

Energy-Aware Issues for Handling Big Data in Mobile Cloud Computing

Chhabi Rani Panigrahi, Rajesh Kumar Verma, Joy Lal Sarkar
and Bibudhendu Pati

Abstract The popularity of mobile devices has been growing at a very fast rate and it is evident from the fact that it is possessed by almost each and every person and some may have even more than a single mobile device. MCC helps in computation and running of various complex applications on the mobile device and also offloads to the cloud when it requires lot of resources for computation or storage purposes. However, as energy is limited in the mobile device, processing of complex applications using big data is a challenge that needs to be addressed using energy efficient architectures. In this work, we mainly focus on identifying energy-aware issues for handling big data in Mobile Cloud Computing (MCC) environment and their current solutions. Also, we have included the review of few techniques available to handle big data in mobile devices. This chapter will also include a brief discussion of techniques available to process big data in MCC in an energy efficient manner. Finally, we conclude with an analysis of identified issues for handling big data in MCC and future scope of research.

1 Introduction

In today's world, around 2.5 Quintillion bytes of data gets generated every day [1]. In fact, data is growing at a very fast pace as there are huge number of data

C.R. Panigrahi · J.L. Sarkar (✉)

Department of Computer Science, Central University of Rajasthan, Ajmer, India
e-mail: joylalsarkar@gmail.com

C.R. Panigrahi

e-mail: panigrahichhabi@gmail.com

R.K. Verma

Cloud and Infrastructure Services (CIS) Lab, Infosys Limited, Hyderabad, India
e-mail: rajeshverma_chicago2004@yahoo.com

B. Pati

Department of Computer Science and Engineering, C.V. Raman College of Engineering,
Bhubaneswar, India
e-mail: patibibudhendu@gmail.com

© Springer International Publishing AG 2018

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_10

producing devices like micro sensing devices, software logs/alerts/notifications, cameras, Radio-Frequency Identification (RFID) and Wireless Sensor Networks (WSN) [2, 3]. The big data needs to be processed in an energy efficient manner and useful information must be derived from this big data, which will be beneficial for the society. Big data refers to data sets or combinations of data sets having features such as big size (volume), complexity (variability), and rate of growth (velocity) (3 V's according to Gartner's research). The conventional technologies and tools, such as relational databases cannot process this big data within the required period of time to make it useful. Most analysts and practitioners currently refer to data sets ranging from 30 to 50 terabytes to multiple petabytes as big data. Also, big data helps to resolve the current challenge of processing large amounts of data of heterogeneous type (it may be either structured, semi or un-structured data). Also, it helps by doing the processing in parallel (like a sorting of an entire country's census data) [4].

The growth of mobile devices in the world has been so rapid that it has far exceeded the human population of the world [5]. The number of smart phone users in the world is greater than 2 billion, and the number is expected to touch 2.7 billion by 2019. Also, greater than 60% of the web traffic is generated from mobile devices, which depicts a radical shift from the conventional desktops and laptops. This leads to generation of humongous volume of data which is coming at huge velocity and has variety in it. In order to handle this type of mobile data flowing to and fro, big data comes to the rescue, as it efficiently handles all related operations and delivers value to the user [6].

MCC refers to the computation which is generally initiated by the mobile device and done both on the Smart Mobile Device (SMD) and the cloud (through offloading process), and finally the results are displayed to the user on his mobile device [7]. The offloading approach is used in case where the computation is pretty complex and the vast cloud resources are required to perform the computation and storage activity. MCC emerges as a new computing paradigm where mobile devices exploit the available cloud computing platform for performing specific tasks and/or accessing data on demand. With the widespread exploitation of information, an increasing number of academic researches and industrial applications result in the appearance of big data from multiple heterogeneous sources in mobile clouds. Storage, transmission, analysis, and processing for such heterogeneous big data are crucially required [8]. Mobile devices have certain limitations such as limited battery energy, CPU, storage, and network bandwidth. So, it is important to consider the battery power of mobile devices in a MCC environment [9].

As per current research on Application Programming Interface, approximately 200 exabytes in 2014 and an estimation of 1.6 zettabytes in 2020 is supposed to be processed, 90% of these data are currently processed locally and the processing rate increases day by day. In the same time the risk of critical data theft, data and device manipulation, falsification of sensitive data as well as IP theft, manipulation and malfunction of server and networks also can not be avoided [10]. There is a great impact of data consolidation and data analytics in network configuration i.e. CISCO, HPE and others. Next, in application platform areas based on clouds and firewalls at the network boundaries are more prone from external attacks [10].

There can be several issues with respect to mobile big data such as latency, faster processing, energy etc. and among them energy plays a significant and indispensable part in all mobile related operations [11]. Hence, it is of utmost importance that one needs to concentrate on conserving energy in the mobile devices and also using it in an effective manner for successful running of applications for a good amount of time. When trying to run big data programs or storing huge amounts of data, it is generally not possible to do so on a mobile device. Under these circumstances, the offloading process comes to the rescue, wherein the computation and storage is done on the cloud. Of course, we need to weigh the advantages of performing the operations on the cloud with respect to the mobile device, as offloading to the cloud does involve a certain amount of energy besides the cost of using the cloud for our computational purposes. The current architecture for processing big data has several limitations. When the integration of big data with MCC is considered, it is advantageous as it can easily offload a huge chunk of computation and storage onto the cloud. For this several parallel programming paradigms have such as Map Reduce and others have been developed to process these big data. Several algorithms and architectures have been developed in the area of MCC that emphasize on energy savings [12]. The existing systems such as ThinkAir, Dream, eTime and others deliver efficient mechanisms of saving energy using offloading mechanism and help to execute mobile applications both in the mobile device and the cloud in an efficient manner [12, 13]. Also, several efficient Map Reduce algorithms have been implemented which require very less energy to execute the mobile big data [14, 15]. The vision here is to introduce a few techniques and architectures which will help to ensure that the big data related computations using MCC are handled in an efficient manner using optimum amount of energy of the mobile devices. Towards this direction of energy efficiency, the authors have made a humble effort to put forth the various architectures in order to achieve energy efficiency in case of MCC.

2 Energy Management Techniques in MCC with and without Big Data

2.1 Energy Efficiency of Mobile Devices Using MCC

Mobile Cloud Computing (MCC) refers to the integration of three different technologies, namely, cloud computing, mobile computing and wireless networks in order to accomplish the computational tasks required by mobile users [16].

The growth of mobile devices (Smart Mobile Devices—SMD's, tablets, laptops, etc.) over the years has been very rapid (around 7.22 billion mobile devices in 2016) that their number has already exceeded the entire population of the world (around 7.19 billion). SMDs are small in size and constrained with respect to their processing power, battery life, network bandwidth and storage. In order to complement the limitations of SMDs, Mobile Cloud Computing (MCC) plays an important role by

helping to offload the complex computation and storage requirement to the cloud which has unlimited processing power and storage [17]. Various researchers have proposed different mechanisms of saving energy which are discussed in the following subsections.

2.1.1 Phone2Cloud

In Phone2Cloud, offloading happens which saves energy required for computation [17]. Phone2Cloud deals with offloading approach which leads to substantial energy savings on Smart Mobile Devices (SMD) in MCC. SMDs like smart phones, tablets, etc., have become very common amongst the users in the world as they can run variety of intelligent applications (in health care, military, gaming, etc.) either on the device itself or by offloading to the cloud.

The YoY (Year on Year) growth of mobile devices is greater than 50% and the total number of mobile devices has far exceeded the total world population. Besides this, mobile access is popular as compared to fixed internet access, and this is evident from the fact that the number of mobiles has already far exceeded the number of Desktop computers in 2014. There are four fundamental approaches related to SMD's for preserving energy consumption which leads to an extended battery life.

1. Use of smart battery models and energy cost models, wherein there are various battery models such as ideal model and other models which are present.
2. Avoiding energy wastage, can be achieved by putting the state of the component (either the entire system or the individual components such as the processor) to a sleep state such that it does not hamper the user's work.
3. Savings related to communication, wherein the radio interface used by the Wi-Fi is automatically shut down by the power management mechanism. Another mechanism known as Coolspots can be used wherein switching happens between Wi-Fi and Bluetooth (consumes lower energy).
4. Savings through offloading of computation, as computationally intensive tasks can be shifted to be executed in the cloud which leads to SMD energy saving. Also, techniques which partition an application and only partially offload the tasks to the cloud are used.

Figure 1 shows the basic architecture of Phone2Cloud and there are seven basic components which comprise the architecture of Phone2cloud [17].

- *Bandwidth monitor and Resource monitor*: The bandwidth monitor monitors the current bandwidth being used and helps to make the offloading decision. Resource monitor will monitor the SMD status.
- Execution time predictor, an important component which tells us the total execution time of the application on the SMD.
- Offloading decision engine, which actually decides whether to offload and the extent of application to be offloaded from the SMD to cloud.

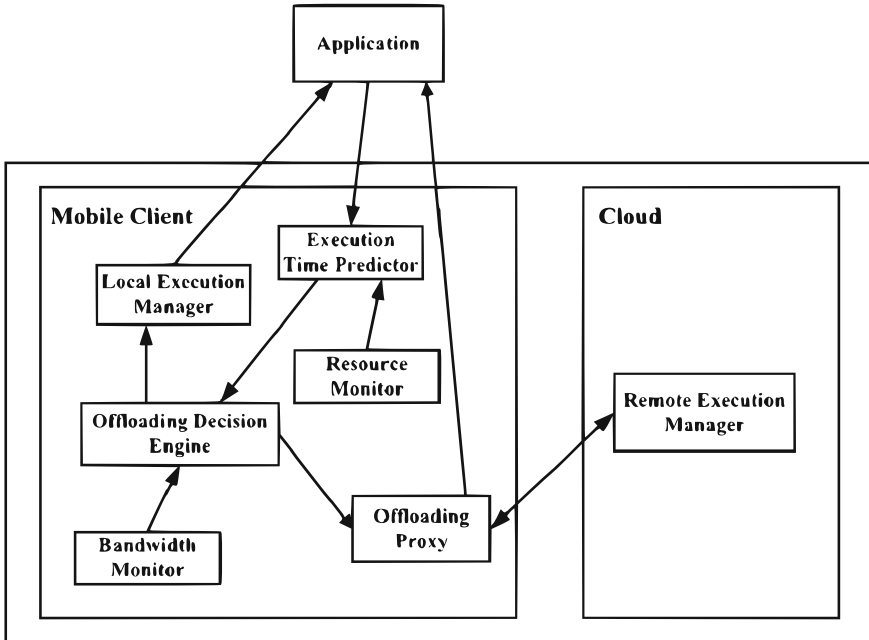


Fig. 1 Basic architecture of Phone2Cloud [17]

- Offloading proxy, which does the task of sending data to remote execution manager and the returned results after computation are sent to the application.
- Local execution manager and remote execution manager, both of these are responsible for managing the execution of application. As the name suggests, the local execution manager will execute application on the SMD (using OS like iOS or Android), and the remote execution manager will execute on the cloud.

Figure 2 shows the working flow for taking the offloading decision that means offload to the cloud or smartphone itself. Initially, an Execution Time Predictor (ETP) predicts the average time for execution denoted by T_{exec} and determines the users delay-tolerance threshold denoted by T_{delay} which is then compared to the T_{exec} . If $T_{delay} > T_{exec}$ then calculate the overall energy consumption for running an application on the smartphone denoted by E_{local} otherwise, the application is offloaded to the cloud. The energy consumption for running application on the cloud is denoted by E_{cloud} and by comparison with E_{local} , Phone2Cloud takes the decision for offloading. Figure 3 shows the application and scenario experiments used in case of Phone2Cloud.

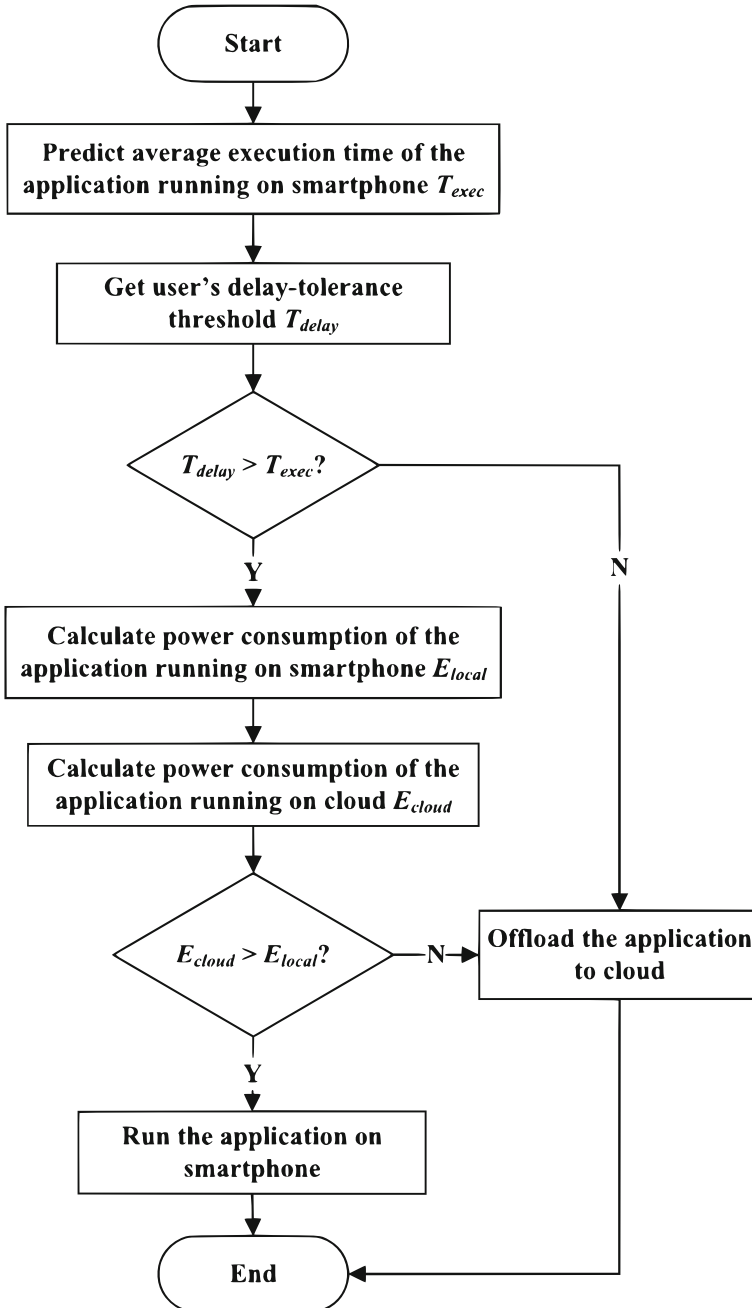
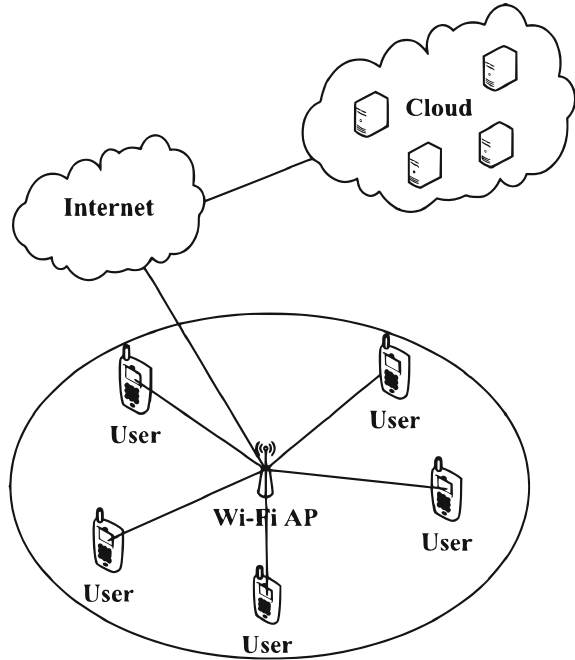


Fig. 2 Working flow for taking offloading decision

Fig. 3 Basic architecture of Phone2Cloud



2.1.2 eTime

eTime is about the energy efficient transmission between the cloud and mobile devices [13]. The need for offloading arises as the Smart Mobile Device (SMD) do not have enough resources and energy to perform complex computations and store huge data. The programs are offloaded from the mobile to the cloud such that they are efficiently executed on the cloud and the results are sent back to the SMD. It uses the Lyapunov technique and saves around one-third of the energy for different type of applications which are present on SMD’s. Also, eTime has intelligence built into it such that it takes advantage of the presence of good network connectivity to perform the offloading process. If the bandwidth of current downlink is $\omega(t)$ and a transmission decision $d(t) \in \alpha = \{transmission\ of\ P_i(t),\ idle\}$. Where, in time slot t , $d(t) = transmission\ of\ P_i(t)$ when the data queuing transmit in $P_i(t)$ of an application i and $d(t) = idle$ when for saving the energy the data transmission deferred. Therefore, the data transmitted from mobile to cloud denoted as $\varphi(t)$ is expressed as follows:

$$\varphi(t) = \begin{cases} \omega(t)\beta, & \text{if } d(t) = \text{transmission of } P_i(t); \\ \omega(t) = \text{idle}; & \end{cases} \quad (1)$$

the time span of one time slot is represented by β eTime takes an intelligent decision which is expressed as follows []:

$$\begin{aligned} \text{minimize } Q &= \lim_{T \rightarrow \infty} \text{Sup} \frac{1}{T} \sum_{t=1}^{T-1} \mathbb{E}|D(t)| \\ \text{Subject to } P &< \infty, d(t) \in \alpha \end{aligned} \quad (2)$$

where, $D(t)$ is the data transmission on a mobile device at the time slot t and T is the long time period and Q represents as the averaged energy consumption of the mobile device.

2.2 DREAM

DREAM stands for Dynamic REsource and task Allocation for energy minimization in Mobile cloud systems (DREAM). Offloading is a popular approach followed to save energy of Smart MobileDevices (SMD) in Mobile Cloud Computing (MCC) [18].

There are several factors that needs to be considered in real world like the offloading policy to the cloud, task allocation for execution on local mobile environment or to transmit to cloud and the CPU speed. The DREAM algorithm can save greater than one-third of the energy than the current algorithms. Figure 4 shows the architecture of DREAM and the different components in this architecture are as below:

- Mobile applications: can be of various types, such as mobile games, email, etc. These may or may not run in the mobile device.
- Mobile Device: consists of SMD.
- Application Interface: which is an interface from the mobile applications to the SMD. This interface is generally based on REST based API's.
- 3G/LTE: It is the network which is either 3G or Long Term Evaluation based and transmits the packets from the SMD to the cloud.

2.3 ThinkAir

In current world, Smart Mobile Devices (SMD) is becoming very popular and the number of SMDs is increasing every day. This popularity of SMDs is forcing developers to put their best of applications (like gaming applications, speech recognition, complex graphical applications, etc.) on the SMDs, so that it can be used by a large number of users and provide a good user experience in the hands of the user. These applications are quite complex in nature and also demand a lot or resources in terms of computational power and battery energy. This requires the support of offloading approach and “ThinkAir” is one such efficient framework used by the developers of these complex applications to offload their applications to the cloud in an efficient manner which enables the applications to run very easily and saves energy [12].

Figure 6 shows the architecture of ThinkAir and keeping the architectural viewpoints in mind about the easy use by the development team, wherein the interface is very simple and hence the developers find it very easy and convenient to use. Also, this framework takes care of not compromising on the performance part of the execution of any complex application and hence it is very easy for any new developer to use without causing any risk or delay for any cloud computation mechanism. This architecture also improves the performance of computation through offloading, and the speed of computation is increased and also less energy is used of the mobile device as the complex methods are offloaded and executed in the cloud and the end result is returned to the calling program. Also, ThinkAir facilitates Parallel execution for faster execution of complex applications, wherein the different methods are executed in parallel across the different Virtual machines (VM's), which helps to reduce the time required and also executes efficiently thereby delivering optimum performance of running complex applications.

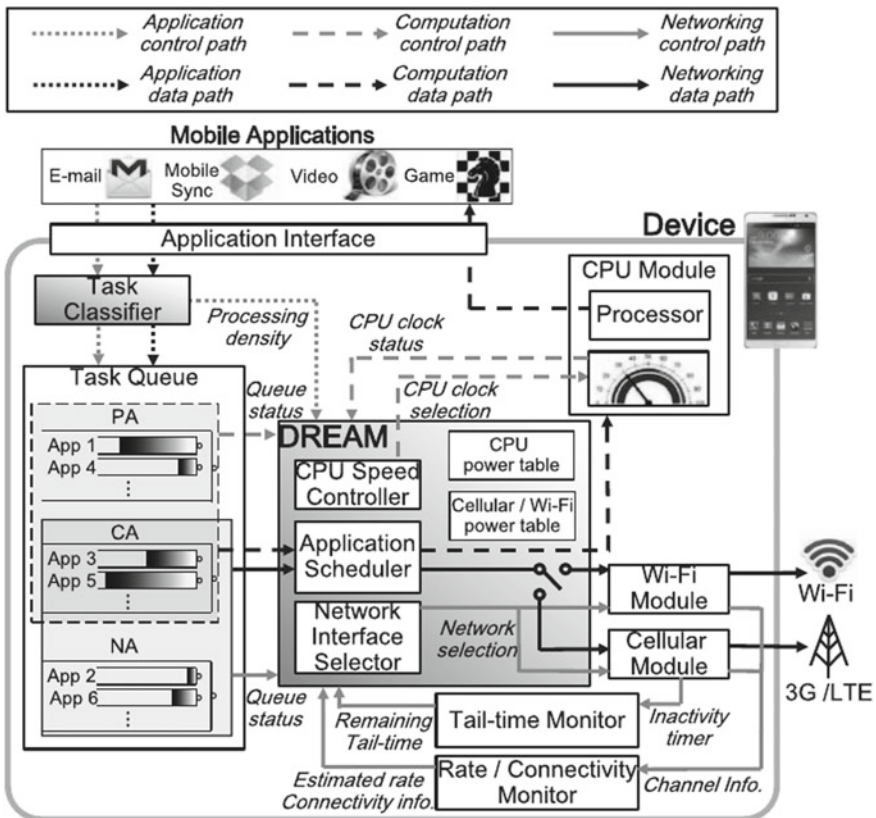


Fig. 4 Dream architecture

The different components of ThinkAir architecture are the Execution platform which is responsible for running of the complex methods in the cloud. It is mainly responsible for making the important decision of whether the particular method should be offloaded to the cloud or be run on the SMD itself. Intelligent algorithms are used which make the decision based on gathered data as well as the previous running of different methods of similar type. Also, the application server component is a very light weight component on the cloud which takes care of managing the code which is offloaded. This server is automatically started when the system is rebooted. The 3 main sub-components of this server includes the client handler, which is mainly responsible for the communication between the SMD and cloud. Besides, cloud infrastructure forms the part of the application server which is pretty flexible as it can be deployed on any of the virtualized cloud environment, whether public or private. Another component called the Profilers is also light weight component and is mainly of 3 different types like software, hardware and network. The Hardware profiler is responsible for inputting the hardware information into the energy model, and subsequently the decision to offload is taken. Different states of CPU, WiFi, Screen and others are monitored using the Hardware profiler. On the other hand, the software profiler is responsible for tracking the parameters connected to program execution like CPU details, time of execution, methods invocation number and others. The most complex of the profiler is the network one which is mainly responsible for collecting metrics like Round Trip Time (RTT), which in turn helps to get the network bandwidth which is very essential for taking decision about offloading (Fig. 5).

2.4 Energy Management Techniques in MCC with Big Data

There are a wide variety of applications available in mobiles ranging from human health care, military, environment, etc. [14]. The mobile applications are complex in nature, and need to process big data in order to come up with results. Under these circumstances, it is very necessary that we need to ensure that we are fully aware about energy related issues in case of mobile big data processing. Also, we need to ensure that we use different techniques and approaches in order to resolve this energy problem. The reason that these energy issues arise in case of mobile devices is because of the fact that the mobile is a small device and the battery is relatively small and hence cannot store huge energy [5]. Subsequently, battery life is less and we need to ensure that the energy is spent efficiently. The different sources of Big Data streams that come into the Mobile devices are shown in Table 1. Also, there are different formats of content that needs to be taken care.

The term Big data refers to data sets (or the combinations of data sets) whose size (volume), complexity (variability), and rate of growth (velocity) 3 Vs (Gartner) make them very difficult to be captured, managed, processed or analyzed by the present day conventional technologies and tools such as RDBMS (Relational Database Management Systems) and desktop statistics or visualization packages, within the time necessary to make them useful and advantageous for the user [19]. The size

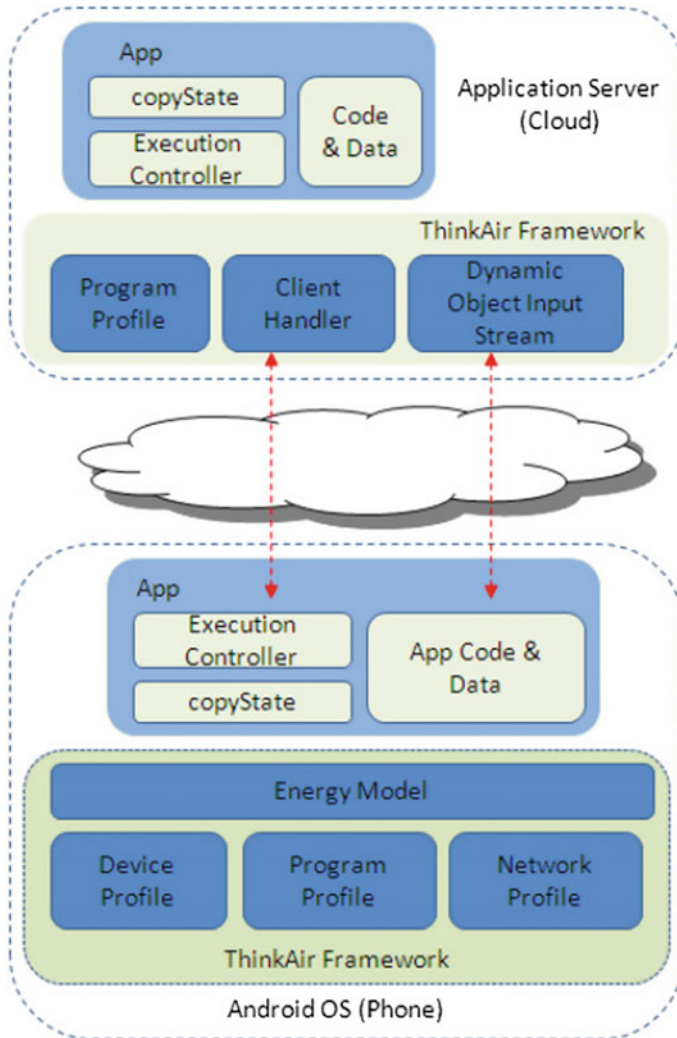


Fig. 5 Basic architecture of ThinkAir

that is fixed in order to determine whether a particular data set is considered big data is not firmly defined and continues to change over time, however, most of the analysts and practitioners currently refer to data sets from 30–50 terabytes (10^{12} or 1000 gigabytes per terabyte) to multiple petabytes (10^{15} or 1000 terabytes per petabyte) as big data. Big data is used to process large amounts of structured, semi-structured or unstructured data like analyzing log files, etc. Also, when the processing can easily be made parallel like a sorting of an entire countries census data, big data becomes very handy. While running batch jobs, big data can prove to be of big help (for example

Table 1 Big data stream sources

Classifications	Descriptions
Social media	Social media helps for communication between the various communities and the social networks such as Twitter, Facebook etc.
Crowdsourcing	Where different kind of people are use SMDs for successfully complete their task
Internet of Things (IoT)	Here the thin devices (e.g., smartphones, tablets, RFIDs, PDAs) have IP addresses and communicate over the internet for enabling services of various types

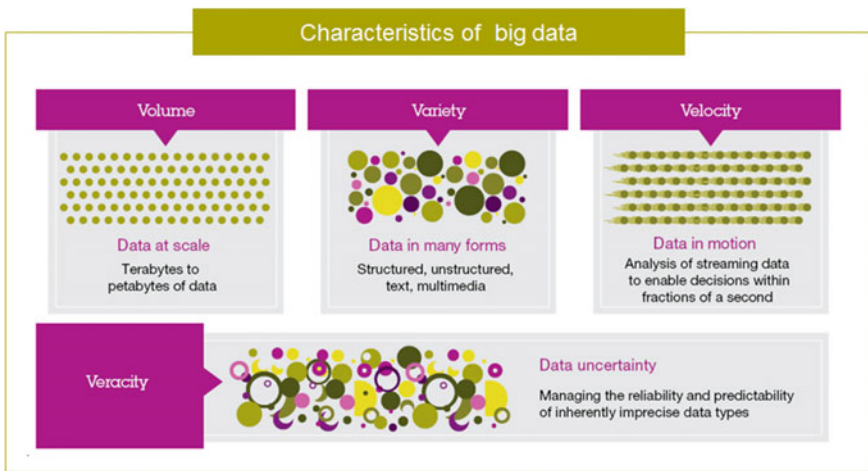


Fig. 6 Characteristics of big data

website crawling by search engines). As Big data involves lots of data it is specially advisable that you run these programs on cheap hardware (commodity hardware). It is not recommended that we use Big data to process GBs or few TBs of data [19].

The characteristics of big data are depicted in Fig. 6 [20].

The current data is growing big because of several factors such as advancements in Mobile Technology, better communication networks, cloud computing availability, etc. The big data solution helps to solve various aspects like fraud detection, get insights into the call center records, discover customer sentiments from social media and better customer segmentation for improved sales and marketing.

There are a wide variety of applications available in mobiles ranging from human health care, military, environment, etc. The mobile applications are complex in nature, and need to process big data in order to come up with results. Under these cir-

cumstances, it is very necessary that we need to ensure that we are fully aware about energy related issues in case of mobile big data processing. Also, we need to ensure that we use different techniques and approaches in order to resolve this energy problem. The reason that these energy issues arise in case of mobile devices is because of the fact that the mobile is a small device and the battery is relatively small and hence cannot store huge energy. Subsequently, battery life is less and we need to ensure that the energy is spent efficiently. There are several Big Data techniques has been proposed which may works with MCC environment which are discussed as follows:

1. *Big Data Stream Mobile Computing (BDSMC)*: The term “Big data stream mobile computing” comprises of two concepts which are merged together. They comprise of the culmination of broadband based stream which provides a constant stream of Big data and the other one is mobile cloud computing which helps in processing this big data at real time [15]. Also, offloading is very important out here and plays an important role to process the big data which is collected by the mobile devices. Figure 7 shows the BDSMC architecture. There are three layers namely, the Radio Access Networks (RANs) or the User layer which is the source for big data, the Internet backbone layer and the remote networked data center layer. Offloading of data and compute intensive applications to the cloud takes place using the best possible manner wherein least energy is consumed and using the appropriate access technologies (e.g., either of WiFi, 3G/4G)
2. *StreamCloud Architecture*: StreamCloud Manager (SeCoM) design was inspired by the workload offered by real time big data streams and also that setup costs and energy wastage due to frequent ON/OFF transitions [15]. SeCoM supports

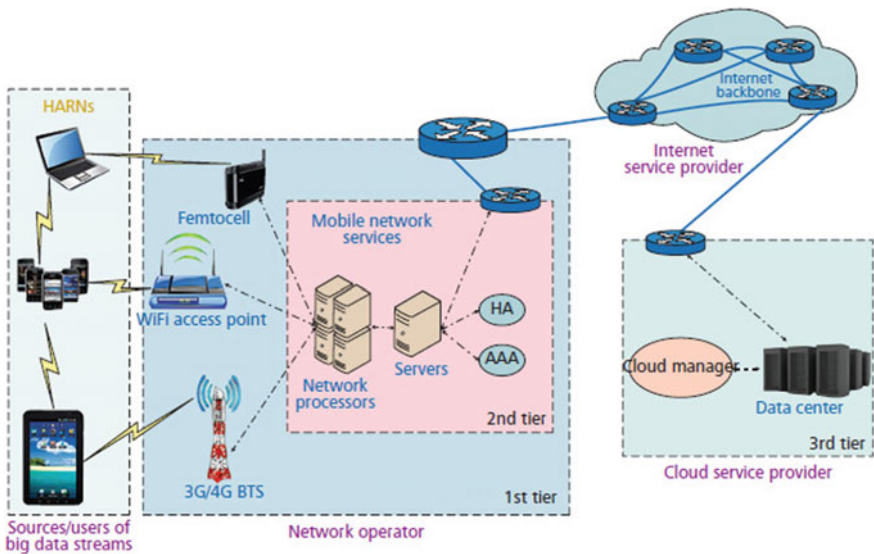


Fig. 7 BDSMC reference architecture [15]

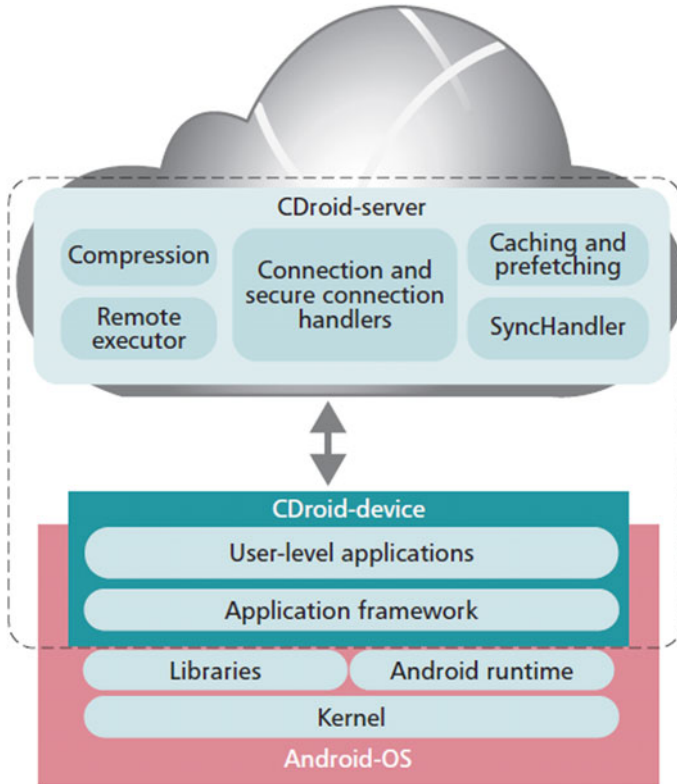


Fig. 8 VNetDC architecture based on StreamCloud platform

HIS (Hibernation stage) which helps to save energy. The main features of SeCoM module are the self-reconfiguration of the overall virtualized networked data centers (VNetDC) which helps to ensure minimum energy resource configuration. In the architecture, the following components are present. CDroid Server, which is responsible for doing the offloading of computation from mobile devices to the cloud. It has two sides, the device side and the cloud side. CDroid architecture has been shown in Fig. 8 and the components of the CDroid which help in energy saving offloading are:

- (i) Communication handling module, responsible for management of all data traffic over mobile Internet connection and also performs http traffic tunnelling of the mobile device via the CDroid server
- (ii) Caching and prefetching module.

Figure 9 shows the VNetDC architecture which is based on the StreamCloud platform which working as an ad-hoc-designed stochastic gradient algorithm and

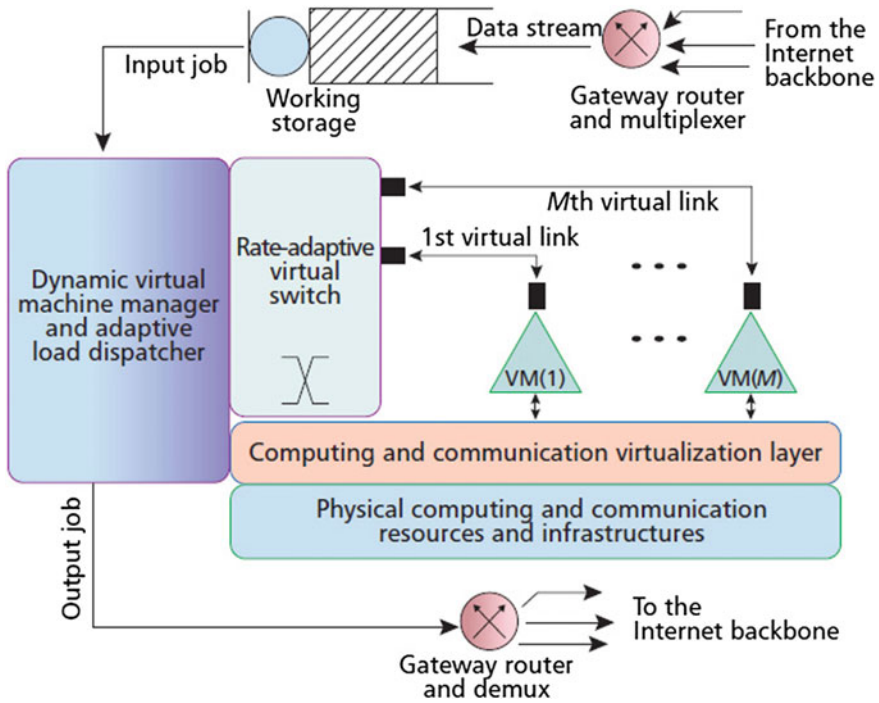


Fig. 9 BDSMC reference architecture

ensure that even if there are unpredictable dynamic fluctuation of the workload, it involves with the minimum-energy resource configuration.

3. *Mobile Cloud Sensing applications:* There are various type of applications which use MCC, Sensing capabilities together with Big data using 5G as the communication medium, which make life more meaningful and worthy [11]. These applications are spread over various areas like Health monitoring, environment monitoring and others. The taxonomy of the mobile sensing applications depends on the size of the applications which determines the volume of data, size of the application, execution speed and other parameters. We take a typical parameter of the size of the applications in mobile sensing area and it can be divided into the below sub-areas [11].

- Personal or Individual sensing, wherein the data is of personal users and used for their betterment, for example, the medical information of a person.
- Collective or Group sensing, and here the data belongs to the group and is aimed at their usage, like restaurant ranking and others.
- Community sensing, where the entire community data is collected and used to predict matters like global trends of warming and climate and other popular trends.

The typical architecture of mobile cloud sensing has been shown in Fig. 10.

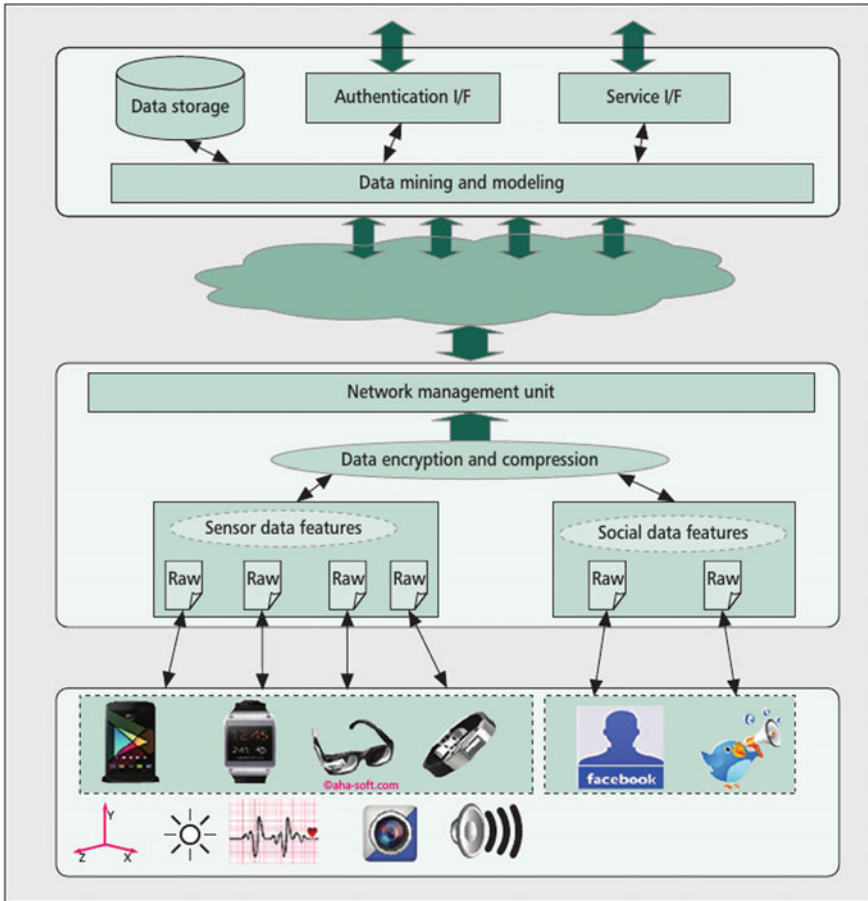


Fig. 10 Architecture of mobile cloud sensing

The different architecture components perform the functionality as explained.

- Data Sensing Unit
- Data Preprocessing Unit
- Network Management Unit
- Cloud Data Mining and Storage

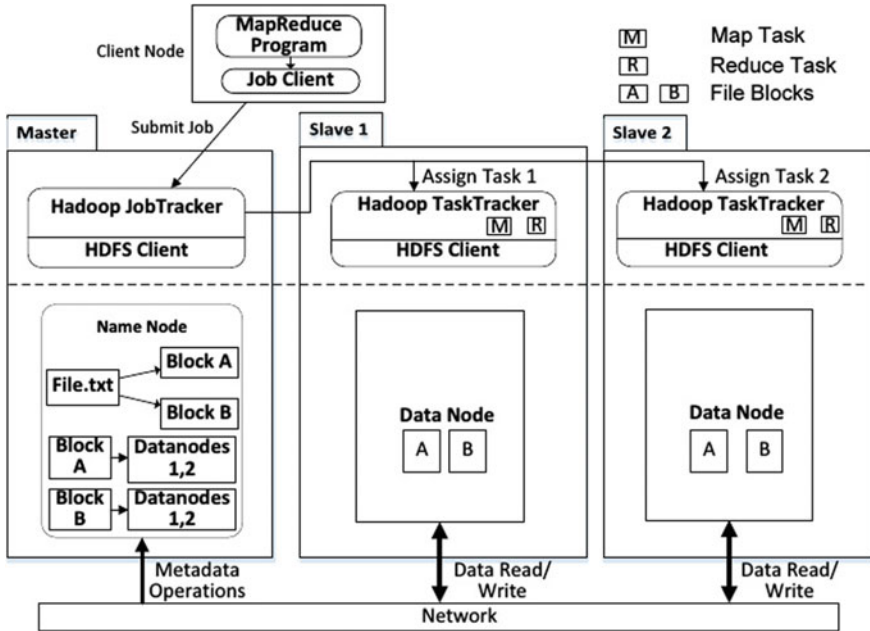


Fig. 11 Hadoop architecture [6]

However, there are also various limitations of Mobile Cloud Sensing, which are listed as follows.

- Currently, it is only in experimental stage and needs to be implemented in large scale
- The limitations of network resources
- Big Data usage

There are few works have been proposed which are based on the Hadoop platform. Figure 11 shows the Hadoop architecture and the various components of the Hadoop architecture are as follows [6]:

(i) Apache Hadoop system comprises of two basic components, mainly the Map Reduce (MR) framework and the Hadoop Distributed File System (HDFS). MR comprises of the Map and Reduce Task which is done in a parallel manner and runs on top of the HDFS. In the Map Task, the input data set is taken, and intermediate $\langle Key, Value \rangle$ pair is produced and is then sorted and partitioned per reducer. In the Reduce Task, the reducer identifies the aggregate function to be applied to the output of the map task and then runs it. The number of times a reducer needs to be run will be set by the user. This mapped output is then sent across to the reducer for deriving the final output. Below is the general form of the Map and Reduce Task:

map: $(K1, V1) \rightarrow list(K2, V2)$
 reduce: $(K2, list(V2)) \rightarrow list(K3, V3)$

In the MR framework, the computation is moved closer to the data node, which will minimize the congestion in the network and also maximize the throughput. Two important modules in MR are the JobTracker which accepts user jobs and does the splitting into multiple tasks and the TaskTracker which are the nodes that execute the tasks in the cluster that run the tasks.

(ii) HDFS is a distributed file system which is very reliable and also fault tolerant. It stores huge datasets in petabytes and beyond and is load balanced to achieve efficiency. Each file is split into blocks where-in blocks are replicated across several devices in the cluster. It is designed to run on commodity hardware. It provides high throughput access to application data and is suitable for applications that have large data sets. The two modules in HDFS are NameNode which holds the metadata information about the different files storing the Inode records of files and directories, and the DataNode which is actually storing the file blocks and helps to complete the read/write requests coming from any client.

The Mobile Distributed File System (MDFS) helps to take care of the big data processing in mobile clouds. It is built on a k-out-of-n-framework which ensures reliability and security of data. Every file uses a secret key to encrypt and is fragmented into several file fragments using Reed Solomon algorithm. Also the secret key is split into fragments using Shamir's secret key sharing algorithm. MDFS helps to ensure high security as data cannot be decrypted until an unless the authorized user does not obtain all the different key fragments. Figure 12 shows the architecture of MDFS. MDFS incorporates a directory service which is distributed and synchronization of each node happens on a periodic basis which ensures that all device directories are updated at all times.

Figure 13 shows the distributed MDFS architecture. In case of distributed MDFS architecture, there is no central controlling authority to manage the cluster. Here, every node has a Name Server and Fragment Mapper. Also, every node will sync up with other nodes in a regular interval. In case of any new node, the broadcast messages are received for all new nodes. The best part is that in this architecture

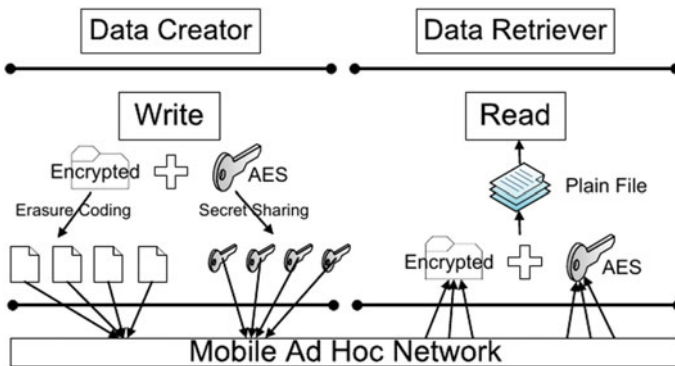


Fig. 12 MDFS architecture

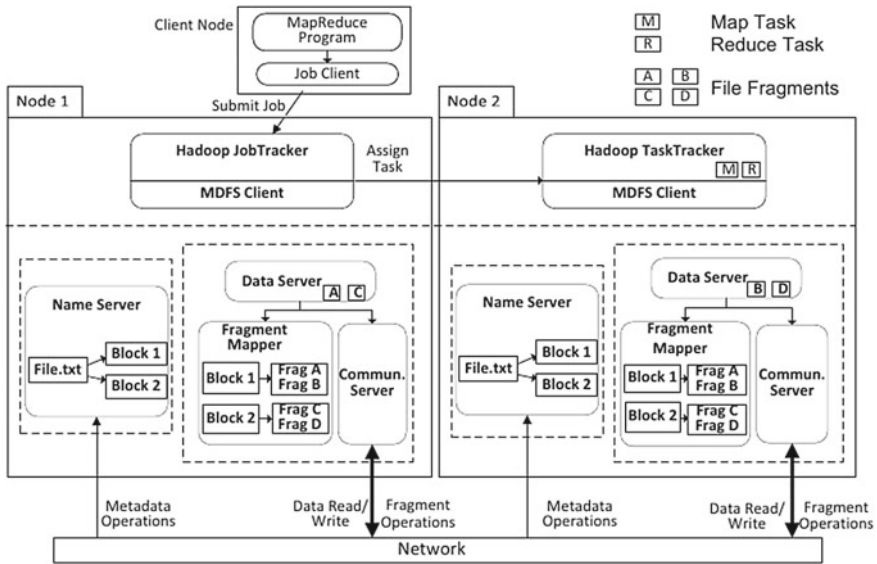


Fig. 13 Architecture of distributed MDFS

there is no single point of failure as each node can operate in an independent manner (since it stores a copy of the namespace and fragment mapping). This ensures that the load is evenly distributed across the cluster [6].

Figure 14 shows the centralized MDFS architecture. In case of centralized MDFS architecture, name server and Fragment Mapper are a single instance across the entire cluster. As daemons can be run in any node, the master node is the node which runs the daemon. It has an advantage as there is no device memory wastage because meta-data is not stored across multiple nodes and also upon creation or change of file, it is not required to broadcast it to other nodes [6].

Figure 15 shows the data flow diagram for the read operation. The overall transmission cost during read operation will vary across nodes based on the different location of fragment and that of the reader. The different steps are summarized as follows [6].

1. User issues read request for file blocks of definite length have a certain byte offset.
2. MDFS client seeks information from Name Server to return all blocks of the file.
3. The data retrieval request from the data server is issues for each block.
4. Data Server returns the block.
5. Data server will then request Fragment mapper to furnish information regarding the key and file fragments, which is then returned.
6. Data Server will then request the communication server to fetch the different fragments from locations which are previously returned by Fragment Mapper.

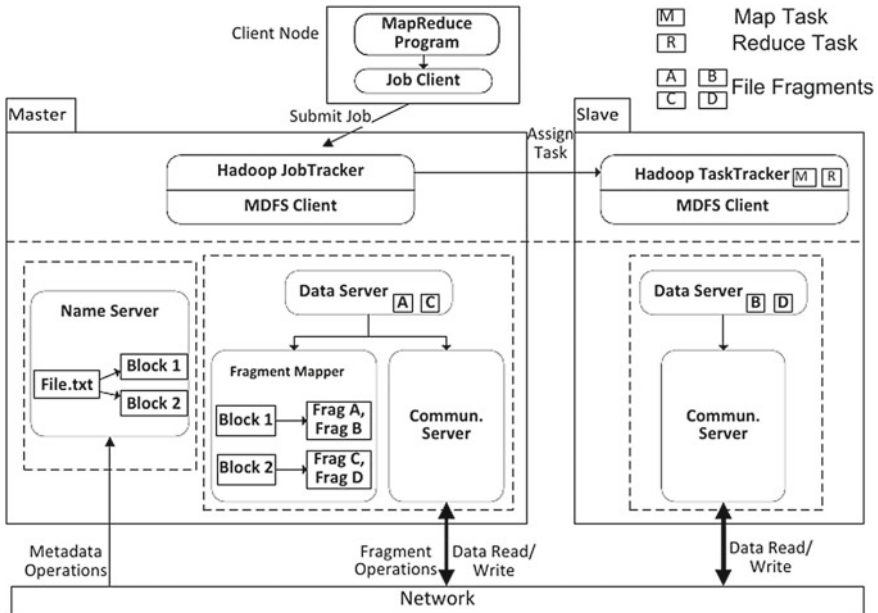


Fig. 14 Architecture of centralized MD FS

The fragments are fetched and stored in local file system of the client who has requested.

7. All the above operations are repeated for fetching the key fragments. Secret key will be created from key fragments.
8. Upon downloading completion of the file fragments, they are decoded and then decrypted to get original block.
9. Key and file fragments are deleted for security purposes.
10. Data Server will then acknowledge the client with the location of the block in the local file system.
11. MDFS client reads the requested number of bytes of the block and Steps 4 to 10 are repeated in case there are many other blocks to read. Once read operation is completed, the block is
12. deleted and original cluster state is restored.

Figure 16 shows the data flow diagram for a write operation. The HDFS write is not possible for MD FS unless the block is decrypted and decoded. The different steps are summarized as follows [6].

1. User issues a write to file request.
2. MD FS client will then request the Name Server for a new block Id.
3. Based on allocation algorithm, the Name Server returns a new block id.
4. MD FS client then issues a creation request to Data Server which then helps in block creation.

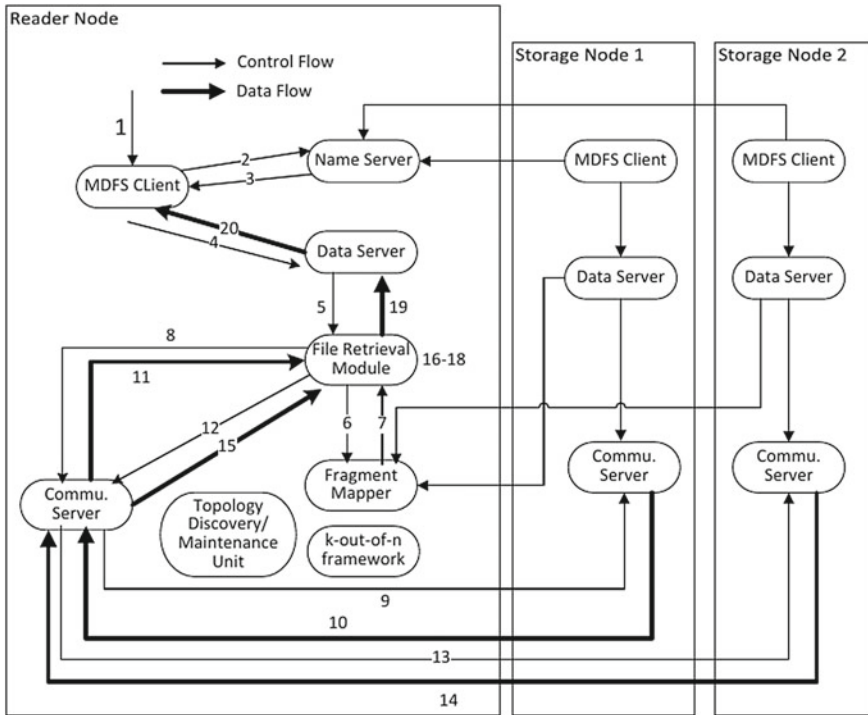


Fig. 15 Data flow diagram for read operation

5. Encryption using secret key takes place for the block in the local file system.
6. Using Shamir’s secret key sharing algorithm, key is split.
7. K-out-of-n framework provides storage nodes.
8. Data Server makes a request to the Fragment mapper to add fragment information.
9. After file and key fragments are distributed in the cluster, Data Server will inform client about the file written successful status. The local file system of the writer is delete after write operation for security purposes.

3 Future Research Direction

There are several areas of research that enthusiastic researchers should delve into in the area of Mobile cloud sensing to make our planet *smart* and *intelligent*. The open research issues are discussed as follows, which will serve as a direction to the future researchers.

- Mobile cloud sensing has immense potential of research. If we take into account the scarce resources present in the mobile devices, there arises a need to be able

to optimize their usage in an efficient manner. This is a very intelligent area and lot of research is currently being pursued.

- Handling large amounts of data which is generated by the mobile sensing data is an area of research and a tough problem to solve.
- Also, 5G as a Infrastructure for Mobile Cloud Sensing is being used in research projects as it supports a peak download speed of 3.6 Gbps as compared to 100 Mbps in 4G. Also, we need to enhance capacity of data movement between the cloud and the mobile device which is expected to rise to 16 Exabytes by 2018 (ten times of the current data volume).
- BDSMC emphasizes on the real time offloading of code and/or data to the cloud through the mobile and Internet network and also real time configuration of the cloud. This area can be dealt with in detail as a research topic.
- As there is a need to provide real-time support of computation in BDSMC applications, the management systems are specially designed for the same. However, BDSMC does not provide a utomatic and dynamic adaptation to fluctuations of time of the input streams processing, and this problem needs to be researched.

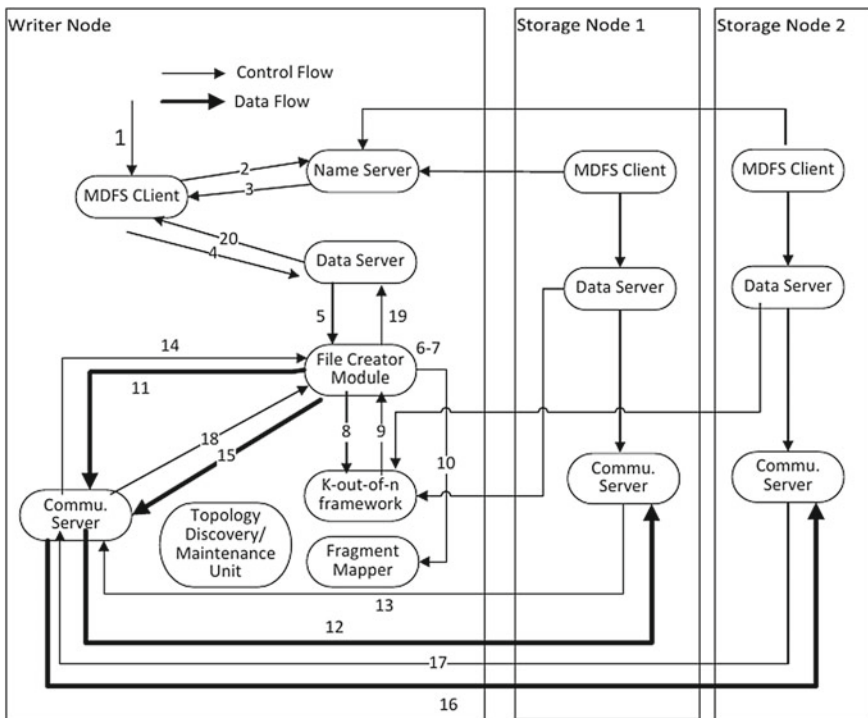


Fig. 16 BDData flow diagram for write operation

4 Conclusion

In this chapter, we have dealt with various techniques related to energy management of mobile devices using MCC like Phone2Cloud, eTime, DREAM and ThinkAir. Also, energy management techniques of MCC using big data have been discussed like BDSMC, StreamCloud and Mobile Cloud Sensing. Besides this, various flavors of MDFS architecture have been discussed as well. There are various limitations discussed here about the various architectures. These present architectures will find it difficult to handle huge volumes of data which are sent to the system at real time. Also, 5G is becoming popular and is necessary for faster data movement during offloading process. The research direction is to delve into the area of Mobile cloud sensing for handling large amount of data and to use 5G as Infrastructure.

References

1. <http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/>. Accessed 30 Dec 2016. Accessed 30 Dec 2016
2. <http://www.bbc.com/news/business-26383058>. Accessed 30 Dec 2016. Accessed 30 Dec 2016
3. Panigrahi, C.R., Sarkar, J.L., Pati, B., Das, H.: S2S: a novel approach for source to sink node communication in wireless sensor networks. In: Proceedings of 3rd International Conference on Mining Intelligence and Knowledge Exploration, pp. 406–414 (2015)
4. Kim, Y., Atchley, S., Valle, G.R., Lee, S., Shipman, G.M.: Optimizing end-to-end big data transfers over terabits network infrastructure. *IEEE Trans. Parallel Distrib. Syst.* **28**(1), 188–201 (2017)
5. Panigrahi, C.R., Pati, B., Tiwary, M., Sarkar, J.L.: EEOA: improving energy efficiency of mobile cloudlets using efficient Offloading Approach. In: Proceedings of IEEE International Conference on Advanced Networks and Telecommunications Systems, pp. 1–6 (2015)
6. George, J., Chen, C.-A., Stoleru, R., Xie, G.G.: Hadoop MapReduce for mobile clouds. *IEEE Trans. Cloud Comput.* **3**(1), 1–14 (2014)
7. Bowen, Z., Dastjerdi, A.V., Calheiros, R.N., Srirama, S.N., Buyya, R.: A context sensitive offloading scheme for mobile cloud computing service. In: Proceedings of the IEEE 8th International Conference on Cloud Computing, pp. 869–876 (2015)
8. Essa, Y.M., Attiya, G., El-Sayed, A.: Mobile agent based new framework for improving big data analysis. In: Proceedings of International Conference on Cloud Computing and Big Data, pp. 381–386 (2014)
9. Rong, P., Pedram, M.: Extending the lifetime of a network of battery powered mobile devices by remote processing: a Markovian decision based approach. In: Proceedings of 2003 Annual Design Automation Conference, pp. 906–911 (2013)
10. <https://dupress.deloitte.com/dup-us-en/focus/tech-trends/2015/tech-trends-2015-what-is-api-economy.html>. Accessed 31 Dec 2016. Accessed 31 Dec 2016
11. Han, Q., Liang, S., Zhang, H.: Mobile cloud sensing, big data, and 5G networks make an intelligent and smart world. *IEEE Network* **29**(2), 40–45 (2015)
12. Kosta, S., Aucinas, A., Hui, P., Mortier, R., Zhang, X.: Thinkair: dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In: Proceedings of 31st IEEE International Conference on Computer Communications, pp. 945–95 (2012)
13. Shu, P., Liu, F., Jin, H., Chen, M., Wen, F., Qu, y., and Li, b.: ETime: Energy-efficient transmission between cloud and mobile devices. *IEEE Infocom*, pp. 195–199 (2013)
14. Tawalbeh, L.A., Mehmood, R., Benkhelifa, E., Song, H.: Mobile cloud computing model and big data analysis for healthcare applications. *IEEE Access* **4**, 6171–6180 (2016)

15. Baccarelli, E., Cordeschi, N., Mei, A., Panella, M., Shojafar, M., Stefa, J.: Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study. *IEEE Netw.* **30**(2), 54–61 (2016)
16. https://en.wikipedia.org/wiki/Mobile_cloud_computing. Accessed 31 Dec 2016
17. Xia, F., DingJie, F., Xi, J., Kong, X., Yang, L.T., Ma, J.: Phone2Cloud: exploiting computation offloading for energy saving on smartphones in mobile cloud computing. *Inf. Syst. Front.* **16**(1), 95–111 (2014)
18. Kwak, J., Kim, Y., Lee, J., Chong, S.: DREAM: dynamic resource and task allocation for energy minimization in mobile cloud systems. *IEEE J. Sel. Areas Commun.* **33**(12), 2510–2523 (2015)
19. https://en.wikipedia.org/wiki/Big_data. Accessed 31 Dec 2016. Accessed 31 Dec 2016
20. <http://www.dataintensity.com/characteristics-of-big-data-part-one/>. Accessed 31 Dec 2016. Accessed 31 Dec 2016