Software Architecture Strategies for Cyber-Foraging Systems

Lewis, G.A.

2016

document version
Publisher's PDF, also known as Version of record

Summary

Cyber-foraging is a technique to enable mobile devices to extend their computing power and storage by offloading computation or data to more powerful servers located in the cloud, or to proximate servers called surrogates. There are two main forms of cyber-foraging. One is computation offload, which is the offload of expensive computation in order to extend battery life and increase computational capability. The second is data staging to improve data transfers between mobile devices and the cloud by temporarily staging data in transit on intermediate surrogates.

One of the main challenges of building cyber-foraging systems is the dynamic nature of the environments that they operate in. For example, the connection to a surrogate may not be available when needed, or may become unavailable during a computation offload or data staging operation. As another example, multiple surrogates may be available but not all have the required capabilities. Adding capabilities to deal with the dynamicity of the environment has to be balanced against resource consumption on the mobile device so as to not defeat the benefits of cyber-foraging. Being able to reason about the behavior of a cyber-foraging system in light of this uncertainty is key to meeting all its desired qualities, which is why software architectures are especially important for cyber-foraging systems.

While there is a large amount of research in cyber-foraging, the reality is that there are not many deployed, operational cyber-foraging systems. As these systems become more prevalent due to their proven benefits, combined with the emergence of micro data centers and edge clouds, a need will arise for guidance on their architecture and development.

This dissertation starts providing this guidance in the form of software architecture strategies for cyber-foraging systems. First, a catalog of architectural tactics for cyber-foraging systems is presented. These tactics were validated through three case studies and can be used by software architects to achieve system qualities such as resource optimization, fault tolerance, scalability, and security, while conserving resources on the mobile device. Secondly, a characterization of usage contexts for cyber-foraging, defined in terms of functional and non-functional requirements is presented in order to understand the usage contexts that benefit the most from cyber-foraging. Finally, a decision model for cyber-foraging systems is presented that maps functional and non-functional requirements for cyber-foraging systems to the set of architectural tactics. The end goal is to help software architects extend their design reasoning towards cyber-foraging as a way to support the mobile applications of the present and the future, while understanding the effects of their decisions.