DRAM Organization Aware OS Page Allocation

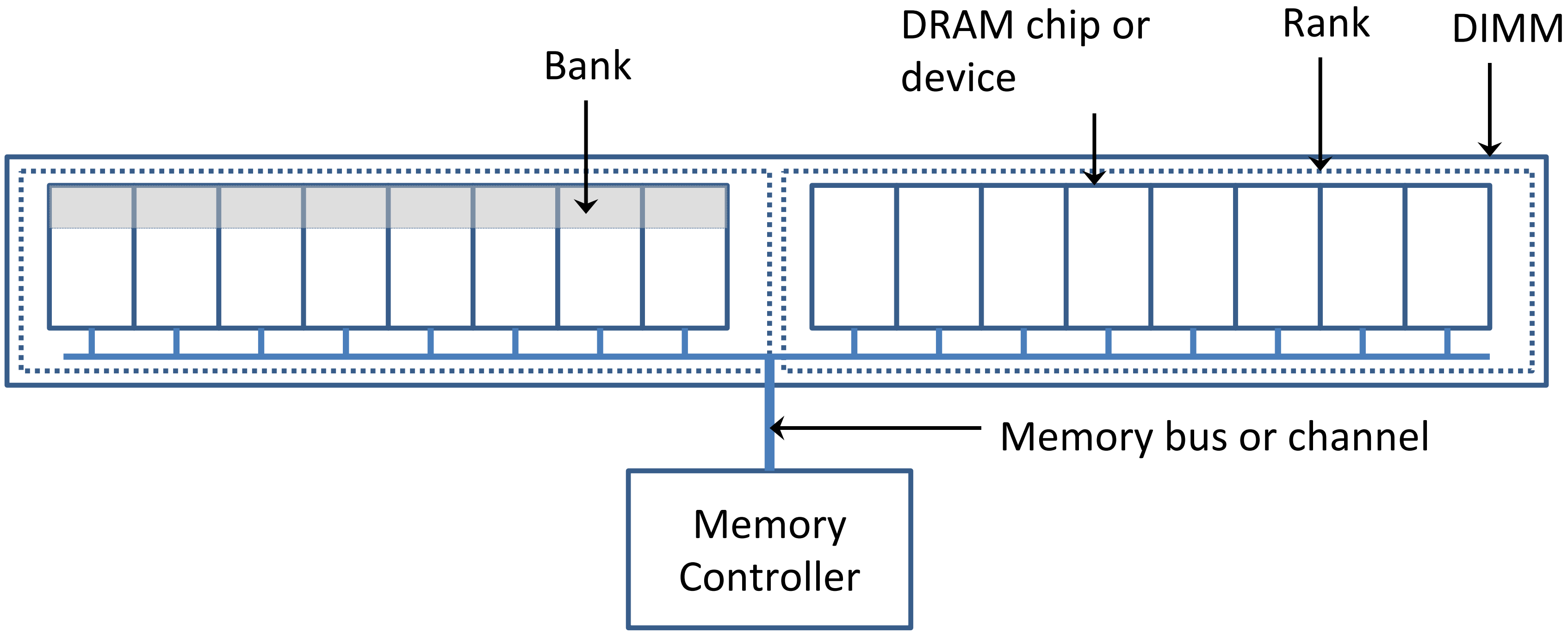
Manu Awasthi

IIT Gandhinagar

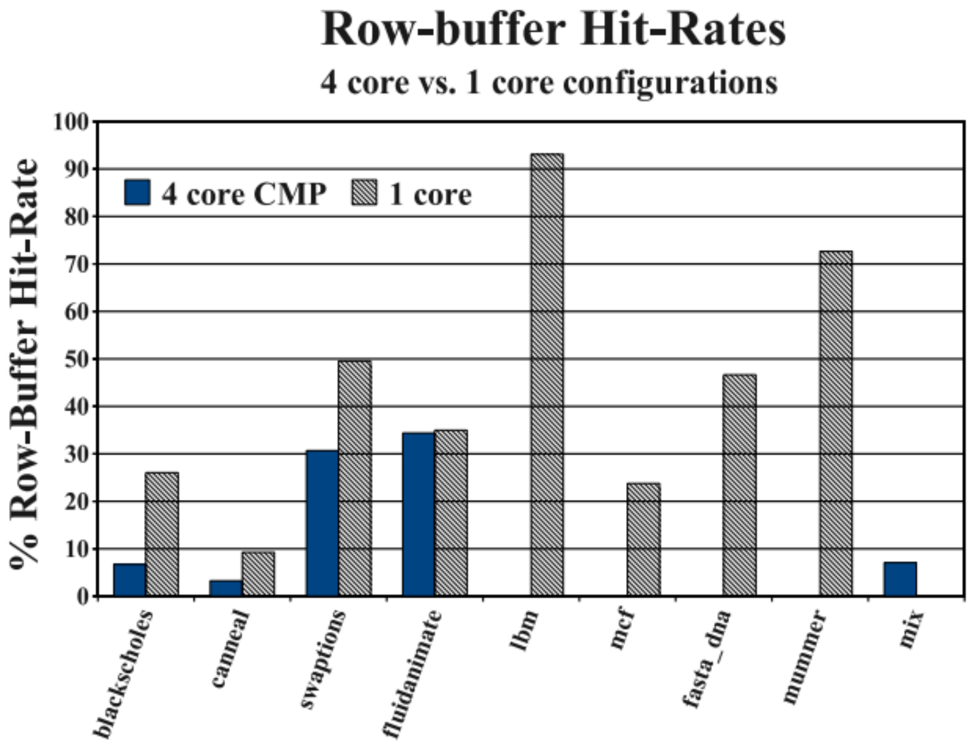
manua@iitgn.ac.in

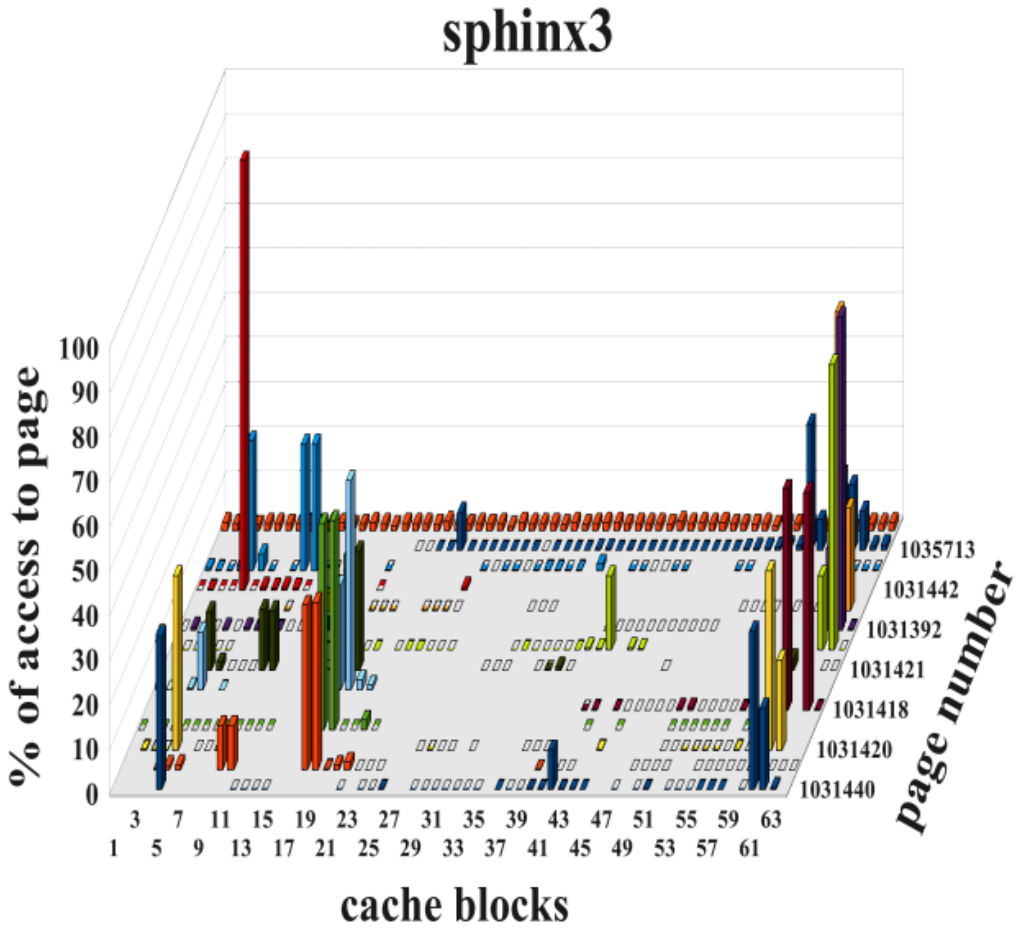
Introduction and Motivation

- Number of NUMA nodes per server are increasing
- Each NUMA node has multiple channels, which in turn has multiple DIMMs
  - Each DIMM is logically arranged into ranks, banks, rows and columns



Motivational Results





Memory Allocation Dynamics

- Two views of memory allocation – OS and DRAM
- Operating System allocates memory in page sized chunks
  - Typically 4 KB in size
  - Gives priority to physically proximal data allocation e.g. same NUMA node
  - However, doesn’t take into account internal organization of DIMM
- Data is distributed across DIMMs by memory controller policy
  - Address determines channel/rank/bank/row/column etc
  - Policy decided for the entire system, or can be per-memory controller

The two views are unrelated to each other!

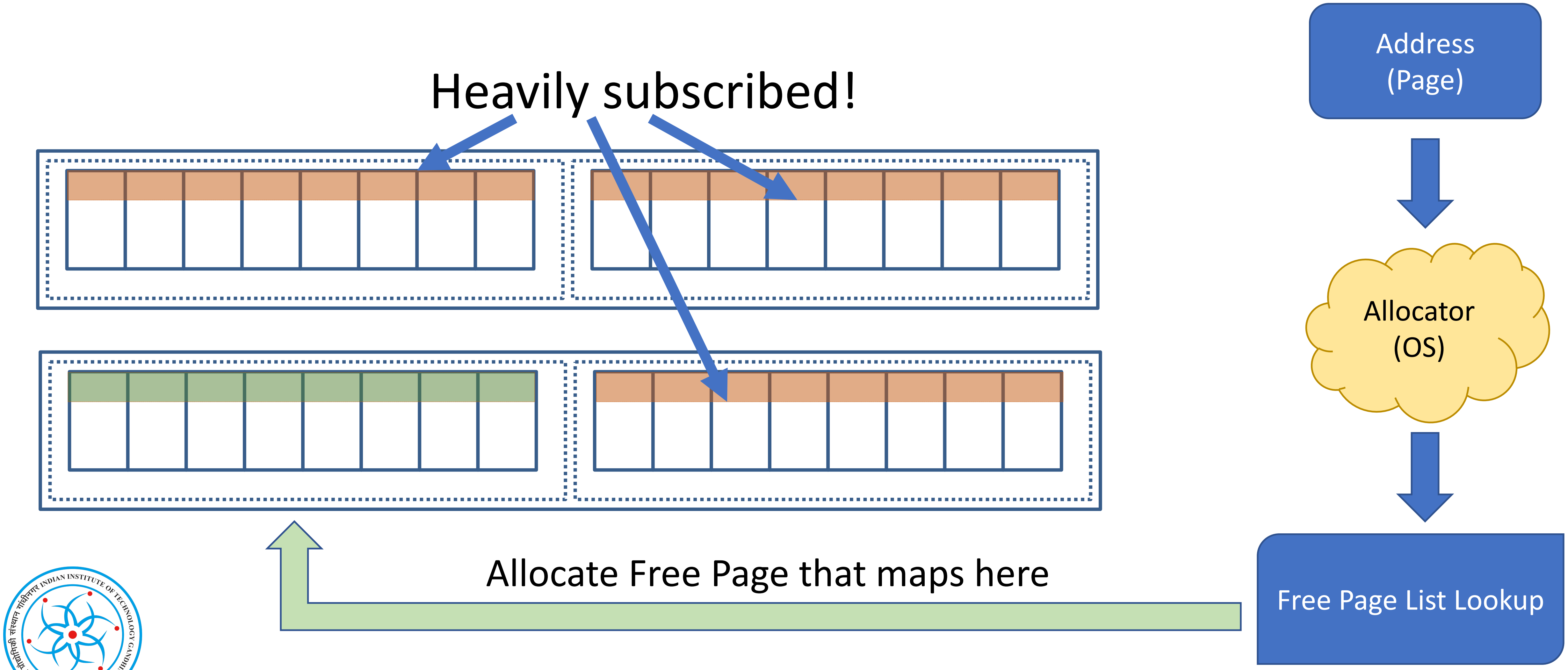
Allocation Policies

- OS allocates memory from the first page in the free page list
  - Doesn’t account for where and how that page maps to in DRAM

Need to account for allocation of pages across DRAM!

- Keep track of how the allocation has been done in recent past; which channels/ranks/banks/arrays are “oversubscribed”
  - Allocate memory across ones that are relatively “free”
  - OS has to work in conjunction with hardware, possibly through performance counters

Allocation Policy



Benefits

- Increases parallelism; increases total memory throughput and bandwidth
- Decreases queuing delays across memory devices/controllers; reduces overall latency of access

Both of these are passed onto end user application, resulting in better application performance