Research Brief
PRiME
run smarter, live longer
Power-efficient, Reliable, Many-core Embedded systems
www.prime-project.org

Adaptive energy optimisation in heterogeneous systems

***PRiME** is a five year EPSRC funded research programme (2013-2018), in which four UK universities address the challenges of **power consumption** and **reliability** of future high-performance embedded systems utilising many-core processors.*

Energy Efficiency in Heterogeneous Systems

Energy efficiency is an important design objective in embedded systems. To accommodate a broader range of applications, modern embedded systems often consist of heterogeneous computing resources, e.g. CPUs, GPUs, DSPs, dedicated hardware - each of which has different energy and performance trade-offs. These computing resources exercise the application tasks differently, generating varying workloads and energy consumption. Hence, minimizing energy consumption in these systems is challenging, as it requires continuous adaptation of application task mapping between the computing resources, as well as dynamic voltage/frequency scaling (DVFS). Existing approaches to energy optimization lack such adaptation.

Adaptive Energy Optimisation

PRiME has developed a novel **adaptive** energy minimization approach for embedded heterogeneous systems. This is based on a **runtime model**, generated through **regression-based learning** of energy and performance trade-offs between different computing resources in the system. Using this model, an application task is mapped to a suitable computing resource during runtime, to minimize energy consumption for the application's performance requirement. Also used is DVFS control which adapts to the performance and workload variations.

Demonstrator: Xilinx Zynq Platform

The PRiME approach has been implemented on a Zynq ZC702 platform - consisting of CPUs, DSPs and FPGA:

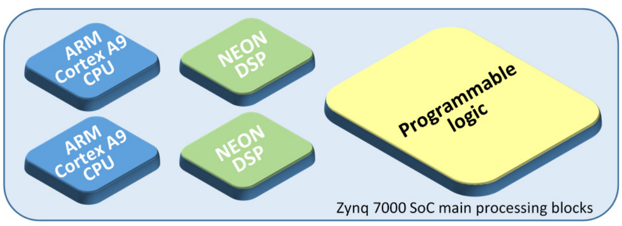

Figure 1. Xilinx Zynq SoC main processing cores

Using several image processing applications as case studies, the PRiME approach has achieved significant energy savings (**more than 70% in some cases**, i.e. from 43mJ per frame to 13 mJ per frame), when compared with existing techniques.

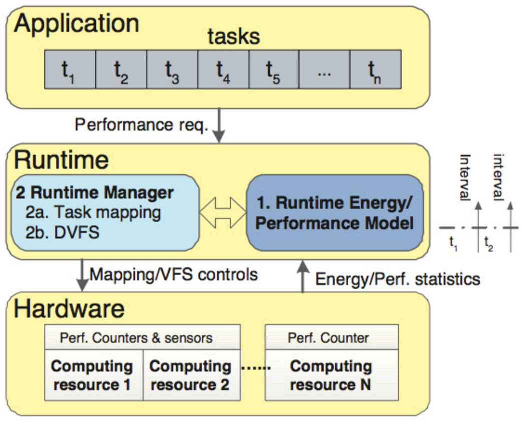

Figure 2. PRiME's multi-layer approach to energy minimization

Figure 2 shows the block diagram of the energy minimization approach, highlighting the interactions between application, runtime and hardware. The application consists of a number of computation tasks. The performance requirement of the application is communicated to the runtime layer, which consists of two components:

- a learning-based energy/performance model
- a runtime manager

The energy/performance model is derived and learnt through runtime measurements from the different computing resources in the system. This model then subsequently guides the runtime manager to carry out optimized application task mapping and DVFS controls, while meeting the application-specified performance requirements.

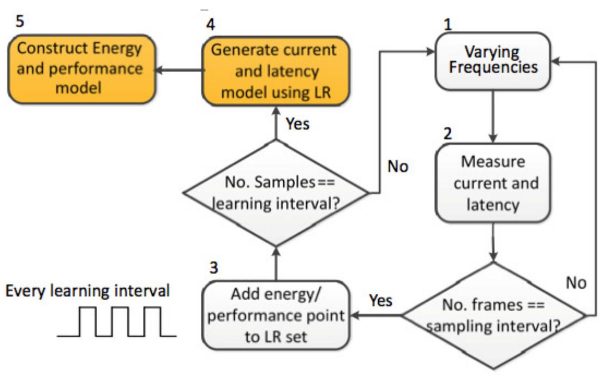

Figure 3. Flowchart of runtime modelling

Figure 3 shows the flowchart of the runtime modelling in 5 steps. It starts by varying the operating frequencies of the computing resources (step 1). For every frame, the current and latency measurements are read from the power sensors and performance counters (step 2). If the number of frames is equal to the sampling interval, the measured current and latency values are averaged over the period to produce a stable learning sample. The sample is then used to generate the model using linear regression (step 3). Such sample collection and hypotheses testing is continued until the learning interval is reached. After this, the current and latency models are generated for the given computing resource (step 4) and these models are used as components to construct the energy/performance model (step 5).

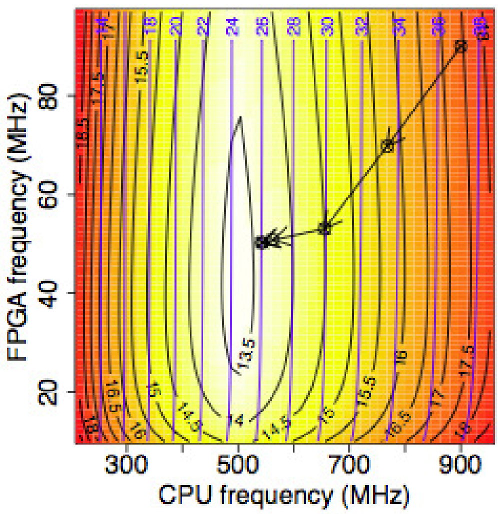

Figure 4. CPU/FPGA energy and performance trade-offs

Figure 4 shows the contour maps of the energy (black contour) and performance (blue contour) trade-offs between CPU and FPGA for different operating frequencies; yellow represents lower energy consumptions, red represents higher energy consumptions. The contours represent the energy/performance models. Gradient descent is used to find the optimal operating points for both CPU and FPGA. The black arrow lines show the optimization path for performance constrained (26 fps) energy minimization. It starts with the highest frequency for both FPGA and CPU and quickly finds the optimal operating point (in this example, FPGA 50MHz and CPU 530MHz) in only 5 steps.

Results

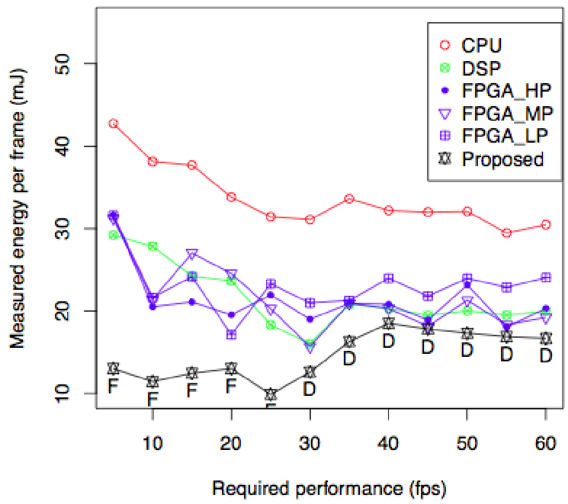

Figure 5. PRiME vs. linux ondemand

Figure 5 compares the PRiME approach with different implementations of **Linux ondemand:**

- CPU with Linux ondemand (red)
- DSP with Linux ondemand (green),
- FPGA with 10 MHz frequency (purple square)
- FPGA with 50 MHz frequency (purple triangle)
- FPGA with 100 MHz frequency (purple dot)
- PRiME's adaptive approach (black star)

The chosen application task mappings are indicated alongside the plotted points with the first letter of the computing resource (C: CPU, D: DSP and F: FPGA). The PRiME approach detects the change in application performance requirements and adaptively selects the most energy-efficient computing resource using optimized task mapping (amongst the CPU, DSP and FPGA) and DVFS controls. As a result, *this adaptation achieves significant reductions in energy compared to linux ondemand for the varying performance requirements.*

More information

Visit the PRiME programme web site, including the opportunity to sign-up for programme updates, or contact the PRiME Impact Manager:

Gerry Scott
Email: gerry.scott@soton.ac.uk
Tel: +44 (0)23 8059 2749

www.prime-project.org