

Chapter III

Adaptive Mobile Applications: Experience With an Approach Based on Mobile Code

Thomas Kunz
Carleton University, Canada

INTRODUCTION

The convergence of two technological developments has made mobile computing a reality. In the last few years, developed countries spent large amounts of money to install and deploy wireless communication facilities. Originally aimed at telephone services (which still account for the majority of usage), the same infrastructure is increasingly used to transfer data. The second development is the continuing reduction in size of computer hardware, leading to portable computation devices such as laptops, palmtops, or functionally enhanced cell phones. Given current technology, a user can run a set of applications on a portable device and communicate over a variety of communication links, depending on his/her current location. For example, the user can access the wired corporate LAN at 10 Mbps or higher in the office. Roaming in the building, connectivity is provided by an indoor wireless LAN at 1-2 Mbps. Outdoors, connectivity is provided by cellular wireless-IP networks, providing bandwidths of a few tens of kbps. Furthermore, the sets of services available in each location will generally differ.

Similar discrepancies will also persist in future wireless networks, such as the ones under study by the International Telecommunication Union (2000) and the European Union's ACTS program (ACTS, Mobility, 1997). Unlike second-generation cellular networks, future cellular systems will cover an area with a variety of nonhomogeneous cells that may overlap. This allows the network operators to tune the system layout to subscriber density and subscribed services. Cells of different sizes will offer widely varying bandwidths: very high bandwidths with low error rates in pico-cells, very low bandwidths with higher error rates in macro-cells. Again, depending on the current location, the sets of available services might also differ. It is generally argued that applications should "adapt" to the current environment, for example, by filtering and compressing data or by changing the functionality offered to the user, see Badrinath, Acharya, & Imielinski, 1994; Welling & Badrinath, 1997; Zenel & Duchamp, 1997. Some researchers even argue that all future applications, not just the ones intended for execution on mobile devices, will have to be able to adapt to changing requirements and changing implementation environments on time scales from microseconds to years (Kavi, Browne & Tripathi, 1999). Our research addresses this problem by proposing an approach to application adaptivity based on code mobility. This chapter explains the basic concept underlying our approach, describes our proposed architecture, and explains our initial experience in developing this architecture.

The alternative to adaptive applications is multiple functionally identical or similar binaries, tuned for specific environments. This is an inferior solution, for a number of reasons. The user of a portable device has to install and maintain multiple applications, which is a drain on the limited storage capabilities typically found on those devices. It also potentially results in different user interfaces and causes high software development overheads when developing the "same" mobile application multiple times. Finally, it forces the user to identify the current execution conditions and select the "right" application.

Our work is based on offloading computationally intensive application components to more powerful servers in the access network. Ideally, this "load sharing" between a portable device and more powerful machines would occur transparent to the user, at runtime, taking into account the currently available resources in the execution environment. To realize this goal, a support infrastructure is required to monitor the available resources, to decide how best to utilize these resources, and that deals with user mobility. The following sections will motivate our approach, introduce a suitable architecture, and explain what experience we collected to date in realizing various components of this architecture.

17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/adaptive-mobile-applications/26016

Related Content

Analyzing the Disruptive Potential in the Telecommunications Industry

Stefan Hüsig (2009). *Handbook of Research on Telecommunications Planning and Management for Business* (pp. 108-124).

www.irma-international.org/chapter/analyzing-disruptive-potential-telecommunications-industry/21660

A Survey of FPGA Dynamic Reconfiguration Design Methodology and Applications

Ming Liu, Zhonghai Lu, Wolfgang Kuehnand Axel Jantsch (2012). *International Journal of Embedded and Real-Time Communication Systems* (pp. 23-39).

www.irma-international.org/article/survey-fpga-dynamic-reconfiguration-design/66430

Spectral Efficiency Self-Optimization through Dynamic User Clustering and Beam Steering

Md Salik Parwez, Hasan Farooq, Ali Imranand Hazem Refai (2017). *Big Data Applications in the Telecommunications Industry* (pp. 137-155).

www.irma-international.org/chapter/spectral-efficiency-self-optimization-through-dynamic-user-clustering-and-beam-steering/174281

Design Flow for Silicon Chip Implementing Novel Platform Architecture for Wireless Communication

Prabhat Avasare, Jeroen Declerck, Miguel Glassee, Amir Amin, Erik Umans, Praveen Raghavanand Martin Palkovic (2013). *International Journal of Embedded and Real-Time Communication Systems* (pp. 42-63).

www.irma-international.org/article/design-flow-silicon-chip-implementing/77309

Real-Time Query Processing on Live Videos in Networks of Distributed Cameras

Rui Peng, Alex J. Avedand Kien A. Hua (2010). *International Journal of Interdisciplinary Telecommunications and Networking* (pp. 27-48).

www.irma-international.org/article/real-time-query-processing-live/40961