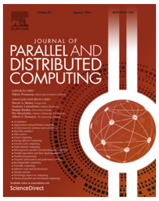
J. Parallel Distrib. Comput. 100 (2017) 30–41

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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



Shared resource aware scheduling on power-constrained tiled many-core processors[☆]

Sudhanshu Shekhar Jha[a,*], Wim Heirman[b], Ayose Falcón[c], Jordi Tubella[a], Antonio González[a], Lieven Eeckhout[d]

[a] *DAC, Universitat Politécnica de Catalunya, Spain*
[b] *Intel Corporation, Belgium*
[c] *HP Inc., Spain*
[d] *ELIS, Ghent University, Belgium*

HIGHLIGHTS

- A low-overhead and high scalable hierarchical power manager on a tiled many-core architecture with shared LLC and VR.
- Shared DVFS and cache adaptation can degrade performance of co-scheduled threads on a tile.
- DVFS and cache-aware thread migration (DCTM) to ensure optimum per-tile co-scheduling of compatible threads at runtime.
- DCTM assisted hierarchical power manager improves performance by up to 20% compared to conventional centralized power manager with per-core VR.

ARTICLE INFO

Article history:
Received 14 April 2016
Received in revised form
1 September 2016
Accepted 2 October 2016
Available online 19 October 2016

Keywords:
Many-core tiled architecture
Thread migration
Power budget
Adaptive microarchitecture

ABSTRACT

Power management through dynamic core, cache and frequency adaptation is becoming a necessity in today's power-constrained many-core environments. Unfortunately, as core count grows, the complexity of both the adaptation hardware and the power management algorithms increases exponentially. This calls for hierarchical solutions, such as on-chip voltage regulators per-tile rather than per-core, along with multi-level power management. As power-driven adaptation of shared resources affects multiple threads at once, the efficiency in a tile-organized many-core processor architecture hinges on the ability to co-schedule compatible threads to tiles in tandem with hardware adaptations per tile and per core.

In this paper, we propose a two-tier hierarchical power management methodology to exploit per-tile voltage regulators and clustered last-level caches. In addition, we include a novel thread migration layer that (i) analyzes threads running on the tiled many-core processor for shared resource sensitivity in tandem with core, cache and frequency adaptation, and (ii) co-schedules threads per tile with compatible behavior. On a 256-core setup with 4 cores per tile, we show that adding sensitivity-based thread migration to a two-tier power manager improves system performance by 10% on average (and up to 20%) while using 4× less on-chip voltage regulators. It also achieves a performance advantage of 4.2% on average (and up to 12%) over existing solutions that do not take DVFS sensitivity into account.

1. Introduction

Industry-wide adoption of chip multiprocessors (CMPs) is driven by the need to maintain the performance trend in a power-efficient way on par with Moore's law [40]. With continued emphasis on technology scaling for increased circuit densities, controlling chip power consumption has become a first-order design constraint. Due to the end of Dennard scaling [12] (slowed supply voltage scaling), we may become so power-constrained that we are no longer able to power on all transistors at the same time—*dark silicon* [16]. Runtime factors such as thermal emergencies [7] and power capping [19] further constrain the available chip power. Owing to all the above factors, power budgeting on many-core systems has received considerable attention recently [22,36,37,39,49,51].

[☆] This work is a collaborative effort.

[*] Corresponding author.

E-mail addresses: sjha@ac.upc.edu (S.S. Jha), wim.heirman@intel.com (W. Heirman), ayose.falcon@hp.com (A. Falcón), jordit@ac.upc.edu (J. Tubella), antonio@ac.upc.edu (A. González), lieven.eeckhout@ugent.be (L. Eeckhout).

http://dx.doi.org/10.1016/j.jpdc.2016.10.001
0743-7315/© 2016 Elsevier Inc. All rights reserved.

Dynamic voltage and frequency scaling (DVFS) for multiple clock domain micro-architectures has been studied extensively in prior work [11,24,25,49,52]. Current commercial implementations of fully integrated voltage regulators (FIVR) [8,32] support multiple on-chip frequency/voltage domains with fast adaptation, although per-core voltage regulators incur significant area overhead—previous works [8,31,48] suggest that the area of on-die per core voltage regulators is approximately 12.5% of core area. Other techniques such as core micro-architecture adaptation [3,13,20,43,30], cache adaptation [1,38,53,46] and network-on-chip adaptation [46] have been shown to be quite effective at managing power in isolation at high to moderate power budgets. Under more stringent power conditions, core gating [36,33] along with the above techniques can be used at the potential risk of starving threads.

Most existing power management schemes use a centralized approach to regulate power dissipation based on power monitoring and performance characteristics. Unfortunately, the complexity and overhead of centralized power management increases exponentially with core count [14]. Moreover, the area overhead of on-chip voltage regulators is significant which limits the number of voltage/frequency domains one can have on the chip. Hence, it becomes a necessity to employ a hierarchical approach as we scale fine-grain power management to large many-core processors at increasingly stringent power budgets. We therefore propose a *two-tier hierarchical power manager* for tile-based many-core architectures; each tile consists of a small number of cores and a shared L2 cache within a single voltage–frequency domain. The two-tier power manager first distributes power across tiles, and then across cores within a tile. The architecture also provides support for core, cache and frequency adaptations to avoid core gating at moderate to stringent power budgets.

Tiled many-core processors pose an interesting challenge when it comes to hardware adaptation and scheduling. Changing frequency and re-configuring the shared L2 cache affects all threads running in the tile. It therefore becomes important to *migrate* threads, such that threads with *compatible* behavior are co-scheduled onto the same tile. Since the execution behavior varies over time, periodic re-evaluation and dynamic thread migration is also required. We therefore classify threads based on their sensitivity to both cache and frequency dynamically at runtime. We propose DVFS and Cache-aware Thread Migration (*DCTM*): a scheduler running on top of the two-tier hierarchical power manager to ensure an optimal co-schedule for all threads running on the power-constrained tiled many-core processor while accounting for the effects of hardware adaptation.

In this work, we make the following contributions:

- We propose an *integrated two-tier hierarchical power management* for tiled many-core architectures, in which we first manage power across tiles and then within a tile.
- For a collection of multi-program and multi-threaded workloads, we report that our two-tier hierarchical power manager outperforms a centralized power manager by 3% on average, and up to 20% for a 256-core setup.
- We make the observation that thread scheduling is essential in a tiled many-core architecture to account for thread sensitivity towards shared resources. We classify threads based on their sensitivity to both cache and frequency adaptation, and we propose *DVFS and Cache-Aware Thread Migration (DCTM)* to optimize per-tile co-scheduling of compatible threads.
- We provide a comprehensive evaluation of *DCTM* on a tiled many-core processor. We use multi-program workloads consisting of both single-threaded and multi-threaded applications, and we report that *DCTM* improves system performance by 10% on average, and up to 20%. *DCTM* outperforms existing solutions by 4.2% on average (and up to 12%).

2. Motivation

2.1. Limitations of a centralized approach

In the context of power management in many-core processors, prior works [38,11,36] have relied on a central entity (micro-controller) to manage power using one or more micro-architectural techniques to trade off performance at high to moderate power budgets. At stringent power budgets, neither of power management schemes like DVFS nor core adaptation nor cache resizing *in isolation* can provide a viable solution. As a result, prior work [33,36] had to resort to core gating at stringent power envelops. Previously proposed state-of-the-art frameworks [38,30,43] provide an integrated framework for multi/many-core architectures by combining and coordinating core adaptation, cache resizing and/or per-core DVFS to maximize system performance across a wide range of power budgets. These frameworks provide some form of global power management that operates on the runtime statistics of each core to decide on an optimal per-core working configuration. During each time slice, a per-core *Performance Monitoring Unit* (PMU) tracks activity statistics using hardware counters, and predicts/projects the performance and power of all possible configurations. Each core's PMU sends a list of optimal configurations to the *Global Power Manager* (GPM), which globally optimizes the many-core configuration within the given power budget. The GPM instructs each core to reconfigure itself based on the global optimization.

In commercial designs, both the per-core PMU and global GPM are already present in some form [45]. The PMU typically collects power consumption and junction temperatures, and performs control functions such as P-state (DVFS) and C-state (various levels of power gating) transitions. The GPM is implemented as an integrated micro-controller and runs firmware algorithms that interface with the PMUs and on-chip voltage regulators. The PMU keeps track of a core's activity and controls the micro-architectural configuration in response to requests made by the GPM; the GPM combines information from all cores and performs the global power/performance optimization, see *Centralized Approach* in Fig. 1. But as core count continues to grow, the centralized approach becomes inviable: Deng et al. [11] report *quadratic* computational complexity, while Li and Martinez [36] suggest the computational complexity to be *logarithmic* to core count. In future many-core processors [6], a centralized GPM – even with logarithmic complexity – would be a severe bottleneck.

Because a centralized power manager does not scale favorably towards large many-core processors and fine-grain hardware adaptations, we propose *two-tier hierarchical* power management (see Section 3)—*first contribution in this work.*

2.2. Cache-aware thread migration (Cruise)

When threads are co-scheduled on a multi-core processor with a shared last-level cache (LLC), conflicting thread behavior can lead to suboptimal performance. For instance, when a thread whose working set fits in the shared cache is co-scheduled with a streaming application, the quick succession of cache misses from the streaming application may push the working set of the first application out of the shared cache, thereby significantly degrading its performance. Jaleel et al. [27] propose Cruise: a hardware/software co-designed scheduling methodology that uses knowledge of the underlying LLC replacement policy and application cache utility information to determine how best to co-schedule applications in multi-core systems with a shared LLC.

Cruise monitors the number of LLC accesses per kilo instructions (APKI) and miss rate (MR) for each application. Application classification based on these metrics along with co-scheduling rules then optimize overall system performance. The applications are classified in the following categories: