GEMSCLAIM
Greener Mobile Systems By Cross Layer Integrated Energy Management

Thomas Fahringer and Peter Thoman, University of Innsbruck, Austria

Consortium:
University of Innsbruck
Queen’s University Belfast
RWTH Aachen University
Politecnica University of Timisoara

Start: September 1, 2012
End: August 31, 2015

Chist-era Projects Seminar 2016, Bern, April 28, 2016
Mobile systems are becoming more energy demanding

Longer battery life is the most desired user feature.

Trade-off between energy and performance for mobile devices and many other computing systems: desktop computing, cloud computing, HPC

Mobile devices are inherently heterogeneous

- Risc-like application processor, DSP for baseband processing
Key challenges and potential impact

Cross layer energy management and optimization for mobile devices: HW/simulator, OS, compiler

- Energy-aware optimizing and parallelizing compiler
- Energy-proportional operating system
- Customizable HW modeling with energy monitoring facilities
  - simulator
  - FPGA

Potential impact

- control trade-off between energy optimization and performance
- additional 30% energy savings for mobile terminals
Consortium and synergies

Cross-layer power optimization

- Compiler/RTL/OS interface, Multi-objective optimisation
- Energy accounting, abstractions, SLAs
- Continuous energy monitors & sensors
- Virtual prototyping & hardware design
- Instruction-level power modelling
- OS-FPGA, OS-VP interface, lightweight RTL
- Energy-efficient programming and energy optimisation of mobile workloads at/across levels
GEMSCLAIM Software and Hardware Stack

- Mobile workloads
- Desktop workloads
- HPC workloads

- OpenMP+
- GEMSCLAIM compiler
- GEMSCLAIM Runtime
- RAPMI
- GEMSCLAIM OS
- LEM drivers

- Hardware Energy Mgmt
- LEM
- Hardware platform

Energy-aware System Software
Parallel SystemC Simulation

- Contemporary heterogeneous multi-core system
  - System parameters (number/type of cores, mem size...) statically configurable
- GEMSCLAIM simulator is based on SystemC
  - Exchanging reference SystemC with ICE’s SCOPE SystemC kernel enables parallel simulation

Instruction set simulators are automatically mapped to simulation threads
- Each thread simulates on its own and maintains its own simulation time
- Maximum allowed time difference ($t_{la}$) is annotated by the system designer
- Time-Decoupling may induce a timing error during cross thread communication

CHIST-ERA Projects Seminar, Bern, Switzerland, April 28-29, 2016
Simulation speed vs. accuracy

- Test setup (parallel simulator uses 4 threads):
  - 8 RISC + 8 VLIW system
  - Application: “ocean” from SPLASH-2 benchmark suite

- 3x speedup possible with ~1% error in timing
- Timing error increases linearly with time decoupling
FPGA prototype with PTEA

- Per Thread Energy Accounting (PTEA)
  - Per-core energy accounting (HW)
    - Shared resources
      - Monitor bus transactions
      - Account the energy of the transaction to the core which initiated it
      - Use the interconnect existing support to identify the source of the transaction
    - Local resources
  - Per-thread energy accounting (OS)
    - Energy counters part of thread context
FPGA prototype with PTEA

Platform
- ZC 702 board with Xilinx Zynq 7020 device
  - PL – programmable logic
  - PS – processing system

System
- ARM Cortex A9 Dual Core processing system
  - ARM0 – used to configure and monitor LEM
- Target reference design block
  - 4 processing cores (Microblaze), with local interrupt controller and local memory
  - LEM sensors: 2 sensors/core, 2 shared sensors/memory
- Target design
  - 2 clusters of 4 cores blocks
  - 8 Microblaze cores, with local interrupt controller, local timer and local memory
  - 2 LEMs
FPGA prototype with PTEA

- Calibrate and validate FPGA prototype
- WCET benchmarks
  - No OS / bare metal implementation
  - Small enough to be executed from BRAM too
  - Repeatable/ deterministic
  - 2-3 minutes/test x 2/3 tests x 2 boards
  - Normalized energy

Normalized power results of the benchmarks

- ACG
- BIT
- FIR
- CRC
- COMPRESSION
- FFT
- PETRI
FPGA prototype with PTEA

- Correlation results between power estimations of LEM and physical measurements: ~95%
Energy-aware OS prototype

- GEMSCLAIM OS prototype
  - Heterogeneous Multi-processing support
  - SystemC simulation
  - FPGA prototype

- Portable Execution Supervisor
  - Execution controlling daemon on Linux systems
  - ARM big.LITTLE (ODROID-XU3)
    - Exynos 5 Octa
    - x86-64 (Intel servers)

- Energy-awareness
  - Account energy per-thread and on different HW components
  - Control execution for reducing energy consumption
Resource Allocation and Power Management Interface (RAPMI)

- Ported on several platforms
- LEM (SystemC and FPGA)
  - CPU, Memory
  - Sampling period <1us
- ODROID board sensors
  - CPU, Memory, GPU
  - Sampling period ~262us
- Intel RAPL
  - Hardware counters
  - Sampling period ~1ms
Energy-aware scheduling

Energy slices (first-class resource)

- Application A
- Application B

Processor time slices (within energy budget)

- DSP
- RISC
- RISC
- RISC

CHIST-ERA Projects Seminar, Bern, Switzerland, April 28-29, 2016
Multi-objective energy-performance optimization

- Target combined energy-performance goals
  - Include SLA information from OpenMP+ extensions
  - Multi-program execution
- Monitor execution of the workload
  - Programs enter and exit
  - Periodically probe programs through the GEMSCLAIM runtime
    - Scalability (multi-threading)
      - Guide DCT
    - Slowdown (frequency throttling)
      - Decide DVFS
Prototype evaluation

- ODROID-XU3 platform
  - Power sensors
    - CPU and Memory
  - DVFS
  - ARM big.LITTLE heterogeneity
    - A15 and A7 cores
- Multi-program execution
  - Workload generator
    - MediaBench, BOTS, NAS, Rodinia
- Baseline
  - Unmanaged
- Optimizers
  - Energy only (E-only)
  - Performance (execution time) only (P-only)
  - Multi-objective Probing Guided Optimizer (MO)
    - EDP target
Results

- MO
  - vs. Unmanaged
    - ~3x less energy
    - ~2x faster execution
  - vs. P-only
    - ~30% less energy
    - ~90% of performance
  - vs. E-only
    - ~6x faster execution
    - ~30% more energy

![Diagram showing performance and energy comparisons](attachment://diagram.png)
Compiler and runtime system

- The GEMSCLAIM Compiler is a source-to-source C compiler
  - supports C programs annotated with OpenMPE
  - OpenMPE is an energy-aware extension to OpenMP

Programs are divided into meaningful code regions
  - division either by manual annotation or automatically (e.g. based on parallel regions)
  - individually tunable and annotated with additional metadata
  - metadata is derived from analysis and/or user annotations

CHIST-ERA Projects Seminar, Bern, Switzerland, April 28-29, 2016
Compiler runtime system

- Compiler runtime available on all relevant platforms
  - GEMSCLAIM virtual platform & ARM boards (e.g. ODROID XU+E) & X86/64-Linux reference version

- Supports *upcalls* from OS/low-level layer
  - E.g. `irt_set_dop(uint32)` allows OS to control per-program degree of parallelism
    → More effective parallelism management than simple thread packing!

- Performs *downcalls* to OS/low-level layer
  - E.g. for application-level frequency scaling control
    → Can make use of information not available at the OS layer!
OpenMPE – OpenMP for energy

- Set of extensions published at IWOMP 2015
  (*Application-level Energy Awareness for OpenMP*)
  - Region construct (handling code outside OpenMP regions)
  - Objective clause (defining multiple optimisation objectives and constraints)
  - Param clause (define tunable parameters for compiler optimisation)

- In addition to energy, power and time: *quality of service* constraints
  - Can be mapped to user-defined parameters
  - Example usage in video decoder:

```c
#pragma omp parallel for schedule(dynamic)
    objective(E : T<1/f_rate; Q<3) param(scaling, range(1:8:1))
for (int y=0; y<rows; y+=2*scaling)
for (int x=0; x<cols_2; x+=scaling) {
    ...
    if(scaling > 1) { ... }
}
```
e-optimizer

- Multi-objective optimization for OpenMPE programs
  - Dynamically searches best configuration for given goals and constraints
  - Combines random sampling (to prevent local minima) with multi-dimensional hill climbing (to quickly converge)
  - Up to 77% energy savings on mobile and 31% on desktop

![Graph](image1)

*Mobile platform, 704x576 resolution*

![Graph](image2)

*Desktop platform, 1408x1152 resolution*
A Multi-Objective Auto-Tuning Framework for Parallel Codes Herbert Jordan, Peter Thoman, Juan J. Durillo, Simone Pellegrini, Philipp Gschwandtner, Thomas Fahringer, and Hans Moritsch. SC ’12 November 11 - 15, 2012


Adaptive Granularity Control in Task Parallel Programs using Multiversioning. Peter Thoman, Herbert Jordan, Thomas Fahringer. Euro-Par 2013 - Aug 26 Aachen, Germany


GEMSCLAIM Publications 2014

Hardware support for performance measurements and energy estimation of OpenRISC processor, 10th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, May 2015

On the Quality of Implementation of the C++11 Thread Support Library. Peter Thoman, Philipp Gschwandtner, Thomas Fahringer. 23rd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2015. Turku, Finland, March 4-6.


Important GEMSCLAIM project objectives

- Cross layer energy management and optimization for mobile devices: HW/simulator, OS, compiler
- Control trade-off between energy optimization and performance
- Additional 30% energy saving for mobile terminals

GEMSCLAIM achieved all of that.
Conclusions and way forward

- Cross-layer energy management and optimization requires breaking barriers between layers
  - Common abstractions, metrics
  - Synergetic optimization approaches

- Measuring energy consumption remains challenging
  - Machine-specific, intrusive, coarse-grain
  - Hybrid modelling & measurement approaches are the only viable

- Software needs to evolve to break the energy wall
  - Energy optimization should become explicit
  - Sensitivity of energy to software structures is not well understood

- *GEMSCLAIM provides promising solutions for these challenges in a holistic approach, demonstrated on a physical HW/SW substrate*
Sustainability and Valorisation

- Joint FETHPC proposal accepted: H2020 FETHPC AllScale
- Part of the SW under Apache 2.0 license (open source)
- Parts of the GEMSMCLAIM OS Runtime are used in current research projects (FP7 NanoStreams) and are a foundation for future research proposals under the Horizon 2020 research programme
- Parallel simulation technology developed in GEMSMCLAIM will be used for upcoming industry collaboration projects.
- The GEMSMCLAIM VP will be used as a basis for future research projects proposed in the context of Horizon-2020 Research Programme of the EU.

www.gemsclaim.eu