Application Virtualization as a Strategy for Cyber Foraging

featuring Grace Lewis interviewed by Suzanne Miller

Suzanne Miller: Welcome to the SEI podcast series, a production of the Carnegie-Mellon Software Engineering Institute. The SEI is a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

My name is Suzanne Miller. I'm a principal researcher here at the SEI, and today I am very pleased to introduce you to Grace Lewis, deputy lead for the SEI's Advanced Mobile Systems Initiative and technical lead for the Edge-Enabled Tactical Systems research team. Her main research interests are mobile computing, service-oriented architecture, and cloud computing. So she's been on the edge of a lot of really interesting things in the last few years. Grace has more than 20 years of professional software development experience, mainly in industry, and is also active in mentoring student teams in Carnegie-Mellon University's Master [of science] in Software Engineering Program. She is a respected thought leader in pretty much every research area that she has been involved with, and I have had the privilege of working with Grace in the past. I am particularly glad to have a chance to catch up with you on your current research today. So, welcome, Grace.

Grace Lewis: Thank you, Suzie. The pleasure is mine.

Suzanne: Today, in our podcast, Grace and I will be discussing her recent technical report—this one's a mouthful—Application Virtualization as a Strategy for Cyber-Foraging in Resource-Constrained Environments, which she co-authored with Dominik Messinger. Let's start off by having you explain exactly what cyber-foraging is, and why is it important to understand how it works?

Grace: So, cyber-foraging is a technique for dynamically augmenting the capabilities of resource-limited devices, like mobile devices. So, basically, you can think of trying to find resources around you on which you can do the stuff that you can't do on your limited-resource mobile devices. So, an example of cyber-foraging is code offload.

So, if you're doing code offload, what you are doing is you are taking an application that has very complex code

Suzanne: Like photo recognition.

Grace: Like photo recognition.

Suzanne: Or voice recognition.

Grace: Or voice recognition, and taking that very expensive code, and offloading it onto a nearby server. So, in that manner, that very expensive code gets executed on the server and not on the mobile device where it would completely drain the battery.

Suzanne: When you say expensive, you mean in terms of things like battery load, memory, that kind of thing.

Grace: Computation.

Suzanne: Computation.

Suzanne: Okay.

Grace: So, that's one form of cyber-foraging. Another form of cyberforaging would be data staging. In this case, what you're offloading is not computation, but what you're doing is you're not only offloading data to a nearby server, but using that server to do expensive computation, such as data filtering, such as data pre-fetching, and anything that basically limits the amount of processing that has to be done on a mobile device.

Suzanne: Okay, so when you're out in the field, that's going to be important if someone gives you a task that involves some of these expensive kinds of tasks.

Grace: Correct. And, if you think of mobile devices now, and also the amount of sensors that people in the edge--which we'll talk about later--get deployed with, those sensors are collecting a large, large, large amounts of data. So, it's very nice to have nearby resources on which that data can be offloaded periodically, or opportunistically, so that it doesn't get lost, for example.

Suzanne: If you weren't on the edge, if you were here in Pittsburgh or the university, you would just use a cloud device, right? You'd use a cloud server. You would have computational resources that are more accessible to you in a lot of sense.

Grace: Yes. Actually, we call that "the Starbuck's scenario," where basically you have all the resources you want. You're relaxed. You're sitting in your office, and you're offloading whatever it is. You don't need to worry about resources.

At the edge, you do need to worry about that, because battery is expensive and heavy. The network is limited. You don't know if you're going to have connectivity around the clock. So, yes.

Suzanne: Okay, very good. So, Grace, I know a lot of your recent research is about helping soldiers out in the field who use smart phones in battle, as we say in the SEI, "on the tactical edge of things." What are some of the issues with respect to resource constraints, some that you've already talked about, and other things that these soldiers at the edge have to contend with?

Grace: So, first of all it's not only soldiers at the edge. Like another type of person that would deal with these same problems is a first responder, which might be easier for some people to relate to. So, for example, people that are on a mission trying to help people after an earthquake or after some kind of accident occurs, or things like that. So, edge environments in general are characterized by high mobility, meaning that one minute you're here, the next minute you're there. Basically, all you have with you is potentially a smart phone, which is happening nowadays.

It's also characterized by rapidly changing mission requirements. Again, let's go to the first responder scenario. You are helping somebody that is injured, and all of a sudden you have an aftershock. So, you quickly go from a health providing mode into another crisis mode.

Suzanne: Right.

Grace: The same happens, obviously, in a battle situation. So, that's another characteristic. Limited computing resources, smart phones, tablets--they're getting more powerful day by day, but they will never ever have the same resources as a desktop. They're always going to lag behind, because that's just the way it is. So, there are limited computing resources.

High levels of stress. You can imagine what these people go through from a stress perspective, and also from what we call "cognitive load"--just a lot of things happening around them. And finally, poor network connectivity.

If you're in a battlefield, you might have an infrastructure at the edge. It might be inside a tactical operations center. If you're talking about first responders, they're also called command centers, but basically they're set up in trailers. So, the network connectivity that you have is whatever you have between your device and those trailers, let's say, or tents. Then, from thereon, you might have other connectivity. But, dealing with that limited network connectivity is another issue that we deal with.

Suzanne: So, tell us how this latest research on virtualization fits into the larger picture of your ongoing effort inesource computation at the edge?

Grace: So, we are investigating a very specific type of cyber foraging, which is called cloudlet-based cyber foraging. Cloudlet was a term that was coined by one of our collaborators on campus, Mahadev Satyanarayanan. We all call him "Satya."

What cloudlets are, are these discoverable, generic, stateless servers that are located in single-hop proximity of mobile devices, meaning that we're not using 3G or 4G or anything like that. We're just using wireless communications to connect to them. They can operate in disconnected mode, meaning that you would pre-provision these cloudlets in the same way that you would provision, like, a private cloud. You do it in advance, which means that you don't need to be connected to the internet to provision them or to operate with them. Finally, they're virtual-machine based—and this will become clear in a second--because you promote flexibility and mobility. Virtual machines you can move from machine from machine, for example, as the mission changes or as things happen.

So, if you take what I just said, and you relate it to the characteristics of edge environments that I just talked about, there's almost, like, a perfect match. I mean, this is exactly what we're trying to solve. So, what makes cyber-foraging especially interesting at the edge--and Suzie, you already talked a little bit about this--is that some of the applications that are useful at the edge are very computation-intensive—face recognition, speech recognition, language translation, air quality analysis—very complex algorithms that are usually run in very complex data centers and things like that.

So, if you divide an application into a very simple client portion—like a very simple client-server application where the client portion is what runs on the mobile device and is a very simple application (taking a picture, taking an air sample, recording some voice)—and then you create the server, which basically has all the other code, the heavy code that has to execute. What happens is that you can offload that server portion from the mobile device to one of these cloudlets. Then you can execute things that you never thought you could execute on a mobile device.

Suzanne: Because you're not taking very much resource to transmit the picture or the voice in comparison to actually processing it.

Grace: Correct. Actually, that's key. It's one of the things that we struggle with in our research, and we like to investigate, which is all that makes sense, as long as it's not expensive to transfer the computation itself. Because, if transferring the computation is going to take up more battery than the computation itself, then you're kind of defeating the purpose.

Suzanne: Okay, that makes sense. So, in your report you talk about this technique of application virtualization as a possible solution to cyber foraging in these kinds of resource-constrained environments. How did you go about evaluating this option, and what did you find to be its strengths and its drawbacks?

Grace: So, as I mentioned earlier, the term *cloudlet* and the term *cyber foraging* were coined by Satya, who is one of our collaborators on campus. When we started investigating systems at the tactical edge, and we found out about his research and what he was doing, we immediately said "this is a perfect match." However, as we started to implement his ideas--because this was all conceptual when we started to work with him--we realized that the process that he had originally envisioned for provisioning, for doing the code offload, was not very appropriate for resource-constrained environments, and let me explain why.

The process that he was using is something called *VM synthesis*. So, what happens with VM synthesis is that there's a portion that is done, let's say, offline--like, before everything happens--and there is a portion that is done at runtime. What VM synthesis does is that, offline, what you do is you create something called a VM overlay. What the VM overlay is, is basically you take a base VM. A base VM is whatever you decided your baseline is within your organization. Then you run that base VM, you install the server portion of what I talked about before, so let's say, the face recognition server, or the voice recognition server. At that point you take a snapshot of the VM. So, now you have a before-state and you have an after-state. If you calculate the binary difference between those two, you have what is called a VM overlay, okay? So, this is what happens offline.

What happens online, or during execution time, is that a person with a smart phone would discover a cloudlet. Then you would send that VM overlay from the mobile device to the cloudlet. It gets processed. Basically it does the opposite, which is it applies that binary difference to the original. Now you have a full base VM, and you start interacting with it. However, the problem that we had was that that VM overlay is quite big because we're talking about a binary difference between two files. It's quite large.

So, when we started doing our experimentation, it just wasn't appropriate because not only were we dealing with the challenge of battery, meaning that the large files took up a lot of battery to transfer. Also, if you have poor network connectivity, you were doing a bunch of retries because you would have to have a stable communication for however long it would take to transfer about 200 megabytes, if you talked about the whole VM overlay. Yes, that's a good bit. So, what we started doing—and actually this is something that we believe has made our research successful—is we're looking at great ideas that are coming from people doing great stuff, for example, Satya. We're looking at the reality of edge environments and trying to find out what those mismatches are and working on those mismatches.

What we focused on in this particular work was: Is there a way in which we could make that overlay, or something similar to an overlay, a lot smaller? So, what we started doing was experimenting with application virtualization, in which what is transferred is not a VM overlay, but it's the virtualized application itself. So, basically, what you try to do is you would try to encapsulate the application with everything it needs inside a package. And our experiments—and you could definitely read all the results in the report—they show a reduction. The new virtualized applications are a fifth of the size of the original. It's not ideal, but it's much better.

Suzanne: 20 percent of 200 megabytes is a lot less than 200 megabytes and a lot faster.

Grace: Yes, I agree. I agree.

Suzanne: Okay. So, this report is one that you worked on with Dominik Messinger, a new name for us at the SEI. He was a student at Carnegie Mellon University. Could you tell us a little bit about Dominik and his involvement? How did you end up collaborating with him on this research?

Grace: Actually, it's very interesting. Because Dominik was at Carnegie Mellon University as the result of an exchange program with the Karlsruhe Institute of Technology in Germany. He was a student of a colleague of mine, Ralf Reussner, who I've known for many years. He said, "I have a very, very bright student who would love to work with you on one of your projects," and it would be great because he could go there via the InterACT [exchange] program. So, that's how we started working with Dominik. We defined our research problem. He was very interested in working on that. We did all the paperwork, and it's been great. Actually, Dominik was our third InterACT student, and we're getting another one this year. So, it's another way of collaborating, not only with CMU, but with other researchers all across the world.

Suzanne: So, we'll get a chance to see some more research coming out from that interaction.

Grace: Yes. Absolutely.

Suzanne: Excellent. So, I'm very impressed that you were able to leverage his research, and I'm sure he was really excited about being able to leverage what your team has done. So, that was very synergistic. Alright, so back to the smart phones and other limited-resource devices. These devices provide both opportunities and risks to our warfighters, right?

Grace: Yes.

Suzanne: So, when you find something out like this about virtualization and its possibilities, what new questions arise, both in terms of opportunities and risks, so that you can think about what's next in terms of your research in this area? Can you give us any insight into that?

Grace: Sure. So, cyber-foraging is only one of three areas that we're currently investigating. We're investigating different ways of doing cyber-foraging, and not only trying to get the size of that file smaller, but also finding, for example, faster ways to provision them. Maybe what we're trying to optimize is not necessarily file size, because imagine a situation where you have your smart phone plugged into an outlet because you have that luxury. Then it doesn't really matter how big the file is, or how long it takes. Or, you have a more relaxed situation. But, maybe there are other things that you want optimized, like you want to be able to provision really quickly. Or, you want to be able to provision only to a cloudlet that is connected to a central data source. So, what we're trying to do is really look at lots of possibilities for cyber-foraging, and trying to create what we call architectural strategies for cyber-foraging, that maybe a particular strategy favors resiliency, or a particular strategy favors survivability, or a particular strategy favors battery consumption. Instead of saying this is the solution for cyber-foraging, [we are] trying to look at a spectrum of solutions. So, that is one area of cyber-foraging, of work that we're doing, and it's what we call cyber-foraging in resource-constrained environments.

A second area is something that is called group context-aware mobile applications, and "group" is really the key word in that sentence because there are lots of context-aware mobile applications. So, every time you use any application to find a restaurant or to do whatever, it is using a part of context which is called *location* to determine what you want to do or where you want to go. Or, for example, Google Places is one that is interesting, because it's constantly tracking you, and it's telling you where traffic is, if you let it, where traffic is. So, basically there are lots of context-aware applications out there, but context is limited to location and time.

It's interesting, because all these devices have— for example, an Android device has 13 built-in sensors, and none of those are exploited. So, the work that we're trying to do in group context-aware mobile applications is to take advantage of all the other sensors that are there on smart phones, for example, and to use not only the context of the mobile device itself as a unit, but if you're deployed in a team—which is usually what happens with first responders and with soldiers, they're never deployed by themselves, they're deployed in teams—can you take advantage of the fact that you have a group of people there that are in potentially close proximity and be able to know not only what your context is, but [also] be able to know what the context of the other people around you is, so you can make better decisions about resource utilization, for example?

So, if we're a group of people, and we're within a certain amount of meters, then it doesn't really make sense for all of us to have GPS active. The person that should have GPS active is the one that has the most battery, and all of the rest should get the coordinates via that person, and not via the GPS signal. So, it's basically forming a very small peer-to-peer network between the team, and being able to share context between everybody, and for the smart phone to make better decisions.

Or, if I'm in a group, and all of a sudden one of the people in my group starts running—and I can detect that easily because I have an accelerometer on my phone—then I can say, "Wow, something is happening," because all of a sudden I see a lot of activity in one of my team members. So, that is the second area of research that we're working on.

Finally, we're working on something called—and this is another mouthful, as Suzie would say—user-configured situational awareness mash-ups. The basic idea behind that is that if you look at the way--especially for soldiers--the way that they have apps on their mobile devices, you have an app for each data feed. So, for example, you have an app for data feed A. You have an app for data feed B. You have an app for Twitter. You have an app for Facebook. And, if you think about it from a situational awareness perspective, it would be great to be able to see all those feeds in a single app. Not only that, but being able to see it on a map, because that is really what I care about.

All those feeds are geo-located, meaning that they have locations. So, for example, I could have a map where I can see every single feed, every single thing that is happening at a particular point in a map. The idea is to give the end user of this smartphone the ability to decide what data sources it wants to see and also to filter that data, or even to say, "I don't want updates every five minutes, I'm okay with updates every hour." By being able to give the user the capability to make decisions on that data, and also to be able to not have to do the filtering locally, but to do the filtering on a server (for example, a cloudlet), so, all that filtering, all that other stuff happens there, and from a user perspective I only see what I've told the server that I want to see.

Suzanne: So, we need to apply that in, you know, commercial. This is your spinout kind of thing. Because there's all these things that want all your data, and they won't let you configure any of it. So, I like that one.

Grace: So, there are two parts of that research. So, one is finding the best ways to visualize geo-located data. The other thing that we're focusing on is, how do we make it easier for people—these are not the end users, but people that are at the command centers—to be able to quickly configure those data sources and say "my people that are going to be out there at this moment are going to need this data source and this data source and this data source," and to quickly set up that server that is going to feed the people.

Suzanne: But also, I can see if you're, say, in a crisis situation--earthquake for example--if I lose a data feed, I need to reconfigure to use a different data feed to give them that same data, because now this one is lost.

Grace: And what we're doing is making that process very easy.

Suzanne: Excellent. Well, it sounds like you have your hands full in terms of the areas of research that you're working on right now. I think I, certainly, and a lot of other people listening, look forward to your results in some of these other important areas. Thank you so much for joining us today, Grace.

Grace: My pleasure.

Suzanne: If you'd like more information on Grace's latest technical report with Dominik, as well as all of the SEI's technical reports and notes, please visit sei.cmu.edu/library/reportspapers.cfm. For more information on the research Grace's team is doing in pervasive mobile computing, please see there our work website at sei.cmu.edu/mobilecomputing /research/index.cfm.

This podcast is available on the SEI website at sei.cmu.edu/podcasts, and on Carnegie-Mellon University's iTunes U site. As always, if you have questions, please don't hesitate to e-mail us at info@sei.cmu.edu. Thank you.