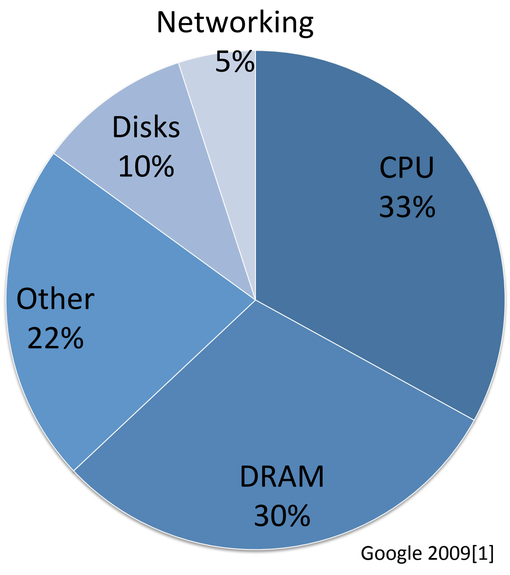
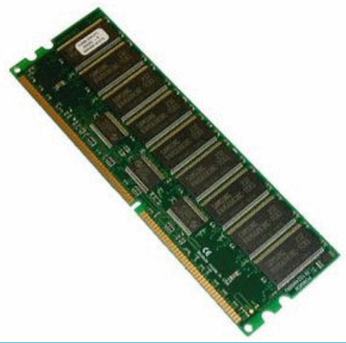
Mercurial Caches: Operating System Support for Energy Proportional DRAM

Asim Kadav, Rathijit Sen and Michael M. Swift, University of Wisconsin-Madison

DRAMs contribute significantly to system power



Google 2009[1]



Servers are provisioned with 10s of GBS of memory and consume 30-57% of total power for a system provisioned with 128 GB DDR3 memory[2]

How can we make DRAM power consumption energy proportional ?

Problem: DRAM is one of the significant consumer of power in modern systems. Unlike CPUs which provide voltage and frequency scaling techniques, DRAM techniques to save power upon partial memory use are limited.

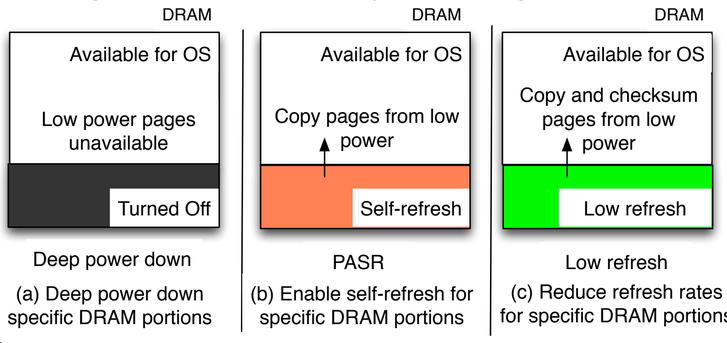
Goal: Provide abstractions in existing operating systems to utilize low power memory technologies using *mercurial caches*.

DRAM Power-saving Technologies

Low power mode in LPDDR/DDR3 DRAM systems. Many technologies in the figure co-exist. TCSR savings are for 40C drop for 64MB DIMM. Mercurial caches use the techniques enclosed in the box.

DRAM Technology	Data Retention	Granularity	Latency	Power Savings
ACPI S4	No	All DRAM	>1s	100%
Self-refresh (SR)	Yes	All DRAM	100ns	33%
Clock stop	Yes	All DRAM	200µs	83%
Temperature Controlled SR	Yes	All DRAM	100ns	60%
Partial Array Self Refresh (PASR)	Yes	1/16th DIMM	100ns	~25-30%
Lower refresh rates	Partial	Any	Clock	~42%
Deep power down (DPD)	No	Per module	200µs	95%

Using low power memory technologies inside OS



OS bottlenecks to adopting low power techniques

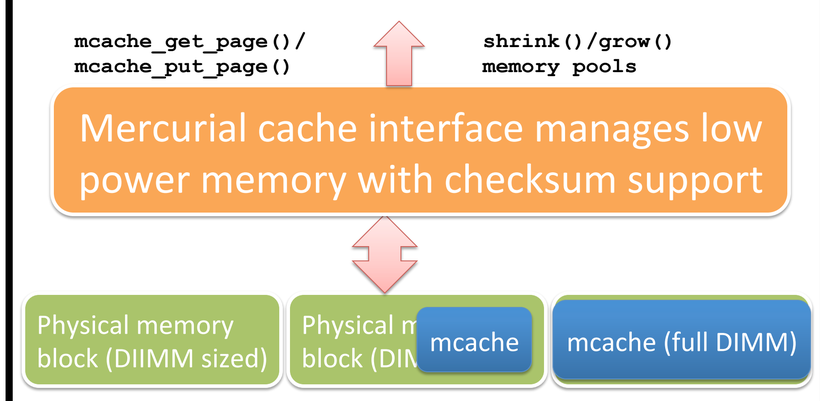
OS Behavior	Policy Rationale	Problem
Use all idle memory as page cache	OS uses all available memory to improve system performance	There is never any "free" space to be put to low power state
Fragments physical address space since address space is virtualized.	Contiguous memory requirement is limited (few MBs), keeping address space defragmented is only an overhead.	Memory needs to be put in low powered state in contiguous segments to save power

"Free memory is bad memory" philosophy and physical memory fragmentation hurt any memory consolidation for power savings

Mercurial Caches

Mercurial Caches provides OS abstractions to use low power DRAM. It occupies portions of DRAM and puts them in low power state (turned off or cache clean data). It uses software checksums to ensure correctness in the absence of reliable hardware. Its goal is to save power with little performance loss during low memory utilization

1. A cache interface to get/put 4K pages reliably



Low power mercurial cache pool per DIMM

2. OS modifications to make mercurial caches transparent to virtual memory subsystem:

Problem	Solution
How to dynamically grow and shrink mercurial caches w/o affecting running applications?	Predict the working set using free page information to quickly satisfy memory demands and slowly reclaim unused memory
How to ensure that mercurial caches do not appear as missing memory to VM system?	Account for mercurial caches when performing VM actions such as prefetching, allocations etc.
What data to store in mercurial caches if they are not turned off completely?	Register as slab cache and use them as a third level eviction cache for page cache

3. OS support to facilitate low power granularity allocations: Pre-reserve and migrate pages

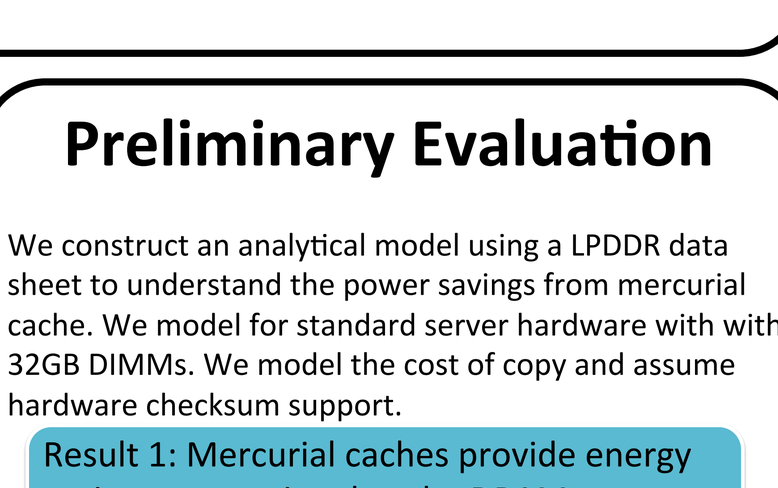
OS Behavior	Solution
Fragmentation of physical address space	Energy-aware migration of movable pages
Migration of pages not possible due to pinned pages	Mark specific segment boundaries dedicated to movable pages

Mercurial caches use reservation policies for movable data and migration mechanisms to switch contiguous memory into low power state

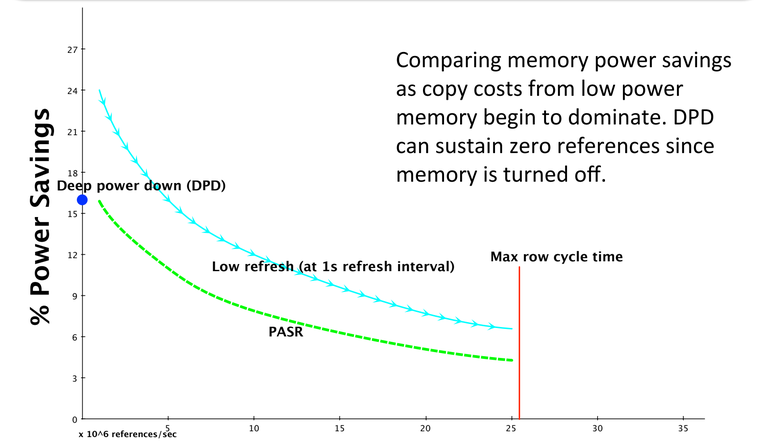
Preliminary Evaluation

We construct an analytical model using a LPDDR data sheet to understand the power savings from mercurial cache. We model for standard server hardware with with 32GB DIMMs. We model the cost of copy and assume hardware checksum support.

Result 1: Mercurial caches provide energy savings proportional to the DRAM usage, across different low power technologies



Result 2: Comparing power cost of copyout against savings from low power, we find mercurial caches can sustain a reference rate up to the row cycle time (~200K references)



Result 3: a) In experiments, completely turning off memory results in swapping and reduces performance by 5x as compared to mercurial caches b) It is also better than swapping out to a low power SSD, where the reference rate that can be sustained is low (upto 10K pages), due to access latency

References

[1] L.A. Barroso and U. Ho¨lzle. The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis Lectures on Computer Architecture*, 2009.

[2] D.H. Yoon, J. Chang, N. Muralimanohar, and P. Ranganathan. BOOM: Enabling mobile memory based low-power server DIMMs. In *ISCA*, June 2012.