2013 IEEE International Congress on Big Data

Service-generated Big Data and Big Data-as-a-Service: An Overview

Zibin Zheng, Jieming Zhu, and Michael R. Lyu
Department of Computer Science & Engineering
The Chinese University of Hong Kong, Hong Kong, China
{zbzheng,jmzhu,lyu}@cse.cuhk.edu.hk

***Abstract*—With the prevalence of service computing and cloud computing, more and more services are emerging on the Internet, generating huge volume of data, such as trace logs, QoS information, service relationship, etc. The overwhelming service-generated data become too large and complex to be effectively processed by traditional approaches. How to store, manage, and create values from the service-oriented big data become an important research problem. On the other hand, with the increasingly large amount of data, a single infrastructure which provides common functionality for managing and analyzing different types of service-generated big data is urgently required. To address this challenge, this paper provides an overview of service-generated big data and Big Data-as-a-Service. First, three types of service-generated big data are exploited to enhance system performance. Then, Big Data-as-a-Service, including Big Data Infrastructure-as-a-Service, Big Data Platform-as-a-Service, and Big Data Analytics Software-as-a-Service, is employed to provide common big data related services (e.g., accessing service-generated big data and data analytics results) to users to enhance efficiency and reduce cost.**

Keywords*—*Big data, service computing, Big Data-as-a-Service

I. Introduction

Upon entering the 21st century, the global economic structure is transferring from “industrial economy” to “service economy”. According to the statistics of the World Bank, the output of modern service industry takes more than 60 percent of the world output, while the percentage in developed countries exceeds 70%. The competition in the area of modern service industry is becoming a focal point of the world’s economy development. Service computing, which provides flexible computing architectures to support modern service industry, has emerged as a promising research area. With the prevalence of cloud computing, more and more modern services are deployed in cloud infrastructures to provide rich functionalities. The number of services and service users are increasing rapidly. There has been enormous explosion in data generation by these services with the prevalence of mobile devices, user social networks, and large-scale service-oriented systems. The overwhelming service-generated data become too large and complex to be effectively processed by traditional approaches.

In information technology, big data has emerged as a widely recognized trend, attracting attentions from government, industry and academia. Big data are high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization [1]. Mentioned by the Compliance, Governance and Oversight Council (CGOC, an organization focused on Information Governance), information volume doubles every 18-24 months for most organizations and 90% of the data in the world has been created in the last two years [2]. In March 2012, the Obama administration announced the big data research and development initiative, which explored how big data could be used to address important problems facing the government. The leading IT companies, such as SAG, Oracle, IBM, Microsoft, SAP and HP, have spent more than $15 billion on buying software firms specializing in data management and analytics. This industry on its own is worth more than $100 billion and growing at almost 10% a year, which is roughly twice as fast as the software business as a whole [3]. How to efficiently and effectively create values from the big data become an important research problem.

The emerging large-scale service-oriented systems often involve a large number of services with complex structures. The big data generated from these systems are typically heterogeneous, of multiple data types, and highly dynamic. Due to the fast increase of system size and the associated massive volume of service-generated data, creating value in the presence of massive system and data becomes an inevitable challenge. Examples of service-generated big data include trace logs, Quality-of-Service (QoS) information, service invocation relationship, etc. Similar to other types of big data, the service-generated big data initiatives span four unique dimensions [4]: (1) volume: nowadays’ large-scale systems are awash with ever-growing data, easily amassing terabytes or even petabytes of information; (2) velocity: time-sensitive processes, such as bottleneck detection and service QoS prediction, could be achieved as data stream into the system; (3) variety: structured and unstructured data are generated in various data types, making it possible to explore new insights when analyzing these data together; and (4) veracity: detecting and correcting noisy and inconsistent data are important to conduct trustable analysis. Establishing trust in big data presents a huge challenge as the variety and number of sources grows. These four unique characteristics of service-generated big data provide great challenge for data management and analysis.

To fulfill the potential of service-generated big data, developing exceptional technologies to effectively process large quantities of data within acceptable processing time is a critical task. Moreover, easy access of the big data and the big data analysis results are important. Big Data-as-a-Service encapsulates various big data storage, management, and analytics

978-0-7695-5006-0/13 $26.00 © 2013 IEEE
DOI 10.1109/BigData.Congress.2013.60

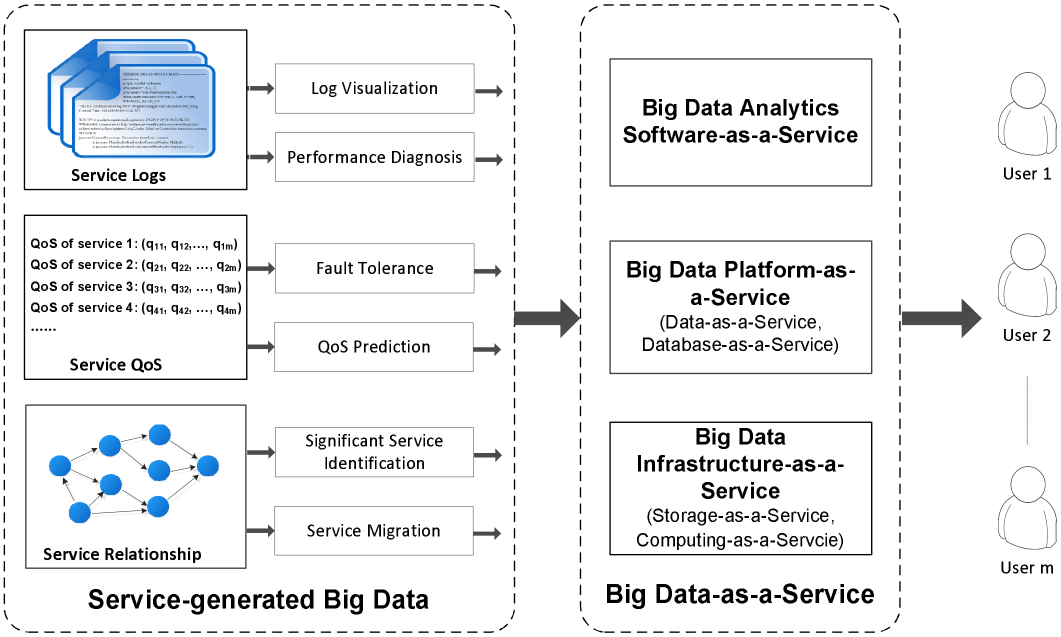
Service Logs
Log Visualization
Performance Diagnosis
QoS of service 1 : (q_{11}, q_{12},..., q_{1m})
QoS of service 2 : (q_{21}, q_{22}, ..., q_{2m})
QoS of service 3 : (q_{31}, q_{32}, ..., q_{3m})
QoS of service 4 : (q_{41}, q_{42}, ..., q_{4m})
......
Service QoS
Fault Tolerance
QoS Prediction
Service Relationship
Significant Service Identification
Service Migration
Service-generated Big Data
Big Data Analytics Software-as-a-Service
Big Data Platform-as-a-Service
(Data-as-a-Service, Database-as-a-Service)
Big Data Infrastructure-as-a-Service
(Storage-as-a-Service, Computing-as-a-Servcie)
Big Data-as-a-Service
User 1
User 2
User m

Fig. 1. Overview of Service-generated Big Data and Big Data-as-a-Service

techniques into services and provides common big data related services to users via programmable APIs, which greatly enhances efficiency, reduce cost and enables seamless integration. To provide big data infrastructure, big data platform, and big data analytics softwares as services, there are a lot of research investigations need to be done. This paper provides an overview of service-generated big data and Big Data-as-a-Service. First, three types of service-generated big data (service trace logs, service QoS information, and service relationship) are exploited to enhance system performance. Then, Big Data-as-a-Service (BDaaS) is investigated to provide friendly APIs to users to access the service-generated big data and data analysis results.

The rest of this paper is organized as follows: Section 2 provides an overview of this paper; Section 3 exploits service-generated big data; Section 4 investigates Big Data-as-a-Service; Section 5 explores business aspects of service-generated big data and Big Data-as-a-Service and Section 6 concludes the paper.

II. Overview

Fig. 1 provides an overview of the service-generated big data and Big Data-as-a-Service. As shown in the figure, on the one hand, we will introduce some typical applications which exploit three types of service-generated big data respectively for system performance enhancement. First, log visualization and performance program diagnosis are investigated via mining service request trace logs. Second, QoS-aware fault tolerance and service QoS prediction are studied based on the service QoS information. Finally, significant service identification and service migration are achieved by investigating service relationship. These concrete applications will shed some light on the problem of big data analytics by mining the service-generated big data.

On the other hand, to provide user-friendly access to the service-generated big data and various big data analytic results, Big Data-as-a-Service will also be introduced as a basic framework to store, manage and create value from the big data. Fig. 1 illustrates the framework of Big Data-as-a-Service, which involves three layers, i.e., Big Data Infrastructure-as-a-Service, Big Data Platform-as-a-Service, and Big Data Analytics Software-as-a-Service. Via standard and programmable APIs, Big Data-as-a-Service enables dynamic integration of different big data and integration of different big data analytics approaches to create value from the service-generated big data.

III. Service-generated Big Data

Nowadays, there are all kinds of online services provided on the Internet and daily used by millions of users. Every time when you perform a search, send an Email, post a microblog or shop on e-commerce Websites, you are generating a trace of data to the services.

As shown in Figure 2, as the number of services and users scales up, the service-generated data (including service logs, service QoS information, and service relationship) are increasing, leading to the big data phenomenon. With the increasing volume of service-generated big data, how to create values from the data becomes an important research problem. The following sub-sections will describe in detail how the service generated data can be processed and analyzed to enhance system performance.

A. Service Trace logs

With the popularization of large-scale service-oriented systems, and the number increasing of service users (e.g., PCs, mobile devices, etc.), a huge volume of trace logs are generated by the service-oriented systems each day. There are billions of

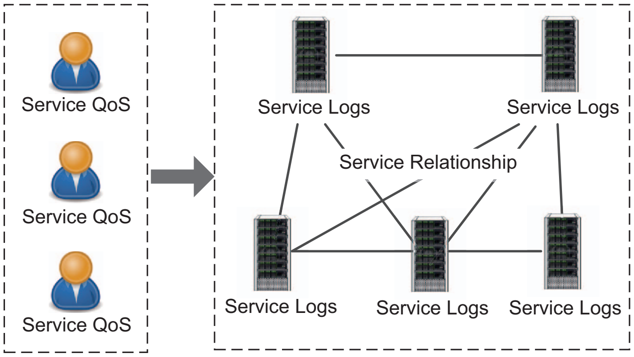

Fig. 2. Service-generated Big Data

daily logs, log files, and structured/unstructured data from a wide variety of service systems. For example, an Email service provided by Alibaba (one of the biggest e-commerce company in the world) would produce about 30-50 gigabytes (around 120-200 million lines) of tracing logs per hour [5]. These logs can be utilized both in the development stages and in normal operations for understanding and debugging the behavior of the complex system.

As the scale and complexity of distributed systems rapidly increase, large-scale distributed systems like cloud computing systems typically involve a large number of interactions between service components, in some cases across hundreds of machines, which makes it very difficult to manually diagnose the performance problems. By the way of mining these service-generated trace logs, valuable information can be obtained to help service designers and developers understand and improve the quality of systems. For example, performance bottleneck localization can be achieved by analyzing the trace logs. However, there are still many challenges to be addressed. On one hand, the huge volume of service-generated trace logs make performance diagnosis labor-intensive; on the other hand, the demand for real-time system diagnosis is ever increasing.

This section discusses how to investigate the trace logs to find the value hidden in it, including trace log visualization and performance problem diagnosis

1) ***Trace log visualization****:* Distributed systems are continuously growing in scale and service component interactions. It is becoming difficult for the system designers and administrators to understand the characteristics of system performance. Request tracing approaches, which are widely adopted by companies such as Google [6], Microsoft [7] and eBay [8], record the execution information of requests (e.g., entering and leaving time when individual requests go through service components). These trace logs contain a lot of hidden treasures for system designers and administrators. Log visualization provides tools for abstract visualization of log files (or results of log queries) in a way that can be best understood by users [9]. With the increasing volume of trace logs, efficient log visualization of become an challenge research problem.

To address this problem, a number of approaches have been proposed. Stardust [10] uses relational databases as repository which suffers poor query efficiency in the environment of massive data volumes. P-tracer [11] is an online performance profiling tool to visualizes multi-dimensional statistical information to help administrators understand the system performance behaviours in depth. DTrace [12] and gprof [13] visualize the execution of systems as call graphs to signify where requests spend time.

Although a number of previous research investigations have been conducted at service log visualization, this research problem becomes more challenging in the scenario of big data, caused by the rapid increase of log files, the unstructured log data, and the requirement of real-time query and display. More research investigations are needed to enable real-time processing and visualization of the huge volume of trace logs.

2) ***Performance problem diagnosis****:* In today's distributed systems, especially the cloud systems, a service request will go through different hosts, invoking a number of software modules. When the service cannot satisfy the promised service level agreement (SLA) to users, it is critical to identify which module (e.g., an invoked method) is the root cause of the performance problem in a timely manner. Trace logs provide valuable information to find the cause of performance problems. How to exploit the tremendous trace logs effectively and efficiently to help designer understand system performance and locate performance behaviours becomes an urgent and challenging research problem.

In recent literature, a large body of research work has been investigated to address this problem. For example, Magpie [14], Pip [15] and Ironmodel [16] are application-specific approaches, which require domain knowledge to construct the performance diagnosis models. Pinpoint [17] employs clustering algorithm to group failure and success logs. Dapper [6] employs Bigtable to manage the large volume of trace logs. Data mining technologies such as principal component analysis (PCA) and robust principal component analysis are also employed for identifying performance bottlenecks via modeling and mining the trace logs [5], [18].

However, the large volume and high velocity of service-generated trace logs make it very difficult to perform real-time diagnosis. Most of the previous solutions suffer from low efficiency in handling large volume of data. More efficient storage, management, and analysis approaches for service-generated trace logs are required.

B. Service QoS information

The current large-scale distributed platforms (e.g., various cloud platforms) provide a number of services to heterogeneous and diversified users. Large volume of QoS data of these services are recorded, in both server-side and user-side. Since different users may observe quite different QoS performance (e.g., response-time) on the same service, the volume of user-side QoS data is much larger than that of server-side QoS data. Moreover, QoS values of service components are changing dynamically from time to time, resulting in explosive increase of user-side service QoS information.

Valuable information can be obtained through investigating these user-side service QoS information in order to enhance system performance, for example, to achieve adaptive fault tolerance and to make personalized QoS prediction for users.

*1) **Fault tolerance**:* The service computing environment is highly dynamic and heterogeneous, where original services may be disabled, new services may be added, and QoS of the services may change from time to time. Building reliable service-oriented systems is much more challenging in this highly dynamic environment compared with the traditional stand-alone software systems. Software fault tolerance [19] is an important approach to build reliable systems via employing functionally-equivalent components to tolerate faults. On the Internet, the functionally-equivalent Web services provided by different organizations can be employed to build fault-tolerant service-oriented systems. When designing fault tolerance strategies, service QoS information can be considered to enhance the performance.

The huge number of services in the large-scale distributed platforms is monitored continuously at runtime. Large volume of QoS data (e.g., response-time, availability, throughput, etc.), is recorded. Moreover, after conducting service invocation, the users also record the QoS of the invoked services. How to efficiently and effectively process these large volume of service QoS data to design fault tolerance strategies which can adapt to the dynamic environment for optimal performance is a challenging research problem.

In our previous work [20], a preliminary middleware has been designed for fault-tolerant Web services. However, this middleware did not provide a personalized fault tolerance strategy for different users. In the dynamic Internet environment, server-side fault tolerance is not enough since the communication connections can fail easily. Personalized user-side fault tolerance needs to be considered. Moreover, to speedup the analysis and computation of the large volume of service QoS information, online learning algorithms [21] will need to be investigated for incremental update of the fault tolerance strategy when new QoS values become available.

*2) **QoS prediction**:* Web service QoS prediction aims at providing personalized QoS value prediction for service users, by employing the historical QoS values of different users. Web service QoS prediction usually includes a user-service matrix, where each entry in the matrix represents the value of a certain QoS property (e.g., *response-time*) of a Web service observed by a service user. The user-service matrix is usually very sparse with many missing entries, since a service user typically only invoked a small number of Web services in the past. The problem is how to accurately predict the missing QoS values in the user-service matrix by employing the available QoS values. After predicting the missing Web service QoS values in the user-service matrix, each service user can have a QoS evaluation on all the services, even on the unused services. As a result, optimal service can be selected for users to achieve good performance.

In service computing, Web service QoS prediction has attracted a lot of attention in recent years. A number of QoS prediction approaches have been proposed to address this paper, including user-based QoS prediction approach [22], combination of user-based and item-based approaches [23], ranking-oriented approach [24], clustering-based approach [25], etc. Some recent work [26], [27] further considers the locations of Web services and service users to enrich the context information of the service environment and thus improve the prediction quality. However, with the rapid increasing of Web services and users, the size of user-service matrix is becoming larger and larger, which makes it not efficient for real-time prediction. Since the Internet environment is highly dynamic, services may add or drop at anytime, so enhancing the robustness of QoS prediction approaches (e.g., solving the cold-start problem) is very essential. In addition, values of some user-side QoS properties (e.g., response time) are varying over time, making the matrix become a three-dimensional user-service-time matrix. Thus, how to efficiently process the large volume of available service QoS data and accurately predict the missing QoS values in the huge user-service-time matrix becomes a very challenging research problem.

C. Service relationship

Nowadays, large-scale distributed systems typically involve a large number of service components. These service components are typically deployed in distributed computer nodes (i.e., physical machines or virtual machines) and have complex invocation relationships. For example, to generate the dynamic Web content for a page in one of the e-commerce sites in Amazon, each request typically requires the page rendering components to construct its response by sending requests to over 150 services [28]. These components are interdependent between each other. The invocation relationship among service components can be modeled as a weighted directed graph, where a node in the graph represents a service component and a directed edge from one node to another represents a component invocation relationship. The weight at each edge may be expressed as the cost or the frequency of the invocation. This service invocation graph can be updated dynamically at runtime, caused by reasons such as service add/drop, service migration, load balance, etc.

By exploiting the service invocation graph, valuable information can be obtained to identify significant service components and to enable better dynamic service migration.

*1) **Significant service component identification**:* Reliability of different service components may impose different impacts on the service-oriented system. By investigating the service invocation graph, the significant service components (e.g., core components or weak components) can be identified. This can greatly help us understand how to improve the structure of a system and how to improve the reliability of the system. For example, additional fault tolerance strategies can be designed for these significant service components to achieve higher reliability. However, due to the nature of dynamic composition of service components, the service invocation graph can be continuously updated at runtime. And the trend towards large-scale systems make the service invocation graph quite large and complex. Stochastic ranking techniques can be employed to identify the significant service component in the graph for a distributed system.

In our preliminary investigation [24], we considered a component significant if it is invoked by many other important components frequently, and proposed a random walk based approach to identify significant components in a cloud system. Besides, Liu et al. [29] propose to exploit the service relationship to assist the reputation computation of services, which further improve the robustness of the system. However, there are still a lot of research problems to be addressed, for example: (1) modeling the relationship between different service components (e.g., components a and b invoke the same component c, then there is implicit relationship between component a and b); (2) modeling the impact of component on the whole system, which can help us improve the robustness of whole system; (3) designing more efficient and effective approaches to build and analyze the service invocation graph and identify significant service components.

2) ***Service migration:*** Distributed systems typically include a number of service components, which need to be deployed to distributed nodes (i.e., physical machines or virtual machines). Since the service environment is highly dynamic, after the initial deployment, it is essential to optimize the service deployment strategy among candidate nodes periodically to achieve optimal overall system performance while minimizing the operational cost. As a result, dynamic service migration is in need by moving the service from one physical machine to another at runtime. It is a common practice in many commercial cloud platforms.

By modeling and exploiting the service invocation relationship and past service usage experiences, a proper migration of the services can improve the experience for existing users. In our previous work [30], a preliminary model has been formulated based on integer programming to make an optimal redeployment of services. However, this integer programming based model suffers from the scaling problem when facing a large number of services and candidate nodes. This model is further improved in [31] by taking service relationship into account, which proposes to use a genetic algorithm to solve the model efficiently. With the proliferation of cloud federation, some other work (e.g., [32], [33], [34]) also considers the dynamic service deployment and migration in geographically distributed clouds to get optimal the service performance.

To cope with the growing size of the service migration problem, more efficient approaches are needed. For example, (1) in addition to considering the invocation relationship among services, locations of service users can be considered to further improve the migration performance; (2) employing QoS prediction techniques, network latencies among service components and between users and services can be predicted to achieve better service migration performance; (3) to speedup the computation of optimal service migration strategy, instead of integer programming, various machine learning algorithms, such as online learning, and k-median optimization model, genetic algorithm, etc., can be investigated to enable efficient service migration.

IV. Big Data-as-a-Service

The value of data has been widely recognized. Data can be analyzed for a lot of purposes, such as enhancing system performance, guiding decision making, assessing risk, trimming costs, lifting sales, and so on [35]. Traditionally, such kinds of data analysis tasks are conducted separately by different organizations, although these tasks include a lot of common steps, such as information extraction, data cleaning, modeling, visualization, and so on. With the increasingly large amount of data, building separate systems to analyze data becomes expensive and infeasible, caused by not only the cost and time of building the systems, but also the required professional knowledge on big data management and analysis. Therefore, it is necessary to have a single infrastructure which provides common functionality of big data management, and flexible enough to handle different types of big data and big data analysis tasks [36].

Big Data-as-a-Service provides common big data related services to users to enhance efficiency and reduce cost. As shown in Figure 1, it typically includes three layers, i.e., big data infrastructure, big data platform, and big data analytics. These three layers in Big Data-as-a-Service provide different level of abstractions to users, where Big Data Infrastructure provides the most basic services and the higher layers provide more advanced services. Although cloud is a nature architecture for Big Data-as-a-Service, the service is not limited to just cloud architecture. Other distributed architecture can also be employed to host the big data services. Section IV-A to Section IV-C provide details discussions on these three layers of Big Data-as-a-Service, respectively.

A. Big Data Infrastructure-as-a-Service

Scale-out infrastructure provides necessary computing and storage capacity for big data. Big data infrastructure can leverage Infrastructure-as-a-Service (IaaS) in cloud computing, including Storage-as-a-Service and Computing-as-a-Service, to store and process the massive data. Although current technologies such as cloud computing provide infrastructure for automation of data collection, storing, processing and visualization, big data impose significant challenges to the traditional infrastructure, due to the characteristics of volume, velocity and variety. Modern Internet and scientific research project produce a huge amount of data with complex interrelationship. These big data need to be supported by a new type of Infrastructure tailored for big data, which must have the performance to provide fast data access and process to satisfy users' just in time needs [37]. Moreover, community standards for data description and exchange are also crucial [38].

One of the challenges of designing big data infrastructure is the requirement to support many different data types, not only the existing data types but also the new types that are emerging. Moreover, the big data infrastructure needs to support reuse and share of the big data. For example, allowing researchers to link their scientific results with the initial data and intermediate data to allow future re-use/re-purpose of the data [39]. Flexible access control of the big data is thus required.

Different from traditional IaaS in cloud computing, in big data infrastructure, the technologies for processing big data have to combine with storage designs [40]. For example, the service-generated big data are generated by Internet services, which are typically deployed in cloud infrastructures. Due to the huge volume of data, it is inevitable that both data and data analytics approaches need to be closely located to reduce the unnecessary network traffic, remove performance bottleneck, and enhance efficiency. Therefore, it is common to develop and deploy big data analytics approaches at the same cloud infrastructure to provide enhanced service offerings.

B. Big Data Platform-as-a-Service

A big data platform allows users to access, analyze and build analytic applications on top of large data sets [41].

One example of Big Data Platform-as-a-Service is Google's BigQuery[1], which let users take advantage of Google's massive compute and storage power to analyze Big Data and get real-time business insights in seconds via REST interface. BigML[2] is another big data PaaS example, which offers a highly scalable cloud based machine learning service that is easy to use, seamless to integrate and instantly actionable. Big data platforms typically include multiple modules, such as analysis task specification, data storage and management, data process and integration, discovery and visualization, and so on.

In a big data platform, the big data analysis typically includes multiple steps. For certain steps, APIs provided by the big data platform can be employed to conduct common processing on the data. For other steps, the users need to define their own specified processing and analysis rules, which requires the big data platform to be flexible enough in expressing various kinds of big data analysis tasks. Appropriate high level declarative language is required to specify different user tasks. The challenges of designing such kind of language include not only the seamless integration of different steps (such as platform-provided APIs and user-defined analysis tasks), but also the consideration of data locations to enable efficient data management, aggregation, and analysis.

At the big data platform, there are different ways for data storage and management, including cloud storage, Data-as-a-Service (DaaS), and Database-as-a-Service (DBaaS). As introduced by Razi Sharir [42], the key differences between these different ways can be summarized as follows:

- Cloud Storage (e.g., Amazon S3, Dropbox, etc.) enables users to store data in the virtual storage of the cloud as they would on any other storage device.
- DaaS (Data-as-a-Service) describes the ability to define data lists in a cloud service and allow controlled access to the data through Web API (e.g., RESTful Web service). Unlike database solutions, DaaS cannot be accessed via languages such as SQL. DaaS is suitable only for basic data management querying and manipulation. One example of DaaS is Google's public data service that provides access to all sorts of data provided by public institutions[3].
- DBaaS (Database-as-a-Service) offers full-blown database service, which can be accessed via predefined common sets of APIs. The provided database services can be traditional relational databases, NoSQL data stores, in-memory databases, and so on.

Nowadays, users are accessing multiple data storage platforms to accomplish their operational and analytical requirements [43]. Efficient integration of different data sources is important. For example, an organization may purchase storage from different vendors and need to combine data with different format stored on systems from different vendors [37]. Data integration, which plays an important role for both commercial and scientific domains, combines data from different sources and provides users with a unified view of these data [44]. How to make efficient data integration with the 4V (volume, velocity, variety, and veracity) characteristics is a key research direction for the big data platforms.

TABLE I. EXAMPLES OF BIG DATA ANALYTICS TECHNIQUES USAGE

Analytic Techniques	Usages
Performance problem diagnosis	Identify the cause system performance problems
Fault tolerance	Improve the reliability of systems
QoS prediction	Enhance quality of the service-oriented systems
Marketing and sales	Identify potential customers, enhance company profit
Manufacturing process analysis	Identify the causes of manufacturing problems
Insurance	Fraudulent claim detection, risk assessment
Item recommendation	Model user preferences from data employing collaborative filtering, etc.
User behavior modeling	Learn user characteristics from data

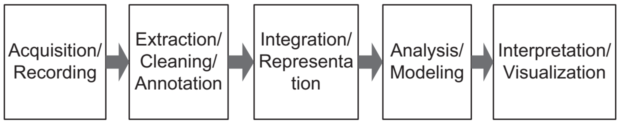
Acquisition/ Recording
Extraction/ Cleaning/ Annotation
Integration/ Representa tion
Analysis/ Modeling
Interpretation/ Visualization

Fig. 3. The Big Data Analysis Pipeline

C. Big Data Analytics Software-as-a-Service

Although big data infrastructure and platform are critical, they can't create the same long-term value as various big data analytics softwares which apply big data to accelerate a market [45]. Big data analytics is the process of examining large amounts of data of various types to uncover hidden patterns, unknown correlations and other useful information [46]. The big data analytics algorithms are complex and far beyond the reach of most organization's IT capabilities. Moreover, there are too few skilled big data practitioners available for every organizations. Therefore, more and more organizations turn to Big Data Analytics Software-as-a-Service to obtain the business intelligence (BI) service that turns their unstructured data into an enhanced asset [47]. Table I provides some examples of the usage of big data analytics techniques. As shown in the table, by exploiting the big data, great values can be created in various areas.

Big Data Analytics Software-as-a-Service exploits massive amounts of structured and unstructured data to deliver realtime and intelligent results, allowing users to perform self-service provisioning, analysis, and collaboration. Big Data Analytics Software-as-a-Service is typically Web-hosted, multi-tenant and use Hadoop, noSQL, and a range of pattern discovery and machine learning technologies [48]. Users would typically execute scripts and queries that data scientists and programmers developed for them to generate reports and visualizations [40]. Various big data analytics approaches can be implemented and encapsulated into services. By this way, users will be able to interact with Web-based analytics services easily without worrying about the underlying data storage, management, and analyzing procedures.

As shown in Fig. 3, the analysis of big data typically involves multiple distinct phases [36]. First, big data are sampled and recorded from some data generation sources (e.g., from large-scale complex service computing systems). Second, since the collected information may not be in a format ready for analysis, we need to extract the required information from the underlying sources, and detect and correct the inaccurate

[1] https://cloud.google.com/products/big-query

[2] https://bigml.com/

[3] http://www.google.com/publicdata

records. Third, given the heterogeneity of the data, data integration and representation are needed. After the above phases, data analysis and modeling can be conducted on the resulting integrated and cleaned big data. Finally, data interpretation and visualization are needed since big data analytics alone is of limited value if users cannot understand the analysis results.

To enable Big data Analytics Software-as-a-Service, the opensource Apache Hadoop software framework is widely employed by leading companies (e.g., Yahoo!, Amazon.com, Apple, eBay, IBM, Facebook, LinkedIn, Microsoft, SAP, etc.). For example, Facebook uses it for storage and for their Facebook Messages service [47]. To store data safely, Hadoop distributes data redundantly on a number of computer servers. To enhance the process efficiency, Hadoop breaks down a task into smaller subtasks, executes the subtasks simultaneously on different computers, and finally reassembles the results. When big data analytics software is offered as a service, improving business practices is easy and cost-effective. The advantages of Big Data Analytics Software-as-a-Service include: Faster deployment, powerful computing and storage capacity, less management, and most importantly, less cost (businesses only pay for the consumed services rather than having underutilized equipment, bandwidth and manpower) [49].

V. Business Aspects

With the development of technology, a flood of data is created every day by the interactions of billions of people using computers, GPS devices, cell phones, and medical devices [50]. For example, nowadays, there are 4.6 billion mobile phone subscriptions worldwide, 1 billion-2 billion people use the Internet [3], and more than 30 million networked sensor nodes are now present in the transportation, automotive, industrial, utilities, and retail sectors [51]. The amount of digital information increases tenfold every five years [3].

According to a recent McKinsey report [51], in 2010, enterprises and users stored more than 13 exabytes of new data. The potential value of global personal location data is estimated to be $700 billion to end users, and it can result in an up to 50% decrease in product development and assembly costs. McKinsey also predicts an equally great effect of Big Data in employment, where 140,000-190,000 workers with ”deep analytical” experience will be needed in the US; furthermore, 1.5 million managers will need to become data-literate [36]. EMC estimates that about 15 percent of all IT spending will move to the as-a-Service delivery models by 2015 and will grow to about 35 percent by 2021. The Big Data-as-a-Service market size is about 2.25 billion by 2015 and about 30 billion by 2021. In other words, about 4 percent of all IT spending will go into Big Data-as-a-Service by 2012 [40].

There is strong business motivation of Big Data-as-a-Service. Based on the ownership of big data, business models of Big Data-as-a-Service can be divided into the following two types: (1)The owner of big data conducts data storage, management, and analysis and provide Web APIs for users to access the data or the analyzed results. For example, Google crawls a huge volume of data, conduct the data management and analysis, and provides various types of APIs with different functionality (e.g., APIs for Webpage search, APIs for map services, etc.) to users. (2) Due to reasons such as cost reduction or lack of professional knowledge, the owner of big data outsources the big data processing (or part of it) to a third party. It consumes the Big Data-as-a-Service provided by third party and allows the service provider to work on it to extract values. The service providers typically charge users on a utility computing basis, i.e., the cost reflects the amount of resources allocated and consumed.

Big Data-as-a-Service provides different levels of services, including infrastructure level, platform level, and software level. These services can be easily used and integrated into other systems. By encapsulating the complex details, Big Data-as-a-Service has the characteristics of cross-language, cross platform, and cross-firewall, making dynamic big data service composition possible, offering great opportunities to create new business values.

VI. Conclusion and Future Work

This paper provides an overview of service-generated big data and Big Data-as-a-Service. Three types of service-generated big data are exploited to enhance quality of service-oriented systems. To provide common functionality of big data management and analysis, Big Data-as-a-Service is investigated to provide APIs for users to access the service-generated big data and the big data analytics results.

In the future, beside the service trace logs, QoS information and service relationship, more types of service-generated big data will be investigated. More comprehensive studies of various service-generated big data analytics approaches will be conducted. Detailed technology roadmap will be provided and security issues beyond the scope of this paper will also be investigated.

Acknowledgment

The work described in this paper was fully supported by the National Basic Research Program of China (973 Project No. 2011CB302603), the National Natural Science Foundation of China (Project No. 61100078), the Shenzhen Basic Research Program (Project No. JCYJ20120619153834216, JC201104220300A), and the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 415212).

References

[1] M. A. Beyer and D. Laney, “The importance of ’big data’: A definition,” Gartner, Tech. Rep., 2012.

[2] D. Austin, “eDiscovery Trends: CGOCs Information Lifecycle Governance Leader Reference Guide,” ”http://www.ediscoverydaily.com”, May 2012.

[3] The Economist, “A special report on managing information: Data, data everywhere,” *The Economist*, February 2010.

[4] IBM, “What is big data? ł bringing big data to the enterprise,” ”http://www-01.ibm.com/software/data/bigdata”, 2013.

[5] H. Mi, H. Wang, Y. Zhou, M. R. Lyu, and H. Cai, “Towards fine-grained, unsupervised, scalable performance diagnosis for production cloud computing systems,” *IEEE Transactions on Parallel and Distributed Systems*, no. PrePrints, 2013.

[6] B. H. Sigelman, L. A. Barroso, M. Burrows, P. Stephenson, M. Plakal, D. Beaver, S. Jaspan, and C. Shanbhag, “Dapper, a large-scale distributed systems tracing infrastructure,” Google, Inc., Tech. Rep., 2010.

[7] S. Han, Y. Dang, S. Ge, D. Zhang, and T. Xie, "Performance debugging in the large via mining millions of stack traces," in *Proc. 34th Int'l Conf. on Software Engineering (ICSE'12)*, 2012, pp. 145–155.

[8] M. Y. Chen, A. Accardi, E. Kiciman, J. Lloyd, D. Patterson, A. Fox, and E. Brewer, "Path-based faliure and evolution management," in *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation*. USENIX Association, 2004, pp. 23–23.

[9] C. Lim, N. Singh, and S. Yajnik, "A log mining approach to failure analysis of enterprise telephony systems," in *Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN'08)*, 2008, pp. 398–403.

[10] E. Thereska, B. Salmon, J. Strunk, M. Wachs, M. Abd-El-Malek, J. Lopez, and G. R. Ganger, "Stardust: tracking activity in a distributed storage system," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 34, no. 1. ACM, 2006, pp. 3–14.

[11] H. Mi, H. Wang, H. Cai, Y. Zhou, M. R. Lyu, and Z. Chen, "P-tracer: Path-based performance profiling in cloud computing systems," in *Proceedings of the 36th IEEE Annual Computer Software and Applications Conference (COMPSAC'12)*. IEEE, 2012, pp. 509–514.

[12] B. M. Cantrill, M. W. Shapiro, A. H. Leventhal *et al.*, "Dynamic instrumentation of production systems," in *USENIX Annual Technical Conference*, 2004, pp. 15–28.

[13] S. L. Graham, P. B. Kessler, and M. K. Mckusick, "Gprof: A call graph execution profiler," *ACM Sigplan Notices*, vol. 17, no. 6, pp. 120–126, 1982.

[14] P. Barham, A. Donnelly, R. Isaacs, and R. Mortier, "Using magpie for request extraction and workload modelling," in *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation (OSDI'04)*, 2004, pp. 18–18.

[15] P. Reynolds, C. Killian, J. L. Wiener, J. C. Mogul, M. A. Shah, and A. Vahdat, "Pip: detecting the unexpected in distributed systems," in *Proceedings of the 3rd conference on Networked Systems Design & Implementation (NSDI'06)*, 2006, pp. 9–9.

[16] E. Thereska and G. R. Ganger, "Ironmodel: robust performance models in the wild," in *Proceedings of the International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '08)*, 2008, pp. 253–264.

[17] M. Chen, E. Kiciman, E. Fratkin, A. Fox, and E. Brewer, "Pinpoint: problem determination in large, dynamic internet services," in *Proceedings of the International Conference on Dependable Systems and Networks (DSN'02)*, pp. 595–604.

[18] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," in *Proceedings of the ACM 22nd Simposium on Operating Systems Principles (SOSP'09)*, 2009, pp. 117–132.

[19] M. R. Lyu, *Software Fault Tolerance*. Trends in Software, Wiley, 1995.

[20] Z. Zheng and M. R. Lyu, "A QoS-aware fault tolerant middleware for dependable service composition," in *Proc. 39th Int'l Conf. Dependable Systems and Networks (DSN'09)*, 2009, pp. 239–248.

[21] H. Yang, Z. Xu, I. King, and M. Lyu, "Online learning for group lasso," in *International Conference on Machine Learning (ICML'10)*, 2010.

[22] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for Web services via collaborative filtering," in *Proc. 5th Int'l Conf. Web Services (ICWS'07)*, 2007, pp. 439–446.

[23] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware Web service recommendation by collaborative filtering," *IEEE Transactions on Service Computing*, vol. 4, no. 2, pp. 140–152, 2011.

[24] Z. Zheng, Y. Zhang, and M. R. Lyu, "CloudRank: A QoS-driven component ranking framework for cloud computing," in *Proc. Int'l Symp. Reliable Distributed Systems (SRDS'10)*, 2010, pp. 184–193.

[25] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-aware Web service recommendation and visualization," *IEEE Transactions on Services Computing*, no. PrePrints, 2011.

[26] M. Tang, Y. Jiang, J. Liu, and X. F. Liu, "Location-aware collaborative filtering for qos-based service recommendation," in *Pro. IEEE 19th Int'l Conf' on Web Services (ICWS'12)*, 2012, pp. 202–209.

[27] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "Collaborative web service qos prediction with location-based regularization," in *Pro. IEEE 19th Int'l Conf' on Web Services (ICWS'12)*, 2012, pp. 464–471.

[28] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: amazon's highly available key-value store," in *Proc. 21st ACM Symposium on Operating Systems Principles (SOSP'07)*, 2007, pp. 205–220.

[29] A. Liu, Q. Li, L. Huang, and S. Wen, "Shapley value based impression propagation for reputation management in web service composition," in *Pro. IEEE 19th Int'l Conf' on Web Services (ICWS'12)*, 2012, pp. 58–65.

[30] Y. Kang, Z. Zheng, and M. Lyu, "A latency-aware co-deployment mechanism for cloud-based services," in *Proceedings of the IEEE 5th International Conference on Cloud Computing (CLOUD'12)*, 2012, pp. 630–637.

[31] J. Zhu, Z. Zheng, Y. Zhou, and M. R. Lyu, "Scaling service-oriented applications into geo-distributed clouds," in *Pro. IEEE Int'l Workshop on Internet-based Virtual Computing Environment (iVCE'13)*, 2013.

[32] Q. Zhang, Q. Zhu, M. F. Zhani, and R. Boutaba, "Dynamic service placement in geographically distributed clouds," in *Proc. IEEE 32nd Int'l Conf. on Distributed Computing Systems (ICDCS'12)*, 2012, pp. 526–535.

[33] M. Steiner, B. G. Gaglianello, V. K. Gurbani, V. Hilt, W. D. Roome, M. Scharf, and T. Voith, "Network-aware service placement in a distributed cloud environment," in *Proc. ACM SIGCOMM'12*, 2012, pp. 73–74.

[34] M. Alicherry and T. V. Lakshman, "Network aware resource allocation in distributed clouds," in *Proc. IEEE INFOCOM'12*, 2012, pp. 963–971.

[35] S. Lohr, "The age of big data," *New York Times*, vol. 11, 2012.

[36] "Challenges and opportunities with big data," leading researchers across the United States, Tech. Rep., 2011.

[37] E. Slack, "Storage infrastructures for big data workflows," Storage Switchland, LLC, Tech. Rep., 2012.

[38] C. Lynch, "Big data: How do your data grow?" *Nature*, vol. 455, no. 7209, pp. 28–29, 2008.

[39] Y. Demchenko, "Bof: Infrastructure issues in big data," "https://tnc2013.terena.org/core/event/15", 2013.

[40] "Big data-as-a-service: A market and technology perspective," EMC Solution Group, Tech. Rep., 2012.

[41] J. Horey, E. Begoli, R. Gunasekaran, S.-H. Lim, and J. Nutaro, "Big data platforms as a service: challenges and approach," in *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing*, ser. HotCloud'12, 2012, pp. 16–16.

[42] R. Sharir, "Cloud database service: The difference between dbaas, daas and cloud storage - what's the difference?" "http://xeround.com/blog/2011/02/dbaas-vs-daas-vs-cloud-storage-difference", 2011.

[43] B. Devlin, S. Rogers, and J. Myers, "Big data comes of age," Tech. Rep.

[44] M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2002, pp. 233–246.

[45] J. LaFayette, "The future of 'big data' is apps, not infrastructure," "http://venturebeat.com/2013/01/04/the-future-of-big-data-is-apps-not-infrastructure/", 2013.

[46] M. Rouse, "Definition of big data analytics," "http://searchbusinessanalytics.techtarget.com/definition/big-data-analytics", January 2012.

[47] "Why big data analytics as a service?" "http://www.analyticsasaservice.org/why-big-data-analytics-as-a-service/", August 2012.

[48] P. O'Brien, "The future: Big data apps or web services?" "http://blog.fliptop.com/blog/2012/05/12/the-future-big-data-apps-or-web-services/", 2013.

[49] "What is big data? analytics as a service in the cloud," "http://www.analyticsasaservice.org/what-is-big-data-analytics-as-a-service-in-the-cloud/", March 2012.

[50] V. W. Consulting, "Big data, big impact: New possibilities for international development," The World Economic Forum, Tech. Rep., 2012.

[51] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," *McKinsey Global Institute*, pp. 1–137, 2011.