Accelerating Compressed Far Memory

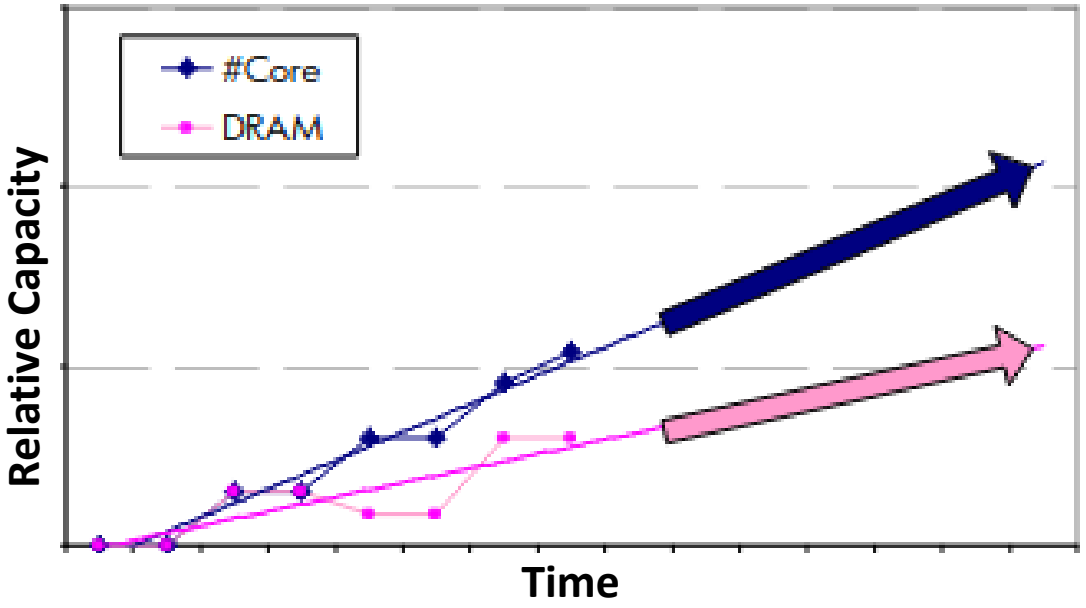
Neel Patel, Mohammad Alian

Electrical Engineering and Computer Science

KU THE UNIVERSITY OF KANSAS

Problem With DRAM

- DRAM is an **inflexible, costly** resource and is extremely **contended** in cloud environments
  - × **Coarse-granularity upgrades**: smallest memory capacity increase for a two-socket Intel Xeon 2nd Gen Scalable server is 25%
  - × **High power consumption**: 38% of Meta's Power Consumption
  - × **High cost**: 33% of Meta's Expenses

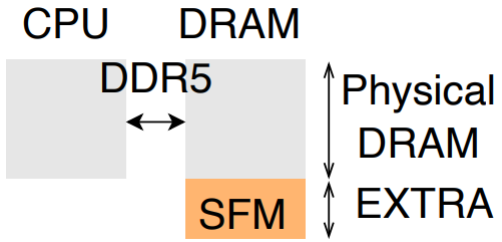


Dropping DRAM Capacity Per Core

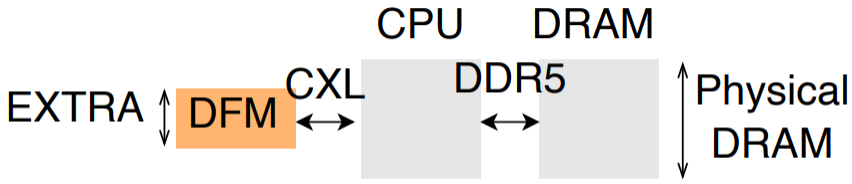
(Source: **Disaggregated memory for expansion and sharing in blade servers**
-Lim et. al. 2009)

Far Memory Solutions

- Software-Defined Far Memory (SFM):
  - ✓**Compressed** memory: shrink “cold” data using compression
- Disaggregated Far Memory (DFM):
  - ✓**Remote-accessible** memory: take advantage of “stranded” memory
  - ✓**CXL-attached** DRAM: add memory capacity to system bus
- Far memory associates cost with accesses
  - Network traversal
  - Decompression
  - CXL latency



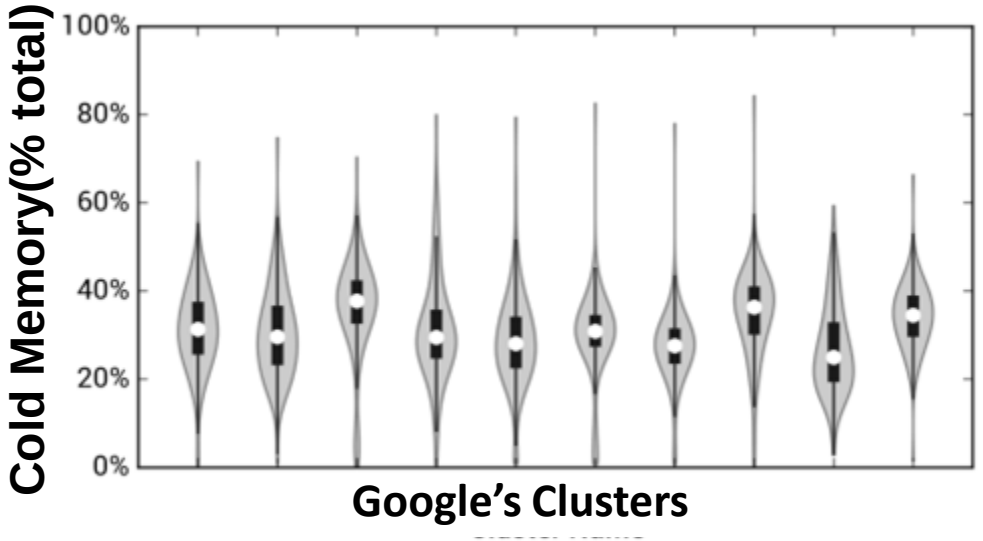
Software-Defined Memory



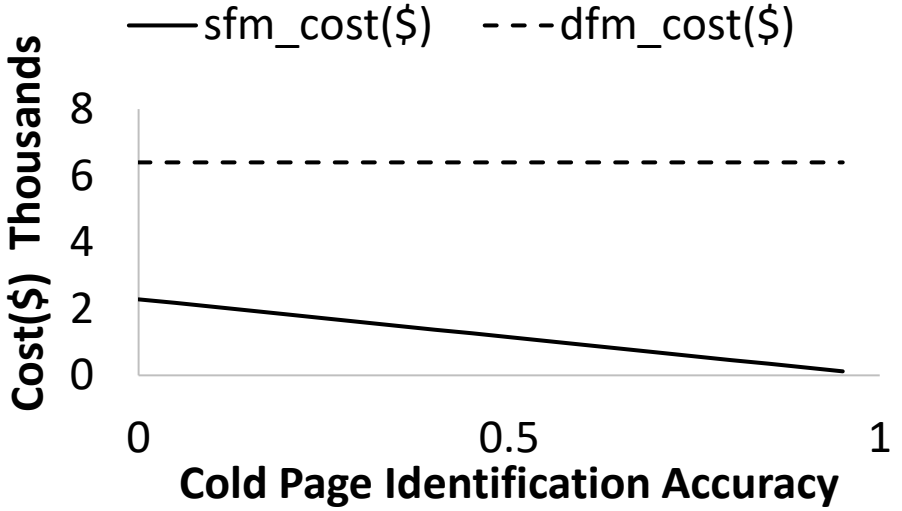
Disaggregated Far Memory

Software-Defined Far Memory (SFM)

- Benefits:
  - ✓Reduced Memory Consumption: Smaller data footprint
  - ✓**Low Risk and Low Cost**
  - ✓Flexible: Dynamically provision “near" and “far" memory capacities
  - ✓Smaller Failure Domain: No pages on remote nodes
- Drawback:
  - × **Suboptimal Datapath**: moves cold data from DRAM on-chip
    - × Cache contention
    - × Memory bandwidth utilization



Source: Software-Defined Memory – Cavilla et. al



XFM

Goal: Accelerate Software-Defined Far Memory using Near-Data Processing

- Benefits:
  - ✓**Cold data stays off-chip**
  - ✓**Freed** (De)compression-related **CPU Cycles**
- Challenges:
  - Mapping cold OS pages to DRAM
  - Preserving and prioritizing host accesses
  - Cache coherency