

# Bring-Your-Own-Application (BYOA): Optimal Stochastic Application Migration in Mobile Cloud Computing

Jonathan Chase<sup>†</sup>, Dusit Niyato<sup>†</sup>, and Sivadon Chaisiri<sup>‡</sup>

<sup>†</sup> School of Computer Engineering, Nanyang Technological University, Singapore

<sup>‡</sup> Cyber Security Lab, Department of Computer Science, University of Waikato, New Zealand

**Abstract**—The increasing popularity of using mobile devices in a work context, has led to the need to be able to support more powerful computation. Users no longer remain in an office or at home to conduct their activities, preferring libraries and cafes. In this paper, we consider a mobile cloud computing scenario in which users bring their own mobile devices and are offered a variety of equipment, e.g., desktop computer, smart-TV, or projector, to migrate their applications to, so as to save battery life, improve usability and performance. We formulate a stochastic optimization problem to optimize the allocation of user applications to equipment despite future uncertainties. Furthermore, we extend the scenario to consider multiple locations and geographic migration. The performance evaluation shows the superior benefit, e.g., higher profit, compared with a baseline algorithm.

**Index Terms**—Mobile cloud computing, application migration, stochastic programming.

## I. INTRODUCTION

Mobile cloud computing is pervasive in all its many forms, such as phones, tablets, and laptops. Such is the ubiquity of these devices, that even users who are not ‘tech savvy’ prefer to use their mobile hardware for work and play, rather than desk-bound equipment. Bring Your Own Device (BYOD) paradigm aims to provide all the features of corporate computing, whilst not inhibiting the convenience and mobility of workers using their own mobile computers. These devices vary in computational resources, operating systems, as well as battery life, and this heterogeneity presents a number of technical challenges. Inspired by this phenomenon, this paper examines a mobile cloud computing setting, with the option to offload computation and migrate user interface to more powerful equipment when constrained by battery life. The surfeit of mobile devices at coffee shops is a clear indicator of the average person’s desire to work ‘on the go’. By employing application migration, users can move away from their battery-limited mobile device with small screen when necessary, and continue working on alternative equipment. We consider an environment in which users can bring their own mobile devices to a location such as a library and offload their application to local equipment such as a desktop computer, smart-TV, or projector. This application migration can be done, e.g., using app and desktop virtualization technology. One example is from Citrix [1]. The term “Bring-Your-Own-Application” expresses the idea that the user brings applications (and

mobile device) and use available facility and equipment. We then examine an extension case, where users are distributed across multiple physical locations, with the option to move between locations to access the best resources. Each problem is formulated as a stochastic optimization and solved, yielding numerical results that support the optimality of our solution.

The structure of the paper is as follows. In Section II, we outline some relevant work to provide a background and basis for our research. In Section III, we provide the context for our two scenarios, which are presented as stochastic programs in Section IV and Section V for each case respectively. Finally, in Section VI we provide test data and numerical results to support our method.

## II. RELATED WORK

Mobile application migration has been a possibility for several years, but may take place at different levels. If a virtual machine (VM) is running on a mobile device, the entire VM can be migrated. Alternatively, applications using middleware platforms such as J2EE can migrate independent of OS. Migration can be seamless, online, directly between mobile devices. An application can be paused on one device and the state transferred to a new device [2] that supports the same middleware.

An application-level approach may be useful to overcome hardware heterogeneity. [3] migrates at the application-level, partitioning the migration to minimize overhead. This also employs Java’s virtualization, similar to the offloading proposed in [4], which partitions to maximize performance. [5] builds on this work, considering the impact of migration on performance, as live migration should be seamless, with minimal overhead. However, whilst application-layer VMs are relatively straightforward to partition for offloading, a significant portion of mobile activity uses web-based technologies, for applications such as social media, and productivity. HTML 5.0’s Web Workers allow easier partitioning of Javascript programs, which allows offloading of web-based applications [6].

The described work focuses on migration to the cloud, but we consider a local offloading point. [7] shows this is possible by using cloud ‘access points’ as base stations. Rather than VMs, however, we employ physical equipment that can be migrated to, similarly to [2]. The ability to carry out a variety of tasks on your own mobile device, with the option

to offload to more potent computers, gives a strong context for the scenario we envisage. Application migration is the logical next step in the development of mobile computing.

This paper aims to exploit an under-researched area, where users may bring their own devices to a commercial location, and continue to work, whilst taking advantage of a variety of offloading options that improve power-efficiency, and simultaneously prove profitable to the service providers.

### III. MIGRATION ENVIRONMENT

We consider a system model, illustrated in Fig. 1, in a mobile cloud computing setting where users travel to a location, such as a library or Internet cafe, with their mobile devices. Here, they can carry out computer-based activities, such as work or communication, by either offloading applications to mains-powered equipment (such as a desktop computer or smart-television) at the location, or by making use of their mobile device (such as a phone or laptop). App and desktop virtualization can be used. Demand is communicated via agent-based discovery, with software running on a user's mobile device communicating over a wireless network with the service provider's Mobile Device Management System (MDMS - a similar concept adopted in the BYOD paradigm). The MDMS can then make a resource provisioning decision and communicate the result to the user via their mobile device.

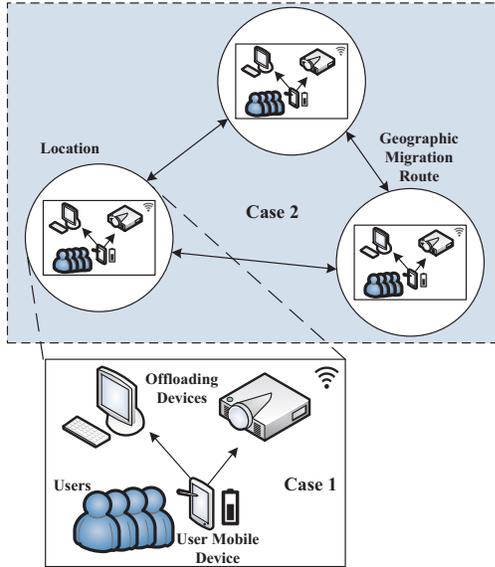


Fig. 1. Illustration of the two problem cases, and their intersection.

A commercial location is managed by a system administrator, who will seek to maximize their profits by charging users for the use of their equipment, much as an Internet cafe would. It is in the administrators' interests, therefore, to maximize the number of users who are able to offload, for as long as they require the equipment available. Users who cannot access equipment, or who run out of battery, are lost customers.

Users do not interact directly with the system administrator, however. A third party who acts as a broker, aims to maximize the profit from user allocation. The broker is location-independent, and is therefore free to consider moving users to alternative locations where equipment may be available. Geographic migration has a negative impact on a user's battery level, and creates a less positive image of the service, which may result in lost business in the future. The broker aims to maximize allocation, and therefore profit.

Users may arrive with their own mobile device, which has a limited battery charge, and varying application requirements. The number of users arriving at any point in time is uncertain, as is the length of their stay, the application that they wish to offload, and the battery level of their mobile device. These factors are only exactly known at the moment of the user's arrival. To optimally allocate equipment to users, we must consider these uncertainties. Thus a decision made in the present will have implications for future equipment availability. We model this using a series of time periods. Users arrive at the beginning of a time period (e.g. every 5 minutes), and are allocated to equipment. In subsequent time periods, users must be allocated to what is available, or be turned away - thus the initial allocation must be well chosen. We can model this scenario using a stochastic optimization problem with multi-stage recourse.

#### A. Uncertainty Parameters

The system model under consideration contains various uncertain factors. For a given time period, the number of users is unknown in advance; it is revealed at the beginning of the time period. Additionally, for each user, the battery level of their mobile device, the application they wish to run, and the duration of their stay are also uncertain. These factors can be modeled as a tuple, containing the number of users, and each of the uncertain factors for each user. Each tuple is a scenario  $\omega \in \Omega$ , where  $\Omega$  is the set of all scenarios. Once the particular scenario that has occurred is known, it is then known as a 'realization', each with a probability. Stochastic optimization aims to maximize the expectation across all scenarios. The set of terms representing all these concepts are given in Table I. In the following two sections, we introduce two allocation cases in this environment.

### IV. CASE 1: SINGLE LOCATION

The first case that we consider concerns a single location. A location administrator aims to maximize their profits by finding the optimal allocation of users to equipment. Users may have a variety of applications that they wish to run, which are compatible with some, but not necessarily all, equipment. Once a user is allocated to a particular piece of equipment, they must be allowed to remain for the full duration of their stay. If a user cannot be allocated to compatible equipment for the full length of their visit, they are turned away. Turning away users does not literally incur a cost to the administrator. However, it can still be regarded as lost business. Thus, to limit the number of rejected users, we model the lost users as

TABLE I  
LIST OF KEY NOTATIONS.

Symbol	Definition
$\mathcal{I}$	Set of all users, while $i \in \mathcal{I}$ denotes the user index.
$\mathcal{J}$	Set of all equipment classes that can be offloaded to. $j \in \mathcal{J}$ denotes the equipment class index.
$\mathcal{T}$	Set of all time periods while $t \in \mathcal{T}$ denotes the time period index.
$\mathcal{P}$	Set of all physical locations while $p \in \mathcal{P}$ denotes the location index.
$F_{ij}(\omega)$	Binary compatibility switch, indicating compatibility between user's application and offloading equipment.
$D_i^{(n)}(\omega), D_i^{(b)}(\omega), D_i^{(m)}(\omega)$	Demand parameters, for network and battery for a user's application, and the battery migration cost for user $i$ .
$D_{it}(\omega), D_{ipt}(\omega)$	Binary demand value for user $i$ and the equivalent value by provider for Case 2.
$T^{(n)}, T_i^{(b)}(\omega)$	Capacity of network, user's mobile device battery under scenario $\omega$ .
$T_j, T_{jp}$	The amount of available equipment of class $j$ , and the equivalent capacity for each provider for Case 2.
$B_j$	Binary parameter indicating whether equipment class $j$ uses mobile battery.
$C_j, C_{jp}, C_i, C_i^{(m)}$	Equipment profits under Case 1 and 2, respectively. Cost of losing users and migrating users, respectively.
$x_{ijt}(\omega), x_{ijpt}(\omega)$	Decision variables for equipment allocation in Case 1 and Case 2, respectively.
$z_{it}(\omega), m_{ipt}(\omega)$	Penalty variables for lost users and user migration.

a negative cost in the problem formulation. The profit gained by a provider in a given scenario is calculated as the sum of profits from each user-equipment allocation across all time periods, minus the total penalty cost of all lost users in the same time frame. Given the uncertainties, we aim to maximize the net expected profit given the set of possible scenarios. This cost function is given in (1).

$$\mathcal{Q}(\omega) = \max \mathbb{E}_\Omega \left[ \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} (C_j x_{ijt}(\omega) - C_i z_{it}(\omega)) \right], \quad (1)$$

Application migration is performed wirelessly across the location's network. Since this network has a bandwidth limit, the bandwidth requirements and application migration bandwidth costs for all applications must be less than the bandwidth capacity of the network. This is modelled in (2), where the sum of the bandwidth costs of each user-equipment allocation must be less than the network capacity in each time slot.

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} D_i^{(n)}(\omega) x_{ijt}(\omega) \leq T^{(n)}, \quad \forall t \in \mathcal{T}, \quad (2)$$

Due to the limited battery life of a mobile device, for any allocation that requires usage of a user's mobile battery, the battery must have sufficient charge to satisfy the application demands for the duration of its usage. We model this in (3). For each user-equipment allocation, we sum their battery demand for the duration of their stay across the set  $\mathcal{T}_{i\omega}$ , which is the subset of  $\mathcal{T}$  containing the time periods that user  $i$  requires equipment for, under scenario  $\omega$ .

$$\sum_{t \in \mathcal{T}_{i\omega}} B_j D_i^{(b)}(\omega) x_{ijt}(\omega) \leq T_i^{(b)}(\omega), \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \quad (3)$$

A physical location has limited equipment capacity. We ensure that the number of users allocated to each equipment class does not exceed the amount of available equipment of that class in a given time slot, given in (4).

$$\sum_{i \in \mathcal{I}} x_{ijt}(\omega) \leq T_j, \quad \forall j \in \mathcal{J}, \forall t \in \mathcal{T}, \quad (4)$$

A user's demand in a given time period is either 1 or 0 - they either want equipment or do not. The demand for a user must be met either by an equipment allocation or by a rejection of the user. The number of users allocated to equipment cannot be greater than the actual user demand (given in (5)), and any unallocated users must be accounted for, they cannot be ignored, this is defined in (6).

$$\sum_{j \in \mathcal{J}} x_{ijt}(\omega) \leq D_{it}(\omega), \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \quad (5)$$

$$(z_{it}(\omega) + \sum_{j \in \mathcal{J}} x_{ijt}(\omega)) \geq D_{it}(\omega), \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \quad (6)$$

Not all applications are suitable or compatible with all equipment. A desktop computer may be valid for all applications, whilst a projector may only be applicable to certain uses. To prevent allocation of users to equipment that is not consistent with their chosen application, we use a binary parameter,  $F_{ij}(\omega)$  to limit the allocation of user to equipment under a given scenario. Since the decision variable  $x_{ijt}(\omega)$  is binary, if  $F_{ij}(\omega)$  is 0, the constraint (7) ensures that  $x_{ijt}(\omega)$  will also be 0.

$$x_{ijt}(\omega) \leq F_{ij}(\omega), \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall t \in \mathcal{T}, \quad (7)$$

Finally, we must restrict the range of the decision variables. A user can either be allocated to equipment or not, thus we use binary variables to indicate the allocation of users to equipment and if a user has been rejected, as shown in (8)

$$x_{ijt}(\omega), z_{it}(\omega) \in \{0, 1\}. \quad (8)$$

To find the optimal user-equipment allocation for Case 1, we use a stochastic optimization to find the maximum expected profit, (1), subject to constraints (2)-(8). Stochastic optimization problems are not straightforward to solve. However, it is reasonable to assume that the problem has finite support. We can formulate the stochastic optimization as a deterministic equivalent. Each constraint is evaluated for all members of  $\Omega$ , with the cost function summed across  $\Omega$ , weighted by the probability of each scenario. This results in a binary

linear program, which can be solved using a branch-and-bound algorithm.

## V. CASE 2: MULTIPLE LOCATIONS

In the second case, a user can migrate application between locations. This is not seamless, rather we assume that migration time is included within a time period, but incurs a battery penalty as the user has to use their mobile device to work while moving. A service provider manages inter-location allocation interacting with the location administrators to maximize total profit across the whole system. Location migration is preferable to rejecting users but still creates a negative user experience. Therefore, we introduce a penalty price, similar to the rejected user penalty but smaller, as migration is preferable to rejection. Total cost is the expected sum of net profit across all time periods, locations and equipment, as given in (9).

$$\begin{aligned} \mathcal{Q}(\omega) = & \max \mathbb{E}_{\Omega} \left[ \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} (C_{jp} x_{ijpt}(\omega) \right. \\ & \left. - C_i z_{it}(\omega) - C_i^{(m)} m_{it}(\omega)) \right], \end{aligned} \quad (9)$$

The Case 1 network capacity constraint, (2), is mirrored as expressed in (10), but is additionally specified to constrain each location, since they do not share a network. Instead, they operate independently of each other. Applications migrating to a location must not overload that location's network.

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} D_i^{(n)}(\omega) x_{ijpt}(\omega) \leq T^{(n)}, \quad \forall p \in \mathcal{P}, \forall t \in \mathcal{T}, \quad (10)$$

We simulate battery consumption for each user-equipment allocation in the same way as (3). However, we must track a user's battery consumption as they move between locations. Since each user is unique, we can find the sum of their battery consumption across all locations. If migration occurs, a penalty,  $D_i^{(m)}(\omega)$ , is imposed, representing the amount of battery used when a user  $i$  physically migrates. This adds to the battery consumption, which must be less than the user's mobile battery capacity. This constraint is given in (11).

$$\begin{aligned} \sum_{t \in \mathcal{T}_{i\omega}} \left( \sum_{p \in \mathcal{P}} (B_j D_i^{(b)}(\omega) x_{ijpt}(\omega)) + D_i^{(m)}(\omega) m_{it}(\omega) \right) \\ \leq T_i^{(b)}(\omega), \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \end{aligned} \quad (11)$$

Offloading equipment is maintained by each location, with equipment remaining stationary. (12) mirrors (4) in Case 1, but must be calculated for each location separately.

$$\sum_{i \in \mathcal{I}} x_{ijpt}(\omega) \leq T_{jp}, \quad \forall j \in \mathcal{J}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T}, \quad (12)$$

Demand is met in the same fashion as Case 1, but must be met across all locations, as shown in (13)- (14). This ensures that the total demand for the whole system is satisfied.

$$\sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{P}} x_{ijpt}(\omega) \leq \sum_{p \in \mathcal{P}} D_{ipt}(\omega), \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \quad (13)$$

$$\left( z_{it}(\omega) + \sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{P}} x_{ijpt}(\omega) \right) \geq \sum_{p \in \mathcal{P}} D_{ipt}(\omega), \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \quad (14)$$

Demand for each individual user at a location must be met by either migration, assignment or rejection, as in (15). This complements the previous constraint, which ensured that the demand is met across the whole system. Thus all demand is satisfied, but any demand not satisfied at a user's arrival location incurs the migration penalty.

$$\begin{aligned} m_{it}(\omega) + z_{it}(\omega) + \sum_{j \in \mathcal{J}} x_{ijpt}(\omega) \geq D_{ipt}(\omega), \\ \forall i \in \mathcal{I}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T}, \end{aligned} \quad (15)$$

(16) performs the same function as (7). Unlike Case 1, compatibility must be checked at each location - a user cannot be migrated only to discover that the equipment available at the other location is not compatible.

$$x_{ijpt}(\omega) \leq F_{ij}(\omega), \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T}, \quad (16)$$

We constrain  $x_{ijpt}(\omega)$  and  $z_{it}(\omega)$  similarly to (8).  $m_{it}(\omega)$  is also constrained to be a binary variable in (17), as a user cannot partially migrate, in the same way that a user cannot be partially rejected.

$$x_{ijpt}(\omega), z_{it}(\omega), m_{it}(\omega) \in \{0, 1\}. \quad (17)$$

Case 2 can also be formulated as a stochastic optimization problem, as laid out in equations (9)-(17). This optimization will maximize profit across multiple locations, even if one location has a smaller amount of equipment. Similarly to Case 1, we can reformulate Case 2 as a deterministic binary linear program, and solve with the same method. It is important to define user demand carefully for Case 2, to ensure that a user does not appear at multiple locations, as this is not explicitly precluded in the formulation.

## VI. PERFORMANCE EVALUATION

### A. Parameter Settings

To test our formulations, we evaluate a moderately sized Internet cafe. The evaluation considers 4 periods. There are four different equipment classes, two of which are mains-powered and two of which are not. Users may require equipment for either one period or two periods, and have a choice of three different applications - either productivity, video streaming, or online communication. The wireless LAN has a total capacity of 300Mbps, achieved by a few access points combined together. Each instance of the productivity application requires 256kbps bandwidth, the video application requires 1500kbps (based on Netflix recommended bandwidth [8]) and the communication application requires 500kbps - based on Skype bandwidth estimates [9]. Equipment prices are based on prevailing Internet cafe rates in Singapore [10], with J1 returning \$2 profit, and the remaining three equipment

classes gaining \$1 per period. Battery charge levels and battery demand are modeled as decimals where 1.00 is a full battery. Applications  $K1, K2$  and  $K3$  use 0.10, 0.20 and 0.30 battery per time period, respectively, while user battery levels may be 0.25, 0.50, 0.75, or 1.00. Users for each combination of application, duration and battery level are unique, only appearing once during the entire 4 periods, with the number of unique users fitting each scenario in each time period being either 15, 30 or 40. The maximum number of users per time period is 40, as users fitting one profile will not reappear with a different set of requirements. Demand settings are synthesized from survey data on the use of a Malaysian Internet cafe [11]. Compatibility is chosen according to the power of the equipment and the demands of the applications.

1) *Case 1:* In Case 1, the default amount of equipment for  $J1, J2, J3$  and  $J4$  are 30, 20, 30 and 20, respectively. This is also applicable to Case 2.

2) *Case 2:* In Case 2, there are three equidistant locations. The amount of available equipment at each is identical, with user demand divided across each location - a user may arrive at any of the locations, although only one location at a time. However, the wireless LAN capacity varies between locations with  $P1, P2$  and  $P3$  having capacities 300, 600 and 450 Mbps, respectively. Travelling between locations drains 10% of the battery lost in one time period by the required application.

### B. Numerical Results

As long as a stochastic optimization has finite support, a deterministic equivalent can be formulated and solved as a linear, mixed integer, or binary integer program.

1) *Case 1: Effect of the amount of available equipment on lost users:* We set the amount of available equipment to 30 and, varying each class in turn, examine the effect on the expected number of lost users. The results, given in Fig. 2, are as expected. Increasing the amount of available equipment reduces the number of lost users. The differing steepness of the lines is primarily caused by the compatibility settings.  $J1$  is compatible with all application types, and is also the most profitable, thus it is prioritised, with reduction in the amount of  $J1$  equipment affecting the number of lost users most. Conversely,  $J3$  is only compatible with one application type, and therefore has the least impact. Equipment class  $J2$ 's larger effect than  $J4$  is most likely caused by the fact that  $J2$  does not consume battery resources, making  $J2$  more desirable, as those users assigned to it are those with lower battery levels that may not last without offloading. The graphs tend to level out at the upper end of the curve, this is likely also due to compatibility, as once all applications of a certain type are covered, the users lost from incompatible applications cannot be helped by further increasing that equipment class.

2) *Case 1: Effect of changing battery level on lost users:* We also examined the effect of changing the battery level on the number of lost users. We specify four battery level scenarios, with the highest and lowest having lower probabilities and the middle values having higher probabilities. We set all the battery scenario levels to 0.5 and modified one scenario across

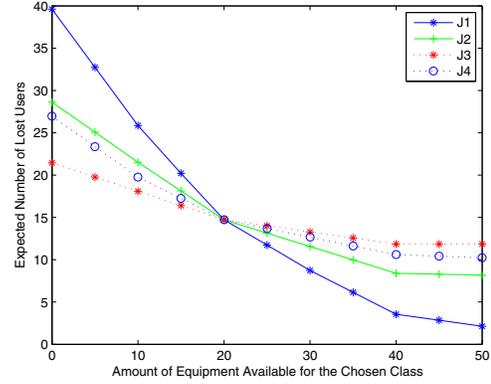


Fig. 2. Variation in lost users by modifying the quantity of each equipment class in turn.

the range of charges from empty to full in turn. The results for the battery scenarios with identical probabilities yield the same shape on the graph, shown in Fig. 3. After a battery level of 0.6, however, both graphs become constant. This is due to the parameter settings, as the maximum battery consumption from an application is 0.3 and the maximum length is 2 periods. Hence, after this, any application can be offloaded to any equipment regardless of whether battery is required or not. Each curve is stepped as there is a finite number of battery demand options, and any battery levels between these demand values will yield the same number of lost users.

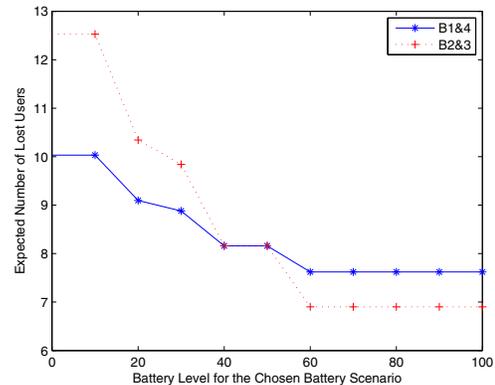


Fig. 3. Variation in lost users by modifying the battery charge level of users in each scenario.

3) *Case 1: Comparison with a greedy approach:* To evaluate the benefits of stochastic optimization, we compare its performance to a greedy allocation of users to equipment on arrival, without any consideration of future user demand. To clearly see the result, we use only equipment class  $J1$  and set its quantity to 30, ensuring that some users are at risk of being lost. We use a two-period scenario, and vary the percentage of users who require the full two periods, rather than one. The difference in profits is shown in Fig. 4. When no users require the full time, the results are identical, as each decision has no impact on later periods. However, as the proportion increases,

the stochastic optimization is able to choose users to maximize profits across both time periods, whilst the greedy approach does not. The profits from each approach rapidly diverge, showing the importance of using stochastic optimization over a simple greedy approach.

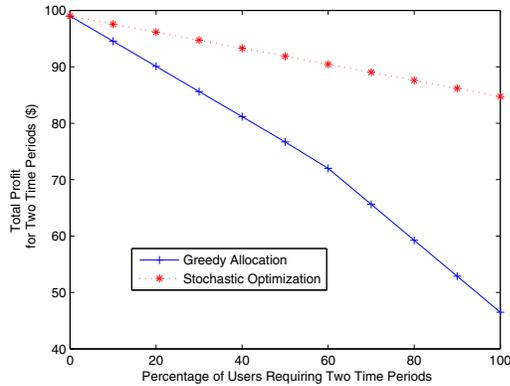


Fig. 4. Demonstration of the value of stochastic optimization over a simple greedy approach.

#### 4) Case 2: Effect of equipment availability on migration:

The key feature of Case 2 is the ability to migrate unallocated demand between locations. To test this scenario, we set the amount of equipment available in each class to 50 for locations  $P2$  and  $P3$ . The amount of equipment available in each class at  $P1$  is then varied to observe the migratory behaviour. In Fig. 5, we show the curve for  $P2$ , as the migrated demand is split evenly between  $P2$  and  $P3$ . This could be altered by setting different migration penalties or the amount of equipment for each provider, an extension we leave for future work. The curves converge, as the amount of equipment at  $P1$  increases and the migrations decrease. This is expected, although it is interesting to observe that the graph is curved, rather than linear. This is likely due to a combination of different probabilities between application types, with the least compatible also being the least frequent. As the amount of equipment increases, the application demands can be managed within a location, rather than migrating to other locations.

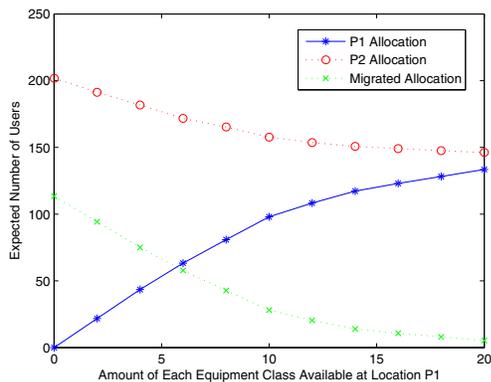


Fig. 5. Effect of varying the amount of available equipment on user migration.

## VII. CONCLUSION

In this paper we have presented a novel scenario in which users can offload applications in a mobile cloud computing setting, to powerful, mains-powered equipment such as PC, smart-TV, or projector. We maximize profit to the service provider, and minimize lost users. We have introduced a second scenario, with multiple locations, using physical migration of users to avoid rejections. The stochastic optimization formulations optimally solve each case, accounting for uncertainties such as the type, quantity, and duration of demand. We have presented results that accentuate the benefits of our methods, showing the effectiveness of stochastic optimization over an alternative, and the value of migration in a combined solution.

Future work would aim to increase the sophistication of the model. Improvements could include the creation of a larger, unified solution, consideration of travel delays when migrating users, bandwidth uncertainty and routing, users changing application requirements during their session, and an expansion to include cloud-based resource offloading.

This research area is a logical next step, given mobile device popularity, and the offering of offloading resources in a mobile cloud computing setting is an opportunity not to be ignored.

## ACKNOWLEDGEMENTS

This work was supported in part by Singapore MOE Tier 1 (RG18/13 and RG33/12).

## REFERENCES

- [1] (2015) App and desktop virtualization. [Online]. Available: <http://www.citrix.com/solutions/desktop-virtualization/overview.html>
- [2] V. Schuchardt, "Moving mobile applications between mobile devices seamlessly," in *Software Engineering (ICSE), 2012 34th International Conference on*, June 2012, pp. 1595–1598.
- [3] R. Ma and C.-L. Wang, "Lightweight application-level task migration for mobile cloud computing," in *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, March 2012, pp. 550–557.
- [4] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud," in *Proceedings of the Sixth Conference on Computer Systems*, ser. EuroSys '11. New York, NY, USA: ACM, 2011, pp. 301–314. [Online]. Available: <http://doi.acm.org/10.1145/1966445.1966473>
- [5] Y. Jiang, J. He, Q. Li, and X. Xiao, "A dynamic execution offloading model for efficient mobile cloud computing," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, Dec 2014, pp. 2302–2307.
- [6] M. Zbierski and P. Makosiej, "Bring the cloud to your mobile: Transparent offloading of html5 web workers," in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, Dec 2014, pp. 198–203.
- [7] Y. Zhang, R. Yang, T. Wo, C. Hu, J. Kang, and L. Cui, "Cloudap: Improving the qos of mobile applications with efficient vm migration," in *High Performance Computing and Communications, Embedded and Ubiquitous Computing (HPCC\_EUC), 2013 IEEE 10th International Conference on*, Nov 2013, pp. 1374–1381.
- [8] (2015) Internet connection speed recommendations. [Online]. Available: <https://help.netflix.com/en/node/306>
- [9] (2015) How much bandwidth does skype need? [Online]. Available: <https://support.skype.com/en/faq/fa1417/how-much-bandwidth-does-skype-need>
- [10] (2015) Internet cafes in singapore. [Online]. Available: <http://www.world66.com/asia/southeastasia/singapore/internetcafes>
- [11] S. S. Alam, Z. Abdullah, and N. Ahsan, "Cyber cafe usage in malaysia: An exploratory study," *The Journal of Internet Banking and Commerce*, vol. 14, no. 1, pp. 1–13, 2009.