J. Parallel Distrib. Comput. 74 (2014) 1872–1890

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

A queueing theoretic approach for performance evaluation of low-power multi-core embedded systems

Arslan Munir [a,*], Ann Gordon-Ross [b,c], Sanjay Ranka [d], Farinaz Koushanfar [a]

[a] Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA
[b] Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA
[c] NSF Center for High-Performance Reconfigurable Computing (CHREC) at the University of Florida, USA
[d] Department of Computer and Information Science and Engineering at the University of Florida, Gainesville, FL, USA

HIGHLIGHTS

- Queueing theory-based modeling technique for evaluating multi-core architectures.
- Enables quick and inexpensive architectural evaluation.
- Architectural evaluation for workloads with any computing requirements.
- Can be used for performance per watt & performance per unit area characterizations.
- Provides insights about shared last-level caches (LLCs) orchestration.

ARTICLE INFO

Article history:
Received 18 June 2012
Received in revised form
6 July 2013
Accepted 26 July 2013
Available online 11 August 2013

Keywords:
Multi-core
Low-power
Embedded systems
Queueing theory
Performance evaluation

ABSTRACT

With Moore's law supplying billions of transistors on-chip, embedded systems are undergoing a transition from single-core to multi-core to exploit this high transistor density for high performance. However, the optimal layout of these multiple cores along with the memory subsystem (caches and main memory) to satisfy power, area, and stringent real-time constraints is a challenging design endeavor. The short *time-to-market* constraint of embedded systems exacerbates this design challenge and necessitates the architectural modeling of embedded systems to reduce the time-to-market by expediting target applications to device/architecture mapping. In this paper, we present a queueing theoretic approach for modeling multi-core embedded systems that provides a quick and inexpensive performance evaluation both in terms of time and resources as compared to the development of multi-core simulators and running benchmarks on these simulators. We verify our queueing theoretic modeling approach by running SPLASH-2 benchmarks on the SuperESCalar simulator (SESC). Results reveal that our queueing theoretic model qualitatively evaluates multi-core architectures accurately with an average difference of 5.6% as compared to the architectures' evaluations from the SESC simulator. Our modeling approach can be used for performance per watt and performance per unit area characterizations of multi-core embedded architectures, with varying number of processor cores and cache configurations, to provide a comparative analysis.

1. Introduction and motivation

With Moore's law supplying billions of transistors on-chip, embedded systems are undergoing a paradigm shift from single-core to multi-core to exploit this high transistor density for high performance. This paradigm shift has led to the emergence of diverse multi-core embedded systems in a plethora of application domains (e.g., high-performance computing, dependable computing, mobile computing, etc.). Many modern embedded systems integrate multiple cores (whether homogeneous or heterogeneous) on-chip to satisfy computing demand while maintaining design constraints (e.g., energy, power, performance, etc.). For example, a 3G mobile handset's signal processing requires 35–40 Giga operations per second (GOPS). Considering the limited energy of a mobile handset battery, these performance levels must be met with a power dissipation budget of approximately 1W, which translates to a performance efficiency of 25 mW/GOP or 25 pJ/operation for the 3G receiver [4]. These demanding and competing power–performance requirements make modern embedded system design challenging.

* Corresponding author.
E-mail addresses: arslan@rice.edu, arslanmn@gmail.com (A. Munir), ann@ece.ufl.edu (A. Gordon-Ross), ranka@cise.ufl.edu (S. Ranka), farinaz@rice.edu (F. Koushanfar).

0743-7315/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jpdc.2013.07.003

Increasing customer expectations/demands for embedded system functionality has led to an exponential increase in design complexity. While industry focuses on increasing the number of on-chip processor cores to meet the customer performance demands, embedded system designers face the new challenge of optimal layout of these processor cores along with the memory subsystem (caches and main memory) to satisfy power, area, and stringent real-time constraints. The short *time-to-market* (time from product conception to market release) of embedded systems further exacerbates design challenges. Architectural modeling of embedded systems helps in reducing the time-to-market by enabling fast application-to-device mapping since identifying an appropriate architecture for a set of target applications significantly reduces the design time of an embedded system. To ensure the timely completion of an embedded system's design with sufficient confidence in the product's market release, design engineers must make tradeoffs between the abstraction level of the system's architecture model and the attainable accuracy.

Modern multi-core embedded systems allow processor cores to share hardware structures, such as last-level caches (LLCs) (e.g., level two (L2) or level three (L3) cache), memory controllers, and interconnection networks [12]. Since the LLC's configuration (e.g., size, line size, associativity) and the layout of the processor cores (on-chip location) has a significant impact on a multi-core embedded system's performance and energy, our work focuses on performance and energy characterization of embedded architectures based on different LLC configurations and layout of the processor cores. Though there is a general consensus on using private level one (L1) instruction (L1-I) and data (L1-D) caches in embedded systems, there has been no dominant architectural paradigm for private or shared LLCs. Since many embedded systems contain an L2 cache as the LLC, we focus on the L2 cache, however, our study can easily be extended for L3 caches and beyond as LLCs.

Since multi-core benchmark simulation requires significant simulation time and resources, a lightweight modeling technique for multi-core architecture evaluation is crucial [11]. Furthermore, simulation-driven architectural evaluation is based on specific benchmarks and consequently only provides performance information for programs similar to the benchmarks. A well-devised modeling technique can model diverse workloads and thus enables performance evaluation for workloads with any computing requirements. Previous work presents various multi-core system models, however, these models become increasingly complex with varying degrees of cache sharing [33]. Many of the previous models assumed that sharing amongst processor cores occurred at either the main memory level or the processor cores all shared the same cache hierarchy, however, multi-core embedded systems can have an L2 cache shared by a subset of cores (e.g., Intel's six-core Dunnington processor has L2 caches shared by two processor cores). We leverage for the first time, to the best of our knowledge, queueing network theory as an alternative approach for modeling multi-core embedded systems for performance analysis (though queueing network models have been studied in the context of traditional computer systems [31]). Our queueing network model approach allows modeling the layout of processor cores (processor cores can be either homogeneous or heterogeneous) with caches of different capacities and configurations at different cache levels. Our modeling technique only requires a high-level workload characterization of an application (i.e., whether the application is processor-bound (requiring high processing resources), memory-bound (requiring a large number of memory accesses), or mixed).

Our main contributions in this paper are:

- we present a novel, queueing theory-based modeling technique for evaluating multi-core embedded architectures that does not require architectural-level benchmark simulation. This modeling technique enables quick and inexpensive architectural evaluation, with respect to design time and resources, as compared to developing and/or using the existing multi-core simulators and running benchmarks on these simulators. Based on a preliminary evaluation using our models, architecture designers can run targeted benchmarks to further verify the performance characteristics of selected multi-core architectures (i.e., our queueing theory-based models facilitate early design space pruning).
- our queueing theoretic approach enables the architectural evaluation for synthetic workloads with any computing requirements characterized probabilistically. We also propose a method to quantify computing requirements of real benchmarks probabilistically. Hence, our modeling technique can provide performance evaluation for workloads with any computing requirements as opposed to simulation-driven architectural evaluation that can only provide performance results for specific benchmarks.
- our queueing theoretic modeling approach can be used for performance per watt and performance per unit area characterizations of multi-core embedded architectures, with varying number of processor cores and cache configurations, to provide a comparative analysis. For performance per watt and performance per unit area computations, we calculate chip area and power consumption for different multi-core embedded architectures with a varying number of processor cores and cache configurations.

We point out that although queueing theory has been used in the literature for performance analysis of multi-disk and pipelined systems [13,18,31], we for the first time, to the best of our knowledge, apply queueing theory-based modeling and performance analysis techniques to multi-core embedded systems. Furthermore, we for the first time develop a methodology to synthesize workloads/benchmarks on our queueing theoretic multi-core models based on probabilities that are assigned according to workload characteristics (e.g., processor-bound, memory-bound, or mixed) and cache miss rates. We verify our queueing theoretic modeling approach by running SPLASH-2 multi-threaded benchmarks on the SuperESCalar simulator (SESC). Results reveal that our queueing theoretic model qualitatively evaluates multi-core architectures accurately with an average difference of 5.6% as compared to the architectures' evaluations from the SESC simulator. The SESC simulation results validate our queueing theoretic modeling approach as a quick and inexpensive architectural evaluation method.

Our queueing theoretic approach can be leveraged for *early design space pruning* by eliminating infeasible architectures in very early design stages, which reduces the number of lengthy architectural evaluations when running targeted benchmarks in later design stages. Specifically, our approach focuses on the qualitative comparison of architectures in the early design stage and not the quantitative comparison of architectures for different benchmarks. Our model is designed to operate using *synthetic workloads* that a designer can categorize for an expected behavior, such as processor or memory-bound workloads, along with an estimate of the expected cache miss rates. The synthetic workloads preclude the need to obtain benchmark-specific statistics from an architecture-level simulator. Furthermore, the cache miss rates are estimates, and thus are not required to be the exact miss rates for any specific benchmark. Our discussions in Section 4.2 regarding statistics obtained from an architecture-level simulator only explains how a real workload can be represented with our queueing theoretic model and is not required for synthetic workloads.

Our investigation of performance and energy for different cache miss rates and workloads is significant because cache miss rates and workloads can significantly impact the performance and energy of an embedded architecture. Furthermore, cache miss rates