Linux VM Infrastructure for memory power management

Ankita Garg
Vaidyanathan Srinivasan

IBM Linux Technology Center

Agenda - Saving Memory Power

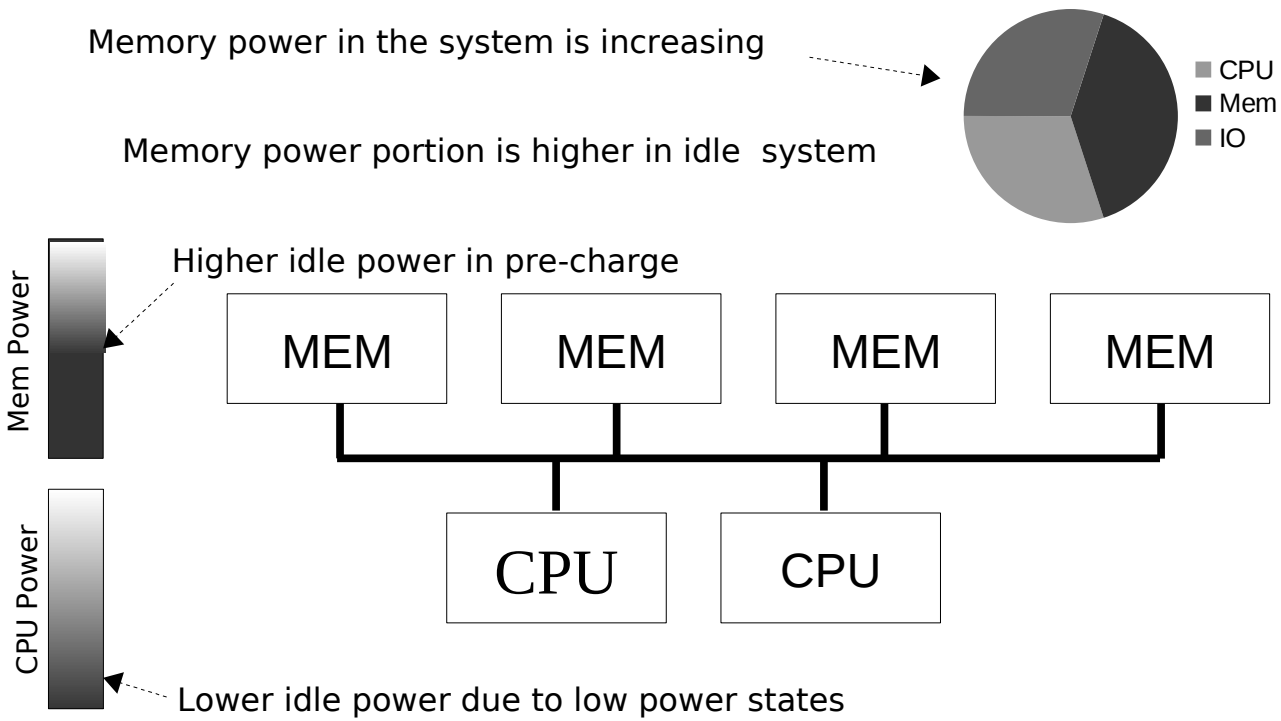
- Motivation - Why Save Memory Power
- Benefits
- How can it be achieved
- Role of Linux Virtual Memory Manager
- Implementation & challenges

Motivation - Why save memory power

- Density and capacity of memory has increased:
  - Leading to increased memory power consumption
  - Modern workload stack and virtualization demands more memory

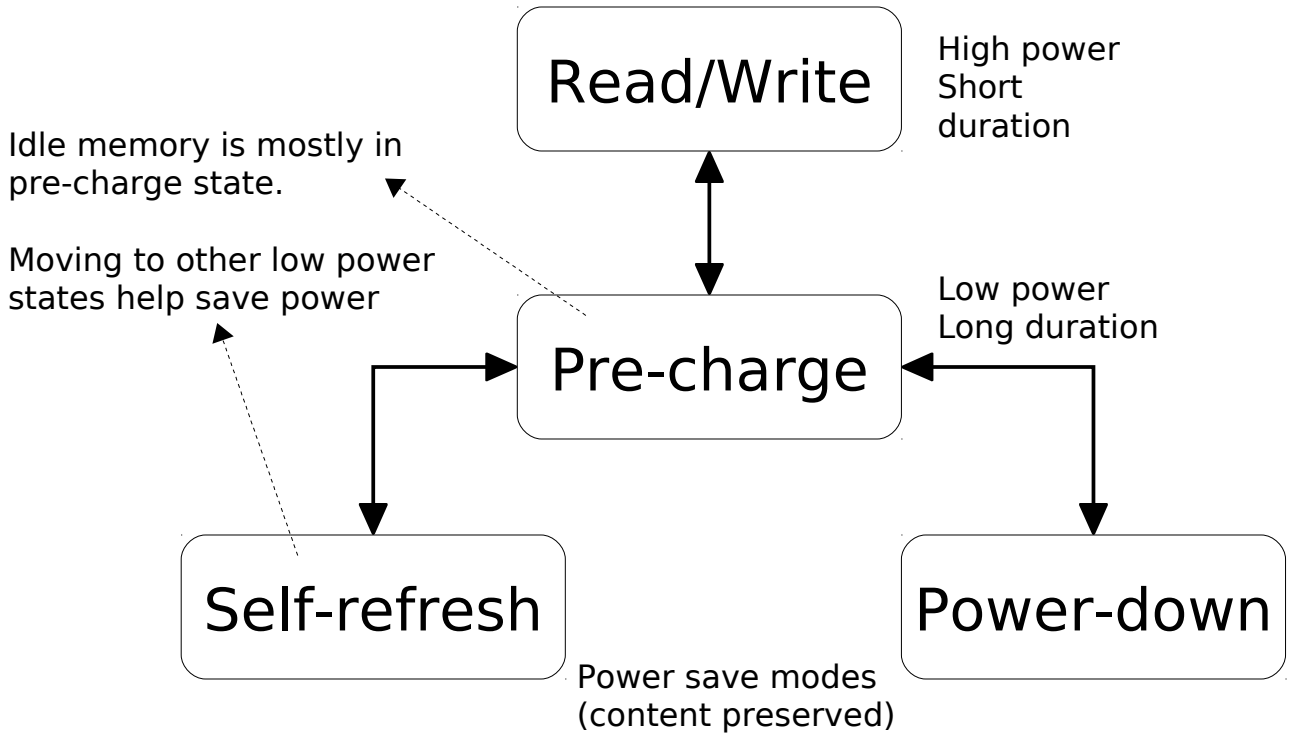
• Advances in OS cooperative runtime CPU power management has reduced cpu power consumption

•OS cooperative memory power savings can help reduce system level power consumption



Runtime low power modes in memory

- DDR3 memory allows different runtime low power states
  - Memory contents are preserved
  - Possible performance reduction

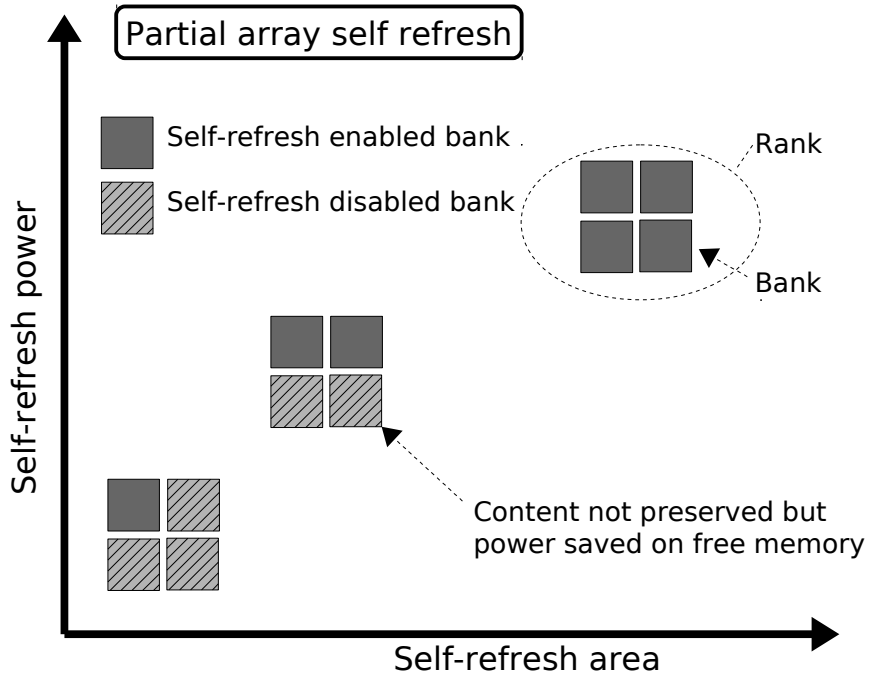


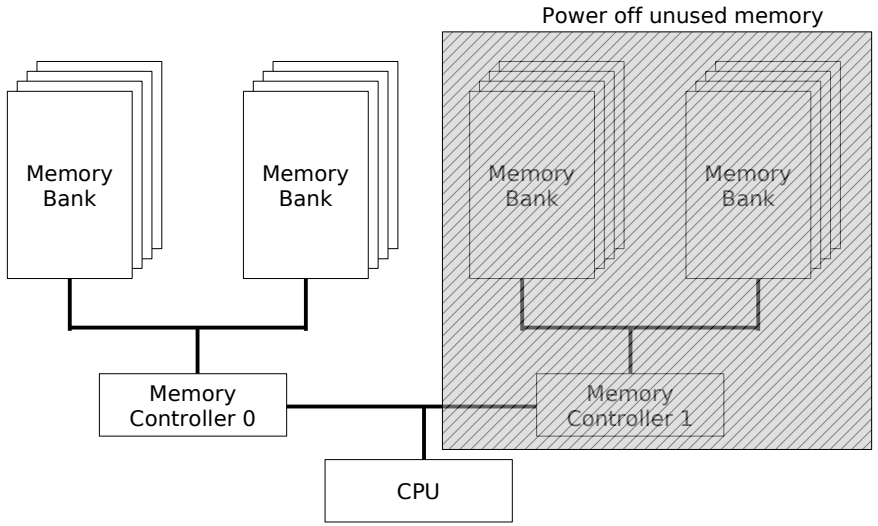
DDR3 Power States

Runtime partial power off

- *Partial Array Self Refresh is* a method to self refresh parts of system memory -
  - When system is active, all regions of memory are refreshed
  - When system is inactive (suspend state), memory is placed in self refresh to save content
  - Unimportant and free memory need not be self refreshed while in suspend mode, leading to longer battery life

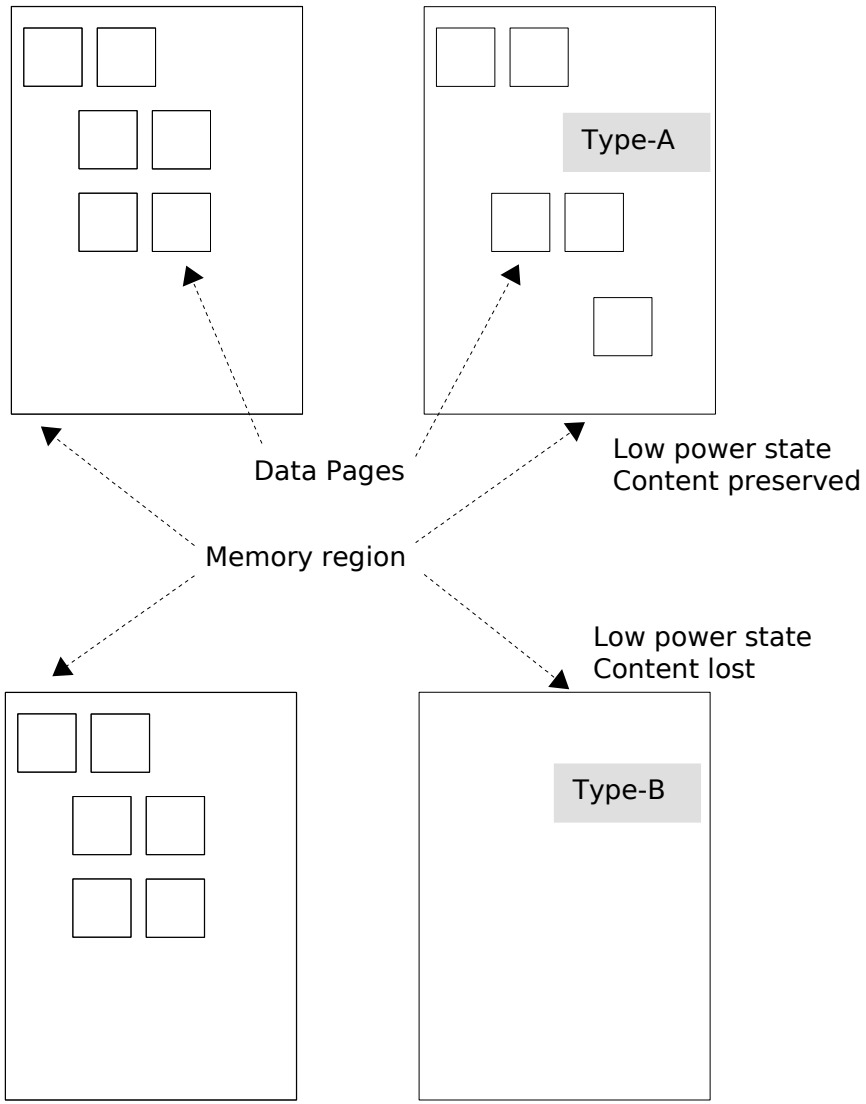
- Unused parts of memory hierarchy can be turned off (contents not preserved) even in an active system



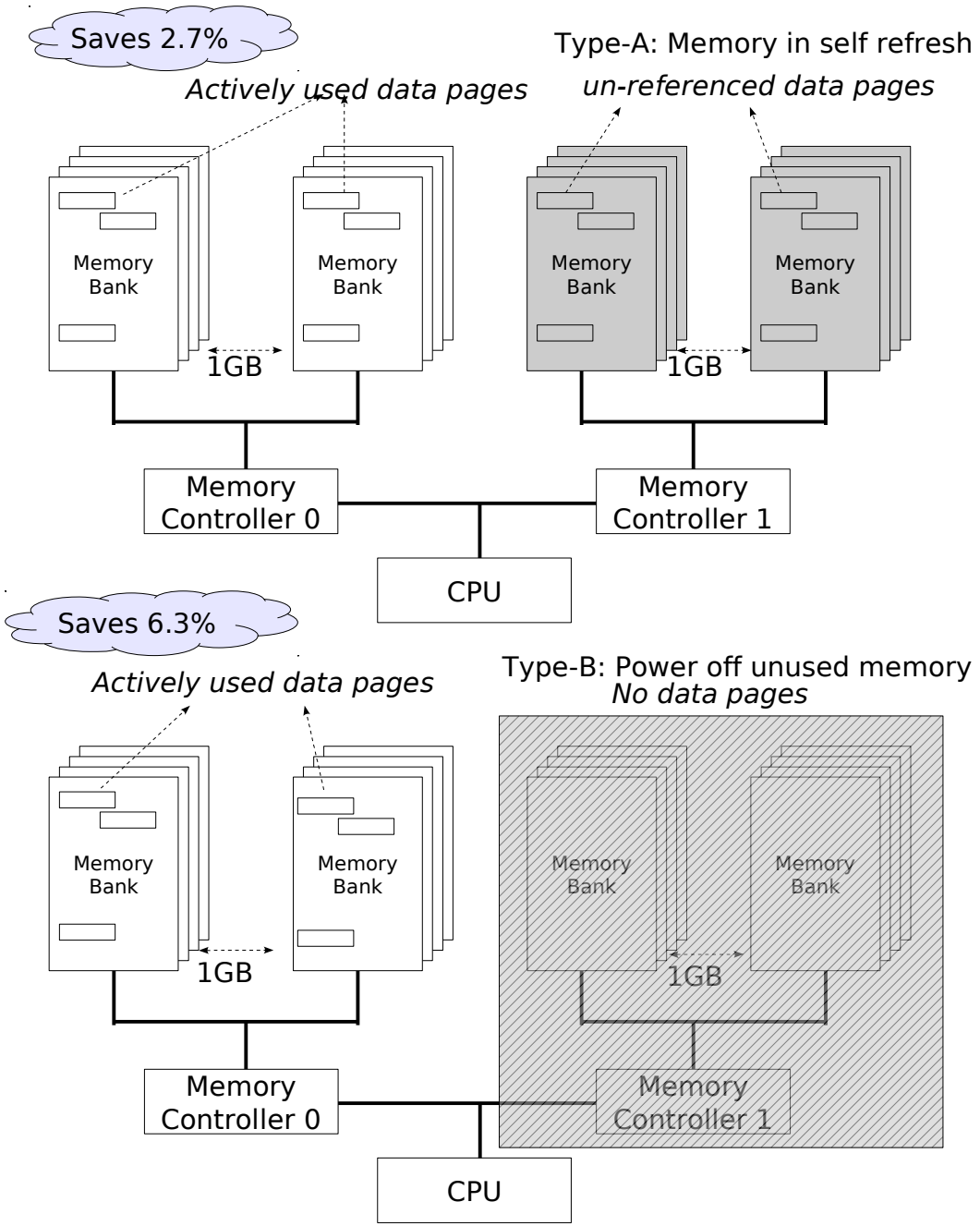


What this means to Linux OS

- Type-A (performance loss)
  - Less memory access consumes less power
    - Memory banks that are not accessed for some time (10s micro seconds) can be transitioned to low power state
    - Performance impact on first access to get the memory out to active state
  - Memory power savings happen at certain granularity
    - Consolidating access around this granularity saves power
- Type-B (content loss)
  - Parts of memory can be turned off losing contents
    - Chunks of free memory or unimportant data is aligned with the power off granularity, then save power during runtime as well as suspend
    - Granularity varies for each of the above feature



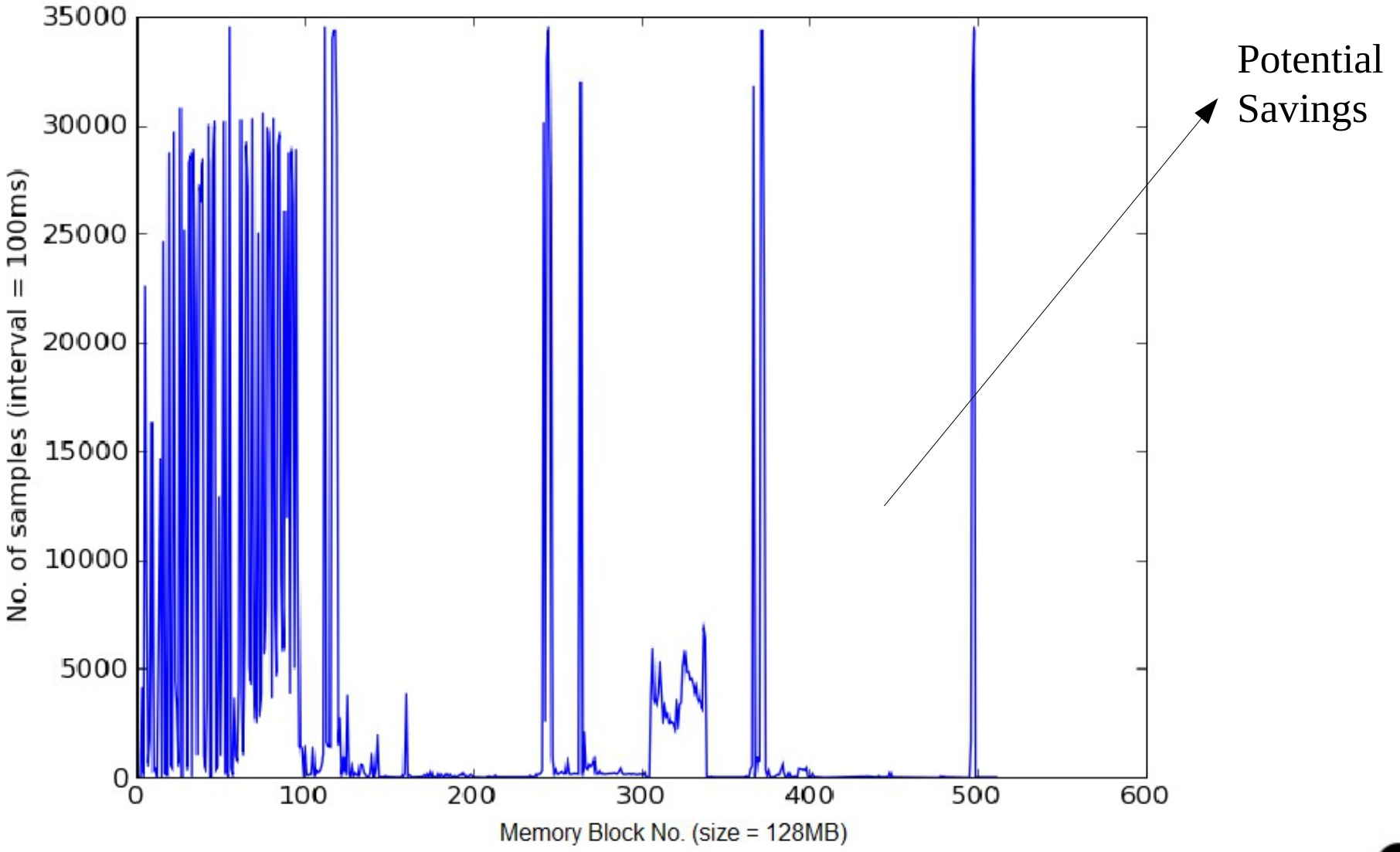
The benefits - Longer battery runtime



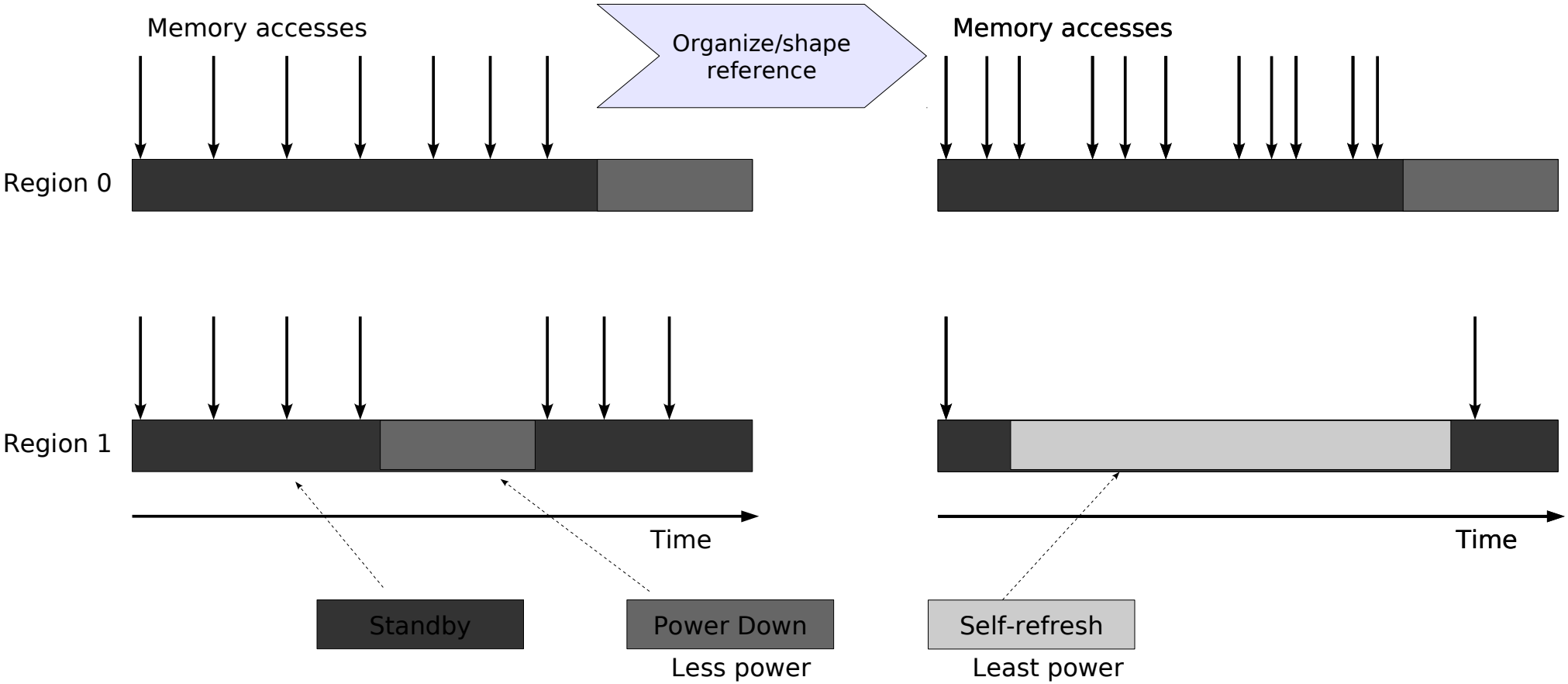
- 2-6% Power savings demonstrated in Samsung EXYNOS board with 2GB RAM
- Further cooperation with DEVFREQ can improve savings
  - Memory bandwidth control

Opportunity for Memory Power Savings

- Memory is over-provisioned
- There are blocks that are not referenced



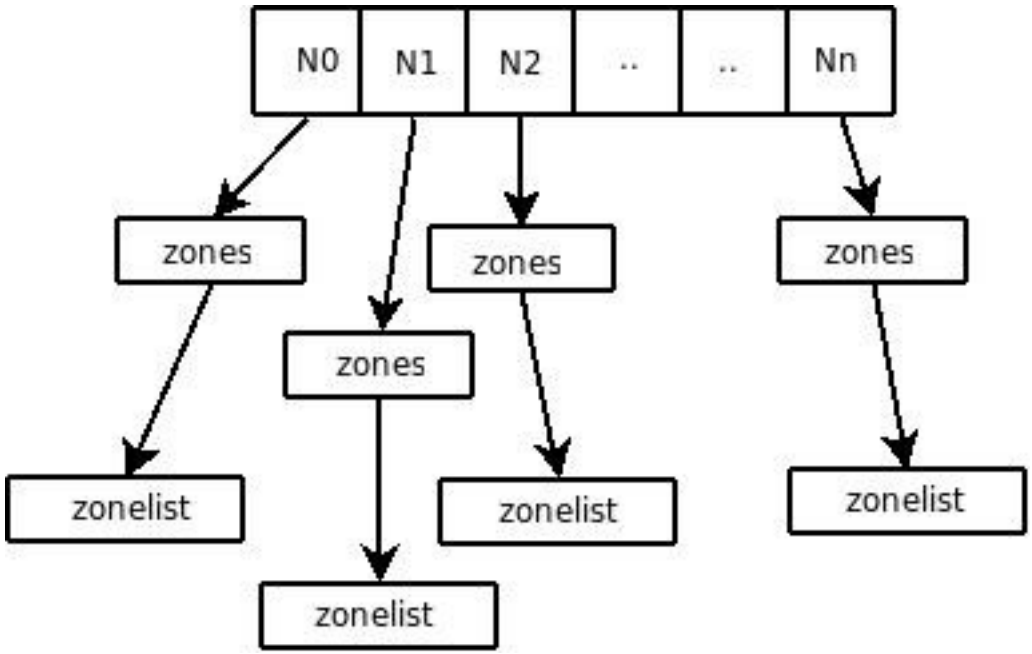
The Goal



- Consolidate memory reference for Type-A savings
- Consolidate allocations for Type-B savings

Role of Virtual Memory Manager

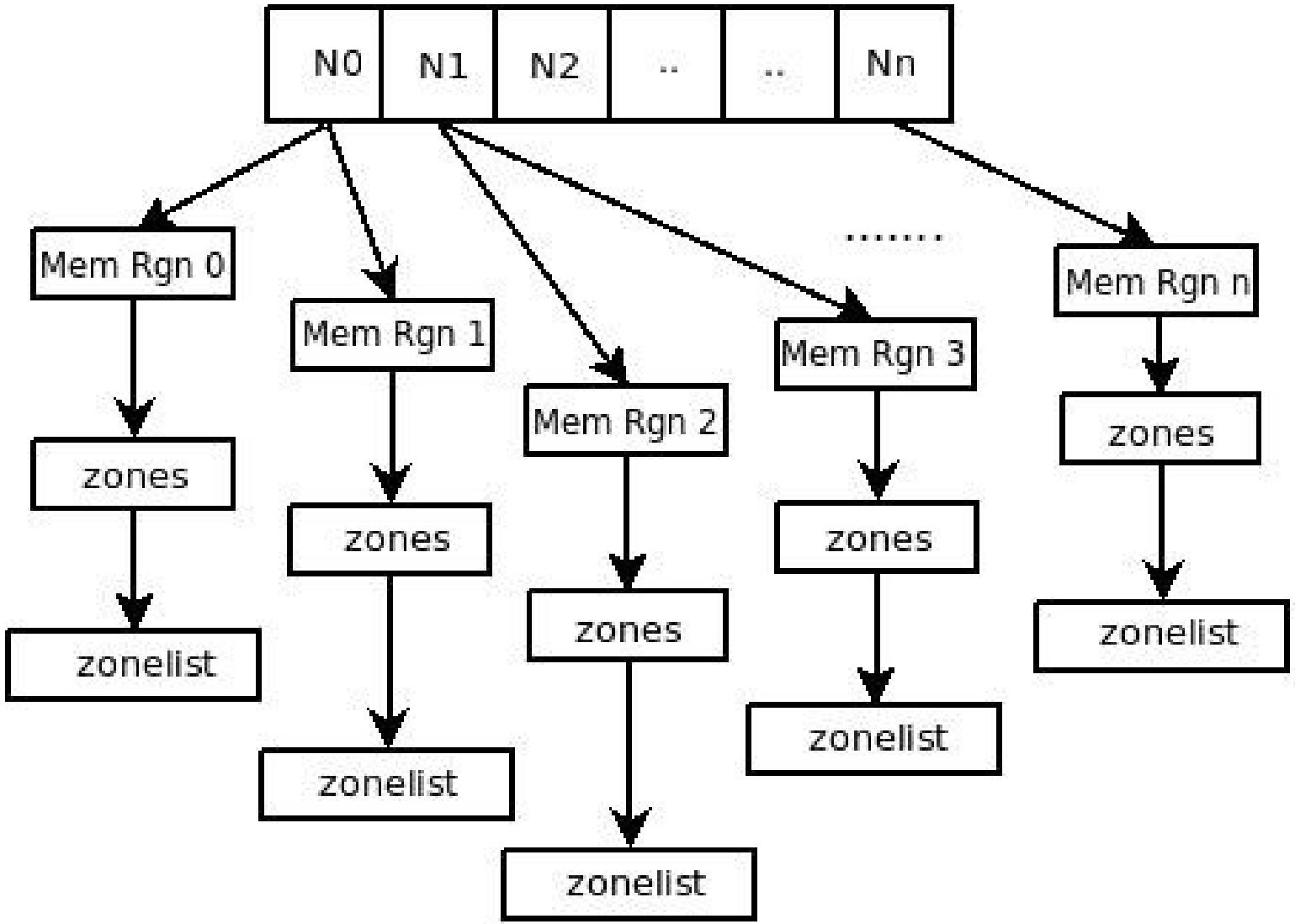
- Organize memory into distinct memory regions
- Align memory operations across these regions
- Allocate pages so that we fill one region at a time
- Maintain per-region allocated/free memory statistics
- Free pages with an intention to completely free a region
- Track page-type (dirty, read-only, etc), to estimate the cost of reclaim vs migrate vs spilling over to next region
- Design OS callbacks to track first page allocation and last page deallocation
- Mechanism to group regions



Present VM organization

Implementation Schemes

- Memory regions
  - Data structure between nodes and zones, that aligns with memory hardware power management granularity



Pros & Cons of Memory Regions

IBM

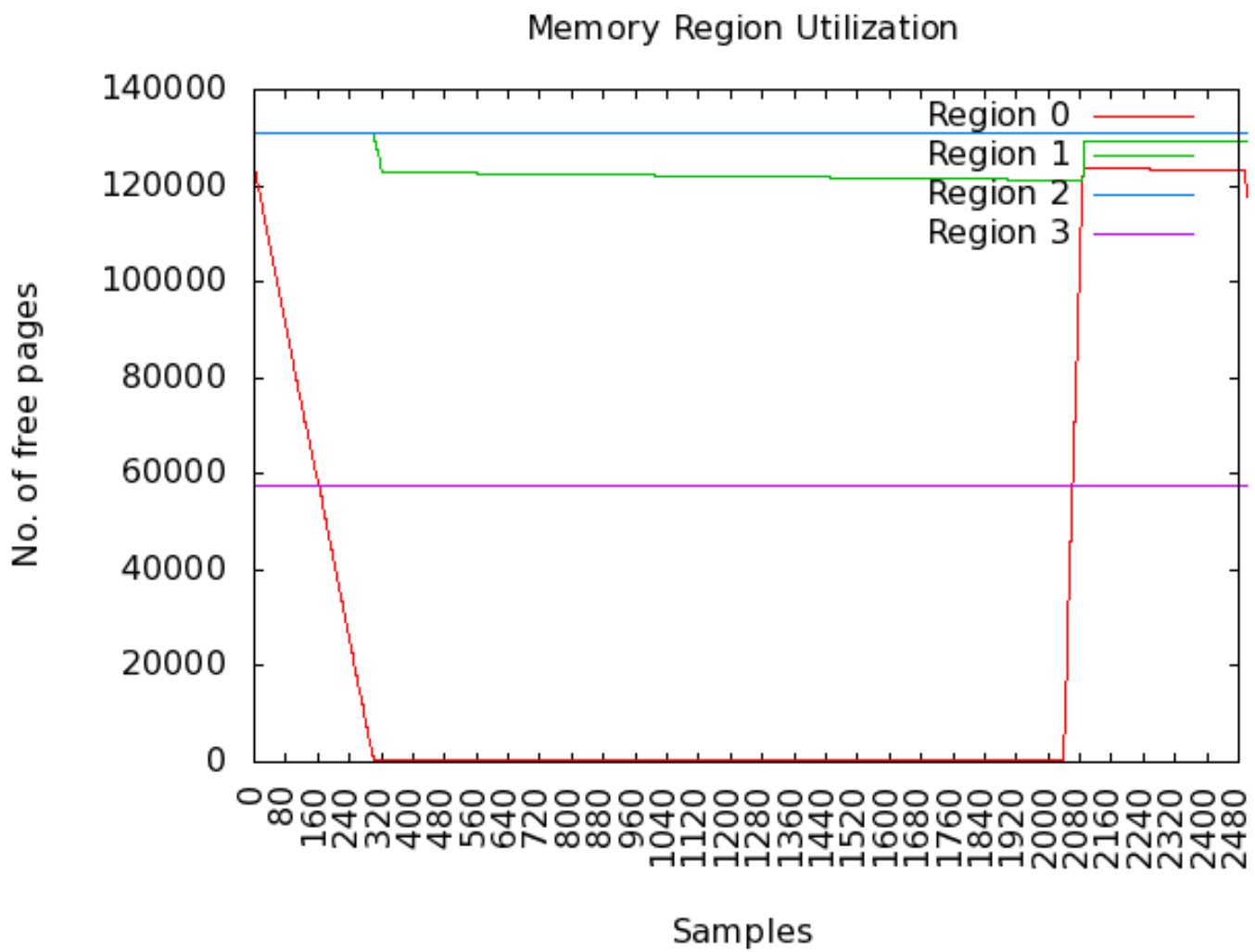
- Advantages
  - A region aligns with hardware address boundary
  - Sequential allocation within region
  - Targeted reclaim within region
  - Statistics gathered on a per-region granularity
  - Grouping of regions

- Disadvantages
  - Zones are fragmented
  - Dereference overhead
  - Modification of core VM structures (memory layout change)

Linux

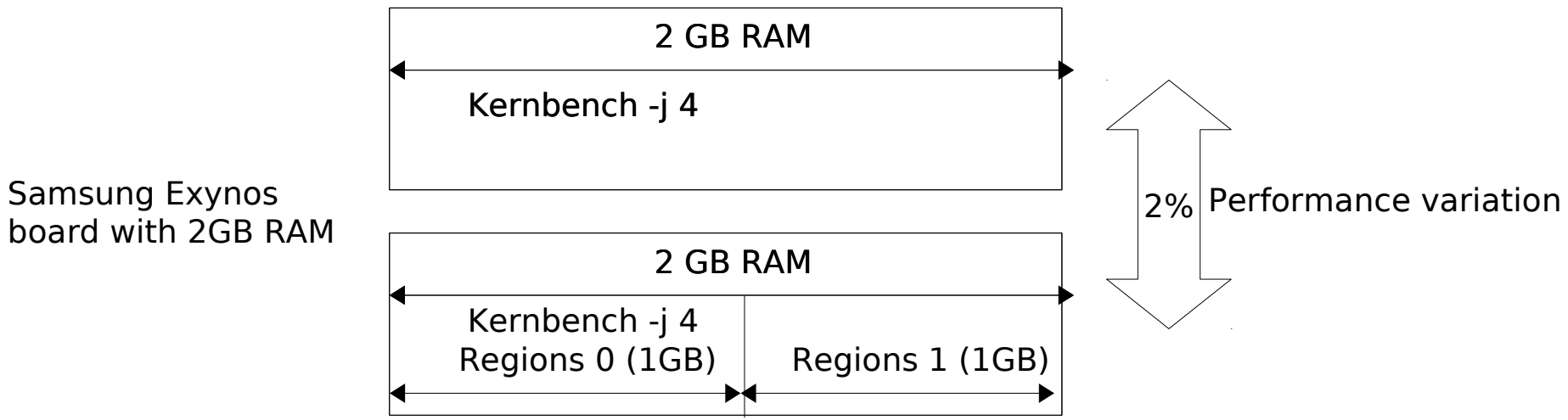
Memory Region Experimental Data

- Allocation of memory from one regions at a time



Memory Region Overhead

- Estimating overhead of memory region -- zone fragmentation
- Lmbench: File & VM latencies vary from 0-3% variation
- Kernbench on 2 GB RAM with two memory regions



Alternate Implementation Schemes

- Track statistics using buddy allocator
  - No address boundary information, so cumbersome to traverse and find pages belonging to the same region
  - Can control access to region
  - Beneficial for PASR
- Memory compaction technique or Lumpy reclaim
  - Free-up region sized chunks of memory
  - Move allocated/free pages around
- Ballooning driver -- create artifical memory pressure
  - Free unused pages to aid PASR
- Contiguous Memory Allocator (CMA)
- Fake NUMA nodes

Implementation Challenges

- Memory interleaving vs power savings
- Accurately capturing rate of memory references
- Low overhead tracking of region usage statistics is difficult
- Kernel memory allocations are non-movable
- Evaluating the threshold for reclaim vs migration
- Determining the amount of memory required to sustain certain performance level (unlike CPU)
- Virtualization – Guest VM does not have complete view of memory

Reference

- Ankita's memory regions (LWN)
- Summary of requirements
- Kernel Summit 2011 proposal and discussions
  - Memory Compaction (Mel Gorman)
- Devfreq, DVFS for devices
- General PASR
- Memory Reference Pattern Instrumentation
- Memory Power Management
  - http://www.energystar.gov/ia/products/downloads/GPapadopoulos_Keynote.pdf
  - http://www.crucial.com/pdf/Crucial_energy_efficient_memory.pdf

Legal statements