Trading Performance and Responsiveness for Energy on Android Devices

Technical Report

Ahmed Hussein

Advised by Antony L. Hosking & Mathias Payer

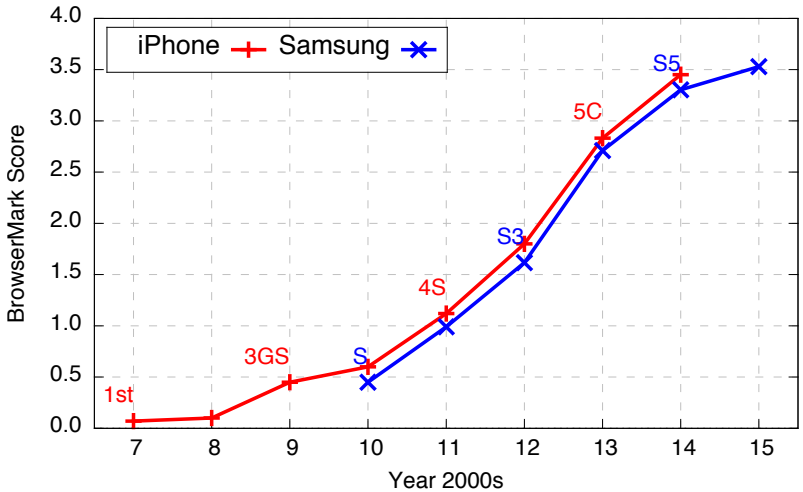
Department of Computer Science
Purdue University

May 2015

Outline

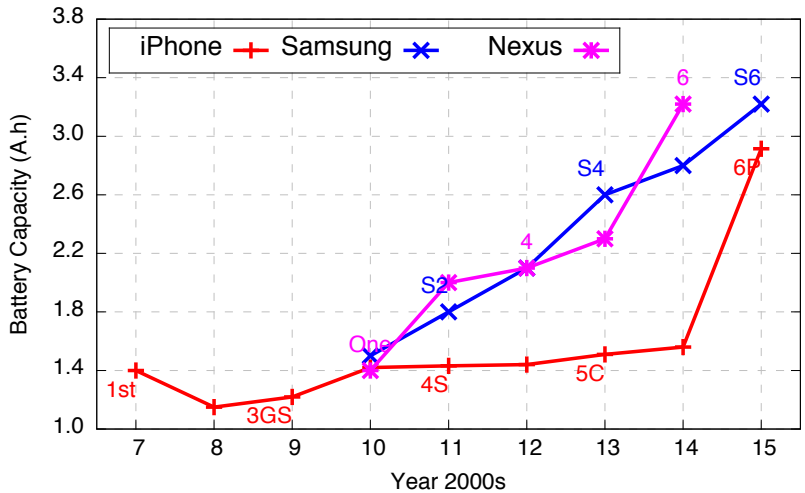
Mobile Devices
Performance & Battery Lifetime

CPU Performance Increase



Source http://browsermark.rightware.com/

Battery Curve



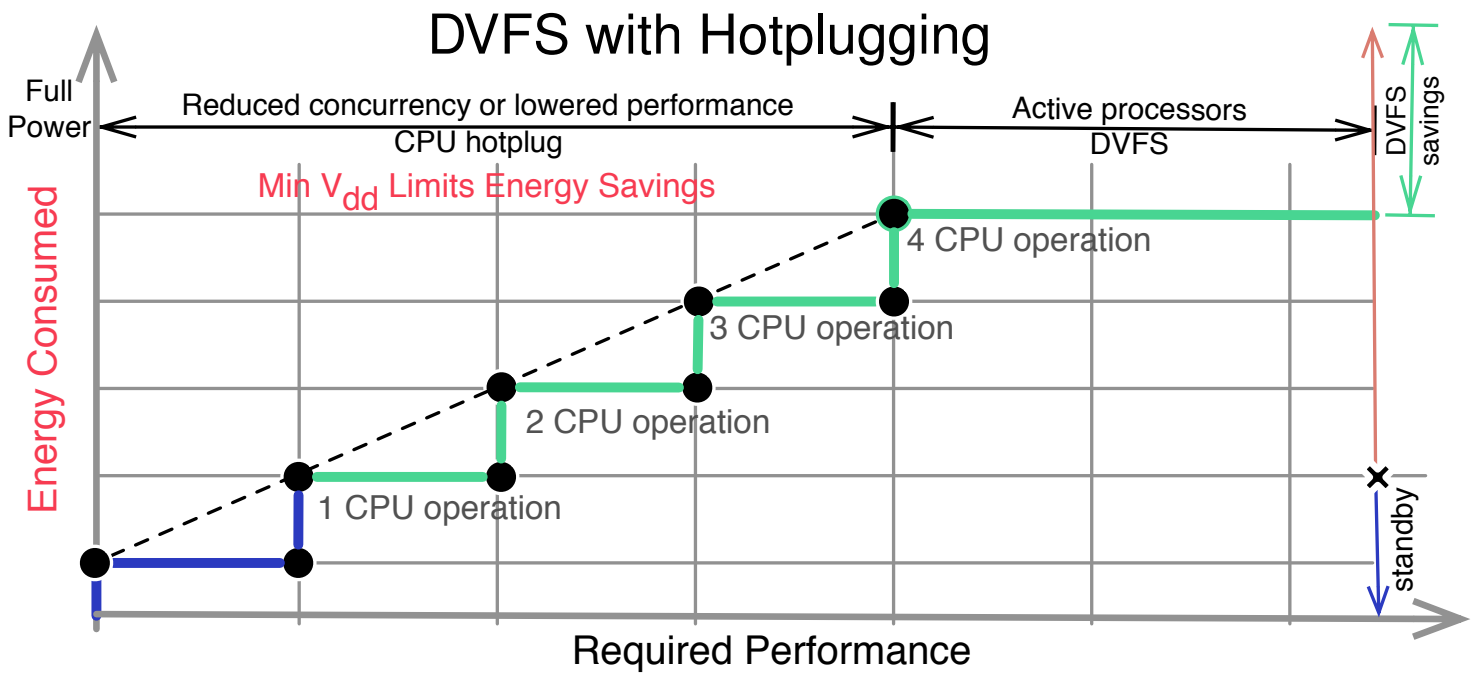
Source: Wikipedia http://en.wikipedia.org/wiki/IPhone

Buying a phone based on a CPU benchmark performance is like buying a car based on what kind of tyres it has. Tim McDonough

[Review14]

Performance & Power Constraints

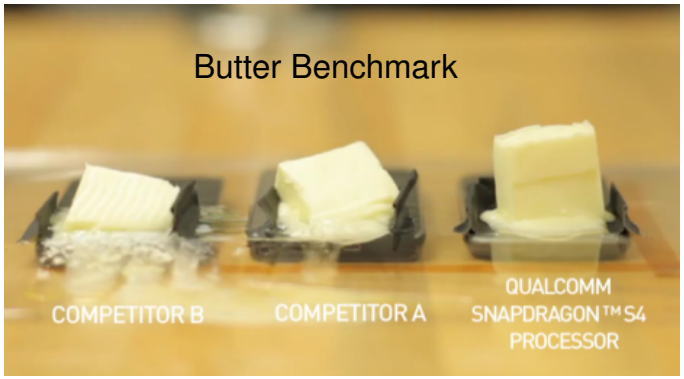
DVFS + Hotplugging



Power & Operational Constraints

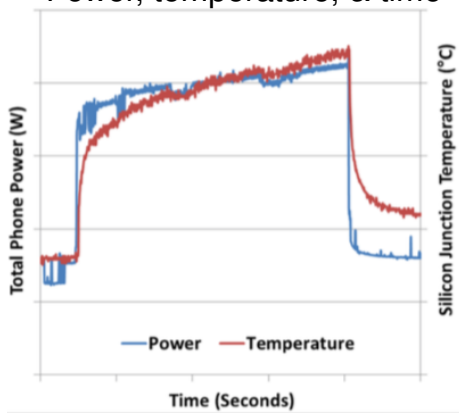
Temperature

- +ve feedback loop between power and temperature.
- Leakage power increases exponentially with temperature.
- Thermal limits are constant (Skin 40–45 °C).
- No Heat Sinks.
- Power management needs to be made temperature-aware.

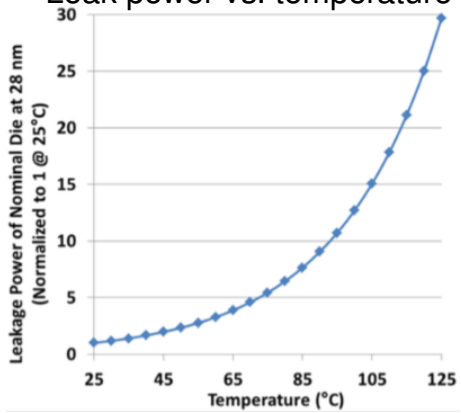


[QComm Butter(2012)]

Power, temperature, & time



Leak power vs. temperature



Statement

Mobile devices that rely on a managed run-time system pay a significant energy overhead for GC. Thus, tuning the GC implementation ameliorates the total device energy consumption with a minimal impact on throughput and responsiveness.

Tangible Conclusions

- Varying GC strategy can reduce on-chip energy by 20-30%
- GC cost function is related to DVFS
- Integration between GC and power mechanisms is effective

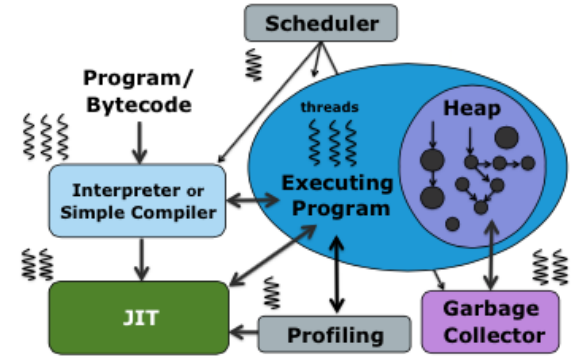
Research Roadmap

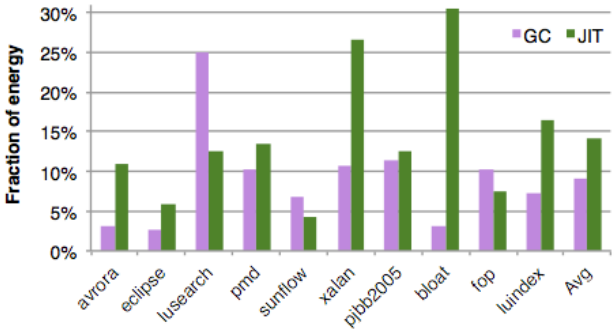
1. Challenges
2. Methodology to evaluate VM services
3. Significance of the GC
4. Impact of GC implementation on a device

Software Evaluation

We Have Been There Before!

- Refined methodology
  - Standard benchmarking.
  - Powerful profiling infrastructures.
- Evaluation scope
  - Tradeoffs are less understood.
  - Research gap between different layers [Kambadur&Kim(2014)].



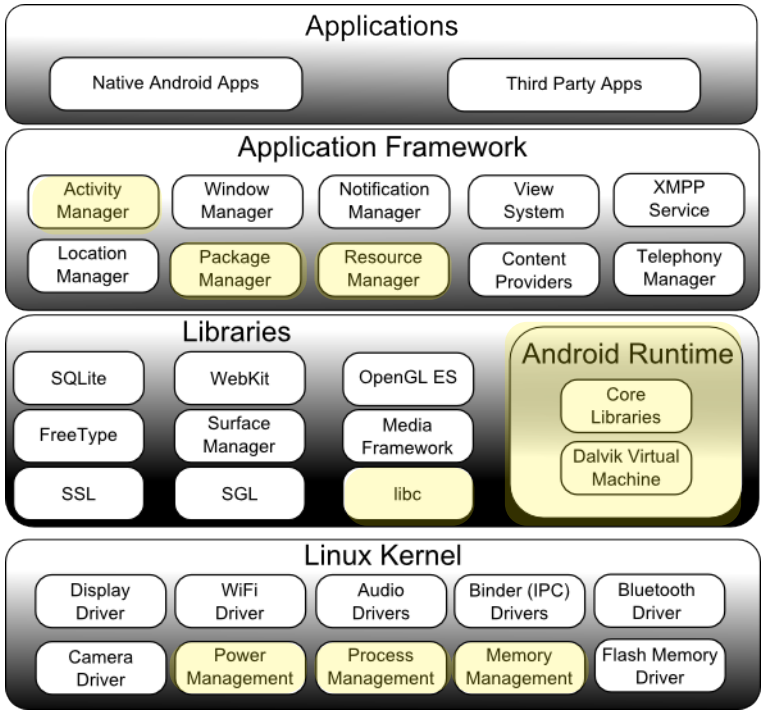


[Cao et al.(2012)]

System Complexity

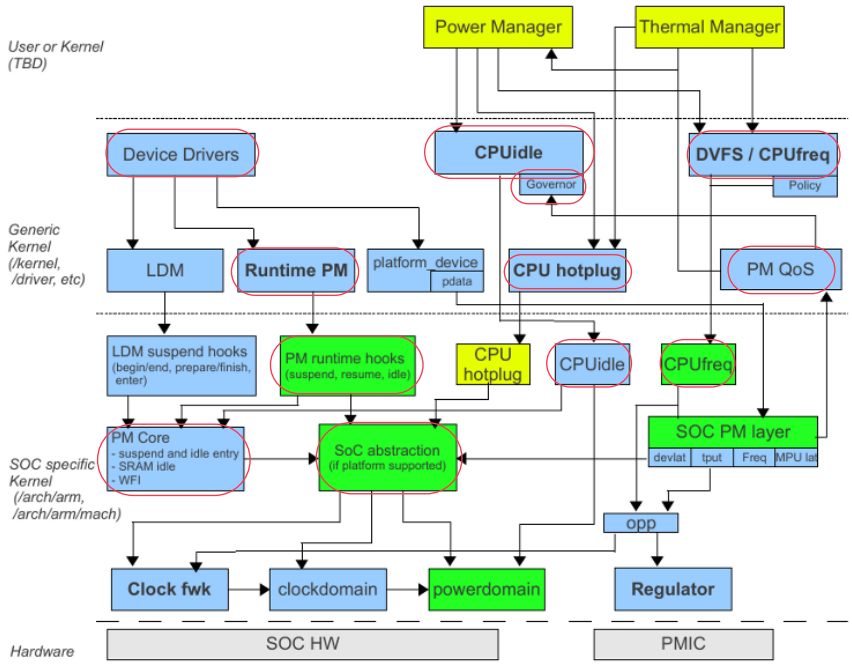
SoC Managed by Stacked Software Layers

Android Architecture



Credit: http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture

Power Management



Credit: http://elinux.org/images/a/a1/Elc2011_kucheria.pdf

Benchmarking
Challenges

- Young platform: benchmarking has yet to emerge
- Adaptive behavior, different functionality
- Synthetic, and measures a single feature
- Blackboxes rely on I/O libraries, irrelevant to VM control

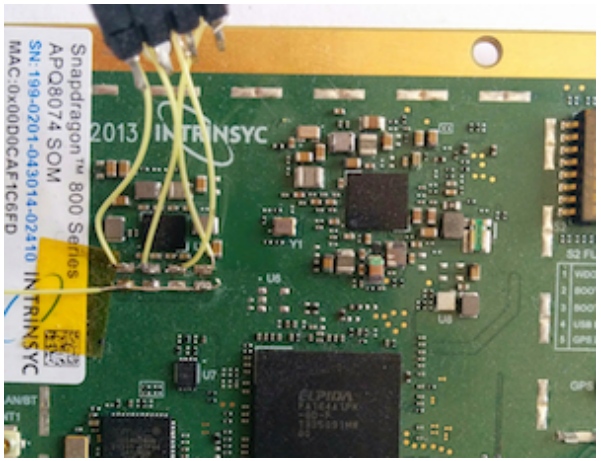
Benchmarking
Porting Java Applications

- Java ports:
  - DaCapo: 4 applications; xalan, lusearch, pmd, and luindex.
  - SPECjvm98: all 8 applications.
  - Small workload.
- Advantage:
  - Enables validation.
  - Enables correlation between VM and device performance.

Power Measurement
Approach

- Common alternatives:
  - Analytical models are restricted.
  - Power rails are not accessible.
- On-Chip Measurements:
  - Account for static & dynamic powers.
  - Isolate environment noise.
  - CPU is significant $\approx 20 - 40\%$

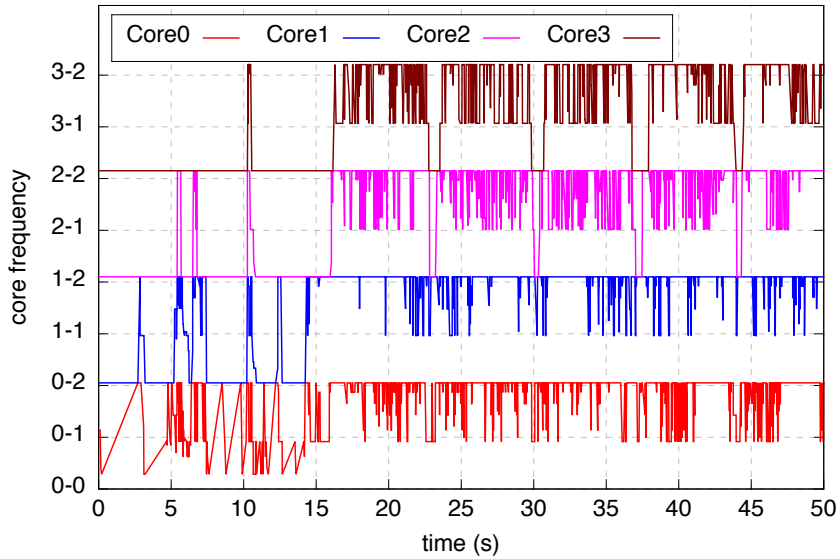
[Carroll&Heiser(2010)].



APQ8074 System-On-Module (SOM) modifications to measure power for the quad-Krait application processors

VM Profiling
Performance Counters & Memory Behavior

- Performance counters:
  - Memory: L1 access and miss.
  - CPU Cycles: Amount of work.
  - Instructions.
- Scheduling statistics: switching, migrations, delays, *etc.*.
- `systrace`: frequency, idle, workqueues [Android-Systrace].



lusearch – Frequency

VM Profiling
Responsiveness

- Mobiles are not real-time systems [RTDroid].
- Humans perceive pauses greater than 50ms [Efron(1973)].
- WCET is not adequate:
  - End-to-end execution.
  - Relation with power is less understood [Wilhelm et al.(2008)].
- Distribution of pauses: *min. mutator utilization* (MMU).
  - Three groups of pauses: *(i)* Safepoints, *(ii)* foreground GC, and *(iii)* waiting for GC.
  - MMU for a window of length w is the minimum $\frac{w-pauses}{w}$ (for all mutators) over all time slices of length w.

GC Cost Function
SYSTOR'15

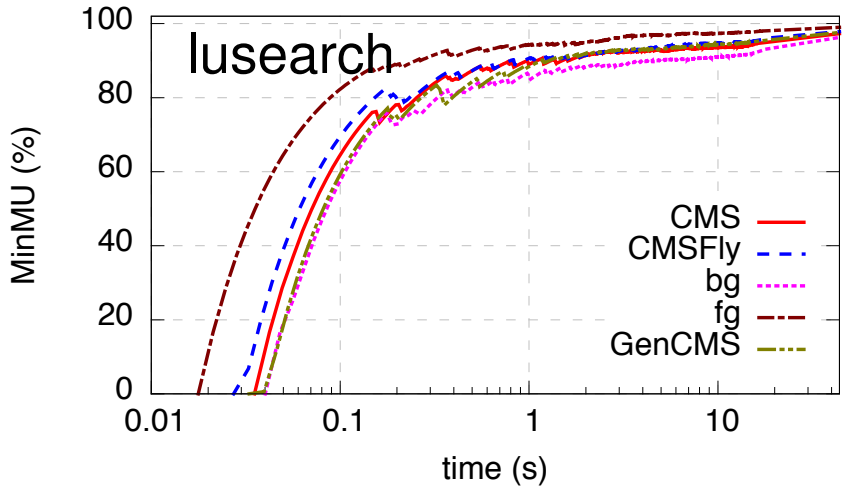
1. Study the design choices by comparison:
   - Android Dalvik has *concurrent mark-sweep*
   - Extend Dalvik's GC with *Generational* & *On-Fly*.
2. Analyze the degree of concurrency:
   - Background GC: mutators yield to GC daemon.
   - Foreground GC: disables GC daemon.
   - Set thread priorities.
3. Revisit GC configurations:
   - Change heap growth policies.

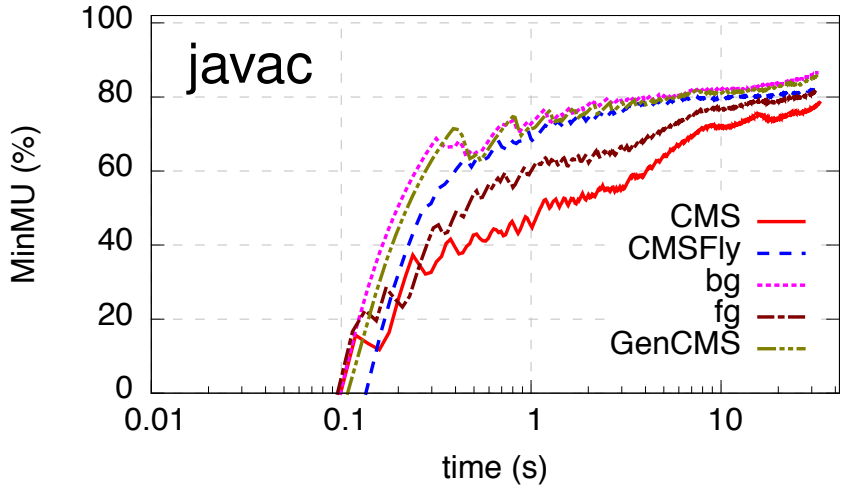
GC Implementations

Impact on Responsiveness and Power

- Background GC lifts MMU for large heaps.
- Foreground has better MMU because of priorities.
- Generational lifts the background performance.

- Background GC consumes more energy.
- Foreground saves energy.

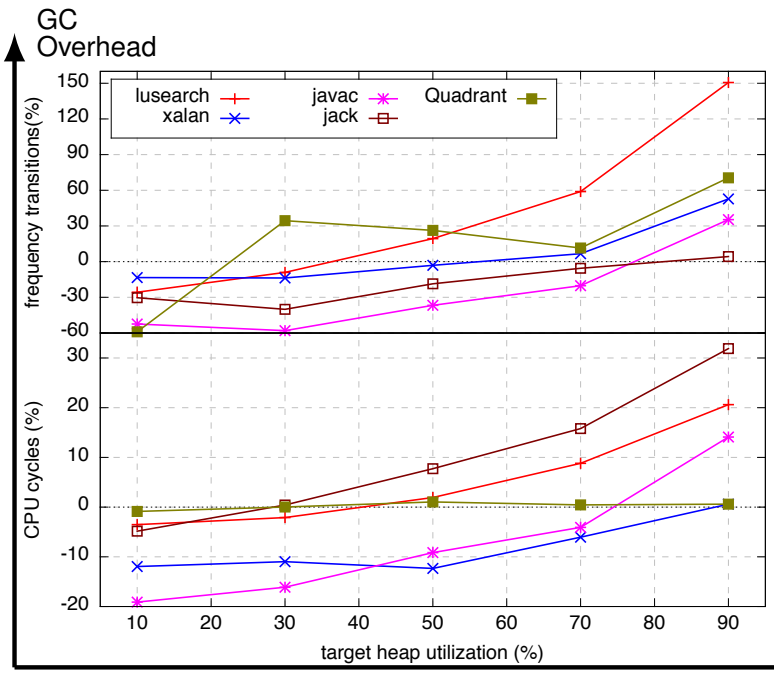




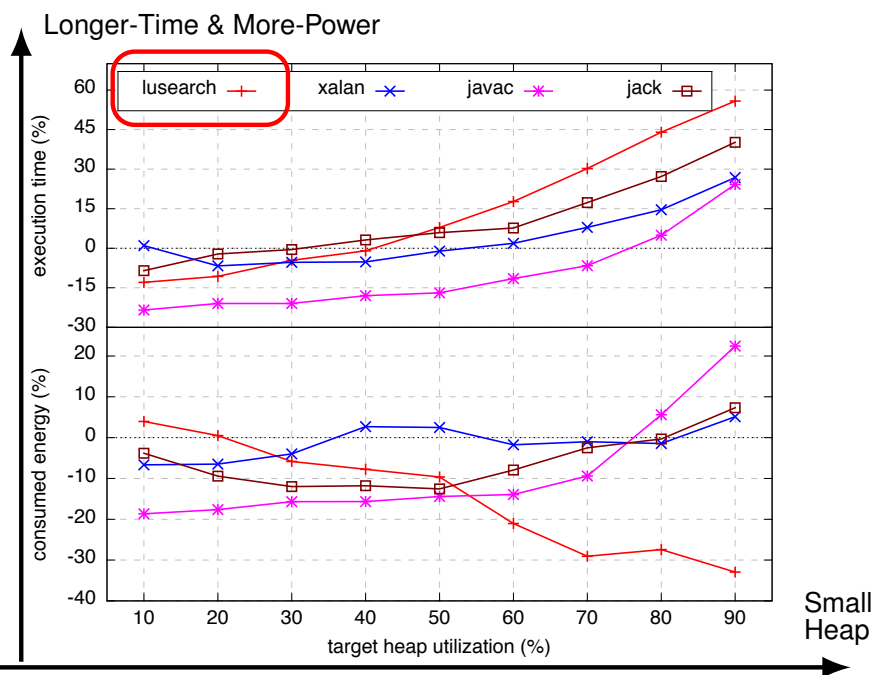
GC Parameters Tuning

Heap Size

- App workload increases with tighter heaps.
- Smaller heaps imply more frequency transitions.



Effect of `targetutil` on CPU cycles (bottom) & frequency transitions (top) normalized to default CMS



Effect of `targetutil` on energy (bottom) & throughput (top) normalized to default CMS

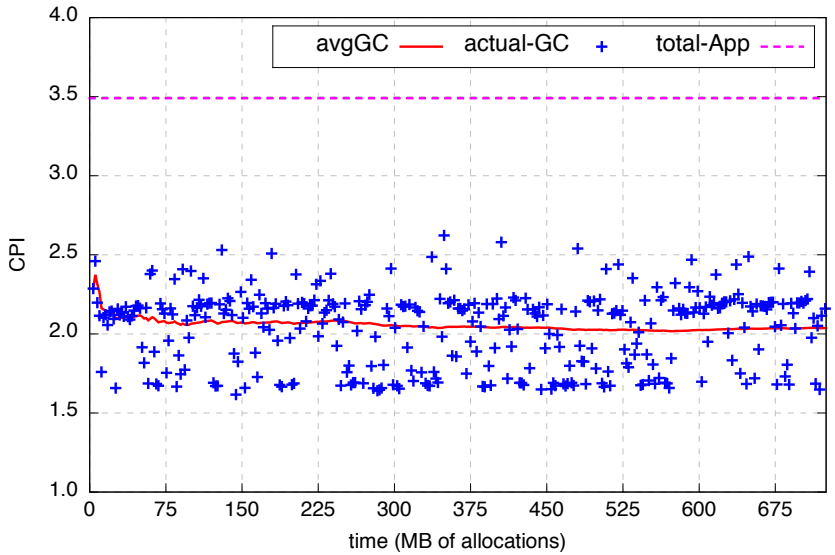
GC Impact on DVFS, ISMM'15

Issues with DVFS

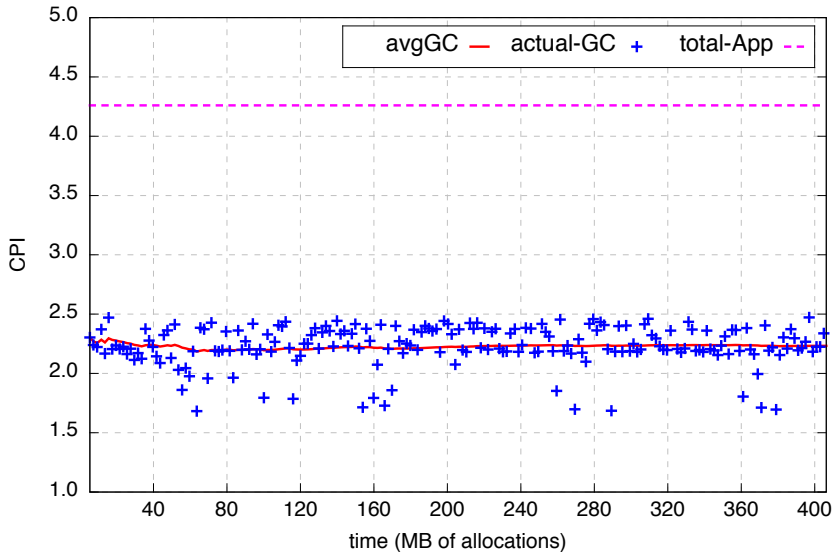
- Increased throughput $\neq$ better energy consumption.
- GC events have a significant impact on DVFS decisions.
- DVFS adopts "*race-to-idle*".
- DVFS has latency (too late).
- DVFS cannot detect impulses.

Exploring Phase Behavior
CPI

- Program's execution changes over time in phases.
- GC has a lower CPI compared to the average mutator workload.



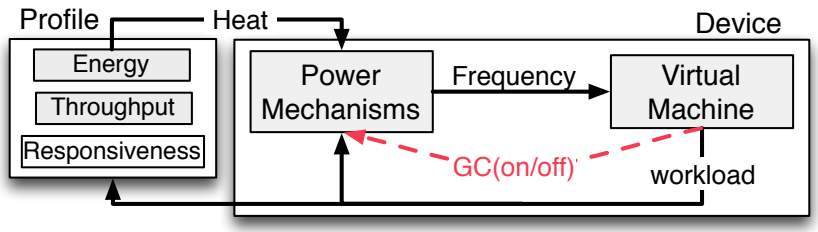
lusearch CPI (GC daemon vs. total app)



xalan CPI (GC daemon vs. total app)

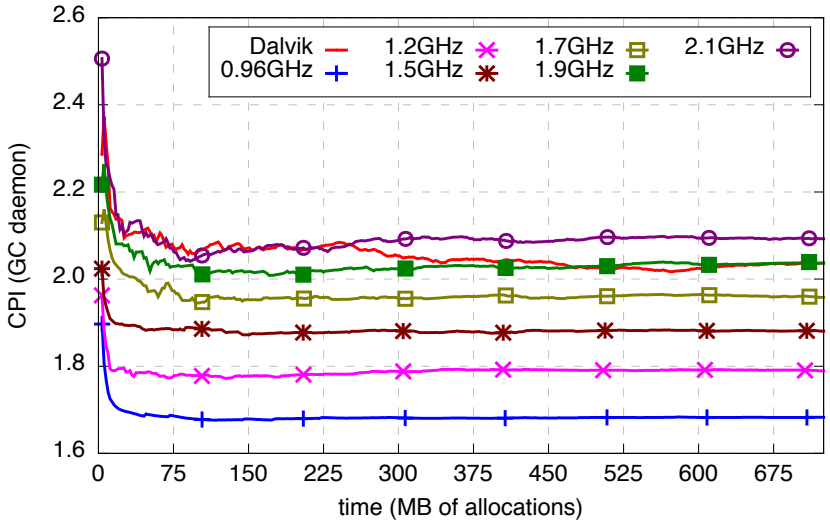
Integrating GC with Power
Throttling

- Reduce wasted cycles during *concurrent GC*:
  - Pin GC daemon to core-0.
  - Cap the maximum frequency of core-0.
- Explore the tradeoff using different frequencies:
  - Vary the maximum core speed.
  - Study the throughput and responsiveness tradeoffs.

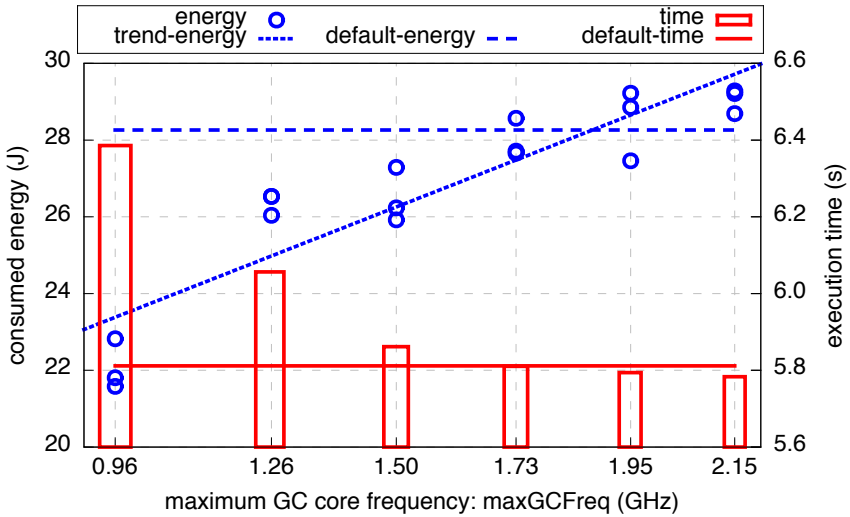


Block diagram of the modified architecture

Reduce the CPI Through DVFS



Cumulative average GC daemon CPI



Total consumed energy and execution time

- The lower the capping frequency, the lower the CPI
- 30% energy reduction at 0.96GHz capping frequency
- 20% time overhead at worst-case

Introducing Android RunTime (ART)

New Android VM

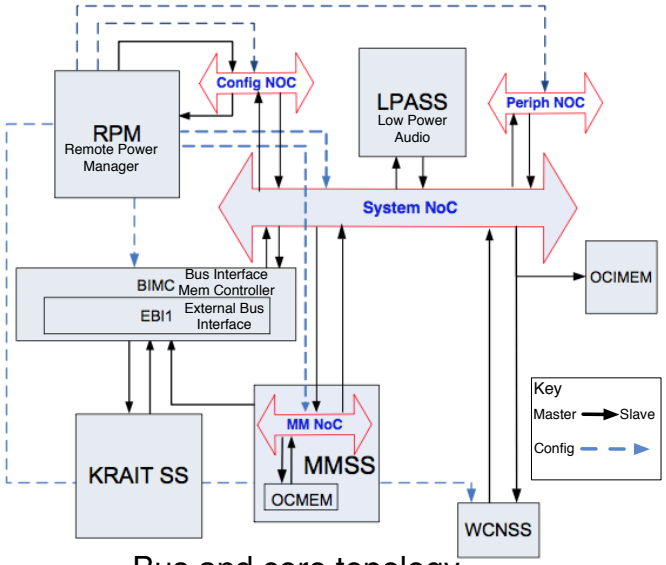
- Ahead-of-time Compiler for Android 4.4
- ART implements several of our GC improvements:
  - One major pause time instead of two
  - *Sticky* collector to deal with short lived objects
  - A separate heap for large objects
  - Parallel processing during the pauses
- Exploring ART's behavior is necessary:
  - Relevance to real devices
  - New VM produces new profiles

Conclusions
What to take?

- GC is significant for energy, responsiveness, and throughput.
- Concurrent GC has different workloads compared to App mutators.
- GC benefits from direct integrations with Power Managers.
- GC-aware governors outperform GC parameter tunings.
- It is necessary to evaluate Mobile systems through non-adjacent layers.
- Mobile platform needs standard benchmarkings and methodology.

Direct Integration with Power
Bus Speed & Memory DVFS

- Power Savings in DRAM:
  - Self Refresh
  - Sleep States
- Memory performance at reduced CPUfreq depends on architecture [Schone et al.(2012)].
- Memory Bandwidth varies with benchmark phases.
- Portion of cache-residency.
- RPM chooses a performance level that satisfies all outstanding requests for a bus.



Bus and core topology


A. Hussein, A. L. Hosking, M. Payer, and C. A. Vick.
Don't race the memory bus: Taming the gc leadfoot.
To appear in proceedings of the International Symposium on Memory Management, Portland, OR, June 2015.


A. Hussein, M. Payer, A. L. Hosking, and C. A. Vick.
Impact of GC design on power and performance for Android.
To appear in proceedings of the International Systems and Storage Conference, Haifa, Israel, May 2015.


A. Hussein.
On tracing the memory behavior of Dalvik applications.
Master's thesis, Purdue University, 2013.


A. Carroll and G. Heiser.
An analysis of power consumption in a smartphone.
In *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, pages 21–21, Berkeley, CA, USA, 2010. USENIX Association.
URL http://dl.acm.org/citation.cfm?id=1855840.1855861.


T. Cao, S. M. Blackburn, T. Gao, and K. S. McKinley.
The yin and yang of power and performance for asymmetric hardware and managed software.
In *International Symposium on Computer Architecture*, pages 225–236, Portland, Oregon, June 2012.
doi: 10.1109/ISCA.2012.6237020.


M. Kambadur and M. A. Kim.
An experimental survey of energy management across the stack.
In *ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, pages 329–344, Portland, Oregon, Oct. 2014.
doi: 10.1145/2660193.2660196.


R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, and P. Stenström.
The worst-case execution-time problem — overview of methods and survey of tools.
ACM Transactions on Embedded Computing Systems, 7(3):36:1–36:53, May 2008.
doi: 10.1145/1347375.1347389.


R. Efron.
Conservation of temporal information by perceptual systems.
Perception & Psychophysics, 14(3):518–530, Oct. 1973.
doi: 10.3758/BF03211193.


R. Schone, D. Hackenberg and D. Molka.
Memory Performance at Reduced CPU Clock Speeds: An Analysis of Current x86_64 Processors.
Presented as part of the 2012 Workshop on Power–Aware Computing and Systems, May 2012.
URL https://www.usenix.org/conference/hotpower12/workshop-program/presentation/Schöne.


Qualcomm, Inc.
Snapdragon S4 Thermal Comparison and Butter Benchmark.
2012
URL https://youtu.be/zPGVGsQ7LrM.


Google.
Systrace.
2015
URL https://developer.android.com/tools/help/systrace.html.


TrustedReviews.
Qualcomm: Don't buy a smartphone because of a CPU benchmark
Tim McDonough is the Head of marketing and the transformational technology marketer at Qualcomm, Inc.
Jan 2014.
URL http://www.trustedreviews.com/news/qualcomm-don-t-by-a-smartphone-because-of-its-cpu.


RTDroid.
Real-time Android varriant powered by Fiji VM
2015
URL http://rtdroid.cse.buffalo.edu/.