Power-Aware Resource Scheduling in Base Stations

Magnus Själander, Sally A. McKee, Bhavishya Goel
Chalmers University of Technology
Gothenburg, Sweden
{hms | mckee | goelb}@chalmers.se

Peter Brauer, David Engdal, and Andràs Vajda
Ericsson AB
Gothenburg, Sweden
{peter.brauer | david.engdal | andras.vajda}@ericsson.se

***Abstract*—Baseband stations for Long Term Evolution (LTE) communication processing tend to rely on over-provisioned resources to ensure that peak demands can be met. These systems must meet user Quality of Service expectations, but during non-peak workloads, for instance, many of the cores could be placed in low-power modes. One key property of such application-specific systems is that they execute frequent, short-lived tasks. Sophisticated resource management and task scheduling approaches suffer intolerable overhead costs in terms of time and expense, and thus lighter-weight and more efficient strategies are essential to both saving power and meeting performance expectations. To this end, we develop a flexible, non-propietary LTE workload model to drive our resource management studies. Here we describe our experimental infrastructure and present early results that underscore the promise of our approach along with its implications on future hardware/software codesign.**

I. Introduction

Power dissipation plays an increasingly important role in system design and specification of operational constraints. Power also influences the manufacturing cost of a system in terms of required cooling equipment. Furthermore, it has a direct impact on the operational cost. Balancing power efficiency with application performance requires intelligent resource management.

In this project, we study the power-aware resource management problem with respect to base stations for Long Term Evolution (LTE) radio based communication and lean DSP multicore clusters. Here, a minimal OS supports dedicated signal processing software with significant Quality of Service (QoS) requirements. These systems afford neither excess (non-dedicated) hardware resources nor spare execution cycles to implement sophisticated monitoring or complex management schemes. On the other hand, such systems tend to be necessarily over-provisioned with respect to LTE processing in order to gracefully perform under peak loads. Such systems could enable flexible task mapping, and when processing non-peak workloads, cores may be put to sleep to save power with no impact on QoS. These two synergistic approaches can be applied together to maximize efficiency and power savings.

Here we describe our experimental software and hardware infrastructure, which we are leveraging to explore more effective and efficient algorithms and policies to realize the potential for core deactivation combined with smarter task management. We report initial results that reinforce the promise of our approach and highlight opportunities for longer-term hardware/software codesign.

II. Experimental Infrastructure

We have developed the infrastructure to study different scheduling approaches, and are continuing to study interactions and tradeoffs with respect to both power and performance. We first describe the design and implementation of our model of the LTE communication "software pipeline", then discuss two different power-measurement approaches to inform system adaptation.

A. LTE workload

The key to this infrastructure is a representative workload of an LTE base station uplink. Developed jointly by Chalmers and Ericsson, this LTE workload is non-proprietary, and thus (pending some in-house optimizations) will be distributed as open source. The workload will provide researchers with realistic software for an LTE base station and means to conduct their own research in this field. The workload's portability and scalability facilitate execution on multiple hardware platforms with different levels of parallelism. This also makes the LTE workload suitable for benchmarking different hardware platforms to assess their appropriateness for modeling LTE baseband systems. Furthermore, the various signal processing kernels can easily be replaced by other, suitable algorithms by simply dropping them into place. We find this flexibility essential for the studies we conduct.

Portability is supported with POSIX threads (Pthreads) and code that avoids architecturally specific solutions. Platform-specific functionality (such as deactivation of cores) is structured such that supporting code is easily identifiable and separable from main workload functionality. Workload behavior remains independent of these architectural-specific constructs. Scalability is supported through a set of parameters that specify the behavior, number of threads, and scheduling of the LTE workload.

We have studied this LTE workload on three different hardware platforms: an Intel Core Duo processor with two cores, a dual Intel Xeon E5620 processor with a total of 8 cores, and a Tilera TILEPro64 processor with 64 cores [1]. The Intel processor allows for easy debugging and ready access to hardware on which the LTE workload can be executed. Furthermore, the platform supports Performance Monitoring Counters (PMCs) used in our work on per-core power prediction, and this gives us more tools for understanding how power measurements correspond to workload behavior (see results below). The TILEPro64 processor is a suitable research

vehicle for this problem, given that it is both highly parallel (64 DSP like cores) and optimized for low power. The TILEPro64 processor also provides means to deactivate cores at runtime and to measure unit activity through the use of PMCs. Linux runs on both platforms; having access to the kernel source has been essential, and the OS support for Pthreads facilitates easy porting of our workload to other platforms. Linux is obviously too heavyweight to represent LTE base station management software, but in this context it is sufficient to support a flexible model of base station operation to support our power-aware resource management studies.

B. Power Measurements

To study power dissipation, we developed two custom hardware infrastructures. The Tilera TILEPro64 development board employs two resistors to load balance across the two phases of the buck converter supplying power to the TILEPro64 chip. We measure the voltage drop across these two resistors to extract current consumption. The measurements are performed with a National Instruments (NI) USB-6210 [2] data acquisitioning unit that can sample the two voltages at a period of eight microseconds. For the Intel platform, we developed a power measurement board that measures current consumption of the individual supply voltage levels on the motherboard's ATX connectors. The current is measured with a current transducer from LEM [3] that uses the Hall effect to measure the current flow and generates respective output voltage. The board is connected between the conventional power supply unit and the power connector on the motherboard, as shown in figure 1. As most current motherboards have a separate supply connector for processor, the processor power dissipation can be isolated from other motherboard components. The output of each transducer is connected to the same NI USB-6210 as the TILEPro64. As the sensitivity of the transducer is 25mV/A and the sensitivity of the USB-6210 is 47.2μV it is possible to measure the current at a sensitivity of 2mA.

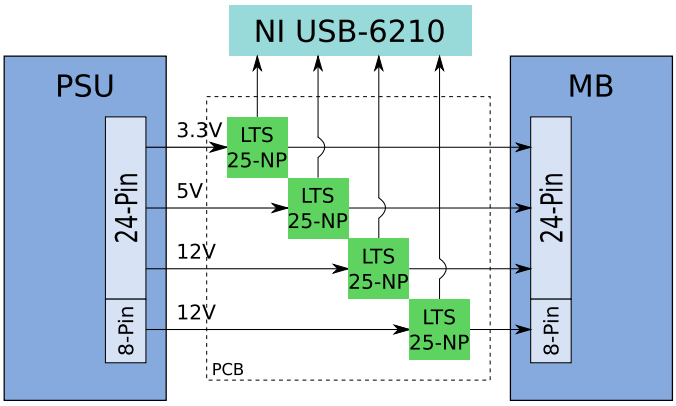


Fig. 1. Illustration of ATX power measurements infrastructure. 8 and 24-Pin are ATX molex connecters used for conventional PSUs and motherboards; LTS 25-NP is a current transducer from LEM; and NI USB-6210 is a data acquisition unit from National Instruments.

C. Per-Core Power Estimation Models

This power measurement infrastructure is sufficient to measure total power dissipation at the chip level. However, today's multi-core era requires means to measure power at the core granularity to enable the development of power aware scheduling techniques (along with the means to analyze their behaviors). Since power dissipation is directly related to circuit activity, we can leverage event-driven performance monitoring counters (PMCs) and temperature sensors, whose implementation per core is becoming increasingly common in contemporary microarchitectures.

We have developed power models that leverage PMCs and temperature sensors to estimate core power dissipation by establishing a statistical relationship between PMC values, temperature, and measured power dissipation [4]. We validated the models for six AMD/Intel platforms with two to eight cores and different memory hierarchies. We ran three different application suites comprising multiprogrammed and multithreaded workloads. We also made an implementation for the high-performance Linux kernel developed by Betti *et al.* to verify that the cost to compute the model every 10 ms is negligible [4].

While such an approach might be appropriate for general purpose operating systems and longer running tasks, it is likely to be prohibitively expensive to implement continuous power estimation in parallel with LTE user workloads. Instead, a monitoring thread could be run occasionally to sample system behavior. These data could then be processed during non-peak times and used to convey more global power dissipation information and longer term patterns to an adaptive system. This enables phase-based behavior recognition and adaptation, which has proved successful in other (non-power critical) scenarios [6], [7].

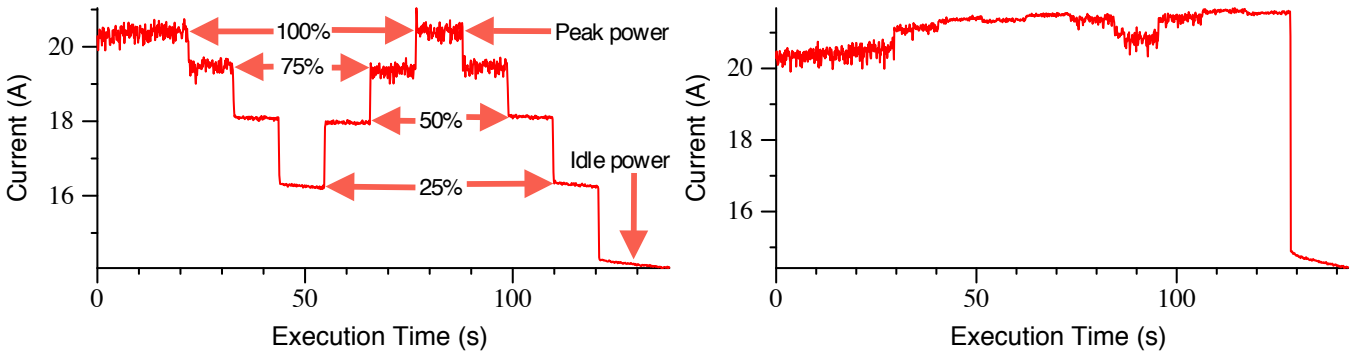
III. Experiments

The LTE workload and platform support allow us to study trends with respect to where and when we turn off cores, along with thread mapping and scheduling policies.

We largely use the Intel platform to develop and debug our LTE workload. In contrast, most of our experiments are run on the TILEPro64, where we use the workload to exercise the many processors in order to better understand the behavior of an LTE uplink. Our first goal was to parallelize the LTE workload such that it can run efficiently on all cores to model maximum load scenarios. Our second goal is to investigate possibilities for deactivating cores to save power under low load scenarios. This work is ongoing.

We have studied different parallelization schemes and scheduling policies for the maximum load scenarios. The current implementation is parallelized across subframes, users, channel estimation, symbols, layers, and receive antennas. Scheduling is handled through a number of queues, and a size-configurable set of cores can be allocated to specific queues to improve data locality.

For low load scenarios, we need to be able to deactivate ("nap") cores. For this we modify the Tilera Linux interrupt



(a) Power dissipation when deactivating cores

(b) Power dissipation without deactivating cores

Fig. 2. Example of power dissipation differences with and without deactivating cores, in accordance with hardware utilization. The y-axis shows measured current in ampers, which has a linear relation to power (since the supply voltage is constant).

handler so that cores deactivated by the user-mode "nap" instruction can be re-activated dynamically. By deactivating cores during low load scenarios, we observe reductions in power.

Figure 2 shows the power dissipation when executing the LTE workload on a TILEPro64 in a scenario where the computational load is changed, in steps of 25%, between 100% and 25%. The total execution time is almost two minutes and each load level lasts for ten seconds. Figure 2(a) and figure 2(b) shows the same workload scenario but in figure 2(a) cores are deactivated using the nap instruction while in figure 2(b) the cores are left spinning in an idle loop. The graphs show a clear potential to significantly reduce the power by core deactivation when the load is low. When the load is reduced from 100% to 25%, the total power reduction is 20%, which represents a 60% reduction of the power difference between peak and idle power.

IV. Ongoing Research

The processing of a subframe for an LTE base station uplink is driven by a number of input parameters that characterize the computational effort required. These parameters are the number of users and the number of resource blocks, layers, and the modulation technique used for each of these users. We will study the correlation of all these parameters with respect to the activity of the cores on the TILEPro64 architecture. Armed with these correlations, we will develop better algorithms to estimate the core utilization. With the capability to accurately estimate core activity, we can then devise models for when to deactivate and activate cores to reduce power.

These models are the first step in creating a complete system that is power and temperature aware. With knowledge of the current temperature of a base station and the capability to accurately estimate the power for a particular resource schedule (given by the parameters of such a schedule), we can set temperature and power limits and gracefully reduce user data transmission capacity if and when it is required, e.g., when a base station is about to overheat.

V. Related Work

Isci et al. [8] analyze power management policies to enforce a given power budget and to minimize the power dissipation for the given performance target. They conduct their experiments on the Turandot [9] simulator. They get their power estimates from IBM PowerTimer instead of developing their own power model. They have developed a global power manager that leverages power-performance data available from locally available monitors per core to enforce the dynamic voltage and frequency scaling (DVFS) policies individually for each core.

Banikazemi et al. [10] present a power-aware meta-scheduler (PAM). PAM monitors the performance, power and energy of the system by using performance counters and built-in power monitoring hardware. It then uses this information to dynamically remap the software threads on multi-core servers for higher performance and lower energy usage. PAM runs in user space and hence does not require kernel changes.

Thread-motion [11] proposes the use of a few but fixed frequency and voltage domains. The application behavior is then monitored and tasks are moved between these domain appropriately to reduce power. For short-lived tasks this is not a viable solution as moving a task would incur high overheads and it would be more efficient to let the task run until completion. Teodorescu and Torrellas propose a power management technique based on linear programming called LinOpt [12]. Coskun et al. [13] propose job scheduling and power management based on a performance database that is periodically updated and queried during scheduling. Both techniques assume a scheduling interval in the range of tens of milliseconds. This is magnitudes slower than what is required for scheduling short-lived tasks that lasts for microseconds.

In contrast to previous work we target many-core architectures where power management is performed by deactivating a core instead of using DVFS or moving tasks between different domains. Our investigation is performed on real hardware with a realistic workload of an LTE base station uplink, where

power can be measured with high accuracy. As we target one application, we leverage application specific knowledge to estimate core activity and use this for developing core deactivation policies.

VI. Conclusions

Our investigations thus far have underscored the need for efficient scheduling and mapping of short-lived tasks. The short task lifetimes require extremely low overhead scheduling and mapping techniques, which argues for custom support in hardware. The LTE workload created in this project will play an important role in this continued research: studying the design software-only (near-term) solutions together with the tradeoffs for systems leveraging application specific hardware design will provide rich research collaboration opportunities far into the future.

References

[1] *Tilera TILEPro64 Processor Product Brief*, Tilera Corporation, http://www.tilera.com/sites/default/files/productbriefs/PB019_TILEPro64_Processor_A_v3.pdf.

[2] *NI USB-6210, 16-Bit, 250 kS/s M Series Multifunction DAQ, Bus-Powered*, National Instruments Corporation, apr 2009, http://sine.ni.com/nips/cds/view/p/lang/en/nid/203223#.

[3] *Current Transducer LTS 25-NP*, LEM Corporation, nov 2009, http://www.lem.com/docs/products/lts%2025-np.pdf.

[4] B. Goel, S. A. McKee, R. Gioiosa, K. Singh, M. Bhadauria, and M. Cesati, "Portable, scalable, per-core power estimation for intelligent resource management," in *Green Computing Conference, 2010 International*, aug 2010, pp. 135–146.

[5] E. Betti, M. Cesati, R. Gioiosa, and F. Piermaria, "A global operating system for HPC clusters," in *Proceedings of the IEEE International Conference on Cluster Computing*, Aug. 2009, pp. 1–10.

[6] T. Sherwood, S. Sair, and B. Calder, "Phase tracking and prediction," in *Proc. 30th International Symposium on Computer Architecture*, jun 2003, pp. 336–447.

[7] Y. Xie and G. H. Loh, "Dynamic classification of program memory behaviors in CMPs," in *Proc. 2nd Workshop on Chip Multiprocessor Memory Systems and Interconnects*, jun 2008.

[8] C. Isci, A. Buyuktosunoglu, C. Cher, P. Bose, and M. Martonosi, "An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget," in *Proc. IEEE/ACM 40th Annual International Symposium on Microarchitecture*, Dec. 2006, pp. 347–358.

[9] M. Moudgill, P. Bose, and J. Moreno, "Validation of Turandot, a fast processor model for microarchitecture exploration," in *Proc. International Performance, Computing, and Communications Conference*, Feb. 1999, pp. 452–457.

[10] M. Banikazemi, D. Poff, and B. Abali, "PAM: A novel performance/power aware meta-scheduler for multi-core systems," in *Proc. IEEE/ACM Supercomputing International Conference on High Performance Computing, Networking, Storage and Analysis*, no. 39, Nov. 2008.

[11] K. K. Rangan, G. Wei, and D. Brooks, "Thread motion: Fine-grained power management for multi-core systems," in *Proceedings of the 36th Annual International Symposium on Computer Architecture*, June 2009, pp. 302–313.

[12] R. Teodorescu and J. Torrellas, "Variation-aware application scheduling and power management for chip multiprocessors," in *Proceedings of the 35th Annual International Symposium on Computer Architecture*, June 2008, pp. 363–374.

[13] A. K. Coskun, R. Strong, D. M. Tullsen, and T. S. Rosing, "Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors," in *Proceedings of the 11th International Joint Conference on Measurement and Modeling of Computer Systems*, June 2009, pp. 169–180.