Application Workload Modelling via Run-Time Performance Statistics

Subayal Khan, VTT Technical Research Centre of Finland, Oulu, Finland
Jukka Saastamoinen, VTT Technical Research Centre of Finland, Oulu, Finland
Jyrki Huusko, VTT Technical Research Centre of Finland, Oulu, Finland
Juha-Pekka Soininen, VTT Technical Research Centre of Finland, Oulu, Finland
Jari Nurmi, Department of Computer Systems, Tampere University of Technology, Tampere, Finland

ABSTRACT

Modern mobile nomadic devices for example internet tablets and high end mobile phones support diverse distributed and stand-alone applications that were supported by single devices a decade back. Furthermore the complex heterogeneous platforms supporting these applications contain multi-core processors, hardware accelerators and IP cores and all these components can possibly be integrated into a single integrated circuit (chip). The high complexity of both the platform and the applications makes the design space very complex due to the availability of several alternatives. Therefore the system designer must be able to quickly evaluate the performance of different application architectures and implementations on potential platforms. The most popular technique employed nowadays is termed as system-level-performance evaluation which uses abstract workload and platform capacity models. The platform capacity models and application workload models reside at a higher abstraction-level. The platform and application workload models can be instantiated with reduced modeling effort and also operate at a higher simulation speed. This article presents a novel run-time statistics based application workload model extraction and platform configuration technique. This technique is called platform COnfiguration and woRKload generatIoN via code instrumeNtation and performAnce counters (CORINNA) which offers several advantages over compiler based technique called ABSINTH, and also provides automatic configuration of the platform processor models for example cache-hits and misses obtained during the application execution.

Keywords: Abstract Instruction Extraction Helper (ABSINTH), Application Workload, Architectural Exploration, COnfiguration and woRKload generatIoN via code instrumeNtation and performAnce (CORINNA), Performance Evaluation, Platform Capacity Model, Workload Model

DOI: 10.4018/jertcs.2013040101
INTRODUCTION

Majority of many digital technology features for example voice, texts, video, pictures, broadcasts, presentation, streaming media, global connectivity and personalized services are currently supported by the highly complex modern high-end nomadic standalone devices. We are indeed seeing the dawn for digital convergence (Covell, 1999).

These devices support diverse applications (Nooergaard, 2005) some of which might be executing sequentially, some concurrently, some independently while others might run interactively. As a result, in the design of these products, the challenges faced in the design of both conventional distributed systems and the resource-constrained real-time systems will apply. Furthermore, the current System on Chip based designs of high-end mobile devices employ multiple processor cores. Some of these cores are general purpose for applications and others are dedicated to multimedia processing.

The system complexity has increased tremendously during the past decade due to the aforementioned advances in technology. The embedded system high end products are usually constraint with strict energy, resource as well as real-time budgets. Apart from that, the platform and application designers have multitude of design alternatives. For faster architectural exploration of the design space, novel systematic approaches are needed. The availability of efficient methods and tools are of pivotal importance so as to steer the architectural exploration trajectory towards the optimal or near optimal design. This will require a few iterations of the application-platform mappings and co-simulations, considerably reducing the design time and time to market.

Different performance modeling approaches have been invented which employ virtual platform models and application models at different levels of abstraction and refinement. SPADE employs a trace driven, system-level co-simulation approach for both the applications and the platform (Lieverse, van der Wolf, Visser & Deprettere, 2001). Artemis is an extension of SPADE which further adds the concepts of virtual processors and bounded buffers to it (Pimentel & Erbas, 2006). The TAPES adopts a different approach and abstracts the functionalities by latencies. The application code is not run and only the interaction between the encompassing sub-functions on the architecture is covered (Wild, Herkersdorf & Lee, 2006). Mesh is another landmark approach which uses resources, software and scheduler/protocol models, each residing at a different abstraction level. From an implementation perspective, they are modeled by software threads running on the target host (Paul, Thomas & Cassidy, 2005).

Application workload models are of pivotal importance in any system-level performance evaluation approach and must provide a proper abstraction of the actual application workloads. Application workload means the computational workload of an application and is closely associated to the set of computer benchmarks employed to measure and the computational effectiveness of different platforms in architectural simulations (Kreku et al., 2008).

ABSOLUT performance evaluation approach follows the Y-chart model consisting of application workload and platform model (Kreku et al., 2008). Applications workload models are automatically extracted from the application source code using a compiler based approach called Abstract Instruction Extraction Helper (ABSINTH) (Kreku, Tiensyrjä & Vanmeerbeck, 2010). The platform models are cycle approximate, operating at transaction-level. SystemC is used as the modeling language for both platform and application. The application models are mapped to platform models for co-simulation to extract performance results. The salient features of ABSOLUT include the ease of modeling, usage of freely available SystemC library, ABSOLUT platform component library and automated application workload extraction (Kreku et al., 2008).
Related Content

Joint Validation of Application Models and Multi-Abstraction Network-on-Chip Platforms
www.irma-international.org/chapter/joint-validation-application-models-multi/65608/

Asymmetric Geographic Forwarding: Exploiting Link Asymmetry in Location Aware Routing
www.irma-international.org/article/asymmetric-geographic-forwarding/60154/

A SEEP Protocol Design Using 3BC, ECC(F2m) and HECC Algorithm
www.irma-international.org/chapter/seep-protocol-design-using-3bc/6050/

Users’ Perceptions of Wireless Networks Usage
www.irma-international.org/article/users-perceptions-wireless-networks-usage/44967/

Checkpointing SystemC-Based Virtual Platforms
www.irma-international.org/article/checkpointing-systemc-based-virtual-platforms/60152/