Energy-Efficient Big Data Analytics in Datacenters

Farhad Mehdipour[1], Hamid Noori[2], Bahman Javadi[3]

[1] Graduate School of Information Science and Electronics Engineering, Kyushu University, Japan

E-mail: farhad@ejust.kyushu-u.ac.jp

[2] Engineering Department, Ferdowsi University of Mashhad, Iran

E-mail: hnoori@um.ac.ir

[3] School of Computing, Engineering and Mathematics, University of Western Sydney, Australia

E-mail: b.javadi@uws.edu.au

Abstract–The volume of generated data increases by the rapid growth of Internet of Things (IoT), leading to the big data proliferation and more opportunities for data centers. Highly virtualized cloud-based datacenters are currently considered for big data analytics. However big data requires datacenters with promoted infrastructure capable of undertaking more responsibilities for handling and analyzing data. Also, as the scale of the datacenter is increasingly expanding, minimizing energy consumption and operational cost is a vital concern. Future datacenters infrastructure including interconnection network, storage and servers should be able to handle big data applications in an energy-efficient way. In this chapter, we aim to explore different aspects of could-based datacenters for big data analytics. First the datacenter architecture including computing and networking technologies as well as data centers for cloud-based services will be illustrated. Then the concept of big data, cloud computing and some of the existing cloud-based datacenter platforms including tools for big data analytics will be introduced. We later discuss the techniques for improving energy efficiency in the cloud-based datacenters for big data analytics. Finally, the current and future trends for datacenters in particular with respect to energy consumption to support big data analytics will be discussed.

Keywords: Datacenter; Big data analytics; Cloud computing; Virtual machine; Interconnection networks; Energy and power optimization.

Acronyms:

Acronym	Definition	Acronym	Definition
AIS	All-In Strategy	**MMF**	Multi-Mode Fibers
APU	Accelerated Processing Unit	**MPI**	Message Passing Interface
BDAS	Berkeley Data Analytics Stack	**$MtCO_2e$**	Metric tons of Carbon Dioxide equivalent
BPaaS	Business Process as-a-Service	**MW**	Mega Watt
CEP	Complex Event Processing	**NoSQL**	Not only SQL
CPU	Central Processing Unit	**NSM**	n-ary Storage Model
CS	Covering Set	**O/E**	Optical-to-Electrical
DAaaS	Data Analytic-as-a-Service	**OFDM**	Orthogonal Frequency-Division Multiplexing
DAG	Directed Acyclic Graph	**OS**	Operating System
DBMS	Database Management System	**PaaS**	Platform as-a-Service
DCN	Datacenter Network	**PB**	Petabytes
DFS	Distributed File System	**PB/s**	Petabytes per second
DVFS	Dynamic Voltage and Frequency Scaling	**PFLOPS**	Peta Floating-Point Operations Per Second
E/O	Electrical-to-Optical	**QoS**	Quality of Service
FPGA	Field-Programmable Gate Array	**RDBMS**	Relational Database Management System
Gbps	Gigabits per second	**SaaS**	Software as-a-Service
GFS	Google File System	**SAN**	Storage Area Network
GPU	Graphics Processing Unit	**SIMD**	Single-Instruction Multiple-Data
HDFS	Hadoop Distributed File System	**SQL**	Structured Query Language
HPC	High-Performance Computing	**Tbps**	Terabits per second (Tbps)
HVAC	Heating-Ventilation and Air-Conditioning	**TCP**	Transmission Control Protocol
IaaS	Infrastructure as-a-Service	**ToR**	Top-of-the-Rack
IO (I/O)	Input/Output	**UPS**	Uninterruptible Power Supply
IoT	Internet of Tings	**VM**	Virtual Machine
IT	Information Technology	**VMM**	Virtual Machine Monitor
KaaS	Knowledge as-a-Service	**WDM**	Wavelength Division Multiplexing
kWh	kilo-Watt hour		

1. Introduction

Due to the latest advances in information technology the volume of generated data further increases by the rapid growth of cloud computing and the Internet of Things (IoT). Widely distributed sensors in an IoT collect and transmit data that should be integrated, analyzed and stored. Such data referred as "Big Data", far surpasses the available computing capacity in quantity, heterogeneity, and speed. The emergence of big data brings excellent development opportunities to datacenters that are rapidly evolving. There is also a great attention to cloud services, virtualization solutions, and high-performance computing to boost service velocity and business agility, support big data analytics, and improve datacenter economics. Conventional computing systems are giving way to highly virtualized environments that reshape datacenter traffic flows and dramatically affect datacenter network designs [61].

Big data requires promotion of datacenter infrastructure in both hardware and software aspects. With the continued growth of the volumes of structured and unstructured data, the data processing and computing capacities of the datacenter shall be greatly enhanced. Although current datacenters provide hardware facilities and storage for data, it is crucial for them to undertake more responsibilities, such as acquiring, analyzing, and organizing the data [24]. Also, as the scale of the datacenter is increasingly expanding, it is also an important issue on how to reduce the operational cost for the development of datacenters.

Power consumption is a major concern in the design and development of modern datacenters. Although the performance per watt ratio has been constantly rising, the total power consumed by computing systems is hardly decreasing [1]. It has been shown that in 2007, the worldwide demand for electricity was around 330 Billion kWh from datacenters [31]. This amount is similar to the electricity consumption in the whole UK. Given the demand for more datacenters and emergent of complex big data processing, it is estimated that the electricity consumption will be triple by 2020 (more than 1000 Billion kWh). This also has a major impact on the environment by greenhouse gas emissions. By 2020, the carbon footprint will be about 275 Metric tons of Carbon Dioxide equivalent (*$MtCO_2e$*) that is more than the double of what we had in 2007 [123]. Therefore, novel power-efficient technologies must be developed to minimize the power consumption in datacenters, especially to handle big data analytics.

In this chapter, we aim to explore different aspects of could-based datacenters for big data analytics. In Section 2, datacenter architecture including computing and networking technologies as well as data centers for cloud-based services will be described. In section 3, the concept of big data, cloud computing some of the existing cloud-based datacenter platforms and tools for big data analytics will be introduced. Section 4 discusses the techniques for improving energy efficiency in the cloud-based datacenters for big data analytic. Section 5 highlights the current and future trends for data centers in particular with respect to energy consumption to support big data analytics. Section 6 summarizes this chapter.

2. Datacenter and Cloud Computing

A datacenter is usually known as the infrastructure used by enterprises and is designed to host computing and networking systems and components for IT service demands. A datacenter typically involves storing, processing and serving large amounts of mission-critical data to clients in client/server architecture. Redundant or backup power supplies, cooling systems, redundant networking connections and security systems for running the enterprise's core applications are extensively needed in datacenters. Datacenter management requires the high reliability of both the connections to the datacenter as well as the information stored within the datacenter's storage. Scheduling of application workloads on the available compute resources cost-effectively is another issue that should be addressed in datacenters. Datacenter size in terms of energy consumption vary widely, ranging from 1 MW to more than 30 MW [11].

2.1. Datacenter Architecture

The design of a datacenter is often divided into four categories as “Tier I–IV”. The 4-tier classification loosely based on the power distribution, uninterruptible power supply (UPS), cooling delivery and redundancy of the datacenter. Tier I datacenters have a single path for power distribution, UPS, and cooling distribution, without redundant components, while Tier II adds redundant components to enhance availability. Tier III datacenters have one active and one alternate distribution path for utilities that provide redundancy even during maintenance. Tier IV datacenters have two simultaneously active power and cooling distribution paths, redundant components in each path and are supposed to tolerate any single equipment failure without impacting the load [11].

The high-level block diagram of a typical datacenter is shown in Fig. 1. A datacenter consists of multiple racks hosting the servers connected through a Top-of-the-Rack Switch (ToR) which are further interconnected through aggregate switches in a tree topology. The datacenter network (DCN) interconnects a massive number of servers and provides routing service for the data flowing through the network among computing elements. The tree architecture is a well-known and common interconnection scheme in the current datacenters. However, it suffers from the problems of low scalability, high cost as well as a single point of failure. An interconnection network should be modular with a fast reconfiguration capability, and a high capacity and scalability [41][74]. In the current datacenters, the network is usually a canonical fat-tree Tier II or Tier III architecture (Fig. 1) [62][71]. In the Tier III topologies (shown in the figure) one more level is applied in which the aggregate switches are connected in a fat-tree topology using the core switches. When a user issues a request, a packet is forwarded through the Internet to the front end of the datacenter. In the front end, the content switches and the load balance devices are used to route the request to the appropriate server. Most of the current datacenters are based on commodity switches for the interconnection network. The topology of the network interconnecting the servers significantly affects the agility and reconfigurability of the datacenter infrastructure to respond to changing application demands and service requirements [79].

There are alternate approaches to scalable, cost-effective network architectures which can be grouped into fixed topology and flexible topology networks. Fixed topology networks can be further categorized into two topologies: tree-based topologies such as fat-tree [1][79] and Clos Network [49], and recursive topologies such as DCell [52] and BCube [51]. Flexible topologies have the ability to adapt their network topology based on the traffic demand, at run time. Examples of such topologies include c-Through [124], Helios [41] and OSA [28]. Every datacenter network has its unique approaches for network topology, routing algorithms, fault-tolerance, and fault recovery [79].

Traditional datacenter architectures including networking and storage resources support the needs of specific client-server applications. The capability of existing datacenter for data processing is limited to compute and storage infrastructures within a local area network, e.g., a single cluster within a datacenter. Further, the conventional datacenters built for client-server computing lack in essential capabilities to meet the requirements of today’s highly virtualized compute environments; they cannot meet the performance and availability demands; make inefficient use of network resources; do not scale in a linear fashion; and are not well suited for the high-bandwidth, low-latency server-to-server traffic flows that is common in current datacenters [61].

However, advances such as powerful multicore processor-based servers, virtualization, cloud and distributed computing are transforming the datacenters to incorporate new features. Today's datacenters contain thousands of switches and servers, run data-intensive applications from cloud services such as search, web email, to infrastructural computations such as MapReduce [35]. Many IT organizations are consolidating compute, network, and storage resources and employing virtualization solutions and cloud-based services based on new networking models.

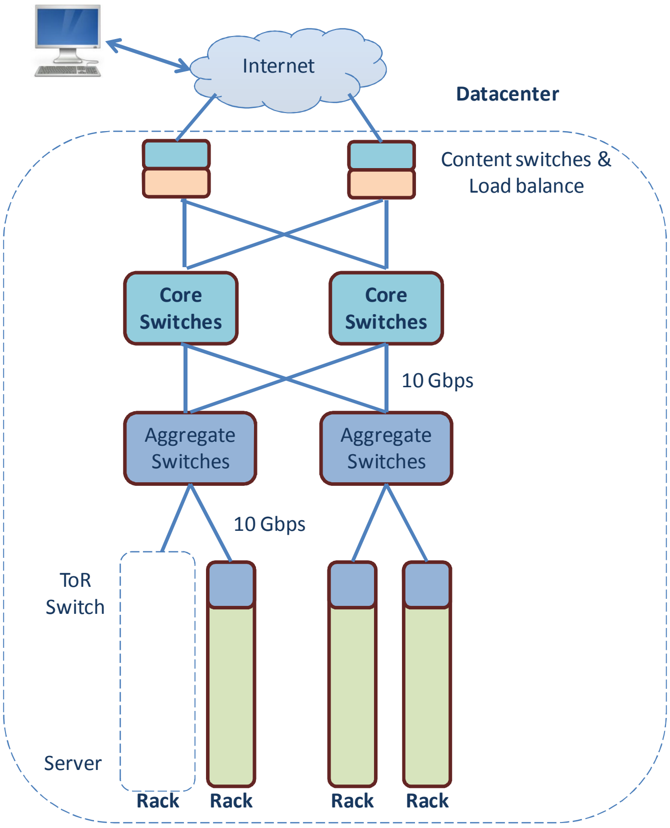

Fig. 1 A datacenter architecture [71]

2.2. Cloud Computing

Cloud computing is the integration of computing and data infrastructures to provide a scalable, agile and cost-effective approach to supporting the ever-growing critical IT needs [74]. Cloud computing relieves its users from the burdens of provisioning and managing their datacenters and allows them to pay for resources only when needed (i.e. pay-as-you-go). It provides more profit to both service providers who capitalize poorly utilized resources and users who only need to pay per their use. Furthermore, increasing the resource utilization results in less energy consumption and carbon footprint that can increase the economical profit of the cloud for providers [1].

Virtualization in Cloud Systems: Physical resources can be divided into a number of logical slices called virtual machines (VMs). Each VM can accommodate an individual operating system (OS) creating for the user a view of a dedicated physical resource and ensuring the performance and failure isolation between VMs that are sharing a single physical machine. The virtualization layer lies between the hardware and OS and, therefore, a virtual machine monitor (VMM) takes the control over resource sharing/multiplexing and has to be involved in the system's power management [1]. The virtualization technology provides the ability to encapsulate the workload in VMs and consolidate them to a single physical server. The consolidation has become especially effective after the adoption of multi-core CPUs in computing environments, as many VMs can be allocated to a single physical node leading to the improved utilization of resources and reduced energy consumption compared to a multi-node setup [1][60].

2.3. Datacenters for Cloud Computing Services

Datacenter infrastructure has been receiving significant interest due to their highlighted role in supporting the rapidly growing cloud-based applications. Datacenters including the computing, storage, and communication resources form the core of the support infrastructures for cloud computing. With the proliferation of cloud computing and cloud-based services, datacenters are becoming increasingly large with massive number of servers and a huge amount of storage. Server and even network virtualization are increasingly employed to make the datacenter flexible and adapt to varying demands [74].

Multi-Cloud Systems: Datacenters based on the interconnection of multiple-cloud systems provide an opportunity to improve overall QoSs parameters by deploying and executing applications over a large cluster of resources [38]. Such datacenters are heterogeneous, distributed, and highly uncertain. The system becomes more complicated where mobile devices are considered as cloud nodes as well. The availability, performance, and state of resources, applications and data may continuously change in a multi-cloud datacenter. Uncertainty is a fact in multi-cloud datacenter environments due to several reasons such as security attacks, network and resource failures, incomplete global knowledge among competing applications leading to non-optimized use of the resources and network bandwidth, etc. The current data-intensive or big data application programming frameworks such as MapReduce and workflow models are inadequate to handle these issues [34].

3. Handling Big Data on Cloud-Based Datacenters

Nowadays the data has reached in different directions in terms of size, type, and speed and has received wide attention as “Big Data”. The huge amount of data coming from various sources with different types and in high speeds needs appropriate infrastructure for efficient processing and knowledge discovery. Currently, datacenters incorporate new features such as virtualized computing in the cloud systems that make them a candidate platform for processing the big data.

3.1. Big Data

Big data refers to the large amounts (volume) of heterogeneous data (variety) that flows continuously (velocity) within data-centric applications. We can classify big data requirements based on its five main characteristics [64]:

- **Volume:** This is the primary characteristic of big data, refers to the large size (Tera or Petabytes) of records, transactions, tables, files, video, web text, and sensor logs, etc. As the size of data to be processed is large, it needs to be broken into manageable chunks. Data needs to be processed simultaneously across multiple systems and several program modules.
- **Variety:** The big data comes from a great variety of sources. Data of different formats, types and structures needs to be processed using efficient solutions.
- **Velocity:** It refers to the frequency of data generation or the frequency of data delivery. The stream of data acquired via multiple sources needs to be processed at a reasonable speed.
- **Ambiguity:** Big data is ambiguous by nature due to the lack of relevant metadata and context in

many cases. For example, "M" and "F" in a data set can mean, Monday and Friday, male and female, or mother and father, respectively.

- **Complexity:** Big data complexity needs to use many algorithms to process data quickly and efficiently. Several types of data need multi-step processing, and the scalability is extremely important as well. Processing large-scale data requires an extremely high-performance computing environment that can be easily managed and its performance can be tuned with linear scalability.

Big data analytics describes the process of performing complex analytical tasks on data that typically includes grouping, aggregation, or iterative processes. The increase in the amount of data raises several issues for analysis software: 1) The amount of data is increasing continuously at a high speed, yet data should be up-to-date for analysis tasks, 2) The response time of a query grows with the amount of data.

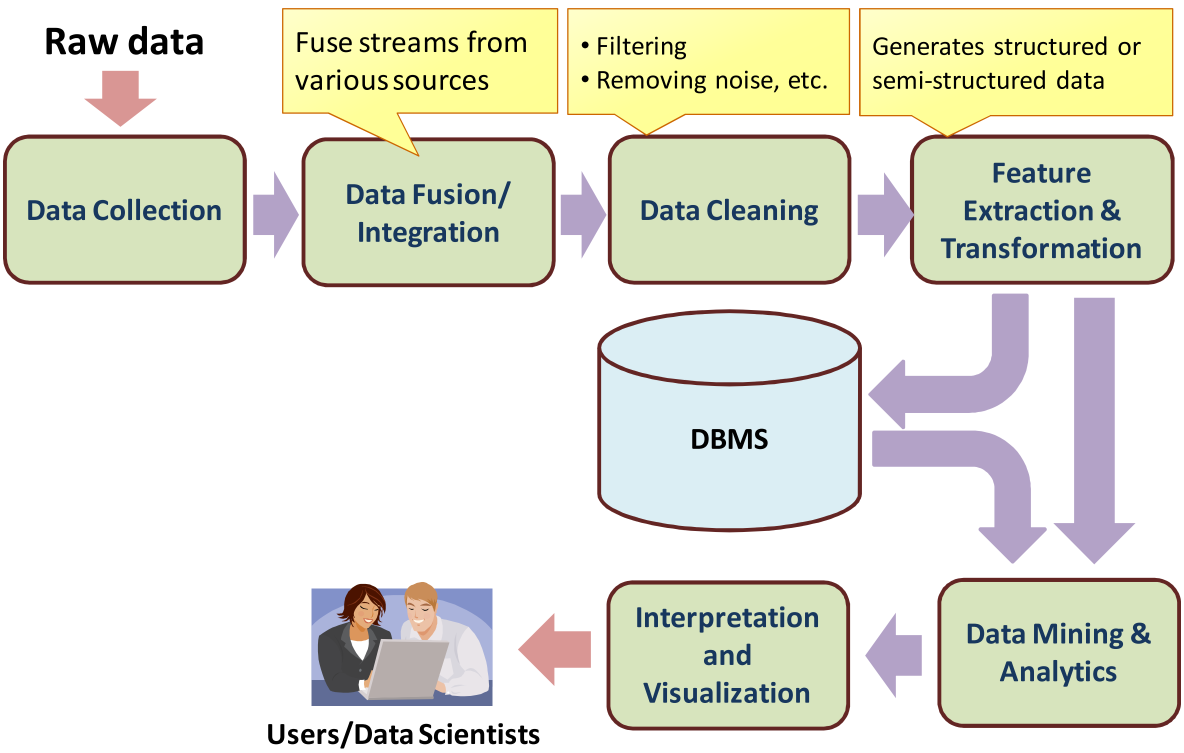

Fig. 2 The flow of big data processing

At the same time, latencies must be reduced to provide actionable intelligence at the right time, 3) Analysis tasks need to produce query results on large data sets in an adequate amount of time [106].

Fig. 2 shows a typical flow for the big data processing. The first step after the acquisition of data is to perform fusion/integration of the data coming from multiple sources. Data cleaning is the next step that may consume large processing time though it may significantly reduce the data size that leads to less time and effort needed for data analytics. The raw data is normally unstructured that neither has a pre-defined data model nor is organized in a pre-defined manner. Thus, the data is transformed to semi-structured or structured data in the next step of the flow [97].

The main phase of big data processing is to perform discovery of data, which is where the complexity of processing data lies. A unique characteristic of big data is the manner in which the value is discovered. It differs from conventional business intelligence, where the simple summing of known values generates a result. The data analytics is performed through visualizations, interactive knowledge-based queries, or machine learning algorithms that can discover knowledge [31]. Due to the variant nature of the data there may not be a single solution for the data analytics problem so, the algorithm may be short-lived.

3.2. Big Data and Cloud Computing

As aforementioned, cloud computing allows access to information and computer resources from anywhere that a network connection is available. It supplies large networks of virtualized services: hardware resources (CPU, storage, and network) and software resources (e.g., databases, load balancers). The datacenter cloud facilitates virtual centralization of application, computing, and data. Nowadays, shifting the big data applications from the physical infrastructure into the virtualized datacenters in computational clouds is a major trend [34]. While cloud computing optimizes the use of resources, it does not provide an effective solution hosting big data applications yet. Gartner’s Inc. Hype Cycle states that cloud computing and big data are the fastest growing technologies that dominate the research world for the next decade [105].

Fig. 3 shows the high-level components that are building blocks of the next-generation cloud-based datacenters supporting big data [72][97]. The lowest layer represents various forms of the data generated by different sources. The next layer interfaces data stream moving toward datacenter cloud system. It may involve real-time processing of data and technologies such as Complex Event Processing (CEP [[102]]) as well. The next tier represents the technologies that will be used in integrating, cleaning and transforming the data to multiple types and sources of data. Storage and distributed data management technologies such as NoSQL for unstructured and semi-structured data and the relational database management system (RDBMS) for structured data are represented in the next tier. The topmost layer indicates the analytical layer that will be used to drive the needs of the big data analytics.

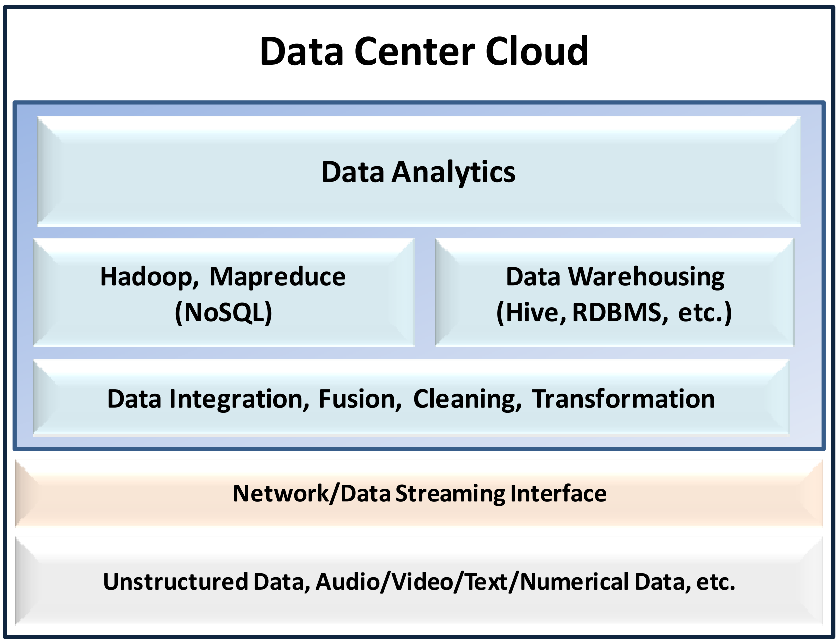

Fig. 3 Components of next generation cloud-based datacenter

The cloud computing system comprises of three service models: Infrastructure as-a-Service, Platform as-a-Service, and Software as-a-Service. A classification of big data functionalities against cloud service models is illustrated in Fig. 4 [31]. At the lowest level, current hardware infrastructure is virtualized with cloud technologies, and the hardware infrastructure, as well as platforms, will be provided as services. In the upper layer software as a service and, on top of it, business processes as a service can be built. Big data can be defined as a *Data Analytic as-a-Service* (DAaaS), an additional cloud layer located between Platform as-a-Service (PaaS) and Software as-a-Service (SaaS) layers. Some of the data analytics functionalities can be implemented through existing tools in SaaS and can be used with minor customizations, or through the software developed in PaaS. Furthermore, most of the cloud services offering big data capabilities can be classified as Infrastructure as-a-Service (IaaS), with minor exceptions. It is because different platforms tend to cover all aspects of big data from data capturing to storage and analysis, and they include the underlying infrastructure needed to handle all the required capabilities, particularly thos of the storage.

Despite rather remarkable number of cloud-based big data solutions that have been released on the market (e.g. OpenCrowd [100], Infochimps Platform [66], Opani [100] and PiCloud [102], Oracle big data appliance [95][34], and IBM Infosphere BigInsights [64]), many of them fall shortly of big data challenge. It is therefore required to provide computing infrastructures such as specialized datacenters, software frameworks, data processing technologies and distributed storage system specifically optimized for big data applications.

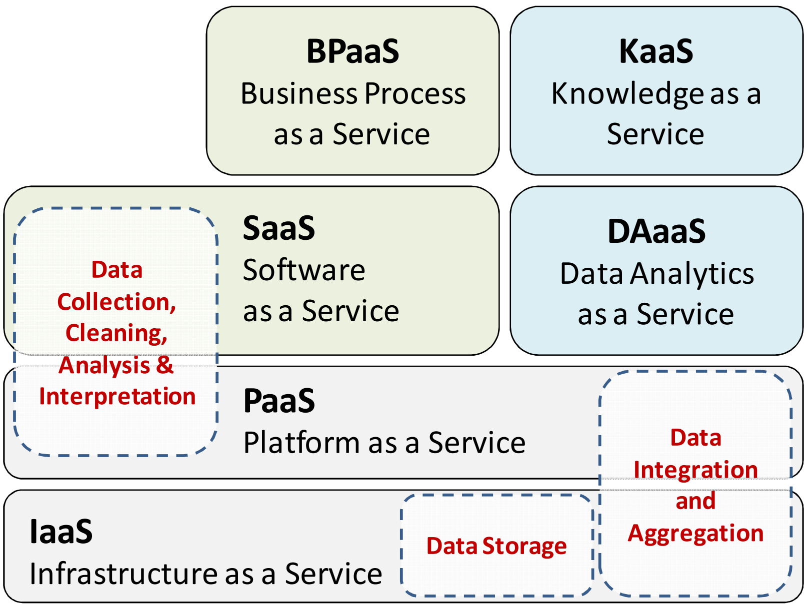

Fig. 4 Big data functionalities in the Cloud service model [5]

3.3. Big Data Analytics on Datacenters

The major driver for modern datacenters is analysing of large-scale and massive data sets [11]. So datacenters need to have adequate infrastructure such as interconnection networks, storage and servers to handle big data applications. While these resources are currently used in datacenters, they should be evaluated for handling a new type of data-centric applications. In the following, these resources including power consumption as a main concern for future datacenters will be reviewed.

3.3.1. Interconnection Networks

The interconnection network in datacenters is the core for supporting big data and is the most important infrastructure that needs urgent attention. Big data applications transfer large-scale data to datacenters for processing and analysis. So we need to consider two types of interconnection: Inter-Datacenter connection and Intra-Datacenter connection.

(i) Inter-Datacenter connection: This is the connection from the outside world to the datacenter, which is normally based on the existing network infrastructure (i.e., Internet). The modern physical infrastructure for most of the countries is the high-bandwidth fiber optic interconnections that can handle the rapid growth of data size. The common architecture in these interconnections is IP-based wavelength division multiplexing (WDM), which is based on multiple optical carriers multiplexing in different wavelength [41]. In this technology, each optical fiber can carry several signals with a different wavelength and, therefore, increase the network bandwidth. While most of the WDM-based networks have deployed with the bandwidth of 40Gbps, recently a new standard for 100Gbps is introduced to address the demand for the high-speed Inter-datacenter connection [8].

Since WDM technology is limited by the bandwidth of the electronic bottleneck, a new technology called orthogonal frequency-division multiplexing (OFDM) is introduced. OFDM is a multi-carrier parallel transmission technology, which divides the high-speed data flow into low-speed sub-data flow to transmit them over multiple orthogonal sub-carriers [7]. This makes OFDM more flexible and efficient compared with the WDM technology and more promising for Inter-datacenter connection that is dealing with big data transferring.

(ii) Intra-Datacenter connection: Intra-Datacenter connection is used to transfer the data within a datacenter. As mentioned earlier, most of the current datacenters consist of high-performance servers interconnected with commodity switches in a multi-level fat-tree topology [11]. While servers in each rack are connected through 1Gpbs links, all the rack servers are interconnected through a set of 10Gbps switches. Finally, these switches will be connected through a set of core switches with 100Gbps links (using a bundle of 10Gbps links). Although this architecture is scalable and fault-tolerant, power consumption is the main issue. The other issue in this architecture is the high-latency due to multiple store-and-forward in the switches hierarchy. For big data applications, we need a network technology for Intra-datacenter connection, which can provide high throughput, low latency, and low energy consumption.

To solve these issues, *optical networks* recently have received great attention as an alternative for interconnection networks in datacenters to address the demands of big data analytics. Currently, optical networks are only used for point-to-point communications in datacenters where the links are based on low-cost multimode fibers (MMF) for short-reach communications [71]. The desired future architecture for intra-datacenter interconnection will be all-optical connections with switching in the optical domain as shown in Fig. 5. In this technology electrical-to-optical (E/O) and optical-to-electrical (O/E) transceivers will be completely removed, which are the main source of power consumption in the current architecture. Therefore, all-optical interconnection networks can provide bandwidth in Terabits per second (Tbps) scale with low power consumption for future datacenter networks.

3.3.2. Storage

An efficient storage mechanism for big data is an essential part of the modern datacenters. The main requirement for big data storage is file systems that is the foundation for applications in higher levels. The Google File System (GFS) is a distributed file system for data-centric applications with robustness, scalability and reliability [41]. GFS can be implemented in commodity servers to support large-scale file applications with high performance and high reliability. Colossus is the next generation of GFS with better reliability that can handle small files with higher performance [88].

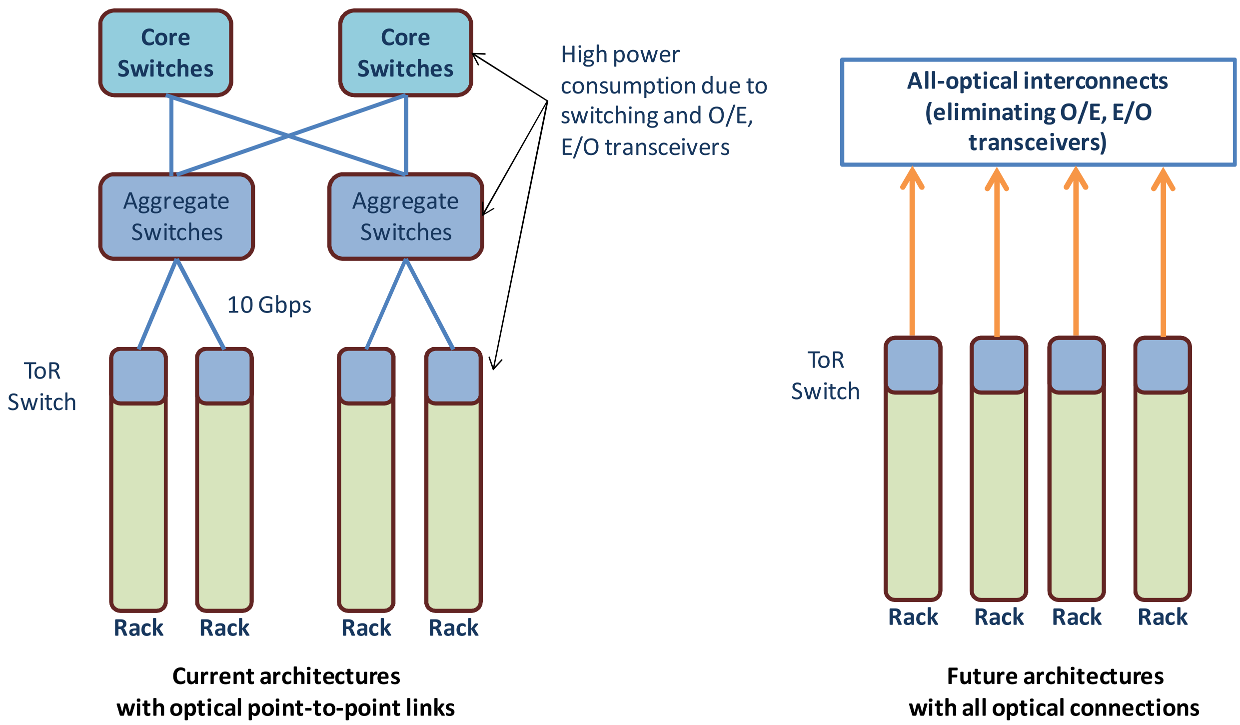

Fig. 5 Point-to-point vs. all-optical interconnections [53]

Hadoop Distributed File System (HDFS) is another file system used by MapReduce model where data is placed more closely to where it is processed [63]. HDFS uses partitioning and replication to increase the fault tolerance and performance of large-scale data set processing. Another file system for storing a large amount of data is Haystack [13] which is used by Facebook for handling a lot of photos storing in this website. This storage system has a very low overhead that minimizes the image retrieval time for users. The above-mentioned file systems are the results of many years research and practice so can be utilized for big data storage.

The second component in big data storage is a database management system. Although database technology has been advancing for more than 30 years, they are not able to meet the requirements for big data. Non-traditional relational databases (NoSQL) are a possible solution for big data storage, which are widely used recently. In the following, we review the existing database solutions for big data storage in three categories: key-value databases, column-oriented databases, and document-oriented databases.

Key-values databases normally have simple data model and data is stored based on key-values. So they have a simple structure with high expandability and performance compared to relational databases. *Column-oriented databases* use columns instead of the row to process and store data. In these databases, both columns and rows will be distributed across multiple nodes to increase expandability.

Table 1. List of databases for big data storage in datacenters

Database	**Type**	**Description**
Amazon: Dynamo [34]	key-value databases	A distributed, high reliable, and scalable database systems used by Amazon for internal applications.
Voldemort [115]	key-value databases	A storage system used in LinkedIn website.
Google: Bigtable [26]	column-oriented databases	A distributed storage system used by several Google produces, which as Google Docs, Google Maps, and Google search engine. Bigtable can handle data storage in the scale of petabytes using thousands of servers.
Cassandra [74]	column-oriented databases	A storage system developed by Facebook to store large-scale structured data across multiple commodity servers. Cassandra is a decentralized database that provide high availability, scalability, and fault-tolerance.
Amazon Simple DB [91]	document-oriented databases	A distributed database designed for structured data storage and provided by Amazon as the web service.

CouchDB [5]	document-oriented databases	Apache CouchDB is document-based storage system where JavaScript is used to query and manipulate the documents.

Document-oriented databases are designed to handle more complex data forms. Since documents can be in any modes (i.e., semi-structured data), so there is no need for mode migration. Table 1 shows the list of big data storages that are classified into three types. These databases are available to handle big data in datacenters and Cloud computing systems.

3.3.3. Processing and Analysis Tools and Techniques

Big data processing is a set of techniques or programming models to access large-scale data to extract useful information for supporting and providing decisions. In the following, we review some tools and techniques, which are available for big data analysis in datacenters.

As mentioned in previous section, big data usually stored in thousands of commodity servers so traditional programming models such as Message Passing Interface (MPI) [113] cannot handle them effectively. Therefore, new parallel programming models are utilized to improve the performance of NoSQL databases in datacenters. MapReduce [35] is one of the most popular programming models for big data processing using large-scale commodity clusters. MapReduce is proposed by Google and developed by Yahoo. Map and Reduce functions are programmed by users to process the big data distributed across multiple heterogeneous nodes. The main advantage of this programming model is simplicity, so users can easily utilize that for big data processing. A certain set of wrappers is being developed for MapReduce. These wrappers can provide a better control over the MapReduce code and aid in the source code development. Apache Pig is an SQL-like environment developed at Yahoo [98] is being used by many organizations like Yahoo, Twitter, AOL, LinkedIn, etc. Hive is another MapReduce wrapper developed by

Facebook [118]. These two wrappers provide a better environment and make the code development simpler since the programmers do not have to deal with the complexities of MapReduce coding.

Hadoop [112][119] is the open-source implementation of MapReduce and is widely used for big data processing. This software is even available through some Cloud providers such as Amazon EMR to create Hadoop clusters to process big data using Amazon EC2 resources [2]. Hadoop adopts the HDFS file system, which is explained in previous section. By using this file system, data will be located close to the processing node to minimize the communication overhead. Windows Azure also uses a MapReduce runtime called Daytone [10], which utilized Azure's Cloud infrastructure as the scalable storage system for data processing.

There are several new implementations of Hadoop to overcome its performance issues such as slowness to load data and the lack of reuse of data [58][59]. For instance, Starfish [58] is a Hadoop-based framework, which aimed to improve the performance of MapReduce jobs using data lifecycle in analytics. It also uses job profiling and workflow optimization to reduce the impact of unbalance data during the job execution. Starfish is a self-tuning system based on user requirements and system workloads without any need from users to configure or change the settings or parameters. Moreover, Starfish's Elastisizer can automate the decision making for creating optimized Hadoop clusters using a mix of simulation and model-based estimation to find the best answers for what-if questions about workload performance.

Spark [132] developed at the University of California at Berkeley, is an alternative to Hadoop, which is designed to overcome the disk I/O limitations and improve the performance of earlier systems. The major feature of Spark that makes it unique is its ability to perform in-memory computations. It allows the data to be cached in memory, thus eliminating the Hadoop's disk overhead limitation for iterative tasks. The Spark developers have also proposed an entire data processing stack called Berkeley Data Analytics Stack (BDAS) [17].

Similarly, there are other proposed techniques for profiling of MapReduce applications to find possible bottlenecks and simulate various scenarios for performance analysis of the modified applications [59]. This trend reveals that using simple Hadoop setup would not be efficient for big data analytics and new tools and techniques to automate provisioning decisions should be designed and developed. This possibly can be a new service (i.e., *Big Data Analytics as-a-Service*) that should be provided by the Cloud providers for automatic big data analytics on datacenters.

In addition to MapReduce, there are other existing programming models that can be used for big data processing in datacenters such as Dryad [68] and Pregel [87]. Dryad is a distributed execution engine to run big data applications in the form of directed acyclic graph (DAG). Operation in the vertexes will be run in clusters where data will be transferred using data channels including documents, TCP connections, and shared memory. Moreover, any type of data can be directly transferred between nodes. While MapReduce only support single input and output set, users can use any number of input and output data in Dryad. Pregel is used by Google to process large-scale graphs for various purposes such as analysis of network graphs and social networking services. Applications are introduced as directed graphs to Pregel where each vertex is modifiable, and user-defined value and edge show the source and destination vertexes.

3.3.4. Platforms for Big Data Analytics

Solutions for the issue of the growing computation power required by big data analytics fall in two different categories. On approach is to scale the current systems and the other one is to look for more suitable and efficient systems as a substitution for current servers. Scaling is the ability of the system to adapt to increased demands for big data processing. Horizontal scaling and vertical scaling are two general approaches.

(*i*) Vertical scaling (also known as scale up) includes empowering machines with more memory and higher performance processors as well as involving specialized hardware such as accelerators. The most popular vertical scale up paradigms are multicore processors, Graphics Processing Unit (GPU), Accelerated Processing Units (APUs), Field-Programmable Gate Array (FPGA) and High-Performance Computing

(HPC) clusters. HPC clusters [21], also called as blades or supercomputers, are machines with thousands of cores. They can have a different variety of disk organization, cache, communication mechanism, etc. depending upon the user requirement. MPI is typically used as the communication scheme for such platforms [113]. Since there are software and programming tools that for easy of vertical scaling (e.g. having multi-core processors, GPUs) easy, most of the software can take advantage of it transparently. However, vertical scaling needs substantial financial investments at one point in time. Furthermore, to cope with future workloads, the system needs to be adequately powerful, and initially, the additional performance goes to waste.

(*ii*) Horizontal scaling includes extending clusters by adding more servers and distributing the work across many servers, which are usually commodity machines. This increases performance in smaller steps, and the financial investment to upgrade is by far smaller. However, to use these large clusters and many servers efficiently, analysis software (parallel programming utilities) on top them has to be developed to handle distribution by itself. Parallel programming uses a divide and conquer approach to breaking down the application into independently processable partitions and send for distributed machines for processing. After finishing the process, the intermediate results have to be merged into a final result. Synchronization among the parallel processors is one of the essentials [106]. Different approaches have evolved for horizontal scaling over the time: parallel systems using an SQL-like interface and massive data analysis systems. Some of the prominent horizontal scale-out platforms include peer-to-peer networks [91][115], Parallel Database Systems, Hyracks [18], Stratosphere [12], since they are samples of infrastructures and tools that support horizontal scaling and provide the facilities to use many machines available in a cluster.

Peer-to-Peer Networks [91] [115] are decentralized and distributed network architecture including millions of connected machines. Each node (machine) in the network serves as well as consumes resources. MPI is the typical communication approach for exchanging the data among nodes. In this structure, scaling out can be unlimited (in case of support of software). The major challenge of these networks is in the communication among different nodes. In case of initiating broadcasting messages in such a network, the messages are sent from a root that is an arbitrary in the form of a spanning tree over the network. The broadcasting messages are cheaper than the aggregation of data/results that are much more expensive. [113].

Parallel DataBase Management Systems such as GRACE [42], Gamma [36] and Teradata [117] are pioneers of parallel relational database systems. Classical database management systems (DBMSs) use an n-ary storage model (NSM) to provide intra record locality. A relation consists of tuples with a defined set of attributes. Gamma [36] is a parallel relational database machine that exploits dataflow query processing techniques. Each operator in a relational query tree is executed by one or more processes. The scheduler places these processes on a combination of processors with and without disk drives. In Gamma's hardware design, associated with each disk drive there is a processor, and the processors are interconnected via an interconnection network. This design provides high disk bandwidth and permits the I/O bandwidth to be expanded incrementally. GRACE [42] is a parallel relational database system adopting data stream-oriented processing. Teradata [117], is a commercial system of that has built a database management system for decision support using a parallel architecture and multiple processors.

Hyracks [18] is a data parallel platform for data-intensive computations that has been developed by the computer science group of the University of California, Irvine. It features a parallel dataflow execution model. Jobs are specified as directed acyclic graphs composed of operators, represented as nodes, and connectors, represented as edges. Hyracks abstracts the MapReduce data model and uses tuples with fields of arbitrary types.

Stratosphere [12] system has been built by a research team comprised of TU Berlin, HU Berlin, and HPI Potsdam since 2008, consists of the PACT programming model and the massively parallel execution engine Nephele. The PACT programming model is a generalization of the MapReduce programming model and operates on a key/value data model.

On the other hand, according to some of current studies, current servers do not match well with the computational requirement of big data analytics. Therefore, they suggest looking for more efficient hardware platforms for future servers such as Micro-servers [6][8][111][83][121][81], Scale-Out Processors [5][42][54][84] and System-Level Integration [79].

4. Energy Efficiency in Datacenters for Big Data Analytics

Today, datacenters and servers consume enormous amounts of energy and, therefore, improving energy consumption in datacenters is crucial. In a typical datacenter, servers, storage, and network devices consume around 40%, 37% and 23% of the total IT power, respectively [49]. Handling big data in datacenters has strong impacts on the power consumption of all three components. In addition, increase of power consumption in datacenters raises the power usage of the HVAC equipment (Heating-Ventilation and Air-Conditioning) to keep the whole system at the working temperature. Therefore, by improving the power efficiency of any component in datacenters, the overall power consumption decreases significantly due to indirect power saving in the HVAC equipment. Reducing power consumption in hardware and network architecture [85][88][1], and smart cooling technologies [102] are effective methods to save energy in datacenters. It has been shown that saving 1.0 Watt power in the IT devices will save around 2.5 Watt in the total power [63]. These evidence indicate that we are in an urgent need for new technologies and techniques to improve the power consumption of datacenters especially in the presence of big data applications.

In a highly-connected datacenter, in practice only a few devices are working at a given time. Therefore, a significant amount of energy can be saved if we can power off those idle devices. Intel Research proposed a proxy architecture that uses a minimal set of servers to support different forms of idle-time behavior for saving energy [95]. In such case, the routing can be adjusted so that data traffic flow through a certain set of devices.

As the network interconnect infrastructure is a significant energy consumer in datacenters, therefore, it is essential to develop intelligent techniques to manage network resources efficiently. Further, optical interconnection networks provide a solution for scalable architectures with optimized energy-efficiency [22].

In the virtualized datacenters, one possible way of reducing energy consumption is to optimize network topologies established between VMs and thus reduce the network communication overhead and the load of network devices steadily. A reduction in the transition overhead caused by switching between different power states and the VM migration overhead can also greatly advance the energy-efficient resource management. [105] proposes consolidation of applications for significant reduction in energy consumption of cloud-based systems. As VMs are increasingly populating the datacenter (42% more per year [53]), by migrating VMs to proper servers the number of active and operating servers can be reduced [123].

In the Cloud federations comprising geographically distributed datacenters, efficient distribution of the workload across geographically distributed datacenters can reduce the costs by dynamically reallocating the workload to a place where the computing resources, energy and/or cooling are cheaper (e.g., solar energy during daytime across different time zones). Moreover, a power-aware OS can monitor the overall system's performance and appropriately apply power management techniques (e.g. dynamic voltage and frequency Scaling- DVFS) to the system components [1].

Regarding the low-level system design, it is important to improve the efficiency of power supplies and develop hardware components supporting the performance scaling proportionally to the power consumption. It is also necessary to develop energy-efficient resource management approaches that leverage multi-core CPUs due to their wide adoption. In this section, we discuss some energy-efficient approaches for big data processing. Fig. 6 shows the general approaches for energy-efficient big data processing. The approaches can be broadly divided into two groups. One group focuses on energy-aware scaling of datacenters including both vertically and horizontally. The second group believes in a new architecture for servers as a replacement for currently used machines for energy enhancement. This is because, according to some studies the architectures of current servers does not match well with the computational requirements of big data processing applications. Using low-power processors (micro-servers), more system-level integration, and a new architecture for servers processors are some of the solutions that have been discussed recently as an energy-efficient replacement for current machines.

4.1. Energy-Efficient Scaled Datacenters

In this section, we discuss some energy-efficient scaling approaches including both horizontally and vertically. In these approaches, while the system is scaled to support computational requirements for big data processing, the efficiency of the system in term of energy is taken into account as well. The energy-efficient horizontally scaled approaches mainly focus on MapReduce programming model and scheduling as well as Distributed File System (DFS) or both. In vertical scaling, FPGAs and GPUs are considered as accelerators for enhancing energy efficiency.

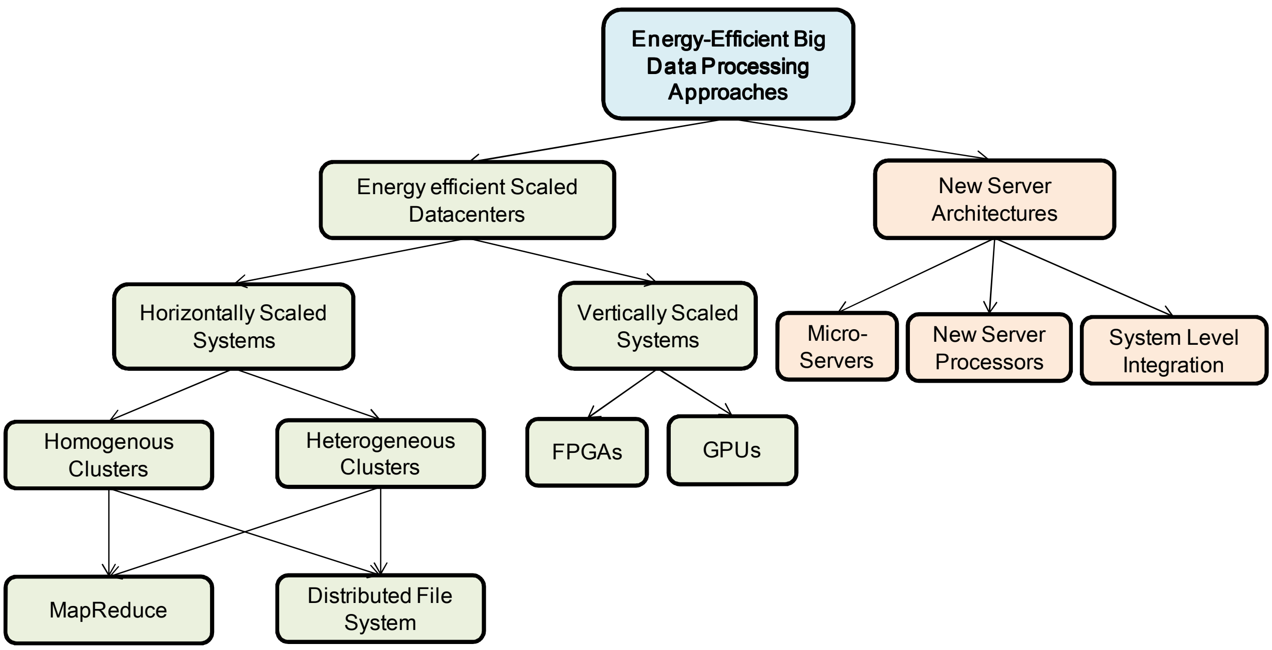

Fig. 6 Classification of various approaches for energy-efficient big-data processing

4.1.1. Horizontally Scaled Systems

MapReduce is used widely by many industries for big data processing through spreading data across datacenters or a large number of clusters. Thus, power management for MapReduce clusters has also become important [119][128]. Unfortunately, the innate features of conventional MapReduce programming model result in energy inefficiency. Examples of these features include [119]:

- Replicating data across multiple machines for fault tolerance,
- Low-average utilization of machines in large datacenters, due to variable distributions of the jobs arrival have,
- Sharing Map-Reduce clusters to run jobs with different behavior (some being CPU‑intensive, and some IO‑intensive), and
- Speculative tasks execution performed by MapReduce to ensure reliability.

Combining workloads on a fewer servers and powering down the idle servers is a common approach to enhance the energy efficiency. However, considering the characteristics of MapReduce, mentioned before, such techniques will not work so effectively because they: [119]

i. affect parallelism and MapReduce performance is proportional to the number of machines due to its parallel nature,

ii. increase latencies because data would have to be fetched from remote locations,

iii. cause data unavailability as data is distributed across servers, and

iv. increase network traffic due to data replication done by NameNode [122].

Therefore, effective energy management techniques should be aware of the underlying distributed file system and programming frameworks otherwise the application performance degrades significantly [119].

Most of the techniques that enhance the energy efficiency in MapReduce environments attempt to reduce the idle periods on nodes with a less number of active nodes in the cluster. To do so, they modify the

MapReduce programming model and DFS to effectively and intelligently 1) merge the jobs, 2) redistribute the data and 3) re-configure the nodes [119]. Tiwari [119] classifies the energy-efficient MapReduce techniques into two MapReduced-based and HDFS-based techniques. Then he categorizes the Map-Reduce programming model modifications for consolidating/distributing workload based on 1) workload characteristics, 2) hardware characteristics [119]. Also, HDFS cluster modification techniques modify HDFS component work by consolidating the data on fewer active nodes so that other nodes can go to sleep state. According to [119], HDFS cluster modification techniques are classified as 1) Replica-binning based data placement and zoning strategies and 2) Temporal-binning based data placement and zoning strategies. The former techniques ensure that at least one copy of every data block is always available. The latter techniques consider the data access patterns based on time and workload to create the zones so that frequently accessed data is always available. Wirtz [128] proposes another classification for energy efficiency techniques for MapReduce including 1) Scheduling, 2) Powering-down nodes, 3) Analyzing how HDFS data is processed and 4) Analyzing system components to address energy conservation.

Our proposed classification for energy efficient MapReduce techniques (shown in Fig. 6) suggests dividing energy-efficient horizontal scaling techniques into two classes of homogeneous clusters versus heterogeneous clusters. Then each of these classes covers energy-efficient techniques for MapReduce programming model and scheduling, and for the DFS. Energy-efficient techniques proposed for homogenous clusters, mostly use scaling-down approach (i.e., transitioning servers to an inactive, low power consuming sleep/standby state) and scaling-up when needed. Alternatively, in a heterogeneous MapReduce cluster can consist of high- and low-power machines. Also, modern machines are equipped with DVFS capabilities. These features provide opportunities to save energy by intelligently placing jobs and data on its corresponding energy-efficient machines.

4.1.1.1. Energy-Efficient MapReduce

In this section, we overview some of the energy-efficient MapRedeuce techniques for homogeneous and heterogeneous clusters. In homogeneous clusters all machines are assumed to be the same, however in the heterogeneous type, machines have different computing and consumption power.

All-In Strategy (AIS) [76] is a framework for energy management in MapReduce clusters by powering down all nodes in the cluster during a low utilization period. It powers down all nodes during low utilization periods, batches the jobs and powers on whole nodes, performs all jobs and again power down all when entire jobs are completed. Due to its batch characteristics, AIS may not support an instant execution. However, it decreases the response time of workload by running the workload on all the available nodes in the cluster [76][39][77][119]. The AIS is difficult to be applied when MapReduce data analysis is interactive since the cluster is never to be completely inactive [39][119]. A method to cope with the problem of energy reduction in the presence of interactive jobs is to avoid replication [29]. This approach motivated by an empirical analysis of MapReduce interactive workload at Facebook. The analysis reveals that the interactive jobs operate on a small fraction of data; hence a small pool of machines (interactive zone) can be dedicated to these jobs. On the contrary, the less time-sensitive jobs can run on the rest of the cluster (batch zone) in a batch way. The interactive zone is always fully powered while the batch zone is powered down between batches. Consequently, energy saving comes from aggregating jobs in the batch zone.

Unlike the techniques that try to save energy by scaling down the nodes, some other approaches choose to run all nodes in medium power state to avoid the peak power states of servers [134]. This needs an energy-efficient job scheduling that can manage system workload so that few peaks observed in power consumption are removed. Further, a cluster can be dynamically reconfigured by scaling up and down the number of nodes based on the cluster utilization and current workload. The number of nodes is scaled up (down) in the cluster when the average cluster utilization rises above (falls below) a threshold specified by the cluster administrator. By doing this, the nodes in the cluster that are underutilized can be turned off to save power [86].

MapReduce efficiency may be varied depending on various factors in the system. Wirtz [128] consider variation of MapReduce efficiency with respect to two kinds of runtime configurations: resource allocation that changes the number of available concurrent workers, and DVFS that adjusts the processor frequency based on the workloads' computational needs. An optimization technique allocates the optimal number of compute nodes to applications and dynamically schedules processor frequency during its execution based on data movement characteristics.

Being aware of whether the system workloads are IO- or CPU-intensive a right strategy can be chosen for scheduling and mapping the tasks onto the server nodes. In a heterogeneous cluster, IO-bound workloads have better energy efficiency on low-power nodes while CPU-bound workloads achieve better energy efficiency on high-power nodes [131]. Further, the Map task is more CPU-intensive while a Reduce task is more IO-intensive. This observation leads to an energy-aware node selection for a given task. In [23] a spatio-temporal trade-off is introduced that includes the efficient spatial placement of tasks on nodes and temporal placement of nodes with tasks having similar runtimes. The objective is to maximize utilization throughout its uptime for a datacenter with heterogeneous VMs. Based on the following two principles (1) efficient spatial fit, and (2) balanced temporal fit, two VM placement algorithms are proposed. The former co-places various MapReduce VMs based on their complementary CPU, memory, storage and network requirements such that the available resources are fully utilized. The latter co-locates MapReduce VMs with similar runtimes so that a server runs at a high utilization rate. Once all collocated VMs have finished, the cloud operator can hibernate the server to conserve energy. Having various VM types differing in the amount of CPU, memory and storage capacity, each job is assigned a type of VM for its virtual cluster (from a set of predefined VM types).

It is not surprising to know that more parallelism does not always results in more energy efficiency or speed-up, specifically when the job is I/O intensive [120]. Network bandwidth is said to be a significant factor that affects the energy efficiency of the map-reduce jobs. Balancing CPU processing rate against I/O processing rate considering the workload characteristics and network bandwidth can improve the energy efficiency of map-reduce jobs.

DyScale [130] is a Hadoop scheduler that takes into account the heterogeneous features of the cores for achieving better performance objectives with a specified power budget. It enables “slow” slots (running on slow cores) and “fast” slots (running on fast cores) in Hadoop. Within the same power budget, DyScale operating on heterogeneous multi-core processors provides significant performance improvement for small and interactive jobs comparing to using homogeneous processors with (many) slow cores. Moreover, DyScale maintains a good performance for large batch jobs compared to using a homogeneous fast core design (with fewer cores).

4.1.1.2. Energy-Efficient DFS

A solution for improving the energy efficiency of a distributed file system such as HDFS is to recast the data layout and task distribution of the file system to enable significant portions of a cluster to be powered down while still fully operational. Covering-Set (CS) approach exploits the replication feature of distributed file systems. In CS strategy some nodes in the system are specified as special nodes, called CS nodes. At least one copy of each unique data block is kept in these nodes. In this technique, during periods of low utilization, some or all of the non-CS nodes are powered down to save energy. The drawback of CS is that the workloads take longer to run when the cluster has been partially powered down, since fewer nodes are available to run the workload. It also requires code modifications in the underlying DFS software [39][77][78].

GreenHDFS [73] divides the cluster into Hot and Cold zones. Data is placed initially in the zones based on high-level data classification policies and later transitioned from one to another based on its access frequencies. The Hot zone consists of files that are being accessed currently and the newly created files. It includes high-performance, high-power, and high-cost CPUs. All nodes in Hot zone remain in high-power mode at all times. The majority of the servers in the cluster are assigned to Hot zone to ensure higher performance and higher data availability. The cold zone contains files with low to rare accesses. It utilizes machines with a higher number of disks to store a large amount of rarely accessed data. Aggressive power management policies are used in the cold zone to keep nodes in low, inactive power mode as long as possible. Nodes are powered-on only on demand [73][119].

4.1.2. Vertically Scaled Systems

Big data analytics including data mining is per essence parallel regardless of the programming model. Some different architectural designs and programming models have been introduced to speed up the execution of applications and improve the energy efficiency with massive data sets. Such platforms range from multi-core clusters, hybrid clusters, clouds, mobile systems, GPUs, and FPGA systems. Also, a great attention is currently paid to substitute architectures for current server processors that will be discussed in the following sections.

4.1.2.1. Energy-Efficient Processing on GPUs

GPUs have been considered an excellent alternative to CPUs for high-performance and high-throughput computing applications. They can exploit data parallelism, increase computational efficiency, and save energy by orders of magnitude. An engine implemented in [133][53][56] is a query co-processing engine on coupled CPU-GPU architecture (referred as Accelerated Processing Unit- APU) that can work as the base of in-memory database systems. This query co-pocessing engine can improve the query processing performance by leveraging the hardware advances with optimized algorithm design. A fine-grained method distributes workload among available processors since the CPU, and the GPU share the same main memory space. Moreover, an in-cache paradigm is employed for query processing to take advantage of shared cache hierarchy to overcome memory stalls of query processing.

The comparison results of discrete CPU-GPUs and coupled CPU-GPUs show that the average power consumption of discrete architecture is between 36% and 44% higher than those of coupled architecture. The results indicate that discrete GPUs always deliver higher performance at the expense of more energy consumption. The coupled architectures win mainly due to the specifically designed architecture of APU so that various complicated interconnections are eliminated. Thus, it is more energy-efficient for database workloads.

4.1.2.2. Energy-Efficient FPGA-based Processing

FPGAs can support very high rates of data throughput when high parallelism is exploited in circuits implemented in the reconfigurable fabric. FPGA reconfigurability offers a flexibility that makes them even superior to GPU for certain application domains. The key features of FPGA that can provide motivation for big data analytics are: parallelism and efficient power consumption (performance/watt). A vital feature of FPGA is its parallelism through a hierarchical style architecture that can be very much suitable for data processing applications. Many of the widely used and typical data operations can be implemented on FPGA through hardware programmability.

IBM’s Netezza [43], which falls under data warehouse appliance category, is a big data infrastructure platform using FPGA. The analytics appliance includes custom-built FPGA accelerators. Netezza minimizes data movement by using innovative hardware acceleration. It employs FPGA to filter out extraneous data as early in the data stream as possible, and as fast as data can be streamed off the disk. Huge benefits by introducing FPGAs in big data analytics hardware has been proved. Specifically saying, the queries are compiled using FPGAs to minimize overhead. Each FPGA on server blades contains embedded engines that perform filtering and transformation functions on the data stream. These engines are dynamically reconfigurable that enables them to be modified or extended through software. They are customized for every snippet through instructions provided during query execution and act on the data stream at extremely high speeds.

LINQits [30] is a flexible and composable framework for accelerating data-intensive applications using specialized logic. LINQits accelerates a domain-specific query language called LINQ. It has been prototyped on a Xilinx Programmable SoC called the ZYNQ, which combines dual ARM A9 processors and FPGA. LINQits improves energy efficiency by 8.9 to 30.6 times and performance by 10.7 to 38.1 times compared to optimized and multithreaded C programs running on conventional ARM A9 processors.

Academic researchers also showcased the vital development in building infrastructure for big data analytics. For example, BlueDBM or Blue Database Machine [69] is a storage system for big data analytics that can dramatically speed up the time it takes to access information. In this system, each inbuilt flash device is connected to FPGA chip to create an individual node. FPGAs are used not only to control the flash device, but are also capable of performing processing operations on the data itself.

4.1.2.3. How to Choose Right Hardware?

A main challenge is to determine what type of hardware system is more suitable for which data type. It is quite essential for the processing infrastructure to be compute-intensive for running and processing such variety of data under time-bounded format. Most organizations with traditional data platforms such as enterprise data warehouses find that their existing infrastructure is either technically incapable or financially impractical for storing and analyzing big data. According to Intel [67], the selection of key specifications for a given application can be specific to each domain and operations involved in each system. Some of the most notable characteristics to help in decision making during the early stages of development are: energy budget, bandwidth, data width, read/write speed, data process mechanism (e.g., batch and stream processing), network switching topology, traffic density, congestion control and security standards. All these characteristics define four distinctive categories of system specifications for hardware infrastructure including compute, memory, storage, and network.

GPU and FPGA are the possible accelerators that can achieve higher performance and energy efficiency than CPUs on certain jobs. GPU is power-efficient but only for SIMD (Single-Instruction Multiple-Data) streams, and the FPGA is hard to program. However, the data flow style architecture in FPGA may dominate CPU and GPU in providing high-performance memory-intensive operations at low power consumption (performance/watt) for a category of operations. The major overhead of GPUs and CPUs comes from executing instructions that rely on memory accesses. FPGA took the advantage of data-flow streaming, thus saving many of the memory accesses. The main drawback with FPGA is the programming complexity. Although, big data processing is performance-intensive, some applications specifically require reduced energy cost. According to [53] and [97], FPGA could be a viable solution on an energy-cost basis for very high performance, large scale applications compared to GPU and CPU.

Although FPGA is a winner with respect to the power-efficiency, still GPU is outperforming its counterparts in terms of performance for some applications such as multimedia and communication algorithms from the HPC domain [33] that often make extensive use of floating-point arithmetic operations. Due to the fact that complexity and expense of the floating-point hardware on a reconfigurable fabric such as FPGA are high, these algorithms are converted to fixed-point operations thus making FPGA less efficient than GPU for achieving higher speeds.

It is worth mentioning that a combination of FPGA, GPU, and CPU hardware infrastructure is giving good results for the applications such as medical imaging [90]. Even though it tends to be very expensive to develop such true heterogeneous infrastructure, the choice is purely based on the user requirement. Another significant requirement of heterogeneity is in signal processing domain that are in need of signal filtering and transformation, encoding/decoding, floating-point operations, etc.

4.2. New Server Architectures

Recent research showes that the architectures of current servers do not comply well the computational requirements of big data processing applications. Therefore, it is required to look for a a new architecture for servers as a replacement for currently used machines for both performance and energy enhancement. Using low-power processors (micro-servers), more system level integration and a new architecture for server processors are some of the solutions that have been discussed recently as performance/energy-efficient replacement for current machines.

4.2.1. Micro-Servers for Big Data Analytics

Processing clusters based on low-power systems such as ARM processors are feasible, and this is an appropriate way to decrease power usage of several server applications. Prior research shows that the processors based on simple in-order cores are well suited for certain scale-out workloads [81]. A comparison between x86 and ARM architectures for server applications [8] concludes that ARM-based processors are three to four times more energy-efficient than the x86-based processors while comparing requests per second per Watt relation [8][111].

Several other studies have shown the efficiency of ARM micro-servers. One of these studies investigates the energy performance of server workloads on ARM Cortex-A9 multicore system through a trace-driven analytical model [121]. The model involves the degrees of CPU core, memory and I/O resource overlap and estimates the optimal number of cores and clock frequency in favor of higher energy efficiency without compromising execution time. Using the collected metrics and a static power profile of the system, the execution time and energy usage of an application is predicted for a various number of cores and clock frequencies. Then, the configuration that maximizes performance without wasting energy is selected. It is predicted that the selected configurations increase energy efficiency by 10% without turning off cores and just through frequency scaling, and up to one-third with shutting down unutilized cores. For memory-bounded programs, the limited memory bandwidth might increase both execution time and energy usage to the point where energy cost might be higher than on a typical x64 multicore system. The conclusion is that increasing memory and I/O bandwidth can improve the execution time and the energy usage of server workloads on ARM Cortex-A9 systems. Moreover, a 3000-node cluster simulation driven by a real-world trace from Facebook shows that on average a cluster comprising ARM-based microservers which support the Hadoop platform reach the same performance of standard servers while saving energy up to 31% at only 60% of the acquisition cost.

Recently, ARM big.LITTLE boards (as small nodes) has been introduced as a platform for big data processing [83]. In comparison with Intel Xeon server systems (as traditional big nodes), the I/O-intensive MapReduce workloads are more energy-efficient to run on Xeon nodes. In contrast, database query processing is always more energy-efficient on ARM servers, at the cost of slightly lower throughput. With minor software modifications, CPU-intensive MapReduce workloads are almost four times cheaper to execute on ARM servers. Unfortunately, small memory size, low memory and I/O bandwidths, and software immaturity ruins the lower-power advantages obtained by ARM servers. Efficient employment of ARM servers while coping with existing architectural constraints may be considered for further research and development.

4.2.2. Novel Server Processors

It has been shown that scale-out workloads have many characteristics that need to be known as a distinct workload class from desktop, parallel, and traditional server workloads [42]. Due to the large mismatch between the demands of the scale-out workloads and today's processor micro-architecture, scale-out processors have been recently introduced that can result in more area- and energy-efficient servers in future [42][54] [84]. The building block of a scale-out processor is the pod. A pod is a complete server that runs its copy of the operating system. A pod acts as the tiling unit in a scale-out processor, and multiple pods can be placed on a die. A Scale-out chip is a simple composition of one or more pods and a set of memory and I/O interfaces. Each pod couples a small last-level cache to a number of cores using a low-latency interconnect. Having a higher per-core performance and lower energy per operation leads to better energy efficiency in scale-out processors. Due to smaller caches and smaller communication distances, scale-out processors dissipate less energy in the memory hierarchy [84].

FAWN architecture [2] is another solution for building cluster systems for energy-efficient serving massive-scale I/O and data-intensive workloads. FAWN couples low-power and efficient embedded processors with flash storage to provide fast and energy efficient processing of random read-intensive workloads.

4.2.3. System-Level Integration (Server-on-Chip)

System-level integration is an alternative approach that has been proposed for improving the efficiency of the warehouse-scale datacenter server market. System-level integration discusses placing CPUs and components on the same die for servers, as done for embedded systems. Integration reduces the 1) latency: by placing cores and components closer to one another, 2) cost: by reducing parts in the bill of material, and 3) power: by decreasing the number of chip-to-chip pin-crossings. Initial results show a reduction of more than 23% of capital cost and 35% of power costs at 16 nm [79].

5. Trends for the Big Data Analytics in Cloud-based Datacenters

Despite great changes in the way data is accessed and managed, the datacenter has kept its server-centric model which is not adequate to keep up with new services, new energy requirements, and new business models [31]. To accommodate big data analytics, the next generation cloud-based datacenter should comply with a new set of requirements. In the following, the challenges and opportunities in big data analytics are discussed for future research and development.

5.1. Energy Efficiency

Measurements on current datacenters indicate significant wasted energy due to underutilization with average utilization values of about 30% [125]. Today's application environments are more distributed, often with multiple tiers, and oriented toward service delivery, eventually resulted in:

- Greater traffic volume on the Ethernet network,
- More storage traffic as applications use distributed file systems and increase the amount of synchronization and replication data across the network,
- Greater traffic flow between peer nodes such as server-to-server or VM-to-VM [34].

The massive number of processing units puts interconnection network systems under pressure to increase performance and accommodate better the communication among them [96]. The higher level of network complexity and higher link bandwidth increase the energy consumed, and the heat emitted. The energy cost and the heat dissipation problems in interconnection networks necessitate future network systems be built with much more efficient power than today. Furthermore, computing with renewable energy is crucial in the future datacenters. There is a strong need for innovative technologies for improved energy capturing, and management of energy usage with energy storage and availability [31].

Table 2. Performance, Bandwidth requirements, and power consumption bound for future systems [15]

Year	Peak Performance (10× every four years)	Bandwidth requirements (20× every four years)	Power consumption bound (2× every four years)
2012	10 PFLOPS	1 PB/s	5 MW
2016	100 PFLOPS	20 PB/s	10 MW
2020	1000 PFLOPS	400 PB/s	20 MW

Table 2 shows the projection of performance, bandwidth and power consumption for future datacenters. As one can see, the peak performance and bandwidth requirements are highly demanding while the affordable power consumption is in the much lower increasing rate. This reveals the great challenges in power consumption of future data centers and the opportunities for new research in this area. Recently, energy proportionality received some attention as a viable solution for power consumption in datacenters [49] though need deeper investigation to be utilized in next generation data centers.

5.2. Interconnection Networks

To support the big data analytics in datacenters, the high throughput/bandwidth interconnection networks with low latency and low power consumption is essential. This includes the data and storage area networks (e.g., SAN) as by increasing the size of the data the demand rises in storage systems as well. Depending on the kind of data management software in use and the kinds of data analyzed, big data can influence the size and frequency of data movements among servers and between servers and storage [69]. Moving large objects (e.g., videos or high-resolution images) from long-term storage to analytical nodes will place a bigger burden on storage systems and data networks than moving myriad small text files [1][18]. Sizing network links, prioritization, traffic shaping, and data compression can be remedies [18][31]. Due to increasing rate of data-intensive applications requiring ubiquitous connectivity, 400G Ethernet will be required in datacenters/routers. Cost considerations is a main reason that copper wires are still favorable even for high-bandwidth networks [94]. Moreover, using optical technologies is a viable solution that needs further research and investigation for future datacenter networks.

5.3. Datacenters Architecture

Distributing data-intensive applications across multiple datacenters seem to be a good solution for efficient big data management in the clouds. However, it should be supported by a reliable framework for managing large and distributed data sets and mobile databases stored on mobile devices over multiple datacenters while optimizing the network (e.g., latency and throughput) and cloud service QoS (e.g., cost, response time, etc.) [31][34]. There is a need for following technologies to realize distributed applications over multiple datacenters:

- Collaborative distributed datacenters that can support both location- and workload-dependent services and redundancy for failure recovery. The system architecture should be based on multiple datacenters, including scheduling, power management, and data/workload placement.

- Innovative networking fabrics and distributed systems to extend services across multiple heterogeneous datacenters.

- Optimal resource provisioning across multiple datacenters. Large-scale distributed data-intensive applications, e.g., environment monitoring applications, need to process and manage massive data sets across geographically distributed datacenters. These kinds of applications that combine multiple, independent, and geographically distributed software and hardware resources require provisioning across multiple datacenter resources.

- Software frameworks and services that allow portability of distributed applications across multiple datacenters. Despite the existing technological advances of the data processing paradigms (e.g., MapReduce, and cloud computing platforms), large-scale, reliable system-level software for big data applications is yet to become commonplace.

- Appropriate programming abstractions, which can extend the capability of existing data processing paradigms to multiple datacenters.

Heterogeneity and Virtualization: The data and processing platform heterogeneity is inevitable in the future of the datacenters. The concept of data center remains the same as before, but the physical implementation will be different from the prior generations. The next-generation datacenter will be deployed on a heterogeneous infrastructure and architectures that integrate both traditional structured data and big data into one scalable environment [64]. New server and processor architectures appropriately matching with big data analytics requirements pose an indispensable trend for the micro-architects. Virtualization infrastructure is the key concept in data centers to support hardware and software heterogeneity and simplify the resource provisioning [1]. Virtualization has a wide concept and has been studied for several years for data centers and cloud computing systems. The high-level language virtualization is the most relevant topic for big data analytics, which allow languages to be executed on multiple computing architectures. This needs to be done through compiler and runtime environment (e.g., JVM). The main challenge here is to consider the programmer's productivity in future big data analytics software since it has a great impact on system cost.

5.4. Resource Provisioning for Big Data Applications

Processing of uncertain and heterogeneous data volumes requires optimal provisioning of cloud-based datacenters. The process of provisioning hardware and software resources to big data applications requires the resource provider to compute the best software and hardware configuration to ensure that QoS targets of applications are achieved, while maximizing energy efficiency and utilization. Schad et al. [109] show that the QoS uncertainty of applications is the main technical obstacle to the successful adoption of cloud datacenters.

The uncertainties of resource provisioning have two aspects. Firstly, from the perspective of big data applications, it is difficult to estimate its workload behavior in terms of data volume to be analyzed, data arrival rate, data types, data processing time distributions, I/O system behavior, number of users, and type of network connecting to the application. Secondly, from a resource perspective, without knowing the behaviors of big data applications, it is difficult to make decisions about the size of resources to be provisioned at a certain time. Furthermore, the availability, load, and throughput of datacenter resources can vary in unpredictable ways, due to failure, malicious attacks, or congestion of network links. In other words, we need reasonable application workload and resource performance prediction models when making provisioning decisions [34].

One possible way to deal with uncertainties is to rely on monitoring the state of both hardware and software resources and taking predefined actions when some events occur. In [129] the operating system level metrics such as available CPU percentage, available non-paged memory and TCP/IP performance are gathered for such purpose. A network QoS-aware provisioning of MapReduce framework for private cluster computing environments has been introduced in [105]. However, this problem is more challenging and poses many open problems on the QoS-based predictive resource provisioning for multiple cloud datacenters [38]. Therefore, there are several challenges in the resource provisioning that should be researched to provide QoS requirements of big data applications.

6. Summary

Highly virtualized cloud-based datacenters currently provide the main platform for big data analytics. As the scale of datacenters is increasingly expanding to accomodate big data needs, minimizing energy consumption and operational cost is a vital concern. The datacenters infrastructure including interconnection network, storage and servers should be able to handle big data applications in an energy-efficient way. The datacenter may be vertically scaled and equipped with more and faster hardware, processors, and memory to be able to handle large future workloads. Furthermore, horizontal scaling distributes the work across many servers which is an economical way to increases the performance compared with vertical scaling. Improving the utilization rate of servers and network resources, adjusting routing to the traffic flow, using optical interconnection networks, consolidating applications through virtualization in the cloud, optimizing network topologies established among VMs, efficient distribution of workload in the federated multi-cloud datacenters, and optimizing the power consumption of servers, processors, power supplies, and memory system are some of techniques for energy-efficient data analytics in data centers. There are still many challenges and open problems as well as the need for improved or new technologies in the architecture and interconnection networks, resource provisioning, etc.

References

[1] M. Al-Fares, A. Loukissas, and A. Vahdat, A Scalable, Commodity Datacenter Network Architecture, Proceedings of the ACM SIGCOMM Conference on Data Communication, pp. 63–74. 2008.

[2] D. G. Andersen, et al., FAWN: a Fast Array of Wimpy Nodes, in SOSP ’09, pp. 1–14, 2009.

[3] Amazon Elastic MapReduce (Amazon EMR), http://aws.amazon.com/elasticmapreduce/

[4] G. Ananthanarayanan and R. H. Katz, Greening the switch, In Proc. of HotPower, Dec 2008.

[5] J. Anderson, J. Lehnardt, and N. Slater, CouchDB: the definitive guide, O'Reilly Media, Inc., 2010.

[6] A. Anwar, K.R. Krish, A.R. Butt, On the Use of Microservers in Supporting Hadoop Applications, IEEE International Conference on Cluster Computing (CLUSTER), 2014.

[7] J. Armstrong, OFDM for optical communications, Journal of Lightwave Technology 27.3:189-204, 2009.

[8] R.V. Aroca, L.M. Garcia Gonçalves, Towards green data centers: A comparison of x86 and ARM architectures power efficiency, J. Parallel Distrib. Comput. 72 (2012) 1770–1780.

[9] M. Balman, Streaming Exascale Data over 100Gbps Networks, IEEE CS Computing Now Magazine, 2014.

[10] R. Barga, J. Ekanayake, and W. Lu, Project daytona: Data analytics as a cloud service, Data Engineering (ICDE), IEEE 28th International Conference on. IEEE, 2012.

[11] L.A. Barroso, J. Clidaras, and U. Holzle, The Datacenter as a Computer An Introduction to the Design of Warehouse-Scale Machines, Second Edition, Morgan & Claypool Publishers, 2013.

[12] D. Battre, S. Ewen, F. Hueske, O. Kao, V. Markl, D. Warneke, Nephele/PACTs: A Programming Model and Execution Framework forWeb-Scale Analytical Processing, In Proceedings of the 1st ACM Symposium on Cloud Computing, SoCC, pp. 119–130. 2010.

[13] D. Beaver, et al., Finding a Needle in Haystack: Facebook's Photo Storage, OSDI. Vol. 10. 2010.

[14] A. Beloglazov, R. Buyya, Y. Choon Lee, and A. Zomaya, A Taxonomy and Survey of Energy-efficient Datacenters and Cloud Computing Systems, Advances in Computers, Vol. 82, 2010 (ISSN: 0065-2458/DOI: 10.1016/B978-0-12-385512-1.00003-7).

[15] A. Benner, Optical interconnect opportunities in supercomputers and high end computing, Optical Fiber Communication Conference, Optical Society of America, 2012.

[16] K. Bergman, et al., ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, 2008.

[17] Berkeley Data Analysis Stack, https://amplab.cs.berkeley.edu/software/

[18] K.S. Beyer, V. Ercegovac, R. Gemulla, A. Balmin, M. Eltabakh, C.C. Kanne, F. Ozcan, E.J. Shekita, Jaql: A Scripting Language for Large Scale Semistructured Data Analysis, In PVLDB, pp. 1272–1283, 2011.

[19] V.R. Borkar, M.J. Carey, R. Grover, N. Onose, R. Vernica, Hyracks: A Flexible and Extensible Foundation for

Data-intensive Computing, In: ICDE, pp. 1151– 1162, 2011.

[20] J. Burke, Big data infrastructure depends on good WAN management, [online] http://searchenterprisewan.techtarget.com/tip/Big-data-infrastructure-depends-on-good-WAN-management.

[21] R. Buyya, High Performance Cluster Computing: Architectures and Systems (Volume 1). Prentice Hall, 1999.

[22] CALIENT Optical Switching in the New Generation Datacenter, [online] http://www.telecomreviewna.com/index.php?option=com_content&view=article&id=151:calient-optical-switching-in-the-new-generation-datacenter&catid=40:march-april-2012&Itemid=80.

[23] M. Cardosa, A. Singh, H. Pucha, and A. Chandra, Exploiting spatio-temporal tradeoffs for energy-aware MapReduce in the Cloud, IEEE Transactions on Computers, pp. 1737–1751, 2012.

[24] P. Castoldi, N. Androiolli, I. Cerutti, O. Liboiron-Ladouceur, P.G. Raponi, Energy Efficiency and Scalability of Multi-plane Optical Interconnection Networks for Computing Platforms and Datacenters, Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2012 (DOI: 10.1364/OFC.2012.OW3J.4).

[25] I. Cerutti, P.G. Raponi, N. Andriolli, P. Castoldi, and O. Liboiron-Ladouceur, Designing Energy-efficient Datacenter Networks Using Space-Time Optical Interconnection Architectures, IEEE Journal of Selected Topics in Quantum Electronics, Vol. 19, No. 2, March/April 2013.

[26] F. Chang, et al. Bigtable: A distributed storage system for structured data, ACM Transactions on Computer Systems (TOCS) 26.2:4, 2008.

[27] M. Chen, S. Mao, Y. Liu, Big Data: A Survey, Mobile Network Applications, Vol. 19, p. 171–209, 2014 (DOI 10.1007/s11036-013-0489-0).

[28] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, Y. Chen, OSA: an optical switching architecture for data center networks with unprecedented flexibility, In Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, 2012.

[29] Y. Chen, S. Alspaugh, D. Borthakur, and R. Katz, Energy efficiency for large-scale mapreduce workloads with significant interactive analysis, in Proc. of the 7th ACM European Conf. on Computer Systems, pp. 43–56, 2012.

[30] E.S. Chung, J.D. Davis, and J. Lee, LINQits: big data on little clients", 40th Annual International Symposium on Computer Architecture, 2013.

[31] G. Cook, How clean is your cloud? Report, Greenpeace International, April 2012.

[32] CORDIS, The next generation datacenter, Report from the Workshop on Next-Generation Datacenters, June 2011.

[33] C. Cullinan, et al., Computing Performance Benchmarks among CPU, GPU, and FPGA, in MathWorks, 2012.

[34] G. DeCandia, et al., Dynamo: Amazon's highly available key-value store, ACM SIGOPS Operating Systems Review, Vol. 41. No. 6. 2007.

[35] J. Dean, and S. Ghemawat, MapReduce: simplified data processing on large clusters, Communications of the ACM, 51.1, pp. 107-113, 2008.

[36] D.J. DeWitt, R.H. Gerber, G. Graefe, M.L. Heytens, K.B. Kumar, M. Muralikrishna, GAMMA - A High Performance Dataflow Database Machine, In Proceedings of the 12th International Conference on Very Large Data Bases

(VLDB), pp. 228–237. Morgan Kaufmann Publishers Inc., 1986.

[37] J.P. Dijcks, Oracle: Big Data for the Enterprise, Oracle White Paper, 2012.

[38] B. Di-Martino, R. Aversa, G. Cretella and A. Esposito, Big data (lost) in the cloud, Int. J. Big Data Intelligence, Vol. 1, Nos. 1/2, 2014.

[39] A. Elsayed, O. Ismail, and M.E. El-Sharkawi, MapReduce: State-of-the-Art and Research Directions, International Journal of Computer and Electrical Engineering, Vol. 6, No. 1, February 2014.

[40] Emerson, Energy-efficient Cooling Solutions for Datacenters, White Paper.

[41] Farrington, N., Porter, G., Radhakrishnan, S., Bazzaz, H., Subramanya, V., Fainman, Y., Papen, G., Vahdat, A., Helios: a hybrid electrical/optical switch architecture for modular data centers, In ACM SIGCOMM Computer Communication Review, vol. 40, pp. 339–350. 2010.

[42] M. Ferdman, et al., Clearing the Clouds: A Study of Emerging Scale-out Workloads on Modern Hardware, 17th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), March 2012.

[43] P. Francisco, The Netezza Data Appliance Architecture: A Platform for High Performance Data Warehousing and Analytics, in IBM Redbooks, 2011.

[44] S. Fushimi, M. Kitsuregawa, H. Tanaka, An Overview of the System Software of a Parallel Relational Database Machine GRACE, In Proceedings of the 12th International Conference on Very Large Data Bases (VLDB), pp. 209–219, Morgan Kaufmann Publishers Inc., 1986.

[45] N. Ghani, S. Dixit, and T.S. Wang, On IP-over-WDM integration, Communications Magazine, IEEE Vol. 38, No.3, pp. 72-84, 2000.

[46] S. Ghemawa, H. Gobioff, and S-T. Leung, The Google file system, ACM SIGOPS Operating Systems Review, Vol. 37, No. 5, 2003.

[47] M. Glick, Optical switching for next generation datacenters, in Proc. Photonics in Switching, 2009.

[48] I. Goiri, K. Le, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini, Greenhadoop: leveraging green energy in data-processing frameworks, in Proc. of the 7th ACM European Conf. on Computer Systems, pp. 57-70, 2012.

[49] GreenDataProject: Where does power go?, 2008 [online] http://www.greendataproject.org.

[50] A. Greenberg, J.R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta, VL2: A scalable and flexible datacenter network, In ACM SIGCOMM, 2009 (doi: http://doi.acm.org/10.1145/1594977.1592576.

[51] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, BCube: A High Performance, Server-centric Network Architecture for Modular Datacenters, In ACM SIGCOMM Comput. Commun. Rev. 39(4), pp. 63–74. 2009.

[52] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, DCell: A Scalable and Fault-Tolerant Network Structure for Datacenters, In ACM SIGCOMM Comput. Commun. Rev. 38(4), pp. 75–86, 2008.

[53] T. Hamada, et al., A Comparative Study on ASIC, FPGAs, GPUs and General Purpose Processors in the O(N^2) Gravitational N-body Simulation, Proc. NASA/ESA Conference on Adaptive Hardware and Systems, 2009.

[54] N. Hardavellas, I. Pandis, R. Johnson, N. Mancheril, A. Ailamaki, and B. Falsafi. Database servers on chip multiprocessors: limitations and opportunities. In The Conference on Innovative Data Systems Research, Jan. 2007.

[55] J. He, M. Lu, and B. He, Revisiting co-processing for hash joins on the coupled CPU-GPU architecture, Proceedings of the VLDB Endowment, Vol. 6, Issue 10, pp. 889-900, 2013.

[56] J. He, S. Zhang, and B. He, In-cache query co-processing on coupled CPU-GPU architectures, Proceedings of the VLDB Endowment, Vol. 8 Issue 4, pp. 329-340, 2014.

[57] B. Heller, S. Seetharaman, P. Mahadevan, ElasticTree: Saving Energy in Datacenter Networks, Proceedings of the 7th USENIX Conference on Networked systems design and implementation, pp. 17-17, Apr. 2010.

[58] H. Herodotou, et al., Starfish: A Self-tuning System for Big Data Analytics, CIDR, pp. 261-272, 2011.

[59] H. Herodotou, and S. Babu, Profiling, what-if analysis, and cost-based optimization of MapReduce programs, Proceedings of the VLDB Endowment, Vol. 4, Issue11, pp. 1111-1122, 2011.

[60] Hewlett-Packard Development Company, Effects of virtualization and cloud computing on datacenter networks, Technology Brief, 2011.

[61] Hewlett-Packard Technical White Paper, Powering the 21st-century datacenter with HP next generation FlexFabric.

[62] U. Hoelzle, L.A. Barroso, The datacenter as a computer: an introduction to the design of warehouse-scale machines, 1st Edition, Morgan and Claypool Publishers, 2009 (ISBN 9781598295566).

[63] L. Huff, The Choice for Datacenter Cabling, Berk-Tek Technology Summit, 2008.

[64] IBM Infosphere BigInsights, 2013 [online] http://www.ibm.com/software/data/infosphere/biginsights/

[65] IBM and Cisco: Together for a World Class Datacenter, An IBM Redbooks Publication, 2013.

[66] Infochimps (2013) [online] http://www.infochimps.com.

[67] Intel Big Data Analytics White Paper, Extract, Transform and Load Big Data with Apache Hadoop, 2013.

[68] M. Isard, et al. "Dryad: distributed data-parallel programs from sequential building blocks." ACM SIGOPS Operating Systems Review. Vol. 41. No. 3. ACM, 2007.

[69] B. Javadi, B. Zhang, and M. Taufer, Bandwidth Modeling in Large Distributed Systems for Big Data Applications, The 15th International Conference on Parallel and Distributed Computing, Applications and Technologies, Hong Kong, 2014.

[70] S.W. Jun, et al., Scalable Multi-Access Flash Store for Big Data Analytics', Proc. Int. Conf. on Field-Programmable Gate Arrays, 2014.

[71] C. Kachris, K. Bergman, and I. Tomkos, Optical interconnects for future datacenter networks, Springer Science & Business Media, 2012.

[72] K. Krishnan, Data Warehousing in the Age of Big Data, Elsevier, 2013 (ISBN 978-0-12-405891-0).

[73] R. T. Kaushik, M. Bhandarkar, and K. Nahrstedt, Evaluation and analysis of GreenHDFS: A self-adaptive, energy-conserving variant of the Hadoop distributed file system, in Proc. 2nd IEEE Int l Conf. on Cloud Computing Technology and Science, pp. 274–287, 2010.

[74] A. Lakshman, and P. Malik, Cassandra: structured storage system on a p2p network, Proceedings of the 28th ACM symposium on Principles of distributed computing, 2009.

[75] C. Lam, K.H. Liu; B. Koley, Z. Xiaoxue, V. Kamalov and V. Gill, Fiber optic communication technologies: What's needed for datacenter network operations, IEEE Communications Magazine, vol. 48, no. 7, pp.32-39, July 2010.

[76] W. Lang and J.M. Patel, Energy management for MapReduce clusters, Proceedings of the VLDB, 3(1-2):129–139, 2010.

[77] K.H. Lee, Y.J. Lee, H. Choi, Y.D. Chung, B. Moon, Parallel Data Processing with MapReduce: A Survey, Newsletter ACM SIGMOD Record Archive, Vol. 40 Issue 4, pp. 11-20, Dec. 2011.

[78] J. Leverich and C. Kozyrakis, On the energy (in) efficiency of Hadoop clusters, ACM SIGOPS Operating Systems Review, vol. 44, no. 1, pp. 61–65, 2010.

[79] S. Li, K. Lim, P. Faraboschi, J. Chang, P. Ranganathan, N. P. Jouppi, System-level integrated server architectures for scale-out datacenters, Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture, 2011.

[80] D. Li, C. X. Guo, H. T. Wu, K Tan, Y. G. Zhang, S. W. Lu, FiConn: Using Backup Port for Server Interconnection in Datacenters, In INFOCOM, 2009.

[81] K. Lim, P. Ranganathan, J. Chang, C. Patel, T. Mudge, and S. Reinhardt, Understanding and designing new server architectures for emerging warehouse-computing environments. In Proc. of the Int'l Symposium on Computer Architecture, June 2008.

[82] Y. Liu, J. K. Muppala, M. Veeraraghavan, and D. Lin, M. Hamdi, Datacenter Networks Topologies, Architectures and Fault-Tolerance Characteristics, Springer, 2013 (ISSN 2191-5768).

[83] D. Loghin, B.M. Tudor, H. Zhang, B.C. Ooi, Y.M. Teo, A Performance Study of Big Data on Small Nodes, Proceedings of the VLDB Endowment, Vol. 8, No. 7, 2015.

[84] P. Lotfi-Kamran, et al., Scale-Out Processors, 39th Annual International Symposium on Computer Architecture International Symposium on Computer Architecture (ISCA), 2012.

[85] G. Magklis, M. Scott, G. Semeraro, et al., Profile-based Dynamic Voltage and Frequency Scaling for a Multiple Clock Domain Microprocessor, In ISCA'03, Jun 2003.

[86] N. Maheshwari, R. Nanduri, and V. Varma, Dynamic energy-efficient data placement and cluster reconfiguration algorithm for MapReduce framework, Future Generation Computer Systems, vol. 28, no. 1, pp. 119–127, 2012.

[87] G. Malewicz, et al., Pregel: a system for large-scale graph processing, Proceedings of the ACM SIGMOD International Conference on Management of data, 2010.

[88] M. McKusick and S. Quinlan, GFS: Evolution on Fast-forward, Queue 7.7 (2009): 10.

[89] D. Meisner, B. Gold, T. Wenisch, PowerNap: Eliminating Server Idle Power, In ASPLOS'09, May 2009.

[90] P. Meng, et al., FPGA-GPU-CPU Heterogenous Architecture for Real-time Cardiac Physiological Optical Mapping, Proc. of International Conference on Field-Programmable Technology, 2012.

[91] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, Z. Xu, Peer-to-peer computing, Technical Report HPL-2002-57, HP Labs, 2002.

[92] J. Murty, Programming Amazon web services: S3, EC2, SQS, FPS, and SimpleDB, O'Reilly Media, Inc., 2008.

[93] R. N. Mysore, et al., PortLand: A Scalable Fault-Tolerant Layer 2 Datacenter Network Fabric, In ACM SIGCOMM, August 2009.

[94] R. Navid, High-Speed and Power-Efficient Serial Links for Big Data, International Workshop on IoT Enabling Chips, Kyoto Institute of Technology, Japan, June 2015.

[95] S. Nedevschi, J. Chandrashekar, and B. Nordman, Skilled in the Art of Being Idle: Reducing Energy Waste in Networked Systems, Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, pp. 381-394, Apr 2009.

[96] H. Nguyen, D. Franco, and E. Luque, Performance-Aware Energy Saving Mechanism in Interconnection Networks for Parallel Systems, ICCS 2014. 14th International Conference on Computational Science, Volume 29, pp. 134–144, 2014.

[97] K. C. Nunna, Farhad Mehdipour, A. Trouve, K. Murakami, A Survey on Big Data Processing Infrastructure: Evolving Role of FPGA, The International Journal of Big Data Intelligence, http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=IJBDI, 2015.

[98] C. Olston, B. Reed, U. Srivastava, R. Kumar, A. Tomkins, Pig latin: a not-so-foreign language for data processing, In Proceedings of the ACM SIGMOD international conference on Management of Data, pp. 1099–1110, 2008.

[99] Oracle Enterprise Architecture White Paper — An Enterprise Architect's Guide to Bi Data [online].

[100] Opani (2013) [online] http://www.opani.com

[101] OpenCrowd: Cloud Computing Vendors Taxonomy [online] http://cloudtaxonomy.opencrowd.com/

[102] C. Patel, C. Bash, R. Sharma, M. Beitelmam, and R. Friedrich. Smart Cooling of datacenters. In Proceedings of InterPack, July 2003.

[103] B. Peer, P. Rajbhoj, N. Chathanur, Complex Events Processing: Unburdening Big Data Complexities, Infosys Labs Briefings, Vol. 11, No. 1, 2013.

[104] PiCloud (2013) [online] http://www.picloud.com/

[105] N. Rizvandi, J. Taheri, R. Moraveji, and A.Y. Zomaya, Network load analysis and provisioning of MapReduce applications, Proc. of International Conference on Parallel and Distributed Computing, Applications, and Technologies (PDCAT), 2013.

[106] M. Saecker, and V. Markl, Big Data Analytics on Modern Hardware Architectures: A Technology Survey, Business Intelligence, Lecture Notes in Business Information Processing, Vol. 138, pp. 125-149, 2013.

[107] K.S. Sangeetha and P. Prakash, Big Data and Cloud: A Survey, Artificial Intelligence and Evolutionary Algorithms in Engineering Systems, Advances in Intelligent Systems and Computing 325, DOI 0.1007/978-81-322-2135-7_81.

[108] Scalable Energy-efficient Datacenters, 2009, [online] http://phys.org/news170618017.html.

[109] J. Schad, J. Dittrich, and J.A. Quiane-Ruiz, Runtime measurements in the cloud: observing, analyzing, and reducing variance, Proceedings of the VLDB Endowment, Vol. 3, Nos. 1–2, pp.460–471, 2010.

[110] Y. Shang, D. Li, and M. Xu, Energy-aware Routing in Datacenter Network, Green Networking, Aug. 2010.

[111] J. Shuja, K. Bilal, S. A. Madani, M. Othman, R. Ranjan, P. Balaji, and S.U. Khan, Survey of Techniques and

Architectures for Designing Energy-efficient Data Centers, IEEE Systems Journal, Issue 99, pp. 1-13, July 2014.

[112] K. Shvachko, et al., The Hadoop distributed file system, In Proc. of the IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), 2010.

[113] D. Singh, and C.K. Reddy, A survey on platforms for big data analytics, Journal of Big Data, 2:8, October 2014.

[114] S. Srikantaiah, A. Kansal and F. Zhao, Energy Aware Consolidation for Cloud Computing, HotPower'08, Dec 2008.

[115] R. Steinmetz, K. Wehrle, Peer-to-Peer Systems and Applications, Springer Berlin, Heidelberg, 2005.

[116] R. Sumbaly, et al., Serving large-scale batch computed data with project voldemort, Proceedings of the 10th USENIX Conference on File and Storage Technologies. USENIX Association, 2012.

[117] Teradata Corporation, Teradata (December 2011), [online] http://www.teradata.com/

[118] A. Thusoo A, et al., Hive: a warehousing solution over a map-reduce framework, Proceedings of the VLDB Endowment 2(2):1626–1629, 2009.

[119] N. Tiwari, Scheduling and Energy Efficiency Improvement Techniques for Hadoop Map-reduce: State of Art and Directions for Future Research, Ph. D Seminar Report, Department of Computer Science and Engineering, Indian Institute of Technology, 2012.

[120] N. Tiwari, S. Sarkar, U. Bellur, M. Indrawan, An Empirical Study of Hadoop's Energy Efficiency on a HPC Cluster, 14th International Conference on Computational Science (ICCS), 2014.

[121] B.M. Tudor, Y.M. Teo, On Understanding the Energy Consumption of ARM-based Multicore Servers, Proceedings of the 34th ACM SIGMETRICS, Pittsburgh, USA, 2013.

[122] N. Vasic, M. Barisits, V. Salzgeber and D. Kostic, Making cluster applications energy-aware, In ACM Proceedings of the 1st Workshop on Automated control for datacenters and clouds (ACDC), pp. 37-42, 2009.

[123] C. Wang, Survey of Recent Research Issues in Datacenter Networking, [online] http://www.cse.wustl.edu/~jain/cse570-13/ftp/dcn.

[124] G. Wang, D. Andersen, M. Kaminsky, K. Papagiannaki, T. Ng, M. Kozuch, M. Ryan, c-Through: part-time optics in data centers, In Proc. of ACM SIGCOMM Computer Communication Review, vol. 40, pp. 327–338, 2010.

[125] H. Wang, K. Bergman, Optically Interconnected Datacenter Architecture for Bandwidth Intensive Energy-efficient Networking, In Proc. of 14th International Conference on Transparent Optical Networks (ICTON), 2012.

[126] M. Webb, Smart 2020: Enabling the low carbon economy in the information age, The Climate Group, 2008.

[127] T. White, Hadoop: The definitive guide, O'Reilly Media, Inc., 2012.

[128] T.S. Wirtz, Energy-efficient Data-Intensive Computing With MapReduce, Master's Thesis, Marquette University, 2013.

[129] R. Wolski, Experiences with predicting resource performance on-line in computational grid settings, ACM SIGMETRICS Performance Evaluation Review, Vol. 30, No. 4, pp.41–49, 2009.

[130] F. Yan, L. Cherkasova, Z. Zhang, E. Smirni, DyScale: a MapReduce Job Scheduler for Heterogeneous Multicore Processors, IEEE Transactions on Cloud Computing, Issue: 99, 2015.

[131] N. Yigitbasi, K. Datta, N. Jain, and T. Willke, Energy-efficient Scheduling of MapReduce Workloads on Heterogeneous Clusters, Proceedings of the 2nd International Workshop on Green Computing Middleware, pp.

1-6, 2011.

[132] M. Zaharia, M. Chowdhury, M.J. Franklin, S. Shenker, Spark SI- Cluster Computing with Working Sets, In Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, 2010.

[133] S. Zhang, J. He, B. He, and M. Lu. OmniDB: towards portable and efficient query processing on parallel CPU/GPU architectures, Proceedings of the VLDB Endowment, Vol. 6, Issue 12, pp. 1374-1377, 2013.

[134] N. Zhu, L. Rao, X. Liu , J. Liu, H. Guan, Taming Power Peaks in MapReduce Clusters. In Proceedings of the ACM SIGCOMM conference. pp. 416-417, 2011.