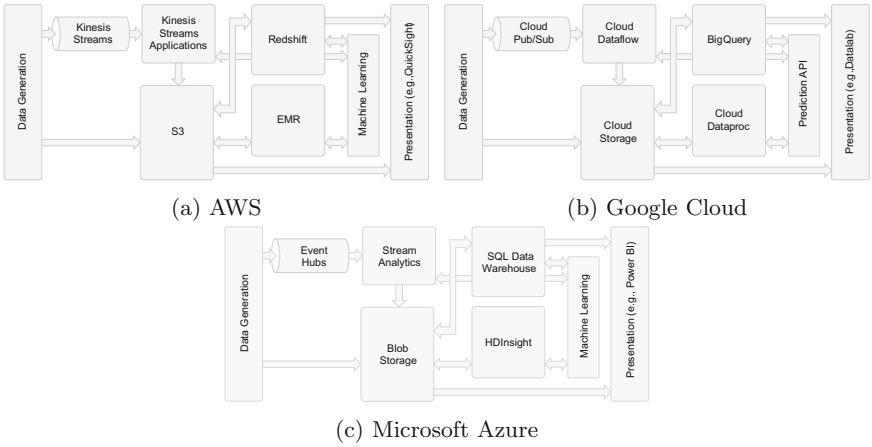
data sources. To harvest its potential, it is necessary to implement applications and methods to further process or analyze streaming data immediately, depending on the processing latency requirements. Therefore, the *stream processing* component contains all logic to immediately utilize streaming data, for instance, by generating alerts or making recommendations based on machine learning tools. If useful for future processing, streaming data is transferred to the central persistent *data storage* component to be stored permanently. This may involve extract, transform, and load (ETL) jobs that reliably export the data into a central storage or directly to a central processing cluster (see, e.g., a recent streaming solution of Spotify using *Google Cloud* [15]). Typically static data with a low velocity is initially stored in a database or file system. To further process and analyze this data, for instance in a data warehouse, those data sources may need to be consolidated and integrated with additional ETL jobs.

Data Storage: For permanent use in form of batch processing, data should be persisted as files or data records in file systems or databases designed to provide scalability, high availability, and low latency. From there on, the data can be used by several distributed applications in parallel. For data analytics in a data warehouse, it is necessary to load the data into database tables of the data warehouse using ETL. A permanent storage of data in an Apache Hadoop cluster is often economically unreasonable as it involves huge costs for using necessary cluster nodes in form of virtual machines (VM). Large cloud consumers, like Netflix [12], show that the integration with low cost cloud storage services instead of using local storage based on the Hadoop Distributed File System (HDFS) [16] can be beneficial. For instance, multiple clusters can access and process the same data for different workloads depending on the data analysis task [12]. However, reading and writing to a central file system is slower than using local storage and thus requires a low-latency and high-bandwidth access. As a compromise, local storage of HDFS is often used for all intermediate stages of MapReduce processes. Consequently, a mixture of a *shared nothing architecture*[1] and *shared storage architecture*[2] approach is often used in practice for operating Hadoop clusters efficiently.

Data Analytics: The generic architecture further supports ad-hoc data queries and advanced data analytics. For providing related cloud services, cloud providers leverage their capabilities in providing massive and scalable computing infrastructure. While a data warehouse aids the processing and analysis of structured data, a Hadoop cluster can be used to process and transform unstructured and semi-structured data into structured data for further processing in databases and *data warehouses*. For the latter, the Hadoop ecosystem provides several extensions for data processing, querying, storage in NoSQL databases (e.g., HBase) and data warehouses (e.g., Hive) as well as advanced statistical and machine learning algorithms

[1] A shared nothing architecture denotes a distributed computing architecture consisting of nodes that only possess and utilize their own computing resources including memory and disk storage. This facilitates, inter alia, a large scale horizontal scaling using commodity machines based on a distributed file system.

[2] In a shared storage environment, a central file storage system is shared among the nodes.



(a) AWS (b) Google Cloud

(c) Microsoft Azure

Fig. 3 Reference architectures of leading cloud service providers

(e.g., Mahout). Thus, a data warehouse is often build upon a Hadoop cluster. Instead of using the MapReduce programming model for processing data in a cluster, it has been shown that Apache Spark, building on in-memory storage to avoid slow disk reads/writes as well as directed acyclic graph (DAG) scheduling to better parallelize processing stages, offers up to two orders of magnitude performance increase [14]. For structured data queries, it is essential that the data warehouse service supports SQL[3] commands. Moreover, a component that provides *machine learning* algorithms and technology for supporting artificial intelligence and predictive analytics shall be provided as a service.

Presentation: Once the data has been analyzed and stored, normally in a data warehouse, resulting insights and results need to be transformed into rich visualizations, dashboards, and reports for individual stakeholders in order to avoid information overload and decrease transaction costs. Therefore, tools to prepare and manage rich visualizations and reports need to be provided.

2.2 Implementations of the Generic Reference Architecture

In this section, we show the technical implementation of the proposed generic big data processing reference architecture. For this purpose, we have reviewed the technical details and architectures of managed cloud services offered by the three leading cloud providers, namely *Amazon Web Services (AWS)*, *Google Cloud*, and *Microsoft Azure*. For each cloud provider, we present a reference architecture in Fig. 3 describing the use of managed cloud services for supporting each phase of

[3] Abbr. for Structured Query Language.

big data processing as described in Fig. 1. A prerequisite for considering a cloud service was that it is fully compatible and integrated with other cloud services of the respective provider. In this regard, all three cloud providers are able to address each phase of a big data processing lifecycle with at least one managed cloud service and further support their integration as well as the use of third-party services. The cloud services conform to the above mentioned requirements (see Sect. 2.1). The proposed composition and an associated integration of those cloud services is fully supported and in line with best practices. As the primary focus is on cost management, details on the implementation of the proposed architectures are out of scope in this chapter. For an overview and technical details, the reader is referred to the documentation of each cloud service given by the respective cloud provider.[4] Generally, those examples demonstrate the applicability of the proposed generic reference architecture.

3 Cloud Pricing and Cost Perspective

After defining the reference architectures for each cloud provider, we investigate possible configurations and the pricing schemes for each category of cloud service in this section. In doing so, we identify main cost factors. Based on a real-time data streaming example, we present a cost comparison for the cases where similar pricing schemes and configurations are possible.

3.1 Data Streams and Stream Processing

By analyzing the cost models of data streaming cloud services, we identify two different approaches. *Amazon Kinesis Streams* charges for each message event occurring during data ingestion and delivery, whereas no additional costs are charged for the required throughput. Although volume-tiered pricing is supported, the prices per million message events are comparatively high. *Kinesis Streams* and *Azure Event Hubs* use a common cost model that calculates the costs based on the volume of incoming message events and required throughput. That is, the stream is grouped into smaller streams (i.e., substreams) with a maximum throughput. Thus, the number of substreams needs to be scaled according to the current message load for achieving real-time processing. Consequently, both volume and speed scaling is covered in the cost models of those cloud providers.

In Table 1, we see that *Kinesis Streams* has competitive prices and is therefore able to provide the most inexpensive solution, also in terms of scaling. A difference between *Kinesis Streams* and *Event Hubs* is the maximum message event size.

[4]The technical documentation and pricing details can be found on the website of the respective cloud providers: AWS (https://aws.amazon.com), Google Cloud (https://cloud.google.com), and Microsoft Azure (https://azure.microsoft.com/).

Table 1 Cost comparison of cloud streaming services (Scenario 1: 100 data records/s, 35 KB record size, 3.42 MB/s input stream, 1 subscriber)

Configuration/pricing	Kinesis streams	Pub/sub*	Event hubs
Max. message event size in KB	25	64	64
Max. thoughput ingress in MB/s per substream	1	N/A	1
Max. thoughput egress in MB/s per substream	2	N/A	2
Message retention in days	1	7**	1
Pricing scheme	Fixed	Tiered	Fixed
Price per million message events per hr ($)	0.014	0.40/0.20/0.10/0.05***	0.028
Price streaming per throughput unit per hr ($)	0.015	N/A	0.03
Required number of substreams (ingress)	4	N/A	4
Additional number of substreams (egress)	0	N/A	0
Number of message events per day in million	17.28	17.28	8.64
Costs ($)			
Overall costs per day	**1.68**	6.91	3.12
Overall costs per month	**52.14**	214.27	96.78

*Assuming that the push subscription mode is used
**If the subscriber is not present; otherwise, messages are dropped after delivery/failure
***First to 250M/next 500M/next 1000M/next 1750M message events

Assuming that the prices of the two providers would be the same, *Event Hubs* would provide a cost advantage for data records sizes greater than 25 KB due to the smaller amount of message events (see Table 1). Thus, also the size of data records of an application may need to be taken into account for cost considerations. When assuming that *Google Cloud Pub/Sub* would provide more attractive prices, the service might be economically advantageous in terms of throughput scaling. In general, we identify two main cost factors: throughput capacity (ingress and egress) and number of message events.

Next, we analyze the portfolio of cloud services for real-time stream processing and analytics. All cloud services allow an individual creation of jobs to read, transform, and analyze streaming data. While Google's *Dataflow* has established a programming model to simplify the implementation of data processing jobs, *Azure Stream Analytics* focuses on structured data processing and allows running SQL-like queries. Although *AWS* has announced that the managed service *Kinesis Analytics* will be available soon, consumers currently have to implement applications using the *Kinesis Client Library* (*KCL*) and different connectors for establishing a link to other cloud services (e.g., *S3*), similar to the *Dataflow* approach. All three cloud services allow an integration with both their own and third-party machine learning cloud services. In general, the costs of running all three cloud solutions are highly dependent on the computational demands of processing and analytics tasks in terms of compute and storage requirements.

Table 2 Cost factors of streaming analytics cloud services

Kinesis analytics	• Number and types of VM cluster nodes
	• Storage capacity
Dataflow	• Number and types of VM cluster nodes
	• Number of GCEU (batch/streaming)
	• Storage capacity
Stream analytics	• Volume of streaming data to be processed
	• Compute capacity in streaming units

Besides costs for using available VM instance types and local storage capacity, *Dataflow* additionally charges per GCEU[5] and differentiates between streaming and batch mode. As shown in a recent implementation of Spotify's event delivery system, the streaming mode considerably decreases end-to-end latency for exporting message events from *Pub/Sub* to *Cloud Storage* [15]. Applying *KCL* further implies costs for using a *DynamoDB* that tracks state information in a cluster of workers that process the data from the stream and transfer the result to respective applications. In general, we see that both cloud providers offer a great flexibility regarding the configuration of the infrastructure and streaming application, but also require a high expertise. The approach of *Stream Analytics* aims to abstract from the infrastructure and defines a price per streaming unit[6] and data volume. Due to the different measures and approaches, it is difficult to compare those cloud services. Estimating the costs requires the collection of empirical data concerning the use of infrastructure for different processing and analytics jobs. However, the flexibility implied by *Dataflow* and *KCL* allows consumers to adapt infrastructure to their individual requirements, e.g., for achieving cost reductions and/or performance boosts. Due to complexity reasons, it is important to implement brokerage mechanisms for supporting related decisions (see, e.g., [8]). To analyze costs, simulation studies that consider different workload scenarios and configurations may provide more insights. In Table 2, the different cost factors of all solutions are shown. In general, we can identify two main cost drivers: compute and storage capacity.

3.2 Data Storage

The costs for storing data in a cloud are a critical aspect to be considered when planning cloud-based big data applications. All three cloud providers offer a variety of

[5]The Google Compute Engine Unit (GCEU) is used as a measure to calculate the total capacity of a virtual central processing unit (vCPU). Google's *Compute Engine* defines the GCEU for each VM instance type depending on the number of vCPUs.

[6]A streaming unit is a measure for expressing the computing capacity in terms of CPU and memory with a maximum throughput of 1 MB/s.

storage options and database systems. In our cost analysis, we focus on inexpensive standard object storage services that can be used to persist data in different formats and massive quantities. Implementations of Netflix [12] and Spotify [15] further emphasize the critical role of a central data object storage component for big data processing. Both volume-tiered and unit pricing schemes are used, as depicted in Table 3. We see that *Azure Storage* undercuts the prices of its competitors. Moreover, it is not differentiated between different types of object requests. The price per request is considerably low. While cost savings of 34.8 % on average can be achieved, the percentage of cost savings slightly increases by the amount of storage capacity. Fig. 4 shows the increase of costs dependent on the volume of data to be persisted. Thus, in particular in terms of scaling, *Azure Storage* offers attractive pricing.

Comparing *Amazon S3* and Google's *Cloud Storage*, we see that the high costs for "expensive" requests (e.g., PUT, LIST, etc.) greatly influence the overall storage costs. Thus, the costs are highly dependent on the use intensity and access patterns of consumers. This emphasizes the importance of data preprocessing activities (e.g., based on ETL). Moreover, all three cloud providers charge for data transfers. Incoming data flows and data transfers within a region is generally free; outgoing data flows and inter-regional data transfers are charged based on different unit prices. The latter needs to be considered, for example, if third-party services, such as machine learning services, are used. In general, we identify three main cost factors of cloud storage services: amount of virtual storage capacity, number and type of data requests, and data transfer.

Table 3 Cost comparison of cloud storage services (assuming that 10 % of the streaming data collected in scenario 1 is persisted in the same region and requires a PUT request and a GET request for each data record)

Pricing ($)	S3	Cloud storage	Azure storage
Pricing model	Tiered	Fixed	Tiered
Storage per GB per month	0.03/0.0295/0.029*	0.026	0.024/0.0236/0.0232*
PUT, COPY, POST, LIST requests	0.005**	0.10***	0.0036****
GET and other requests	0.004***	0.01***	0.0036****
Data transfer (same region)	Free	Free	Free
Costs			
Storage	784.03	680.06	617.69
GET requests	10.71	26.78	0.96
PUT/POST/other requests	133.92	267.84	0.96
Overall costs per day	29.96	31.44	**19.99**
Overall costs per month	928.66	974.69	**619.62**

*First 1 TB/next 49 TB/next 450 TB per month
**Per 1000 requests
***Per 10000 requests
****Per 1000000 requests

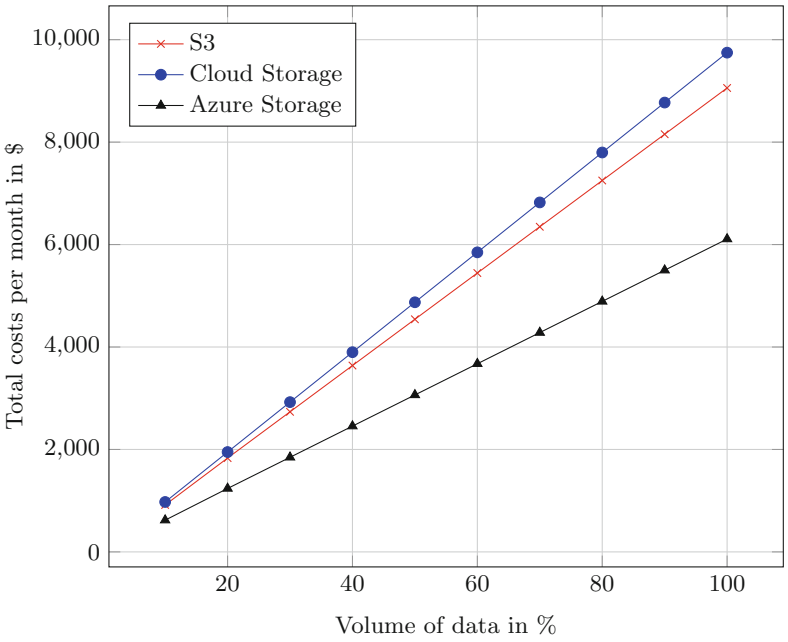


Fig. 4 Cost trends for increasing data volumes (26156–261563 GB)

3.3 Hadoop Cluster

For running a Hadoop cluster, all three cloud providers allow to use a wide range of different VM types and follow comparable pricing schemes. While for *Amazon EMR* and *Google Cloud Dataproc*, it is possible to choose among all VM types provided by *Amazon EC2* and *Google Cloud Compute Engine*, respectively, *Azure HDInsight* limits the range to certain VM types. As shown in Table 4, the price per VM node includes already the costs for using *HDInsight* and is about two times more expensive than using regular nodes of the same VM type. In the other two cases, extra costs for using the Hadoop service occur. *EMR* charges an individual fee per VM type and *Dataproc* calculates an extra fee per vCPU per hour. Thus, three main cost factors need to be considered: number and VM type of cluster nodes, additional usage fee (if applicable), and running time.

To approximately compare costs, we choose a comparable cluster configuration for each cloud provider (see Table 4). In general, we see that the increased price per cluster node leads to considerably higher costs in the case of *HDInsight*. While comparably low storage costs attract consumers to store their data in *Azure Storage*, the inherent comparative cost advantages disappear when it comes to large-scale processing of this data. This further emphasizes the need for an overall cost estimate. We further see that *Dataproc* offers competitive prices for using Google's *Compute*

Table 4 Cost comparison of Hadoop cluster services (clusters with 50 nodes, operated 5 h per day for processing 10 % of the streaming data)

Configuration/pricing	EMR	Dataproc	HDInsight
VM type	m3.xlarge	n1-standard-4	D3 v2
Number of vCPU	4	4	4
Memory in GB	15	15	14
Local SSD storage in GB	80	N/A	200
Additional SSD storage	120	200	N/A
Price VM per hour	0.27	0.20	Incl.
Price SSD per GB per month	0.10	0.22	Incl.
Price Hadoop per hour	0.07	0.04*	Incl.
Price per EMR node per hour	0.35	0.30	0.62
Costs			
Costs for VMs per day	88.03	**74.65**	155.50
Overall costs per month	2729.00	**2314.17**	4820.50

*Number of vCPU × Price per vCPU per hour

Engine services. However, the costs for additional SSD[7] storage are twice as high as the costs in *EMR*. Although the cluster configurations are comparable, the running time needs to be measured in order to estimate costs more accurately.

3.4 Data Warehouse

All three cloud providers offer managed solutions with data warehouse functionality for large-scale data analytics. Querying of massive amounts of data with SQL queries is one of their main features. Besides, the solutions support ETL processes based on different file formats. While the three cloud services offer similar functionality, different pricing approaches are used, as depicted in Table 5. In *Amazon Redshift* consumers define a computing cluster and are charged for the number and hours of utilized VM types. The available VM types are specifically designed for the purposes of *Redshift*. Each node comes with a fixed amount of local storage and it is not possible to separate storage capabilities. Consequently, a trade-off concerning the utilization of processing power and storage capacity will likely occur. That is, increasing the number of nodes for improving the querying performance might lead to unused storage capacities. Instead of charging for infrastructure components, Google's *BigQuery* prices storage capacity, streaming inserts,[8] and querying. Data operations like loading, copying, and exporting are free of charge. While *BigQuery* calculates costs based on the query capacity, which is dependent on the processed

[7] Abbr. for Solid-State Drive.

[8] Allows a direct transfer of data from Pub/Sub to BigQuery.

Table 5 Cost factors of data warehouse cloud services

Redshift	BigQuery	SQL data warehouse
• Number and type of VM cluster nodes	• Queries per capacity*	• Querying in DWU**
	• Amount of storage capacity	• Amount of storage capacity
	• Amount of streaming data***	

*According to the total amount of data processed in the selected columns
**A Database Warehouse Unit (DWU) measures the query performance
***Applies if streaming data is directly transferred to BigQuery

data volume per column and the column's data type, *Azure SQL Data Warehouse* (*DW*) charges in terms of query performance. Thus, the consumer is able to scale the speed of queries in *SQL DW*. As storage is charged separately per volume, consumers are only charged for the exact amount of storage needed to store table data. Due to the differences in the used pricing approaches, it is not possible to compare costs without determining the requirements of individual data analytics tasks. In general, we identify two main cost factors: compute and storage capacity.

3.5 Machine Learning

For predictive analytics, all three cloud providers provide managed machine learning engines that are integrated with both the storage and data warehouse solution of the respective cloud provider. *Amazon Machine Learning* (*ML*) and Google's *Prediction API*[9] focus on basic features to support common activities like the selection of data sources, explorative data analysis, model training, model evaluation, and model deployment. Less rigid is the approach of *Azure ML*, which provides an integrated SaaS application, referred to as *Azure ML Studio*, to individually define all steps of the data mining process as known from other data mining tools (e.g., *RapidMiner*). Besides, artificial intelligence algorithm APIs for vision, language, speech, and recommendations are available.

In general, we can identify three main cost factors: amount of computing capacity, number of transactions, and subscription fees. In Table 6, we give an overview of the individual cost factors per cloud service and provide two cost examples. Although the pricing approach is quite similar between *Amazon ML* and *Prediction API*, the main difference is that the latter calculates computing costs for training based on the volume of datasets. As the running time is closely linked to both the volume of training data and the complexity of the chosen learning algorithm, Google's pricing approach might be beneficial for the consumer as it does not consider complexity.

[9]Abbr. for Application Programming Interface.

Table 6 Cost factors and examples of machine learning cloud services (15 GB dataset size, 15000 streaming updates per day, 150 MB model size, 36 h of model generation, 30000 predictions per month)

Amazon ML	• Hours of data analysis and model training	
	• Number of batch predictions	
	• Number of real-time predictions	
	• Amount of reserved memory capacity	
Example ($)	Price per hour data analysis model training	0.42
	Price per real-time prediction	0.0001
	Price for reserved memory per 10 MB per hour	0.001
	Costs for computing capacities per month	15.12
	Costs for real-time predictions per month	41.16
	Overall costs per month	56.28
Prediction API	• Data volume for model training	
	• Number of streaming updates	
	• Number of predictions	
	• Number of projects (subscription)	
Example ($)	Monthly usage fee per project	10.00
	Price per MB bulk trained	0.002
	Price per streaming update	0.00/0.50*
	Price per prediction	0.00/0.05*
	Costs for computing capacities per month	30.00
	Costs for streaming updates per month	7.75
	Costs for predictions per month	145.00
	Overall costs per month	182.75
Azure ML	• Number of users (subscription)	
	• Number of compute hours	
	• Number of transactions	

*First 10000 predictions/10001 + predictions

Amazon ML does not imply any subscription fees and further differentiates between batch and real-time predictions. For allowing real-time predictions, reserved memory capacity needs to be rented for storing the model. However, streaming updates to further train the model, as supported by the *Prediction API*, are not possible. *Azure ML* charges a monthly subscription fee as well as an hourly fee for using its data mining tools. For its ML APIs, an additional fee per transaction and per computing hour occurs. Due to the different pricing approaches and functionality, it is not possible to precisely compare the cloud services with each other. However, our examples indicate that the number of predictions is an essential cost factor while the cost for generating the model are comparatively low. This emphasizes the need for estimating the business value, i.e., impact of predictions (e.g., for justifying, guiding, and prescribing business actions [13]), which becomes increasingly important, for instance,

to calculate the return of investment (ROI) of cloud-based big data applications. A comprehensive cost and performance benchmark study (as shown in, e.g., [4, 7]) is necessary to further evaluate the different machine learning cloud services.

4 Discussion

In the previous sections, we have shown that the proposed generic cloud-based big data architecture can be implemented in the cloud environments of the three market leading cloud providers. In general, we see that the cloud pricing approaches are quite similar, but also strongly differ in some aspects. To strategically select a cloud provider, consumers have to specify and estimate the characteristics and resource demands of use cases. Given the identified cost factors, a total cost of ownership (TCO) approach will help to better estimate the overall costs of using big data cloud services, for instance, to compare it with an inhouse solution. Besides costs for operating the cloud environment, a TCO approach may need to consider additional costs including costs for installing and configuration, support, and backsourcing. In general, we have seen that each cloud provider has its strengths and weaknesses, for instance, in terms of costs and flexibility. Cost benefits achieved in one cloud service can be exhausted by another cloud service, as shown in the previous sections. All cloud providers support linear scalability in such a way that cost functions are linear. Common pricing schemes for on-demand cloud services are unit-based and tiered pricing, mainly based on the compute or storage capacity. Other pricing schemes, such as for reserved cloud services, have not been considered in this study. Regarding our cost comparisons, we see that the cloud market leader, *AWS*, generally performs well and may be the best choice for implementing a big data streaming application. To better estimate and evaluate total costs, however, the dynamics of big data applications in terms of resource requirements need to be taken into account, for instance, using simulations. For this purpose, application profiling might be necessary [1]. Moreover, benchmark studies are necessary to evaluate the running time of computing clusters for performing certain data processing and data analytics tasks in order to be able to better compare associated costs. The study furthermore emphasizes the need of measures for estimating the value of big data analytics, for instance, in terms of ROI or service quality.

5 Conclusions and Outlook

While the demand for big data applications is growing with the continuous increase of digital data, cloud computing has become essential for meeting infrastructure requirements for big data in terms of computational power and storage capacity. The rapid development of managed big data cloud services makes it possible to support certain activities of big data processing in an on-demand fashion. These cloud ser-

vices need to be integrated to implement sophisticated big data application environments. Likewise, it is important to estimate the costs and value of those environments in order to make strategic decisions.

In this paper, we have presented a generic reference architecture for implementing big data application in cloud environments based on best practices. Moreover, we have defined three specific implementations of this reference architecture, applying cloud services of three leading cloud providers, and analyze important pricing schemes and cost factors. In particular for big data streaming analytics, we identify differences and similarities as well as strengths and weaknesses between cloud providers. Taking a cost perspective, important implications can be derived from the applied case studies. The case studies indicate the importance of an integrated view on big data application environments for estimating and evaluating the overall costs. Therefore, the contribution of this chapter is twofold. First, we proposed a state-of-the-art reference architecture and explained important aspects for implementing big data applications in cloud environments. Second, we analyzed relevant cloud services from a cost perspective and derived important implications for big data management.

Given the insights and implications of this study, the development of a holistic TCO model for big data applications is the object for future research. Moreover, we aim to integrate this model into a simulation framework in order to provide a tool for decision support that is able to consider the dynamics of big data applications in terms of usage patterns and resource demands.

References

1. Assunção MD, Calheiros RN, Bianchi S, Netto MA, Buyya R (2015) Big data computing and clouds: trends and future directions. J Parallel Distrib Comput 79:3–15
2. AWS (2016) Big data analytics options on aws. https://d0.awsstatic.com/whitepapers/Big_Data_Analytics_Options_on_AWS.pdf
3. Chen M, Mao S, Liu Y (2014) Big data: a survey. Mob Netw Appl 19(2):171–209
4. Chen Y, Alspaugh S, Katz R (2012) Interactive analytical processing in big data systems: a cross-industry study of MapReduce workloads. Proc VLDB Endowment 5(12):1802–1813
5. Creeger M (2009) Cloud computing: an overview. ACM Queue 7(5):2
6. Gartner (2015) Magic quadrant for public cloud storage services, worldwide. http://www.gartner.com/technology/reprints.do?id=1-2IH2LGIŹct=150626Źst=sb
7. Ghazal A, Rabl T, Hu M, Raab F, Poess M, Crolotte A, Jacobsen HA (2013) BigBench: towards an industry standard benchmark for big data analytics. In: Proceedings of the ACM SIGMOD international conference on management of data. ACM, New York, NY, USA, pp 1197–1208
8. Heilig L, Lalla-Ruiz E, Voß S (2016) A cloud brokerage approach for solving the resource management problem in multi-cloud environments. Comput Ind Eng 95:16–26
9. Heilig L, Voß S (2014) Decision analytics for cloud computing: a classification and literature review. In: Newman A, Leung J (eds) Tutorials in operations research—bridging data and decisions. INFORMS, San Francisco, pp 1–26
10. Heilig L, Voß S (2014) A scientometric analysis of cloud computing literature. IEEE Trans Cloud Comput 2(3):266–278

11. Jensen M, Schwenk J, Gruschka N, Iacono LL (2009) On technical security issues in cloud computing. In: Proceedings of the IEEE international conference on cloud computing (CLOUD). IEEE, Bangalore, India, pp 109–116
12. Krishnan S, Tse E (2013) Hadoop platform as a service in the cloud. Technical report, Netflix. http://techblog.netflix.com/2013/01/hadoop-platform-as-service-in-cloud.html
13. LaValle S, Lesser E, Shockley R, Hopkins MS, Kruschwitz N (2011) Big data, analytics and the path from insights to value. MIT Sloan Manage Rev 52(2):21
14. Li M, Tan J, Wang Y, Zhang L, Salapura V (2015) SparkBench: a comprehensive benchmarking suite for in memory data analytic platform Spark. In: Proceedings of the 12th ACM international conference on computing frontiers (CF). ACM, Ischia, Italy, pp 53:1–53:8
15. Maravić I (2016) Spotify's event delivery—the road to the cloud (part III). https://labs.spotify.com/2016/03/10/spotifys-event-delivery-the-road-to-the-cloud-part-iii/
16. Shvachko K, Kuang H, Radia S, Chansler R (2010) The Hadoop distributed file system. In: Proceedings of the 26th IEEE symposium on mass storage systems and technologies (MSST). Incline Village, NV, USA, pp 1–10
17. Talia D (2013) Clouds for scalable big data analytics. IEEE Comput 46(5):98–101
18. Tallon PP (2013) Corporate governance of big data: perspectives on value, risk, and cost. Computer 46(6):32–38

http://www.springer.com/978-3-319-45497-9

Big Data Management
García Márquez, F.P.; Lev, B. (Eds.)
2017, XVI, 267 p. 107 illus., 38 illus. in color., Hardcover
ISBN: 978-3-319-45497-9