C2C: Peer-to-Peer Networking of Smartphones on the Cloud

Sokol Kosta, Vasile Claudiu Perta, Julinda Stefa, Pan Hui, Alessandro Mei

Presenter:
Sandeep
Department of Computer Science and Engineering
University of Minnesota, Twin Cities
Research has been done to offload computations to cloud. E.g. MAUI

Aim at smartphone to smartphone communication offloading along with computation

A wireless P2P network is hard to realize due to various limiting factors
Introduction and Motivation

- C2C associates a software clone to every smartphone and interconnects the clones in a P2P fashion.

- CloneDoc is implemented on top of the C2C platform.

- 99%, 80% and 30% of energy saving for security checks, user status update and document editing respectively.
C2C is a distributed platform for cloud clones of smartphones.

MAUI - a method level code offloading system based on the Microsoft .Net framework.

Clonecloud uses a process based offloading methodology.
Above offloading mechanisms do not provide P2P inter-connections of smartphones through the cloud

They do not provide communication offloading

C2C offers the possibility to users to adopt an offloading methodology totally independent from that of other users in the platform
Cloning on the cloud:

- For C2C platform, Android x86 – an Android port to the x86 architecture is exploited
- Two hosting strategies: Private and Public
- Designed a custom Amazon Machine Image for Android x86 that boosted the performance enormously of the clones
CloneDS - Mapping users to clones and clones to IPs. It is always up and its IP is known publicly to the users.
All the entities in the system have a public/private key pair and can securely verify their authenticity.

A user requests a clone equipped with a public/private key pair to any cloud provider.

The user signs the public key so everybody can verify the clone’s owner.
**Steps performed by Clone**

- **DS Register** - clone sends device ID, its ID, IP address, public key

- **DS lookup** - receives a list of all other clones and their public keys and IP addresses

- **C2C connect** – Establish P2P connections with other clones
**User Lookup** - User can get her clone's IP by Clone DS lookup.

**User-Clone connection** - User connects to its clone through its public IP and install whatever she likes. Negotiates a symmetric key to encrypt and sign user-clone communication.
C2C and Security

- Users trust their own cloud provider but not other cloud providers.

- Interaction b/w the user and the clone is secured by using a shared symmetric key.

- CloneDS is trusted by all users in the system. Since, the information is provided by cloud providers of other users, information can be malicious.

- Since the cloud typically guarantees high availability, this distributed alternative would be more available.
Secure, real-time group collaboration on sensitive information can be efficiently deployed.

Force a global order in current users' operations

Due to the sensitiveness of the information, the server is considered as potentially malicious: its goal may be to partition the clients in disjoint groups with different views of document.
Secure real time collaboration

- Prevalent solutions to the problem involve recurrent P2P communication among users as well as heavy cryptography to guarantee crucial security and system properties.

- C2C delivers efficient peer-to-peer networking for smartphones by moving computation and communication to the cloud.
CloneDoc

- P2P like application with communication among peers and not so lite computation

- To make the cloud receive operations from the mobile device, handle many tasks and keep device up to date.

- Clone maintains two states: pending queue and the committed queue
Submits to the server, the operations that he gets from the user's real-device; transforms the operations of the other users received from the server.

Clone sends back the operations to the real device such that the user's view is coherent to that of other users in the system.

Clonedoc includes a clone-user consistency protocol so that the system looks real-time to the user.
When the user is disconnected the clone continues to be on and applies the updates locally. It pulls the updates that it missed during the absence.

Advantages:
- The user's device need not transform these operations past possible
- Reduces the number of possible operations
- Clone sends the whole sequenced bundle. Thus decryption only once.
16 smartphones, 16 clones, 14 Amazon EC2 and 2 on private. Untrusted server on private cloud.

<table>
<thead>
<tr>
<th>Number, type &amp; OS</th>
<th>CPU</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>6×Samsung Galaxy S+ (Android 2.3)</td>
<td>1.4 GHz</td>
<td>512 MB</td>
</tr>
<tr>
<td>2×Nexus S (Android 4.0.1)</td>
<td>1 GHz</td>
<td>512 MB</td>
</tr>
<tr>
<td>2×HTC Desire (Android 2.3)</td>
<td>1 GHz</td>
<td>576 MB</td>
</tr>
<tr>
<td>6×HTC Hero (Android 2.1)</td>
<td>528 MHz</td>
<td>288 MB</td>
</tr>
</tbody>
</table>

Table 2: Specifics of the mobile devices used in the testbed.
Experiment

Test has 4 phases: editing phase, update temporarily disconnected phase, security check to detect misbehavior and user membership remove.

<table>
<thead>
<tr>
<th>Protocol &amp; Smartphone</th>
<th>Phase 1</th>
<th>Phase 2</th>
<th>Phase 3</th>
<th>Phase 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPORC-Hero</td>
<td>2300</td>
<td>140</td>
<td>180</td>
<td>5.5</td>
</tr>
<tr>
<td>SPORC-Samsung</td>
<td>1700</td>
<td>60</td>
<td>130</td>
<td>2.4</td>
</tr>
<tr>
<td>CloneDoc-Hero</td>
<td>1500</td>
<td>22</td>
<td>7</td>
<td>0.9</td>
</tr>
<tr>
<td>CloneDoc-Samsung</td>
<td>1250</td>
<td>18</td>
<td>5</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 3: Energy consumption in mJ during the 4 test phases.
The C2C platform boosts up the performance of old-fashioned low-performing smartphones making them competitive with newer, more expensive, and more performing ones.

Figure 2: Time to (a) update the state of a temporarily disconnected user; (b) apply all edit users’ operations.
Overall network bandwidth used by the smartphones with CloneDoc is 3 times less than with...
Closing Thoughts

- Makes the battery last longer thanks to computation offloading on the cloud
- Makes the device use the network interfaces less, by offloading communication on the cloud;
- Boosts up the performance of old low-performing smartphones making them competitive with newer, more expensive, and more performing ones.