Levering Mobile Cloud Computing for Mobile Big Data Analytics

Yongxin Liu and Houbing Song

Abstract Mobile devices are becoming an indispensable tool for daily life and a considerable number of services are delivered via mobile devices. However, the capacity of mobile devices is constrained for complex interactive and computationally intensive applications (such as Siri on iOS), and therefore, cloud computing is needed to improve user experience. This results in mobile cloud computing. In this chapter, we first review the architectures of popular cloud computing platforms used in enterprise level application scenarios, then we present the requirements and challenges of cloud computing enabled service oriented intelligent mobile applications. After analyzing those challenges on both client side and cloud architecture, we propose the cloud computing architecture for mobile big data analytics and present several application cases.

1 Introduction

In this chapter, we will address several closely related concepts, i.e., cloud computing, mobile cloud computing, and mobile big data. The relationship among these three concepts is given in Fig. 1. As shown in Fig. 1, mobile cloud computing and general cloud computing, are closely related, since both mobile cloud computing and cloud computing are solutions of mobile big data collection, processing, and customized service delivery. Big data provide users with the ability to get insights and discover knowledge from the ocean of data while cloud computing provides the necessary engine.

Y. Liu
School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510640, Guangdong, China
e-mail: yongxin_liu@foxmail.com

H. Song (✉)
Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA
e-mail: h.song@ieee.org

G. Skourletopoulos et al. (eds.), *Mobile Big Data*, Lecture Notes on Data Engineering and Communications Technologies 10,
https://doi.org/10.1007/978-3-319-67925-9_2

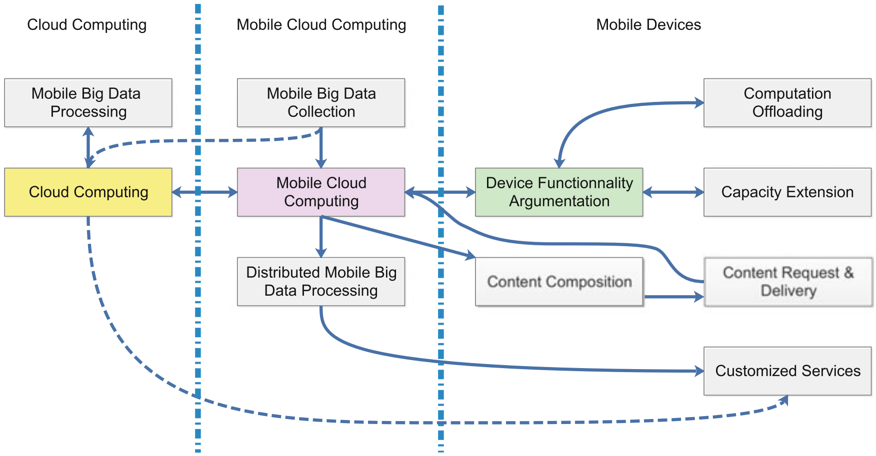


Fig. 1 Relationship among three concepts in this chapter

Mobile big data mining and harvesting is an application paradigm for mobile cloud computing. However, the mobile cloud computing must be specially enhanced from architectures to services for delivering services over resource-constrained mobile devices. With the help of mobile cloud computing, the functionalities of our mobile devices have been significantly enhanced.

The objective of this chapter is to present the opportunities and challenges of applying mobile cloud computing in mobile big data analytics. This chapter is organized as follows: we first provide an overview of general cloud computing application paradigm and its service models along with a logical architecture in Sect. 2. The key technologies of cloud computing, i.e., virtualization and middleware, are presented in Sect. 2.1 and Sect. 2.2, respectively. In Sect. 3, we introduce the challenges (Sect. 3.1) and feasible solutions (Sect. 3.2) for using cloud computing platform to provide service and enhance functionality for mobile devices. The application framework of mobile big data analytics is presented in Sect. 4.

2 An Overview of General Cloud Computing

Cloud computing is an emerging field in information technology that moves computing and data away from desktop and portable PCs into large data centers. Pervasive cloud computing is the antecedent of mobile cloud computing. The word cloud is a metaphor for describing the Web as a space where operating systems [26], applications, storage, data, and processing capacity all have been preinstalled and exist as a service, ready to be shared among users.

The main objective of cloud computing is to make better use of online resources and solve large-scale computation problems [10] (e.g. big data mining). An example

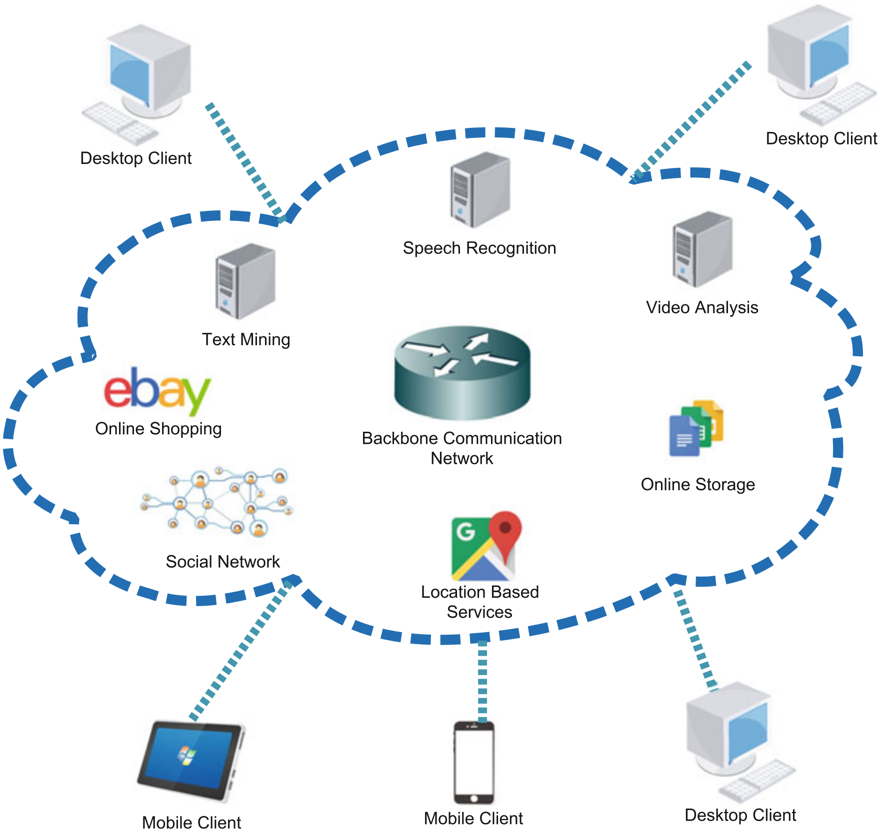


Fig. 2 An overview of cloud computing service

for cloud computing application is illustrated in Fig. 2. Cloud computing enhances the capacity of distributed computers to solve large scale computation problems [26], likewise, resources in the cloud provide transparent resource accessibility to a large number of users who do not need to know their exact locations and specifications [14]. In this scheme, on one hand, cloud applications and data are accessible to authenticated users from anywhere at any time. On the other hand, cloud computing providers offer their "services" in various forms.

According to the National Institute of Standards and Technology (NIST), three standard models are defined, i.e., Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS) [11, 20, 22]. These cloud service models are explained as follows (their hierarchical relationship is illustrated in Fig. 3).

- **Infrastructure as a Service (IaaS)**. Examples include Flexiscale and Amazon's EC2. The service is usually provided in the form of virtualized PC (also called node) where the consumer is able to deploy and run arbitrary software, which can

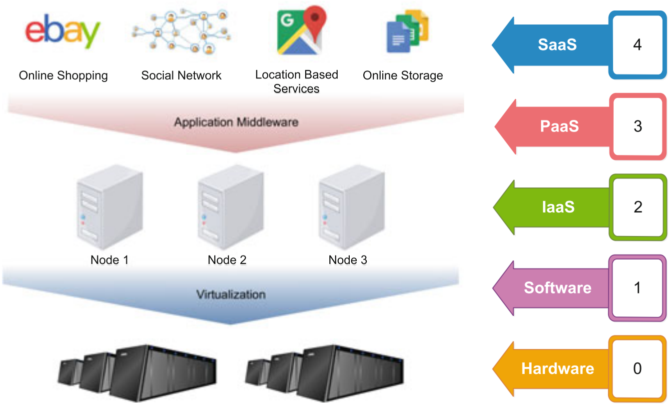


Fig. 3 Architecture of general cloud computing service

include operating systems and applications. Consumers do not manage or control the underlying cloud infrastructure but have full control over the virtualized operating systems, networking components (e.g., host firewalls), storage, and their own applications.

- **Platform as a service (PaaS)**. Examples include Google's Apps Engine, Salesforce.com, Force platform, and Microsoft Azure. In this mode cloud providers deliver a computing platform, which typically includes operating system, programming-language execution environment, database, and web server for end users. In such environments, users develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with hardware and low level operation system APIs.
- **Software as a Service (SaaS)**. Examples include Google Docs, Gmail, Salesforce.com, and Online Payroll. This mode enables users to access providers' applications running on a cloud infrastructure. The applications are accessible from various client devices through either web browser (e.g., web-based email), or a program interface (e.g., navigation service). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, but sends requests and receives responses from the cloud.

In Fig. 3 we can find that there are two supporting layers in cloud computing architecture, i.e., virtualization and application middleware, respectively. A principle of IaaS is that virtualization layer allows one physical server to run several individual computing environments [8]. In practice, it resembles multiple servers for each physical server user. In other words, cloud providers have large data centers which consist of servers to power their cloud offerings, but they cannot devote a single server to each customer [10, 37]. Likewise, application middleware encapsulates the details of virtual nodes and provides higher level developers a unified interface for deploying elastic services. These two supporting layers are discussed in the following two sections.

2.1 Virtualization Framework

An overview of virtualization architecture for cloud computing is illustrated in Fig. 4. This architecture can be divided into three layers.

- **Computing and networking infrastructures**. In this layer, physical servers are connected with high speed network, at the same time hypervisor operating system (e.g. ESX by VMWare or XenSever by Citrix) is installed to manipulate the hardware and network interface on each physical server node. The specialized operation system runs on "bare metal" with its own kernel and provides components for virtualization [8].
- **Resource pooling and management**. In this layer, resource pooling server aggregates resources from individual physical servers and provides unified management interfaces (e.g. XenCenter by Citrix or vCenter by VMWare) for clients to allocate resources and install operation system for virtual PCs [9]. On the other hand, this resource pooling server collaborates with physical severs to achieve the functionality of individual PCs. In some circumstances, where there is only one physical server, resource pooling sever and management client are combined.
- **Virtual nodes and network**. This layer consist of virtual machines simulated by resource pooling server and virtual networks connecting them. Those virtual machines are real objects providing specific services. It's notable that, network virtualization has played an important role in facilitating the flexibility in topology of these virtual nodes [13]. Thus, cloud providers can provide users with a cluster of nodes along with desired networking structure.

The virtualization of physical infrastructure stimulates and provides a consolidated foundation for IaaS, where users interact with the seamlessly simulated virtual nodes rather than the low level hypervisory systems [24].

For certain purposes, specialized hardware, such as Graphical Processing Unit (GPU) can be installed in physical servers from which they can be specially allocated to virtual nodes enabling the parallel computing capacity.

2.2 Middlewares

In cloud computating, middleware refers to the software framework that connects service resource to the application. In many cases, middleware is a major concept of

PaaS that provides users with an encapsulated environment with unified programming interfaces.

Although cloud service providers and subscribers have developed various service oriented middlewares for their applications, traditional middlewares for data manip-

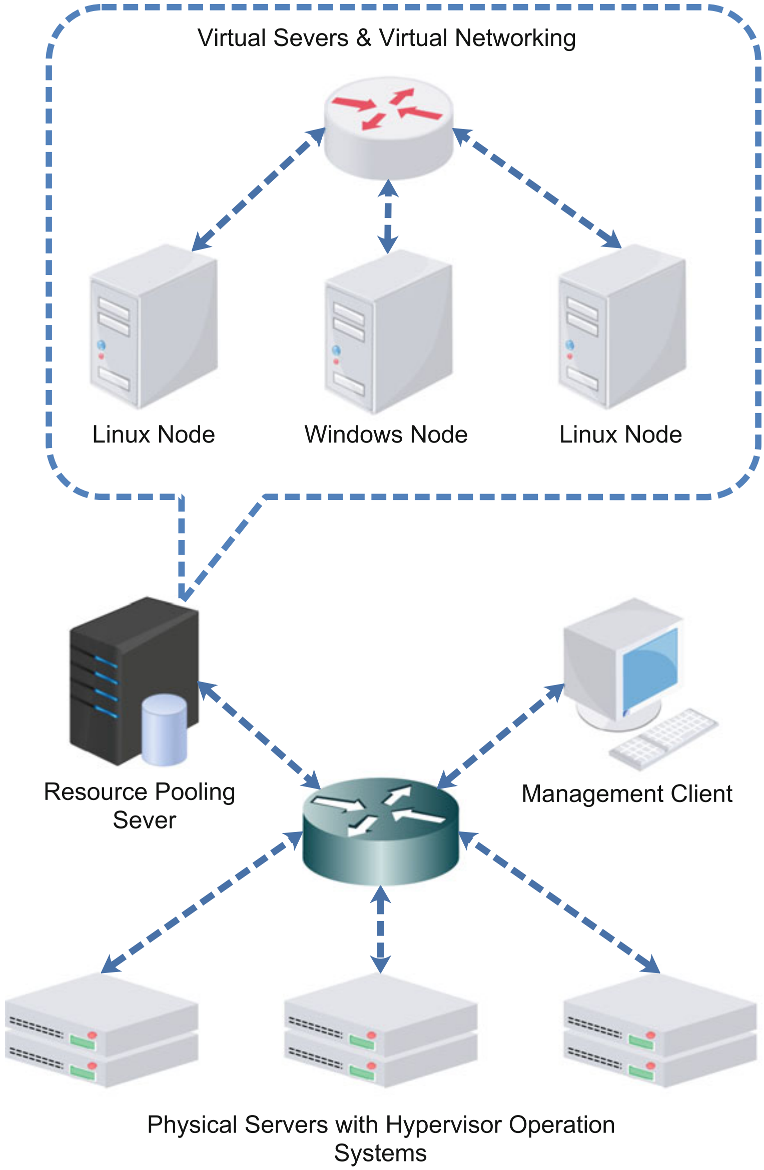


Fig. 4 Architecture of virtualization in cloud computing

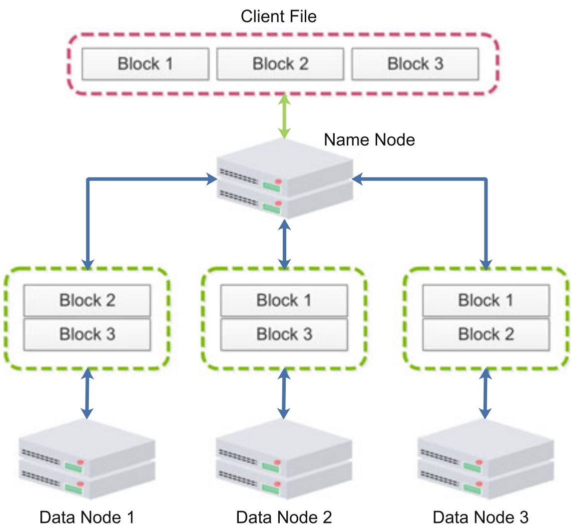


Fig. 5 An overview of mechanism of HDFS

ulation are inadequate for the era of mobile big data. For big data processing, specially for manipulating unstructured data from mobile devices and Internet of Things (IoT), advanced and versatile middlewares are needed to efficiently organize virtualized and even distributed resources (IaaS) to provide subscribers with unified API development and management interfaces. In this manner, the role of middleware for big data processing in cloud resembles a combination of API libraries and resource scheduling engines. In the following subsection, two state-of-art middlewares for cloud based big data storage, management and processing, i.e., Hadoop and Spark, are introduced respectively.

2.2.1 Hadoop

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models [2]. The core of Apache Hadoop consists of two major abstractions: a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce programming model.

HDFS: The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware [3]. It is similar to existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets.

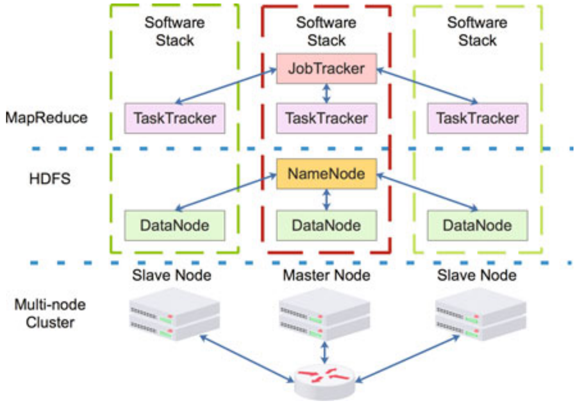


Fig. 6 Logical architecture of MapReduce along with HDFS

A brief architecture of HDFS is given in Fig. 5. HDFS consists of two types of nodes, namely, a NameNode called master and several DataNoes called slaves. HDFS can also include secondary NameNodes. The NameNode manages the hierarchy of file systems and directory namespace (i.e., metadata). File systems are presented in the form of NameNode that registers attributes, such as access time, modification, permission, and disk space quotas. The file content is split into large blocks (in the figure, the client's file is split into 3 blocks), and each block of the file is independently replicated across DataNoes for redundancy. This approach takes advantage of data locality, where nodes, in most of the processing period, manipulate the data within their vicinity. The feature of data locality reduces the dependence on high speed network for information exchange as in conventional parallel computing architectures.

MapReduce: MapReduce is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster (a group of computers) [6]. Processing can occur on data stored either in a filesystem (unstructured) or in a database (structured). MapReduce can take advantage of the locality of data, processing it near the place it is stored in order to minimize communication overhead.

The logical architecture of MapReduce along with HDFS is given in Fig. 6. MapReduce engine relies on the HDFS, which consists of one JobTracker, to which client applications submit MapReduce jobs. The JobTracker pushes work to available TaskTracker nodes in the cluster, striving to keep the work as close to the data as possible. With a rack-aware file system, the JobTracker knows which node contains the data, and which other machines are nearby. If the work cannot be hosted on the actual node where the data resides, priority is given to nodes in the same rack. This reduces network traffic on the main backbone network. If a TaskTracker fails or times out, that part of the job is rescheduled.

In most cases, using MapReduce, programmers are required to specify two functions only: the map function (mapper) and the reduce function (reducer), respectively

[17]. To illustrate the mechanism of MapReduce, we suppose an application scenario where we want to calculate the occurrence frequency of each word within one million articles. In this scenario, these articles have been merged into text blocks and stored in DataNodes. The MapReduce based application then works in the following manner:

1. **Map**: Each slave node counts the occurence frequency of words within the text block in its memory and generates an intermediate word count table. The master node ensures that only one copy of redundant text blocks is processed.
2. **Shuffle**: Reschedule the work load for merging the intermediate word count table generated by each node, e.g. words starting with letter `a` to `e` may be assigned to node 1, likewise, words starting with `f` to `h` can be assigned to node 2.
3. **Reduce**: Corresponding proportion of intermediate word count table is transmitted and processed to generate the overall statistic of word count. In this scenario, the overall count of words starting with letter `a` to `e` are derived by node 1 whilst node 2 provides the overall count of words starting with letter `f` to `h`. Finally, the master node collects the output results of each slave node and merges into our overall word count table.

Although HDFS and MapReduce are most critical components of Hadoop, several other current open-source Apache projects are related to the Hadoop ecosystem. These components can greatly boost the users to implement certain SaaS applications. It's also notable that all the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework [33, 40].

2.2.2 Spark

Compared with Hadoop, Spark is regarded as a more accessible, powerful and capable big data tool for tackling various big data challenges, because Spark enables applications in Hadoop clusters to run up to 10 times faster either in memory and on disk [27, 31]. Spark runs on top of existing HDFS to provide enhanced and additional functionalities, and therefore it is considered as a powerful complement to Hadoop. The architecture of Apache Spark is based on two main abstractions: Resilient Distributed Datasets (RDD) and Directed Acyclic Graph (DAG).

RDD: Resilient Distributed Datasets (RDDs) are collection of data blocks that are split into read-only partitions and can be stored in-memory on workers nodes (similar to the slave nodes in Hadoop cluster) of the spark cluster. In terms of datasets, apache spark supports two types of RDDs: Hadoop Datasets which are created from the files stored on HDFS and parallelized collections which are based on existing Scala collections. Spark RDDs support two different types of operations: Transformations and Actions. Transformations don't return a single value, since RDDs are immutable. The transformation functions just reads in an RDD and return a new RDD. An Action operation evaluates and returns a new value. When an Action function is called on a

RDD object, all the data processing queries are computed at that time and the result value is returned [36].

DAG: Directed Acyclic Graph (DAG) is a sequence of computations performed on data represented as graph, in which each node is an RDD partition and edge is a transformation on top of data. The DAG abstraction helps eliminate the Hadoop MapReduce multi-stage execution model and provides performance enhancements. In conventional hadoop platforms, when dealing with complicated tasks, developers have to connect together a series of MapReduce jobs and execute them in sequence. Each of those jobs is of high-latency. The job output data between each step has to be stored in the HDFS before other procedures can begin. The feature of DAG as well as the RDD, on one hand, replace the disk IO with in-memory operations and supports in-memory data sharing across DAGs, so that different jobs can work with the same data enabling complex work flows [15].

Spark is highly compatible with the Hadoop cluster. However, the logical definitions of nodes are slightly different although both Hadoop and Spark cluster follow a master-slaver hierarchy. An overview of Spark architecture over HDFS is illustrated in Fig. 7. In Fig. 7, the architecture consists of three type of nodes: master, slave, and resource manager. In small clusters, resource manager and master are combined. HDFS is deployed in the cluster, and the Spark's master node within this cluster can be the NameNode of Hadoop. When a task is submitted to the master node, the following steps are executed:

1. When task driver submits a task, it sends the request to the resource manager.
2. Resource manager checks data locality and finds the best available slave nodes for task scheduling.

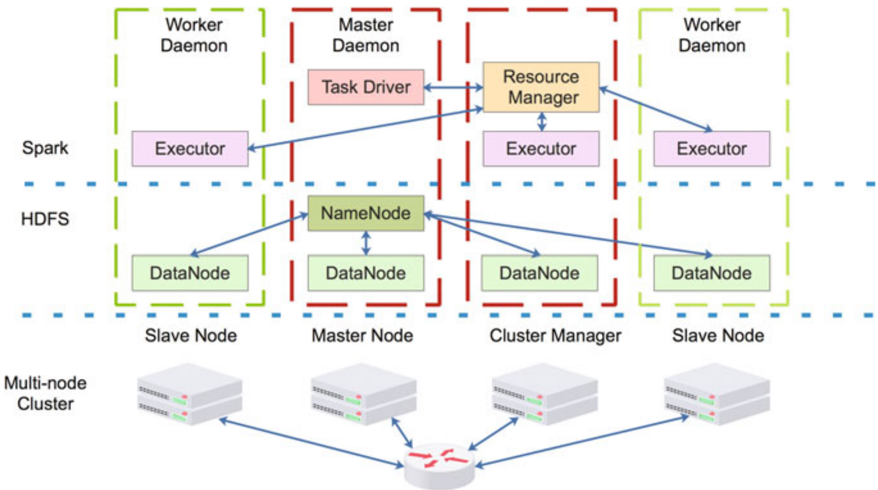


Fig. 7 Logical architecture of spark over HDFS

3. The task is split into different stages based on data locality and resources. Detailed information of the task is sent to slave nodes by the task driver in master node.
4. Task driver keeps track of currently executing task and updates the job monitoring status on master node.
5. Once the task is completed, all the nodes share the aggregated results to the master node.

Nowadays in big data processing applications, specially for mobile big data analytics [1], the fusion of Hadoop and Spark is regarded as an optimal solution.

3 Challenges and Solutions of Mobile Cloud Computing

Mobile devices have become an indispensable tool of daily life. Delivering services and gathering information through mobile devices have also become an inevitable trend in the era of mobile Internet. However, the capacity of mobile devices is constrained where rich media and computationally intensive applications can not be carried out by mobile terminals directly. Hence, as shown in Fig. 1, the cloud computing services with full capacity is considered as a powerful complement to resource-constrained mobile devices to alleviate their burden from heavy load tasks and deliver optimum experience [16, 34]. This scheme has led to the emergence of a Mobile Cloud Computing (MCC). MCC is the combination of cloud computing, mobile computing and wireless networks. The ultimate goal of MCC is to enable the execution of rich mobile applications on capacity limited mobile devices [12].

In this section, we first present the challenges of mobile cloud computing. Then, we introduce solutions for tackling such challenges using enhanced cloud computing infrastructure.

3.1 Challenges

Smartphones have been improved in various aspects such as capability of processor, storage, wireless connectivity and sensory integration. There are still apparent bottlenecks for developing and deploying complicated applications on mobile devices. For instance, 3D and Argumented Reality games may require intense GPU assisted computation which may quickly exhaust the power of batteries. Although MCC is a feasible solution for such resource intensive applications, several challenges exist, resulting in the application development and deployment on mobile devices more complicated than on the desktop cloud clients.

- **Elasticity**: With the increase in the number of mobile users, cloud providers may encounter the phenomenon that in peak period, the amount of service requests may exceed the capacity of their computation resource while in valley period, their resources are far beyond abundance. This scenario may require automatic scheduling of virtual machines (VMs) to achieve the elasticity of services [28].

- **Wireless connectivity**: Wireless networks are supposed to be bandwidth-saving, less-reliable compared with the wired networks. Establishing and maintaining seamless connectivity between mobile MCC users and clouds in a wireless channel with various ISP and protocols are difficult [19]. For instance, mobile terminals have to re-establish their connections due to their roaming, resulting in disconnections of sessions. Hence, the quality of cloud computing services can be significantly degraded. On the other hand, the over crowded wireless channel or signal interference may even disable the access for cloud computing services.
- **Network latency**: Latency impacts the energy efficiency and user experience of cloud-mobile applications by causing delays. Especially in cellular networks, the capacity of cellular base stations, shadowing effect of buildings, and channel inference can all become possible causes of the latency of MCC connections [18]. On the other hand, there will be bottlenecks if mobile applications have to establish long distance connections to the remote cloud servers through the Wide Area Networks (WAN) where latency is an ineligible aspect. To reduce interaction latency, solutions such as Cloudlet have been proposed.
- **Battery duration**: The purpose of cloud computing is to enable resource-constrained mobile devices to deliver computational density services, however, the frequent service requests from wireless networks and inquiries for response from cloud may in contrast become a major source of power consumption, specially for communication network with low QoS. Therefore, it's important to eliminate the abuse of cloud services, in other words, the mobile devices and cloud should collaborate to provide users with better services.
- **Privacy**: Mobile devices are closely related to users' daily life. Information security and user privacy must be addressed when user related information is processed and transmitted through wireless networks. For instance, from a users' location itineraries, data scientists can easily mine their home and work location or other sensitive information [21, 38].

To wrap up, the major challenges for introducing cloud computing to enhance the functionality of mobile devices are: (a) the latency caused by communication networks or resource inadequacy of cloud providers; (b) Extra power consumption of mobile devices caused either by communication latency or response latency.

3.2 Solutions

In this section, we present the solutions to address the challenges of mobile cloud computing identified in the last subsection and provide an overview of the state-of-art architectures for mobile cloud computing.

- **CloudLet**: In conventional scenarios, mobile devices obtain services from a remote cloud infrastructure. In such process, mobile devices usually access the

wide area network via WiFi access points (APs) or cellular networks where, as discussed in previous subsection, issues such as latency, connectivity may occur, degrading users experience. To mitigate this problem, a series of dedicated high speed networks connected cloud servers are distributed in the vicinity of potential users [30], and therefore, low latency can easily be achieved because there are less packet forwarding in congested backbone networks. However, the deployment of Cloudlet is not so straightforward. Service providers need to balance the cost of subscribing spatially separated server infrastructure, the expense of data traffic between remote servers and the profit they can obtain from mobile users. A possible solution is to use data driven approaches to derive mobile users' locations and preference so as to optimize the distribution of either CloudLet infrastructures and distribution of contents iteratively [5].
- **5G Networks**: In the paradigm of 5G networks, the users' handover is the key component, where users' mobility is provisioned and traffic flow is moved to the next point of attachment [23], i.e. the next base station, with no handover request from the mobile devices. This may boost the development of mobile cloud computing with better connectivity.
- **Dynamic partitioning**: Partitioning technology is introduced to automatically decompose applications to mobile devices and cloud servers so that they can be processed optimally. In the current stage, most partitioning strategies try to distribute applications either to mobile devices and cloud servers without an application controller or a user interface [35, 39]. In this paradigm, the capacity of cloud services is considered to be limited, and therefore, applications may automatically coordinate with remote cloud servers and assess the quality of user experience and decide whether to execute tasks remotely or locally. By using dynamic partitioning, we may reduce the burden of mobile cloud servers by collaborating mobile devices with redundant capacities.
- **Access authorization**: Customized services along with precision decision making rely on highly sensitive data, such as location, contact list or physiological sensory data. This type of data could bring unpredictable harms to mobile users if it is used inappropriately [7]. Therefore, it's necessary for mobile operating systems to block unauthorized access for sensitive data in any case. On the other hand, users should be aware of either data security or misappropriation usage.

Based on these solutions, the architecture of mobile cloud computing is depicted in Fig. 8. The MCC uses Cloudlet to send requests and deliver services to mobile users while the center cloud service is responsible for resource management and service deployment. It's notable that Cloudlets and center cloud infrastructure are connected by dedicated network (e.g. VPN). The architecture of single Cloudlet or center cloud are similar to ordinary cloud computing platforms.

4 Application Framework of Mobile Big Data Analytics

As illustrated in the concept diagram (Fig. 1) of this chapter, mobile big data analytics is an important application scenario of cloud computing. A general framework for big data and mobile big data analytics is illustrated in Fig. 9. By using Internet infrastructure in interconnection layer, mobile big data and other sources of data are sent to the data management layer, where the cloud infrastructure (e.g. HDFS, high capacity servers) are deployed. In this layer, data is divided in two types: the structured data (e.g. data tables) and unstructured data (e.g. sounds, videos). The above layer is the cloud computation layer, where data are analyzed using a series of methods as depicted. Finally, knowledge and patterns from data mining are applied for service support and future strategy optimization.

In the following subsection, several application cases for mobile big data analytics are introduced.

4.1 *Case Study: Smart Recommendation*

Accurate recommendation is difficult in many aspects due to the lack of preference information of their customers. Recommendation systems can benefit from the integration of mobile big data and context-aware data mining techniques. An example is provided by Sun et al. in [29]. They proposed a case study of IoT and big data analytics for smart tourism and sustainable cultural heritage in the city of Trento, Italy. Their system, called TreSight, integrated wearable sensors, open data, and participatory sensing enhances the services in the area of tourism and cultural heritage with a context-aware recommendation system.

The target users are cities that want to offer innovative services for citizens and visitors in a cost effective way such as cultural heritage, tourism-related companies

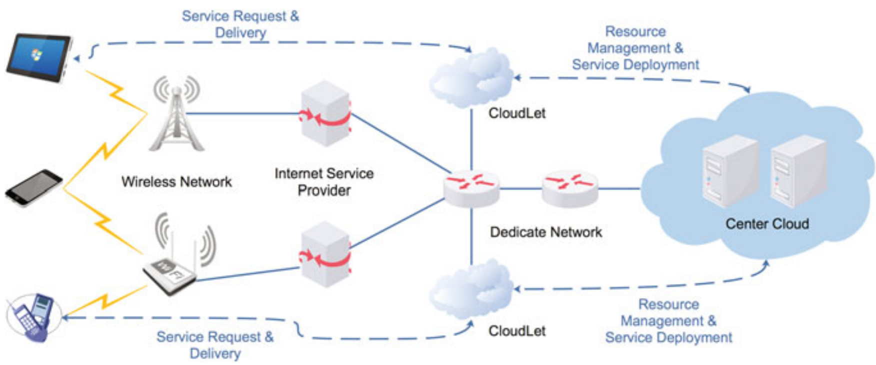


Fig. 8 Logical architecture for mobile cloud computing

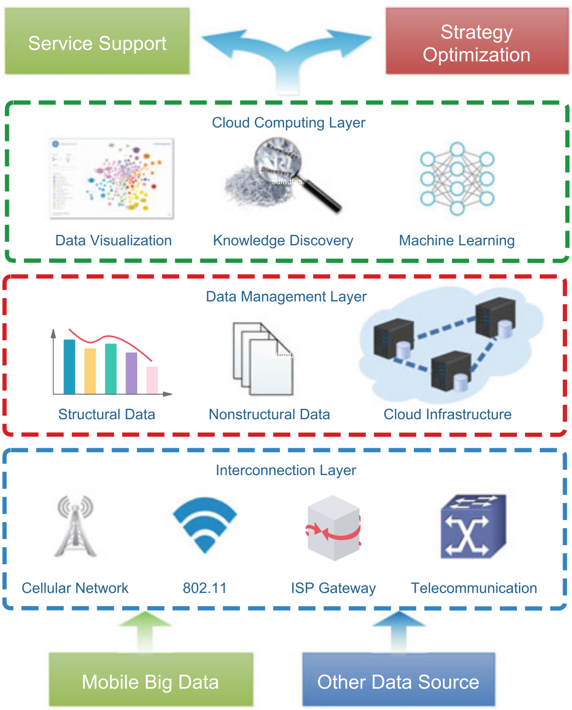
that want to promote themselves (hotels, museums, bars, restaurants, etc.) adding their advertisements, promotion codes, coupons etc. in the mobile app that will be offered to the users for the recommendation system.

Their solution first designed a wearable bracelet for each visitor, which is a crowd-sensing device, interacting with the mobile phone, and the Points of Interest, in order to provide the recommendation system the required data to make it context-aware.

The solution requires the deployment of a hotspot for each relevant place that want to be considered a Point of Interest. The hotspot is required to: (a) Gather the data about how many tourists have attended. (b) Update the data repository for a tourist indicating that he/she has visited this place. (c) Collect the sensed data about surrounding humidity, temperature, noise; (d) Provide additional information and content such as the real-time availability, reservations of a restaurant to the visitors through Bluetooth. (e) Finally, a mobile app will be used by visitors to interact with the bracelet, obtain the recommendations, get promotions (discounts, offers, and coupons from the promoted places and sponsors), and obtain more details about the points of interest (pictures, comments, statistics, open hours, current status information such as availability etc.).

Integration of mobile big data strengthen their approach from these aspects: (a) Better understanding the underlying preference from mobile big data. (b) From the cloud level, their approach manipulates nearly every piece of useful information, this

Fig. 9 General framework for pervasive big data analytics



is an essential quality of a good recommender. (c) Successful interaction with visitors via wearable devices provides real-time recommendation and useful information.

4.2 Case Study: Intelligent Healthcare

Mobile devices are integrating more sensors than ever before and continuous information about its owner is collected as time goes by. By leveraging such data, service providers can get a thorough understanding of the user from various aspects [4] which may potentially stimulate better healthcare services. On the other hand, by leveraging the cloud, medical service providers, the potential patients or companies can be connected together. In this way, we may monitor users' health conditions. An example application is provided by Wan et al. in [32] with a framework for a pervasive healthcare system with MCC capability to provide three types of scenarios (home, hospital, or outdoor environment) for ambulatory monitoring, and support a point of care to patients, the elderly, and infants in different environments.

Their system is composed of four main components: WBANs (Wireless Body Area Networks), wired/wireless transmission, cloud services, and users. WBANs collect various vital signals such as body temperature or heart rate information from wearables or implantable sensors. The collected monitoring data are processed in the cloud and then selectively transmitted to the users. The medical video stream from cameras are transmitted to the adjacent routing equipment via wired or wireless transmission, and then to the cloud server via the Internet. Cloud servers possess powerful VM resources such as CPU, memory, and network bandwidth in order to provide all kinds of cloud services such as automatic diagnosis and alarm, geographical information system (GIS) services, location-based services, and medical decision making (MDM). Different users such as hospitals, clinics, researchers, and even patients ubiquitously acquire multiple cloud services by a variety of interfaces such as personal computers, TVs, and mobile phones. This enables the sharing of monitoring data to authorized social networks or medical communities to search for personalized trends and group patterns, offering insights into disease evolution, the rehabilitation process, and the effects of drug therapy.

In their system, patients' profile and medical history data are maintained by the management center of the local private cloud. According to a user's service priority and/or doctor's availability, the doctor may access the user's information as needed. At the same time, automated notifications can be issued to his/her relatives based on this data via various telecommunication means. Besides these basic services, the cloud services also provide GIS deployment, medical data storage, MDM, virtual resource optimization management, and so on. With cloud support, the mobile devices of medical staff will easily exhibit richer mobile video streaming from remote cameras.

4.3 Case Study: Urban Analytics

The ubiquity of mobile big data for urban analytics can be categorized in two ways: first, in microscopic level, mobile trajectory collection on client side provides individuals' coordinates as well as related timestamps, as the gradual accumulation of data, all types of information including individuals' frequency pattern becomes available. From the frequency pattern set, users' mobility pattern can be derived and their behavior in a short range of future time can be predicted; second, in macroscopic level, the aggregation of mobile trajectories from different groups of users results in a dynamic and insightful image of the flow of crowds, from which we are able to assess the occupancy or quality of service of transportation infrastructure while this kind of work in traditional transportation engineering is undertaken with tiny amount of samples from manual survey.

Qiao et al. in [25] introduced a mobility analytical framework for mobile big data, based on real data traffic collected from second-, third- and fourth-generation networks, which covered nearly 7 million people. To construct a user's historical trajectories, they applied different rules to extract users' locations from different data sources and reduce the noise in their data.

They further explore human movement behavior in densely populated areas. They employ a parameter-free method to identify city hotspots from the view of population, apply a modified version of the Apriori algorithm to mine maximal sequential pattern, discover similar users based on their historical trajectories, and predict users' future movements from both temporal and spatial perspectives. These functionalities are of significance for improving the user experience of location-based service (LBS), for optimizing network resources, and for advising city planning.

5 Closing Remarks

Mobile big data analytics has the potential to benefit our society by enabling the move from data to knowledge to action. In this move, mobile cloud computing, which combines cloud computing, mobile computing, and wireless networks, to bring rich computational resources to mobile users, network operators, as well as cloud computing providers, plays an important role. This chapter presents the opportunities and challenges of leveraging mobile cloud computing for mobile big data analytics. We expect that the mobile big data analytics enabled by mobile cloud computing coudl reduce data transfer times, remove potential performance bottlenecks, and increase data security and enhance privacy while enabling advanced applications.

References

1. Alsheikh, M.A., Niyato, D., Lin, S., Tan, H.P., Han, Z.: Mobile big data analytics using deep learning and apache spark. IEEE Netw. **30**(3), 22–29 (2016)
2. Awadallah, A.: Introducing apache hadoop: the modern data operating system. Lecture given at Stanford University. Retrieved Feb 16, **2014** (2011)
3. Borthakur, D., et al.: HDFS architecture guide. Hadoop Apache Project **53** (2008)
4. Chen, M., Zhang, Y., Li, Y., Mao, S., Leung, V.C.: EMC: Emotion-aware mobile cloud computing in 5g. IEEE Netw. **29**(2), 32–38 (2015)
5. Cheng, X., Fang, L., Yang, L., Cui, S.: Mobile big data: the fuel for data-driven wireless. IEEE Internet Things J. (2017)
6. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
7. Dinh, H.T., Lee, C., Niyato, D., Wang, P.: A survey of mobile cloud computing: architecture, applications, and approaches. Wireless Commun. Mobile Comput. **13**(18), 1587–1611 (2013)
8. García-Valls, M., Cucinotta, T., Lu, C.: Challenges in real-time virtualization and predictable cloud computing. J. Syst. Arch. **60**(9), 726–740 (2014)
9. Gulati, A., Holler, A., Ji, M., Shanmuganathan, G., Waldspurger, C., Zhu, X.: Vmware distributed resource management: design, implementation, and lessons learned. VMware Tech. J. **1**(1), 45–64 (2012)
10. Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U.: The rise of big data on cloud computing: review and open research issues. Inf. Syst. **47**, 98–115 (2015)
11. Hogan, M., Liu, F., Sokol, A., Tong, J.: Nist cloud computing standards roadmap. NIST Special Publication **35** (2011)
12. Huang, D., Xing, T., Wu, H.: Mobile cloud computing service models: a user-centric approach. IEEE Netw. **27**(5), 6–11 (2013)
13. Jain, R., Paul, S.: Network virtualization and software defined networking for cloud computing: a survey. IEEE Commun. Mag. **51**(11), 24–31 (2013)
14. Jula, A., Sundararajan, E., Othman, Z.: Cloud computing service composition: a systematic literature review. Expert Syst. Appl. **41**(8), 3809–3824 (2014)
15. Karau, H., Konwinski, A., Wendell, P., Zaharia, M.: Learning spark: lightning-fast big data analysis. "O'Reilly Media, Inc." (2015)
16. Kumar, K., Liu, J., Lu, Y.H., Bhargava, B.: A survey of computation offloading for mobile systems. Mobile Netw. Appl. 1–12 (2013)
17. Lämmel, R.: Googles mapreduce programming model revisited. Sci. Comput. Program. **70**(1), 1–30 (2008)
18. Lei, L., Zhong, Z., Zheng, K., Chen, J., Meng, H.: Challenges on wireless heterogeneous networks for mobile cloud computing. IEEE Wireless Commun. **20**(3), 34–44 (2013)
19. Liu, F., Shu, P., Jin, H., Ding, L., Yu, J., Niu, D., Li, B.: Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications. IEEE Wireless Commun. **20**(3), 14–22 (2013)
20. Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., Leaf, D.: Nist cloud computing reference architecture. NIST Spec. Publ. **500**, 292 (2011)
21. Liu, H., Zhou, Y., Zhang, Y.: Estimating users' home and work locations leveraging large-scale crowd-sourced smartphone data. IEEE Commun. Mag. **53**(3), 71–79 (2015)
22. Mell, P., Grance, T., et al.: The nist definition of cloud computing (2011)
23. Morales, A.C., Aijaz, A., Mahmoodi, T.: Taming mobility management functions in 5g: handover functionality as a service (FAAS). In: Globecom Workshops (GC Wkshps), 2015 IEEE, pp. 1–4. IEEE (2015)
24. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The eucalyptus open-source cloud-computing system. In: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 124–131. IEEE Computer Society (2009)

25. Qiao, Y., Cheng, Y., Yang, J., Liu, J., Kato, N.: A mobility analytical framework for big mobile data in densely populated area. IEEE Trans.Veh. Technol. **66**(2), 1443–1455 (2017)
26. Sadiku, M.N., Musa, S.M., Momoh, O.D.: Cloud computing: opportunities and challenges. IEEE Potentials **33**(1), 34–36 (2014)
27. Samadi, Y., Zbakh, M., Tadonki, C.: Comparative study between hadoop and spark based on hibench benchmarks. In: 2016 2nd International Conference on, Cloud Computing Technologies and Applications (CloudTech), pp. 267–275. IEEE (2016)
28. Sanaei, Z., Abolfazli, S., Gani, A., Buyya, R.: Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Commun. Surv. Tutor. **16**(1), 369–392 (2014)
29. Sun, Y., Song, H., Jara, A.J., Bie, R.: Internet of things and big data analytics for smart and connected communities. IEEE Access **4**, 766–773 (2016)
30. Verbelen, T., Simoens, P., De Turck, F., Dhoedt, B.: Cloudlets: Bringing the cloud to the mobile user. In: Proceedings of the third ACM workshop on Mobile cloud computing and services, pp. 29–36. ACM (2012)
31. Verma, A., Mansuri, A.H., Jain, N.: Big data management processing with hadoop mapreduce and spark technology: a comparison. In: Symposium on, Colossal Data Analysis and Networking (CDAN), pp. 1–4. IEEE (2016)
32. Wan, J., Zou, C., Ullah, S., Lai, C.F., Zhou, M., Wang, X.: Cloud-enabled wireless body area networks for pervasive healthcare. IEEE Netw. **27**(5), 56–61 (2013)
33. White, T.: Hadoop: the definitive guide. "O'Reilly Media, Inc." (2012)
34. Xu, Y., Mao, S.: A survey of mobile cloud computing for rich media applications. IEEE Wireless Commun. **20**(3), 46–53 (2013)
35. Yang, L., Cao, J., Yuan, Y., Li, T., Han, A., Chan, A.: A framework for partitioning and execution of data stream applications in mobile cloud computing. ACM SIGMETRICS Perform. Eval. Rev. **40**(4), 23–32 (2013)
36. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M.J., Shenker, S., Stoica, I.: Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, pp. 2–2. USENIX Association (2012)
37. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. **1**(1), 7–18 (2010)
38. Zhang, X., Yi, Z., Yan, Z., Min, G., Wang, W., Elmokashfi, A., Maharjan, S., Zhang, Y.: Social computing for mobile big data. Computer **49**(9), 86–90 (2016)
39. Zhu, C., Leung, V.C., Hu, X., Shu, L., Yang, L.T.: A review of key issues that concern the feasibility of mobile cloud computing. In: Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, pp. 769–776. IEEE (2013)
40. Zikopoulos, P., Eaton, C., et al.: Understanding big data: analytics for enterprise class hadoop and streaming data. McGraw-Hill Osborne Media (2011)

Springer
http://www.springer.com/978-3-319-67924-2

Mobile Big Data
A Roadmap from Models to Technologies
Skourletopoulos, G.; Mastorakis, G.; Mavromoustakis, C.; Dobre, C.; Pallis, E. (Eds.)
2018, X, 347 p. 88 illus., 55 illus. in color., Softcover
ISBN: 978-3-319-67924-2