Benchmarking of Dynamic Power Management Solutions

Frank Dols
CELF Embedded Linux Conference
Santa Clara, California (USA)
April 19, 2007
Why Benchmarking?

The Scientific approach

- Benchmarking is science!
- Design an experiment
  - Choose / create benchmark and select system parameters
- Record all information needed to reproduce it
- Perform the experiment
- Record the results
- Test results for correctness / plausibility
- Present results fully qualified and with any supporting context
- Draw conclusions

Science should be done objectively
- ...but we all want our products to look good
  - Realistic or best case results?
    - Doesn’t matter, as long as it’s clearly documented

From Here to There, 2000whatever

Vendor

What do we get?

NXP
Presentation Outline:

- The context;
  - Power management concepts.
  - Hardware and software.

- Benchmarks;
  - The process.
  - Metrics.
  - Findings.

- Conclusions.
  - Relation to other work.
  - What’s next.
Area Of Interest

- Mobile/portable devices mostly exist of:
  1. Battery.
  2. Storage (flash disk, hard disk, ...).
  3. Display (LCD, ...).
  4. Speaker.
  5. Broadband I/O (Bluetooth, UMTS, ...).
  6. Processing (CPU)

- We are initially only focusing on the optimization of the “Processing” power consumption.
  - As next, we are also taking 1-5 into account.
Power Measurements

A board under test

Measurements equipment: high precision DMM (digital Multi-Meters)

Mp3 playback – LabView measurements
Energy Consumers

- Energy saving methods trade performance or functionality for energy:
  - Scaling performance of processors, memories and buses;
  - Using various stand-by modes of peripherals.

- Energy saving is about supplying the right amount of performance at the right time.

- However, the future is unknown!

Energy consuming components in a typical audio/video playing mobile device.
Power Dissipation Basics

\[ E = \int_0^t \left( C (V_{DD})^2 f_c + V_{DD} I_Q \right) dt \]

- Reduce capacitance switched.
- Reduce switching currents.
- Reduce operating voltage.
- Reduce leakage current in active and standby modes of operation.
- Reduce operating voltage.
Dynamic Voltage Frequency Scaling (DVFS)

- Scales performance according to demand (using an estimation of future workload).
- Based on the fact that energy per clock cycle rises with frequency:
  \[ P \propto V^2 \cdot f \]
- Implemented by switching between operating points (voltage and frequency pairs).

Diagram:

- Applications, task scheduler, or other source of information
- Workload estimation
- Clock Generator
- Voltage Controller
- Processor or memory
DVFS: How Does it Save Power

100% CPU usage

- Performance
- Power
- Time

50% CPU usage, no DVFS

- Full speed
- High f,V
- Performance
- Power
- Time

Energy used

50% CPU usage, with DVFS

- 50% speed
- Idle
- Optimal!
- Performance
- Power
- Time

Low f,V

Energy used

Optimal!
Performance Prediction Methods

Interval-based: use CPU-usage of previous interval(s) to determine frequency for next interval.

Problem: late reaction on changing workloads → missed deadlines
**Performance Prediction Methods**

**Application-directed:** use information from the application to change frequencies.

**Problems:** requires changes in the application; not always possible (interactive applications).
Dynamic Power Management (DPM) Concept

- DPM software connects to OS kernel and collects data, with collected data, and usage of policies, try to predict future workload.
- Multiple policies categorize software workload.
- Single global prediction of future performance is made.
Application directed DVFS

- Two main groups of mobile applications:
  
  **Interactive**
  - Internet browsing
  - Gaming

  **Streaming**
  - Audio decoding
  - Video decoding

Future workload unknown (depends on user input)

Future workload known to application
Presentation Outline:

- The context;
  - Power management concepts.
  - Hardware and software.

- Benchmarks;
  - The process.
  - Metrics.
  - Findings.

- Conclusions.
  - Relation to other work.
  - What’s next.
Benchmark Process

- Plan:
  - Define Key Performance Indicators (KPI).
  - Define test to measure KPI.

- Do:
  - Measurement on evaluation platform.

- Check:
  - Analyze gained results and can they be explained.
  - When necessary, cross verify with Vendor.

- Act:
  - Take necessary actions on gained results.
Key Performance Indicators (KPI)

- Memory usage.
  - Footprint of DPM framework and administration.

- CPU usage.
  - Cycles consumed by DPM framework.
  - System behavior, predictability & reproducibility.

- System idleness.
  - Prediction of optimum frequency (minimization of idleness).

- Real time behavior.
  - Application deadlines missed.
  - Responsiveness (latency on events).
  - DPM Policy prediction accuracy.
Test / Use Cases

- Synthetic (fine-grained) benchmark,
  - Simulate different workload levels,
  - Test corner cases.
    - LMbench as available from Open Source; lat_proc, lat_syscall, memory, clock, idle, …

- Application level (coarse-grained) benchmark:
  - Whetstone & Dhrystone (artificial system load).
  - Hartstone (real time behavior).

- Video decoding,
  - ffplay mpeg video decoding use case.
    - use ffplay, part of ffmpeg.

- Audio decoding,
  - mp3 audio decoding use case.
    - use ARM MP3 decoding library.
Metrics for SW based Benchmarking

- Missed deadlines;
- Overdue time;
- Idle time;
- Jitter;
- Responsiveness.
Missed Deadlines

- When operating frequency is too low, deadlines are missed:
Overdue Time

- Are missed deadlines caused by timing inaccuracy, or by performance deficiency?
Idle Time

- When operating frequency is too high, extra slack time is introduced, and power is wasted:

![Diagram showing optimal scenario]
- Execution times vary, so time of completion varies as well:

Arrival times are periodic, but completion times vary → Jitter
In Summary, DPM Metrics

- Missed deadlines (OS schedule);
- Overdue time;
- Idle time;
- Jitter (fluctuation in Completion time);
- Responsiveness.
NXP’s DVFS Benchmark Platform

- Energizer I SoC;
  - ARM1176JZF-s,
  - DVFS-enabled (core also),
  - CPU voltage can be set in 25 mV increments,
  - CPU frequency can be set using 2 PLL’s (300MHz and 400MHz by default) and a divider.

- Energizer I Software;
  - Linux 2.6.15,
  - CPUfreq and PowerOP.
  - MV’s DPM framework ported but not used.
CPU Usage Characteristics

- CPU usage, applied policy reacts slow on required performance.
DPM; Synthetic Workload

- (1st) Synthetic Workload
- (2nd) DPM, Predicted Performance Requirement
- (3rd) Actual CPU Usage

Headroom

Late Response

(Exponential Weighted Average Policy)
Application Level Benchmark

- Video playback (DivX).
  - Open-Source ffplay using frame-buffer device.
  - Instrumented with time measurements.

- Evaluation:
  - How many hard deadlines are missed with & without DPM.
**ffplay Deadlines**

- **ffplay deadline**: time interval in between displaying of successive (decoded) frames.
  - For 10 fps movie -> the deadline is 100 ms.

- With DPM activated, fluctuation increases.

- **REMARK**: video output (display) on CompactPresenter via Compact flash interface, results in high fluctuation oops !! (see next slide).
Kernel & User Space Activity

- While running ffplay, sample the CPU activity in both kernel and user space (based on /proc/stat).
- Kernel activity dominates during the video playback.
DPM; Video Decoding

System Overload

Power Savings

Performance Impact

Policy Overhead

(Exponential Weighed Average Policy)
Video Decoding Jitter

Clock scaling causes jitter increase

Jitter metric gets polluted by high number of deadline misses
Video Decoding and Hints

CITY_5fps.avi

- No DPM
- DPM, No hints
- DPM, with Hints
Comparison

When application-directed hints are enabled:
- idle time reduction increases,
- effective in a wider frame rate window.

DPM without hints is only effective at low workloads.
Presentation Outline:

- The context;
  - Power management concepts.
  - Hardware and software.
  - Relation to other work.

- Benchmarks;
  - The process.
  - Metrics.
  - Findings.

- Conclusions.
  - Relation to other work.
  - What’s next.
Technology Conclusions

- **Dynamic Power Management (DPM).**
  - Added value is in policies.
  - Performs best on low workloads.
  - Adaptation to changing workload is heavily dependent on chosen policy.
  - As application developer, you have to develop your own policies.
  - Be aware, policy adds overhead.

- **Applications cannot feed data directly into DPM.**
  - Performance prediction based on logging application history does not always work.
  - The best performance prediction may come from the application itself (feed forward)!
  - Hinting can give improvements over an interval based approach.
Conclusions

- Interval based DPM at CPU-level results in higher CPU utilisation, but seems only really effective at low workloads;

- Precise energy saving figures should be measured using HW measuring tools, but SW benchmarks give good insight in saving potentials and consequences on real-time behaviour.

- SoC is not the biggest power consumer.
  - Backlight, DC/DC converters & power amplifiers are big consumers, by optimizing these, most can be gained.
What’s Next

1. CELF (you guys), MIPI PM working group
   - Matt Locke and his team


3. Correlation between hardware and software;
   - What to solve at which level?

4. Deadline based scheduling, from priority based to time based.

5. Start contributing to the Community on this.

   Regarding bullet 1, 2 & 3, see extra slides
Overview of Open Source Power Mgt. Software

CPU power management
- Standby (low power mode) when idle
- Dynamic Voltage/Frequency Scaling
  - Task scheduler
  - CPUfreq

Device power management
- Suspend complete system
- (memory) bus frequency scaling
  - APM/ACPI
  - Power OP
- Suspend/Resume of devices
  - Linux Device Model

(also look at Mark Gross presentation)
Mode Transitions

PM policy (performance monitoring, decision taking)

Mode transition in sw domain (state save, sequencing, PM infra control)

Mode transition in hw domain (sequencing, communication)

Mode transition in hw domain (communication, Voltage/Frequency settling, sequencing)

Mode transition in sw domain (warm boot, state restore, sequencing)

ACTIVE mode

LP mode

ACTIVE mode
Improving power management
through co-design of hardware and software

- Other ideas for improving power management
  - Performance counters for other peripherals:
    • Cache miss rate is an indicator for memory bus usage of the CPU.
  - Hardware aided workload prediction algorithms (in embedded FPGA):
    • DVFS algorithm in FPGA, instead of software;
    • Enables fast calculation of estimations and quick operating point changes;
    • Algorithm can be changed for every use case because of the use of an FPGA.

- More hardware acceleration for specific applications:
  • Needs discussion with SW developers in early stage about which parts can be accelerated;
  • Enables running on a lower frequency, and as such saving energy.
To come to the End.
We Started With the Question:

“Why Benchmarking?”

The Scientific approach

- Benchmarking is science!
- Design an experiment
  - Choose / create benchmark and select system parameters
  - Record all information needed to reproduce it
- Perform the experiment
- Record the results
- Test results for correctness / plausibility
- Assemble results, quantify and with any supporting context
- Draw conclusions

- Science should be done objectively
  - ...but we all want our products to look good
  - Realistic or best case results?
  - Doesn’t matter, as long as it’s clearly documented

Do your homework,
don’t bring in the Trojan Horse!
NXP Semiconductors

- Established in 2006 (formerly a division of Philips)
- Builds on a heritage of 50+ years of experience in semiconductors
- Provides engineers and designers with semiconductors and software that deliver better sensory experiences
- Top-10 supplier with Sales of € 4.960 Bln (2006)
- Sales: 35% Greater China, 31% Rest of Asia, 25% Europe, 9% North America
- Headquarters: Eindhoven, The Netherlands
- Key focus areas:
  - Mobile & Personal, Home, Automotive & Identification, Multimarket Semiconductors
- Owner of NXP Software: a fully independent software solutions company
Where to Find Us

Operating systems Knowledge Centre (OKC)

phone: +31 40 - 27 45 131
fax: +31 40 - 27 43 630
email: okc.helpdesk@nxp.com
Web: www.nxp.com
address: High Tech Campus 46
location 3.31
5656 AE Eindhoven (NL)