Energy Discounted Computing On Multicore Smartphones

Meng Zhu & Kai Shen

Atul
Bhargav

Overview

- Energy constraints in a smartphone
    - Li-Ion Battery
- Arm big.LITTLE
    - Hardware Sharing

What is Energy Discounted Computing?

- ➔ Activation of first core consumes incurs much higher power
- ➔ Typical smartphone application have limited parallelism.
- ➔ So we can get rest of the core at deep energy discount.

Multicore Processors

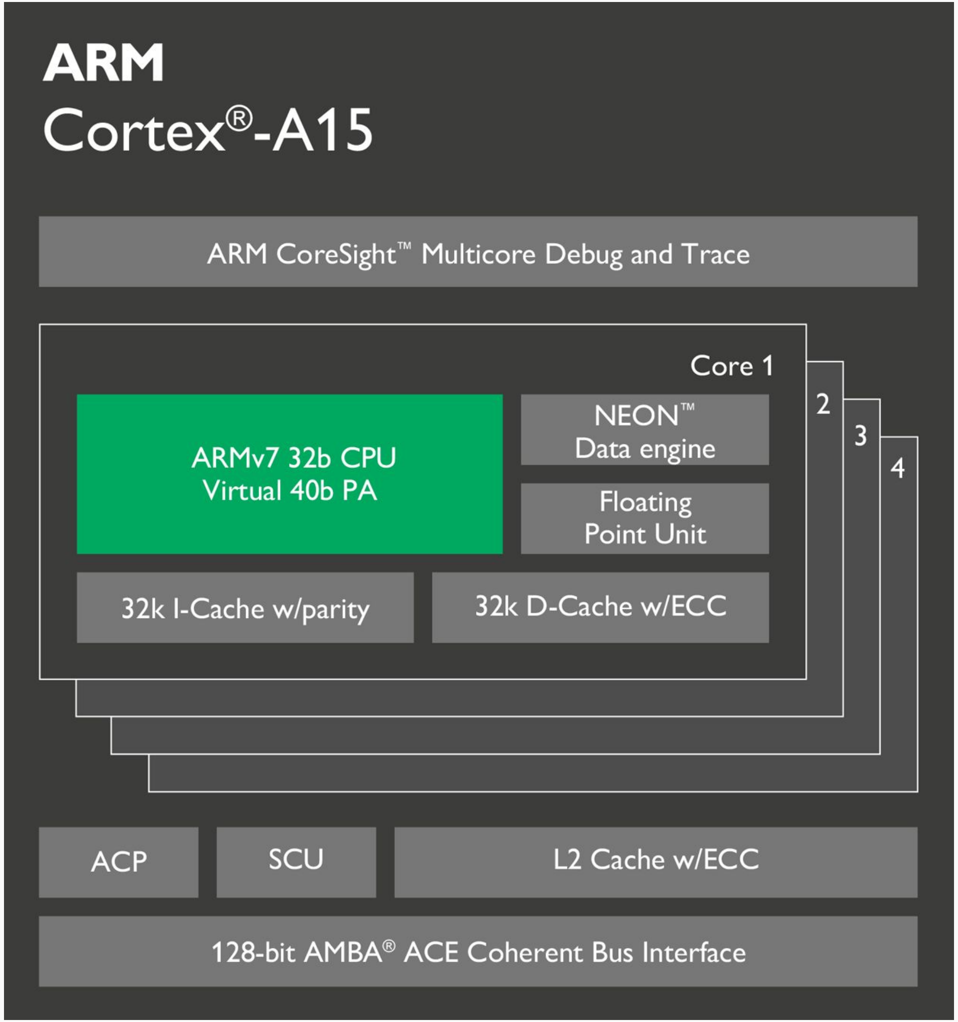
- Latest smartphones are shipping with 4-8 cores
- They have different power saving states.
- Ex: C0, C1, C2.
- They enter different power saving states to adapt to different workloads.
- They use DVFS to achieve different power/performance setting.

Multicores are Energy Disproportional

- Modern processors are good at power gating
  - When the system is idle, most parts of the CPU can be shut down
- Aggressive hardware sharing
  - drive down cost
  - reduce footprint
  - save power
- Example: CPU on one socket usually share power rail and oscillator

ARM Cortex 15

Source: http://www.arm.com/assets/images/Cortex-A15-chip-diagram-16-LG.png



Multicores are Energy Disproportional...

➔ hardware sharing also affects the CPU idle states

State	Name	Power	Target residency	Description
C0	Wait for interrupt (WFI)	403 mW	1 nSec	Processor is clock gated but can respond to cache / TLB maintenance (e.g., L2 snoop) requests without exiting the WFI state.
C1	Individual powerdown	365 mW	1 mSec	Processor is power gated. All state including L1 cache content is lost and the processor is removed from the coherency protocol.
C2	Cluster powerdown	214 mW	4 mSecs	Can only be entered when all processors are in individual powerdown mode. All state including the L2 cache content is lost.

Energy Constraints in a Smartphone

- Slow progress on battery technology, size restrictions
- Quadcore and Octacore processors popularity

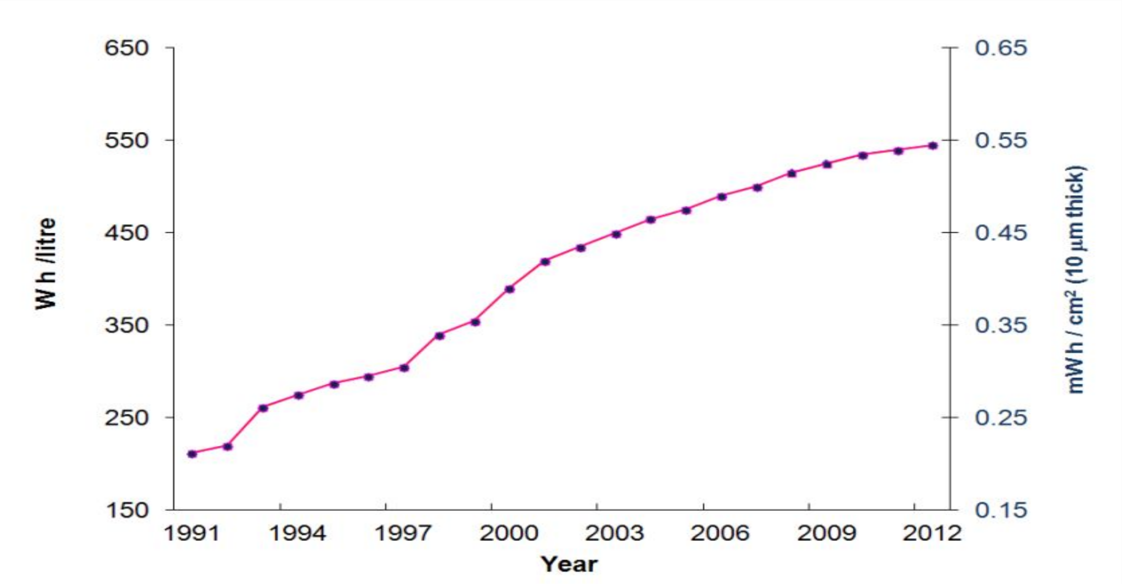
Energy Constraints in a Smartphone

- Slow progress on battery technology, size restrictions
- Quadcore and Octacore processors popularity

Energy Constraints in a Smartphone

- **Slow progress on battery technology**, size restrictions
- Quadcore and Octacore processors popularity

Li-Ion Battery



http://www.intechopen.com/books/energy-storage-battery-materials-and-architectures-at-the-nanoscale

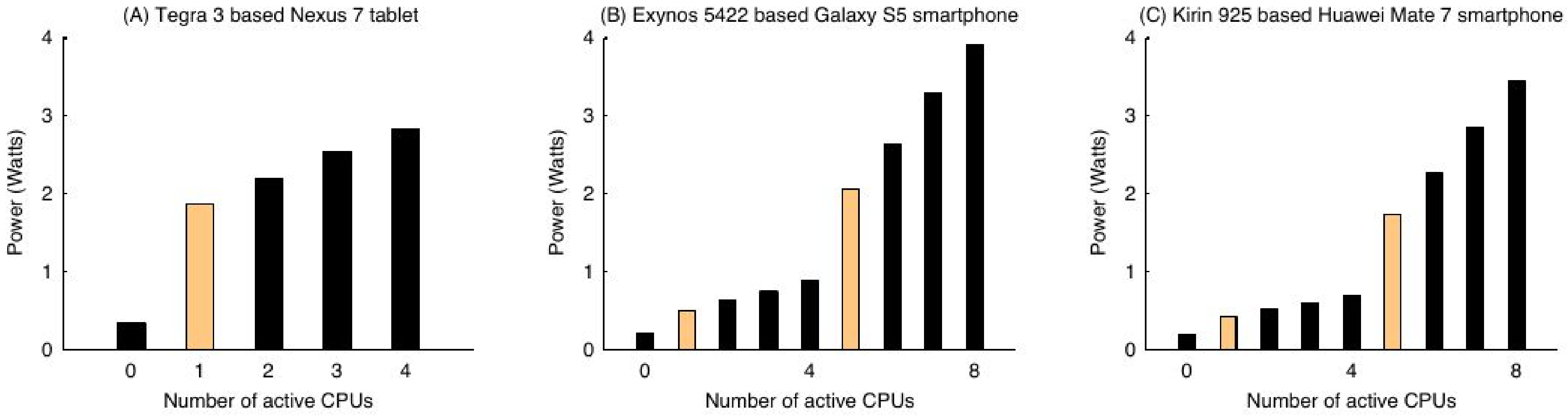
Energy Constraints in a Smartphone

- Slow progress on battery technology, size restrictions
    - Li Ion battery already at 80% energy density
- Quadcore and Octacore processors popularity

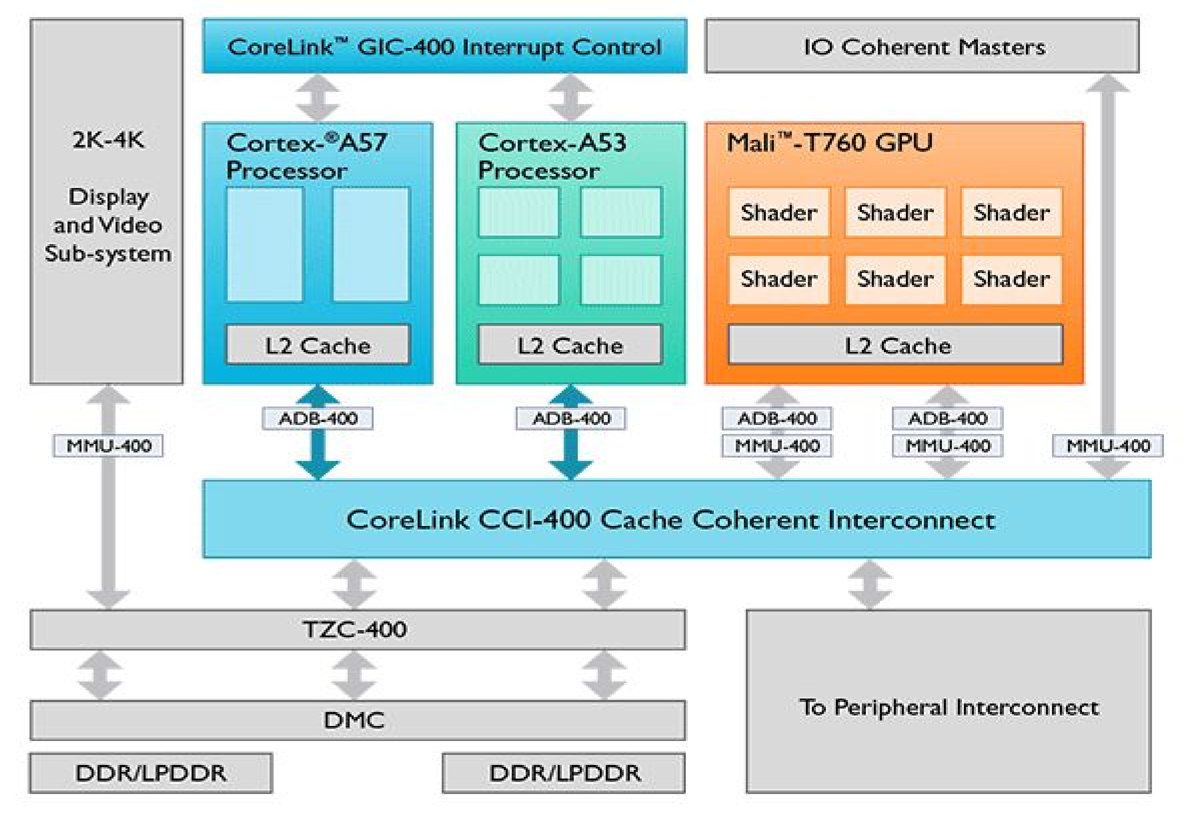
Energy Constraints in a Smartphone

- Slow progress on battery technology, size restrictions
  - Li Ion battery already at 80% energy density
- Quadcore and Octacore processors popularity
  - Heavily energy disproportional on smartphones

Disproportionate Power Consumption



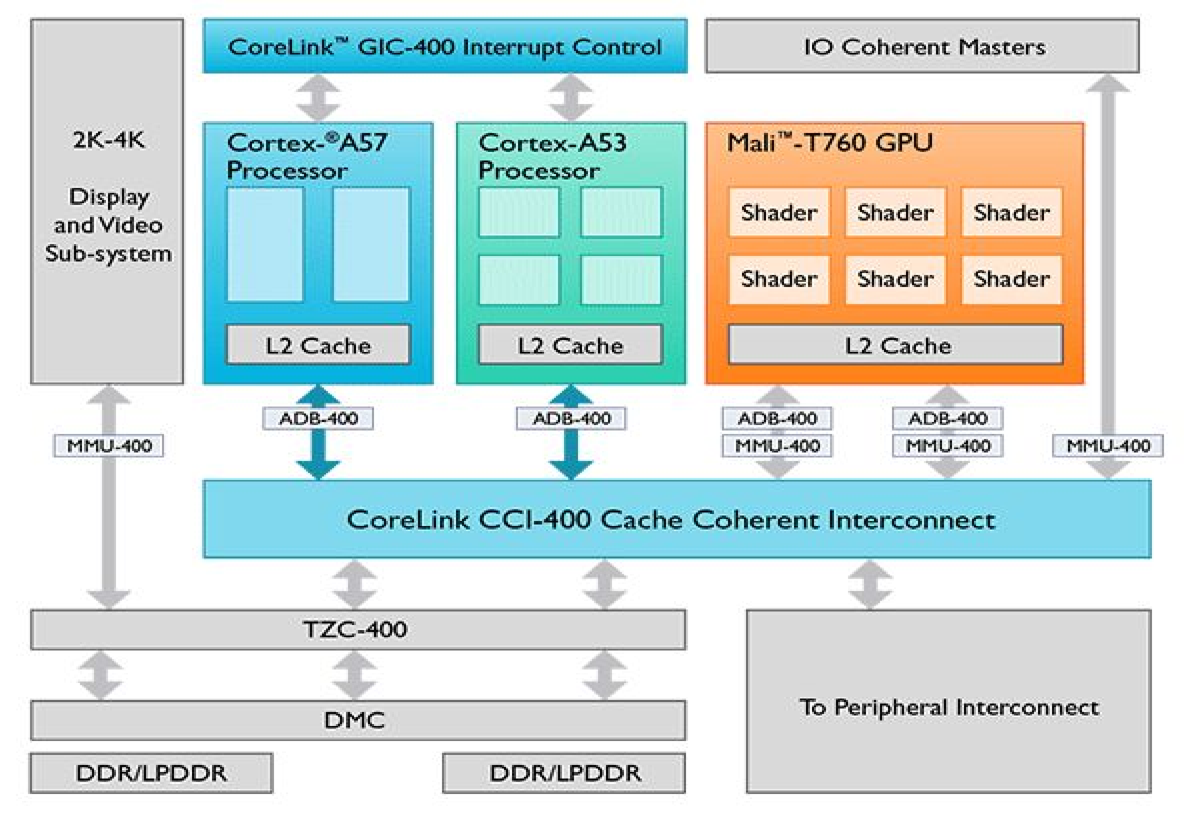
SOC with ARM big.LITTLE



Idle States (‘C’ States)

State	Name	Power	Target residency	Description
C0	Wait for interrupt (WFI)	403 mW	1 nSec	Processor is clock gated but can respond to cache / TLB maintenance (e.g., L2 snoop) requests without exiting the WFI state.
C1	Individual powerdown	365 mW	1 mSec	Processor is power gated. All state including L1 cache content is lost and the processor is removed from the coherency protocol.
C2	Cluster powerdown	214 mW	4 mSecs	Can only be entered when all processors are in individual powerdown mode. All state including the L2 cache content is lost.

SOC with ARM big.LITTLE



Advantages of BigLittle

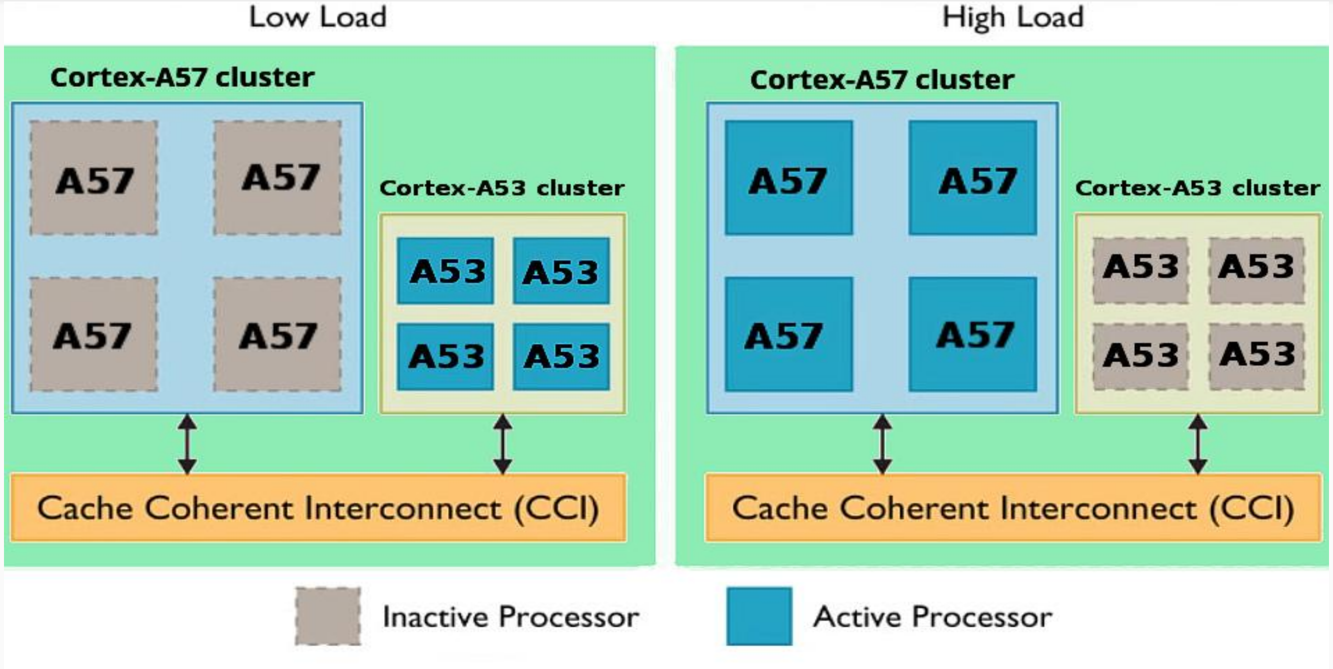
Heterogenous Architecture (Uses all the cores)

- Automatic Task Allocation among the cores
    - Dependent on the work load

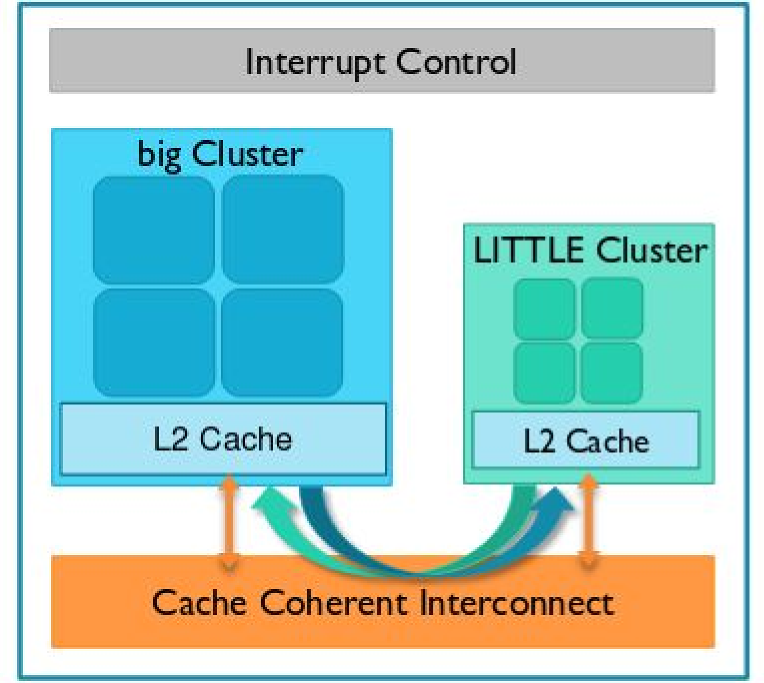
2x higher performance vs. LITTLE only

Up to 75% CPU powersavings vs. big only

ARM Big Little Architecture



ARM Big Little Architecture



Hardware Coherency

- Cache Coherent Interconnect (CCI)
- L1 and L2 snooping between clusters

Use of Multicores on Smartphones?

- ➔ Typical phone apps are built on event-driven, UI-centric framework
- ➔ Don’t have sufficient parallelism to utilize multiple cores simultaneously.
- ➔ Limited multi-processing

Hence,

- ➔ Co-run best effort task.

Best Effort Task(BET)

- ➔ Workloads that are meaningful to the user but do not involve direct interaction.
- ➔ Loose quality-of-service requirements.

Upload & download

- Examples:
  - Syncing data with cloud
  - Posting on social websites
  - Software/Update installation
- Significant energy consumption comes from the transmission module
- CPUs also consume substantial energy
  - Compression/Decompression
  - Encryption/Decryption

System Maintenance Work

- *kswapd* daemon scan for memory pages that can swapped out to free up space
- *dhd_dpc* which analyzes network packets and scans for Wi-Fi hotspots
- Re-compiling the bytecode for better native performance
- May have timing constraints

Background Sensing & Proactive Tasks

➔ Using camera sensors to analyze facial expression or eye movement

➔ Siri can provide recommendations, news and applications even before one asks for it.

While co-execution of applications on multicore processors may improve the energy efficiency, it also risks significant interference on shared hardware resources, memory bandwidth and last-level-cache space in particular, and thereby leads to poor interactive application performance and degraded user experience.

Energy Discounted Computing

POWER STATE PRESERVATION

- CPU idle state, or ACPI “C” state
  - Often long idle gaps between user interactions and CPUs entering deep sleep state.
  - It is crucial to keep best-effort tasks from disrupting these idle periods.
  - During active application executions, due to lack of parallelism, idle CPUs will often enter per-core idle states.
  - These shallow sleep states, do not save much energy.
  - CPU scheduler needs to schedule best-effort tasks opportunistically in accordance with interactive applications.
  - So schedule Best Effort Task only if at-least one core is active

Energy Discounted Computing...

- Core frequency state, or ACPI “P” state
  - CPUs use DVFS to quickly adjust power levels to conserve energy and meet performance needs of different workloads.
  - In our co-run scheme, the system should avoid raising the CPU frequency / voltage levels for best-effort tasks.
  - Otherwise, the extra energy consumption will negate the energy discount.
  - At the same time, such caution should not affect the performance of interactive applications.
  - CPU frequency adjustment should only focus on the needs of interactive applications and ignore the presence of best-effort tasks.

Energy Discounted Computing...

- Smartphone suspension state, or ACPI “S” state
  - Systems in the suspension state consume very little energy by shutting down most parts of the hardware, including the CPU and memory.
  - On some platforms (notably Android), applications can prevent system suspension by making explicit requests to the operating system.
  - Best-effort tasks are not permitted to make such requests. The system should be able to enter the suspension state regardless of best-effort tasks.

Resource Contention Mitigation

- ➔ Co-running tasks on a multicore may slow down each other.
- ➔ One easy mitigation is: adjust CPU scheduling priority(nice value, cpu limit)

But,

- ➔ Due to the hardware resource sharing on multi-core processors, contention could also result from shared hardware resources.
- ➔ Monitor the last-level-cache miss rate using PMU. Contention is identified if the miss rate reaches a threshold

Implementation

- **BUSY** indicates the CPU is running normal tasks(e.g., interactive applications),
- **IDLE** indicates the CPU is in idle state (regardless of the level of idle state),
- **BEST-EFFORT** indicates the CPU is running best-effort tasks,
- **UNDEF** is a transient state (e.g., during context switches).

Implementation

- → **Linux Control Groups**

Implementation

- **Linux Control Groups**
  - Linux Kernel Facility allowing grouping of tasks into tree

Implementation

- **Linux Control Groups**
  - Linux Kernel Facility allowing grouping of tasks into tree
  - Allows the groups for
    - Priority allocation
    - CPU resources
    - Memory B/W
    - Disk
    - Network

Implementation

- **BUSY** indicates the CPU is running normal tasks(e.g., interactive applications),
- **IDLE** indicates the CPU is in idle state (regardless of the level of idle state),
- **BEST-EFFORT** indicates the CPU is running best-effort tasks,
- **UNDEF** is a transient state (e.g., during context switches).

Implementation

- → **Non Work Conserving CPU scheduling**

Linux Complete Fair Scheduler

Implementation

- Non Work Conserving CPU scheduling

Linux Complete Fair Scheduler

Non Work Conserving Scheduler

Implementation

→ **Non Work Conserving CPU scheduling**

Linux Complete Fair Scheduler

Non Work Conserving Scheduler

Three options : nice , cpulimit **,cgroups**

Implementation

- ➔ Non Work Conserving CPU scheduling
- ➔ **Frequency Preservation**

Implementation

- Non Work Conserving CPU scheduling
- Frequency Preservation
- **Suspension Management**

Implementation

- → Non Work Conserving CPU scheduling
- → Frequency Preservation
- → **Suspension Management**

/sys/power/wake_unlock

Implementation

- ➔ Non Work Conserving CPU scheduling
- ➔ Frequency Preservation
- ➔ **Suspension Management**

/sys/power/wake_lock

/sys/power/wake_unlock.

Implementation

- ➔ Non Work Conserving CPU scheduling
- ➔ Frequency Preservation
- ➔ **Suspension Management**

/sys/power/wake_lock

Reject requests from best effort tasks for wake-lock

Implementation

- Non Work Conserving CPU scheduling
- Frequency Preservation
- Suspension Management
- **Contention Triggered Throttling**

Implementation

➔ **Contention Triggered Throttling**

performance monitoring unit:

ARMV7 A15 PERFCTR L2 CACHE REFILL READ

ARMV7 A15 PERFCTR L2 CACHE REFILL WRITE as L2

Implementation

- ➔ **Contention Triggered Throttling**

performance monitoring unit:

ARMV7 A15 PERFCTR L2 CACHE REFILL READ

ARMV7 A15 PERFCTR L2 CACHE REFILL WRITE as L2

Update at 20 ms granularity

Implementation

- Non Work Conserving CPU scheduling -> C State Preservation
- Frequency Preservation -> P State Preservation
- Suspension Management -> S state Preservation
- Contention Triggered Throttling -> Contention Mitigation

Experimental Setup

Device: Huawei Mate7 (late 2014)

– 1.8 GHz ARM Cortex-A15 Quad Core

– 32KB/32KB L1 instruction and data cache

– 2MB L2 cache, 2GB RAM with 12.8 GBps

– Power measurement using Monsoon power meter with smartphone battery detached

Benchmarks

Interactive application:

- Bbench: load locally cached websites
- Angry bird: casual game

Best-effort tasks: Spin, Compression, Encryption,

- AppOpt, FaceAnalysis

Test Flow

- Use automate UI testing tools (RERAN[1]) to minimize variations
- Launch two applications roughly at the same time
- Configure the workload such that application executions mostly overlap

$$\text{Discount } \sigma = 1 - \frac{E_{\text{co-run}} - E_{\text{interactive\_alone}}}{E_{\text{best-effort\_alone}}}$$

Related Work

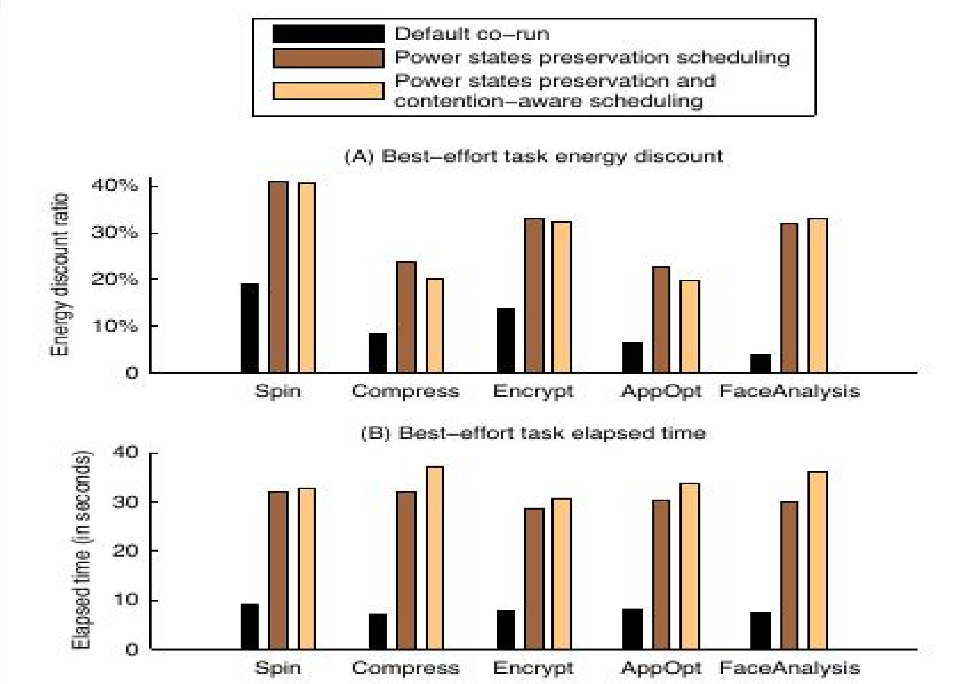
- CARROLL , A., AND HEISER, G. An analysis of power consumption in a smartphone. In Proc. of the USENIX 2010.
- C ARROLL , A., AND H EISER , G. Mobile multicores: use them or waste them. In Proc. of the Workshop on Power-Aware Computing and System (HotPower) (Nov. 2013).
- SONG , W., S UNG , N., C HUN , B.-G., AND KIM , J. Reducing energy consumption of smartphones using user-perceived response time analysis. Santa Bar-bara, CA, Feb. 2014

Test Flow

- Use automate UI testing tools (RERAN[1]) to minimize variations
- Launch two applications roughly at the same time
- Configure the workload such that application executions mostly overlap

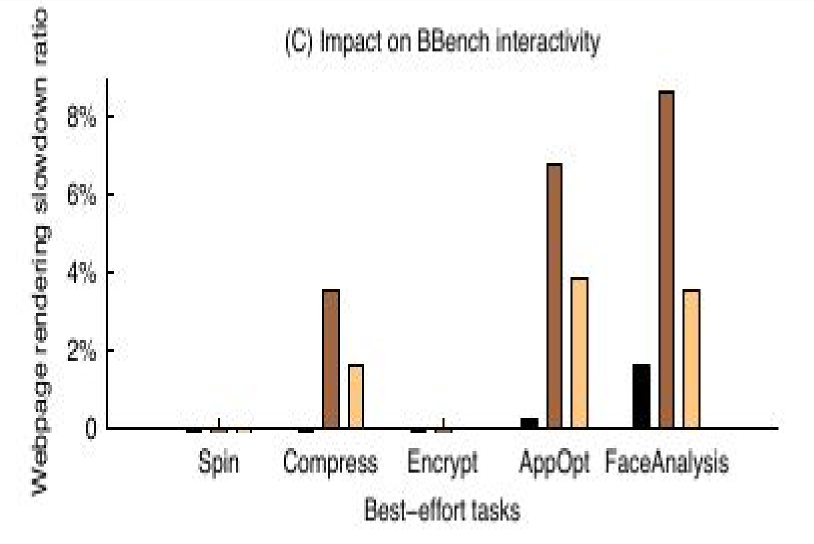
$$\text{Discount } \sigma = 1 - \frac{E_{\text{co-run}} - E_{\text{interactive\_alone}}}{E_{\text{best-effort\_alone}}}$$

BBench

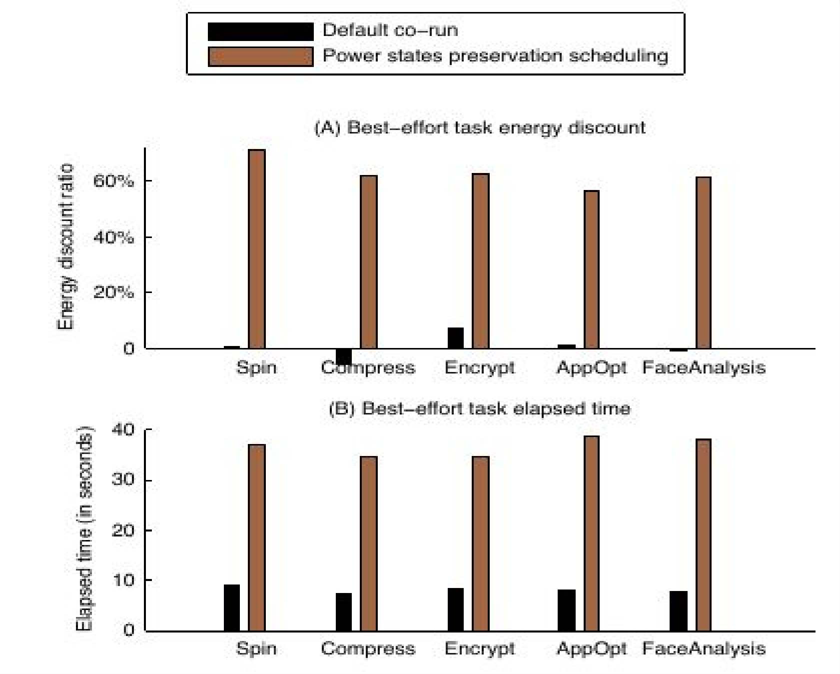


Default co–run
Power states preservation scheduling
Power states preservation and contention–aware scheduling
(A) Best–effort task energy discount
Energy discount ratio
40%
30%
20%
10%
0
Spin
Compress
Encrypt
AppOpt
FaceAnalysis
(B) Best–effort task elapsed time
Elapsed time (in seconds)
40
30
20
10
0
Spin
Compress
Encrypt
AppOpt
FaceAnalysis

BBench

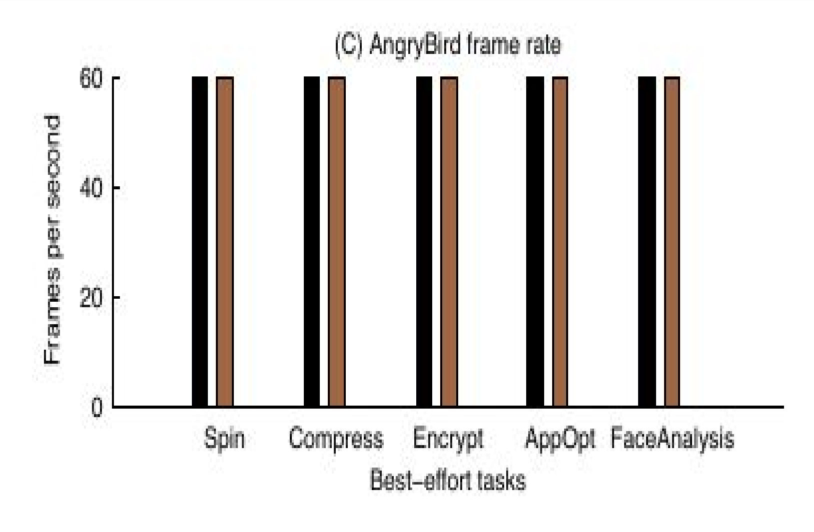

(C) Impact on BBench interactivity
Webpage rendering slowdown ratio
8%
6%
4%
2%
0
Spin
Compress
Encrypt
AppOpt
FaceAnalysis
Best-effort tasks

AngryBird



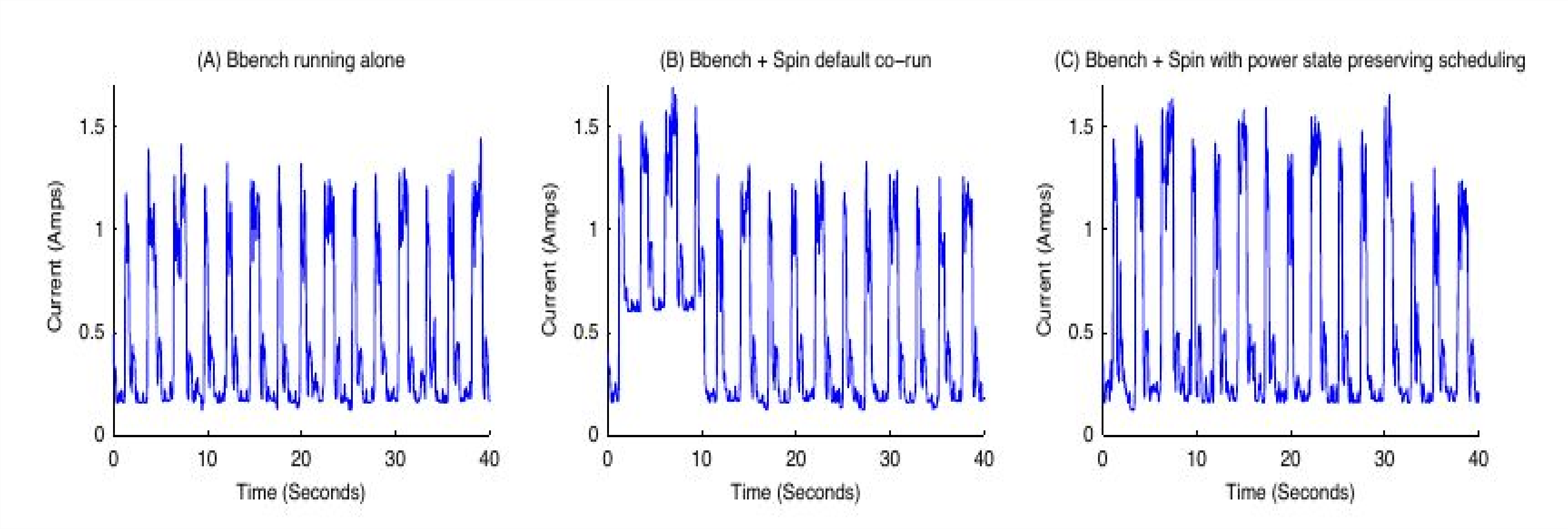
Default co–run
Power states preservation scheduling
(A) Best–effort task energy discount
Energy discount ratio
60%
40%
20%
0
Spin
Compress
Encrypt
AppOpt
FaceAnalysis
(B) Best–effort task elapsed time
Elapsed time (in seconds)
40
30
20
10
0
Spin
Compress
Encrypt
AppOpt
FaceAnalysis

AngryBird



(C) AngryBird frame rate
Frames per second
0
20
40
60
Spin
Compress
Encrypt
AppOpt
FaceAnalysis
Best–effort tasks

BBench



(A) Bbench running alone
(B) Bbench + Spin default co−run
(C) Bbench + Spin with power state preserving scheduling
Current (Amps)
Time (Seconds)
0
0.5
1
1.5
0
10
20
30
40

Bbench

(A) Bbench + FaceAnalysis

Throttling threshold (misses/μSecs)	Best-effort task elapsed time	Bbench slowdown ratio
7.5	42.65 Secs	3.94 %
10.0	36.25 Secs	3.57 %
12.5	32.42 Secs	6.58 %
15.0	30.20 Secs	8.73 %

(B) Bbench + AppOpt

Throttling threshold (misses/μSecs)	Best-effort task elapsed time	BBench slowdown ratio
6.0	43.10 Secs	4.20 %
7.5	36.85 Secs	3.72 %
10.0	33.87 Secs	3.88 %
12.5	29.62 Secs	5.80 %
15.0	29.77 Secs	6.81 %

Category	Abundance of discounted CPU cycles (multicore)	Abundance of discounted CPU cycles (single-core)	Equivalent work of FaceAnalysis (frames of faces can be analyzed)	Equivalent work of Encryption (minutes of video can be encrypted)
Web Browsing	1.63	0.66	30	21
Video Streaming	2.41	0.85	4	3
Gaming	1.61	0.65	21	15
Navigation	2.42	0.85	13	9
Messaging	2.88	0.97	3	2
Social Network	1.88	0.72	12	9
Camera	2.10	0.77	5	4
Music Streaming	1.63	0.66	7	5

Table 4: Results for the trace-based application study. Each usage scenario lasts for one minute. Abundance of

Related Work

- CARROLL , A., AND HEISER, G. An analysis of power consumption in a smartphone. In Proc. of the USENIX 2010.
    - The same paper as Prasanth and Aaskash discussed.
- CARROLL , A., AND HEISER , G. Mobile multicores: use them or waste them. In Proc. of the Workshop on Power-Aware Computing and System (HotPower) (Nov. 2013).
    - That a core should be kept online as long as there is work

Related Work

- SONG , W., SUNG , N., CHUN , B.-G., AND KIM , J. Reducing energy consumption of smartphones using user-perceived response time analysis. Santa Bar-bara, CA, Feb. 2014
    - Decreases the frequency when user facing(display on) tasks are completed.
- M ARTINS , M., C APPOS , J., AND F ONSECA, R.Selectively taming background android apps to improve battery lifetime. In USENIX ATC 15
    - monitor and intercept smartphone background activities while the system is in suspension state to extend the battery life.