

## Special section on advances in methods for adaptive multicore systems

Amir-Mohammad Rahmani · Pasi Liljeberg ·  
Juha Plosila · Hannu Tenhunen

Published online: 3 June 2014  
© Springer Science+Business Media New York 2014

Recent trends in the microprocessor industry have important ramifications for the design of the next generation of high-performance computing systems. By increasing number of cores, it is possible to improve the performance while keeping the power consumption unchanged. This trend has reached all areas of computing, ranging from high-performance supercomputers to embedded multicore systems. Multicore systems need to be able to reconfigure themselves adaptively by monitoring their own condition and the surrounding environment to adapt themselves to different scenarios and performance-power requirements. Runtime monitoring becomes crucial in the emerging and future parallel and distributed multicore systems due to increase in variations and fault occurrences, thermal issues as well as due to the need for various adaptive optimizations.

In this special section, the Guest Editors have put together some of the new developments and trends in the context of monitoring and reconfiguration of parallel and distributed multicore systems. We received 27 submissions and a total of nine were accepted. The high level of competition has led to the selection of top-level contributions covering a wide spectrum of topics in the domain of adaptive multicore systems including transactional memory controlling, cache vulnerability mitigation, task migration, power optimization, job scheduling, and load balancing.

Ansari, in his article entitled “Weighted Adaptive Concurrency Control for Software Transactional Memory”, presents an approach which exploits variance in available parallelism between threads. Transactional memory programs may have dynamic available parallelism, which is defined as the number of transactions that can be

---

A.-M. Rahmani (✉) · P. Liljeberg · J. Plosila  
University of Turku, Turku, Finland  
e-mail: amirah@utu.fi

H. Tenhunen  
Royal Institute of Technology, Stockholm, Sweden

committed concurrently. The basic idea is to consider the case where individual threads exhibit dynamic available parallelism which was not considered in prior work. Deactivating threads with low available parallelism, and vice versa, may improve speedup and reduce wasted work further. In particular, the author introduces weighted adaptive concurrency control to exploit the variance in available parallelism between threads.

Maghsoudloo et al., in their article entitled “Cache Vulnerability Mitigation Using an Adaptive Cache Coherence Protocol”, propose an adaptive cache coherence protocol to improve the reliability of caches against soft errors in shared-memory multi-core processors. In particular, the authors conduct the protocol based on a comprehensive study and analysis intended to determine the effects of cache coherence protocols on the characteristics of cache memories. The outcomes of their analysis indicate that differences in handling dirty data items play an important role to make distinction in favor of or against a cache coherence protocol. Based on their results, the proposed protocol tries to enhance the reliability of caches by means of sharing management.

Request for more computation power steadily forces designers to provide more powerful processors using more number of cores on a single chip. The increasing complexity of processors leads to higher integration density, power density, and temperature. For avoiding thermal emergencies, various dynamic thermal management techniques have been presented. In this context, Salami et al., in their paper entitled “Proactive Task Migration with a Self-Adjusting Migration Threshold for Dynamic Thermal Management of Multi-Core Processors”, present a novel online self-adjusting temperature threshold schema for dynamic thermal management to minimize both average and peak temperature with very low performance overhead. Their proposed algorithm adjusts migration threshold according to work-load and hardware platforms.

Piga et al., in their article entitled “Adaptive Global Power Optimization for Web Servers”, investigate power and performance trade-offs for web servers on a state-of-the-art, high-density, power-efficient SeaMicro SM15k cluster by AMD. They relied on the concept of virtual power states, a combination of CPU utilization rate to the P/C power states available in modern processors, and on their global optimization algorithm called slack recovery, to deploy an adaptive global power management system in a production environment. The main contributions of their article are twofold. First, it presents the slack recovery algorithm deployed on a real cluster, composed of 25 SeaMicro nodes. Second, it proposes a novel mechanism to control utilization rates in each server, a key aspect on the power/performance optimization system which enables the implementation of the virtual power states concept in practice.

Job scheduling strategies in multi-processing systems aim to minimize waiting times of jobs while satisfying user requirements in terms of number of execution units. However, the lack of flexibility in the requests leaves the scheduler a reduced margin of action for scheduling decisions. Many of such decisions consist on just moving ahead some specific jobs in the wait queue. Utrera et al., in their article entitled “Scheduling parallel jobs on multicore clusters using CPU oversubscription” propose a job scheduling strategy that improves the overall performance and maximizes resource utilization by allowing jobs to adapt to variations in the load through CPU oversubscription and backfilling.

Spatial locality of task execution is becoming important in future hardware platforms since the number of cores are steadily increasing. The large amount of cores requires an intelligent power manager and the high chip and core density requires increased thermal awareness to avoid thermal hotspots on the chip. Based on this, Holmbacka et al., in their article entitled “A Task Migration Mechanism for Distributed Many-Core Operating Systems” present a lightweight task migration mechanism explicitly for distributed operating systems running on many-core platforms. As the distributed OS runs one scheduler on each core, the tasks are migrated between OS kernels within the same shared memory platform. In their article, the benefits, such as performance and energy efficiency, of task migration is achieved by re-locating running tasks on the most appropriate cores and keeping the overhead of executing such a migration sufficiently low. They investigate the overhead of migrating tasks on a distributed OS running both on a bus based platform and a many-core networks-on-chip.

Multi-core computing has gone mobile. Managing power consumption within energy-constrained mobile devices demands low-power architectures to increase battery lifespan. One of the promising solutions ordered today by microprocessor architects is hybrid microprocessors that integrate different core architectures on a single die and that are equipped with dynamic frequency-scaling techniques. In this context, Marowka in his article entitled “Maximizing Energy-Saving of Dual-Architecture Processors using DVFS” presents analytical models based on an energy consumption metric to analyze the impact of dynamic frequency scaling on the energy consumption of various architectural design choices for hybrid-architecture chips. He also analyzes the power consumption implications of different processing schemes and various chip configurations.

On today’s multiprocessor systems, simultaneously executing multi-threaded applications contend for cache space and CPU time. This contention can be managed by changing application thread count. Moore et al., in their article entitled “Building and Using Application Utility Models to Dynamically Choose Thread Counts” describe a technique to configure thread count using utility models. A utility model predicts application performance given its thread count and other workload thread counts. In their article, utility models are used online by a system policy to dynamically configure applications’ thread counts. They present a policy which uses the models to maximize throughput while maintaining QoS.

Adaptive routing algorithms improve network performance by distributing traffic over the whole network. However, they require congestion information to facilitate load-balancing. Finally, Farahnakian et al., in their article “Adaptive Load Balancing in Learning-based Approaches for Many-core Embedded Systems” propose a learning method based on dual reinforcement learning approach to provide local and global congestion information. This information can be dynamically updated according to the changing traffic condition in the network by propagating data and learning packets. They utilize a congestion detection method which updates the learning rate according to the congestion level.

We sincerely hope the reader will find this special issue useful and that it will inspire further research in this very important area of electronic and computer system design. We would like to thank all authors who submitted papers to this special issue. Special

---

thanks go to the referees for their time and diligence during the review process and for providing us with high-quality reviews. Finally, we would like to thank Prof. Hamid R. Arabnia, the Editor-in-Chief of the Springer Journal of Supercomputing, for offering us the opportunity to guest-edit this Special Issue.