35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Toward Energy-Efficient and Distributed Mobile Health Monitoring Using Parallel Offloading

Jong Hoon Ahnn and Miodrag Potkonjak
Department of Computer Science, Unviersity of California, Los Angeles, USA
{jhahnn,miodrag}@cs.ucla.edu

Abstract

Although mobile health monitoring where mobile sensors continuously gather, process, and update sensor readings (e.g. vital signals) from patient's sensors is emerging, little effort has been investigated in an energy-efficient management of sensor information gathering and processing. Mobile health monitoring with the focus of energy consumption may instead be holistically analyzed and systematically designed as a global solution to optimization subproblems. We propose a distributed and energy-saving mobile health platform, called mHealthMon where mobile users publish/access sensor data via a cloud computing-based distributed P2P overlay network. The key objective is to satisfy the mobile health monitoring application's quality of service requirements by modeling each subsystem: mobile clients with medical sensors, wireless network medium, and distributed cloud services. By simulations based on experimental data, we present the proposed system can achieve up to 10.1 times more energy-efficient and 20.2 times faster compared to a standalone mobile health monitoring application, in various mobile health monitoring scenarios applying a realistic mobility model.

1. Introduction

Many successful health care applications based on mobile computing and communication technologies have been presented in the literature. Lv et al. [1] utilizes wireless body sensors and smart phones to monitor the wellbeing of the elderly. The key enabler is a smartphone that automatically alerts preassigned people who could be their family and friends, and call the ambulance of the emergency center. As an example of more sophisticated health monitoring systems, Chowdhury et al. proposed MediAlly, a middleware for supporting energy-efficient, long-term remote health monitoring, where sensor data is collected using physiological sensors and transported back to the middleware using a smartphone [4]. Smartphones with medical sensors attached to patients can perform publish/access medical sensor updates such as vital signals, measure personalized estimates of impact and exposure, and share patient's live health information. mHealthMon is similar in a spirit of a middleware-based system but it is fundamentally different in terms of an energy-saving paradigm.

In SensorBase [9], back-end servers (called republishers) further process sensor data to enable sensor data searching. SensorMap [10] is a web portal service that provides mechanisms to archive and index data, process queries, and aggregate and present results on geocentric Web. mHealthMon differs from these approaches in that it focuses on large-scale participatory sensing and facilitates location-sensitive information sharing via a scalable structured P2P overlay that efficiently supports location-sensitive data publish/retrieval. Similar to other works such as GeoServ [8], mHealthMon is a two-tier mobile health monitoring platform that exploits the P2P Internet infrastructure.

The approaches from MAUI [5], CloneCloud [3], Cloudlets [7], and Zhang et al. [14] 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 [15] is similar to above, but it lacks of dynamic adaptation of the computation between mobile devices and cloud services. Prior work mostly focused on saving energy consumption on mobile devices; in contrast, mHealthMon provides analytical cost models to optimize the entire energy consumption including network and cloud at the same time.

The key contributions are summarized in the following. We explicitly model the performance of mobile health monitoring system – mobile clients with medical sensors, wireless network medium, and cloud services – using two aspects: computation and communication cost. We propose a distributed optimized solution of complex mobile health monitoring: program partition-

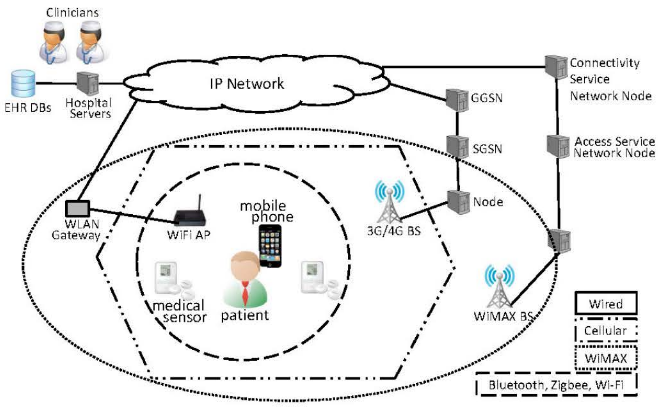

Figure 1. A high-level overview of smartphone-based mobile health network architecture in a heterogeneous wireless network interfaces scenario.

Figure 2. We developed an iOS-based mobile health monitoring application: a screenshot for patient's EHR.

Figure 3. A screenshot for a list of available medical sensors via Bluetooth.

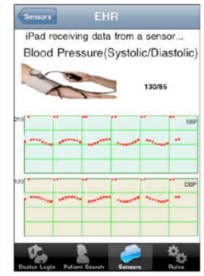

Figure 4. A screenshot for sensor feeding (systolic/diastolic) from a blood pressure sensor.

ing , network resource allocation, and network selection problem. We propose a location-aware sensor data retrieval scheme called mHealthMon that supports geographic range queries, and a location-aware publish-subscribe scheme that enables energy-efficient multicast routing over a group of subscribed users. We prototype energy-optimized mobile health monitoring applications to prove the feasibility of our proposed techniques in various sensing scenarios applying a realistic mobility model utilizing parallel offloading.

2. System Model

Our prior work, GeoServ [8] mainly focuses on how to store in and retrieve sensor data from external storage systems, where the location-awareness is the main consideration on its data management over an overlay-based P2P routing. Thus, GeoServ is a general purpose urban sensing P2P storage with no consideration of performance modeling, energy saving and optimization, and computation offloading.

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 [12].

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 sensing 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.

2.2. Wireless Network Model

We 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 many researchers such as [12] and [11], RATs can be largely characterized into two categories based on means to share their channels: interference constrained RATs and orthogonal RATs [12].. In this paper, we only consider the latter.

3. Optimization Problem Formulation

We mainly solve different problems: a program partitioning program (P1), network resource allocation problem (P2), network selection problem (P3), and cloud resource allocation problem (P4). In this work, we mainly focus on the first three. 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.

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,

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

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'. Furthermore, the calculation of communication cost incurred due to code offload is given by,

$$T_{v'} = n_{v'} \times D_{m,b}, \quad (2)$$

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 $n_{v'}$. 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 2 consists of a concave objective over linear constraints, and it becomes convex. Therefore, there are various convex optimization algorithms to solve it from [16].

3.2. Network Resource Allocation Problem

We consider a utility metric as the effectiveness of allocated resources of networked systems in our optimization problem as $U = \sum_m \sum_b D_{m,b}$. 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 $U = \sum_m \frac{w_m}{1-\alpha} \sum_b D_{m,b}^{1-\alpha}$, where $0 \leq \alpha < 1$. Now, we present an optimization problem as,

$$\begin{array}{ll} max & U, \\ s.t. & \sum_m \frac{D_{m,b}}{\bar{D}_{m,b}} \leq \Gamma_b, \\ & \sum_b D_{m,b} \geq D_{min,b}, \\ & D_{m,b} \geq 0, \end{array} \quad (3)$$

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 3 consists of a concave objective over linear constraints and thus is convex. That means there exists various algorithms to solve the problem immediately [16].

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

3.4. 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

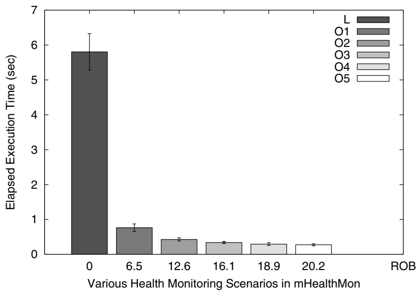

Figure 5. The average execution time of mHealthMon with various offloading scenarios are measured and presented with 95 percent confidence intervals.

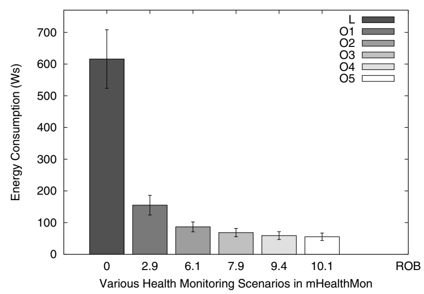

Figure 6. The average energy consumption of mHealthMon with various offloading scenarios are measured and presented with 95 percent confidence intervals.

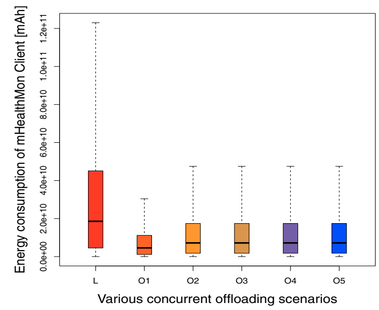

Figure 7. Energy consumption [Watts] for the simulation period is presented. The total amount of energy consumed by each cell is averaged.

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: mHealthMon: an orchestrated approach to four different optimization algorithms to achieve a global optimization objective in a distributed manner.

than stochastic processes. Algorithm 1 solves P2, P3, P4, and P1 in order until the mobile cloud computing facility based on data centers ends.

4. System Evaluation

Evaluating the performance consists of two parts: the performance in a mobile device and the one in a cloud machine. The former is presented in the computation cost and communication cost in five offload scenarios: local execution (L) in a mobile client, offloading with 1-5 concurrent requests (O1-O5). Note that O1 represents serial offloading similar to MAUI [5], [3], and Cloudlets [7]. For our proposed scheme, O2-O5 stands for asynchronous parallel offloading with the different number of concurrent parallel requests.

4.0.1. Evaluating Mobile Clients. This section analyzes computation cost and communication cost in time and energy when applying optimization strategies presented in this paper. We pose a fundamental question about whether or not there exist real benefits when offloading code. Figure 5 compares the execution time between the mobile and the cloud by applying to a typical mobile health monitoring application. Figure 6 compares energy consumption in the same settings. As discussed, we also study how concurrent offloading requests help save time and energy in various scenarios: O1-O5. ROB presents relative offload benefits, comparing each offloading case with the non-offloading case L. We present the proposed system with the help of 5 parallel execution (O5) can perform up to 20.2 times faster and 10.1 times more energy-efficient compared to a standalone mobile health application L. We also observe that our work (O2-O5) outperforms over non-parallel offloading schemes (O1) such as MAUI [5] and CloneCloud [3], resulting in at least 2 times better both in time and energy. We observe the overall time saving and energy saving rate increases as the number of concurrent requests increases. This is done by a non-blocking (asynchronous) offload request.

4.0.2. Evaluating Cloud Servers. For mobile scenarios, we use VanetMobiSim for mobility trace for the duration of 300s [13].. We use the network area size of 12800m×12800m. The Westwood topology from Tigermap (TGR06037, Los Angeles [2]) represents the area in the vicinity of the UCLA campus. We discretize the network area into grids for the Hilbert curve-based linearization, resulting 100×100 grids. Geographic range queries are made by specifying a square area (e.g., 4×4 grids). Each mobile node reports sensor data to its associated overlay node every second. The size of data is set to 128 Bytes (e.g., GPS sample, timestamp, accelerometer samples). We assume that each node knows its accurate geographic coordinate and thus can dynamically change their associated overlay node without any errors (e.g., no bouncing at the boundary). In GeoTable, the number of long links is set to five, as recommended in Symphony DHT [6]. Unless otherwise mentioned, for each configuration we report the average value of 30 runs. For simplicity, orthogonal-based WLAN and TDMA-based GSM are only considered. Unless specified, the network selection follows a strategy given in Equation 6 and network parameters form [11]. We assume that each BS knows its accurate geographic coordinate and thus can dynamically change their associated BS without any errors (e.g., no bouncing at the boundary). In our simulation, each mobile client randomly chooses one of four mobile applications. The duration of execution time and energy of each mobile application is given by experimental results. When one application is done, another random assignment is made automatically during our simulation period.

4.0.3. Large-Scale Simulation. Figure 7 presents the energy consumption with six different offloading scenarios in the case of 128×128 grid cells. We assume each cell has only one base station. As a ground truth, we first show local execution of mobile health applications with no offloading capability, resulting in no energy saving. The mobility is generated by the VanetMobisim simulator. By applying our proposed offloading technique with concurrent request capability, we see clear improvement of time and energy saving in a static viewpoint. From O2 to O5 scenarios, we apply parallelism with the different number of concurrent requests, resulting in slightly 2x more energy saving compared to local execution only. This means applying our optimization techniques saves lots of energy by utilizing cloud

5. Conclusion

We proposed a distributed and energy-saving mobile health monitoring platform. By simulations, we showed the proposed system can perform up to 10.1 times more energy-efficient and 20.2 times faster compared to a standalone mobile health application. We also compared our work with non-parallel offloading schemes such as MAUI [5] and CloneCloud [3], resulting in at least 2 times better both in time and energy.

[1]

[1]

References

[1] Z. Lv, F. Xia, G Wu, L Yao, and Z Chen. iCare: A Mobile Health Monitoring System for the Elderly. In *GreenCom*, 2010.

[2] U.S. Census Bureau. TIGER. Available at www.census.gov/geo/www/tiger.

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

[4] A. R. Chowdhury, B. Falchuk. MediA lIy: A Provenance-Aware Remote Health Monitoring Middleware. In *PerCom*, 2010.

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

[6] G. S. Manku, M. Bawa, P. Raghavan. Symphony: Distributed Hashing in a Small World. *USITS*, 2003.

[7] 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.

[8] J. H. Ahnn, U. Lee, and H. J. Moon. GeoServ: A Distributed Urban Sensing Platform. *CCGRID*, 2011.

[9] S. Reddy, G. Chen, B. Fulkerson, S. J. Kim, U. Park, N. Yau, J. Cho. Sensor-Internet Share and Search – Enabling Collaboration of Citizen Scientists. In *DSI*, 2007.

[10] S. Nath, J. Liu, and F. Zhao. SensorMap for Wide-Area Sensor Webs. In *IEEE Computer Magazine*, 40(7), Jul. 2007.

[11] J. Buhler, G. Wunder. An optimization framework for heterogeneous access management. *WCNC*, 2525-2530, 2009.

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

[13] J. Härri, M. Fiore, F. Fethi. VanetMobiSim: Generating Realistic Mobility Patterns. *VANET*, 2006.

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

[15] 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.

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