usenix ;login:
FEBRUARY 2013 VOL. 38, NO. 1
ELECTRONIC SUPPLEMENT

Conference Reports

2012 Workshop on Power-Aware Computing and Systems (HotPower '12)

Hollywood, CA
October 7, 2012

Keynote Address

Summarized by Vinicius Petrucci (vpetrucci@gmail.com)

Smart Grid—Opportunities and Challenges in the Creation of the 21st-Century Power Grid

Rajit Gadh, Professor, Henry Samueli School of Engineering and Applied Science, Director, UCLA Smart Grid Energy Research Center (SMERC)

Dr. Rajit Gadh presented some major challenges and technologies required to make the US electric grid smart and more efficient, given that energy consumption is rapidly increasing and becoming more expensive. One critical issue is the inflexibility of the current grid system, whose structure assumes a unidirectional flow of power, control, and information. For instance, power generation has a flat profile, but the consumption has typically time-varying behavior. UCLA's Smart Grid Energy Research Center (http://smartgrid.ucla.edu/) proposes a grid structure that is much more flexible, integrating aspects of power generation and consumption.

The opportunistic smart grid presents a new flow model that, for example, would allow for generating energy back to the power distribution subsystem. To provide such smart and flexible power grid systems, technologies such as wireless/mobile devices and sensor/control systems need to be integrated carefully into the existing grid system. The key change in the current grid system deals with providing two-way power communication and control/management. This idea of bidirectional power flow would allow services to aggregate power from consumers and offer it using a distribution network in the smart grid.

A prototype platform for power monitoring and management, WINSmartGrid (Wireless Internet Smart Grid), has been developed and deployed in smart grid research. Dr. Gadh mentioned that they are exploring Arduino and Raspberry Pi controllers. Another interesting example is the WINSmartEV (Wireless Internet Smart Electric Vehicle) framework to enable smart charging of electric vehicles (EVs). Because the price of energy can vary over time, this framework would enable making better decisions on when to charge your car: for example, during non-peak times (at night) when energy is cheaper. Also, with the permission and control of EV owners, local utilities could extract power from their EVs during peak-demand periods and use this aggregated power in utility-grade offerings.

An audience member asked about what else could be done to improve energy efficiency and what are the limits to an energy-efficiency strategy. Gadh cited as an example a smart building that could turn lights on/off automatically when people come in and out; however, such a demand-response strategy improves energy efficiency only up to a point. Additionally, other energy-efficient buildings, for example, could use better lights that use much less power. Another audience member posed an interesting question regarding energy efficient storage and better battery technologies. The discussion focused on the fact that there is a relatively new demand for very large scale battery energy and many issues are still being investigated; for example, what is the exact value and impact for one battery discharge (in the case of the WINSmartEV), among other battery-life issues that need to be addressed to be able to support large scale smart grid systems.

Energy in Smartphones

Summarized by Aaron Carroll (aaron.carroll@nicta.com.au)

Personalized Diapause: Reducing Radio Energy Consumption of Smartphones by Network Context-Aware Dormancy Predictions

Yeseong Kim and Jihong Kim, Seoul National University

About 30% of smartphone energy is consumed in the 3G network, but 33% of that is wasted according to a study of 25 smartphone users presented by Yeseong Kim. Personalized Diapause is a system to optimally move the radio energy into the low power dormancy state automatically and in accord with the behavior of applications and individual users, both of which the authors show to be important.

From the application call-stack, the solution extracts network activities with similar characteristics, and generates for each a model of the distribution of packets in time. From this, a prediction is made as to when the 3G radio connection should enter low-power dormancy mode.

A correct prediction reduces wasted energy; an incorrect prediction incurs the additional cost of reconnection. Limiting the extra reconnections to 10%, Personalized Diapause is able to reduce radio energy by 23%, and by up to 35% if reconnections are unbounded.

Mian Dong (Samsung) asked whether this technique could be applied at the cellular radio base station. Kim responded that the tool can bound the reconnections, and that control of this could be exposed to network operators.

Supporting Distributed Execution of Smartphone Workloads on Loosely Coupled Heterogeneous Processors

Felix Xiaozhu Lin, Zhen Wang, and Lin Zhong, Rice University

Lin presented a position paper on achieving energy-proportionality in smartphones. The authors posit that loosely coupled heterogeneous multicore is the most energy-efficient design, and so have proposed Kage, a system that provides a familiar programming model to smartphone applications running on asymmetric and non-coherent hardware.

Typically, smartphone workloads involve high temporal variation and functional diversity both within and across threads, leading to the conclusion that replication, rather than specialization, is the right OS design. The Kage system achieves this with an application runtime and hardware abstraction layer to achieve replicated and consistent execution of legacy user and OS code.

Jitu Padhye (MSR) asked how app developers are involved. Lin explained that because there is no transparent process migration, the application code needs to be annotated with yield points. Padhye followed up asking about handling asymmetry of CPU features. Lin pointed out that small cores are often feature-symmetric with simpler microarchitectures or lower clock rates, but that where feature differences exist, non-transparency might have to be accepted. Maria Kazandjieva (Stanford) asked what the expected savings are, given that the CPU is not necessarily the main energy consumer. Lin answered that CPU energy is not trivial, but that the approach might be applied to other subsystems. Another questioner asked whether the migration overhead could offset gains. Lin claimed that migration time is low compared to user perception, and that the migration frequency should also be limited. To a follow-up question on ISA asymmetry, Lin said the system restricts the target to systems where the small cores implement a subset of the big-core ISA.

Towards Verifying Android Apps for the Absence of No-Sleep Energy Bugs

Panagiotis Vekris, Ranjit Jhala, Sorin Lerner, and Yuvraj Agarwal, University of California, San Diego

The Android OS conserves energy through suspending to a low-power state opportunistically. An application can override this behavior with the wake lock, which follows the typical acquire-release model. Incorrect use of this API can lead to an "energy bug" whereby the device does not suspend when it ought to. Vekris presented a static-analysis technique for the discovery of energy bugs in Android applications.

The analysis is driven by a policy that defines conditions that are considered energy bugs, such as points in application code where a wake lock should not be held. The static analysis tool produces a control flow graph of the application annotated with the wake lock state, and this is compared to the policy to identify whether any program flow can lead to an energy bug.

In an analysis of hundreds of Android apps, 31 of the policy-violating ones were selected at random and analyzed manually. Of these, 14 contained genuine energy bugs, while 17 were incorrectly marked as buggy, mainly due to complex conditionals of the lock release paths.

Aaron Carroll (NICTA/UNSW) asked whether the analysis exposes an inadequacy of the wake lock API. Vekris responded that Java-like synchronized blocks could alleviate some of the problems, but that higher-level APIs like timed auto-release do exist. Thomas Wenisch (U Michigan) asked at what point the analysis breaks down. Vekris replied that complex conditionals in application code can be problematic. Weisong Shi (Wayne State) wondered whether the approach could be applied to OS services, but Vekris pointed out that the analysis cannot be applied to native code.

Scheduling, Synchronization, and Storage

Summarized by Devesh Tiwari (devesh.dtiwari@ncsu.edu)

Reducing Data Movement Costs Using Energy-Efficient, Active Computation on SSD

Devesh Tiwari, North Carolina State University; Sudharshan S. Vazhkudai and Youngjae Kim, Oak Ridge National Laboratory; Xiaosong Ma, North Carolina State University and Oak Ridge National Laboratory; Simona Boboila and Peter J. Desnoyers, Northeastern University

Devesh first introduced how scientific simulations are performed on large supercomputers, such as Jaguar, Oak Ridge National Lab's leadership computing machine. Scientific discovery is often a two-step process: simulation and data analysis; however, due to huge amounts of data being generated by these machines every hour, traditional data analysis is performed on a set of smaller clusters offline. This offline approach suffers from significant data movement costs such as redundant I/O, storage bandwidth bottlenecks, and wasted CPU cycles. This increases total energy consumption and end-to-end latency.

To address these challenges, the authors propose executing data analysis tasks on the controllers of emerging storage devices, such as SSDs. They call this approach the Active Flash approach, where data analysis is performed in situ on the SSD controller without degrading the performance of the simulation. By migrating analysis tasks closer to where the data resides, this approach helps reduce the data movement cost.

Devesh demonstrated how to model the number of SSDs required in order to hold the data analysis output subject to different application- and device-based constraints. This modeling is done without accounting for Active Flash or active computation on SSD controllers. Interestingly, the

model shows that no additional SSDs are needed to support active computation on these controllers in most cases. They also show that active computation can be performed without any simulation slowdown.

Quantitative Estimation of the Performance Delay with Propagation Effects in Disk Power Savings

Feng Yan and Xenia Mountrouidou, College of William and Mary; Alma Riska, EMC Corporation; Evgenia Smirni, College of William and Mary

Feng Yan presented a new method to save power in disks in datacenter settings, primarily motivated by disk power consumption in large-scale computing facilities. Disks may consume as much as 37% of a facility's power, but they are hard to optimize for two reasons.

First, knowing the arrival distribution of disk requests ahead of time is hard. Second, the performance penalty for resuming from power-saving mode is significant. The authors mitigate these challenges while maximizing power savings and meeting performance guarantees. Feng showed that prior techniques—e.g., aggressive scheduling, util-guided scheduling, and fixed-wait scheduling—are not always efficient in exploiting power-saving opportunities and minimizing performance degradation. Using an analytical model, the authors show why their delay propagation-based approach is more efficient. Finally, they evaluated their scheme on a large enterprise-trace, and results indicate that their technique saves power by up to a factor of 10.

During Q&A, Feng clarified that the overhead of running this decision-making process is minimal. Because the evaluation is done using traces, how much this affects the power consumption of processing cores is not clear and is an issue for future investigation.

The Implications of Shared Data Synchronization Techniques on Multi-Core Energy Efficiency

Ashok Gautham, IIT Madras; Kunal Korgaonkar, IIT Madras and IBM Research; Patanjali SLPSK, Shankar Balachandran, and Kamakoti Veezhinathan, IIT Madras

Ashok Gautham presented an evaluation of different data synchronization techniques on multicore architectures. The evaluation considers three types of techniques: mutex, spin lock, and software transactional memory (STM). The authors considered three metrics: performance, performance per watt, and energy delay product.

The evaluation was done using the STAMP critical section benchmarks on an Intel core-i7 machine running Linux and a competitive STM library from EPFL. Overall, the results showed that STM outperforms the lock-based approach by an order of magnitude; however, sequential code seems to perform better than STM with respect to energy delay product. Lock-based approaches provide the best performance in most cases. In the future, the authors plan to investigate a wider set of benchmarks.

A few attendees (from University of Cambridge) recommended using better lock-based and STM benchmarks and lock-free data structures. Ashok presented an example why transforming some data structures to lock-free versions might be challenging. The discussion concluded with a consensus that evaluating energy-efficiency of lock-free data structures would be interesting.

Lucky Scheduling for Energy-Efficient Heterogeneous Multi-Core Systems

Vinicius Petrucci and Orlando Loques, Universidade Federal Fluminense; Daniel Mossé, University of Pittsburg

Vinicius Petrucci presented a scheduling algorithm and its implementation on heterogeneous multicore architectures. As multicore architectures are evolving, heterogeneity seems to be a promising approach where some cores would be faster than others; however, such heterogeneity in hardware complicates the scheduling of applications on these platforms. Finding an optimal schedule is challenging, and so a rule-of-thumb has been applied in previous techniques: memory-bound applications are executed on a slower (smaller) core, while CPU-intensive cores run on faster cores. This paper presents evidence to show such "bias"-based scheduling results in a monopoly of large cores (i.e., performance degradation for other applications).

To address this issue, Vinicius presented a lottery-based scheduling mechanism that enables fair sharing of cores among all the competing threads. Unlike previous works, it takes both performance and power into account while scheduling applications. The presented results look promising, and the authors are extending this work for more rigorous evaluation and real-time guarantees. Two interesting insights from this study are: the thread that may have similar memory-intensity may prefer different cores for better energy efficiency; some cores may be memory bound, but it is better to run them on faster cores for better energy efficiency.

In the Q&A, Vinicius clarified that the mechanism performs online data collection, and hence can adapt to different phases in a program. He added that the scheduling technique measures relative memory-intensity and hence is not vulnerable to interference-induced errors. Finally, he hinted that he planned to investigate a scheduling mechanism for the whole system instead of individual applications.

Power/Performance Measurement Studies

Summarized by Aaron Carroll (aaronc@cse.unsw.edu.au)

Accurate Characterization of the Variability in Power Consumption in Modern Mobile Processors

Bharathan Balaji, John McCullough, Rajesh K. Gupta, and Yuvraj Agarwal, University of California, San Diego

Variability in the CPU manufacturing processes is seen between production batches, across dies on a wafer, and even within a single die—and is projected to increase. Balaji presented an analysis of the power variation between six instances of "identical" Intel i5 CPUs running on a power-instrumented motherboard.

A power variation of 12–17% between processors is seen across the SPEC CPU2006 benchmark suite, with variability increasing with core frequency. The effect of multithreading, C-states, and turbo boost are also explored, with non-intuitive results. The authors conclude that the main contributor to variation is leakage power.

Thomas Wenisch (U Michigan) asked whether variation due to temperature had been considered. Balaji answered that none had been observed. David Meisner (Facebook) asked why variability was benchmark-dependent. Balaji speculated that it might be related to the proportion of memory-access instructions. Mian Dong (Samsung) asked how knowledge of variability changes optimizations. Balaji admitted that although it is an open problem, one consequence is that devices will need to be characterized individually. Finally, Krishna Ramkumar (Intel) suggested energy MSRs be used in future work.

Memory Performance at Reduced CPU Clock Speeds: An Analysis of Current x86_64 Processors

Robert Schöne, Daniel Hackenberg, and Daniel Molka, Technische Universität Dresden

CPU frequency scaling algorithms are predicated on the assumption that core frequency does not affect memory performance. Schöne presented a survey of the truth of this claim on a range of recent x86_64 processors.

The data show that the relationship between core frequency and memory performance is highly CPU dependent. In some instances, scaling is linear, but in others, memory throughput is frequency independent. Similar complexity is seen in the L3 cache throughput. From this, the authors conclude that profiling the target system before applying DVFS is important.

Thomas Wenisch (U Michigan) asked whether memory latency was also affected, but the authors had not measured it. David Meisner, Facebook, wanted to know why a large variation from one to two cores was seen. Schöne answered that a single core is unable to saturate the memory bandwidth. Aaron Carroll (NICTA/UNSW) asked whether varying RAM frequency could explain the observations. Schöne replied that it had not been investigated.

Power and Performance Analysis of GPU-Accelerated Systems

Yuki Abe and Hiroshi Sasaki, Kyushu University; Martin Peres, Laboratoire Bordelais de Recherche en Informatique; Koji Inoue and Kazuaki Murakami, Kyushu University; Shinpei Kato, Nagoya University

GPUs have become popular for high-performance computing applications, but their power consumption typically exceeds that of the CPU. This naturally leads to the use of DVFS on the GPU. Yuki Abe presented an analysis of the efficacy of both CPU and GPU frequency scaling on GPU-accelerated systems.

A test system featuring an Intel i5 CPU and an nVIDIA GTX480 GPU showed that reducing CPU frequency on a mixed CPU- and GPU-intensive workload only increases energy consumption, which can be traced to the large idle power contribution of the GPU. For GPU-intensive workloads, Abe showed that reducing the GPU memory clock can reduce energy for core-bound tasks, which are characterized by small input data size; however, scaling the GPU core clock generally leads to an increase in energy consumption, independent of the input size.

After the talk, Thomas Wenisch (U Michigan) asked for clarification regarding the idle power consumption of the GPU. The speaker pointed out that it was indeed 46 W, about 1.2x that of the GPU-less system.