Energy-Optimal Mobile Application Execution: Taming Resource-Poor Mobile Devices with Cloud Clones

Yonggang Wen and Weiwen Zhang
School of Computer Engineering
Nanyang Technological Unviersity
Nanyang Avenue, Singapore 639798
Email:{ygwen, wzhang9}@ntu.edu.sg

Haiyun Luo
US Research Center
China Mobile
Milpitas, CA 95035
Email: {haiyunluo}@chinamobile.com

***Abstract*—In this paper, we propose to leverage cloud computing to tame resource-poor mobile devices. Specifically, mobile applications can be executed in the mobile device (known as mobile execution) or offloaded to the cloud clone for execution (known as cloud execution), with an objective to conserve energy for mobile device. The energy-optimal execution policy is obtained by solving two constrained optimization problems, i.e., how to optimally configure the clock frequency to complete CPU cycles for mobile execution, and how to optimally schedule the data transmission for cloud execution in order to achieve the minimal energy within time delay. Closed-form solutions are obtained for both cases and applied to decide the optimal condition under whether the local execution or the remote execution is more energy-efficient for the mobile device. Moreover, numerical results illustrate that a significant amount of energy (e.g., up to 13 times for a typical mobile application profile) can be saved by optimally offloading the mobile application to the cloud clone.**

***Index Terms*—cloud computing, mobile applications, application offloading, dynamic voltage scaling.**

I. Introduction

Nowadays, resource-hungry applications (e.g., multimedia processing) are finding their way into ubiquitous smartphones. However, due to the limited physical size, the mobile device is inherently resource-constrained [12] in computation, energy, bandwidth and information. In particular, the energy supply from the limited battery capacity [10] has been one of the most challenging design issues with mobile device. Therefore, design decisions for mobile applications have to take consideration of the resource limitation in the mobile device.

The emerging cloud-computing technology[1] offers an opportunity to extend the capabilities of mobile device for energy-hungry salient applications. Various cloud-assisted mobile platforms have been proposed, such as cloudlet [14], cloud clone [4], and etc. In particular, each mobile device is associated with a system-level clone in a cloud infrastructure. The mobile clone, which runs on a virtual machine (VM), can execute mobile applications on behalf of the mobile device. This architecture requires both a mechanism to implement task offloading and a policy to decide when to offload applications. Existing research [14], [2], [13], [4], [5], [17] has proposed a variety of application-offloading mechanisms. However, the research on optimal policies for application offloading to cloud execution is limited in that they mostly consider a fixed computation scheduling in the mobile device and a fixed bandwidth model for the wireless channel [6], [10].

In this paper, we focus on the problem of energy-optimal application execution in the cloud-assisted mobile platform. The objective is to minimize the total energy consumed by the mobile device. When the applications are executed in the mobile device, the computation energy can be minimized by optimally scheduling the clock frequency of the mobile device. When the applications are executed in the cloud clone, the transmission energy can be minimized by optimally scheduling the transmission data rate via a stochastic wireless channel. We formulate both scheduling strategies as constrained optimization problems, with a constraint that the application should be completed within a time deadline. The closed-form solutions for the optimal scheduler and the minimum energy consumed by the mobile device are derived, from which we can decide the optimal condition for energy-efficient application execution. Our numerical results indicate that the optimal policy depends on the application profile (i.e., the input data size and the delay deadline) and the wireless-transmission model. Moreover, the cloud execution can result in significant amount of energy saving for the mobile device.

The rest of this paper is organized as follows. In Section II, we present a model for energy consumption in the mobile execution and the cloud execution. In Section III and IV, we solve the optimization problems for the optimal CPU clock-frequency scheduling in the mobile execution and the optimal transmission scheduling in the cloud execution. In Section V, analytical results from previous two sections are applied for the decisions of optimal execution for mobile applications. Section VI summarizes this paper and provides future directions.

II. System Modeling and Problem Formulation

In this section, we present a mathematical model for application execution on the cloud-assisted mobile application platform. First, we define a mobile application profile. Then,

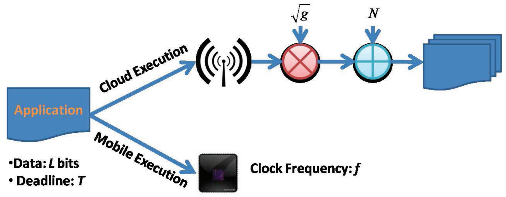

Fig. 1. Mobile application executed in two alternative modes: the mobile execution (lower) and the cloud execution (upper).

we introduce energy consumption models for application execution, including a computation energy model for mobile execution and a transmission energy model for cloud execution.

A. Mobile Application Model

We consider an application-level task execution, as illustrated in Figure 1. On the cloud-assisted mobile application platform, a mobile application can be executed either on the mobile device (known as *mobile execution*) or on the cloud clone (known as *cloud execution*). The objective is to develop an optimal application-execution policy, minimizing the energy consumed by the mobile device.

We denote an application profile as $A(L,T)$, where L and T are the two parameters for the application given as follows:

- Input data size L: the number of data bits as the input to the application;
- Application completion deadline T: the delay deadline before which the application should be completed.

B. Mobile Execution Energy Model

When the application is executed by the mobile device, the energy consumption is determined by CPU workload. The workload is measured by the number of CPU cycles required by the application, denoted as W, which depends on the input data size and the algorithm in the application.

For the mobile execution, its computation energy can be minimized by optimally configuring the clock frequency of the chip, via the dynamic voltage scaling (DVS) technology [11]. In CMOS circuits [3], the energy per operation $\mathcal{E}_{op}$ is proportional to V^2, where V is the supply voltage to the chip. Moreover, it has been observed that the clock frequency of the chip, f, is approximately linearly proportional to the voltage supply of V [3]. Therefore, the energy per operation can be expressed as $\mathcal{E}_{op} = \kappa f^2$, where κ is the energy coefficient depending on the chip architecture. The optimization problem can then be formulated as,

$$\mathcal{E}_m^* = \min_{\psi \in \Psi} \mathrm{E}\{\mathcal{E}_m(L,T,\psi)\}, \tag{1}$$

where ψ is any clock-frequency vector that meets the delay deadline, Ψ is the set of all feasible clock-frequency vectors, $\mathcal{E}_m(L,T,\psi)$ is the energy consumed by the mobile device. This optimization problem will be solved in Section III.

C. Cloud Execution Energy Model

When the application is executed by the cloud clone, the energy consumed by the mobile device depends on the amount of data transmitted from the mobile device to the cloud clone and the wireless channel model. For any mobile application $A(L,T)$, L bits of data needs to be transmitted to the cloud clone. Note that the binary exe file for the application has been replicated on the cloud clone initially. As such, it does not incur additional energy cost. We assume a Markovian fading model for the wireless channel between the mobile device and the cloud clone. A specific model (i.e., the Gilbert-Elliott model) for the channel gain will be presented in Section IV-A.

In this research, we adopt an empirical transmission energy model as in [7], [8] [18], [19]. Specifically, for a wireless fading channel with a gain of g, the energy consumed to transfer s bits of data over the channel within a time slot is governed by a convex monomial function, i.e.,

$$\mathcal{E}_t(s,g,n) = \lambda \frac{s^n}{g}, \tag{2}$$

where n denotes the monomial order, and λ denotes the energy coefficient. It has been shown that some practical modulation scheme exhibits an energy-bit relation that can be well approximated by a monomial. It is normally assumed that $2 \le n \le 5$.

In the cloud execution, it is possible to minimize the total transmission energy consumption by optimally varying the data rate (the number of transmitted bits in a given time slot), in response to a stochastic channel. Under an optimal transmission scheduling, the minimum amount of energy consumed by the mobile device for the cloud execution is given by

$$\mathcal{E}_c^* = \min_{\phi \in \Phi} \mathrm{E}\{\mathcal{E}_c(L,T,\phi)\}, \tag{3}$$

where ϕ denotes a data transmission schedule that meets the delay deadline, Φ is the set of all feasible data schedules, and $\mathcal{E}_c(L,T,\phi)$ denotes the transmission energy. This optimization problem will be solved in Section IV.

D. Optimal Application Execution Policy

The decision for optimal application execution is to choose where to execute the application, with an objective to minimize the total energy consumed by the mobile device. Specifically, the optimal policy is determined by the following rule,

$$\begin{cases} \text{Mobile Execution} & \text{if} \quad \mathcal{E}_m^* \le \mathcal{E}_c^* \\ \text{Cloud Execution} & \text{if} \quad \mathcal{E}_m^* > \mathcal{E}_c^*. \end{cases} \tag{4}$$

To decide an optimal application execution strategy, we will first solve the two optimization problems, as specified in Eq. (1) and Eq. (3).

III. Optimal Computation Energy for Mobile Execution

In this section, we investigate the problem of minimizing the computation energy for mobile execution, by optimally setting the clock frequency of the chip.

A. Probabilistic Task Execution in Mobile Device

Let W indicate the number of CPU cycles needed for an application. For a given input data size, L, it can be expressed as $W = LX$ [10], where X has been shown to be a random variable with an empirical distribution[9]. The estimation of this distribution, which depends on the nature of the application, has been treated in [15], [16], and thus is beyond the scope of this paper. In this paper, we assume that the probability distribution function (PDF) of X is $P(x)$, and its cumulative distribution function (CDF) is defined as $F_X(x) = \Pr[X \leq x]$, and its complementary cumulative distribution function (CCDF), denoted as $F_X^c(w)$, is defined as $F_X^c(x) = 1 - F_X(x)$. Therefore, the CDF of the workload W is given by $F_W(w) = F_X(w/L)$, and its CCDF is given by $F_W^c(w) = F_X^c(w/L)$.

As shown in [9], [15], [16], the number of CPU cycles per bit can be modeled by a Gamma distribution. The PDF of the Gamma distribution is given by

$$p_X(x) = \frac{1}{\beta\Gamma(\alpha)}(\frac{x}{\beta})^{\alpha-1}e^{-\frac{x}{\beta}}, \quad \text{for} \quad x > 0, \tag{5}$$

which depends on 2 parameters(the shape α and the scale β).

In this paper, we adopt a probabilistic performance requirement. Specifically, each application should meet its deadline with a probably of ρ by allocating W_ρ CPU cycles. The parameter ρ is called the application completion probability (ACP). The probability that each job requires no more than the allocated W_ρ cycles is at least ρ, i.e.,

$$F_W(W_\rho) = \Pr[w \leq W_\rho] \geq \rho. \tag{6}$$

Thus, we can obtain the number of CPU cycles, for a given ρ, as

$$W_\rho = F_W^{-1}(\rho) = LF_X^{-1}(\rho), \tag{7}$$

which is the ρ^{th} quantile for the distribution of W.

B. Energy-Efficient Clock-Frequency Configuration

We assume that $f(w)$ be a clock-frequency schedule vector, where w is the number of CPU cycles it has completed previously. Therefore, the energy consumption is given by

$$\mathcal{E}_m = \kappa \sum_{w=1}^{W_\rho} F_W^c(w)[f(w)]^2, \tag{8}$$

where κ is the energy efficiency parameter and $F_W^c(w)$ is the probability in which the application has not completed after w CPU cycles. The optimization problem in Eq. (1) can be rewritten as,

$$\min_{f(w)} \quad \kappa \sum_{w=1}^{W_\rho} F_W^c(w)[f(w)]^2, \tag{9}$$

$$\text{s.t.} \quad \sum_{w=1}^{W_\rho} \frac{1}{f(w)} \leq T, f(w) > 0 \tag{10}$$

where Eq. (10) corresponds to the delay constraint.

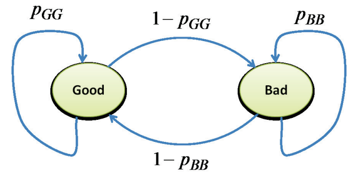

Fig. 2. Channel models: the Gilbert-Elliott (GE) channel model.

For the optimal CPU scheduling problem in Eq. (9), the optimal clock scheduling policy is

$$f^*(w) = \frac{\theta}{T[F_W^c(w)]^{1/3}}, \tag{11}$$

where $\theta = \sum_{i=1}^{W_\rho}[F_W^c(i)]^{1/3}$. The optimal computation energy is

$$\mathcal{E}_m^* = \frac{\kappa}{T^2}\{\sum_{w=1}^{W_\rho}[F_W^c(w)]^{1/3}\}^3. \tag{12}$$

It can also be shown that, as the input data size increases, the minimum computation energy scales cubically with the number of input data bits, i.e., $\mathcal{E}_m^* \sim L^3$.

IV. Optimal Transmission Energy for Cloud Execution

In this section, we consider the problem of scheduling data transmission to wireless (fading) channel variations, under a deadline constraint. As such, we first briefly describe the channel model. Next, we derive the minimum expected energy expenditure for transmission.

A. Wireless Channel Model

We consider the scheduling of L bits data with a deadline in T discrete time slots. The channel state at time slot t is denoted as g_t. We assume that only causal knowledge of the channel state are available.

We adopt the Gilbert-Elliott (GE) channel model [18], [19] in which there are two channel states $\{g_t\}$: "good" and "bad" channel conditions. If the measured channel gain is above some value, the channel is labeled as good. Otherwise, the channel is labeled as bad. Let the (average) channel gains of the good and bad states be g_G and g_B, respectively.

In this model, as illustrated in Figure 2, the state transition matrix is completely determined by the values p_{GG} (for the probability that the next state is the good state, given that the current state is also the good state) and p_{BB} (for the probability that the next state is the bad state, given that the current state is also the bad state). Accordingly, we have $p_{GB} = 1 - p_{GG}, p_{BG} = 1 - p_{BB}$, where p_{GB} denotes the probability in which channel will transit from the good state to the bad state in the next time slot and p_{BG} denotes the probability in which channel will transit from the bad state to the good state in the next time slot. The state sojourn time is geometrically distributed. As such, the mean state sojourn time (duration of being in a state), measured in number of steps in this state, is given by $T_G = \frac{1}{1-p_{GG}}, T_B = \frac{1}{1-p_{BB}}$.

B. Optimal Data Transmission Scheduling

We consider a discrete time model as in [7], [8]. We denote t as discrete time index in descending order (from $t = T$ to $t = 1$). In time slot t, if the number of bits transmitted is s_t, the transmission energy cost is $\mathcal{E}_t(s_t, g_t) = \lambda \frac{s_t^n}{g_t}$. Therefore, the optimization problem in Eq. (3) for the optimal data-transmission schedule can be rewritten as,

$$\begin{aligned} \min_{s_t}: \quad & \mathrm{E}\left[\sum_{t=1}^{T} \mathcal{E}_t(s_t, g_t)\right] \\ \text{s.t.:} \quad & \sum_{t=1}^{T} s_t = L, s_t \geq 0, \forall t. \end{aligned} \tag{13}$$

For the GE channel model, the minimum expected energy depends on the channel state at $t = T + 1$. If, at $t = T + 1$, the channel is in the good state, the optimal number of data bits transmitted in each time slot is given by

$$s_t^*(l_t, g_t) = \begin{cases} l_t \left(\frac{(g_t)^{\frac{1}{n-1}}}{(g_t)^{\frac{1}{n-1}} + (\frac{1}{\zeta_{t-1,G}})^{\frac{1}{n-1}}} \right), & t \geq 2; \\ l_1, & t = 1, \end{cases} \tag{14}$$

where l_t is the number of unfinished bits at time slot t, and

$$\zeta_{t;G} = \begin{cases} p_{GG} \left[\left(\frac{1}{(g_G)^{\frac{1}{n-1}} + (\frac{1}{\zeta_{t-1;G}})^{\frac{1}{n-1}}} \right)^{n-1} \right] \\ + p_{GB} \left[\left(\frac{1}{(g_B)^{\frac{1}{n-1}} + (\frac{1}{\zeta_{t-1;G}})^{\frac{1}{n-1}}} \right)^{n-1} \right], & t \geq 2; \\ p_{GG} \left[\frac{1}{g_G} \right] + p_{GB} \left[\frac{1}{g_B} \right], & t = 1. \end{cases} \tag{15}$$

With this optimal scheduling, the minimum expected energy is given by:

$$\mathcal{E}_t(L; G) = \lambda L^n \zeta_{t;G}. \tag{16}$$

If, at $t = T + 1$, the channel is in the bad state, the optimal number of data bits transmitted in each time slot is given by

$$s_t^*(l_t, g_t) = \begin{cases} l_t \left(\frac{(g_t)^{\frac{1}{n-1}}}{(g_t)^{\frac{1}{n-1}} + (\frac{1}{\zeta_{t-1,B}})^{\frac{1}{n-1}}} \right), & t \geq 2; \\ l_1, & t = 1, \end{cases} \tag{17}$$

where

$$\zeta_{t;B} = \begin{cases} p_{BB} \left[\left(\frac{1}{(g_B)^{\frac{1}{n-1}} + (\frac{1}{\zeta_{t-1;B}})^{\frac{1}{n-1}}} \right)^{n-1} \right] \\ + p_{BG} \left[\left(\frac{1}{(g_G)^{\frac{1}{n-1}} + (\frac{1}{\zeta_{t-1;B}})^{\frac{1}{n-1}}} \right)^{n-1} \right], & t \geq 2; \\ p_{BB} \left[\frac{1}{g_B} \right] + p_{BG} \left[\frac{1}{g_G} \right], & t = 1. \end{cases} \tag{18}$$

With this optimal scheduling, the minimum expected energy is given by:

$$\mathcal{E}_t(L; B) = \lambda L^n \zeta_{t;B}. \tag{19}$$

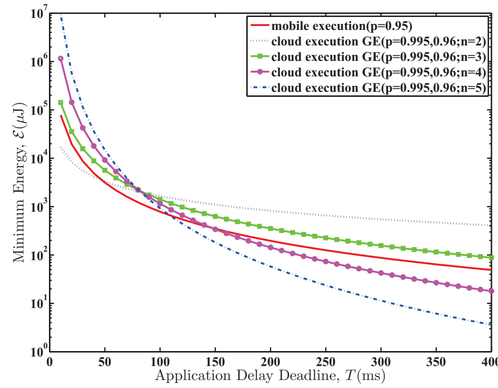

Fig. 3. The minimum energy, $\mathcal{E}^*$, is plotted as a function of the application delay deadline. The task load is modeled as Gamma distribution, with $\alpha = 4$, $\beta = 200$, and $L = 800bits$. The wireless channel is assumed as the Gilbert-Elliott model with $p_{GG} = 0.995$, $p_{BB} = 0.96$, $g_G = 1$ and $g_B = 0.1$.

These results can be proved by induction, which is given in [20]. Given that, at steady state, the probability that a channel is in good or bad state is $\frac{T_G}{T_G+T_B}$ and $\frac{T_B}{T_G+T_B}$, respectively, the minimum expected transmission energy $\mathcal{E}_c^*$ is:

$$\begin{aligned} \mathcal{E}_c^*(L, T) \quad &= \quad \frac{T_G}{T_G + T_B} \mathcal{E}_t(L; G) \\ &+ \quad \frac{T_B}{T_G + T_B} \mathcal{E}_t(L; B). \end{aligned} \tag{20}$$

It can also be shown that, as the application completion deadline of T increases, the minimum transmission energy decreases monotonically and scales with a factor of $T^{-(n-1)}$, where n is the monomial order in Eq. (2).

V. Optimal Application Execution Policy

In this section, we develop the optimal application execution policy, based on the analytical results in Section III and Section IV. In particular, for a given application profile of $A(L, T)$, we compare the minimum computation energy for the mobile execution and the minimum transmission energy for the cloud execution. The optimal application execution policy is to choose whichever consumes less energy by the mobile device, in order to extend the battery life.

As shown In Figure 3, the optimal policy depends on the monomial order of n. On one hand, when n is smaller than 3, the cloud execution is more energy-efficient when the delay deadline is below a threshold. This is because, when $n < 3$, the scaling factor for the cloud execution is slower than T^{-2}, the scaling factor for the mobile execution. On the other hand, when n is larger than 3, the cloud execution is more energy-efficient when the delay deadline is beyond a threshold. This is because, the scaling factor for the cloud execution is faster than T^{-2}, the scaling factor for the mobile execution. Moreover, by optimally deciding where to execute the application, a significant amount of energy can be saved on the mobile devices. For example, for an application profile of $A(800bits, 400ms)$, the mobile execution consumes 13 times energy more than the cloud execution for $n = 5$.

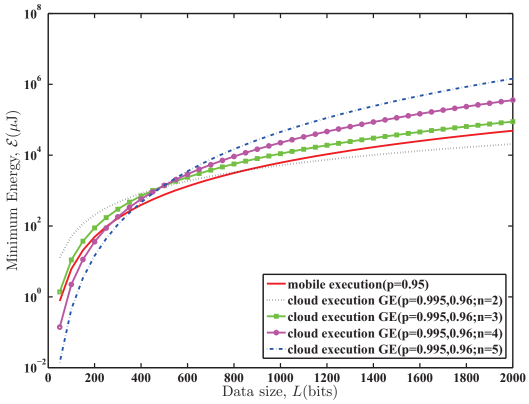

Fig. 4. The minimum energy, $\mathcal{E}^*$, is plotted as a function of the data input size L. The task load is modeled as Gamma distribution, with $\alpha = 4$, $\beta = 200$, and $T = 50ms$. The wireless channel is assumed as the Gilbert-Elliott model with $p_{GG} = 0.995$, $p_{BB} = 0.96$, $g_G = 1$ and $g_B = 0.1$.

As shown in Figure 4, the optimal policy depends on the monomial order of n. On one hand, when n is smaller than 3, the cloud execution is more energy-efficient when the data size is beyond a threshold. This is because, when $n < 3$, the scaling factor for the cloud execution is slower than L^3, the scaling factor for the mobile execution. On the other hand, when n is larger than 3, the cloud execution is more energy-efficient when the input data size is below a threshold. This is because, when $n > 3$, the scaling factor for the cloud execution is faster than L^3, the scaling factor for the mobile execution.

VI. Summary and Future Research

In this paper we investigated the problem of how to conserve energy for the resource-constrained mobile device, by optimally executing mobile applications in either the mobile device or the cloud clone. We proposed an optimization framework for energy-optimal application execution in the cloud-assisted mobile application platform. For the mobile execution, we aim to minimize the computation energy by dynamically configuring the clock frequency of the chip, according to the workload distribution. For the cloud execution, we aim to minimize the transmission energy by optimally scheduling data transmit across a stochastic wireless channel (i.e., the Gilbert-Elliott model). Closed-form solutions were obtained for both scheduling problems and are applied to decide the optimal application-execution condition under which either the mobile execution or the cloud execution is more energy-efficient for the mobile device. Numerical results indicate that the optimal execution policy depends on the application profile and the wireless transmission model. This paper focuses only on the energy issue on the mobile device. For future work, the energy consumption in the cloud side will be taken into consideration.

Acknowledgment

The authors would like to thank Dr. Kyle C. Guan and Dr. Dan Kilper at Bell Laboratories for their insightful suggestions on solving the wireless channel scheduling problem. This work also benefits from discussions with Dr. Xinwen Zhang, Dr. Xiaoqing Zhu, Dr. Shivkumar Kalyanaraman at IBM Research - India, and Prof. Changwen Chen at the University at Buffalo, State University of New York. Moreover, Yonggang Wen would like to thank Singapore Nanyang Technological University for the start-up grant support of this research.

References

[1] Armbrust, M; Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Zaharia, (2010). "A view of cloud computing.". Communication of the ACM 53 (4): 5058.

[2] R. K. Balan, et al, "The Case for Cyber Foraging," In Proceedings of 10th ACM Special Interest Group on Operating Systems European Workshop(SIGOPS), ACM Process, 2002, pp. 87-92.

[3] T. Burd and R. Broderson, "Processor Design for Portable Systems," Journal of VLSI Singapore Process, Vol. 13, No. 2, August 1996, pp. 203-222.

[4] B. G. Chun and P. Maniatis, "Augmented Smartphone Applications Through Clone Cloud Execution," In Proceedings of the 12th conference on Hot topics in operating systems, Berkeley, CA, USA, 2009.

[5] B.G. Chun, S.h. Ihm, P. Maniatis, M. Naik and A. Patti, "CloneCloud: Elastic Execution between Mobile Device and Cloud," In Proceedings of the 6th European Conference on Computer Systems (EuroSys 2011), April 2011.

[6] K. Kumar and Y. H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" IEEE Computer, Vol. 43, No.4, pp. 51-56, April 2010.

[7] J. Lee and N. Jindal, "Delay Constrained Scheduling over Fading Channels: Optimal Policies for Monomial Energy-Cost Functions," IEEE International Conference on Communications (ICC), Dresden, Germany, June 2009.

[8] J. Lee, and N. Jindal, Energy-efficient Scheduling of Delay Constrained Traffic over Fading Channels, IEEE Trans. Wireless Communications, vol. 8, no. 4, pp. 1866-1875, April 2009.

[9] J. R. Lorch and A. J. Smith, "Improving Dynamic Voltage Scaling Algorithms with PACE," Proceedings of ACM SIGMETRICS 2001, Cambridge, MA, USA, June 2001.

[10] A. P. Miettinen and J. K. Nurminen, "Energy Efficiency of Mobile Clients in Cloud Computing," In Proceedings of the 2nd USENIX conference on hot topics in cloud computing, Berkeley, CA, USA, 2010.

[11] J. M. Rabaey, Digital Integrated Circuits, Prentice Hall, 1996.

[12] M. Satyanarayanan, "Fundamental Challenges in Mobile Computing," In Proceedings of ACM Symposium on Principles of Distributed Computing, ACM Press, 1996, pp. 1-7.

[13] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," IEEE Personal Communication, Vol. 8, No. 4, 2001, pp.10-17.

[14] M. Satyanarayanan, P. Bahl, R. Caceres and N. Davies, "The Case for VM-based Cloudlets in Mobile Computing," IEEE Pervasive Computing, Vol. 8, No. 4, pp. 14-23, Oct.-Dec. 2009.

[15] W. H. Yuan and K. Nahrstedt, "Energy-Efficient Soft Real-Time CPU Scheduling for Mobile Multimedia Systems," Proceedings of ACM SOSP'03, October 19-22, 2003, New York, USA.

[16] W. H. Yuan and K. Nahrstedt, "Energy-Efficient CPU Scheduling for Multimedia Applications," ACM Transactions on Computer Systems, Vol. V, NO. N, June 2005, pp.1-36.

[17] X. W. Zhang, A. Kunjithapatham, S. Jeong and S. Gibbs, "Towards an Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing," Mobile Network Applications, 2011(16), pp. 270-284.

[18] M. Zafer and E. Modiano, "Delay Constrained Energy Efficient Data Transmission over a Wireless Fading Channel," Workshop on Information Theory and Application, University of California, San Diego, Feburary 2007.

[19] M. Zafer and E. Modiano, "Minimum Energy Transmission over a Wireless Fading Channel with Packet Deadlines," Proceedings of IEEE Conference on Decision and Control (CDC), New Orleans, LA, December 2007.

[20] Y.G. Wen, W.W. Zhang, K. Guan, D. Kilper and H. Y. Luo, Energy-Optimal Execution Policy for A Cloud-Assisted Mobile Application Platform, Technical Report, September 2011, http://www.ntu.edu.sg/home/ygwen/CAMAP-Technical-Report.pdf