Augmenting Pervasive Environments with an XMPP-Based Mobile Cloud Middleware

Dejan Kovachev, Yiwei Cao, and Ralf Klamma

Information Systems & Database Technologies, RWTH Aachen University
Ahornstr. 55, 52056 Aachen, Germany
{kovachev,cao,klamma}@dbis.rwth-aachen.de
http://dbis.rwth-aachen.de

Abstract. Despite the rapid advances in mobile technology, many constraints still prohibit smartphones to run resource-demanding applications in pervasive environments. Emerging cloud computing opens an access to unlimited resources for mobile devices. However, the combination of both technologies to deliver sound mobile cloud applications and services raises new challenges and requirements. Based on a scenario-based requirement analysis and a comprehensive study on existing work for augmenting mobile devices, we propose a XMPP-based mobile cloud computing architecture employing module partitioning and adaptive offloading to nearby computing infrastructure. Research has also been done in the underlying offloading mechanism based on context-aware cost model. Further problems related to this approach are discussed as well, including selection of most optimal offloading plan, application partitioning and issues with XMPP on mobile systems.

Keywords: mobile cloud computing, dynamic software architecture, pervasive computing, augmented execution, XMPP.

1 Introduction

In recent years, smart mobile devices such as iPhones, Android-based smartphones, BlackBerrys, has become highly accessible. They have rich sensing functionality and data exchange connectivity. The applications running on these smart mobile devices seamlessly integrate with realtime data streams and Web 2.0 applications, such as mashups, open collaboration, social networking and mobile commerce [23,14]. Despite the significant improvements of the mobile hardware technologies, mobile devices will continue to be resource-poor, with unstable connectivity, and with less energy. Users cannot expect that the mobile execution platform supports the same functionality as their desktop's or notebook's counterpart. Resource poverty is a major obstacle for many mobile applications [19]. Therefore, computation capacities on mobile devices will compromise the applications.

On the other hand, cloud computing technologies have been emerging recently as a solution to scalable on-demand computing and storage resources that can

M. Griss and G. Yang (Eds.): MOBICASE 2010, LNICST 76, pp. 361–372, 2012.

be accessed via networks. The conceptual model of cloud computing provides a platform for services and applications with high elasticity. Elasticity means that storage and computation resources are put into use according to actual current requirements. These properties would benefit certainly resource-poor mobile devices. The never ending strife for increasing mobile processing power and more data, clouds can be the best possible solution to augment the mobile execution platform.

However, mobile pervasive computing environments are characterized by severe resource constraints and frequent changes in operating conditions, i.e. unstable connectivity, less energy, limited memory and processing power. The existing cloud computing methods available for desktop clients cannot be applied directly. For example, the computation of current mobile applications happens either mostly on the cloud side (e.g. Twitter) or on the device (e.g. a video game). Such static partitioning of mobile applications does not provide optimal user experience in different scenarios, since mobile devices come with different capabilities. Furthermore, accessing distant clouds introduces more latency, monetary costs and limited interactivity.

In this paper, we envision how to deploy mobile applications that are dynamically partitioned between limited mobile devices and the cloud with "unlimited" resources. Furthermore, to avoid the latency of accessing distant cloud centers and improve the user experience for resource-demanding applications, we consider the idea of cloudlets [19], a computing platform exposing its functionality via wireless access to nearby mobile devices. Such a possibility opens the doorways to more powerful interactive mobile applications (see Fig. 1). Moreover, the partitioned modularized applications are deployed on top of an XMPP-based middleware which enables real-time, flexible, scalable and extensible software architecture. XMPP acts as glue joining the heterogeneous parts of the system, providing bi-directional XML streaming with rich publish/subscribe functionality and presence detection.

In the rest of the paper, we elaborate more on the mobile cloud computing augmentation for pervasive environments. Section 2 surveys briefly the related work. Section 3 motivates this approach with scenarios, from which a set of requirements is derived and explained how these requirements can be met with cloud-related methods. In the Section 4, we propose a modular architecture and describe the enabling technology. Furthermore, we underline the importance of a context-ware cost model for dynamic mobile cloud applications. Finally, we discuss some open issues in Section 5 and conclude the paper in Section 6.

2 Related Work

Over the last two decades, there have been several approaches developed on how to augment capabilities of resource-constrained mobile devices. These approaches include software replication, application partitioning and modularization, process and virtual machine (VM) migration. The previous efforts investigate how to increase the processing power, how to increase data storage or memory [4,21],

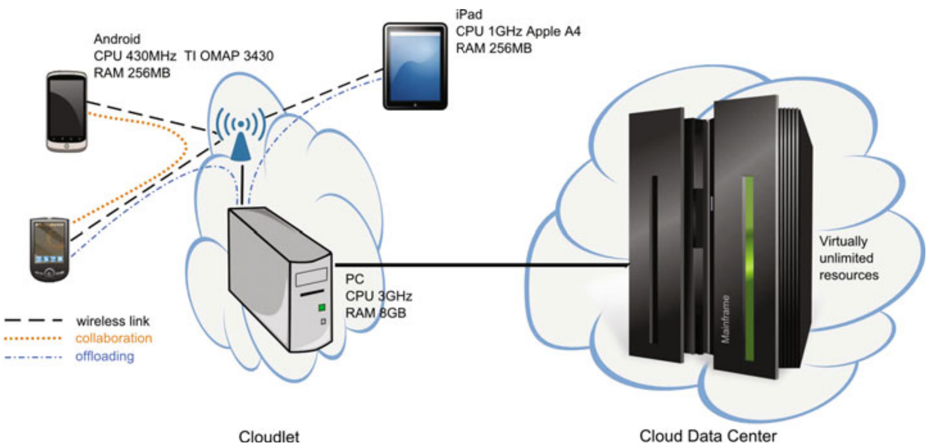

Fig. 1. Mobile devices accessing nearby connected cloudlet and large cloud data centers via Internet

how to balance the performance or QoS with energy consumption, how to ensure application mobility [5], how to secure [7,15] or personalize [20] nearby untrusted computing environments. Our approach draws the best features of previous work, trying to overcome some of their limitations.

The simplest way to augment weak devices such as mobile phones is the application delivery based on the traditional client-server model [12]. However, the client-server model does not consider the changing conditions in pervasive computing environments, causing limited interactivity (thin clients, Web applications) or less portability (fat clients, native mobile applications) [24]. In Spectra [6], programmers define execution plans that run several application partitioning variants which deliver different quality of service. Although Coign [11] is a nice example of automatic partitioning of DCOM applications without source code modification, it outputs client-server applications again statically.

Gu et al. [10] demonstrate an adaptive offloading of certain code classes based on the available resources, class memory footprint and class access frequency. The MAUI system achieves the similar fine grained runtime code offloading to the cloud infrastructure [3]. The primary goal of MAUI is to maximize device battery life with code offload. Developers annotate while programming which methods can be offloaded for remote execution. Since MAUI allows a fine grained offloading mechanism on the level of single methods, the experimental results from MAUI show that the separate method offloading can be contraproductive, i.e. several methods should be combined to achieve benefits. Rellermeyer et al. [16] consider the offloading on complete software modules. Similarly, Zhang et al. [26,25] develop a reference framework for partitioning a single application into elastic components with dynamic configuration of execution. The components, called *weblets*, are platform independent and can be executed transparently on different computing infrastructures. Our approach partitions applications on module level where developers express in the code the mobility of each individual module and its relations with other modules.

Besides code partitioning, the augmented execution can be further achieved with a software replication. Chun and Maniatis [2] propose an architecture that replicates the whole smartphone images and run the application code with few or no modifications in powerful VM replicas. Such architecture, e.g., enables smooth execution of computation and memory-hungry applications, such as virus scanning or file system indexing, without blocking other applications. A similar approach to using virtual machine (VM) technologies executing the computation-intensive software from mobile devices is presented by Satyanarayanan et al. [19]. In this architecture, a mobile user exploits VMs to rapidly instantiate customized service software on a nearby cloudlet and uses the service over WLAN. Rather relying on a distant cloud, the cloudlets eliminate the long latency introduced by wide-area networks for accessing the cloud resources. As a result, the responsiveness and interactivity on the device are enhanced by low-latency, one-hop, high-bandwidth wireless access to the cloudlet. A mobile device delivers a small VMs overlay to the cloudlet infrastructure that already owns the base VM. This approach enables the user to resume a complete personal computing environment including own choices of operating system, applications, settings, and data. We also consider offloading to a nearby computing infrastructure in order to reduce the latency.

3 Scenarios and Requirements

For the purpose of test and evaluation, use case scenarios are designed to illustrate the special usage of the proposed mobile cloud computing framework. The scenarios we consider require more resources than the conventional mobile devices ca provide. We address the issues that may occur related to the scenarios, which helps an in-depth requirement analysis of the framework design.

Scenario 1: Architectural modeling is important in architectural design to see whether the design draft fits to the physical environment. In 1980s, the architect had to build a glass pyramid of the same size in front of the Louvre in order to persuade the Paris citizens to accept his design of the entrance to the national museum. It is almost impossible for the most architects, especially architectural students, to make such a costly experiment. The design models made of plastic foams with a different scale still cannot exactly tell whether a building fits into the real environment. Currently, advanced 3D scanner, 3D printer as well as 3D modeling technologies are able to deploy an augmented model in the physical environment [1]. With advanced cloud services, localization and projection of the model onto the real environment could be realized.

Scenario 2: Virtual archeology deals with those tasks such as 3D reconstruction and context-aware augmented reality [22,9]. These technologies provide researchers appropriate means to explain destroyed structures better or show the original sight. In many cases, research or heritage conservation work is carried out on site at a historical location. The widespread mobile devices help archaeologists during their on-site research. However they are usually expected to access large amounts of heterogeneous multimedia data and run resource-demanding applications, which mobile devices cannot fully deliver. Archaeologists also need to exchange messages and share data collaboratively with the team members.

Based on these and similar use case scenarios, it is obvious that mobile devices like smart phones with an embedded camera and different sensors have great potentials to assist researchers at field work. For example, researchers should able to consume 3D objects from remote data repositories and see virtual objects blended on the video stream from their mobile phone camera. They would be able to see in the ruins virtual historic objects which have been moved to some museums faraway for protection from the severe outdoor environments.

In the reality, problems and challenges are raised from many aspects. A heavy equipment with enough powerful computers lowers the mobility and flexibility of on-site researchers. A large storage capacity for prepared and on-site acquired multimedia data is beyond a mobile device. The advanced 3D technologies cannot be easily used on site, because they require more processing and storage capacity than modern mobile device can provide. Light mobile devices are not able to process that raw data immediately due. A trade-off between reduction of technical infrastructure and provision of high computing capabilities should be made to fulfill the required tasks. Flexibility and mobility feature the applications that need to realize the tasks described in the scenarios.

The tasks in the both scenarios can be well divided into several sub tasks. They address requirements onto the functionality and features of a mobile cloud computing platform as listed in Figure 2.

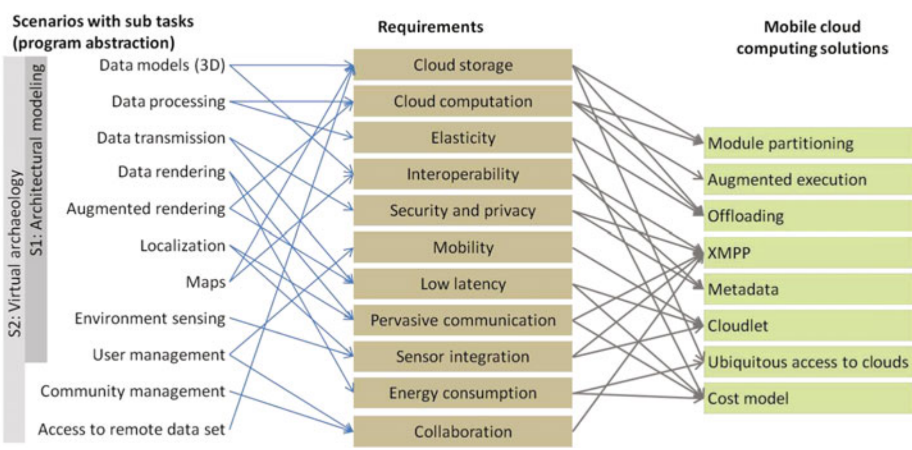

Fig. 2. Scenarios, requirement analysis and the mobile cloud computing solutions

The mobile cloud computing platform should provide a set of solutions to support the communication, collaboration and offloading. The cultural heritage and architecture domains address the scenarios which require an appropriate mobile cloud computing framework.

4 The XMPP-Based Mobile Cloud Middleware

The scenario-based requirement analysis shows that a XMPP based mobile cloud computing solution is promising to meet some requirements simultaneously. A cost model with clear definition and analysis of goals and constraints is the efficient approach to mobile cloud computing. We propose the mobile cloud computing architecture with two crucial aspects: the standardization of XMPP as cloudlet protocol and a context-aware cost model.

4.1 Context-Aware Cost Model

In order to dynamically shift the computation between mobile devices and cloud, applications need to be split in loosely-coupled modules interacting with each other. The modules are dynamically instantiated on and moved between mobile devices and cloud. It depends on several metric parameters modeled in the cost model as depicted in Figure 3. These parameters can include the module execution time, resource consumption, battery level, monetary costs, security, or network bandwidth. The main objective of the cost model is to help make intelligent offloading decisions that fulfill the given goals under certain device constraints based on several input parameters with the minimal overhead.

Several approaches to modeling the parameters have been proposed. Giurgiu et al. [8] model the application modules as resource consumption graph, and try to find a cut in the graph. Every bundle or module composing the application has certain memory consumption, generated input and output traffic, and the code size. The partitioning problem is to find an optimal cut in the graph under device constraints. Meanwhile, end-to-end interaction between phones and servers should be minimized. Gu et al. [10] model the relation between Java classes and use fuzzy logic for decision-making. Zhang et al. [25] use Naïve Bayesian Learning classifiers to find the optimal execution configuration from all possible configurations.

4.2 XMPP: The Glue for Mobile Cloud Services

An essential enabling technology in our approach is the Extensible Messaging and Presence Protocol (XMPP) [18]. The XMPP protocol provides a pure XML foundation for real-time messaging, opening up tremendous possibilities for more advanced real-time applications. XMPP together with its extensions is a powerful protocol for cloud services that demonstrate several advantages beyond traditional HTTP-based Web services (e.g. SOAP and REST):

- decentralized, open and flexible (extensible) communication protocol
- services being discoverable without the need of an external registry
- federation of services enabling easy weaving of cloud services together
- built-in presence functionality providing resource and availability discovery at runtime. XMPP not only allows discovering of services out of the box, but also supports determining their status and availability.

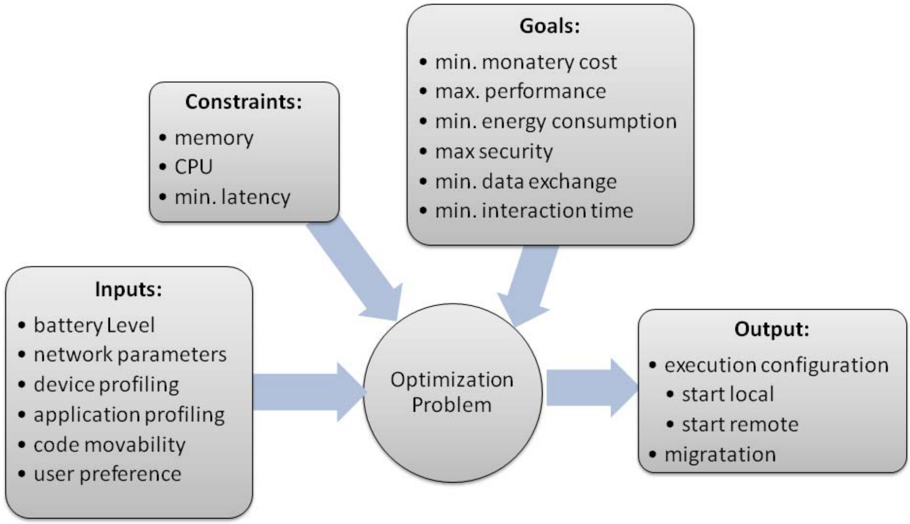

Fig. 3. The cost model with parameters for elastic mobile pervasive cloud applications

– support for real-time data streaming in two directions. Asynchronous invocation eliminates the need for ad-hoc solutions like polling
– interoperability with other protocols and programming language independent
– event notifications
– remote procedure calls (e.g. SOAP over XMPP)
– multimedia session management

Many advantages over existing technologies make XMPP a highly interesting candidate for next generation online services. HTTP was originally designed to accommodate query and retrieval of web pages and does not aim to rather complex communication. The intrinsically synchronous HTTP protocol is unsuitable for time-consuming operations, like computation-demanding database lookups or video processing.

An XMPP network can be seen as a complete XML-based routing framework upon which a messaging middleware can be built. Hence, an XMPP-based middleware can be used to integrate different services into a distributed computing environment. The application modules, external sensors and external services are XMPP entities identified by unique JIDs. They exchange messages through XML stanzas. However, a suitable mechanism to define the interconnections between different entities is needed.

4.3 Modular Architecture of the XMPP-Based Mobile Cloud Middleware

To enable the new mobile cloud application model, the middleware should provide an infrastructure for seamless and transparent execution of elastic applications and offer convenient development support. Furthermore, future mobile applications may utilize and synthesize capabilities from multiple clouds.

We adopt a concept similar to the cloudlets [19] and Slingshot [6] where the computation is offloaded to a nearby more powerful computer called surrogate. A cloudlet is a trusted, resource-rich computer or a cluster of computers that is well connected to the Internet and is available to nearby mobile devices. For example, we employ an access point (AP) with enough resources to run software modules from connected mobile devices. In this way, the latency amongst can be drastically reduced [19]. In contrast to [19] and [6], our approach does not migrate complete virtual machines on the AP. Using VMs for each mobile user connected to the cloudlet will make the cloudlet useless. Therefore, applications coming form different mobile users should share the same cloudlet platform for executing their software modules.

Our architecture relies on the possibility to split applications on loosely-coupled modules that encapsulate a unit of functionality. The developers split the application into loosely-coupled modules. Furthermore, they annotate the modules that can be offloaded and executed remotely. The software development tools determine the dependency between each module which helps further while deciding about module remote execution. The module functionality is exposed through service interface, so modules can easily be instantiated on the mobile device, on the nearby cloudlet, or on a distant cloud infrastructure. The offloading control makes decisions based on a cost model for the optimal execution.

Figure 4 depicts the main architecture of our approach. The XMPP-based mobile cloud middleware runs on mobile devices and cloudlet tier, and connects to other traditional cloud services. The execution manager/validator is responsible to instantiate, migrate and stop the application modules, based on the decisions from the offloading control component. The offloading control component implements the cost model and gives offloading decision for the movable modules based on the sensed context input parameters and designated performance goals . The messaging manager is responsible for synchronizing the different parts of the application. The application itself is split in loosely-coupled modules which can be executed locally on the mobile device or on a nearby cloudlet that hosts the pervasive cloud middleware. There exist some parts of the application which cannot be offloaded such as UI or accessing device hardware functionality. These parts constitute the application core running on the mobile device. The application metadata describes the dependencies and interconnection between modules, and their functionality. The module offloading proxy bridges the module invocations on separate tiers, so that the applications run as one logical entity.

To use XMPP as a communication protocol has several benefits. First, the XMPP publish/subscribe functionality enables easy realtime integration of heterogeneous entities, e.g. modules, services, and sensor streams, which greatly

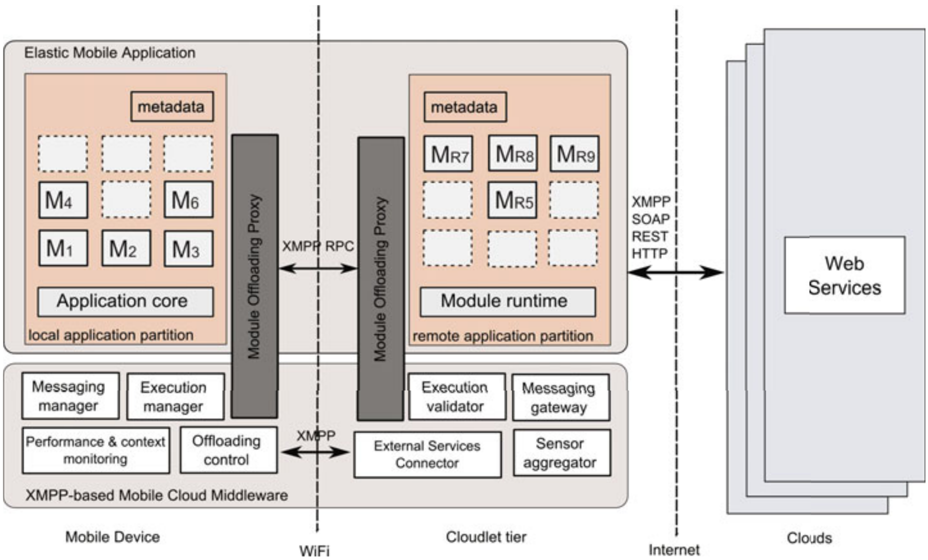

Fig. 4. Modular architecture of the XMPP-based Mobile Cloud Middleware

extends the possibilities of the mobile pervasive cloud applications. This is important in our case since the modules are dynamically started or stopped on different tiers. The XMPP presence awareness simplifies greatly the discovery of new resources and their availability. Second, XMPP enables better interoperability since XMPP uses XML streams for communication. The XML-based communication makes the protocol easily extendable, e.g., in multimedia session management where the signaling is handled by XML messages and media data is sent via other protocols such as Real-Time Transport Protocol (RTP) [17]. Third, XMPP inherently supports bi-directional communication, providing means for collaborative applications. Fourth, the federation model of XMPP enables interconnection with other servers, making the cloudlet a gateway to external cloud services.

5 Discussion

Since the execution of application modules shifts dynamically between mobile devices and cloud, the validity of execution results needs to be checked. The underlying middleware needs to transfer the state of migrating modules and related data. Moreover, requirements for real-time execution integration depend on the application itself. For example, in the previous use scenarios the mobile application would require fast updates from the rendering modules that run on the cloudlet tier. In contrast, an collaborative data sharing between mobile users on site would require more modest communication rate.

Due to the nature of pervasive environments, selection of the most optimal execution plan cannot be done deterministically. Continuous context sensing followed by module, application and network monitoring can help making the decision. It is resource-consuming for mobile devices to solve the optimization problem within itself. Therefore, lightweight, fast and predictive decision algorithms should be executed on the device. Alternatively, the optimization execution itself should be offloaded into the cloudlets.

XMPP is verbose by design, which may influence mobile communications negatively. Although verbosity makes the protocol open and easy for debugging and learning, it introduces also additional processing and communication overhead to the weak mobile devices. That means additional battery and bandwidth consumption for mobile devices. Using the stream compression extension for XMPP can reduce the size of messages.

Our solution is mostly applicable to smart mobile devices with Internet connectivity, with handheld form and a large, high quality graphics display, and significant but limited computing power. We choose these platforms due to their pervasive availability and large sensing capabilities. Their hardware improves constantly, but will still be resource-poor. Therefore, the mobile cloud middleware takes advantage of nearby computers, i.e. cloudlet, serving as a computation and collaboration platform with lower latency. However, the middleware can be extended easily to operate on the other cloud infrastructures. Furthermore, we give developers the control to split the application instead of automatic partitioning as introduced in [11].

6 Conclusions

Mobile cloud computing can augment the limited capabilities of mobile devices via access to elastic cloud computing resources. Based on two scenarios in the domains of cultural heritage and architecture, requirements on mobile cloud computing are derived. Challenges and problems raise related to large-scale data sets, limitations of mobile devices, required nearby equipment, and changing requirements of pervasive environments etc. Starting from the requirements, we propose a mobile cloud computing architecture to provide solution in the domains. Mobile devices are regulated to offload resource-demanding applications onto cloudlets in the pervasive environment aiming to lower the latency, and thus improve user experience. Access to remote cloud services are also supported. The XMPP protocol shows a promising component to meet the platform requirements. Moreover, a cost model is specified to model the factors that influence the performance and cost of this mobile cloud computing model. This cost model is still given at abstract level. Further plans of setting up this mobile cloud computing platform will be carried out with our on-going research on the underlying cost model. Additionally, our goal also is to enbable interoperability between the cloudlet and our existing cloud infrastructure (cf. [13]).

Acknowledgment. This work was supported in part by NRW State within the B-IT Research School and the Excellence Initiative of German National

Science Foundation (DFG) within the research cluster UMIC. We thank our colleague Dominik Renzel, and Georgios Toubekis (RWTH Aachen Center for Documentation and Conservation) for fruitful discussions with them concerning technical issues, and scenarios in the domains of architecture and archeology.

References

1. Broll, W., Lindt, I., Ohlenburg, J., Wittkämper, M., Yuan, C., Novotny, T., Gen Schieck, A.F., Mottram, C., Strothmann, A.: Arthur: A Collaborative Augmented Environment for Architectural Design and Urban Planning. Journal of Virtual Reality and Broadcasting 1(1) (December 2004); urn:nbn:de:0009-6-348
2. Chun, B.-G., Maniatis, P.: Augmented Smartphone Applications Through Clone Cloud Execution. In: Proceedings of the 12th Workshop on Hot Topics in Operating Systems (HotOS XII), Monte Verita, Switzerland. USENIX (2009)
3. Cuervo, E., Balasubramanian, A., Cho, D.-K., Wolman, A., Saroiu, S., Chandra, R., Bahl, P.: MAUI: Making Smartphones Last Longer with Code Offload. In: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (ACM MobiSys 2010), San Francisco, CA, USA, pp. 49–62. ACM (2010)
4. Das, S., Agrawal, D., Abbadi, A.E.: ElasTraS: An Elastic Transactional Data Store in the Cloud. In: Proceedings of the 1st USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2009), San Diego, CA, USA. USENIX Association (2009)
5. David, F.M., Donkervoet, B., Carlyle, J.C., Chan, E.M., Pasquale, F.: Supporting Adaptive Application Mobility. In: Meersman, R., Tari, Z. (eds.) OTM-WS 2007, Part II. LNCS, vol. 4806, pp. 896–905. Springer, Heidelberg (2007)
6. Flinn, J., Narayanan, D., Satyanarayanan, M.: Self-tuned remote execution for pervasive computing. In: Proceedings Eighth Workshop on Hot Topics in Operating Systems (HotOS), Schloss Elamu, Germany, pp. 61–66. IEEE (2001)
7. Garriss, S., Cáceres, R., Berger, S., Sailer, R., van Doorn, L., Zhang, X.: Trustworthy and Personalized Computing on Public Kiosks. In: Proceeding of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys 2008), Breckenridge, CO, USA, pp. 199–210. ACM (2008)
8. Giurgiu, I., Riva, O., Juric, D., Krivulev, I., Alonso, G.: Calling the Cloud: Enabling Mobile Phones as Interfaces to Cloud Applications. In: Bacon, J.M., Cooper, B.F. (eds.) Middleware 2009. LNCS, vol. 5896, pp. 83–102. Springer, Heidelberg (2009)
9. Gleue, T., Dähne, P.: Design and implementation of a mobile device for outdoor augmented reality in the archeoguide project. In: VAST 2001: Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage, pp. 161–168. ACM, New York (2001)
10. Gu, X., Nahrstedt, K., Messer, A., Greenberg, I., Milojicic, D.: Adaptive Offloading Inference for Delivering Applications in Pervasive Computing Environments. In: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), Dallas-Fort Worth, TX, USA, pp. 107–114. IEEE (2003)
11. Hunt, G.C., Scott, M.L.: The Coign Automatic Distributed Partitioning System. In: Proceeedings of the Third Symposium on Operating System Design and Implementation (OSDI 1999), New Orleans, LA, USA, pp. 187–200. USENIX Association (1999)

12. Jing, J., Helal, A.S., Elmagarmid, A.: Client-server Computing in Mobile Environments. ACM Computing Surveys (CSUR) 31(2), 117–157 (1999)
13. Kovachev, D., Klamma, R.: A Cloud Multimedia Platform. In: Proceedings of the 11th International Workshop of the Multimedia Metadata Community on Interoperable Social Multimedia Applications (WISMA 2010), Barcelona, Spain, pp. 61–64. CEUR (2010)
14. Kovachev, D., Renzel, D., Klamma, R., Cao, Y.: Mobile Community Cloud Computing: Emerges and Evolves. In: Proceedings of the First International Workshop on Mobile Cloud Computing (MCC), Kansas City, MO, USA. IEEE (2010)
15. Oberheide, J., Veeraraghavan, K., Cooke, E., Flinn, J., Jahanian, F.: Virtualized In-Cloud Security Services for Mobile Devices Categories and Subject Descriptors. In: Proceedings of the First Workshop on Virtualization in Mobile Computing, Breckenridge, CO, USA, pp. 31–35. ACM (2008)
16. Rellermeyer, J.S., Duller, M., Alonso, G.: Engineering the Cloud from Software Modules. In: Proceedings of the Workshop on Software Engineering Challenges in Cloud Computing (ICSE-Cloud, in Conjunction with ICSE 2009), Vancouver, Canada, pp. 32–37. IEEE (2009)
17. Saint-Andre, P.: Jingle: Jabber Does Multimedia. Multimedia 14(1), 90–94 (2007)
18. Saint-Andre, P., Smith, K., Tronçon, R.: XMPP: The Definitive Guide. O'Reilly (2009)
19. Satyanarayanan, M., Bahl, P., Cáceres, R., Davies, N.: The Case for VM-Based Cloudlets in Mobile Computing. IEEE Pervasive Computing 8(4), 14–23 (2009)
20. Satyanarayanan, M., Kozuch, M.A., Helfrich, C.J., Hallaron, D.R.O.: Towards Seamless Mobility on Pervasive Hardware. Pervasive and Mobile Computing 1(2), 157–189 (2005)
21. Tolvanen, J., Suihko, T., Lipasti, J., Asokan, N.: Remote Storage for Mobile Devices. In: Proceedings of the 1st International Conference on Communication Systems Software & Middleware, New Delhi, India, pp. 1–9. IEEE (2006)
22. Vlahakis, V., Ioannidis, M., Karigiannis, J., Tsotros, M., Gounaris, M., Stricker, D., Gleue, T., Daehne, P., Almeida, L.: Archeoguide: an augmented reality guide for archaeological sites. IEEE Computer Graphics and Applications 22(5), 52–60 (2002)
23. Wright, A.: Get Smart. Communications of the ACM 52(1), 15–16 (2009)
24. Wu, H., Hamdi, L., Mahe, N.: TANGO: A Flexible Mobility-Enabled Architecture for Online and Offline Mobile Enterprise Applications. In: Proceedings of 11th International Conference on Mobile Data Mangement (MDM 2010), Kanas City, MO, USA, pp. 230–238. IEEE (2010)
25. Zhang, X., Jeong, S., Kunjithapatham, A., Gibbs, S.: Towards an Elastic Application Model for Augmenting Computing Capabilities of Mobile Platforms. In: The Third International ICST Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications, Chicago, IL, USA (2010)
26. Zhang, X., Schiffman, J., Gibbs, S., Kunjithapatham, A., Jeong, S.: Securing Elastic Applications on Mobile Devices for Cloud Computing. In: CCSW 2009: Proceedings of the 2009 ACM Workshop on Cloud Computing Security, Chicago, IL, USA, pp. 127–134. ACM (November 2009)