Energy-Efficient Dynamic Traffic Offloading and Reconfiguration of Networked Data Centers for Big Data Stream Mobile Computing: Review, Challenges, and a Case Study

Enzo Baccarelli, Nicola Cordeschi, Alessandro Mei, Massimo Panella, Mohammad Shojafar, and Julinda Stefa

Abstract

Big data stream mobile computing is proposed as a paradigm that relies on the convergence of broadband Internet mobile networking and real-time mobile cloud computing. It aims at fostering the rise of novel self-configuring integrated computing-communication platforms for enabling in real time the offloading and processing of big data streams acquired by resource-limited mobile/wireless devices. This position article formalizes this paradigm, discusses its most significant application opportunities, and outlines the major challenges in performing real-time energy-efficient management of the distributed resources available at both mobile devices and Internet-connected data centers. The performance analysis of a small-scale prototype is also included in order to provide insight into the energy vs. performance tradeoff that is achievable through the optimized design of the resource management modules. Performance comparisons with some state-of-the-art resource managers corroborate the discussion. Hints for future research directions conclude the article.

Consider a spatially distributed networked system with thousands of energy- and resource-limited mobile devices (i.e., "thin" devices, e.g., smartphones, PDAs, tablets, RFIDs), all capable of acquiring and communicating in real time an ever-increasing volume of heterogeneous data streams. Due to the traditional energy-computing-bandwidth limitations of the mobile domain, a few years ago a system of this complexity was unthinkable. However, the incoming big data era is experiencing the convergence of broadband mobile Internet networking (B-MINet) and real-time mobile cloud computing (Rt-MCC), so the new paradigm of big data stream mobile computing (BDSMC) is quickly gaining momentum [1].

BDSMC aims at describing a new generation of mobile/wireless integrated computing-networking infrastructures, which should be designed to extract hidden value from an ever increasing volume of space-time correlated heterogeneous data streams by enabling in real time their energy-efficient acquisition, wireless transport, and processing. According to this picture, in this article we propose a "five Vs" formal characterization of the BDSMC paradigm, that is, variety (data heterogeneity), volume (ever increasing amount of data to be processed), velocity (data generation at fast and unpredictable rates), value (huge value but hidden in massive datasets at very low density), and volatility (the acquired data streams must be transported and processed in real time). While the first four Vs are common to all big data applications, we introduce the last V (i.e., volatility) for featuring big data stream applications. In general, the value of a stream of data is closely related to both its space and time coordinates, and hence, after acquisition, this value quickly decreases if the computing-plus-communication delay is larger than a suitable quality of service (QoS)-dictated hard threshold.

Big data streams are typically acquired and locally preprocessed by a number of heterogeneous spatially distributed mobile/wireless devices and then transported to remote data centers for further post-processing. Hence, according to the proposed five Vs characterization, we further assume that the life cycle of big data streams is composed of three main phases: data acquisition and local preprocessing at the mobile devices, data transport, and data post-processing at the remote data centers. According to this three-phase life cycle, in Fig. 1 we show the main functional blocks of an integrated computing-communication technological platform for the support of BDSMC. It is composed by the interconnection of three tiers, that is, the radio access networks (RANs), the Internet backbone, and the remote networked data center.

The RANs are directly connected to the mobile devices and bridge them to the Internet backbone. On one hand, RANs suffer from fading and interference, which give rise to fluctuating access bandwidths. On the other hand, mobile devices are energy-limited and equipped with scarce computing/storage

The authors are with Sapienza University of Rome, Italy.

 0890-8044/16/$25.00 © 2016 IEEE

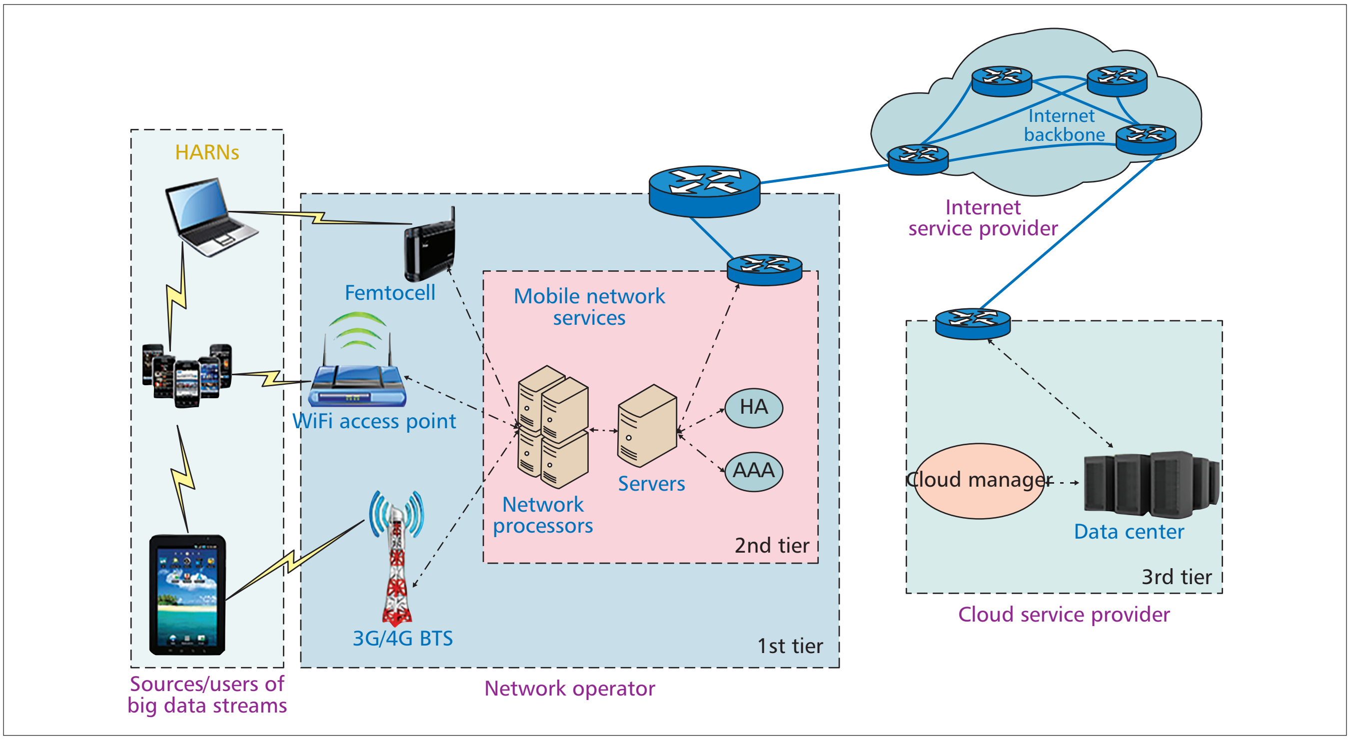

Figure 1. The three-tier Internet-assisted BDSMC reference architecture. Thick (thin dashed-dotted) lines indicate TCP-over-IP (data link) connections. HA: home agent; AAA: authentication, authorization, and accounting; BTS: base transceiver station; HRANs: heterogeneous radio access networks.

resources. This forces them to offload data and computing-intensive application tasks to the remote data center, possibly by opportunistically exploiting the "best" bandwidth/delay/energy consumption mix offered by the available access technologies (e.g., 3G/4G-cellular, WiFi, femto-cellular, WiMAX).

The Internet backbone multiplexes the traffic streams generated by the RANs and forwards them to the remote data center. In order to support multimedia stream applications (e.g., photo and video sharing), the Internet backbone should guarantee the forwarding of massive data at low latency. Currently, this requirement still creates significant challenges to the Internet backbone [2], and requires that the traffic offloaded by the mobile devices is dynamically planned by also accounting for the actual utilizations of the Internet backbone and the data center.

Finally, streams of big data are post-processed and stored in the data center. Today, this is made feasible by virtualization techniques, which allow a clone of a mobile application to synchronously run in an isolated container (e.g., a virtual machine, VM) hosted by the data center. However, in order to both speed up the computing time and reduce the computing energy, modern data centers implement massive parallel processing, which in turn requires large amounts of data to be transferred among different VMs. As a consequence, the energy currently consumed by the data center network may represent more than 20 percent of the overall data center energy consumption, while the inter-VM communication may waste more than 33 percent of the overall processing time, especially for stream workloads [2].

In summary, real-time processing and energy efficiency are two topics of major concern in managing computing-communication technological platforms supporting BDMSC. This is the focus of this position article, which deals with the energy-saving balanced provisioning, dynamic scaling, and distributed QoS management of the communication-plus-computing resources at both the mobile devices and the remote data center of the technological platform of Fig. 1.

Classification of BDSMC and Opportunities for QoS Applications

The aforementioned resource management must be compliant with the (heterogeneous) statistical features of the sources of big data streams and the QoS requirements of the supported applications. In Table 1, we introduce a classification of the BDSMC operative environments, which is based on four main aspects: data sources, content format, data shedding (i.e., compression/fusion) techniques, and QoS supported applications. On the basis of this classification, we individuate three broad classes of potential "killer" applications for the BDSMC paradigm: the mobile Internet of Things (IoT), spatial crowdsourcing (SC), and online social network and nomadic computing (SNNC).

Mobile IoT aims to provide Internet connection to a world of heterogeneous mobile smart devices, which must be capable of supporting both machine-to-machine and machine-to-human communications. The final goal is to make feasible the real-time exchange of information about the surrounding environment by leveraging emerging technologies, such as mobile sensor networks, RFID, and biosensors. Hence, mobile IoT exhibits two basic features that conform to the BDSMC paradigm: a large set of spatially distributed, heterogeneous, and energy-limited wireless devices generate masses of real-time data streams; and IoT data is useful only when it is mined, but this requires a lot of bandwidth and computing resources.

In SC application environments, large populations of non-professional clients utilize their smartphones as basic sensing units for timely distribution of sensed tasks and real-time acquisition/processing/diffusion of sensed data streams. Hence, the key feature that conforms SC applications to the BDSMC paradigm is the collective real-time processing of the acquired streams for spatial monitoring and/or decision making.

Finally, online SNNC applications include context-aware services for personal computing and communication activities on the go, which allow the user to exchange information with other humans in real time through cloud-assisted

Categories of big data stream	
Classification	*Description*
Sources of data stream	
IoT	Thin devices (e.g., smartphones, PDAs, tablets, RFIDs) are identified by IP addresses. Their Internet-based interconnection enables complex services for the support of economic, environmental, and health needs.
Crowdsourcing	Several unskilled users utilize their smartphones as basic sensing units for performing coordinated sensing tasks.
Social media	It is the source of data generated via URL to share or exchange data in virtual communities and social networks.
Content format	
Structured	Structured data assumes the form of ordered records, and include numbers, words, and dates.
Unstructured	Unstructured data do not conform to a predefined template. Examples are location information, messages, videos, and social data sets.
Semi-structured	Semi-structured data are in the form of structured data that are not further organized into relational database models, such as tables and graphs.
Data shedding techniques	
Pseudo-random sampling	It aims to randomly drop segments of acquired data streams while preserving their main statistical features, such as frequencies of symbol occurrences.
Compressive sampling	It performs progressive fusion of the acquired streams while guaranteeing their (quasi) perfect reconstruction via the resolution of suitable optimization problems.
Distributed source coding	It exploits the space-time inter-correlation of the acquired streams for reducing the aggregate data rate of the encoded stream, while preserving the overall information content.

Application vs. QoS requirements			
	IoT	**SC**	**SN&NC**
Bandwidth	$\leq 10^{-1}$ (Mb/s)	≤ 2 (Mb/s)	≤ 10 (Mb/s)
Delay	≤ 500 (ms)	≤ 150 (ms)	≤ 4 (s)
Delay jitter	≤ 40 (ms)	≤ 20 (ms)	≤ 150 (ms)
Packet loss probability	$\leq 10^{-3}$	$\leq 10^{-5}$	$\leq 10^{-2}$
Block probability	$\leq 10^{-1}$	$\leq 10^{-2}$	$\leq 10^{-1}$

Table 1. Categories of big data stream and application-vs.-QoS requirements [3]. The reported numerical data are to be understood as per-class gross upper bounds. IoT: Internet of Things; SC: crowdsourcing; SN&NC: social network and nomadic computing.

social network platforms. The basic feature conforming SNNC to BDSMC is the requirement of massive sets of inter-stream cross-correlation operations, in order to perform real-time detection of new social trends, similarities, and/or anomalies.

Overall, since the energy-saving resource management policies reviewed in the next sections are QoS-based and thus rely on (accurate) characterizations of per-service resource usage, in the last part of Table 1 we provide a gross summary of the main QoS requirements of the aforementioned "killer" applications.

Related Work and Open Challenges

Resource management in BDSMC systems involves real-time offloading of code and/or data to the remote data center through the available mobile access-plus-Internet network, as well as the corresponding real-time reconfiguration of the networked data center (Fig. 1). The final target is minimization of the overall computing-plus-communication energy consumption under the QoS requirements of Table 1.

Work on the Adaptive Reconfiguration of Virtualized Networked Data Centers

Providing real-time computing support to the aforementioned BDSMC applications through remote clouds hosted on virtualized networked data centers (VNetDCs) is the target of a few recent management frameworks, such as S4 [4] and D-Streams [5]. While these management systems are specifically designed for the real-time support of large-scale BDSMC applications, they do not provide automatic and dynamic adaptation to the time fluctuations of the input streams to be processed. Dynamic adaptation of the available computing resources to the ever changing rate of the input streams is, indeed, provided by the more recent Time Stream [6] and PLAstiCC [7] management frameworks. However, these frameworks do not consider simultaneous management of network resources; also, they do not enforce hard real-time processing-plus-communication constraints.

Among the richer set of articles that cover the more general topic of resource management in delay-tolerant cloud-based environments, some recent contributions may be considered representative of the current state of the art. Specifically, since the workload offered by big data streams exhibits almost unpredictable time fluctuations that are hard to forecast in a reliable way, in [8] a Lyapunov-based technique is used to dynamically optimize the provisioning of computing resources by exploiting the available queue information. Although the approach pursued in [8] is of interest, it relies on an inherent delay vs. utility trade-off, which does not account for the hard constraints on the allowed computing delays. Very recently, in [9] the problem of the allocation of computing and networking resources in large-scale data centers is addressed. After recognizing that the resulting optimization problem is NP-hard, a heuristic integrated resource allocator is presented, in order to decide on the admission of dynamically arriving streams of data and allocate resources to the accepted ones. Although performance guarantees in terms of per-stream dedicated minimum bandwidths are accounted for by [9], its target is the maximization of the accepted arrival requests. Hence, the resulting resource allocator:

- Subsumes non-reconfigurable VNetDC platforms
- Does not consider the issue of energy-saving management of computing-bandwidth resources
- Does not enforce constraints on the allowed queue-plus-communication-plus-computing delays

Work on Real-Time Traffic Offloading

Real-time traffic offloading is particularly challenging in BDSCM scenarios, where applications constantly generate large amounts of data. However, the local processing of these data is also expensive for resource-limited devices, so real-time offloading of big data streams currently present two main critical issues.

The first issue concerns the costs to be sustained by the mobile device in terms of bandwidth consumption, which is induced by the Internet-assisted device-cloud communication, and energy consumption, which stems from the intensive usage of networking wireless interfaces (Fig. 1). The authors of [10] analyze these costs on a testbed of 11 Android-equipped smartphones, which are used as the primary phones by the participants in an experiment lasting three weeks and involving five different cellular service providers of two European countries. The obtained numerical results indicate that the offloading costs can be lowered by performing continuous synchronization between devices and the cloud.

In principle, traffic offloading could enable devices to balance the energy consumed by the networking interfaces for device-to-cloud communication and the energy saved by running jobs remotely [11]. Therefore, a second issue with the current offloading approaches is that they can only deal with computationally intensive applications that do not need large volumes of input data to be executed. As a consequence, BDSMC applications should not be offloaded, because the cost to send a large volume of data to the cloud actually offsets the gain from remote execution. However, very recently the authors of [12] mitigate this drawback by proposing a cloud-integrated mobile operating system (OS), the Cloud-anDroid (CDroid) OS. CDroid is a distributed OS residing partially on the mobile device and partially on a cloud software clone synchronized with the device. It is capable of efficiently exploiting the available RANs and Internet backbone of Fig. 1 by using the cloud side as just another resource of the real device. The CDroid system is a building block of the StreamCloud prototype illustrated later.

The Main Open Challenges

From the outset, two main challenges are presented by the BDSMC paradigm.

First, due to the inherent real-time constraints, the performance of BDMSC applications also depends on the physical location of the remote servers, their computation load, and the communication latencies introduced by both inter- and intra-data-center networks. As a consequence, a first challenge regards the provision of cloud support to delay-sensitive applications (Table 1) by exploiting the opportunistic access to the communication and computer resources of nearby mobile devices. In the current literature, this issue is handled by resorting to multi-tier offloading architectures [2]. They harvest computation from nearby less powerful devices (e.g., home computers, femtocells, access points, and/or car computers) for performing low-latency offloading of delay-sensitive "light" applications, while outsourcing the computationally intensive execution of delay-tolerant applications to remote powerful clouds. However, real-time performance of energy-efficient balancing of the total workload over the tiers of these multi-tier systems is still an open research topic.

Second, the energy consumed by intra-data-center networks may represent a large part of the energy demand of the overall BDSMC system, especially when the utilized networks are bandwidth-limited. In fact, current virtualized data centers are not designed to support communication/computer-intensive real-time big data applications, such as real-time video coding/decoding, target recognition, and tracking (Table 1). In fact, imposing hard limits on the overall per-job delay forces the overall virtualized networked infrastructure at the data center to quickly adapt its resource utilization to the (possibly unpredictable and abrupt) time fluctuations of the input workload. Hence, in order to minimize energy consumption, the joint balanced provisioning, scaling, and distributed management of the communication-plus-computer virtual resources of virtualized networked data centers represent a second major challenge.

A Case Study: StreamCloud

The aforementioned challenges are addressed by the self-configuring framework recently presented in [12, 13], which is currently in the prototype phase and is referred to herein as the "StreamCloud paradigm." As detailed in the following, this paradigm relies on an Internet-assisted peer-to-peer service architecture, which minimizes the energy costs at both the mobile devices and the data centers through adaptive synchronization of the corresponding resource management operations and the real-time hibernation of the underutilized networking/computing servers.

Test Scenario

The application scenario considered for testing the performance of the StreamCloud prototype falls into the aforementioned class of nomadic computing. Let us assume that a nomadic user has taken several pictures with her phone during a tour, and she wants to share only pictures of panoramic views with a colleague occasionally met at a bus station. Since the number of pictures is large (100 in our tests), and she has little time (just until the bus comes), she opts for installing a face detection application for her Android smartphone (http://www.anddev.org/quick_and_easy_facedetector_demo-t3856.html).

The application is supposed to help her to quickly and automatically separate the panoramic pictures of interest from those including her or her friends, and it runs on the smartphone. In our field trials, Samsung Galaxy S+ smartphones equipped with 1.4 GHz CPUs and 512 MB RAM modules are used for performing traffic offloading over legacy HTTP/TCP/IP over third generation (3G)/IEEE802.11b connections (Fig. 1). Furthermore, the test data center consists of a server farm with up to 14 IBM BladeCenter HS21 servers equipped with two 3.0 GHz dual-core Intel Woodcrest Xeon processors and 1 GB memory per blade. Depending on the considered test scenarios, Fast, Giga, and Ten-Giga Ethernet LANs are implemented through Cisco Nexus 55548P commodity switches in

	WiFi	3G
C (J/(Mb/s))	1.25	3.75
E_{setup} (J)	5.7	3.4
$R_O{}^{max}$ (Mb/s)	10.5	1.8
$R_O{}^{min}$ (Mb/s)	3.15	0.54
Maximum RTT (ms)	250	300
Minimum RTT (ms)	10	20

Table 2. Measured profiles of wireless access connections.

order to ensure inter-processor communications. We installed and configured Apache as an application server on each blade server in order to process the HTTP transactional requests generated by the CDroid module.

Profile of the Wireless Access Segment

Energy profiling of the tested 3G/WiFi access connections has been performed by sampling the batteries of the tested smartphones at 5000 Hz by using the Mobile Device Power Monitor (http://www.msoon.com/LabEquipment/PowerMonitor). The energy vs. offloading rate model we derived through the least squared fitting of the acquired measurements is the following linear one:

$$E(R_O) = CR_O + E_{setup}.$$

In the above equation:

- R_O (Mb/s) is the offloading throughput measured at the transport layer of the tested mobile device.
- $E(R_O)$ (J) is the corresponding transmission energy wasted by the mobile device.
- E_{setup} (J) is the static energy consumed by the mobile device for building up the wireless connection.
- C (J/Mb/s) is the dynamic energy needed for offloading at R_O =1 (Mb/s).

Table 2 reports the measured values of C and E_{setup}, together with the (measured) values of the maximum and minimum offloading rates R_O^{max} and R_O^{min}, and the (measured) round-trip times (RTTs) of the overall TCP/IP connections built up by the mobile devices.

Profile of the Networked Data Center Segment

The measured setup costs and power consumptions of the employed computing server farm are: T_{OFF} = 200 (s), T_{SLEEP} = 60 (s), P_{OFF} = 0 (W), P_{IDLE} = 150 (W), and P_{ON} = 240 (W). The congestion control algorithm of the data-center-oriented TCP transport protocol [14] has been implemented in software for building up reliable end-to-end connections among the instantiated VMs. The resulting per-connection power vs. TCP throughput model we derived through the least squared fitting of the acquired measurements reads as

$$P(R_{DC}) = K(R_{DC})^{\alpha} + P_{setup}.$$

In the above equation:

- R_{DC} (Mb/s) is the throughput of the considered intra-data-center TCP connection.
- $P(R_{DC})$ (W) is the resulting transmission power.
- α is a dimensionless positive exponent.
- P_{setup} (W) is the static power needed for building up the intra-data-center TCP connection.
- K (W × (s/Mb)$^{\alpha}$) is the dynamic power for operating at R_{DC} = 1 (Mb/s).

Table 3 reports the measured (average) values of K, α, and P_{setup}, together with the measured maximum TCP throughput R_{DC}^{max} (Mb/s), and the measured average per-connection RTT (e.g., $\overline{RTT_{DC}}$) (ms).

Mobile Traffic Offloading in StreamCloud: The CDroid Module

StreamCloud relies on the CDroid module in order to efficiently offload computation tasks from the mobile devices to the remote data center [12]. For this purpose, the CDroid is split into two parts as illustrated in Fig. 2: the device-side, residing on the user device; the cloud-side, residing on the data center. According to the P2P service model, in the CDroid framework each mobile device comes with its own cloud-hosted counterpart. The latter acts as just an additional resource of the mobile device, which is placed at a cellular-plus-Internet connection away from it (Fig. 1).

By referring to Fig. 2, the components of the CDroid module involved in energy-saving offloading are:

- The *communication handling module*, which manages all the data traffic over the available mobile Internet connection and also performs the http traffic tunneling of the mobile device through the CDroid-server
- The *caching and prefetching module*, which promptly fetches the content from the CDroid-server, without waiting to connect to the (possibly, overloaded) remote server
- The *traffic compression module*, which performs load shedding by compressing the big data streams offloaded to the remote data center
- The *synchronization module*, which guarantees that newly generated or updated files available at the mobile device are promptly sent to the remote cloud server

The CDroid platform performs real-time traffic offloading by implementing the cognitive-based decision framework recently proposed in [15]. Specifically, the communication handling module of CDroid attains maximization of the per-client offloaded traffic rate at the minimum computing-plus-communication energy cost of the overall multi-tier system of Fig. 1 by allowing each client to opportunistically select the more energy-efficient RAN technology. This is done, in turn, by jointly accounting for:

- The currently available energy and queue backlog at the mobile device
- The fading level affecting the available access links
- The communication-plus-computing energy cost required by the remote data center for processing in real-time the offloaded workload

Additionally, the synchronization module adopts an updating technique similar to Dropbox: it only sends to the cloud side the differences of the content in time. In so doing, it minimizes the differences by automatically saving on the cloud side the content that the user downloads from/uploads to the Internet.

The obtained experimental results show that the CDroid system leads to noticeable reductions in:

- The offloading-induced energy-consumption at the device side
- The execution time of the applications run by the mobile device
- The traffic offloaded over the access network
- The sync-induced energy overhead

Specifically, a comparative examination of an Android system with a bare ThinkAir [15] and a CDroid-equipped Android leads to the conclusion that the ThinkAir tool decides to locally execute the job directly on the phone, due to the high predicted energy costs of transmitting the pictures and the code remotely.

In agreement with these findings, the performed tests point out that the offloading-induced energy consumption (averaged over 100 independent field trials) of the Android OS equipped with CDroid under WiFi IEEE802.11b connections are about 4.7 (J) at 100 (kb/s), 4.5 (J) at 512 (kb/s), 4.2 (J) at 1 (Mb/s), and 4.0 (J) at 10 (Mb/s). Furthermore, they increase up to 22 (J) by using 3G-Universal Terrestrial Radio Access Network (UTRAN) connections, so the mobile device incurs about 239 (J) of energy cost. On the contrary, the proposed CDroid system continuously synchronizes with the data center. As a consequence, thanks to the optimized sync mechanism implemented by the CDroid system, a large part of both the files and code already resides at the data center when the execution of the application is required. The net result is that the job is offloaded even when the input data to process is large, while the energy consumption of the mobile device is about 90 percent lower than that experienced when executing the job locally, even when the most expensive 3G-cellular connection is used [12].

Regarding the sync overhead, the measured data we collected over a two-week experiment in which participants exploited CDroid on their own personal smartphone point out that CDroid is capable of reducing the device-cloud synchronization energy overhead to 2.5 percent of daily battery use. Regarding the reduction of the execution time at the device side, the test applications took 88 (s) to run on the plain Android OS, while only 14 (s) suffice by resorting to the CDroid system. This is due to the fact that the remote data center is equipped with more computing-plus-storage resources, and the data center-to-device synchronization implemented by the CDroid system is tight enough to reduce up to 6.5 times the overall latency of the executed applications. Interestingly enough, the corresponding reduction of the cellular data traffic to be conveyed by the access network of Fig. 1 is also noticeable, and, in our tests, approaches 62 percent.

Real-Time Management of the Data Center Resources: The SeCoM Module

A sketch of the VNetDC architecture subsumed by the StreamCloud paradigm at the cloud side is reported in Fig. 3. It supports the CDroid-server of Fig. 2, which runs at the cloud side on top of the VNetDC of Fig. 3 as a guest OS. Specifically, the platform of Fig. 3 resides at the middleware layer of the underlying protocol stack. It is composed of multiple reconfigurable VMs interconnected by a switched rate-adaptive virtual LAN (VLAN) and managed by a central controller, that is, the StreamCloud Manager (SeCoM). According to some recent contributions on the architecture of VNetDCs [17], time is slotted in Fig. 3, T_t (s) is the slot duration, and the implemented service model is software as a service (SaaS). The multiplexed big data stream received by the Internet backbone during a time slot undergoes load shedding; hence, it is temporarily buffered by the working storage of Fig. 3. At the beginning of each slot, all the current backlog of the working storage is drained and passed to the VNetDC platform as the current input job. In turn, the VNetDC processes the current input job within a time interval that is limited up to a time slot. This ensures that the overall per-job queue-plus-computing-plus-communication delay of the VNetDC of Fig. 3 is limited in a hard way up to $2T_t$.

The task of the virtualization layer of Fig. 3 is to guarantee that the demands for the computing and communication virtual resources of the CDroid-server are mapped onto adequate computing (e.g., CPU cycles) and communication (e.g., link bandwidths) physical supplies. For this purpose, the virtualization layer of the StreamCloud prototype implements the recently proposed *mClock* and *SecondNet* schedulers [13].

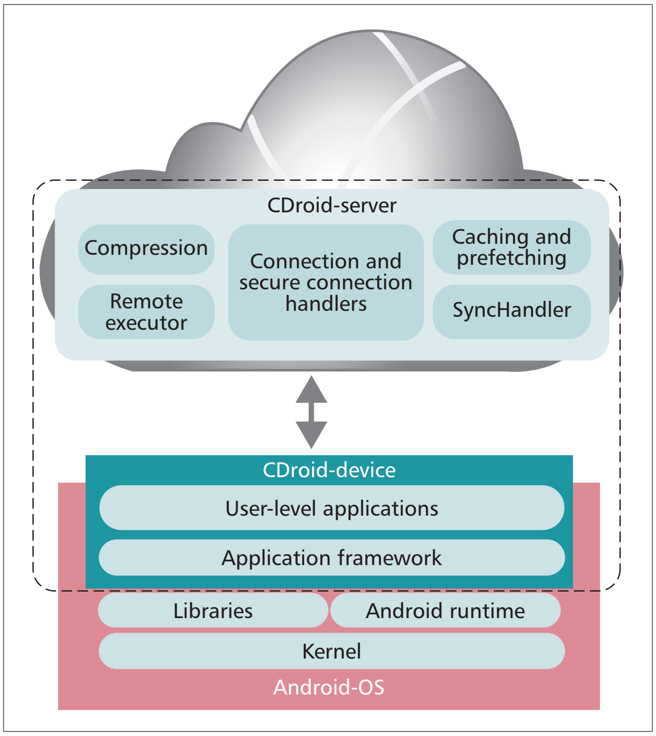

Figure 2. A sketch of the mobile Internet-assisted CDroid architecture.

	Fast Ethernet	Giga Ethernet	10G Ethernet
K ($W \times$ (s/Mb)$^{\alpha}$)	1.7×10^{-3}	1.2×10^{-3}	9×10^{-4}
α	1.3	1.2	1.1
P_{setup} (W)	1.8×10^{-4}	1.2×10^{-4}	9×10^{-5}
R_{DC}^{max} (Mb/s)	32	320	3,200
$\overline{RTT}_{DC}$ (ms)	4	0.4	0.04

Table 3. Measured profiles of the intra-data-center TCP connections.

The SeCoM module is the "core" engine that performs the adaptive joint management of the virtual communication-plus-computer resources and implements the *dynamic virtual manager and load dispatcher* and the *rate-adaptive virtual switch* of Fig. 3. Two key observations inspired the SeCoM design [13]. First, the workload offered by real-time big data streams is highly time-varying and bursty, so forecasting the instantaneous arrival rate is a challenging task. Second, due to the setup costs, too frequent ON/OFF transitions of the currently underutilized computing/communication servers waste energy [17]. Hence, in order to cope with these challenges, the SeCoM supports a novel hibernation state (HIS), which allows currently underutilized physical servers and switches to quickly enter an energy-saving mode, in which only the synchronization task required by the CDroid module is performed. The real-time self-reconfiguration of the overall VNetDC of Fig. 3 is guaranteed by an ad-hoc-designed stochastic gradient algorithm, which ensures global fast convergence to the current minimum-energy resource configuration, even in the presence of unpredictable random fluctuations of the offered workload. These are the main novel features of the SeCoM module.

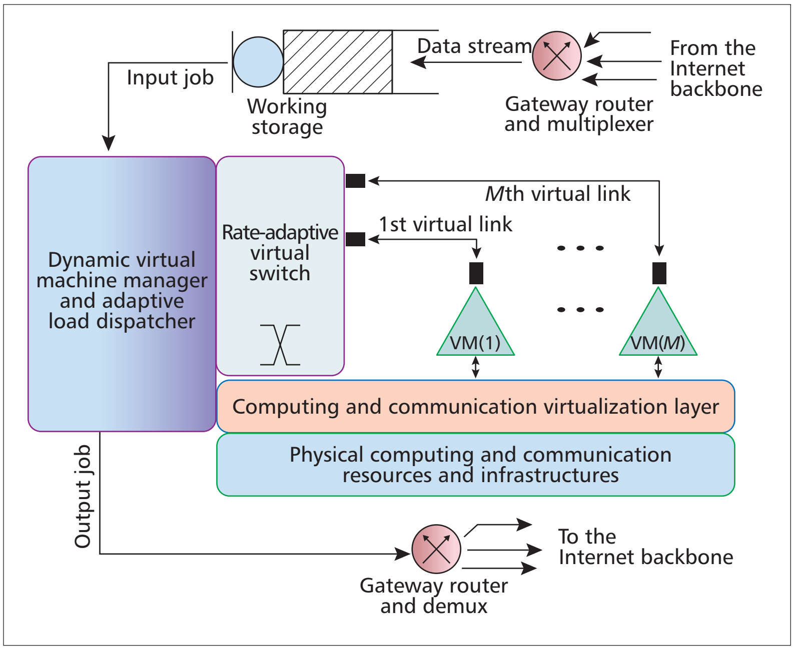

Figure 3. The VNetDC architecture of the StreamCloud platform [13]. Black boxes indicate virtual network interface cards. Each virtual link subsumes an end-to-end TCP-based transport connection.

We have implemented in software a prototype of this module at the driver domain of the para-virtualized Xen 3.3. Xen is an open source software tool that works at the middleware layer. It provides the basic primitives for allowing multiple VMs to share storage, computing, I/O, and networking physical resources [18]. Linux 2.6.18 has been used as a guest OS kernel. In order to perform suitable energy performance comparisons, the hybrid manager (HM) recently considered in [2, 7] has also been implemented as a benchmark resource manager. In order to cope with the abrupt time fluctuations of the workload offered by big data streams, the HM adaptively scales up/down the processing rates of the instantiated VMs, but it holds the communication rates of all implemented virtual links fixed at the maximum values required for transporting the (expected) peak workload.

A typical example of an actual measured workload trace is shown in Fig. 4a; the resulting energy curves (averaged over 100 independent field trials) are reported in Fig. 4b. They refer to an overall per-job processing time limited to 2 s (i.e., $2T_t = 2$). The test workload traces are the ones actually generated by the CDroid module under the aforementioned nomadic computing scenario. An examination of the plots of Fig. 4b leads to three main conclusions. First, the total communication-plus-computing energies of Fig. 4b scale down for increasing value M of the instantiated VMs, and then approach minimum values that do not vary when M is further increased. Second, due to the large average per-job workload featuring big data applications (around 39.6 (Mb) in our tests), the overall consumed energies scale down by passing from the Fast Ethernet LAN to the 10G one. Third, due to the large peak-to-mean ratio (PMR) exhibited by the workload of big data streams (around 2.5 in our tests), the energy reductions of the SeCoM over the benchmark HM are noticeable and approach 27 percent.

These conclusions are also corroborated by the plot of Fig. 4c, which reports the percent energy saving of the SeCoM over the benchmark HM for various values of the time correlation coefficient $\rho \in [0,1]$ of the input workload traces. Deeper load shedding operations performed by the CDroid's compression module of Fig. 2 lead to smaller values of ρ that, in turn, increase the rate of occurrence of abrupt up/down spikes in the resulting workload traces. In agreement with this consideration, the plot of Fig. 4c points out that the percent energy gain of the SeCoM quickly increases for decreasing values of ρ. In our tests, it approaches 30 percent for values of below 0.55.

Conclusions and Hints for Future Developments

A key challenge for coping with the unpredictable volume of data generated by the emerging BDMSC applications is the design of integrated computing-networking technological platforms, which allow fully adaptive energy-efficient joint reconfigurations of the virtual resources available at data centers and mobile devices under hard real-time constraints. In this position article, we identify both the major opportunities and challenges of the BDSMC paradigm, and point out some relevant Internet-based mobile applications that could benefit from this paradigm. General concepts have been coupled with a concise presentation of the StreamCloud architecture and prototype, in order to contextualize the presentation within a practical framework. The reported numerical results corroborate the conclusion that noticeable energy savings may be attained by the resource management framework implemented by StreamCloud over the state-of-the-art HM.

In this regard, we observe that the performance of the proposed StreamCloud platform could be improved by moving along two main research directions. First, the current prototype does not perform forecasting of the volume of offloaded data. Hence, the design of reliable techniques for real-time forecasting of the workload offloaded by big data streams is a first topic for future research. Second, in order to reduce both the processing latency and congestion of the Internet backbone, the second tier of the BDSMC architecture could also host local Fog centers. By design, Fog centers cooperatively process the less intensive offloaded tasks, while forwarding the most expensive ones to remote data centers [19]. Hence, the development of CDroid-supported energy saving mechanisms/protocols for implementing live migrations of VMs among Fog centers, and between Fog centers and remote data centers is a second promising research topic.

References

[1] S. Schneider, M. Hirzel, and B. Gedik, "Tutorial: Stream Processing Optimizations," *ACM DEBS*, June 2013, pp. 249–58.
[2] M. Shojafar *et al.*, "Energy-Saving Adaptive Computing and Traffic Engineering for Real-Time-Service Data Centers," *IEEE ICCW '15*, 2015, pp. 1800–06.
[3] "Big Data in Big Companies," SAS research rep., 2013.
[4] L. Neumeyer *et al.*, "S4: Distributed Stream Computing Platform," *Proc. ICDMW '10*, 2010, pp. 170–77.
[5] M. Zaharia *et al.*, "Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters," *HotCloud*, 2012.
[6] Z. Qian *et al.*, "TimeStream: Reliable Stream Computation in the Cloud," *EuroSys*, 2013.
[7] A. Kumbhare, Y. Simmhan, and V. K. Prasanna, "PLAstiCC: Predictive Look-Ahead Scheduling for Continuous Data-Flows on the Clouds," *CCGRID 2014*.
[8] Z. Zhou *et al.*, "Carbon-Aware Load Balancing for Geo-Distributed Cloud Services," *Proc. IEEE MASCOTS '13*, 2013, pp. 232–41.
[9] D. M. Divakaran, T. N. Le, and M. Gurusamy, "An Online Integrated Resource Allocator for Guaranteed Performance in Data Centers," *IEEE Trans. Parallel Distrib. Sys.*, vol. 25, no. 6, June 2014, pp. 1382–92.
[10] M. V. Barbera *et al.*, "To Offload or Not to Offload? The Bandwidth and Energy Costs of Mobile Cloud Computing," *IEEE INFOCOM 2013*.
[11] K. Lee et al., "Mobile Data Offloading: How Much Can Wifi Deliver?" *IEEE/ACM Trans. Networking*, vol. 21, no. 2, Apr. 2013, pp. 536–50.
[12] M. V. Barbera *et al.*, "Mobile Offloading in the Wild: Findings and Lessons Learned through a Real-Life Experiment with a New Cloud-Aware System," *IEEE INFOCOM 2014*.

[13] N. Cordeschi *et al.*, "Energy-Efficient Adaptive Networked Datacenters for the Qos Support of Real-Time Applications," *J. Supercomputing*, vol. 71, no. 2, Feb. 2015, pp. 448–78.
[14] M. Alizadeh *et al.*, "Data center TCP (DTCP)," *ACM SIGCOMM*, 2010.
[15] N. Cordeschi, D. Amendola, and E. Baccarelli, "Resource-Management for Vehicular Real-Time Applications Under Hard Reliability Constraints," *IEEE/ACM DS-RT '14*, 2014.
[16] S. Kosta *et al.*, "Thinkair: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading," *IEEE INFOCOM 2012*.
[17] N. Cordeschi, M. Shojafar, and E. Baccarelli, "Energy-Saving Self-Configuring Networked Data Centers," *Computer Networks*, vol. 57, no. 17, Oct. 2013, pp. 3479–91.
[18] D.Chisnall, *The Definitive Guide to the Xen Hypervisor*, Prentice Hall, 2007.
[19] I.Stojmenovic, and S.Wen, "The Fog Computing Paradigm: Scenarios and Security Issues," *Proc. FedCSIS 2014*.

Biographies

ENZO BACCARELLI (enzo.baccarelli@uniroma1.it) is a full professor in information and communication engineering at the DIET Department of Sapienza University of Rome. He received his Laurea degree in electronic engineering and Ph.D. degree in communication theory and systems from Sapienza University. In 1995, he received a post-doctorate degree in information theory and applications from the INFOCOM Department of Sapienza University of Rome. His current research focuses on data networks, distributed computing, and adaptive optimization.

NICOLA CORDESCHI (nicola.cordeschi@uniroma1.it) is a postdoctoral researcher in information and communication engineering at the DIET Department of Sapienza University of Rome. He received his five-year Laurea degree (summa cum laude) in communication engineering and Ph.D. degree in information and communication engineering from Sapienza University of Rome in 2004 and 2008, respectively. His research activity focuses on data communication, distributed computing, and design/optimization of QoS transmission systems for wireless/wired multimedia applications.

ALESSANDRO MEI (mei@di.uniroma1.it) is a full professor in computer science at Sapienza University of Rome. His research focuses on distributed, networked, and cloud systems and security, and social networks. He did his studies and worked at the University of Pisa, Trento University, and the University of Southern California. During 2010–2011, he was a Marie Curie visiting professor at the University of California, San Diego. He has received several international awards for research achievements, including the Google Faculty Research Award 2013.

MASSIMO PANELLA (massimo.panella@uniroma1.it) is an associate professor of circuit theory and learning systems for pervasive computing at Sapienza University of Rome. He received his five-year Laurea degree (summa cum laude) in electronic engineering and Ph.D. degree in information and communication engineering from Sapienza University of Rome in 1998 and 2002, respectively. His research activity pertains to circuit theory, computational intelligence, and pervasive computing for modeling, optimization, and control of complex systems.

MOHAMMAD SHOJAFAR (shojafar@diet.uniroma1.it) is currently a Ph.D. student in information and communication engineering in the DIET Department of Sapienza University of Rome. He received his B.S. in computer engineering-software from Iran University Science and Technology of Tehran (2001–2006), and his M.Sc. from Qazvin Islamic Azad University (2007–2010). His current research focuses on data communications, distributed computing, and optimization.

JULINDA STEFA (stefa@di.uniroma1.it) is a postdoctoral researcher in computer science at Sapienza University of Rome. She obtained her Ph.D. (2010) and Laurea degree in computer science (2006), summa cum laude, from the same university. She received the Working Capital PNI price offered by Telecom Italia in 2011, and the IEEE INFOCOM 2013 and IEEE SECON 2013 Best Demo Awards. Her research focuses on distributed and networked systems and security, mobile cloud, and social mobile networks.

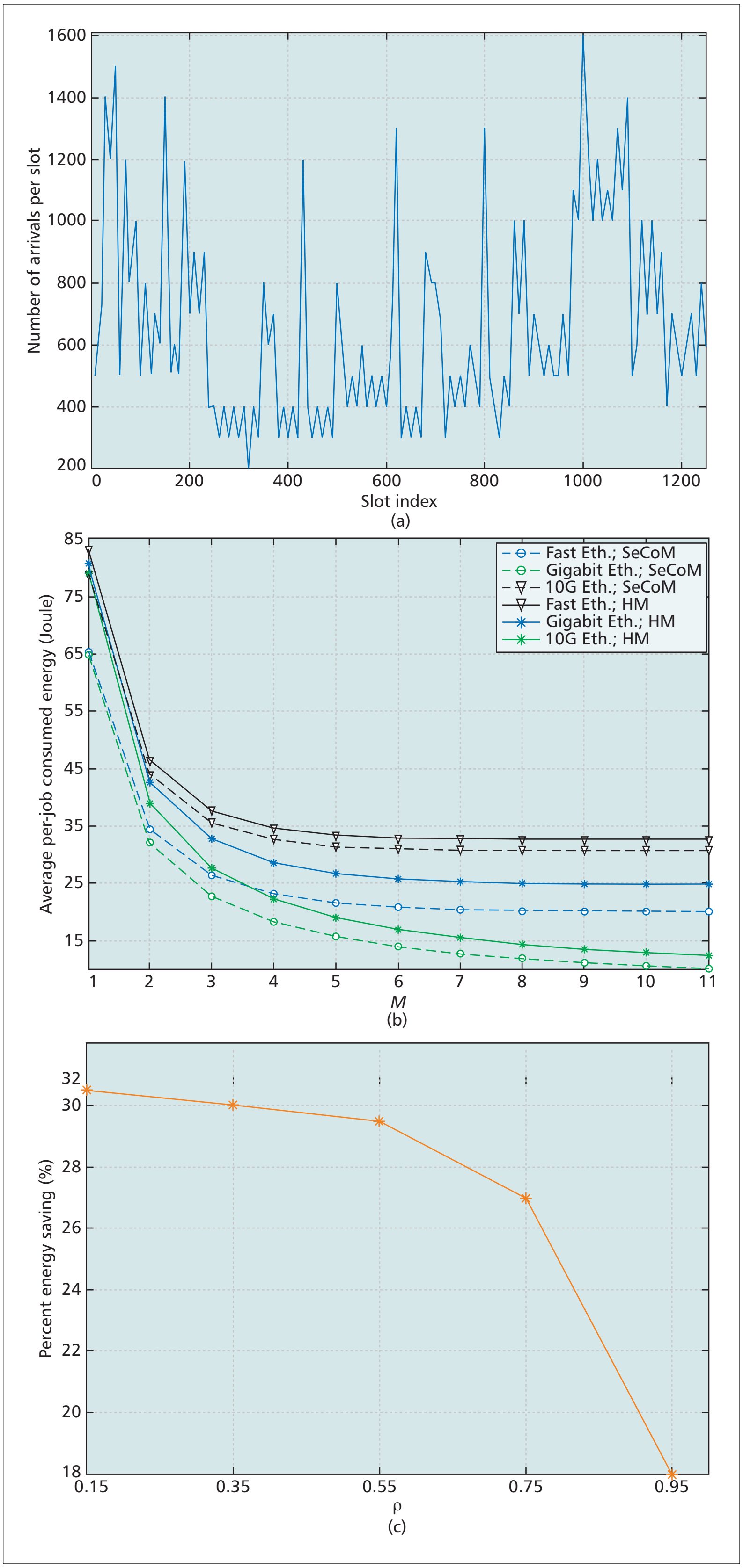

Figure 4. a) An instance of measured workload trace for the considered BDSMC scenario. The corresponding *PMR* is 2.5, and the workload carried out by each arrival is 62 (kb); b) effects of the aggregate rates of the tested Ethernet LANs on the energy consumption of the SeCoM (dashed plots) and HM (continuous plots); c) percent energy saving of SeCoM over HM at $PMR = 2.5$ and $M = 10$ for the10G Ethernet LAN.