Modeling Mobile Cloud Computing Using Greenmetrics

Jong Hoon Ahnn
Department of Computer Science, UCLA
Cloud Research Lab, Samsung Information Systems America
jhahnn@cs.ucla.edu

Miodrag Potkonjak
Department of Computer Science, UCLA
miodrag@cs.ucla.edu

Abstract

Although a mobile cloud computing paradigm has obtained significant attentions from research community, we note that most of work is based on an ad hoc fashion. Furthermore, little work has shown a model-based cost optimization of offloading. Mobile cloud computing may instead be holistically analyzed and systematically designed as distributed solutions to some global optimization problems. Such a paradigm enables optimized code offloading of mobile applications, where mobile devices can be thought of a waypoint of powerful cloud resources. This paper tackles a suite of optimization subproblems: a program partitioning, network resource allocation, network selection, and cloud resource allocation problem. The key objective is to satisfy the mobile application's quality of service requirements by quantifying the performance of each subsystem: mobile clients, wireless network medium, and cloud services. By extensive experiments, we present mobile clients can have up to 73.69x and 39.69x offloading benefit in terms of time and energy.

1 Introduction

Mobile cloud computing is a model for transparent elastic augmentation of mobile device capabilities via ubiquitous wireless access to cloud storage and computing resources, with context-aware dynamic adjusting of offloading in respect to change in operating conditions, while preserving available sensing and computing capabilities of mobile devices [1]. Mobile devices can be seen as entry points and interfaces of cloud services. The combination of cloud computing, wireless communication infrastructure, portable computing devices, location-based services, and mobile Web has paved the foundation for a novel computing model.

For instance, Chang et al. studied a cloud-assisted speech recognition application and showed experimental results in their recent work [5] (See Figure 1). The Android-based voice dialer allows the user to dial the phone number to a friend by speaking the friend's name to the smartphone. The application recognizes the speech from the user first, searches the phone book for the corresponded phone number, and dial the phone number. An automatic speech recognition (ASR) engine presented in [5] is used as the backend for processing the voice signal and translate it to text.

Although many researcher have studied an energy-saving issue in mobile cloud computing, little work has shown a model-based cost optimization of offloading with the consideration of energy saving of mobile devices and cloud services at the same time. The overall system model may include several subsystems: mobile terminals, multiple wireless network interfaces, and cloud services. Recent research efforts have proposed several system architectures and its communication mechanism between a mobile and cloud side. However, without having concrete models in the performance of application and wireless communication medium, it is difficult to quantify the cost of code offload due to dynamic nature of mobile cloud applications. Therefore, we model the computation cost statistically while we model the communication cost theoretically. The reasoning behind is that once an application is profiled on a specific mobile device and cloud machine, the computing cost stays static unless network conditions change. For example, mobile network traffic is highly bursty in many cases. Therein, obtaining real-time network parameters can be costly due to the heavy scanning cost, and this situation is against our goal to save energy. In order to profile network conditions in an energy-efficient manner, we adopt an analytic cost of communication by compromising little bit of accuracy. We believe combining empirical and analytical profiling costs can enhance the overall

This work is funded in part by Samsung global research outreach program 2011-2013, award number 20112465.

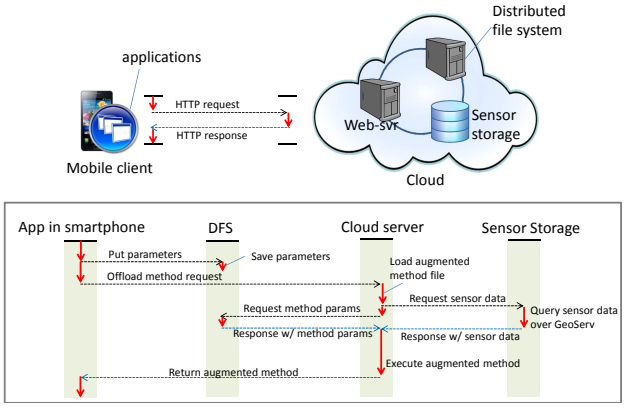


Figure 1: An automatic speech recognition application in our setting is presented. A solid line represents a control flow of a program while a dotted line represents networked message flow between mobile clients and cloud machines.

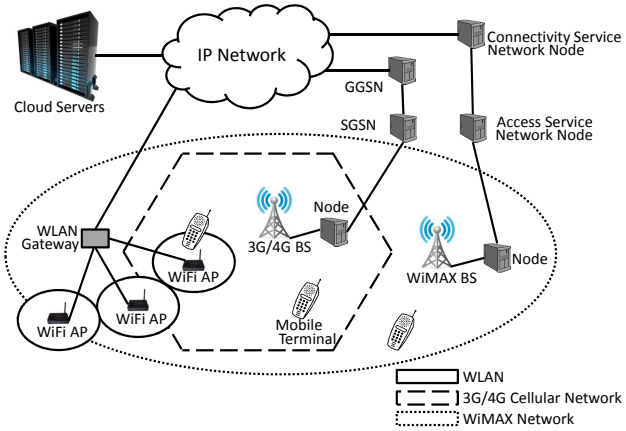


Figure 2: A high-level overview of mobile cloud architecture in heterogeneous wireless network interfaces.

system performance in providing real-time optimal offloading strategies for resource- and energy-constrained mobile clients.

The main contribution is three fold. We provide analytical cost models of computation and communication in mobile cloud computing. Upon the cost models, we decompose a complex mobile cloud system into subsystems (mobile clients, network, and cloud servers) to systematically provide distributed solutions to energy optimization problems. We finally propose a middleware-based system that enables optimized code offloading of mobile applications with the help of cloud services and integrate it with the energy optimizer.

Mobile Cloud Computing. The approaches from MAUI [7], CloneCloud [9] and Zhang et al. [16] seem promising because their model incorporates a cost model for deciding best execution configuration, and they can be also adapted dynamically according to real-time conditions. The approach in [17] is similar to above, but it lacks of dynamic adaptation of the computation between mobile devices and cloud services. Cloudlets [11] and ISR [15] allow high abstraction and personalization of the computing environment by using VMs, but lack from fine-grained execution adaptation. Prior work mostly focused on saving energy consumption on mobile devices; in contrast, our work provides analytical cost models to optimize the entire energy consumption including network and cloud at the same time.

2 Cost Models

We start this section by studying the current status of mobile applications as shown in Figure 2. Several researchers have identified the fundamental challenges in mobile computing due to severe resource constraints and dynamic changes in operating conditions. There are two types of mobile applications: offline and online applications. The computation cost acts as rich mobile client that perform all the computation locally, while the latter splits its computation into local and remote parts, thus may incur additional communication cost to transfer necessary its binary and necessary data, however save the total amount of local computation. We apply a regression theory in modeling computation of mobile applications based on empirical measurement data. To model heterogeneous air interfaces such as WLAN, cellular network, and WiMAX, we apply a state-of-the-art mathematical network model based on empirical system parameters [3].

2.1 Computation Model

We define a software program as a set of basic functional blocks (BFB)s, where a basic functional block corresponds to a single method or function in a program. Each BFB consists of a set of inputs as required knowledge of computation, and a set of outputs as an outcome of computation. These include both global and local variables defined in a program.

In order to model the performance of mobile applications in heterogeneous hardware environments, we apply a regression theory to derive statistical inference models, by taking a small number of samples, where each sample denotes the execution time of a BFB on a particular machine. In our regression model, a response is modeled as a weighted sum of predictor variables. By adopting statistical techniques, we then assess the effectiveness of model's predictive capability.

We suppose there are a subset of observations $\hat{\Theta}$ in a

large observation space Θ for which values of response and predictor variables are known. A observed response vector is denoted by $\mathbf{y} = [y_1, ..., y_i, ..., y_\theta]$, where y_i denotes it response variable for a single observation $i \in \hat{\Theta}$ and a Φ predictor vector is denoted by $x_i = [x_i^\phi, ..., x_i^\phi]$. The corresponding set of regression coefficients is expressed by a vector $\Gamma = [\gamma_0, ..., \gamma_\phi]$. Thus, a linear function of predictors Φ is given by,

$$f(y_i) = \Psi(x_i)\Gamma + \varepsilon_i = \gamma_0 + \sum_{j=1}^{\Phi} \Psi_j(x_i^j)\gamma_j + \varepsilon_i \quad (1)$$

γ_i can be seen as the expected change in y_i per unit change in the predictor variable x_i^j. An independent random error ε_i has mean $E(\varepsilon_i) = 0$ and constant variance $Var(\varepsilon_i) = \sigma^2$.

In order to determine the best fitting model, we consider least squared errors commonly used in minimizing $\Omega(\Gamma)$ the sum of squared deviations of predicted responses give by the model from observed responses.

$\Omega(\Gamma) = \sum_{i=1}^{\hat{\Theta}} (y_i - \gamma_0 - \sum_{j=1}^{\Phi} \gamma_j x_i^j)^2$ Obtaining estimates of the coefficients Γ is the goal of this approach. The correlation of response-to-predictor relationship is used in identifying the significance of the estimates. We express residuals to answer the problem of how well the model captures observed trends as, $\hat{\varepsilon} = y_i - \hat{\gamma}_0 - \sum_{j=0}^{\Phi} \hat{\gamma}_j x_i^j$ Model fitting can be assessed by the F-test [2] which is a standard statistical test method using multiple correlation statistic R^2 given by $R^2 = 1 - \frac{\sum_{i=1}^{\hat{\Theta}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{\hat{\Theta}} (y_i - \frac{1}{\hat{\Theta}} \sum_{i=1}^{\hat{\Theta}} y_i)^2}$ A larger R^2 value indicates better fits, while over-fitting if R^2 is close to 1. The over-fitting may occur when data sets are small and the number of predictors are large. A typical strategy is to set the number of predictors less than the number of observations given by $|\Phi| < \frac{\hat{\Theta}}{20}$ according to [2].

$$\hat{y}_i = E[\gamma_0 + \sum_{j=1}^{\Phi} \gamma_j x_i^j + \varepsilon_i] = \gamma_0 + \sum_{j=1}^{\Phi} \gamma_j x_i^j \quad (2)$$

The Equation 2 presents the expected value of y_i, $E[y_i]$ and its corresponding estimate $\hat{y}_i$ with $E[\varepsilon_i] = 0$. We herein define a coefficient of performance η of a computer such as a mobile client and cloud server. The coefficient converts the performance in time $\hat{y}_i$ into the one in power or energy $\hat{P}_i$ on a computer j as $\hat{P_i^j} = \eta \cdot \hat{y}_i^j$, where η can be experimentally obtained.

2.2 Communication Model

As mobile cloud is built on top of standard multi-RATs that have been heavily studied, we feel no need to model them in a different way. Rather, we focus on ways to decompose a complex mobile cloud system into analyzable sub-systems and to coordinate sub-systems. We here consider multiple wireless network interfaces scenario where heterogeneous radio access technologies (RAT)s such as WIFI, WiMAX, UMTS, and GSM work together with their overlapping network coverage in a given area. According to [3], RATs can be largely characterized into two categories based on means to share their channels: interference constrained RATs and orthogonal RATs. For the page limitation, we only consider the latter in this paper.

Orthogonal RATs are network interfaces where they assume a fixed transmission power over all base stations (BS)s. In this sort of systems, time and frequency slots are thought to be their bandwidth per a base station, and their resources are assigned to mobile devices by different resource allocation techniques under various objectives such as utility, fairness, and priority. An typical example of such system is GSM/EDGE technology which is based on TDMA. The signal to interference and noise ratio (SINR) between a mobile client m and BS b can be given by

$$\nu_{m,b} = \frac{q_{m,b}\bar{P}_m}{\omega_b + \pi_b} \quad (3)$$

The thermal noise is denoted by π_b, the transmission power of BS is denoted by $\bar{P}_b$, and the intercell interference is given by ω_b for each BS b. THus, the assigned bandwidth $r_{m,b}$ to a mobile client m in BS b does not depend upon the SINR $\nu_{m,b}$. Therefore, the link rate $D_{m,b}$ of the orthogonal-based system is given by,

$$D_{m,b} = \bar{D}_{m,b} r_{m,b} \quad (4)$$

In TDMA system, $\bar{D}_{m,b}$ denotes the link rate of a unit time while it denotes the rate of a unit frequency between a mobile client m and m in BS b. $\bar{D}_{m,b}$ can be rewritten as a function of SINR $\nu_{m,b}$ defined in Equation 3.

3 Optimization Problem Formulation

We mainly solve three different problems: a program partitioning problem, network resource allocation problem, and network selection problem. A solution to the partitioning problem gives an optimal set of code offloading decisions in terms of computation cost and communication cost, while a solution to the network resource allocation problem gives an optimal allocation strategy toward maximizing the utility of network systems. It is obvious that solving the latter problem provides a way to choosing the best communication cost in the former problem.



Figure 3: (left) mobile application (MA) on an Android OS in Samsung Galexy S device; (middle) MA running in an iOS emulator; and (right) MA running in a Windows Mobile emulator.

3.1 Program Partitioning Problem

Let us consider a mobile application A and its call function graph $G = (V, E)$, where each vertex $v \in V$ denotes a method in A. An invocation of method v from one another u thereby is denoted by an edge $e = (u, v)$. We annotate each vertex with the execution time T_v of the method v and each edge with the data transfer time $T_{u \to v}$ incurred when the method v is offloaded from the method u. We reconstruct a new graph $G' = (V', E')$ from G by adding corresponding offloading methods to V. The code partitioning problem based on G' can be formulated as,

$$\begin{aligned} min \quad & \textstyle\sum_{v' \in V'} T_{v'} + \sum_{e' \in E'} T_{e': u' \to v'}, \\ s.t. \quad & \frac{\sum_{v' \in V'} T_{v'} + \sum_{e' \in E'} T_{e'}}{\sum_{v \in V} T_v + \sum_{e \in E} T_e} \leq 1, \\ & T_{v'} \geq 0, T_v \geq 0, T_{e'} \geq 0, T_e \geq 0 \end{aligned} \tag{5}$$

The calculation of computation cost $T_v, T_{v'}$ depends upon the performance estimate $\hat{y}_i$ for each basic functional block (BFB) v, v' (see Equation 2). Furthermore, the calculation of communication cost incurred due to code offload is given by,

$$T_{v'} = n_{v'} \times D_{m,b}, \tag{6}$$

where the assigned data rate is denoted by $D_{m,b}$ for a mobile client m in BS b, and the size of data to be transferred due to offloading for BFB v' is given by $n_{v'}$. The data rate for orthogonal RATs is presented in Equation 4. We formulate further problems for how to assign the data rate to each mobile client in Section 3.3 and how to select one of the heterogeneous network interfaces in Section 3.2. The problem formulated in Equation 6 consists of a concave objective over linear constraints, and it becomes convex. Therefore, there are various convex optimization algorithms to solve it from [4].

3.2 Network Resource Allocation problem

A resource allocation problem in a single RAT has been well studied. Several approaches including convex optimization [12], stochastic optimization [13], and dynamic programming [14] have been applied to such resource allocation problems. In this work, we consider a utility maximization strategy which is similar to [3]. We consider a utility metric as the effectiveness of allocated resources of networked systems in our optimization problem as,

$$U = \sum_m \sum_b D_{m,b} \tag{7}$$

In order to deal with fairness in resource allocation among mobile clients, the utility function with a weight variable w can construct the α proportional fairness as,

$$U = \sum_m \frac{w_m}{1-\alpha} \sum_b D_{m,b}^{1-\alpha}, \tag{8}$$

where $0 \leq \alpha < 1$. Now, we present an optimization problem as,

$$\begin{aligned} max \quad & U, \\ s.t. \quad & \textstyle\sum_m \frac{D_{m,b}}{\overline{D}_{m,b}} \leq \Gamma_b, \\ & \textstyle\sum_b D_{m,b} \geq D_{min,b}, \\ & D_{m,b} \geq 0, \end{aligned} \tag{9}$$

where $D_{min,b}$ is the minimum data rate assigned to mobile clients. Our goal is for a network operator to maximize the sum of utility of all mobile users in all base stations. Note that Equation 9 consists of a concave objective over linear constraints and thus is convex. That means there exists various algorithms to solve the problem immediately [4].

3.3 Network Selection Problem

The network system model in this paper considers multiple wireless network interfaces with different radio access technologies (RAT)s such as WIFI, WiMAX, UMTS, and GSM having different capacity constraints and channel conditions. The problem we would construct is a decent network selection strategy that minimizes the expected mean cost of data transfer from a mobile client to a target offloading agent such as cloud machines. We develop a simple heuristic-based strategy $S(m, b, l) \in S$ where it takes into account current workload $l_m \in L$ for a mobile client $m \in M$ in a base station $b \in B$ which corresponds to the size of data to be transfered over the wireless network medium. The strategy $S(m, l)$ selects the best RAT which can support its workload l satisfying QoS requirements. We assume each

Data: mobile client $m \in M$, mobile application $a \in A$, BFB $i \in a$, BS $b \in B_r$, RAT $r \in R$ cloud server $j \in J$
Result: Optimal offloading graph on optimal resources assigned: $\hat{G}'$
while *There exist jobs to be scheduled* **do**
  U = solve **P2**(D);
  $D_{m,b}$ = solve **P3**(U, D);
  $(\hat{y}, P)$ = solve **P4**(A, J);
  $\hat{G}'$ = solve **P1**$(m, a, D_{m,b}, \hat{y})$;
end

Algorithm 1: An orchestrated approach to a suite of optimization algorithms to achieve a global optimization objective in a distributed manner.

mobile user belongs to one of K different user classes. With probability p_k, an arriving mobile user is characterized by a specific class k in the network system. Let $\bar{X}_k \in X$ denote a set of states loaded for class k specifying whether it allows the admission of a mobile user in class k to one of the RATs $r \in R$. Therefore, the strategy S(m,b,l,r) can be defined as,

$$S(m,b,l,r) := max \quad l_m \times D_{m,b,r}, \quad \forall l_m \in L \forall m \in M, \forall b \in B, \forall r \in R \tag{10}$$

If there are several $S(m,b,l,r)$ that maximizes the performance of data transfer cost, among them the strategy chooses the one which is equivalent to the RAT having minimum current total workload.

4 Algorithm

We present an algorithm used in mobile cloud computing operated by data centers. In the dynamic scenario, mobile clients or users request and their mobility are subject to a given mobility and traffic model rather than stochastic processes. Algorithm 1 solves P2, P3, P4, and P1 in order until the mobile cloud computing facility based on data centers ends. The network resource allocation problem P2 is for a network operator to maximize the sum of utility U of all mobile users in all base stations (Eq.9). The following network selection problem P3 that maximizes the performance of data transfer cost, resulting in an optimal choice $D_{m,b}$ of RAT among candidate RATs. The data center job allocation problem P4 minimizes the total amount of power consumption within a data center. It results in the performance estimate $\hat{y}$ for given computation to be offloaded by mobile applications, and its power consumption footprint P. Finally, the program partitioning problem collects computation cost and communication cost, and makes a decision of whether or not given computational blocks are outsourced based on optimal resources provided by P2, P3, and P4. The final offloading decision is denoted by a graph of BFBs $\hat{G}'$ (Eq.5) with a set of optimal resources assigned.

5 Results

We implement the proposed mobile cloud system, where each mobile device dynamically partitions its target application to offload computationally expensive components or methods to the cloud (See Figure 3). Recall that the ultimate goal of this partitioning is to maximize offloading benefits in mobile clients. By design, a cloud service can host any types of Web servers because our architecture supports multi- and cross-platforms. In our prototype we implemented offloaded methods as java servlets in an Apache Tomcat Web server. In this work, we show experimental results based on Samsung Galexy S (Android OS Gingerbread, single-core 1GHz Cortex-A8 CPU, and 512MB RAM). The dedicated cloud machine is configured by Intel core-2 quad 2.4GHz CPU, 64bits Windows Vista, and 4G memory. For simplicity, WLAN and AT&T 3G cellular network are only considered. Unless specified, the network selection follows a strategy given in Equation 10. The energy is measured by Monsoon power monitor.

We evaluate three mobile cloud applications: a chess game, puzzle game, and cryptographic algorithm DES. Each of application with different scenarios clearly shows offloading help improve significant performance in terms of execution time in Figure 4-6 and energy in Figure 7-9. L stands for local execution. P1-5 stands for parallel execution with the different number of concurrent parallel requests. M5-20 stands for scenarios where a different number of mobile clients concurrently request offloading to one cloud service. ROB presents relative offload benefits, comparing each offloading case with L. We present the proposed system with the help of 5 parallel execution (P5) can perform up to 73.69 times faster and 39.69 times energy-efficient compared to a standalone mobile application L.

6 Conclusion

We proposed a model-based energy saving technique for mobile cloud computing. For tested applications, we showed the proposed system can perform up to 73 times faster and 37 times more energy-efficient compared to a standalone mobile application.

References

[1] D. Kovachev, Y. Cao and R. Klamma. Mobile Cloud Computing: A Comparison of Application Models. In *CoRR*, 2011.

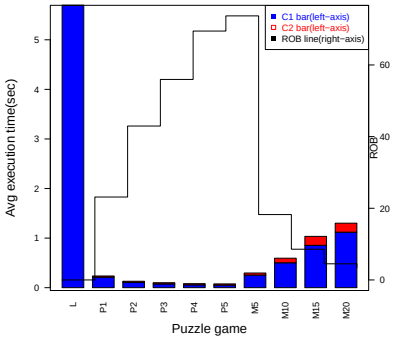


Figure 4: Execution time of puzzle game: max. 73.69x faster in P5.

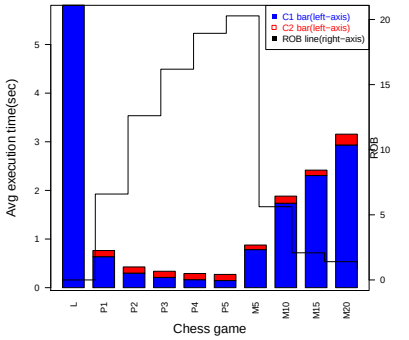


Figure 5: Execution time of chess game: max. 39.69x faster in P5.

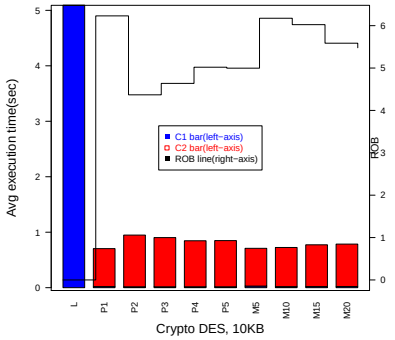


Figure 6: Execution time of DES crypto: max. 6.17x faster in P5.

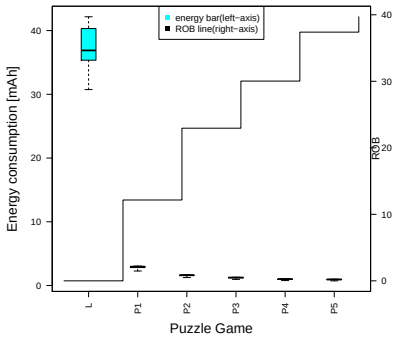


Figure 7: Energy consumption of puzzle game: max. 39.69x energy-efficient in P5.

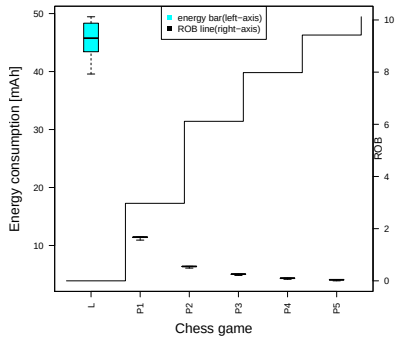


Figure 8: Energy consumption of chess game: max. 10.12x energy-efficient in P5.

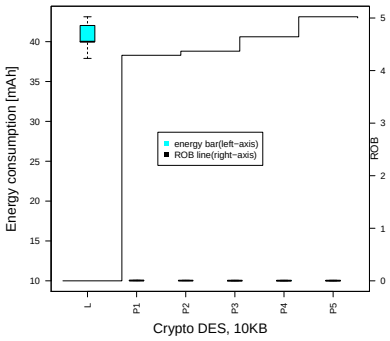


Figure 9: Energy consumption of DES crypto algorithm: max. 5.00x energy-efficient in P5.

[2] F. Harrell. Regression modeling strategies. In *Springer*, 2001.

[3] I. Blau, G. Wunder, I. Karla, and R. Sigle. Decentralized Utility Maximization in Heterogeneous Multicell Scenario with Interface Limited Orthogonal Air Interfaces. In *JWCN*, Jan. 2009.

[4] S. Boyd and L. Vandenberghe. Convex Optimization. *Cambridge University Press*, New York, NY, USA, 2004.

[5] Y-S Chang and S-H Hung. Developing Collaborative Applications with Mobile Cloud -A Case Study of Speech Recognition. In *JISIS*, 1(1):18-36, 2011.

[6] K.P. Birman. Reliable Distributed Systems: Technologies, Web Services and Applications. In *Springer Verlag*, 1997.

[7] E. Cuervoy, A. Balasubramanianz, D-K Cho, A. Wolmanx, S. Saroiux, R. Chandrax, P. Bahl. MAUI: Making Smartphones Last Longer with Code Offload. In *MobiSys*, 2010.

[8] Monsoon Solutions Inc. Monsoon Power Monitor. http://www.msoon.com/.

[9] B-G Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. CloneCloud: Elastic Execution between Mobile Device and Cloud. In *EuroSys*, 2011.

[10] X. Zhang, S. Jeong, A. Kunjithapatham, and Simon Gibbs. Towards an Elastic Application Model for Augmenting Computing Capabilities of Mobile Platforms. In *MOBILWARE*, 2010.

[11] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 8(4):14-23, 2009.

[12] D. P. Bertsekas. Nonlinear Programming. Athena Scientific. In *Belmont*, Massachusetts, 2nd. edition, 1995.

[13] H. Kushner and G. Yin. Stochastic approximation and recursive algorithms and applications. In *Springer*, 2nd edition, 2003.

[14] D. P. Bertsekas. Dynamic Programming and Optimal Control. In *Athena Scientific*, volume I, Massatchusetts, 1995.

[15] M. Satyanarayanan, M. A. Kozuch, C. J. Helfrich, and D. R. O. Hallaron. Towards Seamless Mobility on Pervasive Hardware. In *PMC*, 1(2):157-189, 2005.

[16] X. Zhang, S. Jeong, A. Kunjithapatham, and Simon Gibbs. Towards an Elastic Application Model for Augmenting Computing Capabilities of Mobile Platforms. In *Mobileware*, 2010.

[17] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso. Calling the Cloud: Enabling Mobile Phones as Interfaces to Cloud Applications. In *Middleware*, 1-20 2009.

[18] SciMark 2.0 available at http://math.nist.gov/scimark2.