Research, Technology, and System Solutions

A New Approach for Handheld Devices in the Military

By Edwin Morris, Senior Member of the Technical Staff

Many people today carry handheld computing devices to support their business, entertainment, and social needs in commercial networks. The Department of Defense (DoD) is increasingly interested in having soldiers carry handheld computing devices to support their mission needs in tactical networks. Not surprisingly, however, conventional handheld computing devices (such as iPhone or Android smartphones) for commercial networks differ in significant ways from handheld devices for tactical networks. For example, conventional devices and the software that runs on them do not provide the capabilities and security needed by military devices, nor are they configured to work over DoD tactical networks with severe bandwidth limitations and stringent transmission security requirements. This article describes exploratory research we are conducting at the Software Engineering Institute (SEI) to (1) create software that allows soldiers to access information on a handheld device and (2) program the software to tailor the information for a given mission or situation.

To motivate the need for tactical handheld devices, imagine a U.S. soldier on patrol, deployed abroad, and walking into an unfamiliar village. Many pieces of information would be useful to that soldier in that situation. For example, it would be useful to know who the village elders are and to have pictures to identify them. It would also be useful to access information about previous improvised explosive device (IED) attacks, reports detailing the results of other contact that soldiers have had with villagers, and whether any friendly villagers speak English. We face the following challenges when creating software for tactical handheld computing devices that can provide this information:

- *Developing applications that can support the full range of military missions.* In recent years, soldiers have provided humanitarian assistance to victims of natural disasters in Haiti and countries in Asia, patrolled our country's borders, protected global waterways from piracy, and performed many types of military operations in Iraq and Afghanistan. These missions are sufficiently diverse that a one-size-fits-all software solution is not practical. For example, consider the different goals of clearing a route in a combat zone versus delivering humanitarian supplies in a relief effort or the different information required to protect from IED attacks versus treat a critically ill child. Not only is different information required, but also the rules for sharing it can vary. In a combat environment, security concerns require limiting access, while information in a relief mission may be shareable with nongovernmental organizations responding to the crisis.
- *Processing large amounts of data available through the rapid computerization and internetworking of various military missions.* For example, the military employs hundreds of unmanned aerial vehicles that generate large amounts of data. There are also increases in the number of sensors, such as auditory, biological, chemical, and nuclear, that are network enabled. All the data generated from these devices makes it hard to pinpoint the right information for a given mission and situation.

Our goal is to ensure the capabilities provided on tactical handheld computing devices are flexible enough to allow solders to control the amount and type of data that they receive and adaptive enough to meet the needs of particular missions. To achieve this goal, we are exploring the integration of end-user programming techniques, active data filtering and formatting, and confidence-building strategies. End-user programming techniques enable soldiers to program software on tactical handheld devices without requiring them to be professional software developers. Filtering incoming information and displaying it in intuitive formats helps avoid inundating soldiers on patrol with too much data. Confidence-building strategies promote trust that applications programmed by soldiers work correctly and safely. We are currently developing software for Android devices, but the fundamental concepts are applicable to other mobile platforms as well.

A key concern is designing software that has an intuitive and simple-to-use interface since the soldiers customizing these capabilities are not programmers; they are war fighters. The software we build must therefore help them readily find and assemble the types of information they need. It should reduce the soldier's workload by filling in (auto-complete) as much information for the soldier as possible. The software should require soldiers to learn only a few different types of screens (for example, screens for entering data and for establishing filters should be substantially the same). In addition, confidence-building feedback should be integrated into the interface so that soldiers are sure that what they build will work and are informed early if it will not.

Our work also focuses on ensuring that the information—whether from central command or a local unit—makes its way quickly and efficiently to the handheld computing device used by soldiers. For example, user-programmable data filtering allows soldiers to specify what information is important. Likewise, optimized protocol implementations ensure this information is exchanged quickly.

Last year, we conducted a research project that involved taking a service-oriented architecture approach to provide real-time situational awareness data to Android smartphones. We worked with soldiers through the Naval Postgraduate School's Center for Network Innovation and Experimentation (CENETIX) to test our applications. They told us what capabilities they need, and what did not work. These collaborations tie our work firmly into both the research and military communities and keep us focused on providing a useful and cutting-edge capability. In addition to continuing our collaboration with CENETIX, we are working with Dr. Brad Myers of the Carnegie Mellon University Human Computer Interaction Institute. Dr. Myers is helping us define an appropriate interface for soldiers to use the handheld software in the challenging situations they face.

Related Web Sites

http://blog.sei.cmu.edu/post.cfm/a-new-approach-for-handheld-devices-in-the-military

For General Information

For information about the SEI and its products and services, contact
Customer Relations
Phone: 412-268-5800
FAX: 412-268-6257
info@sei.cmu.edu
www.sei.cmu.edu

Software Engineering Institute | Carnegie Mellon

Research, Technology, and System Solutions

Cloud Computing for the Battlefield

By Grace Lewis, Senior Member of the Technical Staff

The Department of Defense (DoD) is increasingly interested in having soldiers carry handheld mobile computing devices to support their mission needs. Soldiers can use handheld devices to help with various tasks, such as speech and image recognition, natural language processing, decision making, and mission planning. Three challenges, however, present obstacles to achieving these capabilities. The first challenge is that mobile devices offer less computational power than a conventional desktop or server computer. A second challenge is that computation-intensive tasks, such as image recognition or even global positioning systems, take heavy tolls on battery power. The third challenge is dealing with unreliable networks and bandwidth. This article explores our research to overcome these challenges by using cloudlets, which are localized, lightweight servers running one or more virtual machines (VMs) on which soldiers can offload expensive computations from their handheld mobile devices, thereby providing greater processing capacity and conserving battery power.

Leveraging external resources to augment the capabilities of resource-limited mobile devices is a technique commonly known as cyber-foraging. The use of VM technology provides greater flexibility in the type and platform of applications and also reduces setup and administration time, which is critical for systems at the tactical edge. The term *tactical edge* refers to systems used by soldiers or first responders that are close to a mission or emergency executing in environments characterized by limited resources in terms of computation, power, and network bandwidth, as well as changes in the status of the mission or emergency.

Cloudlets are located within proximity of handheld devices that use them, thereby decreasing latency by using a single-hop network and potentially lowering battery consumption by using WiFi instead of broadband wireless, which consumes more energy. For example, a cloudlet might run in a Tactical Operations Center (TOC) or a Humvee. From a security perspective, cloudlets can use WiFi networks to take advantage of existing security policies, including access from only specific handheld devices and encryption techniques.

Related work on offloading computation to conserve battery power in mobile devices relies on the conventional Internet or environments that tightly couple applications running on handheld devices and servers on which computations are offloaded. In contrast, cloudlets decouple mobile applications from the servers. Each mobile app has a client portion and an application overlay corresponding to the computation-intensive code invoked by the client. On execution, the overlay is sent to the cloudlet and applied to one of the VMs running in the cloudlet, which is called dynamic VM synthesis. The application overlay is pre-generated by calculating the difference between a base VM and the base VM with the computation-intensive code installed. The only coupling that exists between the mobile app and the cloudlet is that the same version of the VM software on which the overlay was created must be used. Since no application-specific software is installed on the server, there is no need to synchronize release cycles between the client and server portions of apps, which simplifies the deployment and configuration management of apps in the field.

Dynamic VM synthesis is particularly useful in tactical environments characterized by unreliable networks and bandwidth, unplanned loss of cyber foraging platforms, and a need for rapid deployment. For example, imagine a scenario where a soldier needs to execute a computation-intensive app configured to work with cloudlets. At runtime, the app discovers a nearby cloudlet located on a Humvee and offloads the computation-intensive portion of code to it. Due to enemy attacks, network connectivity, or exhaustion of energy sources on the cloudlet, however, the mobile app is disconnected from the cloudlet. The mobile app can then locate a different cloudlet (e.g., in a TOC) and—due to dynamic VM synthesis—can have the app running in a short amount of time, with no need for any configuration on the app or the cloudlet. This flexibility enables the use of whatever resources become opportunistically available, as well as replacement of lost cyber-foraging resources and dynamic customization of newly acquired cyber-foraging resources.

As part of our research, we are focusing on face recognition applications. Thus far we have created an Android-based facial recognition app that performs the following actions:

1. It locates a cloudlet via a discovery protocol.
2. It sends the application overlay to the cloudlet where dynamic VM synthesis is performed.
3. It captures images and sends them to the facial recognition server code that now resides in the cloudlet.
4. The application overlay is a facial recognition server written in C++ that processes images from a client for training or recognition purposes. When in recognition mode, it returns coordinates for the faces it recognizes as well as a measure of confidence. The first version of the cloudlet is a simple HTTP server that receives the application overlay from the client, decrypts the overlay, decompresses the overlay, and performs VM synthesis to dynamically set up the cloudlet.

The first phase of our work has focused on creating the cloudlet prototype described above. In the second phase, we will conduct measurements to see if computations in a cloudlet provide significant reductions in device battery power. In addition, we will gather measurements related to bandwidth consumption of overlay transfer and VM synthesis to focus on optimization of cloudlet setup time. Assuming we are successful, our third phase will create a cloudlet in the RTSS Concept Lab to explore other ways to take computation to the tactical edge.

As part of our research, we are collaborating with Mahadev Satyanarayanan, the creator of the cloudlet concept and a faculty member at Carnegie Mellon University's School of Computer Science.

Related Web Sites

www.sei.cmu.edu/sos/research/cloudcomputing
www.sei.cmu.edu/library/abstracts/webinars/Cloud-Computing.cfm
http://blog.sei.cmu.edu/archives.cfm/category/cloud-computing

For General Information

For information about the SEI and its products and services, contact
Customer Relations
Phone: 412-268-5800
FAX: 412-268-6257
info@sei.cmu.edu
www.sei.cmu.edu

07/28/2011