Chapter 4
Intelligent Management of Mobile Systems Through Computational Self-Awareness

Bryan Donyanavard
University of California, Irvine, USA

Amir M. Rahmani
University of California, Irvine, USA

Axel Jantsch
TU Wien, Austria

Onur Mutlu
Swiss Federal Institute of Technology in Zurich, Switzerland

Nikil Dutt
University of California, Irvine, USA

ABSTRACT

Runtime resource management for many-core systems is increasingly complex. The complexity can be due to diverse workload characteristics with conflicting demands, or limited shared resources such as memory bandwidth and power. Resource management strategies for many-core systems must distribute shared resource(s) appropriately across workloads, while coordinating the high-level system goals at runtime in a scalable and robust manner. In this chapter, the concept of reflection is used to explore adaptive resource management techniques that provide two key properties: the ability to adapt to (1) changing goals at runtime (i.e., self-adaptivity) and (2) changing dynamics of the modeled system (i.e., self-optimization). By supporting these self-awareness properties, the system can reason about the actions it takes by considering the significance of competing objectives, user requirements, and operating conditions while executing unpredictable workloads.

DOI: 10.4018/978-1-7998-7156-9.ch004

INTRODUCTION

Battery powered-devices are the most ubiquitous computers in the world. Users expect the devices to support high performance applications running on same device, sometimes at the same time. The devices support a wide range of applications, from interactive maps and navigation, to web browsers and email clients. In order to meet the performance demands of the complex workloads, increasingly powerful hardware platforms are being deployed in battery-powered devices. These platforms include a number of configurable knobs that allow for a tradeoff between power and performance, e.g., dynamic voltage and frequency scaling (DVFS), core gating, idle cycle injection, etc. These knobs can be set and modified at runtime based on workload demands and system constraints. Heterogeneous manycore processors (HMPs) have extended this principle of dynamic power-performance tradeoffs by incorporating single-ISA, architecturally differentiated cores on a single processor, with each of the cores containing a number of independent tradeoff knobs. All of these configurable knobs allow for a large range of potential tradeoffs. However, with such a large number of possible configurations, HMPs require intelligent runtime management in order to achieve application goals for complex workloads while considering system constraints. Additionally, the knobs may be interdependent, so the decisions must be coordinated. In this chapter, we explore the use of computational self-awareness to address challenges of adaptive resource management in mobile multiprocessors.

Computational Self-awareness

Self-aware computing is a new paradigm that does not strictly introduce new research concepts, but unifies overlapping research efforts in disparate disciplines (Lewis et al., 2016). The concept of self-awareness from psychology has inspired research in autonomous systems and neuroscience, and existing research in fields such as adaptive control theory support properties of self-awareness. This chapter addresses key challenges for achieving computational self-awareness that can make the design, maintenance and operation of complex, heterogeneous systems adaptive, autonomous, and highly efficient. Computational self-awareness is the ability of a computing system to recognize its own state, possible actions and the result of these actions on itself, its operational goals, and its environment, thereby empowering the system to become autonomous (Jantsch et al., 2017). An infrastructure for system introspection and reflective behavior forms the foundation of self-aware systems.

Reflection

Reflection can be defined as *the capability of a system to reason about itself and act upon this information* (Smith, 1982). A reflective system can achieve this by maintaining a representation of itself (i.e., a self-model) within the underlying system, which is used for reasoning. Reflection is a key property of self-awareness. Reflection enables decisions to be made based on both *past* observations, as well as predictions made from past observations. Reflection and prediction involve two types of models: (1) a self-model of the subsystem(s) under control, and (2) models of other policies that may impact the decision-making process. Predictions consider *future* actions, or events that may occur before the next decision, enabling "what-if" exploration of alternatives. Such actions may be triggered by other resource managers running with a shorter period than the decision loop. The top half of Figure 1 shows prediction enabled through reflection that can be utilized in the decision making process of a feedback loop. The