Demo: Transient Clouds

Terry Penner, Alison Johnson, Brandon Van Slyke, Mina Guirguis, Qijun Gu
Computer Science Department
Texas State University
{t_p68,abj25,brandon.vanslyke,msg,qijun}@txstate.edu

***Abstract*— In this paper, we present our demo of Transient Clouds – a platform that allows for mobile devices within range of each other to form an ad-hoc network and collaboratively execute tasks. Transient Clouds are utilized in temporal scenarios in which the cloud is created on-the-fly by the devices present in an environment and would disappear as the devices leave the network. In Transient Clouds, each device offers a different set of capabilities based on its software and hardware. When a particular device has a task to execute, it can send it into the Transient Cloud that would assign the subtasks to the devices, based on their capabilities, and collect the results back. We use a modified version of the Hungarian method to decide the assignments to achieve certain properties such as load balancing and collocating subtasks. This demo showcases an implementation of a Transient Cloud, albeit limited by Android's Wi-Fi Direct framework, in which a number of devices provide their capabilities as a cloud service. The demo is based on an Android app that connects the devices and allows for the remote execution of subtasks on devices based on the assignment algorithm.**

I. Motivation and Background

Transient Clouds [1] is platform that allows a group of mobile devices to offer their capabilities as a cloud service solely among themselves without requiring access to any existing infrastructure. They are envisioned to be utilized in temporal scenarios in which the cloud is created on-the-fly and would disappear as the context ceases to exist. Transient Clouds are motivated by the fact that mobile devices have grown more and more powerful, and by using them as a single big collection of various capabilities rather than individual devices the scope of the tasks that they can accomplish should grow more significantly. Currently, any single device is constrained by what functions its hardware and software can provide. With Transient Clouds, this constraint is lifted. If a device is missing a particular needed capability (or a resource), it can ask another device in the Transient Cloud to supply that capability/resource on its behalf. For example, if a device needs its approximate GPS coordinates but lacks a GPS chip, it can use the Transient Cloud that it is a member of to get the coordinates of a nearby device, and use that as an approximate location for itself. Transient clouds are different from other mobile clouds (e.g., [2]–[4]) in that they deal with the dynamic nature of devices joining and leaving the network as well as dynamically adapt the assignments of capabilities to devices to achieve load balancing and task collocation properties.

We decided to use the Hungarian method [5] as the basis for our assignment algorithm due to its efficiency and simplicity. It finds the minimum overall cost of assigning tasks to devices. This is demonstrated in the selection of elements in a matrix where no selections are in the same row or column as any others. We set up our matrix to have the columns as capabilities and the rows as devices where each device were expanded so that each capability that it offered is in a separate row. This can be seen in the change from Figure 1a to 1b.

Devices	Capabilities 0	1	2	3	4	5
0	1	1	1	1	1	1
1	1	0	0	0	0	0
2	0	1	0	1	0	0
3	0	0	0	1	0	0
4	0	0	0	0	1	0
5	0	0	0	0	1	0
6	1	0	1	1	1	0

(a) Compressed capability table

Devices with Available Capabilities	Capabilities 0	1	2	3	4	5
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1
1	1	0	0	0	0	0
2	0	1	0	0	0	0
2	0	0	0	1	0	0
3	0	0	0	1	0	0
4	0	0	0	0	1	0
5	0	0	0	0	1	0
6	1	0	0	0	0	0
6	0	0	1	0	0	0
6	0	0	0	1	0	0
6	0	0	0	0	1	0

(b) Standard selection

Devices with Available Capabilities	Capabilities 0	1	2	3	4	5
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1
1	1	0	0	0	0	0
2	0	1	0	0	0	0
2	0	0	0	1	0	0
3	0	0	0	1	0	0
4	0	0	0	0	1	0
5	0	0	0	0	1	0
6	1	0	0	0	0	0
6	0	0	1	0	0	0
6	0	0	0	1	0	0
6	0	0	0	0	1	0

(c) Modified selection

Fig. 1. Standard vs. modified Hungarian method selection

We then modified the Hungarian method to discourage the selection of multiple capabilities offered by the same device to achieve load balancing. The idea is to inflate the costs of those other capabilities offered by that device, if one of its capabilities is already chosen. This ensures that a device will be assigned multiple capabilities only if there are no other options. The result of this modification can be seen in the change of selection in Figure 1b to Figure 1c. Similarly, the costs can be deflated to encourage collocating tasks on the same device.

MobiCASE 2014, November 06-07, Austin, United States

DOI 10.4108/icst.mobicase.2014.257792

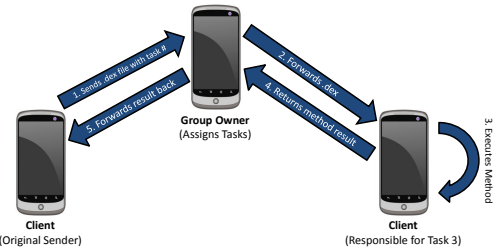

Fig. 2. Illustration of code distribution in Transient Clouds

II. Implementation

Our goal in this demo is to show the feasibility of Transient Clouds even with today's technology. We decided to use the Android operating system because it offers the most flexibility. We also wanted our framework to work with any off-the-shelf Android device so we made the decision to work within the confines of unrooted devices. We use the Wi-Fi Direct framework, a relatively recent addition to the Android operating system that allows devices to connect to each other directly over Wi-Fi and exchange data. Unfortunately, the stock Android implementation of Wi-Fi Direct requires that one device be designated as the "Group Owner" and that all other devices must connect to it in order to join the network. As a result, our demo does not use a complete ad-hoc network like our conceptual model, and instead forms a network with a star topology. We decided, however, to take the full advantage of this star topology, and gave the Group Owner full responsibilities for creating and maintaining the capabilities table and assignments. Because every device in the network is required to connect to the Group Owner to join the network, the Group Owner is only required to manage the table rather than provide any capabilities itself.

An illustration of our demonstration Transient Cloud design can be seen in Figure 2. When the first two devices connect to each other, one of them is elected by the Wi-Fi Direct protocol to be the Group Owner. The Group Owner then receives the list of capabilities offered by the client device and starts building the capability table. When additional devices want to join the network, they connect to the Group Owner and send it their lists of capabilities as well. Each time the capability table is changed, either through a device joining the cloud or leaving it, the Group Owner re-runs the modified Hungarian method and updates its assignments.

When a client device decides that it would like to have a task offloaded into the Transient Cloud, it sends the task number and the code that it would like to have executed to the Group Owner. It sends the code in a pre-compiled format, along with the name of the class and method that it would like to be executed and any parameters the method requires. The Group Owner looks at its current assignments and forwards all of this information to the client device that has been assigned that particular task. When it receives the code, this client device uses Java's Reflection library to find the specified class and method inside the pre-compiled file and executes it. The result is captured and returned to the Group Owner, which then forwards the result to the original requesting client.

This implementation style is more of a proof-of-concept. As mentioned earlier, one of the goals was to prove that this could be done using today's available consumer technology with a minimal amount of set up for the users, which is why we worked within the limitations of Android's Wi-Fi Direct framework. On a rooted device, however, a full implementation of a complete ad-hoc network becomes feasible.

III. Demonstration

Our demonstration is composed of several Android devices which we will connect together with Wi-Fi Direct. As the devices connect to the Group Owner, their "capabilities" will be assigned at random to simulate the idea that they are all different devices. When a client device connects to the network, it will gain the ability to request a capability be offloaded into the Transient Cloud. When it makes this request, the process illustrated in Figure 2 and described above begins. Notifications will appear on the screens of the involved devices as the different stages of the remote execution occur. For the final stage, the result will be displayed on the screen of the client that requested the offloaded capability.

The pre-compiled code that will be offloaded as the task is fairly straightforward, containing some dummy methods that are used to simulate capabilities that can be requested. These methods perform simple arithmetic operations that are easily verifiable. This way, a device can request a specific capability be offloaded and will get back the type of result that it is expecting.

IV. Technical Requirements

This demonstration has no special requirements. The mobile devices involved will be supplied by us, and there are no other components of this demonstration to consider. The only possible exception to this would be if there was a way to easily display the screens of all of the devices to a group of people all at once.

Acknowledgments

This research is funded by NSF REU award #1156712 that is co-funded by the Department of Defense and by NSF CNS award #1149397.

References

[1] T. Penner, A. Johnson, B. V. Slyke, M. Guirguis, and Q. Gu, "Transient Clouds: Assignment and Collaborative Execution of Tasks on Mobile Devices," in *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, Austin, TX, Dec 2014.

[2] E. Miluzzo, R. Cáceres, and Y. Chen, "Vision: mClouds - Computing On Clouds of Mobile Devices," in *Proceedings of the third ACM workshop on Mobile Cloud Computing and Services*, Ambleside, UK, June 2012.

[3] OSGi Alliance Staff, "OSGi Alliance," http://www.osgi.org/Main/HomePage, 2013.

[4] M. Yuriyama and T. Kushida, "Sensor-Cloud Infrastructure-Physical Sensor Management with Virtualize d Sensors on Cloud Computing," in *Proceedings of NBiS*, Takayama, Gifu, Japan, Septemeber 2010.

[5] H. Kuhn and B. Yaw, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, pp. 83–97, 1955.