RUN-TIME POWER MANAGEMENT OF MULTI- AND MANY-CORE SYSTEMS

Dr Geoff Merrett
Adaptive Many-Core Architectures and Systems workshop
13-15 June 2018 | York, UK
THE PRiME PROJECT

“Enable the sustainability of many-core scaling by preventing the uncontrolled increase in energy consumption and unreliability through a step change in holistic design methods and cross-layer system optimisation.”

www.prime-project.org
WE ARE MANY-CORE

42 Years of Microprocessor Trend Data

Year

Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Transistors (thousands)
Single-Thread Performance (SpecINT x 10^3)
Frequency (MHz)
Typical Power (Watts)
Number of Logical Cores

Image from www.karlrupp.net/
MANY-CORE PLATFORMS

Intel Xeon Phi - Homogeneous 61 Cores

Odroid XU3 – 8 core big.LITTLE CPU + 6 cores GPU

Nvidia Jetson TK1 - Quad core CPU + 192 cores GPU

Parallella - Dual core CPU + FPGA + 16 cores NoC
THE PRiME PROJECT

www.prime-project.org
RUNTIME POWER MANAGEMENT

PRiME
Q-Learning RTM

ondemand
Linux Governor

LEARNING OPTIMAL DVFS CHOICES

Reinforcement Learning

- Observes the current system state
- Selects an action (V-F pairs)
- Changes the state (workload)
- Leads to a payoff (reward/penalty)

<table>
<thead>
<tr>
<th>STATES (Tasks)</th>
<th>ACTIONS (Power Modes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WD0</td>
<td>P0 128 P1 128 P2 128 P3 128</td>
</tr>
<tr>
<td>WD1</td>
<td>P0 128 P1 128 P2 128 P3 128</td>
</tr>
<tr>
<td>WD2</td>
<td>P0 128 P1 128 P2 128 P3 128</td>
</tr>
<tr>
<td>WD3</td>
<td>P0 128 P1 128 P2 128 P3 128</td>
</tr>
<tr>
<td>WD4</td>
<td>P0 128 P1 128 P2 128 P3 128</td>
</tr>
<tr>
<td>WD5</td>
<td>P0 128 P1 128 P2 128 P3 128</td>
</tr>
</tbody>
</table>

### MANAGING THERMAL (LIFETIME) RELIABILITY

#### Application Data Set

<table>
<thead>
<tr>
<th>Application</th>
<th>Data Set</th>
<th>Average Temperature (Celcius)</th>
<th>Peak Temperature (Celcius)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Linux</td>
<td>Ge et al.</td>
</tr>
<tr>
<td>tachyon</td>
<td>set 1</td>
<td>69.2</td>
<td>52.6</td>
</tr>
<tr>
<td></td>
<td>set 2</td>
<td>50.5</td>
<td>44.5</td>
</tr>
<tr>
<td></td>
<td>set 3</td>
<td>50.8</td>
<td>44.7</td>
</tr>
<tr>
<td>mpeg2_dec</td>
<td>clip 1</td>
<td>36</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>clip 2</td>
<td>35.6</td>
<td>34.4</td>
</tr>
<tr>
<td></td>
<td>clip 3</td>
<td>34.3</td>
<td>34.4</td>
</tr>
</tbody>
</table>

**Average MTTF improvements:** 5x (thermal aging); 4x (thermal cycling)

OVERVIEW

Applications
• From single > sequential > concurrent execution

Offline Characterisation
• Can we improve RTM through offline characterisation?

Towards Many-Core
• How do RTM approaches scale with number of cores?

Novel Platforms
• Can our RTM approaches be applied to novel platforms?

http://www.prime-project.org/
RTMs and Application Workloads

From single > sequential > concurrent execution
QUALITY OF EXPERIENCE

• User cares about **observable performance**
  – Responsiveness, battery life, consistency, uninterrupted service
  – Doesn’t really care about FLOPS, FPS, bandwidth, latency (QoS)

• Therefore, optimise for **quality of user experience** (QoE)
  – “good-enough” performance
  – Minimum energy usage


QUALITY OF EXPERIENCE

Example Scenario
QUALITY OF EXPERIENCE
Workload Classification

<table>
<thead>
<tr>
<th>Applications</th>
<th>Type of QoE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>Throughput</td>
</tr>
<tr>
<td>Video</td>
<td>Throughput</td>
</tr>
<tr>
<td>Application Loading</td>
<td>Latency</td>
</tr>
<tr>
<td>Web Page Loading</td>
<td>Latency</td>
</tr>
<tr>
<td>Downloading a File</td>
<td>Latency</td>
</tr>
<tr>
<td>3D Gaming</td>
<td>Throughput</td>
</tr>
<tr>
<td>Word Processing</td>
<td>Latency</td>
</tr>
</tbody>
</table>

- Delay of
  - <0.1 s appears instant,
  - 1 s becomes noticeable
  - 10 s become disruptive

- Types of QoE:
  - Latency sensitive - complete workload in short time period
  - Throughput sensitive - complete at minimum rate

QoE CHARACTERISTICS

Latency Sensitive

Inverse exponential

Throughput Sensitive

Sigmoid function
TUNING DPM/RTM PARAMETERS

• Tune governor parameters for the executing (interactive) workload
• Account for variability in access times and user input
• Prediction/detection dependent
• Energy saving/QoE improvement compared to ‘default’, e.g.
  – 13% energy saving
  – 27% QoE improvement
  – 9% energy + 15% QoE

Exynos-5422 A15/A7, Android 6.0
Google Chrome browser workloads
Touch input emulation
Network throttling (UL, DL, RTT latency)

EXECUTING MULTIPLE APPLICATIONS

- Workload and performance variation due to:
  - Changes within an application
  - Changing applications (sequential execution)
  - Overlapping applications (concurrent execution)

DETECTING WORKLOAD CHANGES

- Density ratio-based statistical divergence between overlapping sliding windows of CPU cycles
- Use this information to clear learning table (i.e. start afresh)

TRANSFERING LEARNING

- Detect workload changes
- Transfer knowledge where possible
- Learn again fresh when not

RTM FOR CONCURRENT EXECUTION

- Approaches so far instrument a single application executing at a time
- How can we manage multiple applications executing concurrently?

Online vs Offline

Can we improve RTM through offline characterisation?
RTM FOR CONCURRENT EXECUTION

MRPI (Memory Reads Per Instruction)

- Supports concurrent execution of applications
- Inter-cluster Thread-to-core Mapping (ITM).
- MRPI informs DVFS control

MODEL-BASED RTM: HETEROGENEITY

Heterogeneous Platforms

Run-time changes in:
- Performance requirements
- Application workload changes

Towards Many-Core

How do RTM approaches scale with number of cores?
In this document, we discuss Intel's 64-core champion in depth on Xeon Phi.
MODEL-BASED RTM

Stereo Matching Application: http://github.com/PRiME-project/PRiMEStereoMatch

- Processes still images, video or a camera feed
- OpenCL supported
- Includes test datasets

MODEL-BASED RTM
Model Building

MODEL-BASED RTM

Runtime Management

ENERGY RTM ON HPC SYSTEMS

- Applications targeted for HPC are usually multi-threaded
- Modern HPC often based on Non-Uniform Memory Access (NUMA) architecture
- Our Approach:
  - Platform characterized offline
  - Workload estimated based on memory-intensity, thread synchronization contention, NUMA latency
  - V-f determined using binning, while accounting for contention due to concurrent execution

Illustration of various steps in the proposed approach

An example of V-f setting selection using binning-based approach

ENERGY RTM ON HPC SYSTEMS

- Xeon E5-2630 (12 cores, 24 threads) and Xeon Phi 7620P (61 cores, 244 threads); NAS and Rodinia benchmarks

- Proposed (Prop) approach achieves energy savings of up to 81% (Xeon) and 61% (Phi) compared to Linux’s governors

- Outperforms Sundriyal et al. by 10% in energy efficiency and 3.7% in performance

RTM of Novel Platforms

Can our RTM approaches be applied to novel platforms?
RTM ON SPINNAKER

- Implemented 4 RTMs
  - **User (G1)**: user-defined static $f$
  - **On-demand (G2)**: Highest $f$ when CPU load is high, lowest when it’s low
  - **Conservative (G3)**: Increase or decrease $f$ by fixed step according to load.
  - **Proposed (G4)**: As G3, but using a non-linear $f$ step

<table>
<thead>
<tr>
<th>App.</th>
<th>Res.</th>
<th>Governor</th>
<th>Timing (ms)</th>
<th>Energy Consumption (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
</tr>
<tr>
<td>A1</td>
<td>vga</td>
<td>955</td>
<td>976</td>
<td>976</td>
</tr>
<tr>
<td></td>
<td>xga</td>
<td>2444</td>
<td>2498</td>
<td>2498</td>
</tr>
<tr>
<td>A2</td>
<td>vga</td>
<td>2670</td>
<td>3080</td>
<td>3080</td>
</tr>
<tr>
<td></td>
<td>svga</td>
<td>4408</td>
<td>4737</td>
<td>4737</td>
</tr>
<tr>
<td></td>
<td>xga</td>
<td>7114</td>
<td>7342</td>
<td>7342</td>
</tr>
<tr>
<td>A3</td>
<td>vga</td>
<td>437</td>
<td>454</td>
<td>454</td>
</tr>
<tr>
<td></td>
<td>xga</td>
<td>674</td>
<td>696</td>
<td>696</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1111</td>
<td>1150</td>
<td>1150</td>
</tr>
</tbody>
</table>

RTM ON THE GRACEFUL PLATFORM

Approach

• Opportunity for Hierarchical RTM
  – Local RTM (DVFS, local mapping etc) on each node
  – Higher level ‘strategic’ RTM (mapping within cluster, migration, load balancing etc) in clusters
  – Potential for a third level negotiating between clusters

• See our (early) demonstration of this

Poster and Demo Session
Thursday 14:30
Example Application

- Face/Object Detection/Classification
- Uses OpenCV classifiers
  - Detect faces/animals/objects
  - Classify gender
  - Estimate age
OPEN SOURCE TOOLS
POWMON: STABLE POWER MODELLING

www.powmon.ecs.soton.ac.uk

Our stable approach achieves a low average error and narrow error distribution compared to existing techniques.

Training: Small set of 20 workloads
Testing: Full set of 60 workloads


1. **Run workloads**
   @ different DVFS levels

39 workloads used: MiBench, LMBench, Roy Longbottom, ParMiBench and ALPBench

2. **Record**
   • PMCs
   • Power, Voltage, Temperature, etc.

3. **Choose PMCs**
   Hierarchical cluster analysis, Correlation matrix analysis, Exhaustive search, etc.

4. **Build Model**
   • OLS multiple regression
   • Considers collinearity and heteroscedasticity
   • “sensible” equation

5. **Validate**
   • K-fold cross validation
   • $R^2 : > 0.99$
   • Error: 2.8 – 3.7%

6. **Uses**
   • OS Run-time management
   • Reference for research
   • gem5 add-on

---

POWMON: METHODOLOGY
Increasing RTM Usability/Comparative Evaluation

### Hardware Layer
- **GPU**
- **DSP**

### Application Layer
- **App 1**
- **App 2**
- **App N**

#### PRiME App Knobs:
- Parallelism (No. Kernels)
- Heterogeneity

#### PRiME Dev Knobs:
- V-F
- Core Affinity

#### PRiME App Monitors:
- Performance (fps)
- Accuracy (error rate)

#### PRiME Dev Monitors:
- Power
- Energy
- Temperature

<table>
<thead>
<tr>
<th>Plat. Const. Space Type</th>
<th>For</th>
<th>No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>disc GOVERNER FREQ FREQ_EN PMC_CTRL</td>
<td>A7 cluster GPU DVFS A7 cores 16 A15 cores 24</td>
<td></td>
</tr>
<tr>
<td>cont TEMP CYCLE PMC_CTRL</td>
<td>A15 cores GPU A7 cores 4 A15 cores 24</td>
<td></td>
</tr>
<tr>
<td>cont VOLT PMC_CTRL</td>
<td>A9 cluster, peripherals FPGA, peripherals 4 3</td>
<td></td>
</tr>
<tr>
<td>cont VOLT PMC_CTRL</td>
<td>A9 cluster, peripherals FPGA, peripherals 5 4 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Const.</th>
<th>Space</th>
<th>Allowed/target values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobi Iterations</td>
<td>knob</td>
<td>disc</td>
<td>( N \in [1, \infty) )</td>
</tr>
<tr>
<td>Jacobi Data type</td>
<td>knob</td>
<td>disc</td>
<td>{ float, double }</td>
</tr>
<tr>
<td>Jacobi Device type</td>
<td>knob</td>
<td>disc</td>
<td>{ CPU, GPU/FPGA }</td>
</tr>
<tr>
<td>Jacobi Throughput</td>
<td>mon</td>
<td>cont</td>
<td>( R \in [10, \infty) )</td>
</tr>
<tr>
<td>Jacobi Error</td>
<td>mon</td>
<td>cont</td>
<td>( R \in (-\infty, 10^{-12}) )</td>
</tr>
<tr>
<td>Video decoder Throughput</td>
<td>mon</td>
<td>cont</td>
<td>( R \in [25, \infty) )</td>
</tr>
<tr>
<td>Whetstone Threads</td>
<td>knob</td>
<td>disc</td>
<td>( N \in [1, \infty) )</td>
</tr>
<tr>
<td>Whetstone Throughput</td>
<td>mon</td>
<td>cont</td>
<td>( R \in [2.5, \infty) )</td>
</tr>
</tbody>
</table>
CONCLUSIONS

Runtime Power Management
• Single > multiple > concurrent applications
• Online vs offline+online approaches
• >> Number of cores
• COTS > Novel multi-/many-core platforms
• Homogeneous vs Heterogeneous platforms

Tools and Support  www.prime-project.org
• PowMon power estimation
  www.powmon.ecs.soton.ac.uk
  www.gemstone.ecs.soton.ac.uk
• PRiME RTM Framework
  github.com/PRiME-project/PRiME-Framework
• PRiMEStereoMatch application
  github.com/PRiME-project/PRiMEStereoMatch

http://www.prime-project.org/
Any Questions?

Dr Geoff V Merrett
Associate Professor | Head of Centre

Centre for IoT and Pervasive Systems
Tel: +44 (0)23 8059 2775
Email: gvm@ecs.soton.ac.uk  |  www.geoffmerrett.co.uk
Highfield Campus, Southampton, SO17 1BJ UK