Mobile Devices as Interfaces for Steering Cloud-Based High-Performance Computations

Young-Woo Kwon
Dept. of Computer Science
Utah State University, USA
young.kwon@usu.edu

Eli Tilevich
Dept. of Computer Science
Virginia Tech, USA
tilevich@cs.vt.edu

Abstract

As mobile devices have been steadily overtaking the personal computer as a primary computing platform, mobile applications deliver increasingly complex functionality. Furthermore, for next generation mobile applications to be proactive in their functionality, they need to be able to collect and process massive amounts of context-sensitive information on the fly. Leveraging high-end computing resources offers a promising avenue to address these emerging computational needs of mobile computing, both improving performance and saving battery power. These computational resources can now be conveniently accessed via standardized cloud-based interfaces. However, several research challenges must be addressed to be able to seamlessly use mobile devices as convenient interfaces for steering cloud-based high-performance computations. This position paper presents our view of the research agenda that must be followed to achieve these objectives as well as reports on our initial efforts in this endeavor.

Categories and Subject Descriptors H.3.3 [*Systems and Software*]: Distributed Systems; C.2.4 [*Distributed Systems*]: Distributed applications

General Terms Design

Keywords HPC, Mobile Computing, Cloud Computing

1. Introduction

We foresee that next-generation mobile applications will be proactive in their functionality and thus will collect and process massive amounts of context-sensitive information. To improve performance and save battery power, mobile applications will need to be able to leverage high-end computing resources including cloud infrastructures and computational science & engineering software, even though adopting the legacy code in computational science & engineering software can be non-trivial. We see automated refactoring techniques as a powerful tool [2], capable of streamlining the process of adopting extant computational science & engineering codebases for next-generation mobile applications.

Leveraging high-end computing resources in mobile applications presents both challenges and opportunities. The challenges lie in the necessity to evolve the legacy computational science codebases for the requirements of next-generation mobile applications. The scientific and engineering domains are characterized by massive volumes of legacy code that needs to be maintained and evolved. The key opportunities that we identify are in applying automated software refactoring techniques and tools for the needs of computational science & engineering in mobile applications. State-of-the-art software refactoring techniques and tools hold a huge promise as a means of increasing programmer productivity in mobile software development. This position paper describes our vision for leveraging high-end computing resources for next-generation mobile applications through automated software refactoring and tools.

2. Our Vision

In the following discussion, we outline our vision for next-generation mobile applications.

2.1 Leveraging HPC Resources

Our vision for the technological future is that the software engineering research agenda will be dominated by the needs to support next-generation applications such as a Personal Mobile Assistant (PMA). These applications will collect and process massive amounts of context-sensitive information on the fly, and hence will need to be able to leverage high-end computing resources for their efficient execution.

We also observe that leveraging high-end computing resources in the mobile domains presents both challenges and opportunities. The challenges lie in the necessity to adapt the legacy codebases for the requirements of future comput-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

MobileDeLi'15, October 26, 2015, Pittsburgh, PA, USA
© 2015 ACM. 978-1-4503-3906-3/15/10...$15.00
http://dx.doi.org/10.1145/2846661.2846676

ing systems, which remains an elusive goal despite multiple targeted research efforts. The primary opportunities that we identify is in developing and applying automated refactoring techniques that can streamline the process of adapting both existing mobile and computational science/engineering codebases, so that they can be harmoniously bridged together to enable the next generation of PMA applications expeditiously and cost-efficiently.

Hence, in summary, the software engineering research challenges that need to be addressed to realize this vision are (1) enabling mobile applications to seamlessly access high-end computational resources, (2) developing effective refactoring techniques that can adapt the massive volumes of legacy code, both mobile and high-end computing, for the new requirements and execution environments.

2.1.1 Automated Software Refactoring

In the last decade, the productivity of desktop application programmers has been significantly increased due to the development of software tools such as Eclipse, Microsoft Visual Studio, NetBeans, and IntelliJ IDEA. Software refactoring is the process of changing source code without changing its semantics, in order to improve non-functional aspects of the software. Aside from readability and maintainability, many other aspects (e.g., performance, responsiveness) can benefit from refactoring technologies.

We have been working in the area of automated refactoring for utilizing high-end computing resources. We introduced a novel cloud offloading approach that leverages the resources of cloud-based servers to execute portions of a mobile applications functionality [1]. Then, we created novel refactoring techniques that facilitate the process of transitioning existing applications to make use of cloud-based resources [2].

2.1.2 Offloading Local Computations

The idea of computation offloading has been widely explored to improve the performance of local applications [1]. The concept behind computation offloading is relatively simple: When a local application identifies some functionality as "hot" (that is, computation intensive), it transforms to a distributed application and offloads the required data to a cloud server, which executes the hot functionality and then transfers back the results. Recently, cloud offloading has gained prominence to reduce the energy consumption of mobile applications, whose functionality sets and computational needs expand continuously. Furthermore, developments in mobile hardware and software technologies increasingly enable the integration of complex computational science algorithms into mobile applications. However, their functionality is often limited by the shortage of mobile hardware resources, including the CPU, storage, and battery. Computation offloading can help address these limitations.

2.1.3 Adopting Legacy Code in the Cloud

The scientific and engineering domains are characterized by massive volumes of legacy code that needs to be maintained and evolved. One of the evolutionary thrusts is adopting the legacy code for modern computing environments. Cloud computing has been shown to deliver tangible benefits in the enterprise domain by providing massive computing resources at scale and at cost. Leveraging cloud computing for the needs of computational science & engineering has great potential benefit, even though adopting the legacy code for the cloud computing interfaces can be non-trivial [2].

2.1.4 Dynamically Adapting Execution Patterns

The quality (e.g., performance, energy-efficiency) of mobile applications accessing cloud-based high-end computing resources highly depends on the underlying hardware and execution environments (e.g., latency, bandwidth, network connectivity such as WiFi or cellular). Thus, our vision hinges on the ability to adaptively change execution patterns at runtime based on the current environmental conditions to further improve the QoS of mobile applications. To that end, middleware presents a promising target by dynamically adapting various facets of distributed functionality (e.g., data compression and transmission) and mobile execution (e.g., limiting or deferring executions) in response to fluctuations in runtime conditions [3].

3. Conclusion

It is commonly the case that many application developers have limited experience with computational science software, which is created, maintained, and used by domain experts (i.e., scientists) and runs in high-end computing resources. Thus, the dichotomy between the intrinsic difficulty of high-end computing and the non-expert status of many application developers is a major bottleneck in developing proactive mobile applications safely and efficiently. Thus, automated software refactoring and tools will be able to keep improving the productivity of mobile software developers as well as the performance and energy efficiency of next-generation mobile software.

References

[1] Y.-W. Kwon and E. Tilevich. Energy-efficient and fault-tolerant distributed mobile execution. In *Proceedings of the 32nd International Conference on Distributed Computing Systems (ICDCS)*, June 2012.

[2] Y.-W. Kwon and E. Tilevich. Cloud refactoring: Automated transitioning to cloud-based services. *Automated Software Engineering Journal*, 2014.

[3] Y.-W. Kwon and E. Tilevich. Configurable and adaptive middleware for energy-efficient distributed mobile computing. In *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services (MobiCase 2015)*, November 2014.