Received September 10, 2015, accepted September 30, 2015, date of publication October 14, 2015, date of current version October 27, 2015.

Digital Object Identifier 10.1109/ACCESS.2015.2490085

CF4BDA: A Conceptual Framework for Big Data Analytics Applications in the Cloud

QINGHUA LU[1], ZHENG LI[2], MARIA KIHL[2], LIMING ZHU[3], AND WEISHAN ZHANG[1]

[1]College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China
[2]Department of Electrical and Information Technology, Lund University, Lund 223 63, Sweden
[3]Data61, Commonwealth Scientific and Industrial Research Organisation, Sydney, NSW 2015, Australia

Corresponding author: Z. Li (imlizheng@gmail.com)

This work was supported by the National Natural Science Foundation of China under Grant 61402533.

ABSTRACT Building big data analytics (BDA) applications in the cloud introduces inevitable challenges, such as loss of control and uncertainty. To address the existing challenges, numerous efforts have been made on BDA application engineering to optimize the quality of BDA applications in the cloud, such as performance and reliability. However, there is still a lack of systematic view on engineering BDA applications in the cloud. Therefore, in this paper, we present a conceptual framework named CF4BDA to analyze the existing work on BDA applications from two perspectives: 1) the lifecycle of BDA applications and 2) the objects involved in the context of BDA applications in the cloud. The framework can help researchers and practitioners identify the research opportunities in a structured way and guide implementing BDA applications in the cloud. We perform a preliminary evaluation of the usefulness of CF4BDA by applying it to analyze a set of representative studies.

INDEX TERMS Big data analytics, cloud computing, conceptual framework, software engineering.

I. INTRODUCTION

Big data analytics (BDA) applications are defined as a new category of software applications that process large amounts of data using large scale parallel processing infrastructure to uncover hidden value [1]. For instance, eBay has deployed BDA applications to optimise product search by processing 5 PB data with more than 4000 CPU cores [2], and Facebook repeats collecting and analysing more than 700 TB data every day to drive its core business [3]. Considering that BDA applications generally require efficient high-performance processors to produce timely results, the booming Cloud computing has been recognised to be an effective paradigm for addressing both the intensive computational and data storage needs of BDA [4]. In fact, given the increasing trend of storing commercial data in the Cloud [5], implementing pervasive and scalable data analytics would naturally and gradually rely on the Cloud infrastructure. A recent report [6] has pointed that 53% of enterprises have deployed (28%) or plan to deploy (25%) their BDA applications in the Cloud.

Although the benefits of employing Cloud computing is clear, BDA in the Cloud has inevitable challenges. Firstly, ''Cloud computing'' implies invisibility and loss of control from the Cloud consumers' perspective [7] (e.g., configurations of physical infrastructure) of Cloud services. Unlike traditional consumer-owned computing systems, Cloud users have little knowledge and control over the precise nature of Cloud services even in a ''locked down'' environment [8]. Secondly, there is uncertainty in the runtime Cloud services. On the one hand, it has been recognized that the given indicators often lack providing comprehensive information about the overall performance of a service regarding specific tasks [9]. On the other hand, it is difficult to precisely predict how the same Cloud services support various Quality of Service (QoS) consumption requirements of different applications [10].

In order to improve the quality of BDA applications in the cloud, novel software engineering methods, techniques, and tools are needed for developing and deploying BDA applications in the cloud. We anticipate that engineering platforms for BDA applications in the cloud will become common in the near future to serve data scientist who do not have much software engineering background.

To address the existing challenges, a number of research projects have been working on developing and deploying BDA applications [11]–[14] to optimise quality of BDA applications in the Cloud, such as performance, reliability, and security. The BDA process proposed in [1] is too general,

which can hardly be used to guide BDA engineering. There is still a lack of a systematic and holistic view on what concerns we should keep in mind when implementing BDA. Therefore, we develop a conceptual framework named CF4BDA to locate the implementation challenges and the identified solutions in a structured fashion. In particular, we consider a BDA application from the perspective of software engineering, and construct our CF4BDA framework into two dimensions. Based on the lifecycle analysis, the first dimension is to take different implementation stages to examine where the possible BDA challenges might exist. By object-oriented thinking, the second dimension is to identify different elementary entities of BDA implementation to explore what potential solutions could be employed. Please note that the proposed conceptual framework is an initial outcome of our ongoing systematic literature review. The selection of the surveyed papers followed the guideline of systematic literature review [15].

The contribution of this work is mainly threefold. First, the proposed CF4BDA conceptual framework supplies a structured way to investigate the existing relevant studies, and then helps both researchers and practitioners to understand the state-of-the-practice of BDA implementations in the Cloud. Second, based on the aforementioned investigation, this conceptual framework would significantly facilitate the identification of research gaps and opportunities in this domain. Third, and more importantly, by analysing and synthesising the practical elements, this conceptual framework would enable generating and evolving strategic approaches to guide implementing BDA applications in the Cloud.

The rest of the paper is organised as follows. Section II covers the related work. Section III presents a conceptual framework for big data analytics applications in the Cloud. Section IV evaluates the usefulness of the CF4BDA framework by applying CF4BDA to examine a set of papers. Section V concludes the paper and outlines the future work.

II. RELATED WORK

Currently, there have been a number of efforts to implement BDA applications in the Cloud with various focus of optimisation.

Some research work have been carried out on the development of BDA applications, which can be viewed as optimisation of BDA applications. G-Hadoop [11] is a fined-grained data processing framework that leverages MapReduce [16] implementations for large-scale distributed data processing across multiple data centres to achieve high throughput and fault tolerance. As BDA applications often run for long time which increases the possibility for being attacked, Bendahmane et al. [17] extend MapReduce to facilitate computation integration of the long-running BDA applications to avoid the possibility of being attacked.

More work has been done from the perspective of the running environment of BDA applications, which are actually within the scope of deployment. Jayalath et al. [12] present data transformation graphs that can be used to execute jobs on geo-distributed data sets in a way that minimises job execution time and cost. Gu et al. [13] propose a joint-optimization based algorithm to address the problem of cost minimisation by jointly considering data placement, task assignment and data routing. AROMA [18] is a system that automates Hadoop configuration and resource allocation to achieve high performance while minimising cost. Loughran et al. [14] propose a MapReduce architecture that enables dynamic deployment of BDA applications on different combination of public and private Clouds in a declarative way.

Since there are many aspects of BDA application engineering worth to concern, the current work is scattered and confusing across diverse directions, e.g. there are overlaps between optimisation and deployment. When developing and deploying BDA applications in the Cloud, there is a lack of big pictures which can be used to understand the existing concerns.

A couple of surveys have been performed in the area of BDA application engineering. Sakr et al. investigated the existing approaches on deploying data-intensive applications in the Cloud [19] and provided a comprehensive survey the current large scale data processing mechanisms based on the original MapReduce [20]. The BDA process proposed in [1] is too general, which can hardly be used to guide BDA engineering. To our best knowledge, there is still a lack of work that systematically analyses engineering of BDA applications in the Cloud, which proves the usefulness of our conceptual framework. Our work provides a big picture which help researchers and practitioners understand the concerns about BDA implementations and guide building BDA applications in the Cloud.

III. A CONCEPTUAL FRAMEWORK FOR BDA APPLICATIONS IN THE CLOUD

We developed this conceptual framework CF4BDA along two dimensions, and both dimensions were designed to help implement Cloud-based BDA applications, as explained in the following subsections respectively.

A. LIFE CYCLE OF BDA APPLICATIONS

The first dimension of CF4BDA is about the generic life cycle of BDA applications. Although each life phase defined in software engineering might further comprise a set of specific steps, we mainly consider nine stages in a BDA implementation, as illustrated in Fig. 1. The rectangles represent implementation stages of a BDA application while arrows represent transitions between stages. Our consideration has been largely adapted to the current investigations into BDA. For example, we particular ignored the stage requirement specification, because most of the existing BDA studies were based on predefined BDA requirements as their motivations. Nevertheless, we can naturally grow this dimension by adding other implementation stages if needed in the future.

(1) *Development:* Given particular BDA requirements, the developers build a BDA application by gradually

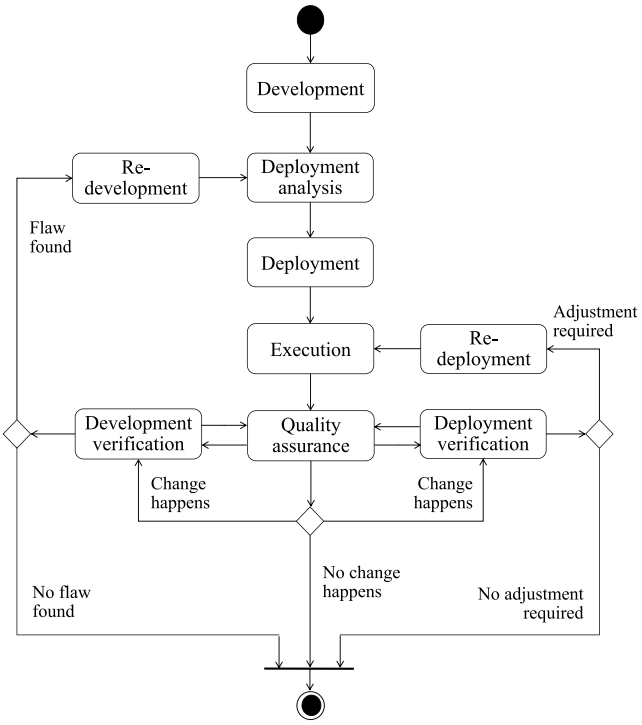


FIGURE 1. Life cycle of a BDA application.

implementing its predefined logic and functionality. The logic and functionality keep being tested during development with small sets of sample data.

(2) *Deployment Analysis:* Once there is an executable version of the BDA application, the developers and the operators analyse the characteristics of the BDA application, source data, data processing platforms and candidate Cloud infrastructures. The aim of this stage is to find the optimal deployment architecture and optimal deployment procedure including job execution sequences, individual deployment steps and parameter settings at each step.

(3) *Deployment:* Based on comprehensive analyses, the operators place the BDA application in an execution environment with appropriate settings to make sure that the BDA application is ready to work. Note that the execution environment indicates either a local/pseudo Cloud environment for testing or a real Cloud environment for production.

(4) *Execution:* Once the deployment is completed in the execution environment, the logic and functionality of the BDA application can be triggered at any time. The predefined instructions essentially drive the data processing and the task running.

(5) *Quality Assurance (QA):* The QA engineers monitor the development and testing processes and check whether the application performance and/or results meet pre-specified requirements. In particular, administrative tools can be developed and used to record development activities and application execution into logs. If the application quality is not satisfying, the monitoring further drives the verification of development and deployment to identify root causes. For the purpose of error prevention, in particular, quality assurance can directly include development verification and deployment verification.

(6) *Development Verification:* Usually driven by the unsatisfying application quality, the developers and the QA engineers examine if there is any flaw related to the development process and activities, for example, by review meeting and checking relevant source codes.

(7) *Deployment Verification:* Usually driven by the unsatisfying application quality found by performance testing, the operators and the QA engineers examine if there is any flaw related to the deployment process and activities, for example, by review meeting and identifying unusual execution records in the log.

(8) *Re-Development:* If any development flaw is found during development verification or any requirement change is happened to the BDA application, the developers should redevelop the BDA application. Meanwhile, the QA engineers can help improve the development and testing processes if necessary.

(9) *Re-Deployment:* If any deployment flaw is found during deployment verification or any environmental change is happened to the BDA application, the operators should redeploy the BDA application. Meanwhile, the QA engineers can help improve the deployment process or parameter settings if necessary.

B. CLASS DIAGRAM OF BDA APPLICATIONS

The second dimension of CF4BDA is about the elementary entities of a BDA application. Following the object-oriented thinking, we identified ten main objects and their relationships in the context of Cloud-based BDA applications, and represented them into a class diagram by using the Unified Modeling Language (UML), as shown in Fig. 2. Furthermore, we classify the identified objects into three layers, namely the application layer, the platform layer, and the infrastructure layer, as specified below.

1) APPLICATION LAYER

We investigate the application layer by considering the workflow of a BDA application. In fact, it has been realized that developers can apply workflow approaches to data analytics design to address the complexity of scientific and business BDA applications. In particular, a data analytics workflow "encompasses all the steps of data analytics, from data access and filtering to data mining and sharing produced knowledge" [4].

Although there could be various workflow techniques and designs [4], a common BDA workflow nowadays is typically driven by MapReduce [16]. Since proposed by Google in 2004, MapReduce has become the standard programming model for BDA applications. As shown in Fig. 3, the key steps in a MapReduce workflow are: (1) The initial input source data are segmented into blocks according to the

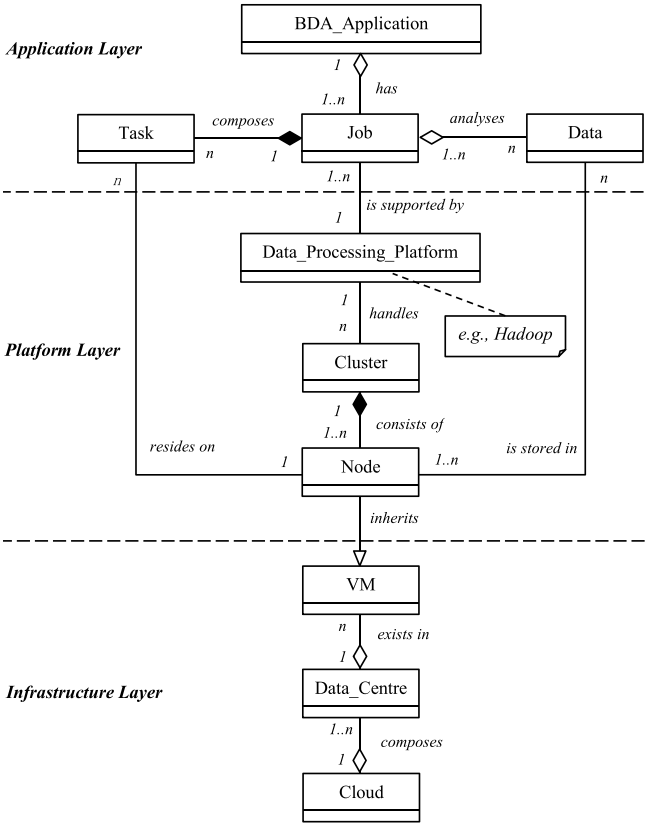


FIGURE 2. Class diagram in the Unified Modeling Language (UML) for understanding BDA applications in the Cloud.

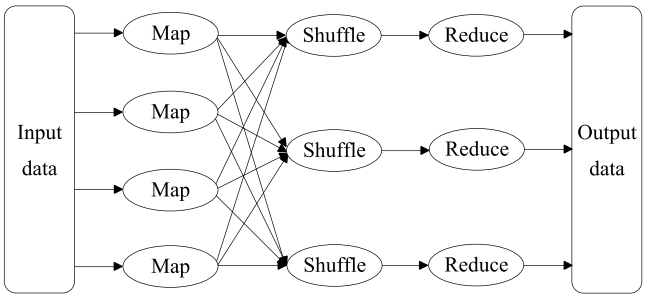


FIGURE 3. Demonstration of a MapReduce workflow.

predefined split function and saved as a list of key-value pairs. (2) The mapper executes the user-defined map function which generates intermediate key-value pairs. (3) The intermediate key-value pairs generated by mapper nodes is sent to a specific reducer based on the key. (4) Each reducer computes and reduces the data to one single key-value pair. (5) All the reduced data are integrated into the final result of a MapReduce job.

By referring to this typical BDA workflow, we focus on four internal objects of a BDA application that constitutes the application layer, i.e. BDA application, job, task, and data.

- *BDA Application:* A BDA application is a software application that processes large amounts of data using parallel data processing platforms to capture intended knowledge. A BDA application can have a single job or multiple jobs being processed either concurrently or sequentially.
- *Job:* A job is a batch process which performs data transforming and analytic computation following a predefined workflow. A job can be partitioned into numerous and various tasks, and then be accomplished by farming and harvesting the tasks.
- *Task:* A task is an atomic job unit operating on a slice of data. Individual tasks within a job can have diverse operations with different purposes. For example, driven by a MapReduce workflow, a task can be a Map, a Reduce, or a Shuffle operation. Since the atomic tasks are essentially components of a particular job, there is a composition relationship between classes task and job (cf. Fig. 2).
- *Data:* In a broad sense, data indicates a massive amount of complex and pervasive unstructured data that need to be analysed. Here we treat a data object particularly as a data slice being operated by a task. Unlike the object task that composes a job, data can exist independently and therefore there is an aggregation relationship between classes data and job (cf. Fig. 2).

2) PLATFORM LAYER

To facilitate implementing BDA in the Cloud, there are usually software middleware as platforms between BDA applications and the Cloud infrastructure. A platform provides a large-scale distributed processing capability that can help operators to deploy, execute and manage BDA applications. A well-known and de facto BDA platform is Apache Hadoop. Thus, we take Hadoop as a concrete example to investigate potential platform objects. In particular, we mainly focus on the interactions between a BDA application and such a platform, instead of being concerned with Hadoop's ecosystem that includes various modules and components.

As illustrated in Fig. 4, the platform can be further divided into a distributed processing layer (i.e. MapReduce that corresponds to the aforementioned typical BDA workflow) and a distributed file-system layer (i.e. Hadoop Distributed File System (HDFS) in this case). When dealing with a BDA application, the main interactions are: (1) A MapReduce

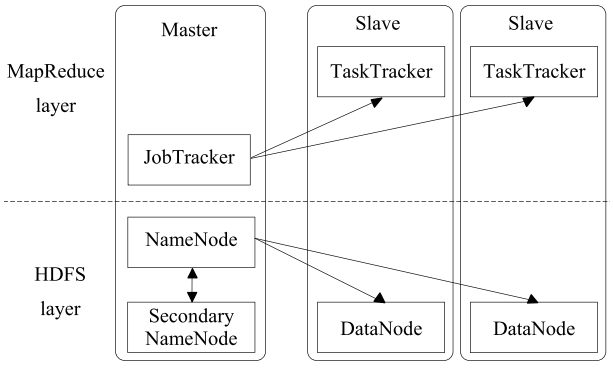


FIGURE 4. Demonstration of a Hadoop cluster.

job can run in a Hadoop cluster. (2) The JobTracker in the cluster accepts a job from the BDA application, and locates relevant data through the NameNode. (3) Suitable TaskTrackers are selected and then accept the tasks delivered by the JobTracker. (4) The JobTracker communicates with the TaskTrackers and manages failures. (5) When the collaboration among the TaskTrackers finish the job, the JobTracker updates its status and returns the result.

Considering that the JobTracker, TraskTrakcer, NameNode and DataNode are all logic nodes in the platform, we highlight three abstract objects in those interactions, namely data processing platform, cluster, and node.

- *Data Processing Platform:* A data processing platform refers to a software middleware that can help manage a distributed computing environment for storing and analysing large amounts of data. Note that the platform object is in a generic sense and it does not have to be Hadoop.
- *Cluster:* A cluster consists of a group of logically interconnected nodes that work together to store and analyse unstructured data. Different nodes in a cluster can play different roles in the collaboration on completing a data analytics job. Since these roles are constrained to be in the same cluster, the nodes and their cluster have a composition relationship (cf. Fig. 2).
- *Node:* A node is a logic unit where the data processing platform manages and stores data, as well as schedules and executes tasks on top of VMs. As indicated in the class diagram (cf. Fig. 2), the derived class node inherits the base class VM, and it can be specialized into different roles through the inheritance. In practice, it is common to locate multiple nodes with different roles in the same VM. For example, a TaskTracker node and a data node can coexist together to reduce the communication overhead.

3) INFRASTRUCTURE LAYER

When it comes to the infrastructure layer, we are concerned with both virtual and physical infrastructures in a Cloud. In general, Cloud infrastructural resources can be specifically distinguished between computation, communication (network), memory, and storage. Considering the relation to the node in the platform layer, however, we particularly extract the object VM that integrates the above four types of Cloud resources. The other two objects we have identified in the infrastructure layer are data centre and cloud, as shown in Fig. 5 and specified below.

- *VM:* As the name suggests, a virtual machine (VM) is an imitated server that runs on top of a shared pool of physical machines. The end user experience of a VM is the same as that of a physical machine. In fact, a predefined VM type can also be specified by a set of hardware indexes.
- *Data Centre:* A data centre is an organized facility that centralises IT operations and equipments to house both data and data-related systems. As mentioned previously, although a data centre comprises various computing resources, here we only highlight the resource type VM. Note that VM instances do not have to exist only in data centres, thus the classes VM and data centre have an aggregation relationship (cf. Fig. 2).
- *Cloud:* The Cloud is a shared pool of virtualised computing resources (e.g., servers, storage) that can be timely provisioned and released as a utility (just like electricity and water) and paid only for the consumed resources and workloads. For reasons such as disaster discovery or performance improvement, a Cloud might be constructed of many data centres in different geographical regions.

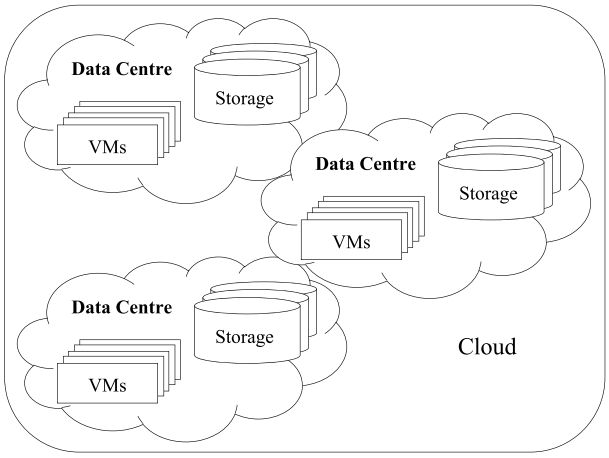


FIGURE 5. Demonstration of Cloud architecture.

IV. APPLICATION OF THE CONCEPTUAL FRAMEWORK CF4BDA

There are three application scenarios of applying our conceptual framework CF4BDA. First of all, by developing CF4BDA, we have identified and clarified relevant concepts, objects and their relationships. Thus, CF4BDA can in turn act as an anatomy of BDA application in the Cloud for new comers to get familiar with this domain-specific knowledge. Furthermore, benefiting from the clarified elements, CF4BDA can be used to systematically examine the existing relevant studies to help understand the available methods and techniques as well as their constraints and maturity levels when using them to implement BDA in the Cloud. More importantly, based on examining the existing practices, CF4BDA would be beneficial for both researchers and practitioners to investigate suitable solutions and develop automatic tools to support BDA implementations in the Cloud.

At this current stage, we have used CF4BDA to analyse a set of representative studies in the literature. The analysis results are briefly listed in Table 1, which can also be viewed as a preliminary evaluation of the usefulness of this framework. To avoid duplication, here we only highlight the identified BDA implementation techniques without re-elaborating their positions in the conceptual framework.

TABLE 1. Using the conceptual framework to identify the existing concerns and corresponding techniques about BDA implementation.

	Job	Data	Node	Cluster	Platform
Development	Programming model extension [21]	N/A	N/A	N/A	N/A
Re-Development	Pipeline development [22]	N/A	N/A	N/A	N/A
Development Verification	Result verification [17]	N/A	N/A	N/A	N/A
Deployment Analysis	Performance modelling [23]	Performance modelling [24]	Performance modelling [25]	Performance modelling [26]	Performance modelling [27]
Deployment	Job scheduling [28]	Data movement/placement [12]	Configuration optimisation [18]	Configuration optimisation [29]	Configuration optimisation [14]
Re-Deployment	Configuration optimisation [30]	Workflow management [31]	Configuration optimisation [32]	Configuration optimisation [33]	Workflow management [34]
Deployment Verification	Log analysis [35]	Log analysis [36]	Log analysis [37]	Log analysis [38]	Log analysis [2]
Execution	Configuration optimisation [39]	N/A	N/A	N/A	N/A
Quality Assurance	Monitoring [40]	Monitoring [41]	Monitoring [42]	Monitoring [43]	Monitoring [44]

1) PROGRAMMING MODEL EXTENSION

Programming model extension offers developers a new logical execution flow based on the existing BDA programming model (such as MapReduce), which is mainly focused on adding new features for building the logic and functionality of jobs (programs) at the development stage of the BDA application. The motivation is to comprehend the parallel processing capability of the existing programming model (such as processing distributed data across multiple Clouds) and improve certain quality attributes of the BDA applications (such as performance and security). Ding *et al.* [21] extend MapReduce with a simple communication mechanism that can effectively obtain certain shared information to reduce the time of processing unnecessary intermediate data.

2) CONTINUOUS DEVELOPMENT

Real-world BDA applications often require to process a pipeline of jobs. To enable efficient development of BDA applications with sequence jobs, continuous development focuses on adding coordination code to chain separate computation stages together and manage the results between pipeline stages. Google's FlumeJava [22] is a java library that wraps around MapReduce to optimise for better execution plans. Google has developed a runtime execution system for automatically running the selected optimised execution plans while managing low-level details.

3) RESULTS VERIFICATION

During or after development of the BDA applications, developers need to examine if there is any errors related to the development process and artefacts. Results verification refer to the mechanisms that can efficiently detect the errors related to programs for BDA jobs to guarantee high computation accuracy with an acceptable overhead. As BDA applications often execute for long time which increases the possibility for being attacked, [17] presents a mechanism that enables computation integration to improve the security of BDA applications.

4) PERFORMANCE MODELLING

Performance modelling refers to the behaviour modelling of BDA applications with various loads of jobs and data, as well as different settings of resources and platforms. For example, Hadoop has around 200 parameters that can be adjusted to affect BDA applications' performance. Numerous studies have examined the performance characteristics of BDA applications associated with different scopes (such as applications, clusters or processing platforms) [23]–[27].

5) JOB SCHEDULING

When multiple jobs are included in a BDA application, job scheduling is required to decide when and how to allocate resources for job execution and communication coordination to achieve performance goals. To understand the workload characteristics and optimise configuration for BDA applications, [28] conducts a comprehensive workload study using data collected from a 2000-node cluster at Taobao and introduce a job scheduling algorithm called Fair4S based on the observations derived from the collected data to optimise the job completion time.

6) DATA MOVEMENT/PLACEMENT

Data movement/placement refer to the mechanisms of transferring and storing data to optimise the quality of BDA applications. When the source data for BDA applications are distributed across different clouds, the location of data for computation have serious impact on the quality of BDA applications. Moving large amounts of data across different clouds may consume long time and cause data lost and source data explosion. To minimise job execution time and cost, [12] presents a framework that determines schedules for a given job sequence based on the proposed data

transformation graphs and executes the job sequence on geo-distributed data sets.

7) CONFIGURATION OPTIMIZATION

Configuration optimisation is defined as the selection of the most appropriate settings for all relevant objects including physical nodes, clusters, and processing platforms to ensure the overall quality of BDA applications. Various configuration optimisation approaches [14], [18], [29], [30], [32], [33], [39] are proposed to configure resources and/or processing platforms at runtime to optimise performance and cost taking into account the needs of job requests and resource availability.

8) WORKFLOW MANAGEMENT

Workflow management provides an infrastructure to setup, execute, and monitor the workflows of BDA applications which contain diverse jobs with the focus of continuity and scalability. Nova [31] is developed by Yahoo to deal with batched incremental processing and manage workflow by continually pushing coming data through Pig program graphs running on clusters. Oozie [34] is designed to satisfy the four major requirements for large scale BDA workflows: scale, multi-tenancy, processing platform security, and operability.

9) LOG ANALYSIS

Log analysis refers to examination and analysis of the generated logs during deployment in order to verify whether there are any deployment errors happened. To improve the effectiveness of deployment verification, numerous log analysis approaches [2], [35]–[38] are proposed to abstracts the logs and reports the different behaviour of BDA applications between running with small sample data and real-life data in the Cloud.

10) MONITORING

Monitoring consists of a set of activities undertaken to measure technical QoS metrics (e.g., response time, throughput, availability) and/or business metrics (e.g., cost) to ensure correct computations and operators, as well as optimal business benefits. Various monitoring mechanisms and systems [40]–[44] are proposed to examine the traces of job processing with BDA applications at runtime in order to take further adaptation actions.

V. CONCLUSIONS AND FUTURE WORK

The paper proposes a conceptual framework called CF4BDA that systematically and holistically identifies the challenges and solutions of implementing BDA applications in the Cloud. The CF4BDA framework is constructed in two dimensions: the first dimension lists different stages to understand where the possible BDA implementation challenges might exist while the second dimension takes object-oriented thinking and uses different elements of BDA implementation to investigate what potential solutions could be adopted.

By developing CF4BDA, we have identified and clarified relevant concepts, objects and their relationships. Thus, CF4BDA can be used as an anatomy of BDA applications in the Cloud for researchers and practitioners to get familiar with this domain-specific knowledge. Furthermore, benefiting from the clarified elements, CF4BDA can be used to systematically examine the existing relevant studies to help understand the existing methods and techniques as well as their limitations and maturity levels for developing and deploying BDA applications in the Cloud. More importantly, CF4BDA would facilitate development of automatic engineering tools to support BDA implementations in the Cloud.

By analysing a set of representative papers related to BDA applications in the Cloud, we priliminarily evaluated the usefulness of CF4BDA. Since BDA implementation techniques are still maturing, it would be necessary and valuable to keep this conceptual framework up to date. Therefore, we will conduct a systematic literature review in the domain of Cloud-based BDA to both validate CF4BDA in further details and expand its dimensions in a smooth way.

REFERENCES

[1] D. Fisher, R. DeLine, M. Czerwinski, and S. Drucker, "Interactions with big data analytics," *Interactions*, vol. 19, no. 3, pp. 50–59, May/Jun. 2012.

[2] W. Shang, Z. M. Jiang, H. Hemmati, B. Adams, A. E. Hassan, and P. Martin, "Assisting developers of big data analytics applications when deploying on Hadoop clouds," in *Proc. 35th Int. Conf. Softw. Eng. (ICSE)*, San Francisco, CA, USA, May 2013, pp. 402–411.

[3] H. Yang, Z. Luan, W. Li, D. Qian, and G. Guan, "Statistics-based workload modeling for MapReduce," in *Proc. IEEE 26th Int. Parallel Distrib. Process. Symp. Workshops PhD Forum (IPDPSW)*, Shanghai, China, May 2012, pp. 2043–2051.

[4] D. Talia, "Clouds for scalable big data analytics," *Computer*, vol. 46, no. 5, pp. 98–101, May 2013.

[5] A. Stevenson. (Jan. 2, 2015). *How to Pick a Cloud Storage Provider*. [Online]. Available: http://www.v3.co.uk/v3-uk/analysis/2386102/how-to-pick-a-cloud-storage-provider

[6] A. Brust. (Nov. 2014). *Big Data Analytics in the Cloud: The Enterprise Wants it Now*. [Online]. Available: http://research.gigaom.com/2014/11/big-data-analytics-in-the-cloud-the-enterprise-wants-it-now/

[7] B. Cromwell. (Feb. 7, 2013). *How Much of Cloud Security is New and Different?* [Online]. Available: http://cybersecurity.learningtree.com/2013/02/07/how-much-of-cloud-security-is-new-and-different/

[8] W. Sobel *et al.*, "Cloudstone: Multi-platform, multi-language benchmark and measurement tools for Web 2.0," in *Proc. 1st Workshop Cloud Comput. Appl. (CCA)*, Chicago, IL, USA, Oct. 2008, pp. 1–6.

[9] A. Lenk, M. Menzel, J. Lipsky, S. Tai, and P. Offermann, "What are you paying for? Performance benchmarking for infrastructure-as-a-service offerings," in *Proc. 4th IEEE Int. Conf. Cloud Comput. (CLOUD)*, Washington, DC, USA, Jul. 2011, pp. 484–491.

[10] M. Zhang, R. Ranjan, A. Haller, D. Georgakopoulos, and P. Strazdins, "Investigating decision support techniques for automating cloud service selection," in *Proc. 4th IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Taipei, Taiwan, Dec. 2012, pp. 759–764.

[11] L. Wang *et al.*, "G-Hadoop: MapReduce across distributed data centers for data-intensive computing," *Future Generat. Comput. Syst.*, vol. 29, no. 3, pp. 739–750, Mar. 2013.

[12] C. Jayalath, J. Stephen, and P. Eugster, "From the cloud to the atmosphere: Running MapReduce across data centers," *IEEE Trans. Comput.*, vol. 63, no. 1, pp. 74–87, Jan. 2014.

[13] L. Gu, D. Zeng, P. Li, and S. Guo, "Cost minimization for big data processing in geo-distributed data centers," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 314–323, Sep. 2014.

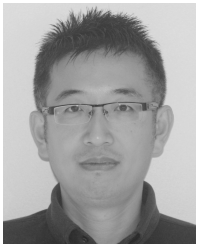
[14] S. Loughran, J. M. A. Calero, A. Farrell, J. Kirschnick, and J. Guijarro, "Dynamic cloud deployment of a MapReduce architecture," *IEEE Internet Comput.*, vol. 16, no. 6, pp. 40–50, Nov./Dec. 2012.

[15] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," School Comput. Sci. Math., Keele Univ., Keele, U.K., EBSE Tech. Rep. EBSE-2007-01, 2007.
[16] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
[17] A. Bendahmane, M. Essaaidi, A. El Moussaoui, and A. Younes, "Result verification mechanism for MapReduce computation integrity in cloud computing," in *Proc. Int. Conf. Complex Syst. (ICCS)*, Agadir, Morocco, Nov. 2012, pp. 1–6.
[18] P. Lama and X. Zhou, "AROMA: Automated resource allocation and configuration of MapReduce environment in the cloud," in *Proc. 9th ACM Int. Conf. Auto. Comput. (ICAC)*, San Jose, CA, USA, Sep. 2012, pp. 63–72.
[19] S. Sakr, A. Liu, D. M. Batista, and M. Alomari, "A survey of large scale data management approaches in cloud environments," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 311–336, Third Quarter 2011.
[20] S. Sakr, A. Liu, and A. G. Fayoumi, "The family of MapReduce and large-scale data processing systems," *ACM Comput. Surv.*, vol. 46, no. 1, Oct. 2013, Art. ID 11.
[21] L. Ding, G. Wang, J. Xin, X. Wang, S. Huang, and R. Zhang, "ComMapReduce: An improvement of MapReduce with lightweight communication mechanisms," *Data Knowl. Eng.*, vol. 88, pp. 224–247, Nov. 2013.
[22] C. Chambers *et al.*, "FlumeJava: Easy, efficient data-parallel pipelines," *ACM SIGPLAN Notices*, vol. 45, no. 6, pp. 363–375, Jun. 2010.
[23] S. Aggarwal, S. Phadke, and M. Bhandarkar, "Characterization of Hadoop jobs using unsupervised learning," in *Proc. 2nd IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Indianapolis, IN, USA, Nov./Dec. 2010, pp. 748–753.
[24] F. Zhang and M. Sakr, "Cluster-size scaling and MapReduce execution times," in *Proc. 5th IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Bristol, U.K., Dec. 2013, pp. 240–249.
[25] Z. Zhang, L. Cherkasova, and B. T. Loo, "Performance modeling of MapReduce jobs in heterogeneous cloud environments," in *Proc. IEEE 6th Int. Conf. Cloud Comput. (CLOUD)*, Santa Clara, CA, USA, Jun. 2013, pp. 839–846.
[26] E. Feller, L. Ramakrishnan, and C. Morin, "Performance and energy efficiency of big data applications in cloud environments: A Hadoop case study," *J. Parallel Distrib. Comput.*, vols. 79–80, pp. 80–89, May 2015.
[27] X. Wu, Y. Liu, and I. Gorton, "Exploring performance models of Hadoop applications on cloud architecture," in *Proc. 11th Int. ACM SIGSOFT Conf. Quality Softw. Archit. (QoSA)*, Montreal, QC, Canada, May 2015, pp. 93–101.
[28] Z. Ren, J. Wan, W. Shi, X. Xu, and M. Zhou, "Workload analysis, implications, and optimization on a production Hadoop cluster: A case study on Taobao," *IEEE Trans. Services Comput.*, vol. 7, no. 2, pp. 307–321, Apr./Jun. 2014.
[29] H. Mao, Z. Zhang, B. Zhao, L. Xiao, and L. Ruan, "Towards deploying elastic Hadoop in the cloud," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Beijing, China, Oct. 2011, pp. 476–482.
[30] Y. Kang, Y. Zhou, Z. Zheng, and M. R. Lyu, "A user experience-based cloud service redeployment mechanism," in *Proc. IEEE 6th Int. Conf. Cloud Comput. (Cloud)*, Washington, DC, USA, Jul. 2011, pp. 227–234.
[31] C. Olston *et al.*, "Nova: Continuous Pig/Hadoop workflows," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, Athens, Greece, Jun. 2011, pp. 1081–1090.
[32] Y. Yao, J. Wang, B. Sheng, C. Tan, and N. Mi, "Self-adjusting slot configurations for homogeneous and heterogeneous Hadoop clusters," *IEEE Trans. Cloud Comput.*, in press.
[33] N. Maheshwari, R. Nanduri, and V. Varma, "Dynamic energy efficient data placement and cluster reconfiguration algorithm for MapReduce framework," *Future Generat. Comput. Syst.*, vol. 28, no. 1, pp. 119–127, Jan. 2012.
[34] M. Islam *et al.*, "Oozie: Towards a scalable workflow management system for Hadoop," in *Proc. 1st ACM SIGMOD Workshop Scalable Workflow Execution Engines Technol. (SWEET)*, Scottsdale, AZ, USA, Apr. 2012, pp. 1–10.
[35] J. Tan, S. Kavulya, R. Gandhi, and P. Narasimhan, "Visual, log-based causal tracing for performance debugging of MapReduce systems," in *Proc. IEEE 30th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Genoa, Italy, Jun. 2010, pp. 795–806.
[36] J. Tan, X. Pan, S. Kavulya, R. Gandhi, and P. Narasimhan, "Mochi: Visual log-analysis based tools for debugging Hadoop," in *Proc. Workshop Hot Topics Cloud Comput. (HotCloud)*, San Diego, CA, USA, Jun. 2009, pp. 1–5.
[37] J. Tan, X. Pan, E. Marinelli, S. Kavulya, R. Gandhi, and P. Narasimhan, "Kahuna: Problem diagnosis for MapReduce-based cloud computing environments," in *Proc. 12th IEEE/IFIP Int. Netw. Oper. Manage. Symp. (NOMS)*, Osaka, Japan, Apr. 2010, pp. 112–119.
[38] E. Chuah, A. Jhumka, S. Narasimhamurthy, J. Hammond, J. C. Browne, and B. Barth, "Linking resource usage anomalies with system failures from cluster log data," in *Proc. IEEE 32nd Int. Symp. Rel. Distrib. Syst. (SRDS)*, Braga, Portugal, Sep. 2013, pp. 111–120.
[39] X. Hou, A. Kumar, J. P. Thomas, and V. Varadharajan, "Dynamic workload balancing for Hadoop MapReduce," in *Proc. IEEE 4th Int. Conf. Big Data Cloud Comput. (BDCloud)*, Dec. 2014, pp. 56–62.
[40] J. Conejero, O. Rana, P. Burnap, J. Morgan, B. Caminero, and C. Carrión, "Analyzing Hadoop power consumption and impact on application QoS," *Future Generat. Comput. Syst.*, in press.
[41] B. Balis, M. Bubak, and M. Pelczar, "From monitoring data to experiment information—Monitoring of grid scientific workflows," in *Proc. IEEE Int. Conf. e-Sci. Grid Comput. (E-SCIENCE)*, Bengaluru, India, Dec. 2007, pp. 77–84.
[42] M. Mohandas and P. M. Dhanya, "An approach for log analysis based failure monitoring in Hadoop cluster," in *Proc. Int. Conf. Green Comput., Commun., Conservation Energy (ICGCE)*, Chennai, India, Dec. 2013, pp. 861–867.
[43] C. Delimitrou and C. Kozyrakis, "Quasar: Resource-efficient and QoS-aware cluster management," in *Proc. 19th Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLOS)*, Salt Lake City, UT, USA, Mar. 2014, pp. 127–144.
[44] X. Wu, Y. Liu, and I. Gorton, "Scalability and cost evaluation of incremental data processing using Amazon's Hadoop service," in *Big Data—Algorithms, Analytics, and Applications*. London, U.K.: Chapman & Hall, 2015, pp. 21–38.

QINGHUA LU received the Ph.D. degree from the University of New South Wales, in 2013. She is currently a Lecturer with the Department of Software Engineering, China University of Petroleum, Qingdao, China. Her research interests include software architecture, dependability of cloud computing, big data architecture, and service computing.

ZHENG LI received the Ph.D. degree from Australian National University. He is currently a Post-Doctoral Researcher with the Department of Electrical and Information Technology, Lund University, Sweden. His research interests include cloud computing, performance engineering, empirical software engineering, software cost/effort estimation, and Web service composition.

MARIA KIHL is currently a Professor of Internetworked Systems with the Department of Electrical and Information Technology, Lund University, Sweden. Her work focuses on performance modeling, analysis, and control of distributed Internet-based systems, currently cloud systems and media distribution architectures.



LIMING ZHU received the Ph.D. degree in software engineering from the University of New South Wales (UNSW). He worked in several technology lead positions in the software industry. He was the Research Group Leader of the Software Systems Research Group with National ICT Australia, Australia. He holds conjoint positions with UNSW and the University of Sydney. He is currently the Research Director of the Software and Computational Systems Program with Data61, Commonwealth Scientific and Industrial Research Organisation, Australia. His research interests include data analytics platforms and infrastructures, software engineering, and distributed systems.

WEISHAN ZHANG is currently a Full Professor and the Deputy Head for research with the Department of Software Engineering, China University of Petroleum. He has authored over 50 papers, and his current h-index according to Google scholar is 10. His current research interests are big data platforms, pervasive cloud computing, and service oriented computing.