This is the Pre-Published Version.

The following publication Y. H. Ho, P. M. F. Ho and H. C. B. Chan, "Mobile Intercloud System and Objects Transfer Mechanism," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-6 is available at https://doi.org/10.1109/GLOCOM.2017.8254706.

Mobile Intercloud System and Objects Transfer Mechanism

Yik Him Ho, Peter Man Fai Ho, Henry C. B. Chan
Department of Computing
The Hong Kong Polytechnic University

Abstract—**In recent years, there has been considerable interest in studying intercloud with the aim of supporting interactions among different clouds, possibly managed by different service providers. The extension of intercloud to a mobile environment, called a mobile intercloud system is a relatively new research area. In this paper, we present a mobile intercloud system, which is developed based on the IEEE P2302 Intercloud architecture and inspired by the mobile Internet protocol. Basically a mobile terminal is associated with a home cloud in which a virtual mobile terminal can be set up together with the mobile applications and data. When a mobile terminal moves to a new area and joins a foreign cloud, certain mobile applications and data can be transferred to the foreign cloud. For data/files, we present a rule-based policy so that users can specify the rules for transferring data/files flexibly. A prototype has been developed to demonstrate the basic concept. For applications, we formulate a Markov decision process model to study how the applications should be transferred to minimize the overall cost. Analytical results are presented to provide valuable insights into the design of the mobile intercloud system.**

I. INTRODUCTION

With the advance of cloud computing and smartphones, there has also been considerable interest in mobile cloud computing, which extends cloud computing to mobile terminals. In general, mobile cloud computing can be classified into three service models [1]: receiving service from a cloud, serving as a service provider, and functioning as a service broker. Computation offloading is a major function of mobile cloud computing [2]. Due to the limitations of mobile terminals, it is desirable to offload certain computations and/or programs to a cloud. For instance, according to [3], energy can be saved by means of computation offloading, which depends on the link bandwidth and other factors. Two case studies can be found in [4] to illustrate the advantages of computation offloading. [5] presents an offloading system using a genetic algorithm to make offloading decisions with the aim of minimizing processing time and power consumption of the mobile terminals.

In general, there are five main cloud computing-related standards to support various cloud operation [6]: Open Virtualization Format by Distributed Management Task Force, Cloud Data Management Interface standard by Storage Networking Industry Association, Open Cloud Computing Interface standard by Open Grid Forum, ID Cloud standard by the Organization for the Advancement of Structured Information Standards, and P2301 and P2302 by IEEE. Our work is based on the IEEE P2302 draft standard, a draft standard for intercloud interoperability and federation. The aim is to study how to support mobile cloud computing over an intercloud system.

An overview of intercloud systems can be found in [7] and [8]. Generally speaking, there are three approaches to creating an intercloud system: Internet-like, new framework, and overlay. P2302 is based on an Internet-like framework. That means, it seeks to develop an intercloud framework based on the Internet model (i.e., cloud of clouds inspired by the network of networks, with similar architecture and protocols). In [9], a new framework called the Intercloud Architecture Framework is proposed for realizing intercloud. Research on intercloud is still relatively new. The fundamental problem is to study how to support collaboration over heterogeneous clouds, possibly run by different cloud providers with different cloud platforms.

Extending intercloud to a mobile environment poses new research challenges. We call this intercloud system the Mobile Intercloud System. To the best of our knowledge, there has been little research conducted on the mobile intercloud. For example, in a mobile intercloud system, not only can mobile terminals be moved, but so can virtual terminals, data and applications (i.e., different types of handoff can be possible). More importantly, for mobile intercloud, heterogeneous clouds (i.e., clouds using different systems) can collaborate in a mobile environment. A similar migration scenario has been studied in [10] such that a mobile user can be served by a local cloud and a backend cloud by transferring certain resources as well as tasks. However, the scenario is based on intracloud rather than intercloud (i.e., supported by a single cloud provider). The mobile intercloud system has a wider scope and is more flexible. As discussed later, the mobile intercloud system is also inspired by the mobile Internet protocol (i.e., mobile IP [11]). Note that while mobile cloud computing focuses on computation offloading in general, mobile intercloud seeks to support cloud computing services through collaboration among heterogeneous clouds in a mobile environment. In other words, it seeks to support inter-operations rather than intra-operations. Note that the migration processes of mobile intercloud are more complex because they are conducted over heterogeneous clouds and cover virtual terminals, data and applications. In this paper, we present the basic concept of mobile intercloud and study how data/files and applications (referred to as objects in general) can be transferred. For data/files, a rule-based approach is studied. For applications, a Markov decision process is formulated to determine the transfer policy for supporting mobile intercloud.

This rest of the paper is organized as follows. Section II presents the overall architecture of our system. Section III introduces the rule-based approach for moving files between clouds. We demonstrate the ability to transfer files over different clouds through the control of a smartphone application on Android devices. Section IV introduces the use of the Markov decision process to find optimum policies when deciding the migration of applications. Section V provides an analysis of the use of a Markov decision model in different situations. Section VI concludes our work.

II. SYSTEM ARCHITECTURE

Fig. 1 shows the basic concept of the mobile intercloud system. Every user owns a mobile device, called a physical mobile terminal. They also own a virtual terminal on the cloud. The virtual terminal cloud could store more contents than the physical terminal, for example, files and applications. Note that the virtual terminal usually has higher processing power and storage capacity. Because the physical mobile terminal has limited storage, not all contents are downloaded onto the device. The contents will be stored on the virtual terminal and transferred to the mobile terminal only when required. Also, some processing intensive tasks could be offloaded to the virtual terminal, in order to save battery life of the mobile terminal, and in some cases the response time will also be shortened.

We categorize the contents of the virtual terminal into data/files and applications, which are referred to as objects in general. When users want to access files, for example a text document and a spreadsheet, these need to be transferred and downloaded completely in order to view or use them. When users want to use an application, they do not need to download the entire application to the mobile terminal, but instead can send the request and input to the virtual terminal and wait for the response. The main applications are stored in the cloud and the processing job will be finished by the virtual terminal. A web service or web API is a similar example.

In order to increase the response time, the virtual terminal will be located as close as possible to the physical terminal. We call the cloud storing in the virtual terminal the home cloud. Consider the case when a user travels to a foreign area; the waiting time when accessing the content will be increased because it is physically located further away. If a closer cloud is available, the waiting time to access the content will be lower if that content is moved to the closer location, namely a foreign cloud.

To minimize cost and to increase the response time for accessing objects, the user can decide to move the entire virtual terminal to the foreign cloud. However, if some of the files and applications are rarely used, moving these to a foreign cloud might be wasteful. Therefore, we have proposed two approaches when deciding which content should be moved to a foreign cloud. We present a rule-based approach for moving files and a threshold-based approach based on a Markov decision process model for moving applications.

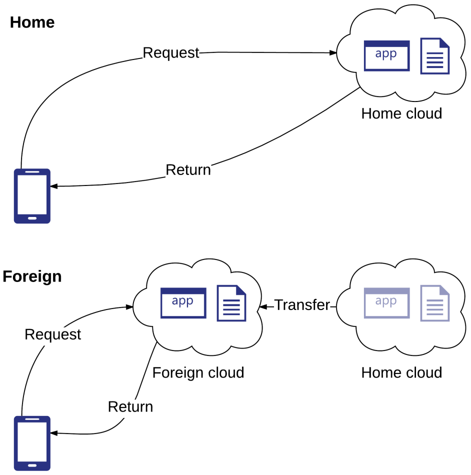


Fig. 1: Basic concept of the mobile intercloud system

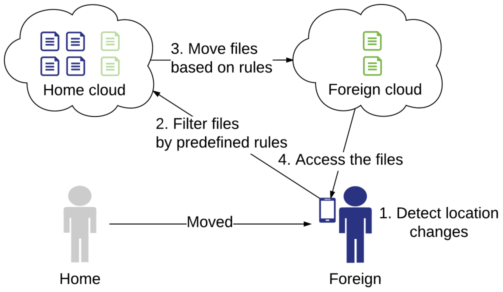


Fig. 2: The flow of moving files

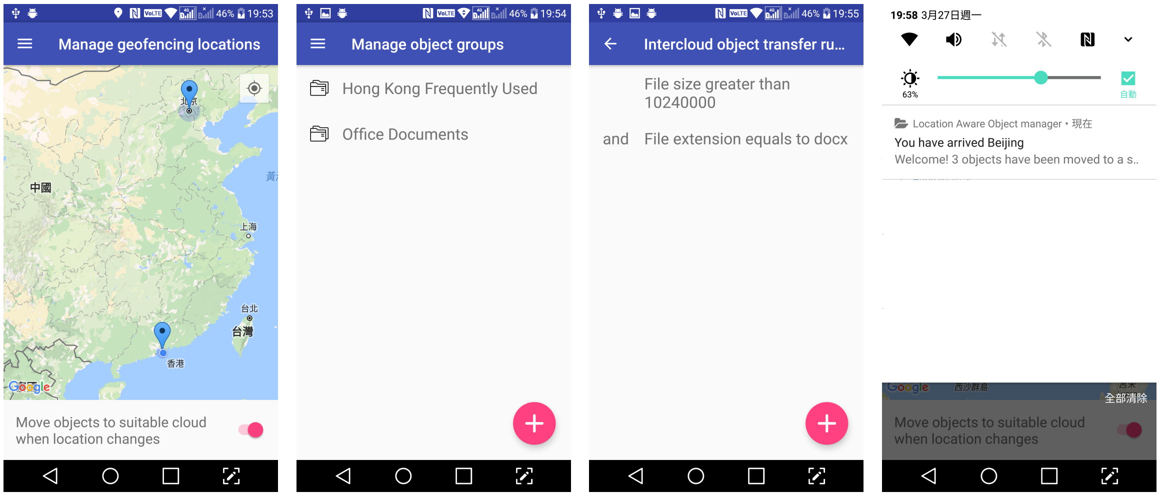
III. RULES BASED APPROACH FOR MOVING FILES

When users travel to a foreign area, they can choose to move which files to a closer virtual terminal according to their preferences. Users can decide this based on file type, file size and frequency of access.

Fig. 2 shows the general flow when moving files between clouds. When a user moves from one location to another, the mobile terminal detects a location change. This can be done by using the Global Position System (GPS) and geofencing technology. The location change triggers an inter-cloud file transfer process that moves necessary files to a foreign cloud. These necessary files include all files from the home cloud, but filtered by a set of rules, which are defined by the user prior to the transfer process. Any files that pass the rule filter are transferred to the foreign cloud.

To allow users to define different rules for different sets of files, we organize files into different groups, and each group is associated with one or multiple rules. Rules can be defined based on file attribute, file status, file access pattern, user location or user status. In this paper, we propose four kinds of rules to illustrate our approach as follows:

- File byte size (e.g. move the files that size is larger than 20 megabytes)



Move objects to suitable cloud when location changes

File size greater than 10240000

and File extension equals to docx

Location Aware Object manager • 現在

You have arrived Beijing

Welcome! 3 objects have been moved to a s..

(a) Geofences to monitor location change events (b) Group objects by folder (c) Defining rules (d) Files being moved after location changed

Fig. 3: Demonstration of mobile intercloud migration using an Android application

- File extension type (e.g. move the files with extension of jpg/pdf/ppt)
- Per week access frequency (e.g. move the files that the user has accessed more than 10 times per week)
- Current user location (e.g. move the selected files when the user is in London)

The following illustrates an example. Assume that there are two clouds on the network, A and B. Cloud A and cloud B have a shorter network latency to Hong Kong and Beijing respectively. A user organizes 10 text files into a group called text documents. Among these files, six of them are stored inside cloud A and the remainder are stored inside cloud B. The user also defines the following rules within the group:

- Transfer the file if it is larger than 10 megabytes
- Transfer the file if it is a text file (i.e. file extension type is txt)

At the beginning of the day, the user is geographically located in Hong Kong and plans to travel to Beijing. When the user arrives in Beijing, their mobile terminal detects a location change and triggers the file transfer process. The process executes the rules defined in the text documents group, and determined there are five text files larger than 10 megabytes within that group, with three of those files stored in a home cloud (i.e. cloud A, which has a longer network latency to Beijing than cloud B). Finally, the system automatically transfers those three files from the home cloud to the foreign cloud (i.e. cloud A to cloud B).

The rule-based model has been implemented on the intercloud testbed with an Android smartphone. It is capable of transferring bucket objects between different clouds, including Amazon Web Services (AWS), Microsoft Azure and Minio, an open-sourced distributed object storage system. Fig. 3a illustrates two geofences to monitor location change events in the application. Fig. 3b and Fig. 3c show two user interfaces for managing groups and rules. Fig. 3d demonstrates that the file transfer process is done in the background and the user receives a notification once the transfer process is completed.

Based on the rules, necessary files are able to migrate between clouds when a closer cloud with a lower latency is available. The policies and migration abilities not only work between the same cloud platform, but also in an intercloud environment, i.e., between different cloud platform and service providers.

IV. MARKOV DECISION PROCESS FOR MOVING APPLICATIONS

In order to minimize the overall cost when accessing an application, one cannot always simply transfer the application to the foreign (closer) cloud when the user travels. While the size of application is large, the transfer cost is actually a huge cost for the network. If the user plans to return home after a short period of time, or seldom uses these applications, moving them to a foreign cloud would be wasteful.

For example, when a user is at a foreign location, the cost of using the application in the home cloud is five units, while in a foreign cloud, it is two. The moving cost is ten. Even the access cost at a foreign cloud is lower, if users only use the application once when they are at a foreign location, the cost of moving the application is larger than using it on the home cloud $(10+2 > 5)$. The situation is more complicated because the optimum decision would be affected by the probability of the user using the application, the cost of moving the application, the cost of using the application at a different location, and the period the user will stay at the foreign location.

To decide whether the application should be moved when the user travels to a foreign region, we formulate a Markov decision model by following the methodology/notations in [12]. We assume that only two clouds exist (home and

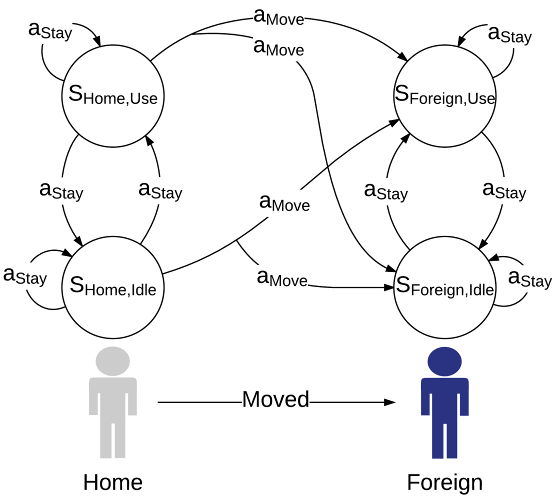


Fig. 4: Symbolic representation of Markov decision process

foreign). A decision is performed at the start of each epoch. The use of an application is carried out after the action is made. For each epoch, there will only be a maximum of one usage. We use a discrete time system to represent the situation, for example at time 1,2,3 to n where n is a finite number. In other words, this is a finite-horizon discrete time Markov decision process.

The symbolic representation of the model is shown in Fig. 4. We define the location and status of the application as the state. The application can either be located at the home cloud or at the foreign cloud. No matter where the application is located, the user may use the application or let it idle each time. As there are two clouds, there will be four states:

$$S = \{s_{h,u}, s_{h,i}, s_{f,u}, s_{f,i}\} \tag{1}$$

where $s_{h,u}$ indicate the application is at home and being used, $s_{h,i}$ means home and idle, $s_{f,u}$ means foreign and use, $s_{f,i}$ means foreign and idle.

Actions of the system includes moving the application from home cloud to foreign cloud, retaining the application at the home cloud, or retaining the application at the foreign cloud, provided that it has been moved at a previous time.

$$A_s = \begin{cases} \{a_{move}, a_{stay}\}, if state = s_{h,u} or s_{h,i} \\ \{a_{stay}\}, if state = s_{f,u} or s_{f,i} \end{cases} \tag{2}$$

Note that the application will only be moved to the foreign cloud if it is at home, while there is no point in moving the application from the foreign cloud to the home cloud (which is further away) if the user is in a foreign area.

The cost of using the application is affected by the network speed between mobile device and cloud, the network speed between clouds, input and output size, and the application size on the cloud. We assume the processing time of the clouds are the same. The flow and related cost of staying and moving action is shown in Fig. 5.

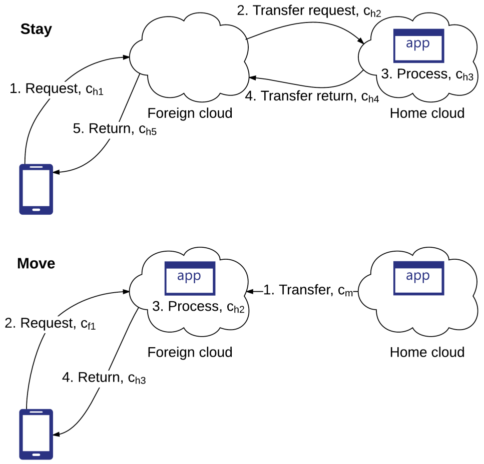


Fig. 5: Flow of stay and move situation, and related cost c

Here, we define the following costs for the performance analysis:

- c_h: cost of using the application at home cloud
- c_f: cost of using the application at foreign cloud
- c_m: moving cost of application from home cloud to foreign cloud

Note that the costs have the properties of $c_m > c_h \geq c_f$, as the cost of accessing the application is the lowest when the user and application both locate in a foreign region. The one-time moving cost would be the largest.

The cost of using the application at time t is defined as:

$$Cost^t(s, a) \tag{3}$$

Note that this cost depends on the states and actions as shown below:

$$Cost^t(s, a_{stay}) = \begin{array}{c|cccc} & s_{h,u} & s_{h,i} & s_{f,u} & s_{f,i} \\ \hline s_{h,u} & c_h & 0 & 0 & 0 \\ s_{h,i} & c_h & 0 & 0 & 0 \\ s_{f,u} & 0 & 0 & c_f & 0 \\ s_{f,i} & 0 & 0 & c_f & 0 \end{array}$$

$$Cost^t(s, a_{move}) = \begin{array}{c|cccc} & s_{h,u} & s_{h,i} & s_{f,u} & s_{f,i} \\ \hline s_{h,u} & 0 & 0 & c_m + c_f & 0 \\ s_{h,i} & 0 & 0 & c_m + c_f & 0 \\ s_{f,u} & 0 & 0 & c_f & 0 \\ s_{f,i} & 0 & 0 & c_f & 0 \end{array}$$

The transition probability from current state s^t at time t to next time $t+1$: s^{t+1} is

$$P(s^{t+1}|s, a) \tag{4}$$

Let the probability of the user using the application be β, the transition matrix is defined as follows:

$$P(s^{t+1}|s, a_{move}) = \begin{array}{c c} & \begin{array}{cccc} s_{h,u} & s_{h,i} & s_{f,u} & s_{f,i} \end{array} \\ \begin{array}{c} s_{h,u} \\ s_{h,i} \\ s_{f,u} \\ s_{f,i} \end{array} & \begin{pmatrix} 0 & 0 & \beta & 1-\beta \\ 0 & 0 & \beta & 1-\beta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$P(s^{t+1}|s, a_{stay}) = \begin{array}{c c} & \begin{array}{cccc} s_{h,u} & s_{h,i} & s_{f,u} & s_{f,i} \end{array} \\ \begin{array}{c} s_{h,u} \\ s_{h,i} \\ s_{f,u} \\ s_{f,i} \end{array} & \begin{pmatrix} \beta & 1-\beta & 0 & 0 \\ \beta & 1-\beta & 0 & 0 \\ 0 & 0 & \beta & 1-\beta \\ 0 & 0 & \beta & 1-\beta \end{pmatrix} \end{array}$$

Let $\theta^t(s)$ be the minimum expected accumulated cost for using the application when the state is s at time t. The minimum cost is expressed as follows:

$$\theta^t(s) = min\Big\{Cost^t(s,a) + \sum_{s\in S} P(s^{t+1}|s,a)\cdot\theta^{t+1}(s^{t+1})\Big\}$$

Because it is a finite-horizon situation, the minimum expected cost can be calculated by backward induction algorithm. The action associated with the minimum cost can also be found.

For example, when the user is currently located in a foreign region, let the cost of using the application at home cloud c_h be 8 and foreign cloud be c_f be 2. Let the one-time moving cost of transferring the application c_m from home to cloud be 10. Let the probability of the user using the application be 0.8. The set of states is the same as in equation 1: $S_t = \{s_{h,u}, s_{h,i}, s_{f,u}, s_{f,i}\}$ for $t = 1, 2, ..., n$.

The minimum expected cost is:

$$\theta^t(s,a) = min\Big\{Cost^t(s,a) + P(s^{t+1}|s,a)\cdot\theta^{t+1}\Big\} \tag{5}$$

Let the period of the user staying in a foreign area be 3, i.e. $t = 1$ to 3, by backward induction algorithm we found that the best decision to move is when $t = 1$ if there are only 3 epochs, otherwise 'stay' is preferred, as the high moving cost cannot be compensated for in a short period of time.

V. ANALYSIS

The best decision depends on three factors, namely (1) the difference between c_h, c_f and c_h, (2) the probability of using the application and (3) length of the period staying at the foreign. We compare the result between decisions using a Markov decision process model with a always move and always stay policy. In the following analysis, home cost c_h, and moving cost c_m is normalized into a ratio of c_f, that is c_f:c_h:c_m. For example, if $c_f = 3, c_h = 8, c_m = 20$, it is presented as $1 : 2.67 : 6.67$.

Given the example of probability of using the application is 0.8, Fig. 6 shows the preferred action at different epochs. The shading area is the condition that moving is preferred, and the curve is the threshold value. Note that the y-axis is the ratio between normalized moving cost and normalized home cost $\frac{c_m{}'}{c_h{}'}$ where $c_m{}'$ and $c_h{}'$ is the normalized moving cost and normalized home cost respectively. The policy can be seen as: at time t, moving the application to a foreign cloud is preferred if the moving cost is larger than home cost at a certain threshold. For example, at $t = 7$, the ratio is 9.5, indicating the application should be moved if the moving cost is 9.5 times or smaller than the home cost. (e.g, $c_f = 3, c_h = 8, c_m = 76$, $1 : 2.67 : 25.33$)

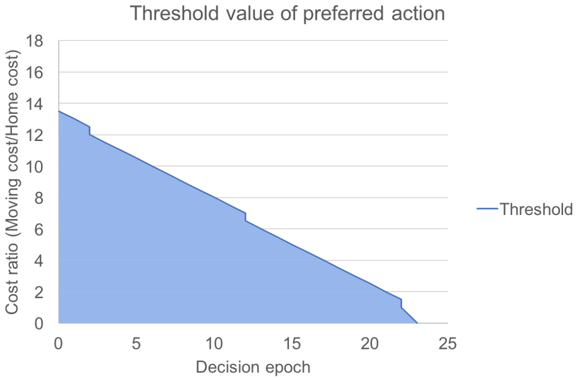
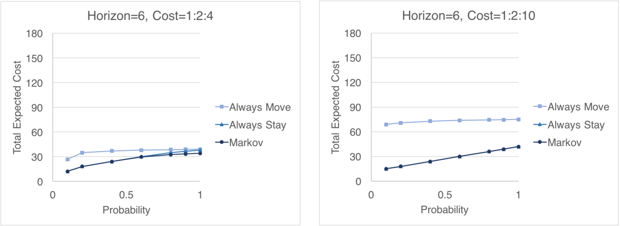
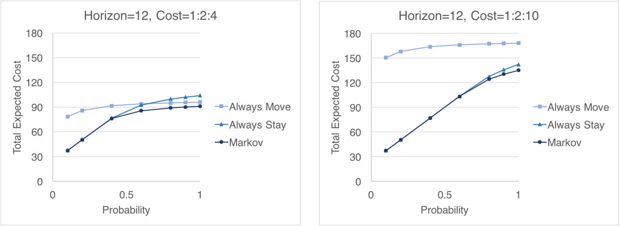


Fig. 6: Preferred action at different epochs



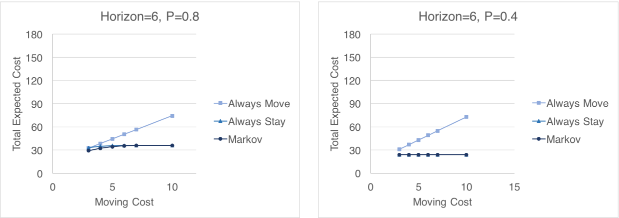
(a) Horizon: 6 time units, cost: 1:2:4 (b) Horizon: 6 time units, cost: 1:2:10



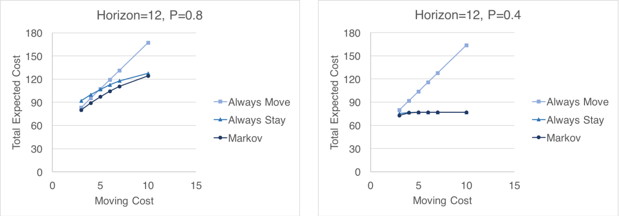
(c) Horizon: 12 time units, cost: 1:2:4 (d) Horizon: 12 time units, cost: 1:2:10

Fig. 7: Total expected cost when probability varies

Fig. 7 shows the relationship between total cost and probability. In general, a 'staying' policy incurs a lower cost than moving. Only if the horizon is long enough, or the probability of using the application is higher or the moving cost is lower (Fig. 7c), moving to a foreign area is worthwhile. With reference to the figures, normally the curve of 'Always Stay' policy is the same as the decision suggested by the Markov decision process. The condition where two curves split is the condition where the model decide to 'move'.

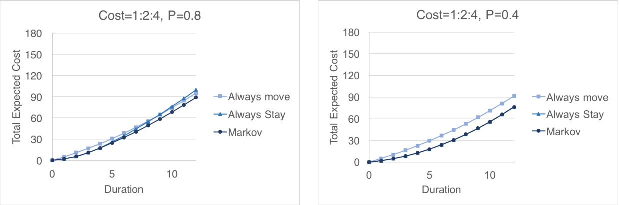


(a) Horizon: 6 time units, probability: 0.8 (b) Horizon: 6 time units, probability: 0.4

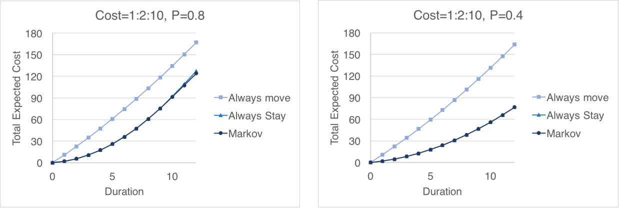


(c) Horizon: 12 time units, probability: 0.8 (d) Horizon: 12 time units, probability: 0.4

Fig. 8: Total expected cost when moving cost varies



(a) Probability is 0.8, cost is 1:2:4 (b) Probability is 0.4, cost is 1:2:4



(c) Probability is 0.8, cost is 1:2:10 (d) Probability is 0.4, cost is 1:2:10

Fig. 9: Total expected cost when duration varies

Fig. 8 shows the relationship between total expected cost and moving cost. Staying is a better approach, when the moving cost is high. It is worthwhile to move only if the probability is high, the horizon is long and the moving cost is lower (e.g., 1:2:5) as shown in Fig. 8c.

Fig. 9 shows the relationship between duration of stay and total cost. Similar to the previous results, staying is a better approach. Fig. 9d shows that if the staying period is larger than 7, there is enough time for the moving cost to be compensated.

To summarize the analysis, in most cases staying is preferred, because moving an application from one cloud to another cloud is costly. If the moving cost is relatively smaller (i.e., below 1:2:5), the probability is larger than 0.6, and the staying period (the length of the horizon) is larger than 7, moving would incur a lower total expected cost. With the aid of a Markov decision prorcess, the system is able to suggest moving under certain conditions, therefore the overall cost would be lowest.

VI. CONCLUSION

In conclusion, we have presented a mobile intercloud system based on the IEEE P2302 draft standard and inspired by mobile IP. We have also studied how objects (i.e., data/files and applications) can be transferred between a home cloud and a foreign cloud. A rule-based approach is proposed and demonstrated for allowing users to design different criteria to move data/files if a location change event is detected by a mobile terminal. A Markov decision process is formulated for the system to determine under which situation it is better to move the application to a foreign cloud so as to minimize the overall cost. In general, a threshold-based policy can be used to transfer the applications so that the overall cost can be minimized. The analytical results should provide valuable insights into the design of the mobile intercloud system.

VII. ACKNOWLEDGMENT

This project is supported by The Hong Kong Polytechnic University under account G-YBAJ.

REFERENCES

[1] D. Huang, T. Xing, and H. Wu, “Mobile cloud computing service models: a user-centric approach,” *IEEE Network*, vol. 27, no. 5, pp. 6–11, September 2013.

[2] Y. Xu and S. Mao, “A survey of mobile cloud computing for rich media applications,” *IEEE Wireless Communications*, vol. 20, no. 3, pp. 46–53, June 2013.

[3] K. Kumar and Y. H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” *Computer*, vol. 43, no. 4, pp. 51–56, April 2010.

[4] X. Ma, Y. Zhao, L. Zhang, H. Wang, and L. Peng, “When mobile terminals meet the cloud: computation offloading as the bridge,” *IEEE Network*, vol. 27, no. 5, pp. 28–33, September 2013.

[5] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, “Computation offloading for service workflow in mobile cloud computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3317–3329, Dec 2015.

[6] S. Ortiz, “The problem with cloud-computing standardization,” *Computer*, vol. 44, no. 7, pp. 13–16, July 2011.

[7] A. N. Toosi, R. N. Calheiros, and R. Buyya, “Interconnected cloud computing environments: Challenges, taxonomy, and survey,” *ACM Comput. Surv.*, vol. 47, no. 1, pp. 7:1–7:47, May 2014.

[8] N. Grozev and R. Buyya, “Inter-cloud architectures and application brokering: taxonomy and survey,” *Software: Practice and Experience*, vol. 44, no. 3, pp. 369–390, 2014. [Online]. Available: http://dx.doi.org/10.1002/spe.2168

[9] Y. Demchenko, M. X. Makkes, R. Strijkers, and C. de Laat, “Intercloud architecture for interoperability and integration,” in *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, Dec 2012, pp. 666–674.

[10] L. Gkatzikis and I. Koutsopoulos, “Migrate or not? exploiting dynamic task migration in mobile cloud computing systems,” *IEEE Wireless Communications*, vol. 20, no. 3, pp. 24–32, June 2013.

[11] C. E. Perkins, “Mobile ip,” *IEEE Communications Magazine*, vol. 35, no. 5, pp. 84–99, May 1997.

[12] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1994.