Resource Orchestration for Multi-task Application in Home-to-home Cloud

Qi Qi, *Member*, IEEE, Jingyu Wang, *Member*, IEEE, Qi Li, Tonghong Li, and Yufei Cao

Abstract —*Along with the increasing capability of consumer electronic devices, it becomes the new trend that the home cloud computing takes household consumer electronic devices as a sharing resource pool, which can bring great convenience to consumers. However, how to find the most suitable resources or services in the home cloud is a challenge for the consumers. When offloading a complicated application to the home cloud, each task of the application can be executed individually with its own computation, storage and bandwidth requirements. The heterogeneous consumer electronic devices that contain different performance metrics affect the destination choice. This paper proposes a distributed home-to-home cloud infrastructure where volunteer users allow the resources of their consumer electronic devices to be shared with the neighboring houses. By extending the home gateways, a decentralized controlling is responsible for resource monitoring, orchestration, task offload and migration. With consideration of load balance and volunteers' opinion, the resource orchestration is formulated as an optimization problem with one objective and multi-constraints. Finally, a Particle Swarm Optimization (PSO)-based algorithm is used to obtain the approximate optimal solution. Simulation results show that the solution for all cases studied almost achieves 90% objective value in acceptable time*[1].

Index Terms — household consumer electronic device, home cloud computing, multi-task application, resource orchestration

I. INTRODUCTION

Nowadays, household consumer electronic devices such as mobile phone, tablet, laptops, TV, cleaner robots and sensors (e.g., temperature, humidity, brightness) are all becoming smart, which always contain their own computation, storage and network modules. The home gateway provides management for these household consumer electronic devices, and provides a home network with various applications [1]. As these consumer electronic devices are rapidly evolving in terms of both hardware and software, resources in these devices are not fully used in most of the time [2]. The technologies such as home network, pervasive computing, and cloud computing influence each other, which results in the emergence of home cloud computing. The goal of the home cloud is exploiting unused computing resources in various devices at home in order to deliver more powerful and novel functionalities to users.

Within a house, the virtualization of home gateway and household consumer electronic devices, as well as the openness services of some applications enable idle resources to be shared in home cloud. In addition, the Internet-connected consumer electronic devices, provided by volunteer homes, can be used as a source of computing power and storage to construct the underlying resource pool in distributed computing paradigms. Then, the applications are not constrained to operate within the consumer's private devices; instead, they are able to run on the shared resources among neighboring homes. The different resources of multiple houses can cooperate with each other, in a manner best suited for the application and current operating constraints [3]. In Fig.1, the home-to-home cloud for enabling convenient, on-demand access to a shared pool of configurable computing resources (e.g. networks, servers, storage, and services) is shown.

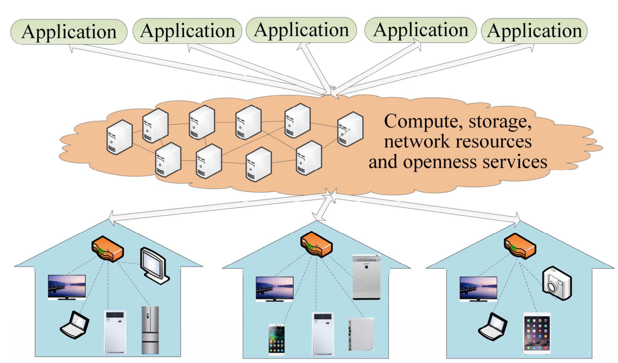


Fig. 1. Home-to-home cloud scenario. The compute, storage and network resource of devices, as well as openness service are provided.

However, there have been no technologies and standards for controlling and managing heterogeneous home-to-home

[1] This work was supported in part by the (1) National Basic Research Program of China (No. 2013CB329102); (2) National Natural Science Foundation of China (No. 61471063, 61421061, 61372120, 61302087, 61271019); (3) Key Project of Chinese Ministry of Education (No. MCM20130310); (4) Beijing Municipal Natural Science Foundation (No. 4152039); (5) Beijing Higher Education Young Elite Teacher Project (No. YETP0473).

Qi Qi is with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications; Beijing, P.R. China, 100876 (e-mail: qiqi8266@bupt.edu.cn).

Jingyu Wang is with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications; Beijing, P.R. China, 100876 (e-mail: wangjingyu@bupt.edu.cn).

Qi Li is with department of computer science, Beijing University of Posts and Telecommunications, Beijing, 100876, P.R. China (e-mail: liqi2001@bupt.edu.cn).

Tonghong Li is with department of computer science, Technical University of Madrid, Madrid, Spain, 28660 (e-mail: tonghong@fi.upm.es).

Yufei Cao is with the EBUPT Company, Beijing, 100191, P.R. China (e-mail: caoyufei@ebupt.com).

Contributed Paper
Manuscript received 03/30/16
Current version published 06/29/16
Electronic version published 06/29/16

cloud resources in a consolidated manner. The resource sharing pool does not scale for various demands from individual houses. Moreover, the consumer's requirements become more complicated, one single service can hardly satisfy such requirements, and multiple tasks may be composed in a complex application to execute a complicated function [3]. When offloading a complex application to the home-to-home cloud, each task's own computation, storage and bandwidth requirements should be considered.

Traditionally, heavy computing applications are usually deployed within one home cloud. Nevertheless, for the complicated application in the home-to-home cloud, the appropriate resource for one of the tasks may be located in another house. When consumers offload tasks to the home-to-home cloud for improving the capability of application, neighboring houses may provide multiple choices for each task. Thus, resource management for the home-to-home cloud requires the consideration of multiple aspects of the resource orchestration, such as device capability, resource availability, cost, and volunteers' option. For the distributed resources that lack the integrated management, how to orchestrate the multiple resources by choosing the appropriate offload destination is a challenging problem.

To make sure that neighboring home resources cooperate with each other, a two-layer home-to-home cloud management architecture in the home gateway is proposed, where the user can utilize the idle resources of consumer electronic devices provided by several volunteer houses. Moreover, for offloading the task to appropriate consumer electronic devices, the resource orchestration problem is formulated and the approximate optimal solution is obtained in acceptable time by Particle Swarm Optimization (PSO)-based algorithm.

In the remaining part of the paper, the related work is first described in Section II. Then the system architectural design is presented in Section III. The resource orchestration problem in the home-to-home cloud and its solution are presented with details in Section IV. Numerical evaluation and real world experiment are presented in Section V. Section VI concludes the paper and presents the future work.

II. Related Work

A. Home Cloud Computing

Along with the development of cloud computing technology, researchers focus on the resource sharing among consumer electronic devices in the home network.

Wei *et al.* [4] work on combining clouds and smart home together by Peer-to-Peer (P2P) technology, where each smart home shares its appliances resources like memory and computing power with others. Son *et al.* [5] provide a secure cloud computing service based on personal virtualization for content sharing among consumers. Lee *et al.* [2] construct a home cloud in order to exploit unused computing resources in various devices at home to deliver more powerful and novel functionalities to users. Takatori *et al*. [1] delegate the home gateway to a cloud for saving the cost of purchasing a home server for a house. They propose a novel cloud-based architecture, i.e. home gateway as a service, for the home network with security isolation, fault isolation and resource isolation. Similarly, to address the resource-constrained limitations, Igarashi *et al.* [6] present a new cloud-enhanced home controller where the resources of the local home controller are augmented with external cloud resources accessed over the network. Then this architecture is scaled to support multi-vendor [7]. Moreover, Lee *et al*. [8] propose an auto-configuration of home networks in home cloud environments using the Software Defined Networking (SDN) controller. The SDN controller can auto-recognize new home devices, and manage home devices according to the connection state, without the middleware to be installed. Also, Jo *et al*. [9] presents in-home consumer electronic devices that incorporate SDN and provide a high degree of flexibility for intra-home networking as well as wider connectivity for inter-home networking. For achieving high-performance home cloud environments, Lee *et al.* [10] propose a novel solid state drives caching scheme and virtual disk image format based on the workload characteristics.

For home services, Fu *et al*. [11] work on the semantic service search in cloud for home network consumers, where the data are encrypted for security. Jeong *et al*. [12] propose a zone-aware service system which utilizes the user's nomadic resource, i.e. the user's hand-held device that has a sensing or networking capability, such as a smartphone, smart watch, and mobile sensor, as part of the smart home service infrastructure. The new system is easily modified, updated, and cost- effectively implemented. Díaz-Sánchez *et al.* [13] present cloud computing middleware Media Cloud for set-top boxes for classifying, searching, and delivering media inside home network and across the cloud. Guo *et al*. [14] solve the intra-datacenter traffic problem in a chunk-based public cloud storage service for smart homes such as delivering content service. The batching smooth intra-datacenter traffic scheme is used to reduce the peak load.

B. Resource Orchestration in Cloud Computing

Resource orchestration is similar to the task scheduling. Tao *et al*. [15] work on the resource service composition in an industry grid system, and exploit the standard particle swarm algorithm to find the solution. Tian *et al*. [16] develop a toolkit that enables developers to compare different resource scheduling algorithms in Infrastructure as a Service (IaaS) regarding both computing servers and workloads. Yuan *et al*. [17] propose new memetic algorithms for the multi-objective flexible job shop scheduling with the objectives to minimize the makespan, total workload, and critical workload. For resource orchestration in the cloud computing, Horizon project [18] firstly points out that the heterogeneous cloud computing resources from multiple providers should be composited as a service to enhance availability, flexibility, elasticity and to meet the targeted performance constraints. Bittencourt *et al*. [19] divide the application into dependent modules and make use of the directed acyclic graph to allocate cloud computing resource for each task. The authors introduce the scheduling algorithms in hybrid clouds with object of reducing the execution time. Zuo *et al*. [20] propose a resource allocation framework to take use of the private cloud and the external public cloud together when its own resources are not sufficient to meet the demand. The problem is how to allocate users' tasks to maximize the profit of IaaS provider while guaranteeing Quality of Service (QoS). It is formulated as an integer-programming model and solved by a self-adaptive learning particle swarm optimization algorithm.

In conclusion, although computation offloading can improve the capability of consumer electronic device through migrating heavy computation tasks to powerful servers in clouds [21], the selection of multiple resources for each task is an open issue. This selection is important to the performance of the application, especially for the home cloud where there is a limitation for the computation, storage and bandwidth resources in electronic devices. Therefore, the resource orchestration in the home-to-home cloud should take into consideration the resource load and application performance. In this work, a distributed system that controls the heterogeneous resources is designed, and a particle swarm algorithm is proposed to find the appropriate solution for users.

III. HOME-TO-HOME CLOUD ARCHITECTURE

In this section, the motivation of home-to-home cloud is first presented. Then the distributed system model and its functionalities are introduced. Finally, the interactive protocol among neighboring houses is defined.

A. Motivation Example

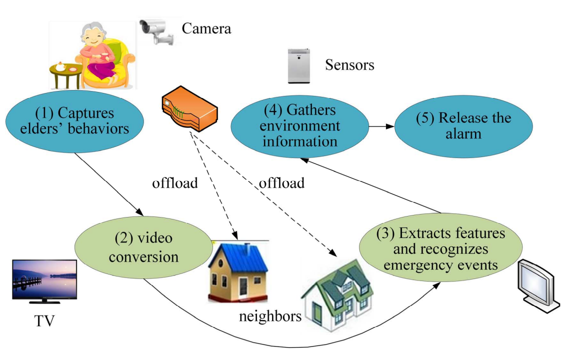


Fig. 2. An example service of home-to-home cloud. The domestic user offloads some tasks to consumer devices in neighboring houses.

Traditional home cloud provides the resource sharing among one house [1] [22]. Along with the development of cloud computing, however, sharing their idle resources for the neighbors in need becomes popular. The domestic users may require to offload tasks to foreign execution environments in neighboring houses, because task offloading to electronic devices in neighboring houses can get a faster transmission speed comparing to offloading to the remote public cloud. For example, in Fig. 2, the home-to-home cloud provides an elders emergency monitoring service. Firstly, the domestic camera captures the elders' behaviors. But the raw video data are not convenience to make pattern recognition. Thus, the home gateway offloads the data to home-to-home cloud where a TV located in a neighboring house can provide a powerful video conversion. Then the new video data as well as the analyzing application is transferred to another powerful electronic device within the home-to-home cloud, in order to make further feature extraction and recognize the emergency events, such as fainting. In the meanwhile, the home gateway gathers the environment information, air pressure, temperature, humidity, and the location provided by the domestic sensors. Finally, the home gateway releases the alarm to family members. This complicated service which needs heterogeneous cloud resources can be provided by the sharing home-to-home cloud.

B. Home-to-home Cloud Infrastructure

The proposed cloud infrastructure for the home-to-home network contains two layers, depicted in Fig. 3. In a house inside, smart electronic devices connect to the home gateway via Wireless Local Area Network (WLAN), Bluetooth [23], and so on. The virtualization container, such as Linux Container (LXC), can be installed in each intelligent operation system to control the devices and allocate Virtual Machine (VM) resources [24]. Then the offload tasks can efficiently run on VMs of consumer devices like smartphones, tablets, netbooks, smart TVs, and game consoles. Thereby, these devices can share their computation and storage among their family users.

Among the houses, home gateways are connected with each other via the Internet. The volunteer users provide several VMs or openness services in their idle electronic devices, so that much more resources can be shared in the community. The home gateways are distributed, with a resource controller inside for managing internal electronic devices, monitoring performance and orchestrating resources. In Fig.3, the home gateways are connected, and provide both the inner resource controlling and the external resource orchestration. Base on this two-layer infrastructure, a user in house "A" initiates an application containing four tasks that can be executed by the consumer electronic devices in the house "B", "C", and "D".

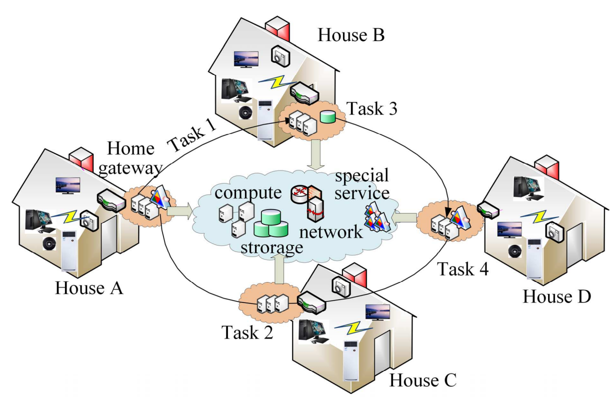


Fig. 3. Two-layer home-to-home cloud infrastructure. The first layer is that consumers share resources within a house, and the second layer is that consumers allow their resources to be shared among houses.

C. Extended Home Gateway Functionalities

For realizing resource orchestration in home-to-home cloud, the home gateway is extended by a new module, i.e. Resource Controller. Taking use of the information provided by application provider, volunteer resource provider, and the end user, the Resource Controller decides which device in the home-to-home cloud will be used to execute the offload task for an application. The interaction and information sharing mechanism in the system guarantee the functions cooperate with each other. By this way, the home gateway can make the resource orchestration for their own requesting users according to their application requirements.

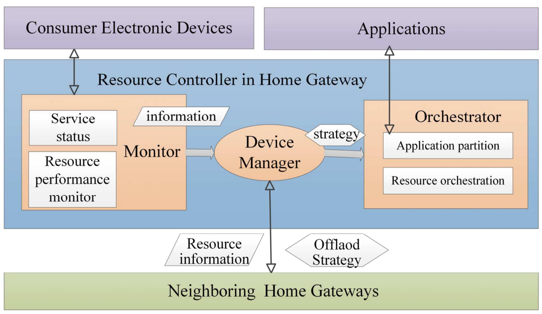


Fig. 4. Resource controller inside a home gateway. Three functions, device manager, monitor and orchestrator are contained.

The Resource Controller is deployed in home gateway, which contains three functions, device manager, monitor and orchestrator, depicted in Fig. 4. Device Manager receives the register message of connecting electronic devices inside house, and records the sharing cost and resource parameters. It also gathers the perceiving and monitoring information from the distributed home-to-home cloud resource, and transmits the resources orchestration policy from the Orchestrator function to the selected home gateways.

Monitor collects and aggregates the status and performance metrics of consumer electronic devices. By this function, the Resource Controller obtains the resource parameters such as CPU, storage and communication capability, or the number of sharing services such as firewall or detailed weather report.

Orchestrator is the central calculation module in the Resource Controller, which takes use of the monitoring information and obtains the resource orchestration policy from the optimization problem. The application partition is used to divide the complicated applications if the application itself does not contain several separated tasks. Then according to resource requirements, this function executes the intelligent algorithm and provides the optimal resource allocation strategy for an application.

D. Interactive Procedure

An appropriate incentive-compatible policy such as cooperative game theory for multiple cloud providers [25] makes the resource provision among different houses practical. Based on this, the goal is providing powerful service for family users and improving the application performance by offloading each task to the appropriate resource.

The home-to-home cloud application for intelligent operation system is deployed in each electronic devices, which can communicate with the Resource Controller in home gateway, execute the task, and provide openness interfaces of some services. The interaction procedure is depicted in Fig. 5. When the consumer electronic devices in a house start, they register their sharing resource information, including the resource configuration, openness services, and their device identifications. Moreover, the consumer electronic devices update their statuses to the home gateway periodically and the home gateway advertises the latest resource information to the neighboring houses in the home-to-home cloud.

After a user's request arrives at the home gateway, it finishes the resource orchestration and transmits the strategies to the home gateway that controls the target electronic device. The strategy is depicted by Extensive Markup Language (XML), including the identification of resource area, task and the depiction of resource requirement. The target home gateway is responsible for notifying its electrical device to execute the task offloaded. When the application in the electrical device receives the message, it performs the service migration, executes the task, and sends the result back to its home gateway. The service deployment and migration procedure can consult the work proposed by Lin *et al.* [26]. Finally, the home gateway transmits the results to the next home gateway according to the orchestration strategy.

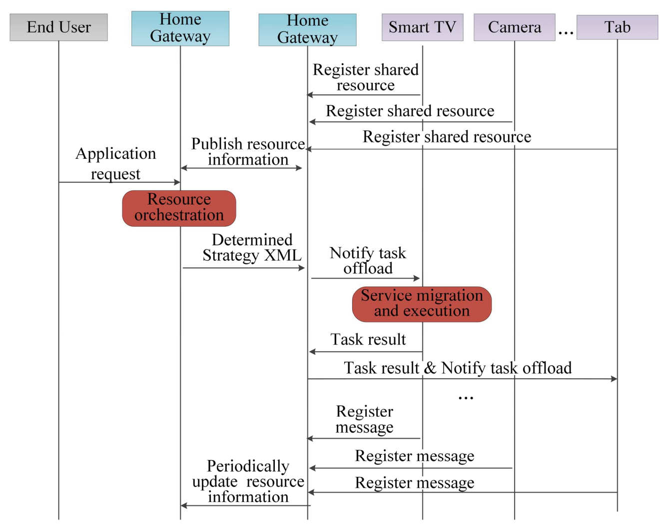


Fig. 5. Interactive procedure in home-to-home cloud.

IV. Resource Orchestration

A. Problem Definition

The heterogeneous resources from volunteer homes are abstracted as the services with specific functions and parameters. The application is composed by several dependent tasks, which is modeled as a specific dataflow graph and denoted by Directed Acyclic Graph (DAG), depicted in Fig. 6. As several resources may provide the same function, the orchestrator chooses the appropriate resource for each task according to the global optimization goal.

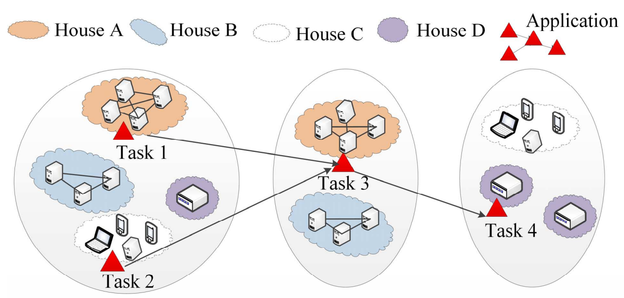


Fig. 6. Example of resource orchestration in home-to-home cloud.

Suppose there are V candidate *Resource Orchestration Paths* (*ROP*) in the problem. The objective is finding a ROP_d, which minimizes the load balance metric while satisfying the constraints. The mathematical formulation is presented below:

$$Min\ \ Z_{ROP_d} = \alpha \cdot Cur_{ROP_d} + \beta \cdot Sur_{ROP_d} + \gamma \cdot Bur_{ROP_d} \quad (1)$$

Subject to:

$$C_{ROP_d} \le C_o$$
$$T_{start} \le T_{current} \ \&\ (T_{current} + D_{ROP_d}) \le T_{end}$$
$$R_{ROP_d} \ge R_o$$

where d=1,…,V. C_o, T_{start}, T_{end} and R_o are the corresponding constraints value of cost, available time (start to end) and reliability respectively. C_{ROP_d}, D_{ROP_d} and R_{ROP_d} are the cost, using time and reliability of ROP_d, respectively.

The objective function combines three parts to evaluate the load pressure of consumer electronic devices. α, β and γ are the corresponding weighted factors to control the relative significance of compute, storage and network capability. *Cur*, *Sur* are the compute and storage load level, which reflect the workload of a consumer electronic device. *Bur* is the bandwidth, which reflects the load pressure of the link of a consumer electronic device. This objective function guarantees that the resource orchestration for the application avoids heavy load spots and heavy load lines in the home-to-home cloud.

Computing utilization rate (Cur_r^i) is the parameter depicting the percentage of the computing ability when offloading task i to the resource r provided by a consumer electronic device in the home-to-home cloud. Cur_r^i can be calculated using

$$Cur_r^i = (Cur_{required}^i + Cur_{occupied}^r) \big/ Cur_{total}^r \ . \quad (2)$$

Here, $Cur_{required}^i$ is the computing ability required by tasks i, which is provided by the service provider when the application is downloaded from the store. $Cur_{occupied}^r$ is the current occupied computing resource in r for both the local and the foreign applications. Cur_{total}^r represents the total computing resource in r. In some situations, the required computing ability is affected by the amount of related data, i.e. more data needs more computing resource. Here, only a general value of the task is considered, whereas the storage resource and the bandwidth resource are used to express this factor.

Storage utilization rate (Sur_r^i) is the parameter depicting the percentage of the storage ability when offloading task i to resource r provided by the consumer device in the neighboring house. Sur^i can be calculated using

$$Sur_r^i = (Sur_{required}^i + Sur_{occupied}^r) \big/ Sur_{total}^r \ . \quad (3)$$

The definition of parameters in (3) is similar to (2), which is used to describe the estimated storage usage percentage of a VM in the home-to-home cloud.

Bandwidth utilization rate (Bur_r^i) is the parameter depicting the percentage of the bandwidth ability of the home gateway when offloading task i to resource r provided by the consumer electronic device in the neighboring house. It reflects the available bandwidth level of neighboring houses.

$$Bur_r^i = (Bur_{required}^i + Bur_{occupied}^n) \big/ Bur_{total}^n \quad (4)$$

Here, $Bur_{required}^i$ is the bandwidth required by tasks i. $Bur_{occupied}^n$ is the occupied bandwidth that is used to access the Internet of home gateway n, and $Bur_{occupied}^n$ is the total bandwidth of home gateway n.

Cost (C_r^i) refers to the price that task i pays the consumer electronic device owner to offload its request. When the resource of the consumer electronic device is registered, the cost per second Cs_r is added to the home gateway in the meantime. If this cost changes, the information must be updated.

$$C_r^i = Cs_r \cdot (De^i + Dt_r \cdot V_{off}^i) \quad (5)$$

Here, De^i is the expected exaction time of task i and Dt_r is the transport delay of resource r for per byte data, which is the sum of transmission and propagation delay. V_{off}^i is the data volume that the task offloads to the neighboring consumer electronic devices.

Sharing Time (T) represents the resource's sharing time from the start T_{start} to the end T_{end}. As the consumer electronic device may not always be online, this parameter is provided by the owner to present the resource's available period.

Reliability (R) represents the proportion of the resource usable time. Because practical smart house are generally reparable systems, they work in the cycle composed of failure and normal status. $T^{k}_{available}$ is the k^{th} time slice when the resource in a consumer electronic device works normally. $T_{observe}$ is the total observed time. Then, for N time slices, the reliability is calculated by:

$$R_r = \sum_{k=1}^{N} T^{k}_{available} \Big/ T_{observe} \,. \tag{6}$$

B. Denotations in the Problem

The denotations in the resource orchestration are shown in Fig. 7. When a request is forwarded to the home gateway, it can be denoted by a task set $T_{Set}=\{Tk_1,\ldots, Tk_a, Tk_M\}$. Then Resource Controller maps the task set to the corresponding resources of neighboring houses, and determines each task's resource candidates $R_{Set}i=\{Rs_1,\ldots, Rs_j, Rs_m\}$. The Resource Controller chooses one resource Rs_i for each task and forms a candidate orchestration $RSset=\{Tk_1Rs_3,\ldots, Tk_1Rs_m, Tk_MRs_j\}$.

The resource orchestration is to select the optimal candidate from all possible candidates by considering the objective and multiple constrains (e.g. cost, sharing time, and availability). The sequential coding discrete PSO is adopted and the position of each particle represents a candidate solution to the problem. The number of tasks corresponds with the dimension numbers of the particle. Position values of a particle represent the resource service indexes for a task.

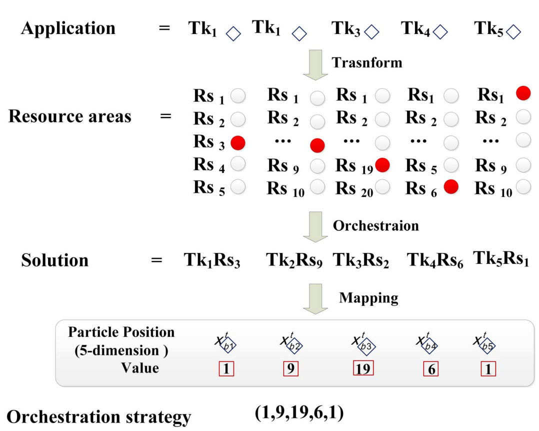


Fig. 7. Mapping resource orchestration to discrete PSO.

C. Resource Orchestration Algorithm

Assume there are M tasks in the application. Then particle is set to M-dimension. Let $x_b^t=(x_{b1}^t, x_{b2}^t,\ldots,x_{bi}^t,\ldots,x_{bM}^t)$ be the particle's b^{th} -dimensional position at the t^{th} iteration and $v_b^t=(v_{b1}^t, v_{b2}^t,\ldots,v_{bi}^t,\ldots,v_{bM}^t)$ be the particle's velocity of b^{th} -dimensional at the t^{th} iteration. Let $p_b^t=(p_{b1}^t, p_{b2}^t,\ldots,p_{bi}^t,\ldots,p_{bM}^t)$ and $p_g^t=(p_{g1}^t, p_{g2}^t,\ldots,p_{gi}^t,\ldots,p_{gM}^t)$ be the local and global optimal position, respectively. T_{max} is the maximum iteration number, and m is the size of the PSO swarm. The algorithm is shown as follows.

1. **For** each particle b in the swarm:
2. t=1;
3. Initialize x_b^1, v_b^1 randomly;
4. Let $p_b^1 = x_b^1$;
5. Initialize p_g^1 to be the one with the best fitness in p_b^1, b=1, …, m
6. Calculate fitness (x_b^t) according to (1);
7. **If** fitness (x_b^t) is better than fitness (p_b^t)
8. $p_b^t = x_b^t$;
9. **End If**
10. Let p_g^t be the one with the best fitness in p_b^t, b=1,…, m;
11. Update the particle's position and velocity with (7).
12. **If** t++<= T_{max}
13. Go to 6;
14. **End If**
15. **End For**

After initiation, the fitness value according to the objective function, i.e. (1) is calculated. If the current fitness value is better than the local optimal value, the local optimal is updated. The best value of all particles is chosen as the global optimal value. In the $(t+1)^{th}$ iteration, the b^{th} particle's i-dimensional velocity is updated according to three values, i.e. the velocity at the last iteration, its local optimum p_{bi}^{t+1} and the global optimum p_{gi}^{t+1}. Then its i-dimensional position is updated by the updated velocity and the position at the last iteration. This process is updated by (7).

$$\begin{aligned} v_{bi}^{t+1} &= \omega^t v_{bi}^t + c_1^t \xi^t (p_{bi}^t - x_{bi}^t) + c_2^t \eta^t (p_{gi}^t - x_{bi}^t) \\ x_{bi}^{t+1} &= \begin{cases} \lceil x_{bi}^t + v_{bi}^{t+1} \rceil \\ \text{with prob. } p_1 = (x_{bi}^t + v_{bi}^{t+1}) - \lfloor x_{bi}^t + v_{bi}^{t+1} \rfloor \\ \lfloor x_{bi}^t + v_{bi}^{t+1} \rfloor \\ \text{with prob. } p_2 = 1 - p_1 \end{cases} \end{aligned} \tag{7}$$

where ω^t is the time-variant inertia weight which controls the inheritance from the velocity at the last iteration. c_1^t and c_2^t are the time-variant learning factors which let the particle move towards the local optimum and the global optimum respectively. ξ^t and η^t are random numbers uniformly distributed between zero and one, which enable the particle to escape the sub-optima. If the value of x_{bi}^{t+1} is not an integer, it is rounded to the nearest small integer or the nearest large integer randomly such that the mean error is zero. Parameters in (7) are calculated as (8).

$$\begin{cases} \omega^t = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times t / T_{\max} \\ c_1^t = (c_{1f} - c_{1i}) \times t / T_{\max} + c_{1i} \\ c_2^t = (c_{2f} - c_{2i}) \times t / T_{\max} + c_{2i} \end{cases} \tag{8}$$

where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum value of inertia weight [27]. $T_{\max}$ is the maximum iteration number. In (13), c_{1i} and c_{1f} are the initial and final values of c_1^t. c_{2i} and c_{2f} are the initial and final values of c_2^t [28].

V. Performance Evaluation

A. Simulation Scenarios

The implementation set-up consists of a test bed with four VMs of desktop PCs performing as the distributed home gateways. The smart phones and several notebooks with LXC container provide the VM resources in home-to-home cloud. The home gateways are all configured with 1 GB memory, CPU 1.5 GHz, gigabit Ethernet network interfaces, and Linux operating system (nowadays configuration of home gateways), and are deployed with the Resource Controller for orchestrating the resources of consumer electronic devices.

In the simulation scenarios, various applications with different number of tasks executing sequentially are generated. The other structures of multi-task can be transformed to the sequential workflow by the existing techniques [29]. The resource parameters of consumer electronic devices as well as the tasks are generated according to a uniform distribution listed in TABLE I. Cur_r^i, Sur_r^i and Bur_r^i are configured directly. The ω_{max} and ω_{min} are set according to the work proposed by Zitzler *et al.* [27] and the c_{1i}, c_{1f}, c_{2i} and c_{2f} are set according to the work proposed by Veldhuizen *et al.* [28]. All experimental results represent an average of five trials. The available time T_{start} and T_{end} are set to 0 and 24, respectively.

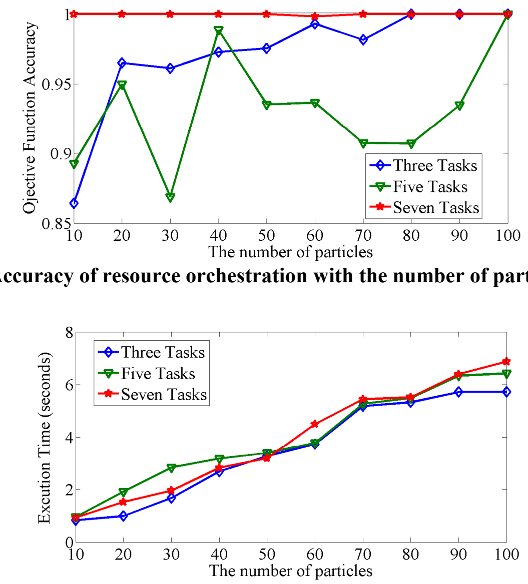
TABLE I
The Value of Simulation Parameters

Symbol	Value	Symbol	Value	Symbol	Value
ω_{max}	0.9	Cur_r^i	[0.05,0.95]	V_{off}^i	[100, 1000]
ω_{min}	0.4	Sur_r^i	[0.05,0.95]	De^i	[0.1,1] hours
c_{1i}	2.5	Bur_r^i	[0.05,0.95]	C_o	8
c_{1f}	0.5	Cs_r	[1,5]	R_o	0.7
c_{2i}	0.5	Dt_r	[0.01,0.1] seconds		
c_{2f}	2.5	R_r	[0.5,0.99]		

The users usually concern about whether the resource orchestration can archive the optimal result in acceptable time. The objective function accuracy, i.e. the ratio of objective function values of the PSO solution and the true optimal solution, as well as the exaction time for resource orchestration are analyzed in the following subsections.

B. Different Number of Tasks in Application

Three cases for test are taken, where the number of tasks in an application is three, five and seven, respectively. In addition, for these three cases, the number of resource candidates in the home-to-home cloud is 15, 5 and 3 for each task. The number of particles grows from 10 to 100. In all tests, PSO algorithm is excuted100 iterations.



(a): Accuracy of resource orchestration with the number of particles.

(b): Execution time with the number of particles.

Fig. 8. Comparison of different numbers of tasks. Three tasks, five tasks and seven tasks are configured in an application. The number of particles is set from 10 to 100.

Fig. 8 depicts the accuracy of resource orchestration and the corresponding execution time in home-to-home cloud. Generally, the variation of task numbers has little influence over the accuracy, but may lead to a little increment of the execution time. Along with the increasing of particles in PSO algorithm, the accuracy of resource orchestration among these houses increases. Nevertheless, the execution time rises greatly. For computation limited home gateway, the execution time will reduce significantly if obtaining an approximate solution.

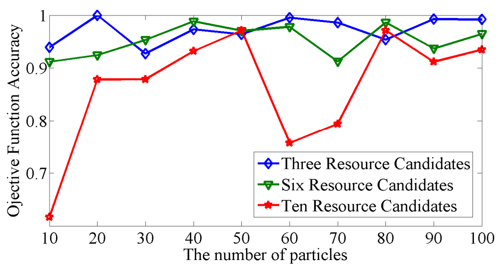
When the number of particles is more than 80, the accuracy of resource orchestration achieves 100% for the case of task numbers three and seven. However, in the case of task number five, 100% accuracy needs more particles, i.e. 100. Therefore, when the orchestration algorithm is taken into practice, the appropriate parameters should be chosen according to the specific application.

C. Different Numbers of Resource Candidates

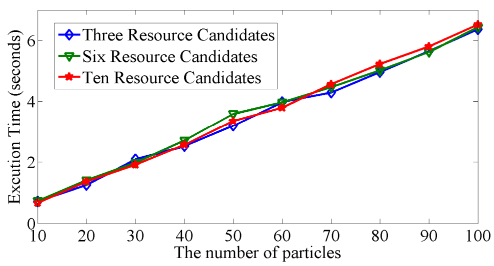
Fig. 9 illustrate the accuracy of orchestration and the execution time when an application contains four tasks and the number of resource candidate for each task is three, six and ten. In the above three cases, the candidate orchestration solution is 81, 1296 and 10000, respectively. The number of iterations maintains 100 in all tests.

As shown in Fig. 9(a), the number of resource candidates is the key factor that affects the accuracy of resource orchestration. When the number of particles is 10, the accuracy with three resource candidates is much higher than that with ten resource candidates. The difference in terms of accuracy under three cases narrows down when the number of particles increases.

In another aspect, as depicted in Fig. 9(b), the execution time is hardly affected by increasing the number of resource candidates when the swarm size is stationary. The execution time also increases when the swarm size increases.



(a): Accuracy of resource orchestration with the number of particles.



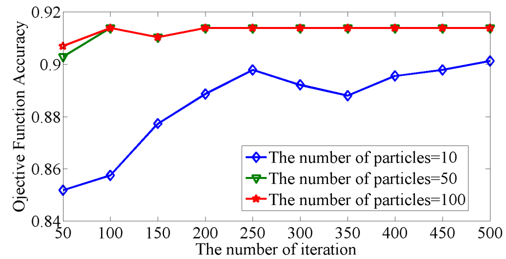
(b): Execution time with the number of particles.

Fig. 9. Comparison of different numbers of resource candidates. Three, six and ten resource candidates are set for each task. The number of particles is set from 10 to 100.

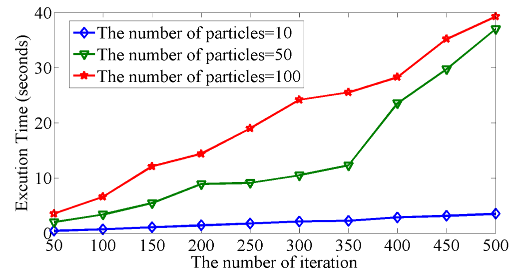
D. Different Number of Increasing Iterations

Fig. 10 presents the accuracy rate and execution time along with iterating from 50 to 500 times. As shown in Fig. 10(a), the accuracy of resource orchestration improves when the number of iterations increases, especially when the number of particles is 10. For 50 and 100 particles, 0.9139 is the largest accuracy. Thus, 200 iterations are enough. In Fig. 10(b), the execution time increases when the number of iterations increases. In the case of 10 particles, the execution time increases slowly; and in the case of 50 and 100 particles, the execution time increases sharply with 400 and 150 iterations, respectively.

According to the above results, the fast convergence of the PSO-based resource orchestration algorithm is suitable for the nowadays home gateway with low processing power. Furthermore, an effective method to improve the accuracy of resource orchestration with a little increase of the execution time is to increase the number of particles instead of the number of iterations.



(a): Accuracy of resource orchestration with increasing iterations.



(b): Execution time with increasing iterations.

Fig. 10. Comparison of different numbers of particles. The numbers of particles are 10, 50 and 100. Four tasks are taken in an application, and each task has ten resource candidates among their neighbors.

VI. CONCLUSION

With the fast-developing consumer electronics, smart house comes to the people life and the virtualization of consumer electronic devices becomes practical. This paper constructs a two-layer home-to-home cloud, which allows intra-home resources to be shared among consumer electronic devices and inter-home resources to be shared among neighboring houses. By extending the functionalities of the home gateway, the multi-task application can be offloaded to the home-to-home cloud. To utilize the resources shared by consumer electronic devices effectively, the resource orchestration is defined as an optimization problem that consider the cost, sharing time and reliability. The optimal orchestration strategy is calculated by the PSO-based algorithm and the simulation results show that all of the studied cases can archive almost 90% optimal solution in acceptable time.

Future work will investigate the security mechanisms of exploiting the volunteer resources in home-to-home clouds for multi-task applications.

REFERENCES

[1] S. Takatori, S. Matsumoto, S. Saiki. S. Tokunaga, J. Lee and M. Nakamura, "A cloud-based architecture for home network system, " *in Proc. IEEE International Conference on Cloud Computing Technology and Science*, Singapore, pp. 964-969, Dec. 2014.

[2] J. Lee, K. Choi, Y.J. Kim and S. Kang, "Design and implementation of the lightweight home cloud computing framework," *in Proc. IEEE Third International Conference on Consumer Electronics-Berlin*, Berlin, Germany, pp. 264 - 267, Sept. 2013.

[3] M. Fazio and A. Puliafito, "Cloud4sens: a cloud-based architecture for sensor controlling and monitoring," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 41-47, Mar. 2015.

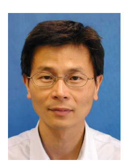
[4] Z. Wei, S. Qin, D. Jia and Y. Yang, "Research and design of cloud architecture for smart home," *in Proc. IEEE International Conference on Software Engineering and Service Sciences*, Beijing, China, pp. 86-89, Jul. 2010.
[5] J. Son, R. Hussain, H. Kim and H. Oh, "SC-DVR: a secure cloud computing based framework for DVR service," *IEEE Trans. Consumer Electron.*, vol. 60, no. 3, pp. 368-374, Aug. 2014.
[6] Y. Igarashi, M. Hiltunen, K. Joshi and R. Schlichting, "An extensible home automation architecture based on cloud offloading," *in Proc. IEEE International Conference on Network-Based Information Systems*, Taipei, pp. 187-194, Sept. 2015.
[7] S. Takatori, S. Matsumoto, S. Saiki and M. Nakamura, "A proposal of cloud-based home network system for multi-vendor services," *in Proc. IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, Las Vegas, USA, pp. 1-6, Jun. 2014.
[8] M. Lee, Y. Kim and Y. Lee, "A home cloud-based home network auto-configuration using SDN," *in Proc. IEEE International Conference on Networking, Sensing and Control*, Taipei, pp. 444-449, Apr. 2015.
[9] J. Jo, S. Lee, and J. Kim, "Software-defined home networking devices for multi-home visual sharing," *IEEE Trans. Consumer Electron*., vol. 60, no. 3, pp. 534-539, Aug. 2014
[10] D. Lee, C. Min and Y. Ik Eom, "Effective flash-based SSD caching for high performance home cloud server," *IEEE Trans. Consumer Electron*., vol. 61, no. 2, pp. 215-221, May 2015.
[11] Z. Fu, J. Shu, X. Sun and N. Linge, "Smart cloud search services verifiable keyword-based semantic search over encrypted cloud data," IEEE *Trans. Consumer Electron.,* vol. 60, no. 4, pp. 762-770, Nov. 2014.
[12] D. Jeong, J. Byun and S. Park, "Zone-aware service system with nomadic resources for cost-effective pervasive infrastructure," IEEE *Trans. Consumer Electron.*, vol. 60, no. 3, pp. 329-337, Aug. 2014.
[13] D. Diaz-Sanchez, F. Almenarez, A. Marin, D. Proserpio, and P.A. Cabarco, "Media cloud: an open cloud computing middleware for content management," *IEEE Trans. Consumer Electron.*, vol. 57, no. 2, pp. 970-978, 2011.
[14] S. Guo, L. Chen, G. Zhang and D.M. Chiu, "Batching for smart home leveraging delay-insensitive workload in cloud storage," *in Proc. International Conference on Communication Systems and Networks*, Bangalore, India, pp. 1-8, Jan. 2015
[15] F. Tao, D. Zhao, Y. Hu, and Z. Zhou, "Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system," *IEEE Trans. Ind. Inform.*, vol. 4, no. 4, pp. 315-327. Nov. 2008.
[16] W. Tian, Y. Zhao, M. Xu, Y. Zhong and X. Sun, "A toolkit for modeling and simulation of real-time virtual machine allocation in a cloud data center," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no.1, pp. 153-161, Jan. 2015.
[17] Y. Yuan and H. Xu, "Multiobjective flexible job shop scheduling using memetic algorithms," *IEEE Trans. Autom. Sci. Eng.*, vol.12, no.1, pp. 336-353, Jan. 2015.
[18] "Information and Communication Technologies", Horizon 2020 Work Program, 2014-2015.
[19] L. F. Bittencourt, E. R. M. Madeira and N. Fonseca, "Scheduling in hybrid clouds," *IEEE Commun. Mag.*, vol. 50, no. 9, pp. 42-47, Sept. 2012.
[20] X. Zuo, G. Zhang and W. Tan, "Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IaaS cloud," *IEEE Trans. Autom. Sci. Eng.*, vol.11, no. 2, pp. 564 -573, Apr. 2014.
[21] S. Abolfazli, Z. Sanaei, M. Alizadeh, A. Gani and F. Xia, "An experimental analysis on cloud-based mobile augmentation in mobile cloud computing," *IEEE Trans. Consumer Electron.*, vol. 60, no. 1, pp. 146 - 154, Feb. 2014.
[22] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, and C. Lung, "Smart home: integrating internet of things with web services and cloud computing," *in Proc. IEEE International Conference on Cloud Computing Technology and Science*, Bristol, pp. 317 - 320, Dec. 2013.
[23] J. Etxaniz and G. Aranguren, "Modeling of the data transportation network of a multi-hop data-content-sharing home network," *IEEE Trans. Consumer Electron.*, vol. 61, no. 1, pp. 31-38, Feb. 2015.
[24] J. Jeong and S. Hur, "Sensor device virtualization for mobile clouds," *in Proc. IEEE International Conference on Consumer Electronics*, Las Vegas, USA, pp. 288 -289, Jan. 2014.
[25] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing", *IEEE J. Sel. Area. Commun*., vol. 31, no. 12, pp. 2685 - 2700, Dec. 2013
[26] Shiraz M., Gani A., Khokhar R.H. and Buyya R., "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Commun. Surv. & Tutor*., vol. 15, no. 3, pp. 1294-1313, Third Quarter 2013.
[27] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca and V.G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Trans. on Evol. Comput.* vol. 7, no. 2, pp. 117-132, Apr. 2003.
[28] D.A.V. Veldhuizen and G.B. Lamont, "On measuring multiobjective evolutionary algorithm performance", *in Proc. IEEE Congress on Evolutionary Computation*, La Jolla, CA, pp. 204-211, Jul. 2000.
[29] J. Liao, Y. Liu, X. Zhu, and J. Wang, "Accurate sub-swarms particle swarm optimization algorithm for service composition", *J. Sys. and Soft.*, vol. 90, pp. 191-203, Apr. 2014.

BIOGRAPHIES

Qi Qi (M'14) was born in 1982, obtained her PhD degree from Beijing University of Posts and Telecommunications in 2010. Now she is an assistant professor in Beijing University of Posts and Telecommunications, China. Her research interests include cloud computing, home network, mobility management, resource scheduling, ubiquitous services, and multimedia communication.

Jingyu Wang (M'13) was born in 1978, obtained his PhD degree from Beijing University of Posts and Telecommunications in 2008. Now he is an associate professor in Beijing University of Posts and Telecommunications, China. His research interests span broad aspects of home network, software defined networking, multi-path transmission, overlay network, traffic engineering, and multimedia communication over wireless network.

Qi Li was born in 1982, obtained her PhD degree from Beijing University of Posts and Telecommunications in 2010. Now she is an associate professor in Beijing University of Posts and Telecommunications, China. Her research interests include mobile sensors network, internet of things, software security, and multimedia communication.



Tonghong Li was born in 1968, obtained his PhD degree from Beijing University of Posts and Telecommunications in 1999. He is currently an assistant professor with the department of computer science, Technical University of Madrid, Spain. His main research interests include resource management, distributed system, middleware, wireless networks, sensor networks, and optimization algorithms.

Yufei Cao was born in 1974, obtained his PhD degree from Beijing University of Posts and Telecommunications in 2008. He joined the EBUPT Information Technology Company, China, in 2008, and is currently working as a Research Engineer. His research interests include cloud computing, home gateway design, and network protocol.