Advancing the State of Mobile Cloud Computing

Paramvir Bahl[‡] Richard Y. Han[†] Li Erran Li[⋆] Mahadev Satyanarayanan[*]

Microsoft Research[‡] University of Colorado at Boulder[†] Bell Labs[⋆] CMU[*]
bahl@microsoft.com rhan@cs.colorado.edu erranlli@research.bell-labs.com
satya@cs.cmu.edu

ABSTRACT

The capabilities of mobile devices have been improving very quickly in terms of computing power, storage, feature support, and developed applications. However, these mobile applications are still intrinsically limited by a relative lack of bandwidth, computing power, and energy compared to their tethered counterparts. Cloud computing offers abundant computing power that can be tapped easily. Apple iCloud and Amazon Silk browser are two recent mobile applications that leverage the cloud. In this paper, we systematically explore the fundamental research questions when combining mobile and cloud computing. We will highlight some of the challenges we face and some of the solutions we are pursuing.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Client/server

General Terms

Design

Keywords

Mobile cloud, programming models, platform services

1. INTRODUCTION

We are living in a compelling new era for mobile computing. Technological innovations are occurring at an accelerated rate: (1) increasingly, mobile devices are much more capable in terms of processing speed and storage; and (2) the wireless network is becoming much faster and has lower latency, with new deployments such as LTE shaping the field.

Parallel to these innovations, cloud computing has soared in popularity. The cloud computing paradigm offers a novel approach for utility computing with unprecedented resource flexibility, agility, and scalability [3]. A recent report by Gartner research [10] predicts that cloud computing is poised for active enterprise adoption within the next two to five years.

Compared with their tethered counterparts, mobile devices are intrinsically limited by computing, storage and energy limit. This is fundamental to mobile computing. Given the abundance of and easy access to public cloud computing resources, the natural question to ask is, can cloud computing bridge the resource gap of mobile computing?

The answer is definitively yes. Recently, we have witnessed several cases which cloud computing is called in to solve mobile computing problems. Apple's iCloud stores customers' music, photos, apps, calendars, documents, etc, and wirelessly pushes them to all their devices automatically. Apple's iCloud stores are hosted in Amazon EC2 and Microsoft Azure. Amazon has released its new "cloud-accelerated" Web browser Silk. Silk a "split browser" whose software resides both on Kindle Fire and EC2. With each web page request, Silk dynamically determines a division of labor between the mobile hardware and Amazon EC2 (i.e. which browser sub-components run where) that takes into consideration factors like network conditions, page complexity and the location of any cached content. We refer to mobile applications that leverage the public cloud (e.g. Amazon EC2 and Windows Azure) as mobile cloud applications or mCloud apps for short. We refer to the research area of mobile computing that taps in cloud resources as mobile cloud computing or mCloud computing for short. The public cloud today are designed for enterprise applications without any explicit consideration of mobile applications. Mobile computing demand fundamental changes to the public cloud. We refer to a public cloud that supports mobile applications seamlessly as mCloud.

How to transition from a cloud with no explicit support of mobile applications to mCloud? In this paper, we try to address this question systematically from multiple perspectives below.

- Cloud computing has been designed for enterprises. The public cloud computing infrastructure that exists today may not be the perfect architecture to support mobile computing. The question is what architectural support does mobile computing need besides what current cloud computing offers?
- What is the programming model for mobile devices to tap into public cloud computing resources? Do we need to tightly synchronize mobile devices with the cloud and treat mobile devices as just a display? Should our computing unit be a virtual machine (VM) or a method invoked by a remote method invocation (RMI)?
- What are the basic services or building blocks public cloud computing can offer to mobile applications?
- What mechanisms should cloud computing offer in order to foster a new generation of collaborative mobile applications?

In this paper, we will address each of these fundamental questions. we will put forth a vision of mobile computing that breaks free of

MCS'12, June 25, 2012, Low Wood Bay, Lake District, UK.
Copyright 2012 ACM 978-1-4503-1319-3/12/06 ...$10.00.

the fundamental constraints that have been keeping us from discovering an entirely new world in which mobile computing seamlessly augments the cognitive abilities of users using compute intensive capabilities such as speech recognition, natural language processing, computer vision and graphics, machine learning, augmented reality, planning and decision making. By thus empowering mobile users, we could transform many areas of human activity like never before. In this vision, mobile users seamlessly utilize the cloud to obtain the resource benefits without incurring delays and jitter and without worrying about energy. We will highlight some of the challenges we face and some of the solutions we are pursuing.

The rest of the paper is organized as follows. In Section 2, we motivate the need for offloading tasks of mobile applications. In Section 3, we present mCloud architecture. In Section 4, we discuss mCloud app programming models. In Section 5, we outline the basic building blocks of mCloud apps. In Section 6, we propose mCloud system support for service interaction so that collaborative services can be built. We conclude in Section 7.

2. THE NEED FOR OFFLOADING

A new generation of mobile applications in Apple Appstore, and Google Android Marketplace, etc are pushing the boundary on how we interact with the physical world and the cyber world. For example, a new design of a floor plan overlays on top of the physical floor allows the user to vividly test how a design will look like as if it is realized in the physical world. A navigation system that points to the recognized street signs, and blinks or speaks to the user is much easier to use than traditional navigation system that is only based on GPS coordinates. Sophisticated multi-player shared games will require pose and gesture recognition, and rich graphics.

A common theme of these applications is that they require compute intensive capabilities such as speech recognition, natural language processing, computer vision and graphics, machine learning, augmented reality, planning and decision making. These capabilities run counter to the resource poverty nature of mobile devices. This constraint is not just a temporary limitation of current technology, but is intrinsic to mobility.

On one hand are small form factor handheld devices and on the other is the cloud, a nearly limitless pool of computing resources that is being heavily touted as the future of computing. It is natural to connect and combine the two to enable a new class of CPU and data intensive applications that seamlessly augment the cognitive abilities of users.

3. MOBILE CLOUD COMPUTING SYSTEM ARCHITECTURE

We first give an overview of the current cloud computing architecture, and how current mobile cloud services make use of cloud computing resources. We then discuss recently proposed alternative or complimentary architectures—Cloudlet and peer. Finally, we offer our vision and open questions.

3.1 Current cloud computing architecture and mobile cloud services

Current cloud computing providers typically allow customers to rent computation and storage such that customers can start instances of their cloud applications as VMs within the provider's cloud of servers. Cloud providers may provide additional services, such as backup and traffic accounting, to ease the process of managing VM instances. The distribution of the VM instances is largely transparent to the customers, and cloud providers mainly focus on providing guarantees of CPU time, memory usage, storage, server availability, networking throughput, etc. However, some cloud providers offer customers the additional ability to choose geographically from among a small number of data centers where their VM instances will run, e.g. Amazon has several regional data centers such as US East, US West, etc. The intent is to lower network latency by locating data centers near where their output will be used, and as such these data centers are mostly located in places with large population densities.

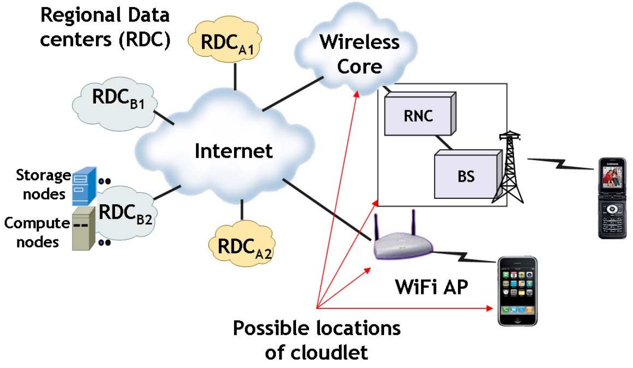

Figure 1: Components of mobile cloud architecture

Such cloud computing is suitable and popular for small startups and medium-sized businesses, since the management of servers and many basic application services can be outsourced to the cloud. Its suitability for large organizations is still being proven in the marketplace, as each large company must investigate the price/performance tradeoff between building and managing their own private cloud or contracting out those services to a third party cloud as traffic scales to high volumes. A key consideration that factors into this decision is whether an organization wishes to store its private or proprietary data on a third party's cloud, and to what extent that cloud provider provides protection to ensure the privacy of such data.

We envision that the future of cloud computing will be heterogeneous, and include many diverse clouds with different capabilities and protections, offered by different vendors. A large company that builds its private cloud may still bridge into a larger public cloud for some of its services. The diverse application-level services embedded within these various clouds will likely be merged in a seamless manner via interoperable standards based on Web services that span these heterogeneous clouds.

Today's mobile applications have already begun to adapt to cloud computing. A common theme emerging from the large wave of mobile applications developed for smartphones such as the iPhone and Android is that these mobile applications are often linked to server instances operating in the cloud. However, there is much duplication of effort, as these server instances reimplement many of the same elements of mobile support, such as location awareness, adaptation to mobility, and computational partitioning of execution between the mobile and the cloud.

We believe that fusing mobile and cloud computing will require a rethinking of the architecture of cloud computing to accommodate common themes of mobile computing, including adaptation to limited resources and mobility.

3.2 The case for a middle tier

A natural question to ask is will distribution to the closest regional data center be enough? The key to answering this question is the end-to-end performance such as bandwidth, delay and jitter.

Even with LTE, access to the closest data center will incur a latency of at least 70ms. This latency can still be problematic for perception applications. For example, it is reported in [19] that transmitting a large image to a server on a network with RTT of 40ms degrades the frame rate to 1.8 frames per second whereas in a LAN of 100Mbps, the frame rate is about 8 frames per second. Perception applications call for a middle tier such as Cloudlet [21]. A cloudlet is a trusted, resource rich computer or cluster of computers that is well connected to the Internet and is available for use by nearby mobile devices.

One incentive for wireless providers to deploy computing and storage nodes is to reduce resource consumption within its access network. The possible places for deploying resources that are closer to mobile devices are the wireless access networks, WiFi hotspots, peer mobile devices. The key advantage of deploying cloudlets in wireless access networks is that there is minimal security, privacy and trust problems because wireless providers see all traffic from its subscribers. It also simplifies billing. The key drawback for deploying cloudlet type of resources in public WiFi hotspots are the lack of security, trust and billing infrastructure.

3.3 Cloud Infrastructure Optimization for Mobile Applications

The performance of public cloud infrastructure is adequate for many mobile applications. However, they may fall short for certain demanding mobile applications. One such type of application is social games which are largely played on mobile devices. Unlike most web applications such as e-commerce or search which are read heavy, social games are write heavy. This is due to the interaction between the user and the game state and between users themselves. In social games the ratio of reads to writes can be as high as 1:1. In addition, to achieve good user experience, social games require low latency and high availability. As a result, the leading social game company, Zynga built its own cloud, zCloud [24]. zCloud is designed specifically for social games in terms of availability, network connectivity, server processing power and storage throughput.

zCloud provides redundant power to each rack, uses state-of-the-art server with high memory capacity. It is a fully non-blocking network infrastructure and uses in-line hardware-based load balancers and local disk storage. zCloud also optimized game servers [23]. Instead of using Memcache and MySQL, zCloud makes use of Membase. Membase has built in persistence and replication mechanism. Membase is also optimized with write throughput besides reads. As a result, zCloud offers 3 times the efficiency of standard public cloud infrastructure. For example, where Zynga games in the public cloud would require three physical servers, zCloud only uses one.

3.4 Leveraging peer mobile devices

It has been demonstrated [12, 16] that one can leverage peer mobile devices to perform cloud computing functions. A system called Misco [16], a version of MapReduce, can be handled by a "server farm" comprised of 20-odd Nokia N95 smartphones. The choice of using peer mobile devices for cloud computing faces many other hurdles. The security, trust, privacy issue is even greater. There is also the incentive issue.

3.5 Our vision and research agenda

Our vision of a mCloud architecture is the seamless integration of cloudlet and public cloud, and infrastructure specialization for mobile applications. We believe the dominant architecture will be the regional data centers of public cloud providers. Cloudlet is necessary to reduce the delay of latency sensitive perception applications. There are two convincing deployment settings. One is for wireless providers to deploy cloudlet like nodes within their wireless access networks as a premium service for its subscribers. The other is for cloud providers to co-locate cloud resources in wireless access networks through co-location agreement with wireless providers.

For optimal performance, we believe the middle tier needs to be integrated with the region data centers of public cloud seamlessly. Seamless integration reqires the following:

- The network needs to support high bandwidth and low latency connection to the regional data centers of public cloud. This can be achieved through various VPN technologies such as BGP/MPLS VPN. This support is crucial for fast migration of computation and data from Cloudlet to the public cloud due to local resource overload.
- Cloudlet and public cloud needs to support high performance VM migration. When Cloudlet faces resource limitation, this support makes it easy for the Cloudlet to seamlessly migrate the VM to the public cloud. Support for RPC, thread migration can also be very helpful.
- Cloudlet and public cloud should have a common computing platform, and the cloud should support "automatic resource augmentation". For example, a computing job at Cloudlet may have access to a few VMs. When the job is migrated to and executed in the cloud, the cloud should automatically expand the job to use many more VMS, e.g. hundreds according to application needs or service agreements. MapReduce is such a common computing platform which makes automatic resource augmentation easier.
- Cloudlet should store a copy of persistent data to the public cloud, and should keep this loosely synchonized.

As the zCloud example shows, public cloud infrastructure needs to be specialized for mobile applications. We believe server, file system, networking, and memcache technologies should all be specialized for mobile applications.

In our future research, we will pursue these open research topics.

4. PROGRAMMING MODELS

How should mobile applications tap into the resources of public cloud? In other words, what components should run local and what should be done in remote? This will depend on the application, the device capability and the operating environment (e.g. delay, bandwidth). There are several factors to consider. First, what is the objective? Various objectives are offload computation, reduce latency, minimize energy consumptions, etc. Second, we need a profiler to understand the resource usage of the components, and the impact of offloading. Third, we need a solver to decide what to offload. Profiler and decision engine are common to all programming models, and very similar. We do not go into details of these two components.

4.1 Existing Programming Models

The first is the recently proposed CloneCloud [8]. It is proposed that the mobile device will have a clone copy in the cloud. The two work in synergy to enhance the application experience while minimizing resource consumption. This programming model allows a mobile client to fully utilize the cloud resources. The optimization solver decides on what executions should be offloaded based on a dynamic profiler and a static analyzer. Remote execution mechanism is thread migration. CloneCloud leverages an application level VM which is an abstract computing machine that provides hardware and operating system independence.

The second one makes use of RPC to remotely execute resource intensive methods. MAUI [9] proposes such a programming model. It leverages Microsoft .NET runtime to annotate methods that are remotable. A profiler at both the client and server will evaluate whether it is beneficial to remote a method. Since RPC is widely used in client server computing, this programming model can be readily used to take advantage of cloud resources. However, it is language and platform dependent.

The third one is Odessa [19] which imposes a specific programming model. Application programmers have to structure the application as a data flow graph. This model is well suited for media processing applications that perform a series of operations to an input video or audio stream. The vertices of the graph are processing steps called stages and the edges are connectors which represent the data dependencies between the stages. Stages do not share any state. This model allows programmers to express coarse grained application parallelism while hiding the complexity of parallel and distributed programming from the developers. The advantage of this model is that it enables parallel processing. However, this model does not support existing applications.

The fourth one, Orleans [7] is proposed for cloud computing. It has no mobile computing support currently. Orleans is a software framework for building reliable, scalable, and elastic cloud applications. It is based on distributed actor-like components called grains. Grains are isolated units of state and computation that communicate through asynchronous messages. Since the Orleans runtime provides scalability, availability, and reliability, application developers can focus on application logic. Because of the natural isolation of grains and the Orleans runtime support, Orleans looks very promising as a programming model for mCloud.

4.2 A case for a RESTFUL Programming Model

We propose a fifth alternative. Since many media applications make use of standard components for face recognition, gesture recognition, object and pose recognition, packaging them as cloud services with standard APIs can be more appealing. This motivates our RESTful programming model. It is inspired by the Amazon EC2 API. For RESTful model, there is no state kept in the cloud. Whenever a computing task is needed, the mobile device just invoke a function with appropriate parameters through http or https protocol. For inter-operability among cloud providers, the API has to be standardized. There is also the issue whether there are enough meaningful common functions that mobile devices typically use. The functions we have in mind are invoking an image recognition software with appropriate input parameters. For example, a user can take a picture of a book, and asks the cloud service to extract the text in the picture. A user can take a picture of the Statue of Liberty and ask for the current location. This programming model is suitable for well-defined tasks of common services. As a real example, recently Google Android offers a speech recognition service to developers. This service is stateless.

4.3 Comparison and Use cases

Comparison of the programming models: Table 4.3 is a comparison of these models in terms of whether mobile applications at the client side blocks or not when tasks are migrated to the cloud, how much state is kept at the cloud, and the remote execution unit.

Programming model usage cases: We believe that several programming models will co-exist. Which one to use will depend on the application context. For example, certain well-defined tasks such as speech recognition can make use of the RESTful design. Rich media processing applications such as gaming may want more control on how the media is processed. For example, an application may want to implement a fast algorithm to extract text from images which may not be available from generic image recognition service built using RESTful model. In this case, Odessa will be an ideal programming model.

Models	Blocking	Cloud state	Remote exec. unit
CloneCloud	Yes	full thread	Thread
MAUI	Yes	partial	Method
Odessa	Yes	partial	App task
Orleans	No	partial	Grains
RESTful	No	No	Cloud task

Table 1: Comparison of different programming models

Unlike RESTful and Odessa, MAUI, CloneCloud and Orleans are applicable to any application. MAUI and CloneCloud are based on traditional RPC and thread migration respectively. MAUI is more fine-grained. However, MAUI requires modification of existing applications. CloneCloud supports existing applications. Grains are ideal for applications that need to manage persistent data. Since it is new, it does not support existing applications. In addition, it does not provide specific support for mobile applications.

4.4 Programming model implementation in public cloud

Amazon has recently provided mobile access to its cloud computing services [2]. Android or Apple iOS developers can now create applications that will enable users to access Amazon EC2: S3, SimpleDB, Amazon Simple Queue Service, Amazon Simple Notification Service, Amazon CloudWatch, Amazon Simple Email Service, Elastic Load Balancing, and Auto Scaling, all from their mobile devices. With this mobile access capability, one can implement all five programming models using Amazon EC2.

4.5 Our vision and research agenda

We believe there are still lots to be done for mCloud programming model. The key is to hide the complexity from the developers, and support legacy applications with minimal or no change. We believe the key challenge is to support mobile perception applications. These applications often require cognition and recognition from user input data. These tasks are computational intensive even when operated on user input data alone. In addition, these tasks often involve computation on "big data". For example, speech recognition or language translation benefits from matching user input data with these big data.

Many perception applications make use of the OpenCV library [18] to process images or videos. Instead of letting every such application to program in the Odessa model, we propose to provide system support for offloading tasks of OpenCV. Our system, mCloudCV will determine in run time whether a OpenCV task should run local or offload to the cloud.

We believe there is a need for reliability. For example, suppose an application offloads a bank transaction to the cloud using RPC. If the transaction succeeds, but RPC fails to return to the client. The client may perform the same transaction directly. This may end up with execuating the transaction two times, e.g. debit $2X$ dollars instead of X dollars. Orleans provide reliability. However, it does not support existing applications. The question is, can we provide reliability without changing existing applications?

5. BASIC MOBILE CLOUD COMPUTING SERVICES

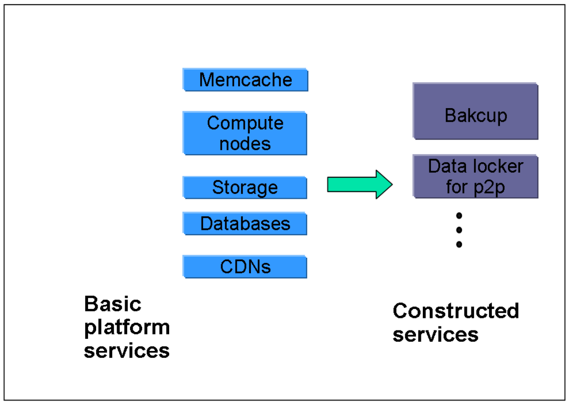

Figure 2: Platform services

We envision cloud computing providers will provide a set of basic services for mobile computing. There are three types of services. The first one is what we refer as *platform services*, the second is *application services*, and the third is *context-rich support services*.

5.1 Platform services

Platform services include computing, storage, database, memcache, content distribution as shown in Figure 2. Currently all EC2 services accessible from mobile devices are considered platform services. Some of these basic services can benefit from application sharing. Take distributed memcache service for an example. Many application may create same or access same data sets. With a shared memcache service, it will be more likely to have a cache hit due to the larger cache size. It will reduce computation demand to re-generate the cached results. Of course, sharing bring forth the issues of security, privacy as well as how much storage each application should have.

Out of the basic platform service, one can already build very useful applications. For example, with storage service, and computing service, one can build file backup service, and file syncing service (keep all registered devices in sync of the user content). One can also build a data locker service [1]. In essence, the data locker protocol works with p2p protocols closely to service files on behalf of end hosts. It is particularly appealing in the mobile device context as it minimizes the usage of wireless access links.

5.2 Application services

Public cloud provider can also offer a set of essential application services. For example, people may not trust each individual applications and thus, may not reveal their location information. This can hamper the development of location based services. If mobile devices are using the cloud services, then there is prior trusted relationship. For example, Apple iCloud users are comfortable that their private data will be protected from un-authorized use. So it is easier to trust the cloud provider for location privacy. Thus, a presence service can be an essential service so that any application that needs location information can talk to the presence service. The presence service will implement location privacy policies according to what are stipulated by the mobile subscribers. We recognize that different people have different level of privacy requirements. It is conceivable that some people may not want to sign up with a presence service. However, the presence service will facilitate the development of location-based services. Presence service will save resources as it is not replicated for each location based application.

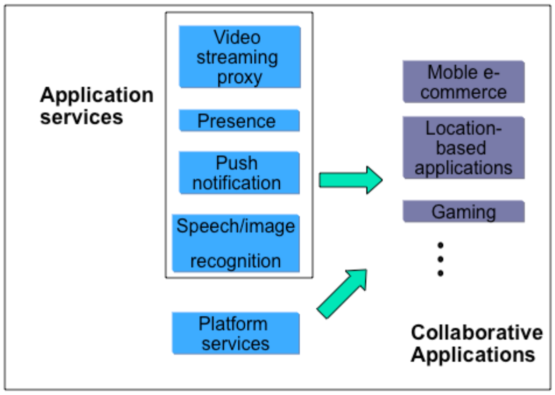

Figure 3: Application services

Given the popularity of video streaming applications, cloud providers can offer a video transcoding and streaming proxy. The reason is that mobile devices are limited by the availability of video codec as well as bandwidth variability. A proxy service alleviates this problem by performing transcoding. In addition, the proxy can take advantage of certain codec's inherent bandwidth adaptation capability, for example, H.264SVC can adapt in three dimensions with finer granularity of network bandwidths.

Many mobile applications need to send push notification to mobile devices. Because many mobile devices are behind NAT, in order to send push notification, a persistent TCP connection is needed. To maintain such a persistent TCP connection, periodic heartbeat messages have to be sent. Thus, it will be very inefficient if each application has to maintain a persistent TCP connection. To avoid such situation, Android offers a push notification service through an API so that one TCP connection is maintained between a mobile device and a Google server for the purpose of push notification.

Push notification services typically are used by servers to reach mobile clients. To allow mobile devices to communicate with each other, Microsoft Research Project Hawaii [14] has developed a relay service. The Hawaii Relay Service provides a relay point in the cloud that mobile applications can use to communicate. It provides an endpoint naming scheme and buffering for messages sent between endpoints. It also allows for messages to be multicast to multiple endpoints.

There are many applications that do speech and image recognition. It makes sense to provide a common service to implement the best algorithm while amortizing the cost. In fact, Google Android have a speech recognition API which enables developers to integrate speech input capabilities into their applications. Developers stream audio to Google's servers which then convert speech into text and feed it back to the applications. Project Hawaii [14] also provides a speech to text service.

5.3 Context-rich services

We envision that many mobile applications will become more personalized, and more context aware, recognizing not only the location of the user and the time of day, but also a user's identity and their personal preferences. To support these mCloud services, we believe mCloud providers need to provide a set of context-rich support services. Application developers can use these context-rich support services as building blocks to build a large class of new mCloud services. We envision several context-rich support services such as context extraction service, recommendation service, and group privacy service. Context extraction service provides data mining analysis of mobile data combined with other forms of data, such as social networking data and sensor network data, in order to extract contextual clues relevant to the user. For example, recognizing the user's activity based on mobile accelerometer and audio data is one such contextual mining service that is currently being explored [15]. The context extraction service will be a common service that relieves each context-rich application from replicating context extraction, thus saving energy and reduce computation costs of mobiles.

Based on these contextual clues, a layer of cloud recommendation services can be built that creates output that is tailored to an individual or set of individuals with those contextual characteristics. For example, some applications have begun to combine together mobile location with social networks to generate multimedia content, e.g. a song playlist or a recommended video [4], that is tailored to the individual or individuals who are nearby an audio jukebox or video screen that is aware of their presence.

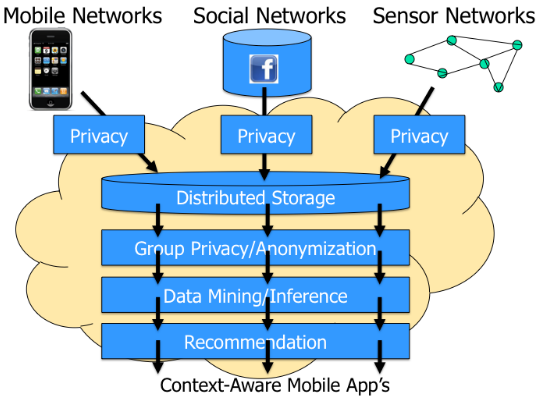

Figure 4: Privacy, Data Mining, and Recommendation Services in the Context-Aware Mobile Social Cloud.

Such contextual mobile applications would be composed as shown in Figure 4. This architecture fuses together multiple layers of cloud application services, as described in the SocialFusion architecture [6], wherein mobile, social, and sensor networks supply streams of data into a distributed storage service. Data mining/inference cloud services then operate on the assembled data to extract contextual clues. Finally, recommendation services in the cloud generate tailored multimedia output, either for the mobile device or for nearby multimedia devices such as LCD displays or loudspeakers.

We imagine that privacy protection services will emerge as a key component of context-aware mobile cloud services, as there is a fundamental tradeoff between supplying personal information to receive contextual services, and revealing too much private information for those services. Location privacy has already been discussed, but we think that new privacy services will have to be developed to protect user data from data mining services that analyze mobile smartphone data, such as activity recognition services. New privacy services will also need to be devised to protect and anonymize information released from social networks [5] and sensor networks [11, 20]. Moreover, we believe a new concept of "group privacy" or "collective privacy" will emerge, requiring privacy services that protect groups of individuals from collective inferences on their joint actions, tastes, and preferences.

6. SUPPORT FOR COLLABORATIVE APPLICATIONS

Mobile participatory sensing applications are becoming increasingly popular. In such applications, large numbers of mobile devices contribute their own sensor data, such as video clips, image captures, audio snippets, temperature data, location information, and/or text metadata to a collaborative application located in the cloud. This cloud application then generates compelling crowd-sourced output that could not otherwise be easily obtained. Example applications include traffic jam/congestion detection [17], bus arrival forecasting [22], parking space discovery [13], localization of weather phenomena, distributed pollution detection, etc. We imagine that such collaborative applications will expand to include mobile epidemiology and disease outbreak detection, spontaneously coordinated crowd activities at concerts and sporting/cultural events, etc.

New cloud infrastructure beyond the application services proposed above for single mobile applications will be needed to support such large scale collaborative mobile applications. Cloud-based data mining services will need to scale to analyze large groups of people and the large quantities of data that they generate in order to extract collective trends among the population of users in real time. In addition, new crowd actuation services will need to be created and scaled, such as recommendation services based on collective group context rather than individual context. Privacy services that scale to large numbers of people, and preserve a sense of "collective privacy of the group" will become more important and will need to be devised.

Because applications servicing a region co-locate in regional data centers, there are ample opportunities for synergy. Co-location enables intimate collaboration of applications and performance optimization which are not possible before. For example, for file sharing, traditionally, the application has to transfer the file remotely. If it is a big file, many applications will not work, e.g. collaborative games. If the two applications are co-located, rather than sending the file, a pointer will be sufficient. If server 1 of application A talks to server 2 of application B a lot, then the cloud provider can even co-locate these two servers in the same physical machine.

7. CONCLUSION AND FUTURE WORK

Today's mobile applications are demanding compute intensive capabilities such as speech recognition, natural language processing, computer vision and graphics, machine learning, augmented reality, planning and decision making. These demands will not be met solely by making more powerful mobile devices. Mobile computing is poised to demand fundamental changes to cloud computing such as programming models to enable seamless remote execution, a low-latency middle tier, cloud infrastructure optimization for mobile applications, basic mobile cloud services such as presence services, memcache services etc. In this paper, we envision that these fundamental new capabilities will enable mobile users to seamlessly utilize the cloud to obtain the resource benefits without incurring delays and jitter and without worrying about energy. By thus empowering mobile users, mobile computing will be able to break free of the fundamental constraints that have been keeping us from transform many areas of human activity. We envision the future of mobile computing applications will be built on top of a rich eco-system of basic mobile cloud services. We are pursuing many of the research topics outlined in this paper.

8. REFERENCES

[1] R. Alimi et al. Open content distribution using data lockers. Technical Report TR1426, Yale, Feb. 2010.

[2] Amazon AWS. Mobile developer center. `http://aws.amazon.com/mobile`.

[3] M. Armbrust et al. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, UC Berkeley, 2009.

[4] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, M. Terada, and R. Han. Whozthat? evolving an ecosystem for context-aware mobile social networks. *IEEE Network*, 22(4):50–55, July-August 2008.

[5] A. Beach, M. Gartrell, and R. Han. q-Anon: Rethinking anonymity for social networks. In *IEEE SocialCom Conference*, 2010.

[6] A. Beach, M. Gartrell, X. Xing, R. Han, Q. Lv, S. Mishra, and K. Seada. Fusing mobile, sensor, and social data to fully enable context-aware computing. In *HotMobile '10: Proceedings of the 11th Workshop on Mobile Computing, Systems, and Applications*, pages 61–66, 2010.

[7] S. Bykov, A. Geller, G. Kliot, J. R. Larus, R. Pandya, and J. Thelin. Orleans: cloud computing for everyone. In *SOCC '11: Proceedings of the 2nd ACM Symposium on Cloud Computing*, pages 16:1–16:14, 2011.

[8] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. Clonecloud: elastic execution between mobile device and cloud. In *EuroSys '11: Proceedings of the sixth conference on Computer systems*, pages 301–314, 2011.

[9] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. MAUI: making smartphones last longer with code offload. In *MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 49–62, 2010.

[10] Gartner, Inc. Gartner says worldwide cloud services market to surpass $68 billion in 2010. `http://www.gartner.com/it/page.jsp?id=1389313`, 2010.

[11] P. Gilbert, L. P. Cox, J. Jung, and D. Wetherall. Toward trustworthy mobile sensing. In *HotMobile '10: Proceedings of the 11th Workshop on Mobile Computing Systems & Applications*, pages 31–36, 2010.

[12] G. Huerta-Canepa and D. Lee. A virtual cloud computing provider for mobile devices. In *MCS '10: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services*, pages 1–5, 2010.

[13] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe. Parknet: drive-by sensing of road-side parking statistics. In *MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 123–136, 2010.

[14] Microsoft Research Project Hawaii. Hardware and software platforms for developing cloud-enabled applications for Windows Phone 7. `http://research.microsoft.com/en-us/um/redmond/projects/hawaii/students/`.

[15] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *SenSys '08: Proceedings of the 6th ACM Conf. on Embedded Network Sensor Systems*, 2008.

[16] MIT technology review. Building a cloud out of smart phones. `http://www.technologyreview.com/blog/mimssbits/25609/`, 2010.

[17] NewVoice Social, LLC. Traffic talk: People-powered traffic information. `http://www.traffictalk.info/`.

[18] OpenCV. Open source computer vision (OpenCV) library. `http://opencv.willowgarage.com/wiki/`.

[19] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan. Odessa: enabling interactive perception applications on mobile devices. In *MobiSys '11: Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 43–56, 2011.

[20] S. Saroiu and A. Wolman. I am a sensor, and I approve this message. In *HotMobile '10: Proceedings of the 11th Workshop on Mobile Computing Systems & Applications*, pages 37–42, 2010.

[21] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for VM-based Cloudlets in mobile computing. *IEEE Pervasive Computing*, 8:14–23, 2009.

[22] J. Zimmerman, A. Tomasic, C. Garrod, D. Yoo, C. Hiruncharoenvate, R. Aziz, N. R. Thiruvengadam, Y. Huang, and A. Steinfeld. Field trial of tiramisu: crowd-sourcing bus arrival times to spur co-design. In *CHI '11: Proceedings of the 29th annual conference on Human factors in computing systems*, pages 1677–1686, 2011.

[23] Zynga, Inc. Building a scalable game server. `http://code.zynga.com/2011/07/building-a-scalable-game-server/`, 2012.

[24] Zynga, Inc. The evolution of zcloud. `http://code.zynga.com/2012/02/the-evolution-of-zcloud/`, 2012.