Journal of Cloud Computing
a SpringerOpen Journal

RESEARCH Open Access

Enabling actionable analytics for mobile devices: performance issues of distributed analytics on Hadoop mobile clusters

Seungbae Lee*, Kanika Grover and Alvin Lim

Abstract

Significant innovations in mobile technologies are enabling mobile users to make real-time actionable decisions based on balancing opportunities and risks to take coordinated actions with other users in their workplace. This requires a new distributed analytic framework that collects relevant information from internal and external sources, performs real-time distributed analytics, and delivers a critical analysis to any user at any place in a given time frame through the use of mobile devices such as smartphones and tablets. This paper discusses the advantages and challenges of utilizing mobile devices for distributed analytics by showing its feasibility with Hadoop analytic framework.

Keywords: Mobile device; Mobile cloud; Distributed analytics; Performance analysis; Android; MapReduce; Hadoop

Introduction

Many IT industry analysts predict that ubiquity of mobile systems is the most significant trend in the near future. According to International Data Corporation (IDC) estimates, the recent surge in demand for mobile systems will lead to mobile devices surpassing PCs as the method of choice for online access [1]. Market growth of smartphones and tablet computers will outpace PCs in 2013. The number of U.S. mobile users will outnumber those using PCs to access the Internet in 2015. Gartner also forecasts sales of 1.2 billion mobile devices in 2013, including smartphones and tablets, a 50 percent increase over 2012 [2].

The rising demand for mobile devices and mobility-related services has led smartphones and tablets to become far more powerful. Mobile devices with a quad-core 1.9 GHz processor and 2 GB memory are already widely available. Even octa-core mobile processors and 3 GB memory are planned for release at the end of this year. Along with enhancements in battery capacity and network capabilities, mobile devices are now capable of sharing their resources for distributed processing of critical analytics as resource providers of cloud computing.

Significant innovations in mobile technologies enable mobile workers to make real-time decisions based on balancing opportunities and risks in order to support collaboration with other workers through coordinated actions. These actions can be generated by mobile ad hoc actionable analytics that may consist of simulation, prediction, and/or optimization. This capability leads to more flexible decision-making that can be optimized for a specific scenario at a certain time and place.

The actionable analytics requires the ability to analyze all relevant data from internal or external sources and deliver a critical analysis to any individual at any location within the expected response time. This creates a great challenging opportunity for driving a new distributed analytic method based on the convergence of the latest mobile technologies. Gartner included actionable analytics among the top technologies that will be strategic for most organizations in 2013 [3].

To lay the groundwork for building a new mobile distributed processing framework for actionable analytics, Apache Hadoop [4] can serve as a good starting point for the mobile analytic platform. It is an open-source software framework for distributed processing of large unstructured data sets. Hadoop, based on Google MapReduce [5] and Google distributed File System (GFS) [6], has become

*Correspondence: sblee@auburn.edu
Department of Computer Science and Software Engineering, 3101 Shelby Center for Engineering Technology, Auburn University, Auburn, AL 36849-5347, USA

the de facto standard tool for distributed data mining in the academic and industrial world.

Unlike earlier approaches, this study examines the performance of Hadoop mobile clusters by conducting distributed analytics using typical Hadoop benchmarks with a CPU, memory and/or I/O intensive workload. The newest release of Hadoop software framework with its enhancements is entirely ported to the latest Android-based mobile devices (e.g. Google NEXUS 7 tablet) without degradation of the system performance and side effects on the Android operations.

Through performance analysis, it is observed that the overall computing power of the mobile cluster is no longer significantly bounded by typical processing capabilities of each individual mobile node (e.g. CPU speed and memory capacity) since the processing power of mobile devices has been constantly enhanced. On the other hand, the performance of distributed computing in the mobile cluster is strongly influenced by reliable network capabilities for continuous data interchange between mobile nodes.

Current distributed systems including Hadoop employ Transmission Control Protocol (TCP) to reliably collect input data from remote sources and deliver analytic results to destinations. The performance of mobile distributed analytics largely relies on how effectively every mobile device utilizes the available network resource through TCP connections. Despite advances in mobile technologies, mobile devices still face significant limitations on transmitting and receiving constant TCP data streams necessary to provide users with seamless services.

The goal of this study is to discuss the advantages of using mobile devices for distributed analytics by showing its feasibility with today's mobile technologies. We also investigate design challenges and considerations for mobile distributed applications, focusing on critical performance issues of reliable data communications between mobile devices for interchanging large amount of analytical data and monitoring real-time status of cluster nodes.

The rest of the paper is organized as follows. We first introduce the motivation and challenges of mobile ad hoc analytic frameworks for mobile devices in Section "Motivation and challenges" and then summarize related work in Section "Related work". Section "Performance analysis of Hadoop mobile clusters" provides our experimental observations on the performance of Hadoop mobile clusters with details of the experimental setup and Section "Performance issues of mobile cloud clusters" discusses performance issues of mobile distributed analytics. We conclude this paper in Section "Conclusion".

Motivation and challenges

With the increasing popularity of mobile devices, a growing number of organizations are adopting "Bring Your Own Devices" (BYOD) policies [7]. With BYOD, workers bring their own mobile devices to their workplace and use those devices to access privileged information and run applications of their organization. It provides a great opportunity for improving productivity by accelerating the speed of decision-making and problem-solving. This section describes the mobile trends in workplace, which are the motivation of this study.

Mobile device capabilities

The strong demand for diverse mobile devices has led smartphones and tablets to offer the latest advanced features. Mobile platforms are leveraging multi-core processor architecture to dramatically increase processing power at a low cost. Manufacturers are introducing high-speed memory and increasing storage capacity for mobile devices. Moreover, advances in battery capacity and power saving techniques enable mobile devices to support large complex computations and long-running processes and provide more reliable high-speed wireless connectivity with more optional features, including 4G LTE, Wi-Fi, Bluetooth, and Near Field Communication (NFC).

Furthermore, recent mobile devices provide innovative visual interaction through the use of advanced high-resolution touchscreens. Also, they integrate a variety of sensors that are constantly improved, including the microphone, image sensor, 3-axis accelerometer, gyroscope, atmosphere pressure sensor, digital compass, optical proximity sensor, ambient light sensor, humidity sensor and touch sensor. These cutting edge touchscreens and sensors enable mobile devices to monitor the operating environment in real time and adapt to the situation accordingly.

Mobility in workplace

Today, people are connected in more ways than before. By using mobile devices, they no longer need to sit in front of a desktop PC at office or at home in order to communicate with others. People are making new connections anywhere, anytime, and on any device. As a result of the use of mobile equipment, mobility is having a huge impact on the way people work. A recent IDC report shows the world's mobile worker population will reach 1.3 billion, representing 37.2 % of the total workforce by 2015 [8]. The number of mobile workers in the U.S. will grow from 182.5 million in 2010 to more than 212.1 million by 2015.

The increasingly mobile and remote workforce is driving organizations to support a wide range of mobile applications and services, which enables workers to proactively detect and collect more information from internal and external sources by using mobile devices, and perform real-time analytics for rapid decision, thus improving collaboration and productivity. A Gartner's report shows

that more than 40 percent of all enterprise web applications will support mobile environments by 2014 and 90 percent of companies will support corporate applications on mobile devices by 2014 [3].

Mobile cloud computing

The increasing number of mobile applications require more complex processing and more operational data. These applications include real-time mobile analytics that enhances situational awareness, risk assessment, distributed decision making, coordinated action planning, team collaboration, and instant responsiveness. Despite the increasing use of mobile devices, exploiting its full potential is difficult due to the inherent problems such as resource limitations (e.g. computational capability and battery capacity) and frequent disconnections from mobility.

Mobile cloud computing can solve these problems cost-effectively by utilizing computing and storage capabilities from remote resource providers or other mobile devices. Although current cloud applications that connect to a remote infrastructure are becoming popular, they can perform well only under high speed connectivity. It is not practical to assume high-speed connections, seamless handovers, and fast responses on mobile devices. Thus, clustering with nearby mobile devices will promise faster connectivity and better availability. This study focuses on mobile ad hoc cloud where the remote resources are mobile and available only within the range of the wireless transmission.

Actionable analytics

Conventional explanatory analytics usually focused on what happened in the past. Such analytics may be outdated and ineffective approaches that do not offer timely, accurate, and actionable insights needed for distributed decision making and coordinated action planning today. What is happening now? What is going to happen in the future? The ability to answer these questions in real time or near real time can provide a competitive advantage.

Recent advances in mobile technologies enable mobile users to collaborate with their network team through coordinated actions by balancing opportunities and risks. These actions can be generated by ad hoc distributed analytics that may consist of simulation, prediction, and/or optimization. This capability leads to a great opportunity for reducing cost while improving outcomes through more flexible decision-making that can be optimized for a specific scenario at a certain time and place [3].

Challenges of mobile distributed analytics

Despite recent advances in mobile technologies and analytic methods, mobile devices still face great challenges in delivering distributed analytics to mobile users without interruptions. The challenges include the followings:

- Reliable access to remote resources is the first challenge since the analytic information is commonly distributed across a variety of remote sources. The mobile environment is subject to the high probability of disruptions due to mobility, where fixed infrastructure is frequently unavailable and network partitions are common.
- Given that wireless communication bandwidth is relatively low, collecting large data sets from various source systems in a short or limited period of time, integrating the data into a combined view using distributed computing resources, and delivering analytic results to mobile destinations without delay are all difficult challenges.
- A mobile device that initiates distributed analytics needs to dynamically take advantage of mobile cloud resources depending on specific requirements of workload since the internal status and the external environment are subject to change. Monitoring and scheduling of available resources are the most critical operations for mobile distributed computing.

Related work

Many researchers have identified key attributes, technologies, and challenges that distinguish cloud computing from traditional computing paradigms [9-13]. Briefly, cloud computing provides reliable, customizable and dynamic computing environments with Quality of Service (QoS) guarantee for end-users [14]. Also, many studies have focused on mobile cloud services on the Internet as summarized in [15,16].

This study pays particular attention to the performance of mobile ad hoc cloud, where ad hoc networks of mobile devices themselves work as resource providers of the cloud as in [17]. In this type of cloud, the workload and data reside on individual mobile devices rather than on remote resource servers. The following studies have tried to implement ad hoc resource clouds using practical mobile devices.

Hyrax [18,19] explores the feasibility of using a cluster of mobile phones as resource providers by porting Hadoop to Android smartphones. For a sample application, they present a simple distributed mobile multimedia search and sharing program. However, their performance evaluations for the mobile ad hoc cloud are limited since they completed only a partial implementation of the Hadoop architecture, where many core features were removed due to difficulties and obstacles in Hadoop migration. Even the major controllers of Hadoop framework, such as JobTracker for MapReduce and NameNode for HDFS (Hadoop Distributed File System), are not installed on

the mobile node. A similar approach to implementing the Hadoop framework on mobile devices is found in [20].

Serendipity [21,22] discusses the challenges of remote computing using mobile devices and introduces a framework that enables a mobile computation initiator to use remote computational resources available on mobile devices. They implement an actual prototype on Android devices and evaluate their system using two sample applications, a face detection application and speech-to-text application. However, no performance comparison with existing distributed frameworks is made. Another study, Cuckoo [23], proposes a computation offloading framework for Android smartphones and illustrates its utility with an application for multimedia content analysis.

In short, several studies on the ad hoc analytic framework for mobile devices have been conducted by implementing only part of an existing distributed analytic framework or by proposing a customized framework similar to the existing one. The previous studies are mostly evaluated using just one or two domain-specific applications and fail to provide comparative analysis of their performance and efficiency. To the best of our knowledge, there has been no comparable framework and performance analysis for practical mobile cloud clusters running distributed analytic applications.

Although we mostly focus on the performance of practical distributed analytics on mobile cloud clusters in terms of job processing time and response time, some studies concentrate on energy efficiency which is a key aspect to enable data analysis and mining over mobile devices. For example, [24] proposes an energy-aware scheduling strategy that assigns data mining tasks over a mobile cluster to optimize energy utilization. Our future work should take into account efficient power utilization for mobile distributed analytics.

Performance analysis of Hadoop mobile clusters

When reviewing multiple data analytic models, we found that Apache Hadoop can provide a good starting point for mobile distributed analytics since it supports cost-effective and high performance analysis of a large volume of unstructured data on a set of commodity hardware. This section examines the performance of Hadoop distributed processing in practical mobile cluster setups.

Overview of Apache Hadoop

Apache Hadoop [4] is an open-source framework that uses a simple distributed processing model based on Google MapReduce [5] and Google file system (GFS) [6]. It effectively handles massive amount of information by either transforming it to a more useful structure and/or format, or extracting valuable analytics from it. Hadoop runs on any machines equipped with a lower cost processor and storage, and automatically recovers from hardware, software, and system failures by providing fault tolerance through software. Therefore, Hadoop is more cost-effective for handling large unstructured data sets than conventional data mining approaches. Moreover, Hadoop offers great scalability and high availability.

Google MapReduce is the fundamental software programming model in the Hadoop architecture, which performs distributed processing of large data sets on a computing cluster. A single large workload (job) is divided or mapped into smaller sub-workloads (tasks) to be processed in parallel. The results from the sub-workloads are merged, condensed, and reduced to produce the final result. Both input and output are stored on the nodes throughout the cluster in the distributed file system known as Google file system.

Numerous factors can affect the performance of the Hadoop cluster. The typical performance factors such as workload type, cluster size, input/output data size, and characteristics of computing nodes (e.g. CPU, Memory, and I/O resources) have significant impacts on the processing time. In addition, the network is also a critical factor on the Hadoop performance since the nodes are interconnected through the network in order to transfer data for distributed processing during one or more phases of MapReduce execution consisting of Map, Shuffle, Reduce, and optional Replication phase, as illustrated by Figure 1.

This paper omits some details of Hadoop framework which can be found in many papers on key techniques, data mining algorithms, and performance analysis, some of which (e.g. [25-27]) are referenced in this paper.

Assumptions on mobile cloud

This study performs experiments based on the following assumptions on the basic, common configuration of practical mobile cloud clusters for ad hoc analytics. However, our future work will consider extensive scenarios that include dynamic node mobility and various analytic workloads under actual mobile environments.

- Mobile devices may process computational workload that exceeds their capability by offloading portions of the workload to remote resources for distributed execution. All mobile devices are capable of sharing their computing resources, and behave in a collaborative and trustworthy manner.
- Clustering with nearby mobile devices to build a mobile ad hoc cloud provides faster connectivity and better availability because the actual connectivity with typical remote cloud infrastructures may be intermittent and unpredictable due to the mobility of mobile devices.
- All mobile nodes belonging to a cluster move in the same direction and at the same speed when

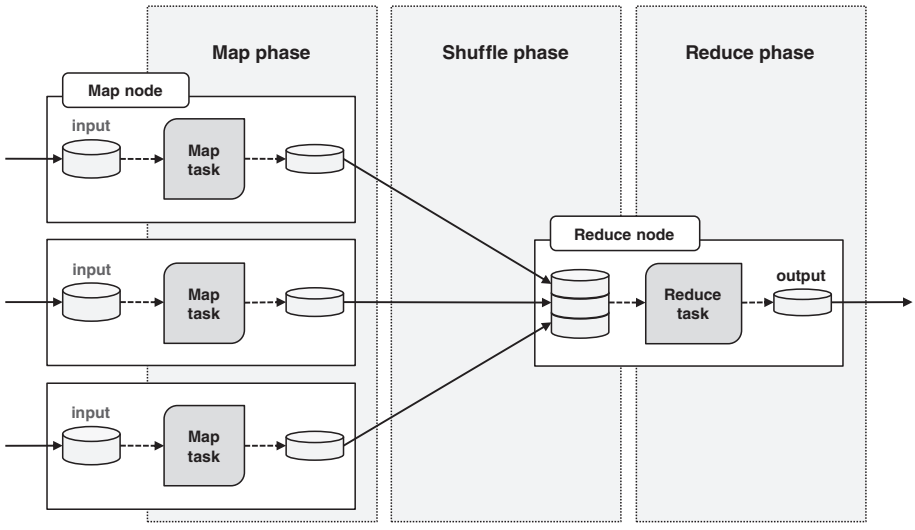

Figure 1 Data flows of Google MapReduce.

processing the workload, i.e. there is no significant change in network connectivity. This assumption is needed for this study to evaluate the distributed processing capability of the mobile cluster under reliable connectivity.

- The mobile cluster runs a single workload at a time, either transforming the unstructured input data to a more useful structure without adding new data, or extracting small but valuable analytics from the input data. The amount of intermediate and output data generated by mobile devices depends on the type of workload.

Experimental setup

In the experiments, we measured the performance of Hadoop clusters using Android-based mobile platforms including smartphones (e.g. Samsung Galaxy S2, Google Galaxy NEXUS), media players (e.g. Samsung Galaxy player), and tablets (e.g. Samsung Galaxy Tab, Google NEXUS 7) under extensive distributed configurations. This paper presents experimental results from one of those cluster setups, which consists of eleven NEXUS 7 tablets developed by Google in conjunction with Asus. Figure 2 displays the experimental mobile cluster with Google NEXUS 7 tablets.

The experimental platform, NEXUS 7, is the first tablet in the Google Nexus series that implements the Android operating system. The Nexus 7 features a 7-inch display, NVIDIA Tegra 3 quad-core processor, 1 GB of memory, and 16 GB of internal storage, and incorporates built-in Wi-Fi, Bluetooth, and NFC connectivity [28]. The tablet runs the latest Android operating system (version 4.2.2, nicknamed "Jelly Bean") and Hadoop stable release (version 1.1.2) with Oracle JDK (version 1.6) at the time of writing this paper. The detailed specifications of experimental platforms are listed in Table 1. All platforms are reliably interconnected with a Wi-Fi based wireless access point, Asus RT-N66U, in an IEEE 802.11n [29] infrastructure mode.

Porting Hadoop on the Android operating system was a big and significant challenge at the early stage of this study. Android supports the Dalvik process virtual machine for running mobile applications written in Java, but Hadoop software framework is not fully compatible with this runtime environment. Thus, Hadoop can be ported by either converting from Java Virtual Machine (JVM) compatible source codes and libraries to Dalvik compatible ones or installing a specific JVM recommended by the Hadoop project to run the original Hadoop software.

Most of previous work [18-20] had difficulties with rewriting Hadoop codes for Android. They implemented

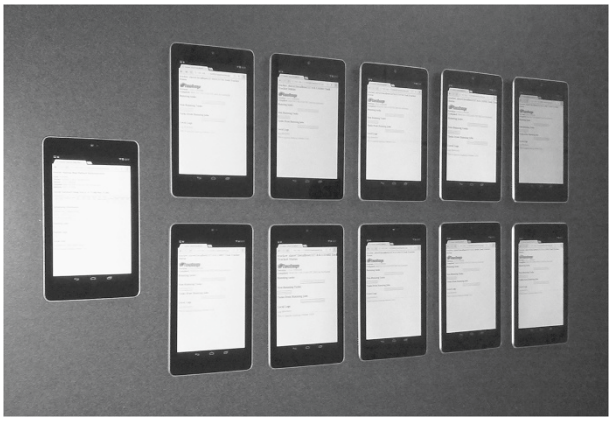

Figure 2 Experimental mobile cluster using Google NEXUS 7.

Table 1 Hardware and software specifications of experimental nodes

Platform	Google NEXUS 7
CPU	NVIDIA® Tegra® 3 quad-core processor
Memory	1GB, RAM
Storage	16GB, Nand flash
Network	Wi-Fi 802.11 b/g/n, Bluetooth, NFC
OS	Android 4.2, Jelly Bean (Build number: JDQ39)
Kernel	Linux 3.1.10
Linux extension	Ubuntu 12.04 for ARM
JVM	JDK 1.6.0_32 (Oracle Java SE for Embedded 6u32 ARMv7 Linux)
Hadoop	1.1.2 stable release
Resource monitoring	Sysstat 10.0.3-1 stable version

only a small number of Hadoop functions and removed many core features that are incompatible with the Dalvik environment. In contrast to earlier approaches, we successfully installed the Oracle JDK that is recommended for running Hadoop on the Linux-based Android operating system by adding a Linux extension [30], Ubuntu 12.04, to the Android kernel. We carefully ensured that there was no degradation of the hardware performance or adverse effect on Android operations. As a result, the experimental mobile cluster runs all existing and experimental features of the Hadoop architecture, including MapReduce 2.0, also known as YARN [25].

The mobile cluster that runs the Hadoop software consists of one Master node and ten Slave nodes which are configured with the default values for parameters of Android OS and Hadoop. The Master node coordinates the Slave nodes to get the workload done and the Slaves run the sub-workloads, Map and Reduce tasks, assigned by the Master node. The usage of computing and networking resources on each node is carefully monitored with a performance monitoring tool, Sysstat. To investigate node's behavior in the Hadoop workflow, two typical workloads – WordCount and TeraSort – are tested with associated Hadoop benchmark tools on the mobile cluster.

- WordCount: this workload counts the occurrence of each word in the input data sets generated by the Hadoop RandomTextWriter tool. It represents workload that extracts small but valuable analytics from the input data.
- TeraSort: this workload sorts the input data sets generated by the Hadoop TeraGen tool in a predefined order. It represents workload that transforms unstructured source data to a more useful structure or format without adding new data.

The input and output data usually need to be replicated to a small number of physically separate nodes (typically three) to insure against data block corruption and hardware failure. However, we disabled the replication of both input and output data in the experiments to concentrate on core behaviors of the MapReduce workflow.

I/O performance of mobile nodes

Before analyzing the performance of Hadoop mobile clusters, stress tests are performed on the mobile node, Google NEXUS 7, to identify the maximum operating capability of hardware resources (e.g. CPU, memory, file system, network, etc.). We also investigate which resource may cause performance degradation. In the stress tests, a distinct performance characteristic between the file system I/O and network I/O is observed, where the Hadoop TestDFSIO benchmark that tests the I/O performance of HDFS by sampling the number of bytes read/written at fixed time intervals is used to measure the file system I/O (in a single-node cluster setup) and the Iperf network performance measurement tool that generates constant TCP or UDP traffic flows is employed to compute the actual network throughput (between two cluster nodes).

Figure 3 displays throughput measurements of the file system and network in the load tests. The result shows that the network I/O is far slower that the file system I/O which is in complete contrast to the observations [27] made in wired Hadoop operating environments, where the network throughput is much higher than the data transfer rates of internal disks because typical Hadoop clusters are built with one or two 1 Gbps wired connections per node. Since the actual effect of the network throughput on Hadoop performance is relatively low in conventional Hadoop setups with high-speed wired connectivity, not much attention has been paid to Hadoop operations under network bandwidth constraints that are critical for reliable data transfers.

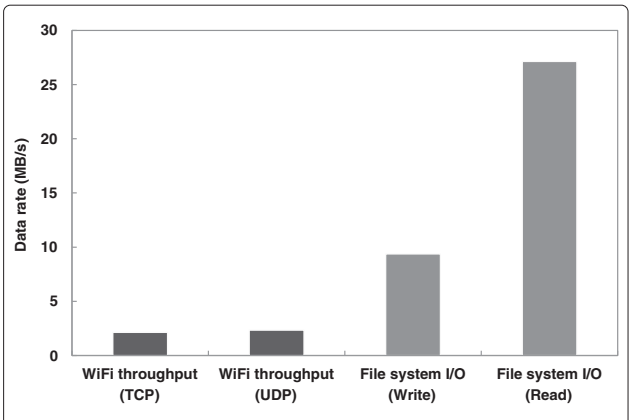

Figure 3 Performance comparison between file system and network I/O of mobile nodes.

Consequently, the performance of mobile distributed computing may be strongly influenced by the characteristics of wireless links in the mobile cluster. Although computing capabilities of nodes are a significant performance factor, each node also needs the capability to read and write large amounts of data to and from the distributed file system that is implemented on remote nodes. In wireless networks with relatively low network bandwidth, time required to transfer data blocks can significantly contribute to the total processing time even though the distributed computing power generally decreases the amount of time needed for job completion.

Performance of WordCount workload

The WordCount workload that counts the occurrence of each word in the input data sets produces small final output. The Map phase is generally computation intensive, compared to other phases. Network utilization is low in the Shuffle phase, in which the Map tasks transfer their output (i.e. intermediate results) to the Reduce task as input, because the Map output is a small subset of the large input data set in this kind of workload.

Figure 4 shows the network utilization with MapReduce task progress of the WordCount workload that starts with 1 GB input data. In the workload, 20 Map tasks corresponding to the 1GB input size are equally distributed over 10 Slave nodes. One node is chosen to run the single Reduce task that produces the final output. Figure 5 displays resource utilization on two typical Slave nodes; the Map node runs only two of 20 Map tasks and the Reduce node runs both the Map tasks and the additional Reduce task.

Figure 4 contains an aggregate data traffic pattern receiving from all nodes running Map tasks, which is denoted by the solid line and a single data flow transmitted by a typical Map node, denoted by the dash line. The graph shows two bursts of received traffic since each node finishes two assigned Map tasks one at a time and transmits the intermediate result at the same time to the single node running the Reduce task.

Although Hadoop has the ability to process multiple tasks simultaneously within resource bounds, the experimental nodes run tasks sequentially due to lack of memory (see Figure 5). This explains the separated bursts of traffic and corresponding delays in the Map and Reduce progress. The network bandwidth is saturated during each burst, but it only lasts for a short period of time since the output of the Map tasks is very small.

Performance of TeraSort workload

The TeraSort workload that sorts input data sets generates a large amount of intermediate data in the Map phase, which needs to be transmitted to the Reduce task over the network to produce the final output. Both Map and Reduce phase are commonly computation and I/O intensive. Network utilization is very high in the Shuffle phase because the output of Map tasks is the same size as the input data sets in this workload.

Figure 6 shows the network utilization with MapReduce task progress of the TeraSort workload initialized with 1 GB input data. The configuration is identical to the WordCount workload; 20 Map tasks are equally distributed over 10 Slave nodes and one node runs the single Reduce task. The resource utilization of two different Slave nodes is detailed in Figure 7 in the same way as the WordCount workload analysis.

Figure 6 illustrates a large volume of aggregate traffic made up of data flows transmitted at the same time by multiple nodes because the entire input data needs to be shuffled to the single node running the Reduce task. The network bandwidth is saturated while the output of all Map tasks is being transferred. This traffic

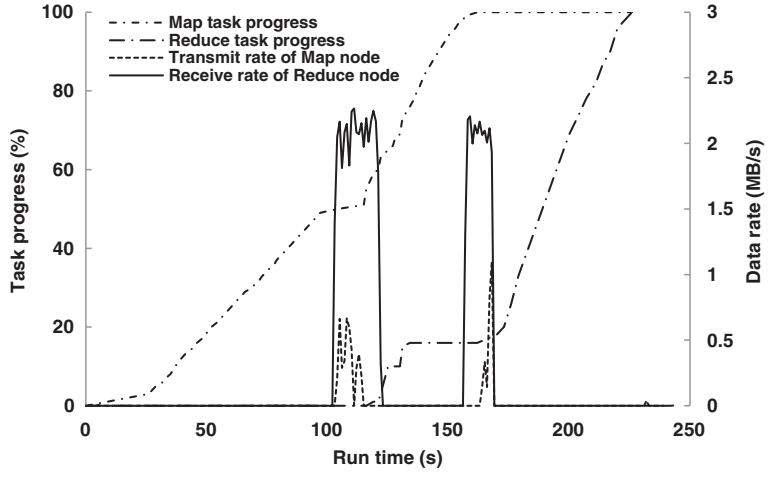

Figure 4 Network utilization of Hadoop mobile cluster running 1 GB WordCount workload.

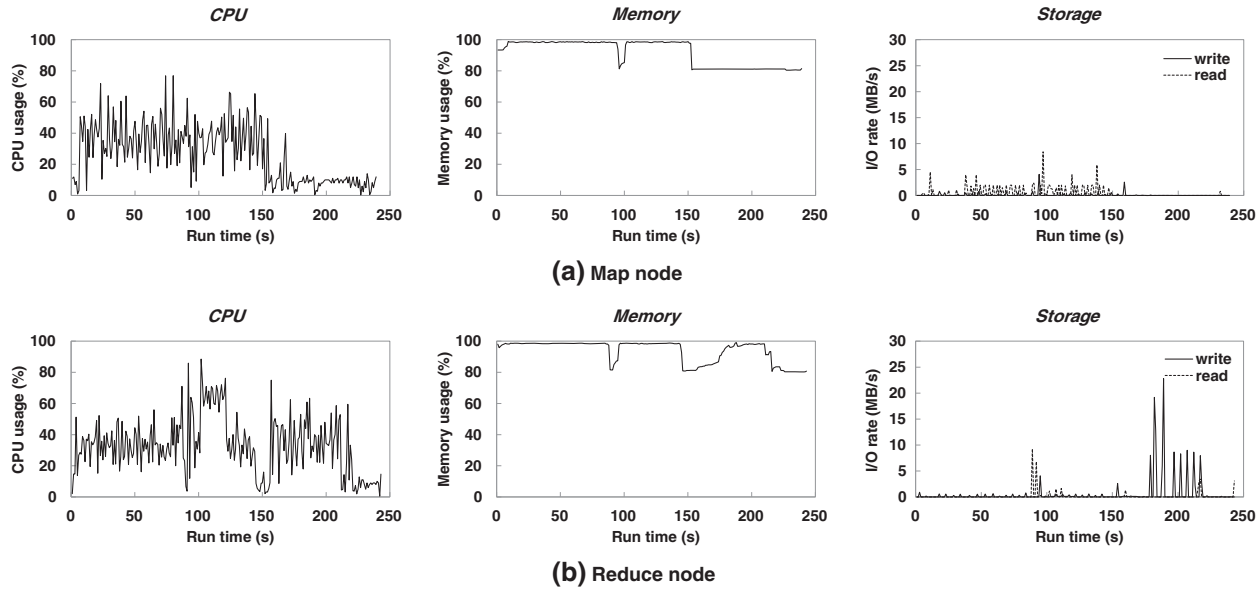

Figure 5 Resource utilization of (a) Map and (b) Reduce node with 1 GB WordCount workload.

pattern increases the possibility of packet loss, resulting in throughput reduction and fluctuating performance; a significant number of TCP packets are dropped during the data interchange. Consequently, the Map tasks finish relatively quickly but the Reduce task makes slow progress since it spends a great deal of time in receiving the large input data (i.e. the output of Map tasks) and processing the entire data sets.

Performance of scaling tests

This section examines the effects of scaling up the cluster size, data block size, and input data size that represents the variability in configuring the mobile ad hoc cloud.

In general, an optimally configured cluster has the ability to improve performance by scaling up the cluster size. Figure 8 shows the results from the experiments which are intended to verify how the cluster size affects the performance of the mobile distributed framework. The job completion time of two typical workloads (WordCount and TeraSort) with 1 GB input data is measured as the number of Slave nodes participating in the cluster gradually increases.

As indicated in Figure 8, increasing the number of nodes considerably decreases the job completion time of the WordCount workload. On the other hand, in the cluster scaling with the TeraSort workload, the increase in cluster

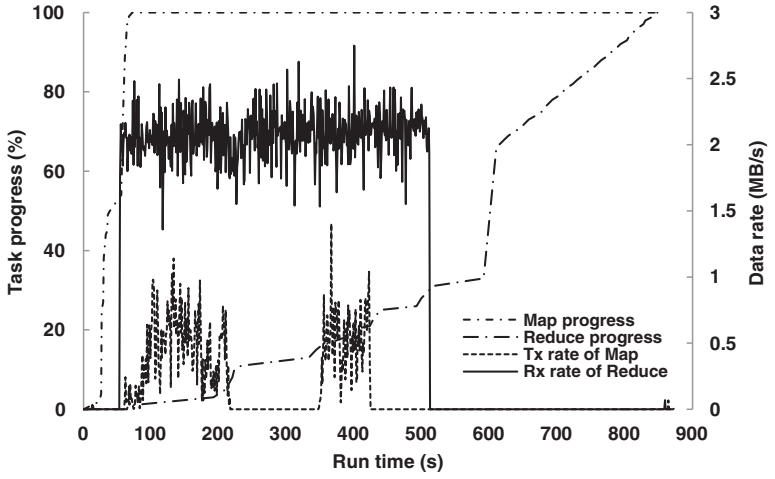

Figure 6 Network utilization of Hadoop mobile cluster running 1 GB TeraSort workload.

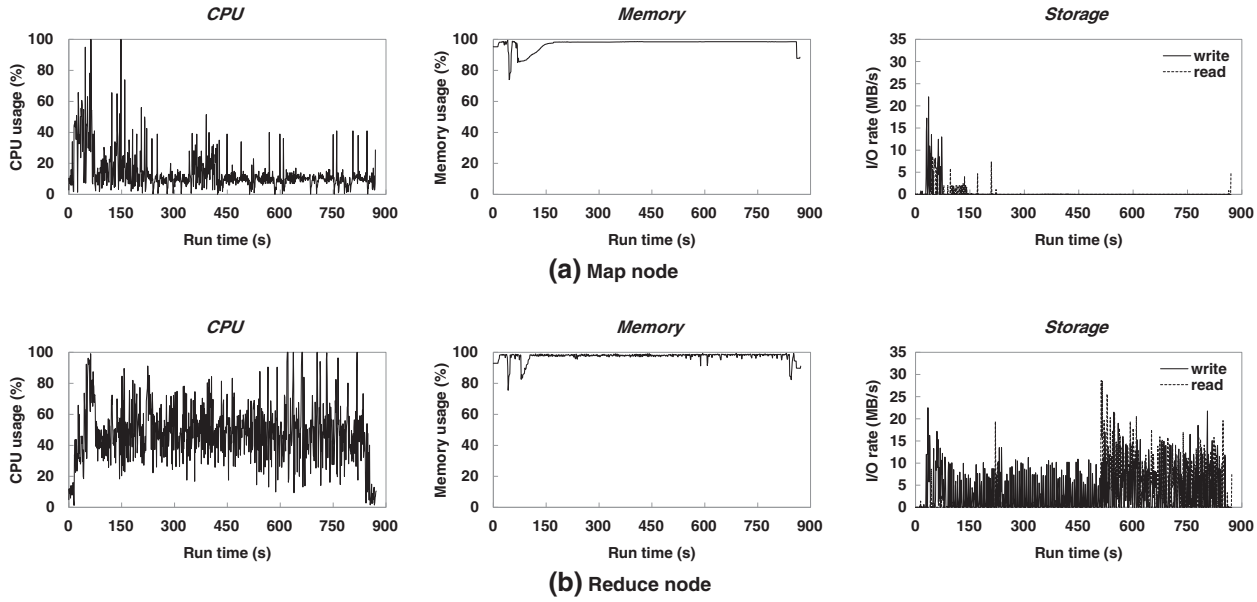

Figure 7 Resource utilization of (a) Map and (b) Reduce node with 1 GB TeraSort workload.

size does not lead to a significant decrease in job completion time because the performance of the mobile cluster is bounded by the time taken by the entire input data to be shuffled under the limited wireless bandwidth that is also highly variable.

The unit of input for a Map task is a data block of the input file. A single large input file is split into many blocks which are distributed over the nodes in the Hadoop cluster. The size of a data block stored in Hadoop file system is large – 64 MB by default, compared to a block size in traditional file systems – normally 512 bytes. By making a block large enough, the data transfer time from the disk becomes significantly larger compared to the time required to seek the start of the block. Thus, the transfer operation of a large file made of multiple blocks becomes faster by minimizing the seek time [25].

What is the effect of the data block size in wireless configurations where one or more phases of MapReduce transfer a considerable number of data blocks over wireless links with low throughput? The previous work [18] suggested the use of a small block size in consideration of the lengthy transfer time and delay of the large block in the wireless network. However, they did not provide any comparative measurements to validate their suggested value. To determine an appropriate data block size for the Hadoop mobile cluster, the job completion time of the I/O intensive TeraSort workload with 1 GB input data is measured as the data block size gradually increases.

Contrary to expectations, Figure 9 displays performance degradation in small data block sizes. A Map task handles a data block of input at a time. If the data block is very small (i.e. there are a large number of data blocks),

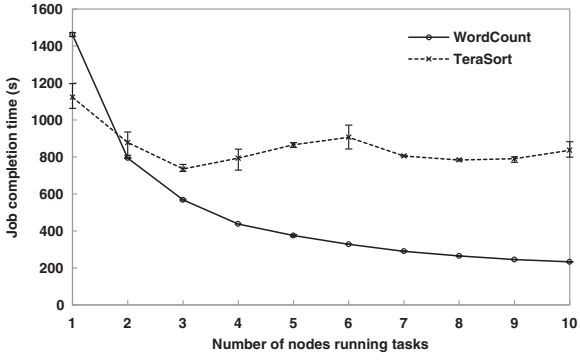

Figure 8 Cluster size scaling of WordCount and TeraSort workload.

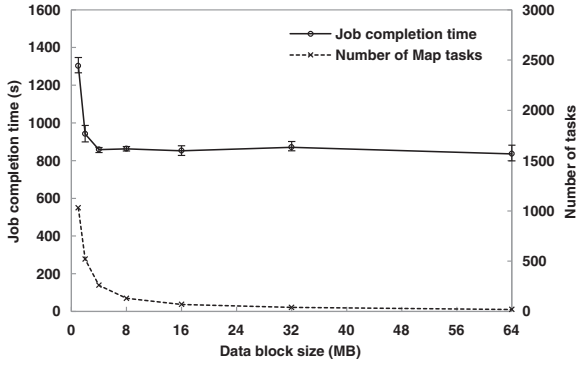

Figure 9 Data block size scaling of TeraSort workload.

more map tasks are required to process each data block as also shown in Figure 9. This may impose an inefficient data access pattern by causing frequent seeks to retrieve each small block. Furthermore, resources may be scarce for an excessive number of Map tasks. Hence, configuration parameters for the mobile cluster should be carefully determined by taking into account various other performance aspects.

Finally, Figure 10 demonstrates the impact of input data set size on the job completion time of the WordCount workload as the data set size increases. The larger the input data, the longer it takes to process the workload and produce the output result. Meanwhile, we encounter a problem in plotting the same measurements from the TeraSort workload because the performance is extremely variable and unreliable due to an increasing number of task failures (from task response timeouts and intermittent node disconnections) and re-runs. This paper identifies the cause of the failures and discusses performance issues of mobile cloud clusters in the following section.

Performance issues of mobile cloud clusters

Most of the current distributed systems including Hadoop employ Transmission Control Protocol (TCP) for reliable communications between cluster nodes. The performance of mobile distributed processing largely relies on how effectively each mobile device exploits the available network resources through TCP connections. Despite advances in mobile technologies, mobile devices still face significant limitations on transmitting and receiving reliable TCP data streams required to avoid any interruptions while performing distributed analytics.

Limitations on TCP performance over mobile devices

Mobile devices use a wireless channel as a transmission medium. Unlike wired networks, the time-varying condition on the wireless channel is the dominant cause of packet loss. TCP proposals mostly designed for wired networks are unable to react adequately to the packet loss due to channel noise, fading, or interference since they assume the only source of packet loss is congestion [31]. The random packet loss in the wireless channel makes it difficult for mobile nodes using one of those proposals (e.g. TCP CUBIC [32]) to estimate available channel bandwidth and achieve optimal TCP throughput. In addition, most of wireless protocols allow wireless devices to share the same channel through contention-based media access control (MAC) that includes procedures for initiating a new transmission, determining the channel state (e.g. available or unavailable), and managing retransmissions in the event of a busy channel or data loss. This has several limitations. If many nodes attempt to communicate at the same time, for example, many collisions may occur lowering the available bandwidth. Furthermore, there is no appropriate method to prioritize data traffic and prevent unfair transmissions without pre-coordination. Not many studies have been made on TCP performance of mobile distributed applications under these limitations.

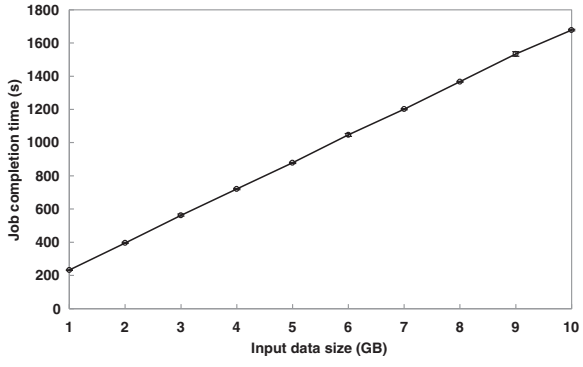

Figure 10 Input size scaling of WordCount workload.

The IEEE 802.11 standard for WLANs [33] defines several Physical-layer (PHY) data rates (e.g. recent mobile devices supporting IEEE 802.11n [29] use eight rates: 6.5, 13, 19.5, 26, 39, 52, 58.5, and 65 Mbps) to provide more robust communication by falling back to a lower rate in the presence of a high noise level. Rate adaptation algorithms of the media access control (MAC) make runtime prediction of changes in the channel condition and select the most appropriate PHY rate. Although the PHY rate change is critical to the TCP performance, the cross layer interaction between the TCP flow control and MAC rate adaptation is yet to be thoroughly investigated [34]. A problematic issue arises when the rate adaptation algorithm aggressively and rapidly reduces the PHY rate due to short-term degradation of channel quality. TCP reacts to the sudden PHY rate reduction but needs a substantial amount of settling time to converge into a stable rate by updating its congestion window size corresponding to the PHY rate. In the case of frequent occurrence of rate changes in the PHY layer, it is hard to utilize the available bandwidth to the fullest extent using TCP. In addition, TCP performance can drastically deteriorate if inappropriate PHY rates are selected by mistake.

Constraints of using mobile devices for mobile cloud

Some low-end mobile devices continue to have resource limitations compared to traditional PCs and laptops in spite of the advances made in their hardware capabilities. Especially, their wireless capability is limited by several factors including power-saving operations (e.g. lower communication quality and intermittent connectivity), form factor constraints (e.g. challenges in antenna implementation and placement), and minimal production costs (e.g. small network buffer/queue due to low memory capacity), which subject them to throughput reduction

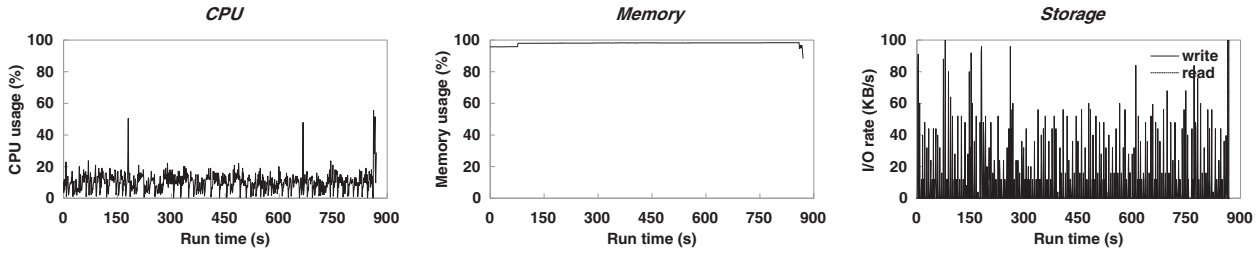

Figure 11 Resource utilization of Hadoop Master node running 1 GB TeraSort workload.

and fluctuating performance [35]. Moreover, when an application on the receiver is not able to process TCP packets as fast as senders transmit due to lack of processing resources, the receiver sets the TCP flow limit by decreasing its receive window size. As a result, the sender's transmission will eventually be restricted by the receiver's processing rate. Thus, the processing capability of a mobile device potentially becomes a significant factor (i.e. TCP transmit rate bound) in network performance when the device experiences resource scarcity on processing requests.

Interestingly, most of mobile devices are optimized to improve receive performance. This characteristic can be found when looking into the mobile OS kernel and wireless drivers. For example, the mobile devices have an asymmetric resource scheduling (or distribution) scheme for transmitting and receiving packets, where the mobile kernel allocates more resources to speed up processing of MAC frames on arrival and the minimum number of frames necessary to acknowledge the received frames is scheduled for transmission while receiving data. In addition, the mobile OS does not alert the user to runtime errors of its wireless kernel or hardware faults nor display information about the internal problems directly, which makes it difficult to identify critical performance factors and improve the performance of mobile applications such as distributed analytics. Besides, it is not an easy task to customize the OS kernel and wireless driver of mobile devices for the variable operating environment although the mobile OS is open-source. Hence, the network performance observed on mobile devices may not be optimal and it is hard to find out the performance limit.

Performance problems of Hadoop mobile clusters

From the performance analysis, it is found that the overall computing power of the mobile cluster is no longer significantly bounded by internal resource capabilities of each individual node since mobile devices have been constantly enhancing their resources and processing power. On the other hand, this study identifies distinct problems in conducting Hadoop distributed analytics on the mobile clusters, which come in the form of longer job completion time or frequent task failure from task response timeout and node disconnection.

In distributed systems where a controller usually makes control decisions with limited information from remote components, a timeout control provides a key mechanism through which the controller can infer valuable information about unobservable states and events in the system when direct feedback is either impossible or costly [36]. The timeout control is configured using a timer which expires after a timeout threshold. This defines an expected time by which a specific event should occur. If no information arrives within this period, a timeout event occurs and the controller triggers corresponding reactions. In fact, timeout control is an integral component for building up reliable distributed systems.

The Hadoop distributed system also adopts the timeout control for both job scheduling and progress monitoring. A MapReduce job initiates long-lived batch tasks running on Slave nodes, which usually take a few minutes or hours. Because of the significant length of run-time, it is important for the Master node to get feedback on how the job is progressing in a timely fashion. It enables the Master to keep track of task status and restart failed or slow tasks.

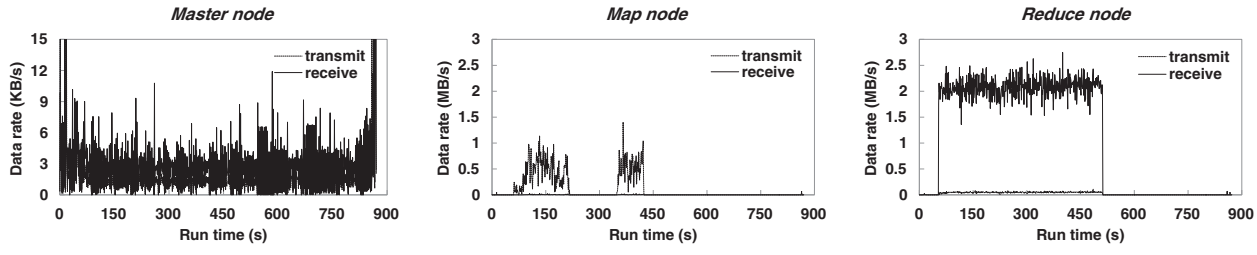

Figure 12 Comparison of traffic patterns on Hadoop cluster nodes with 1 GB TeraSort workload.

If a Slave (task) fails by crashing or running very slowly, for example, it stops sending (or sends intermittently) current status and progress updates, called Heartbeats, to the Master; the Master then marks the Slave (task) as failed after the timeout threshold which is 10 minutes by default [25].

In the previous experiments, the frequent timeout occurrences (task failures) with corresponding performance degradation while running the I/O intensive TeraSort workload with large input data are observed in the Hadoop mobile clusters. The problems can be summarised as follows.

First, the job execution time is sensitive to slow-running tasks as only one slow task makes the time significantly loner. When a mobile node running Map tasks has significant delays in transmitting a large amount of intermediate result to Reduce tasks through wireless connections (i.e. tasks are running slower than expected due to the lengthy transfer time of Shuffle phase), the Master launches another, equivalent tasks as a backup instead of diagnosing and fixing the slow-running tasks. The slow-running (or hanging) tasks are considered failed and automatically killed after the timeout period. The Master also tries to avoid rescheduling the tasks on the Slave node where they have previously failed.

Second, depending on the size of the cluster, the Master node has high resource requirements as it manages the cluster resources, schedules all user jobs, and holds block metadata of the distributed file system. On a busy cluster running a heavy workload, the Master uses more memory and CPU resources. Thus, the Master node based on a mobile device is subject to resource scarcity and bottlenecks in processing receive data in a timely fashion; the high memory usage and steady storage utilization of the Master node are commonly observed as shown in Figure 11. Its incessant sort-lived traffic pattern compared to the Map and Reduce node is also displayed in Figure 12. When the Master has not received an expected progress update from a Slave node for the timeout threshold, it arranges for all the Map tasks that were scheduled on the failed node, whether completed or not, to be rerun since intermediate output residing on the node may not be accessible to the Reduce task.

Consequently, theses failures and reactions lead to a significant increase in job execution time. Therefore, it is critical to mitigate the effect of the timeout occurrences in the Hadoop mobile clusters where the chance of particular node failures and communication problems is comparatively high.

Conclusion

This paper studies the advantages and challenges of using mobile devices for distributed analytics by showing its feasibility and conducting performance analysis. The empirical study focuses on how to build mobile ad hoc cloud by clustering with nearby mobile devices to reliably support practical distributed analytics such as actionable analytics. For enabling actionable analytics in mobile devices, the following questions should be addressed:

- What are the limitations in enabling mobile devices to offload portions of the workload to remote computing resources and share their resources for distributed processing? How efficiently can the controller node initiate distributed analytics using dynamic mobile cloud resources under the time-varying operating environment?
- In what ways, is the mobile cluster able to mitigate the effect of frequent task failures while supporting large complex computations and long-running processes for distributed analytics, which are usually caused by hardware/software faults (or slow-running tasks) and communication problems?
- How can reliable data communications between mobile devices for analytical data transfers in the workflow of distributed analytics be guaranteed under the limitations of TCP performance over wireless links? What is the best way to control TCP flows on mobile devices for improving performance of mobile distributed analytics?

To resolve these questions, this study will continue to conduct performance analysis using various benchmarks and sample applications, e.g. web search, machine learning, etc., and identify more critical performance issues. Based on the performance studies, our future work will propose adaptive TCP algorithms for enhanced analytic performance of mobile cloud clusters.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

SL designed and performed experiments, analyzed data and drafted the manuscript. KG reviewed the manuscript and discussed the results and implications. AL provided direction and advice for the study and revised the manuscript. All authors read and approved the final manuscript.

Authors' information

Seungbae Lee received his B.S. degree in Computer Science and Statistics from Air Force Academy, South Korea in 1998 and his M.S. degree in Technology Management, Economics and Policy from Seoul National University, South Korea in 2008. Since 2010, he has been working towards his Ph.D. degree in Computer Science and Software Engineering at Auburn University, USA. His current research interests include performance analysis of wireless communications and implementation of mobile systems.

Kanika Grover is currently doing her Ph.D. in Computer Science and Software Engineering at Auburn University. Her current research interests include security measures in wireless sensor/ad hoc networks, mobile and pervasive computing, broadcast authentication in vehicular networks and performance analysis of mobile cloud clusters.

Alvin Lim is currently an associate professor of Computer Science and Software Engineering at Auburn University. He received his Ph.D. degree in Computer Science from University of Wisconsin at Madison in 1993. His research interests include self-organizing sensor networks, mobile and

pervasive computing, network security, wireless networks, reliable and dynamically reconfigurable distributed systems, complex distributed systems, mobile and distributed databases, distributed operating systems, and performance measurement and analysis.

Received: 5 March 2013 Accepted: 27 July 2013
Published: 1 October 2013

References

1. IDC Inc. (2012) IDC Predictions 2013: Competing on the 3rd Platform. Market Analysis. http://www.gartner.com/technology/research/top-10-technology-trends. Accessed Mar 2013
2. Gartner Inc. (2012) Gartner Says 821 Million Smart Devices Will Be Purchased Worldwide in 2012; Sales to Rise to 1.2 Billion in 2013. Press Release. http://www.gartner.com/newsroom/id/2227215. Accessed Mar 2013
3. Gartner Inc. (2012) Top 10 Strategic Technology Trends for 2013. Market Analysis. http://www.gartner.com/technology/research/top-10-technology-trends. Accessed Mar 2013
4. Hadoop projects, Apache Software Foundation. http://hadoop.apache.org. Accessed Mar 2013
5. Dean J, Ghemawat S (2008) MapReduce: Simplified Data, Processing on Large Clusters. Communications of the ACM, ACM Volume 51(Issue 1): pp 107–113
6. Ghemawat S, Gobioff H, Leung ST (2003) The Google file system. In: ACM SIGOPS Operating Systems, Review. ACM, Bolton Landing, New York, USA, pp 29–43
7. The White House (2012) A Toolkit to Support Federal Agencies Implementing Bring Your Own Device (BYOD) Programs. http://www.whitehouse.gov/digitalgov/bring-your-own-device. Accessed Mar 2013
8. IDC Inc. (2011) Worldwide Mobile Worker Population 2011–2015 Forecast. Market Analysis. http://www.idc.com/getdoc.jsp?containerId=238366#.USZZJ6WNEwE. Accessed Mar 2013
9. Vaquero LM, Rodero-Merino L, Caceres J, Lindner M (2008) A Break in the Clouds: Towards a Cloud Definition. ACM SIGCOMM Computer Communications Review, ACM Volume 39. Issure 1, pp 50–55
10. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, et.al (2010) A View of Cloud Computing. Communications of the ACM, ACM Volume 53(Issue 4): 50–58
11. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud Computing and, Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Future Generation Computer Systems, Elsevier Science Publishers B. V. Volume 25, Issue 6, pp 599–616
12. Zhang Q i, Cheng L, Boutaba R (2010) Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications, Springer-Verlag. Volume 1, Issue 1, pp 7–18
13. Mei L, Chan WK, Tse TH (2008) A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues. In: The 3rd IEEE Asia-Pacific Services Computing Conference (APSCC) 2008. IEEE, Yilan, Taiwan, pp 464–469
14. Wang L, Von Laszewski G, Younge A, He X, Kunze M, Tao J, Fu C (2010) Cloud Computing: a Perspective Study. New Generation Computing, Verlag Omsha, Tokio. Voulume 28, Issue 2, pp 137–146
15. Dinh HT, Lee C, Niyato D, Wang P (2011) A survey of mobile cloud computing: architecture, applications, and approaches. Wireless Communications and Mobile Computing, John Wiley & Sons, Ltd. http://dx.doi.org/10.1002/wcm.1203
16. Fernando N, Loke SW, Rahayu W (2012) Mobile cloud computing: A survey. Future Generation Computer Systems. Elsevier Science Publishers B. V. Volume 29, pp 84–106
17. Zachariadis S, Mascolo C, Emmerich W (2004) Satin: A Component Model for Mobile Self Organisation. In: On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and, ODBASE. Agia Napa, Cyprus, Springer Berlin Heidelberg, pp 1303–1321
18. Marinelli EE (2009) Hyrax: Cloud Computing on, Mobile Devices using MapReduce. Master's thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
19. Teo CLV (2012) Hyrax: Crowdsourcing Mobile, Devices to Develop Proximity-Based Mobile Clouds. Master's thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
20. Huerta-Canepa G, Lee D (2010) A Virtual Cloud, Computing Provider for Mobile Devices. Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, San Francisco, California, USA, ACM. p 6
21. Shi C, Lakafosis V, Ammar MH, Zegura EW (2012) Serendipity: Enabling Remote Computing among Intermittently Connected Mobile Devices. In: Proceedings of the thirteenth, ACM international symposium on Mobile Ad Hoc Networking and Computing, Hilton Head, South Carolina, USA, ACM, pp 145–154
22. Shi C, Ammar MH, Zegura EW, Naik M (2012) Computing in Cirrus Clouds: the Challenge of Intermittent Connectivity. In: Proceedings of the first edition of the, MCC workshop on Mobile cloud computing. ACM, Helsinki, Finland, pp 23–28
23. Kemp R, Palmer N, Kielmann T, Bal H (2012) Cuckoo: A Computation Offloading Framework for Smartphones. Mobile Computing, Applications, and Services, Springer Berlin Heidelberg. Volume 76, pp 59–79
24. Comito C, Falcone D, Talia D, Trunfio P (2013) A Distributed Allocation Strategy for Data Mining Tasks in Mobile Environments. In: Intelligent Distributed Computing VI, Proceedings of the 6th International Symposium on Intelligent Distributed Computing - IDC 2012. Springer Berlin Heidelberg, Calabria, Italy, pp 231–240
25. White T (2012) Hadoop: The definitive guide. O'Reilly Media, Sebastopol, California, USA
26. Rajaraman A, Ullman JD (2011) Mining of Massive Datasets. Cambridge University Press, Cambridge, UK
27. Cisco Systems Inc. (2011) Big Data in the Enterprise: Network Design Considerations. White paper. http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9670/white_paper_c11-690561.html. Accessed Mar 2013
28. Google Inc. (2012) NEXUS 7. http://www.google.com/nexus/7. Accessed Mar 2013
29. (2009) IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput. IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009), pp 1–565
30. LinuxonAndroid project. http://linuxonandroid.org. Accessed Mar 2013
31. Afanasyev A, Tilley N, Reiher P, Kleinrock L (2010) Host-to-Host Congestion Control for TCP. Communications Surveys & Tutorials, IEEE. Volume 12, Issue 3, pp 304–342
32. Ha S, Rhee I, Xu L (2008) CUBIC: A New TCP-Friendly, High-Speed TCP Variant. ACM SIGOPS Operating Systems Review, ACM. Volume 42, Issue 5, pp 64–74
33. (2012) IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007), pp 1–2793
34. Khademi N, Welzl M, Gjessing S (2012) Experimental evaluation of TCP performance in multi-rate 802.11 WLANs. In: IEEE International Symposium on a, World of Wireless, Mobile and Multimedia Networks (WoWMoM). IEEE, San Francisco, California, USA, pp 1–9
35. Sanadhya S, Sivakumar R (2011) Adaptive Flow Control for TCP on Mobile Phones. In: Proceedings IEEE INFOCOM. IEEE, Shanghai, P.R. China, pp 2912–2920
36. Kebarighotbi Ali, Cassandras CG (2011) Timeout Control in Distributed Systems using Perturbation Analysis. In: 50th IEEE Conference on, Decision and Control and European Control Conference (CDC-ECC). IEEE, Orlando, Florida, USA, pp 5437–5442

doi:10.1186/2192-113X-2-15
Cite this article as: Lee *et al.*: **Enabling actionable analytics for mobile devices: performance issues of distributed analytics on Hadoop mobile clusters.** *Journal of Cloud Computing: Advances, Systems and Applications* 2013 **2**:15.