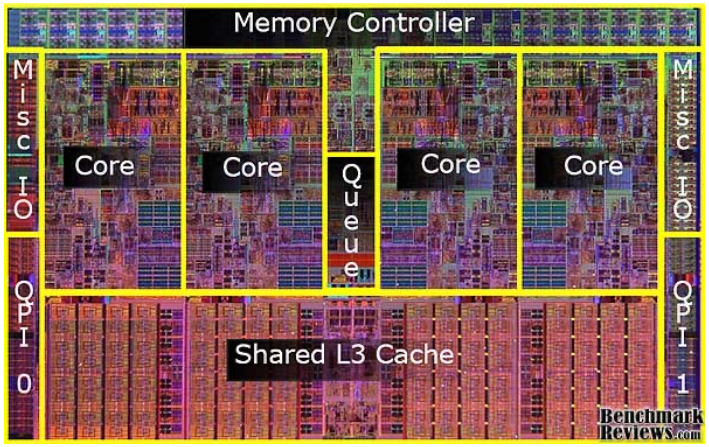
PALLOC: DRAM Bank-Aware Memory Allocator for Performance Isolation on Multicore Platforms

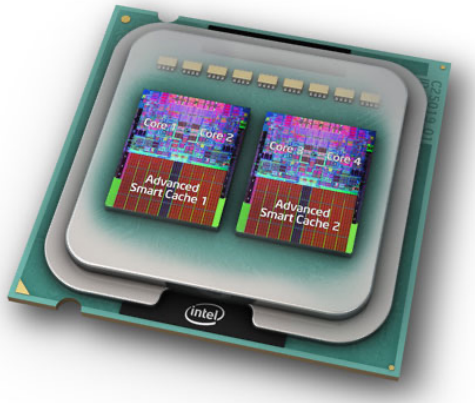
Heechul Yun*, Renato Mancuso+, Zheng-Pei Wu#, Rodolfo Pellizzoni#

*University of Kansas, +University of Illinois , #University of Waterloo

KU THE UNIVERSITY OF KANSAS

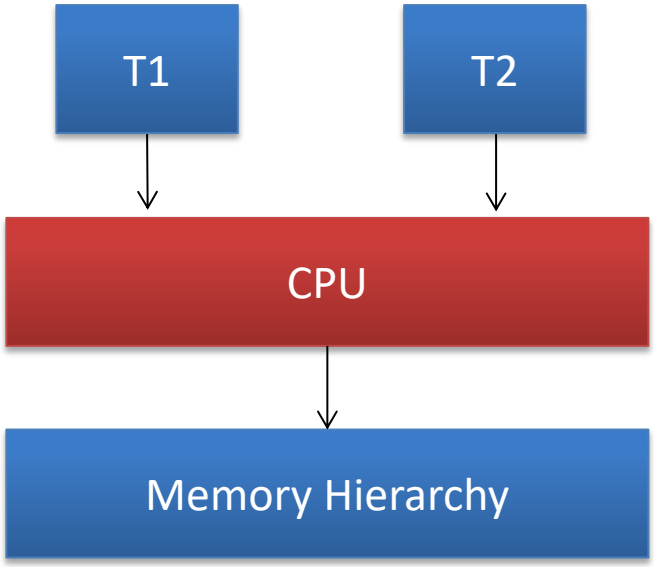
Multicore



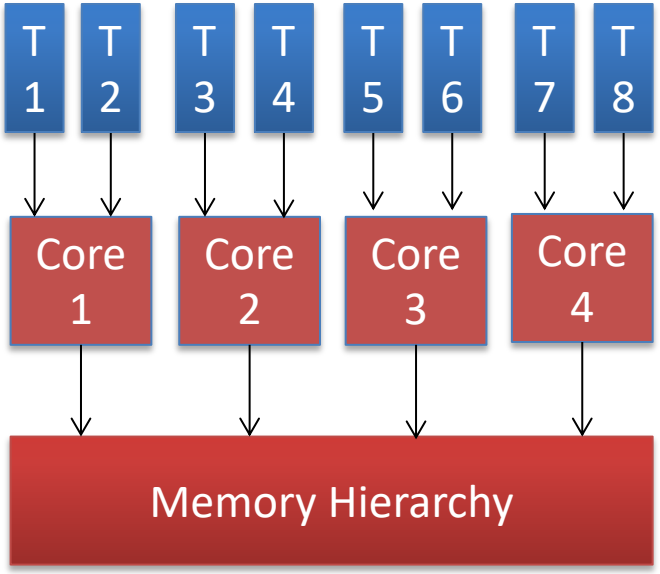


Server ⇨ Desktop ⇨ Mobile ⇨ **RT/Embedded**

Challenges: **Shared Memory Hierarchy**



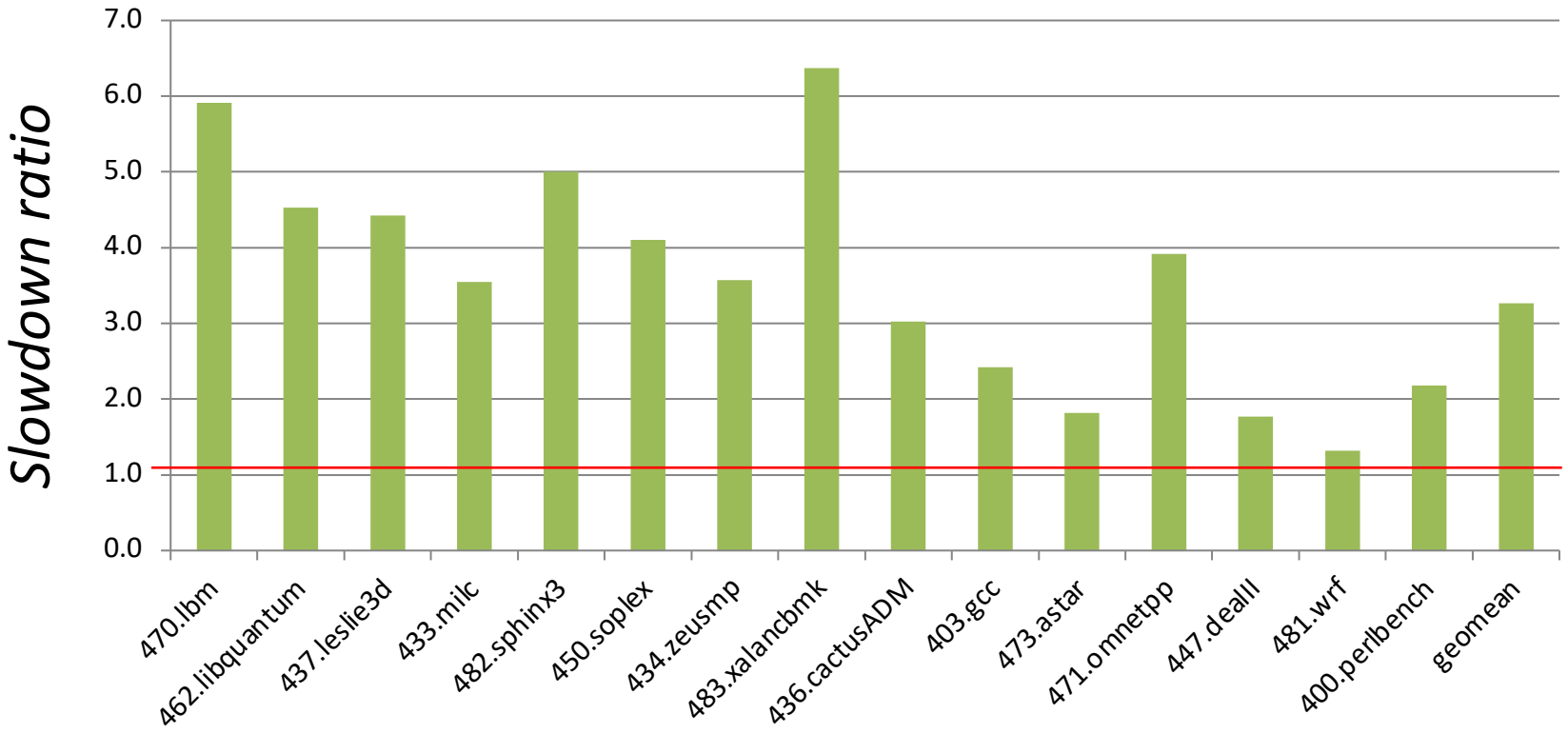
Unicore



Multicore

Performance Impact

Impact of Memory Interference

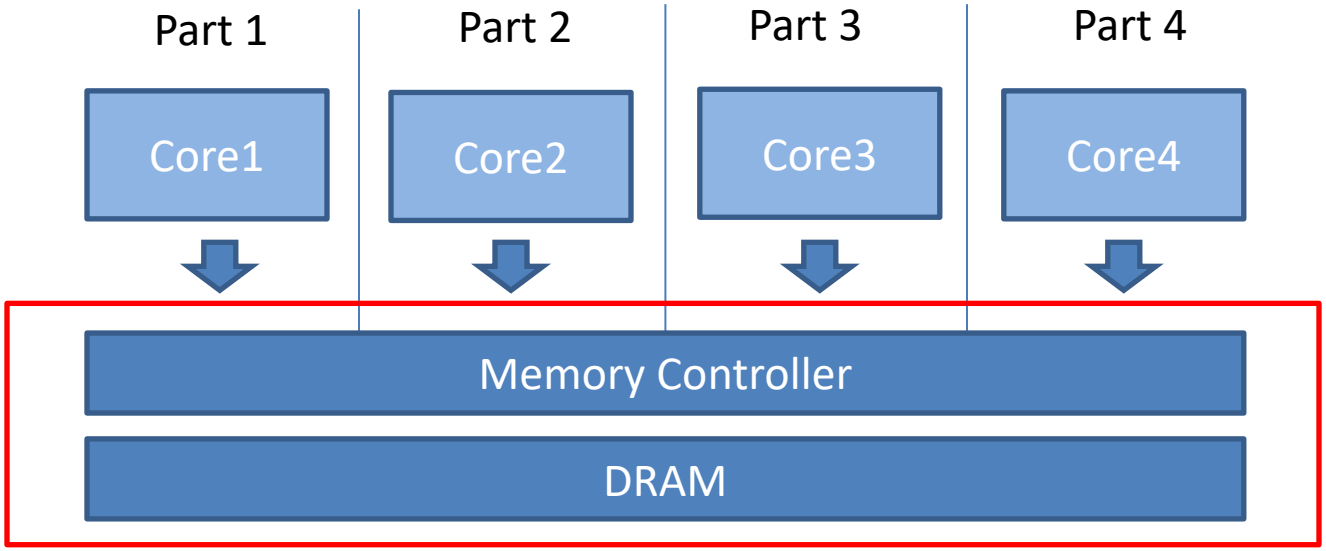


- Setup: Core0: X-axis, **Core1-3: 470.lbm x 3 (interference**)
- Slowdown ratio = Solo IPC / Corun IPC

KU THE UNIVERSITY OF KANSAS

(*) Measured on Intel Xeon 3530 (4 cores), 8GB 1ch DDR3 DRAM

Memory Performance Isolation

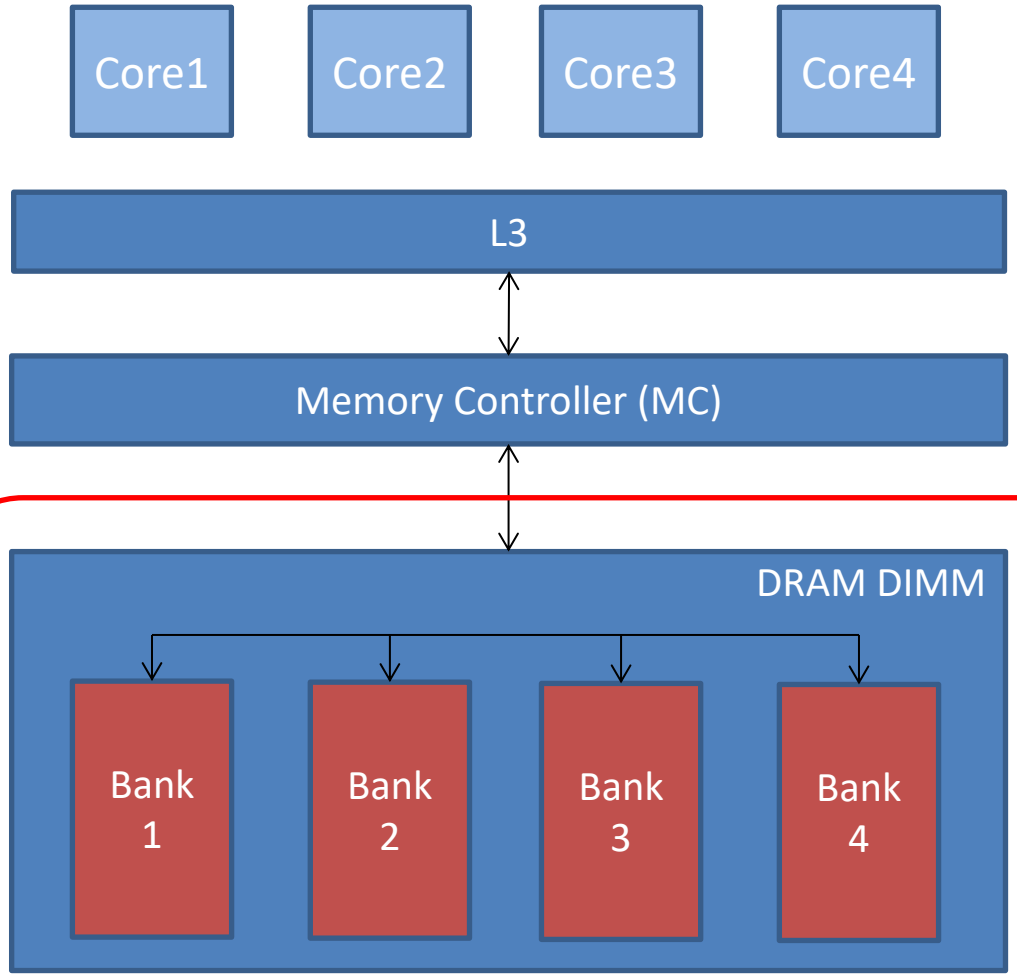


- Q. How to reduce memory contention?

Outline

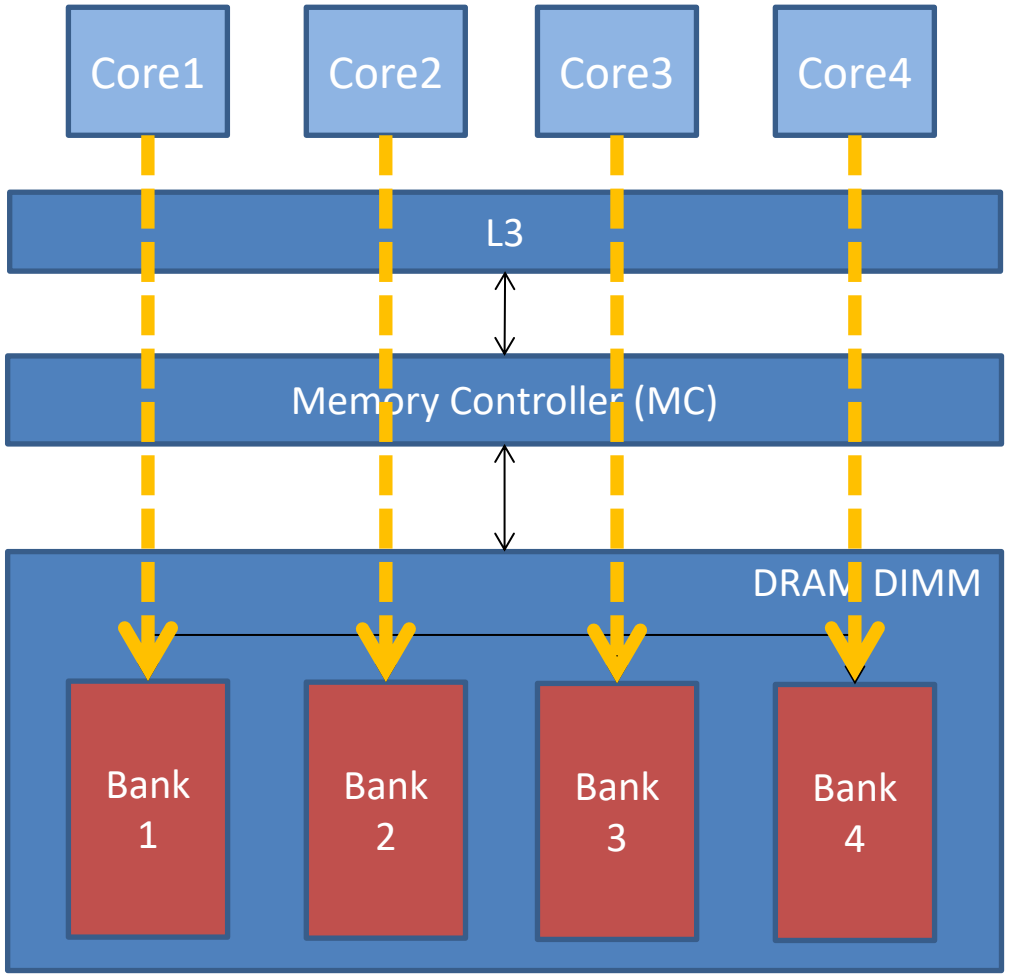
- Motivation
- **Background & Problems**
- PALLOC: DRAM Bank Allocation Mgmt.
- Evaluation
- Conclusion

Background: DRAM Organization



- Have multiple banks
- Different banks can be accessed in parallel

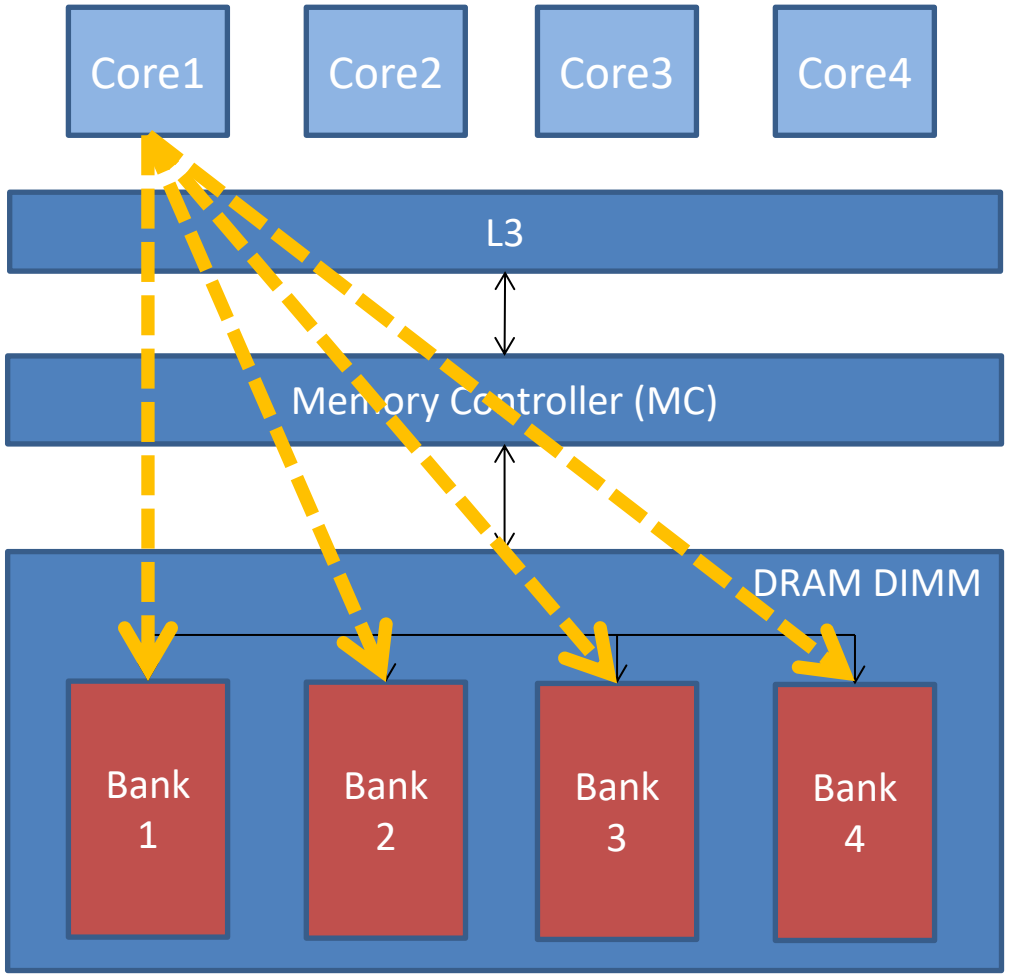
Best-case



Fast

- Peak = 10.6 GB/s
  - DDR3 1333Mhz

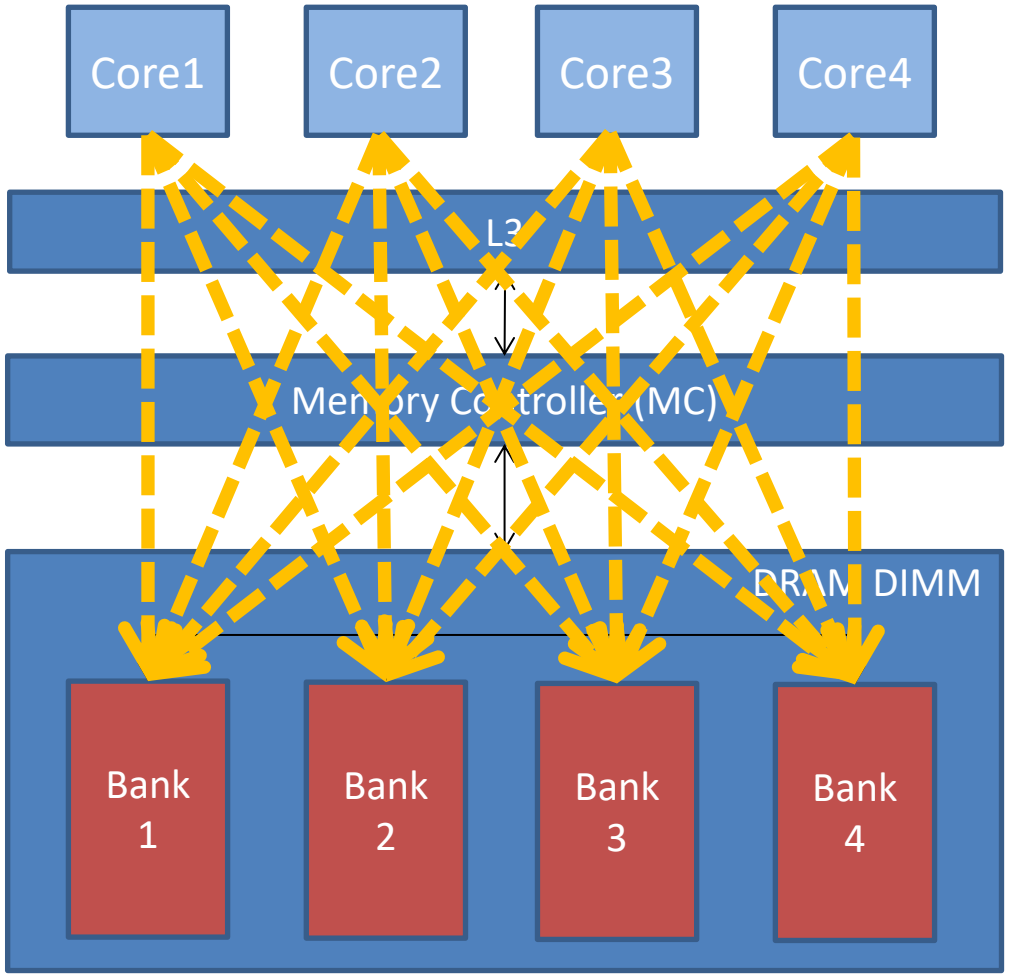
Best-case



Fast

- Peak = 10.6 GB/s
  - DDR3 1333Mhz
- Out-of-order processors

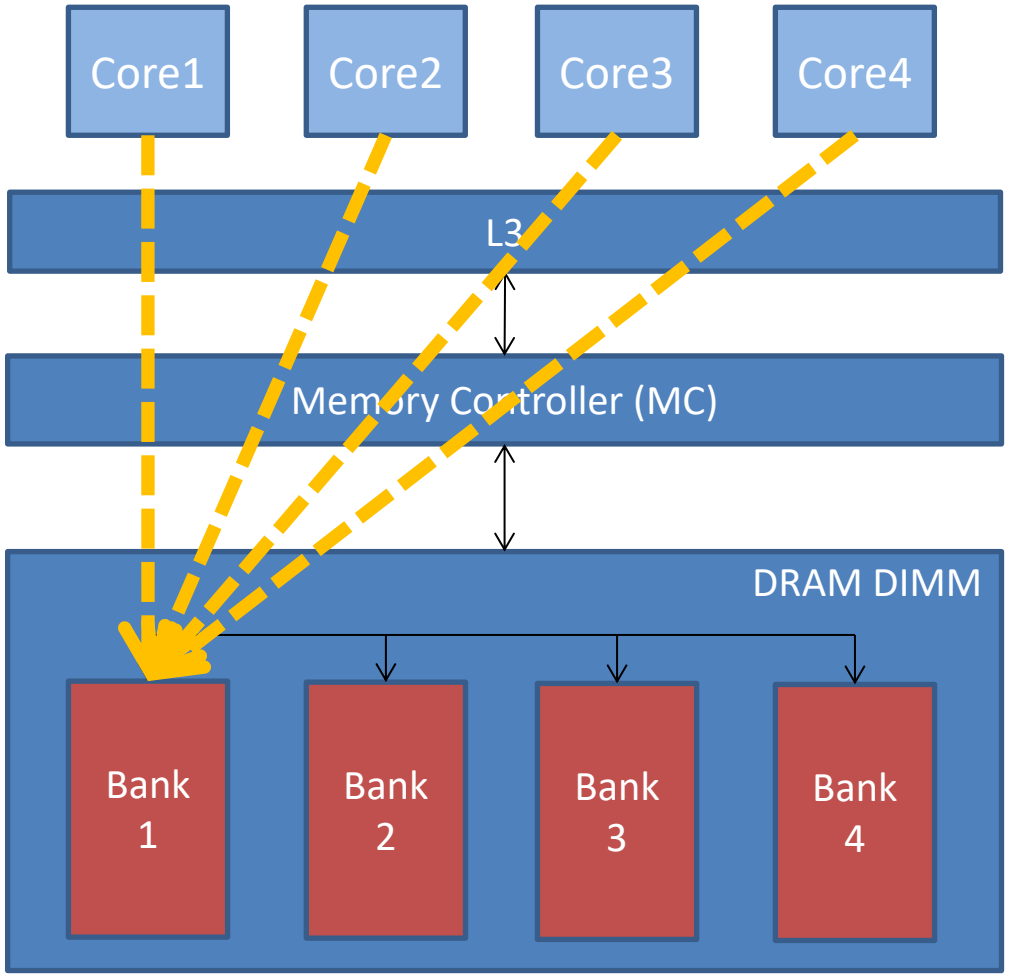
Most-cases



Mess

- Performance = ??

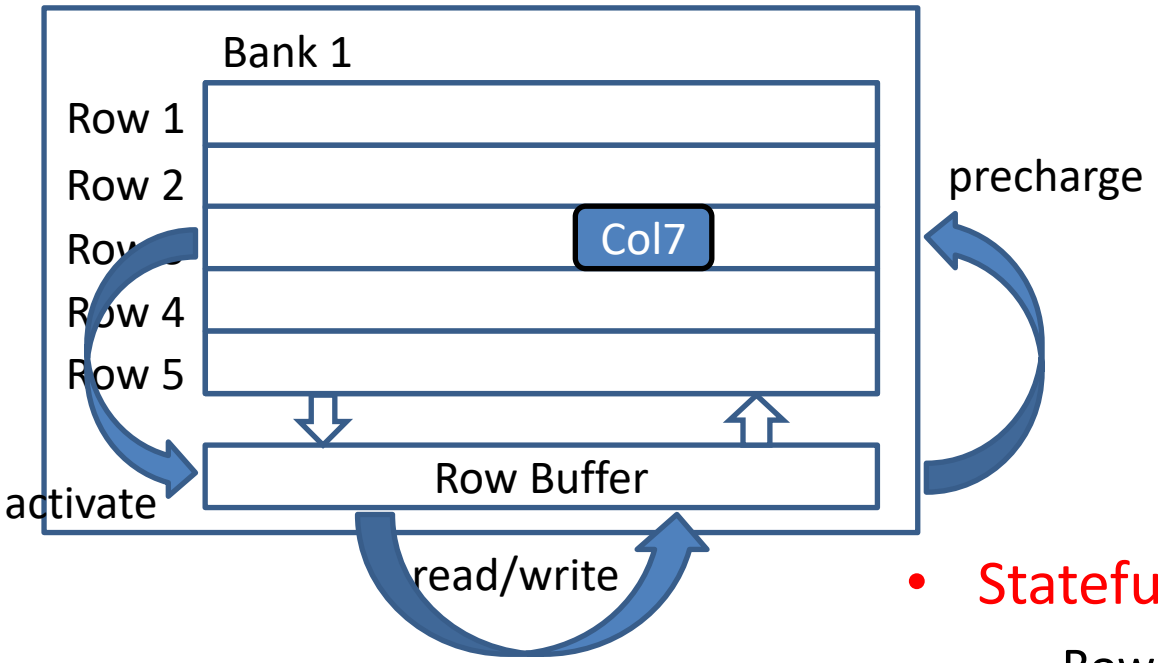
Worst-case



Slow

- 1bank b/w
  - Less than peak b/w
  - How much?

Background: DRAM Operation

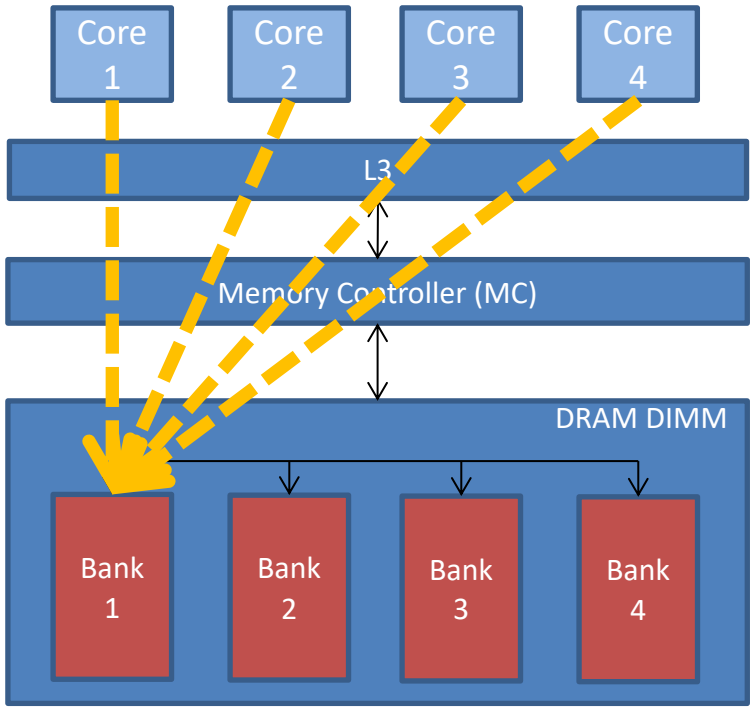


READ (Bank 1, Row 3, Col 7)

- Stateful per-bank access time
  - Row miss: 19 cycles
  - Row hit: 9 cycles

(*) PC6400-DDR2 with 5-5-5 (RAS-CAS-CL latency setting)

Real Worst-case



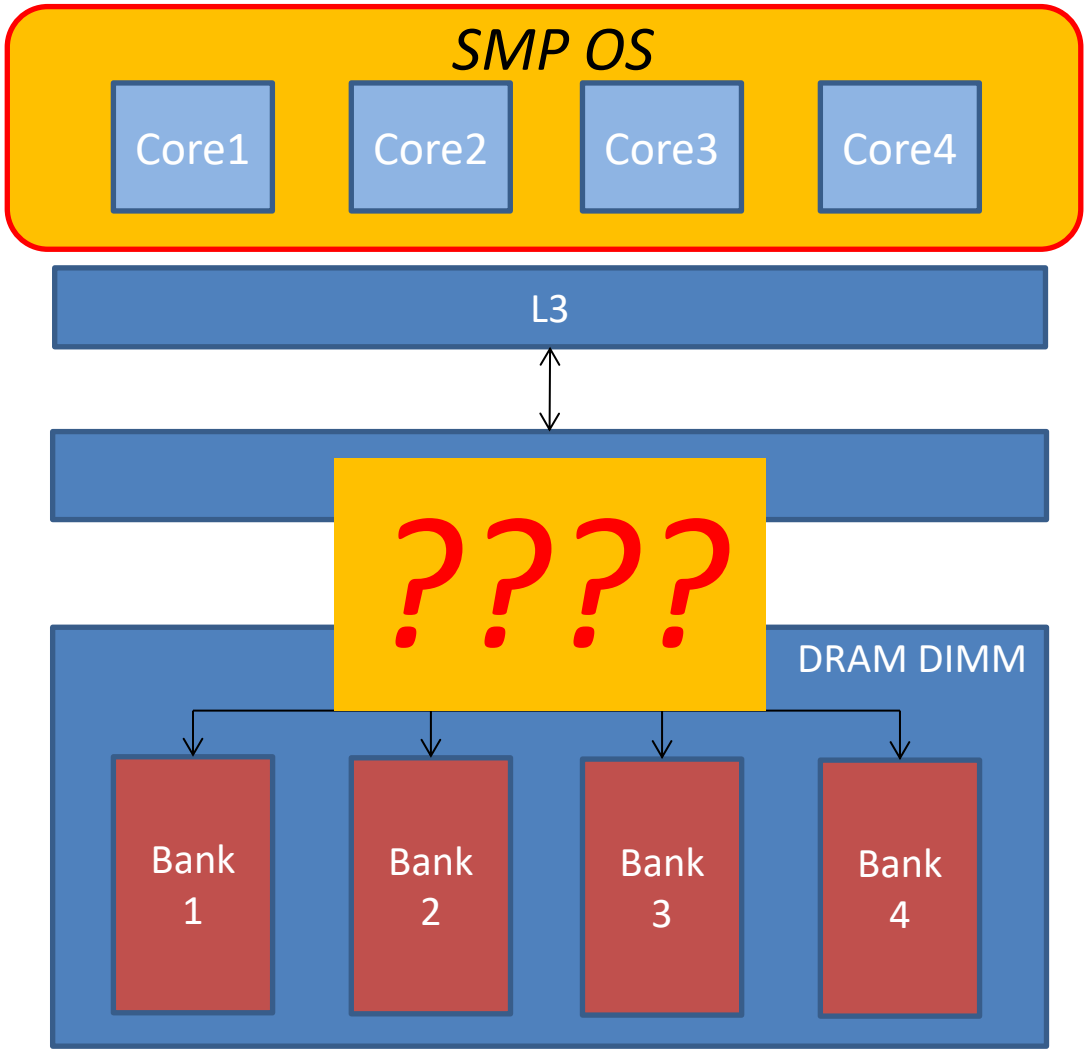
Request order

time

Row 1
Row 2
Row 3
Row 4
Row 1
Row 2
...

1 bank & always row misses → ~1/10 peak b/w

Problem



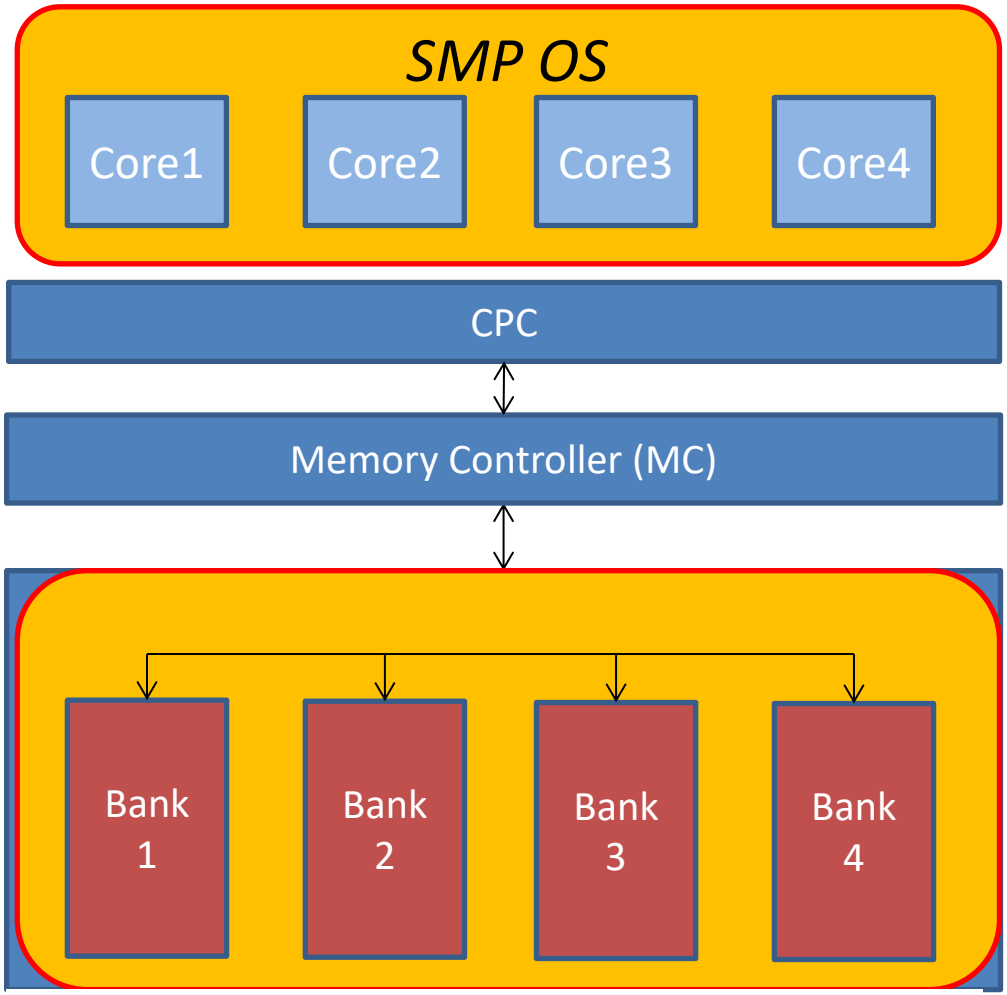
- OS does NOT know DRAM banks
- OS memory pages are spread all over multiple banks

Unpredictable memory performance

Outline

- Motivation
- Background & Problems
- **PALLOC: DRAM Bank Allocation Mgmt.**
- Evaluation
- Conclusion

PALLOC



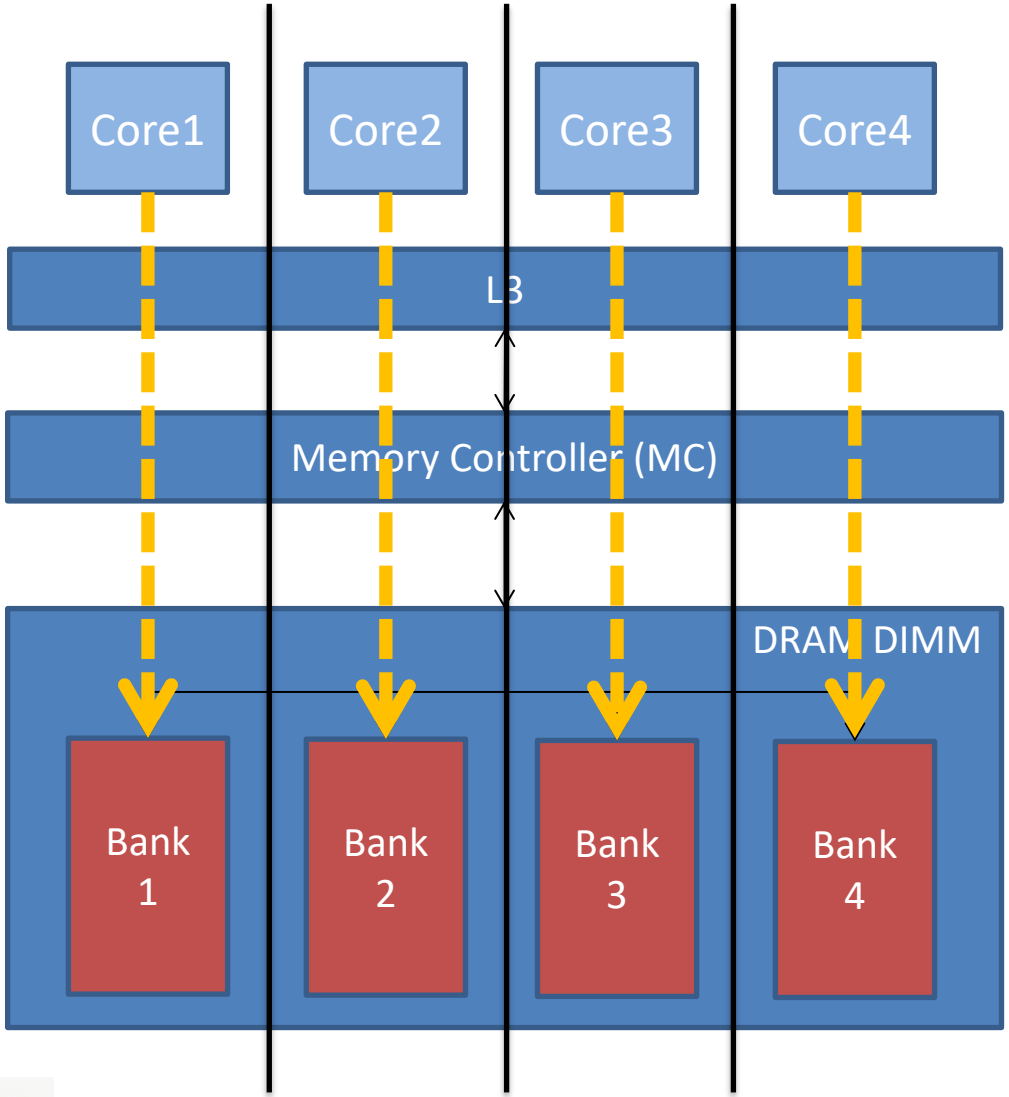
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33-35
—	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R	S	S	B	B	B	C	C	C	C	C	C	C	M	C	C	C	O

Figure 12. Cache-Line and Chip Select Interleaved Address Bit Encoding for 14 × 10 × 2 DDR

- OS is aware of DRAM mapping
- Each page can be allocated to a desired DRAM bank

Flexible allocation policy

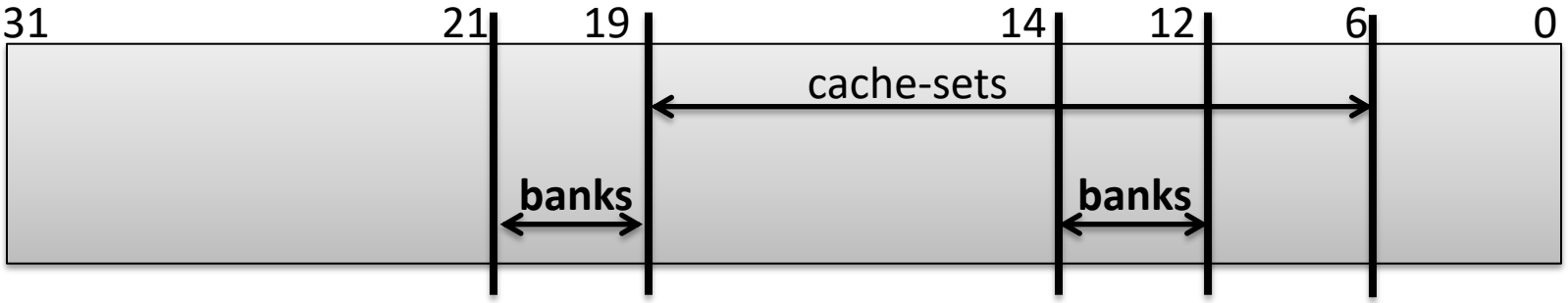
PALLOC



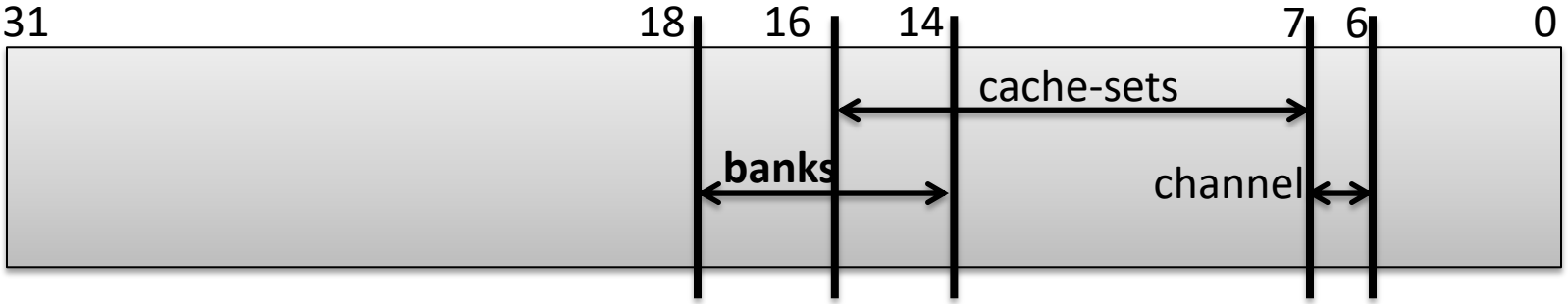
- Private banking
  - *Allocate* pages on certain *exclusively* assigned banks

Eliminate Inter-core bank conflicts

Identifying Memory Mapping



Intel Xeon 3530 + 4GiB DDR3 DIMM (16 banks)



Freescale P4080 + 2x2 GiB DDR3 DIMM (32 banks)

- Memory mappings are platform specific
- We developed a detection tool software

PALLOC Implementation

- Modified Linux kernel's buddy allocator
  - DRAM bank-aware page frame allocation at each page fault

Simplified Pseudocode

```
/* return a free page frame from the selected banks */
struct page *palloc_find_page(bankmap)
{
    for (bank <- bankmap) {
        if (!empty(bank_bins[bank]) {
            page = pop(bank_bins[bank])
            return page;
        }
    }
    return NULL;
}

/* return a free page frame (4KB) */
struct page *__rmqueue_smallest(...)
{
    freelist <- free pages
    bankmap <- selected banks

    /* search page from bank cache */
    page = palloc_find_page(bankmap);
    if (page)
        return page;

    /* build bank cache & search page */
    for(page <- freelist) {
        bank = addr_to_bank(page);
        push(bank_bins[bank], page);
        page = palloc_find_page(bankmap);
        if (page)
            return page;
    }
    return NULL;
}
```

PALLOC Interface

- Example: per-core private banking (PB)

```
# cd /sys/fs/cgroup
# mkdir core0 core1 core2 core3
  ➔ create 4 cgroup partitions

# echo 0-3 > core0/palloc.dram_bank
  ➔ assign bank 0 ~ 3 for the core0 partition.
# echo 4-7 > core1/palloc.dram_bank
# echo 8-11 > core2/palloc.dram_bank
# echo 12-15 > core3/palloc.dram_bank
```

KU THE UNIVERSITY OF KANSAS

Outline

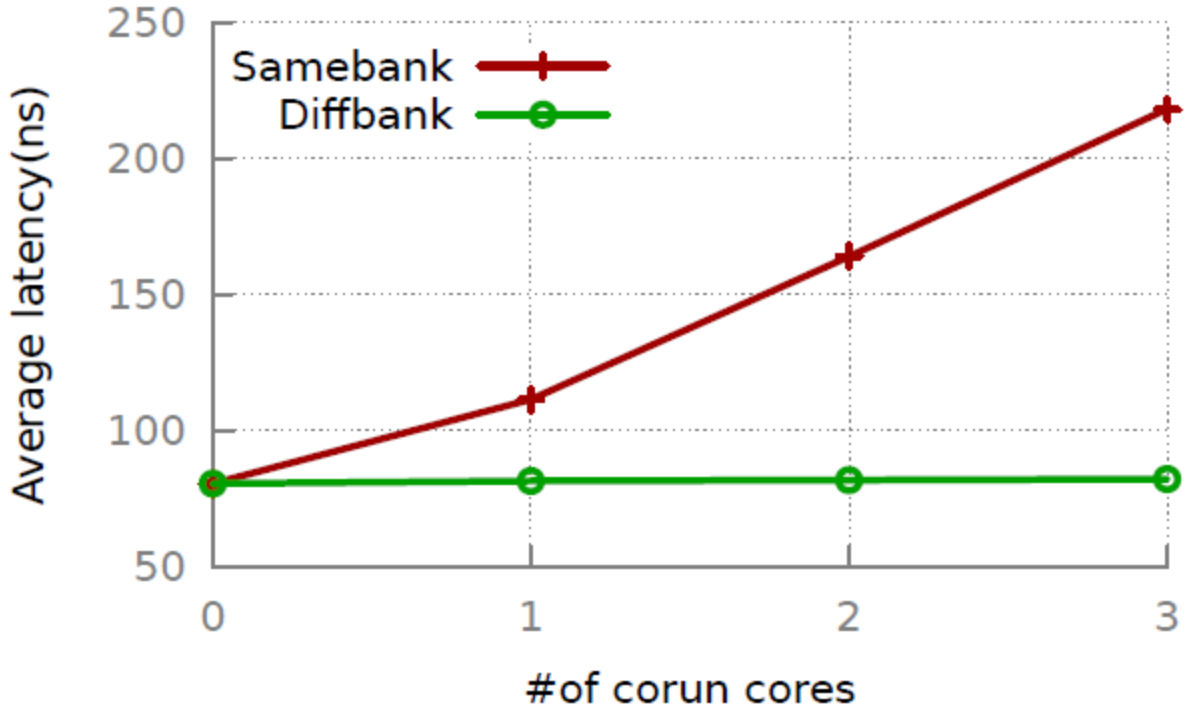
- Motivation
- Background & Problems
- PALLOC: DRAM Bank Allocation Mgmt.
- **Evaluation**
- Conclusion

Evaluation Platforms

- **Platform #1: Intel Xeon 3530**
  - X86-64, 4 cores, 8MB shared L3 cache
  - 1 x 4GB DDR3 DRAM module (16 banks)
  - Modified Linux 3.6.0

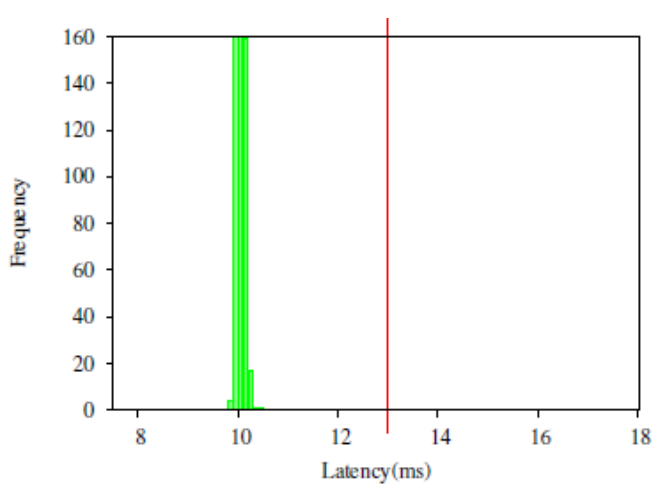
- Platform #2: Freescale P4080
  - PowerPC, 8 cores, 2MB shared LLC
  - 2 x 2GB DDR3 DRAM module (32 banks)
  - Modified Linux 3.0.6

Samebank vs. Diffbank

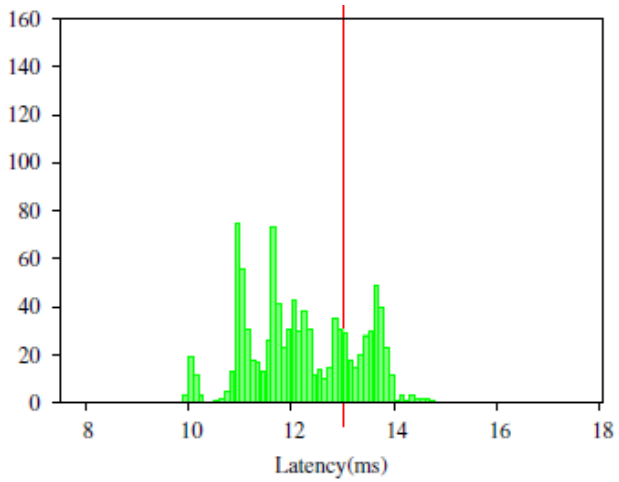


- Samebank: All cores → Bank0
- Diffbank: Core0 → Bank0, Core1-3 → Bank 1-3
  - **Zero interference !!!**

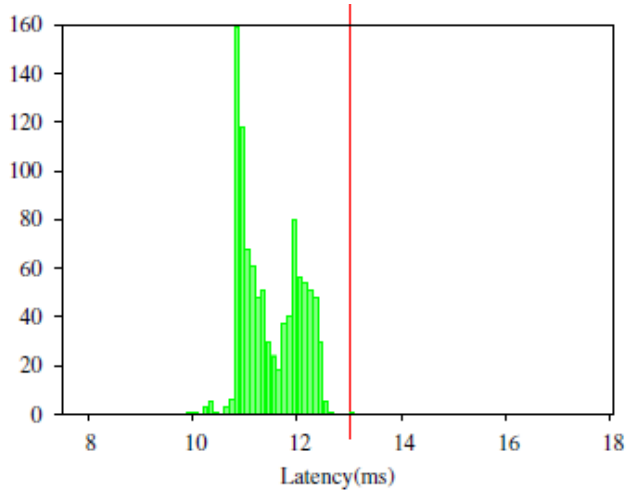
Real-Time Performance



Buddy(solo)



Buddy



PALLOC(diffbank)

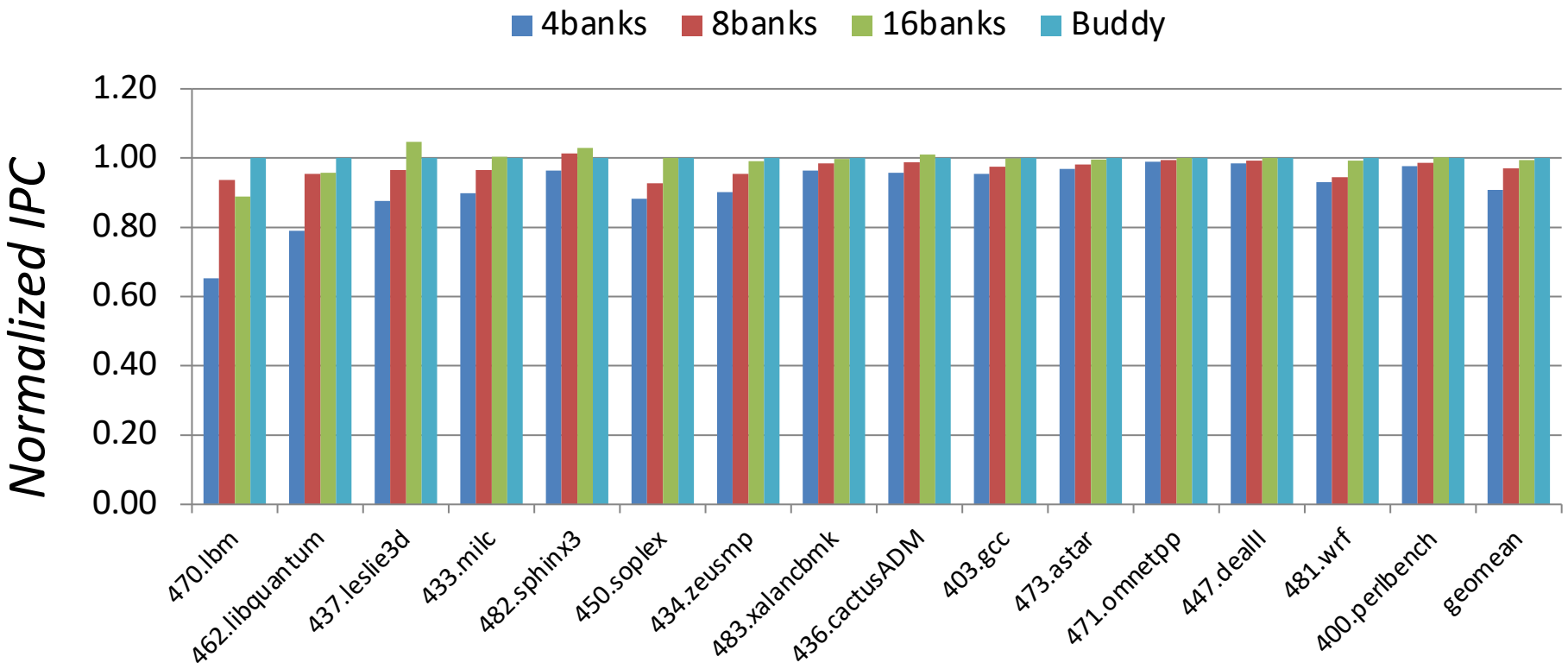
- Setup: HRT → Core0, X-server → Core1
- Buddy: no bank control (use all Bank 0-15)
- Diffbank: Core0 → Bank0-7, Core1 → Bank8-15

SPEC2006

benchmark	bandwidth (MB/s)	RSS (MiB)	average IPC	memory intensity
470.lbm	3158	409	0.88	
462.libquantum	3124	64	0.83	
437.leslie3d	2346	123	0.76	
433.milc	2313	523	0.80	High
482.sphinx3	1649	40	1.11	
450.soplex	1211	108	0.70	
434.zeusmp	1122	502	1.13	
483.xalancbmk	798	110	0.63	
436.cactusADM	702	623	0.93	
403.gcc	618	196	1.02	
473.astar	378	325	0.66	Medium
471.omnetpp	203	173	1.09	
447.dealII	136	6	1.61	
481.wrf	131	570	1.89	
400.perlbench	124	147	1.50	

- Use 15 high-medium memory intensive benchmarks

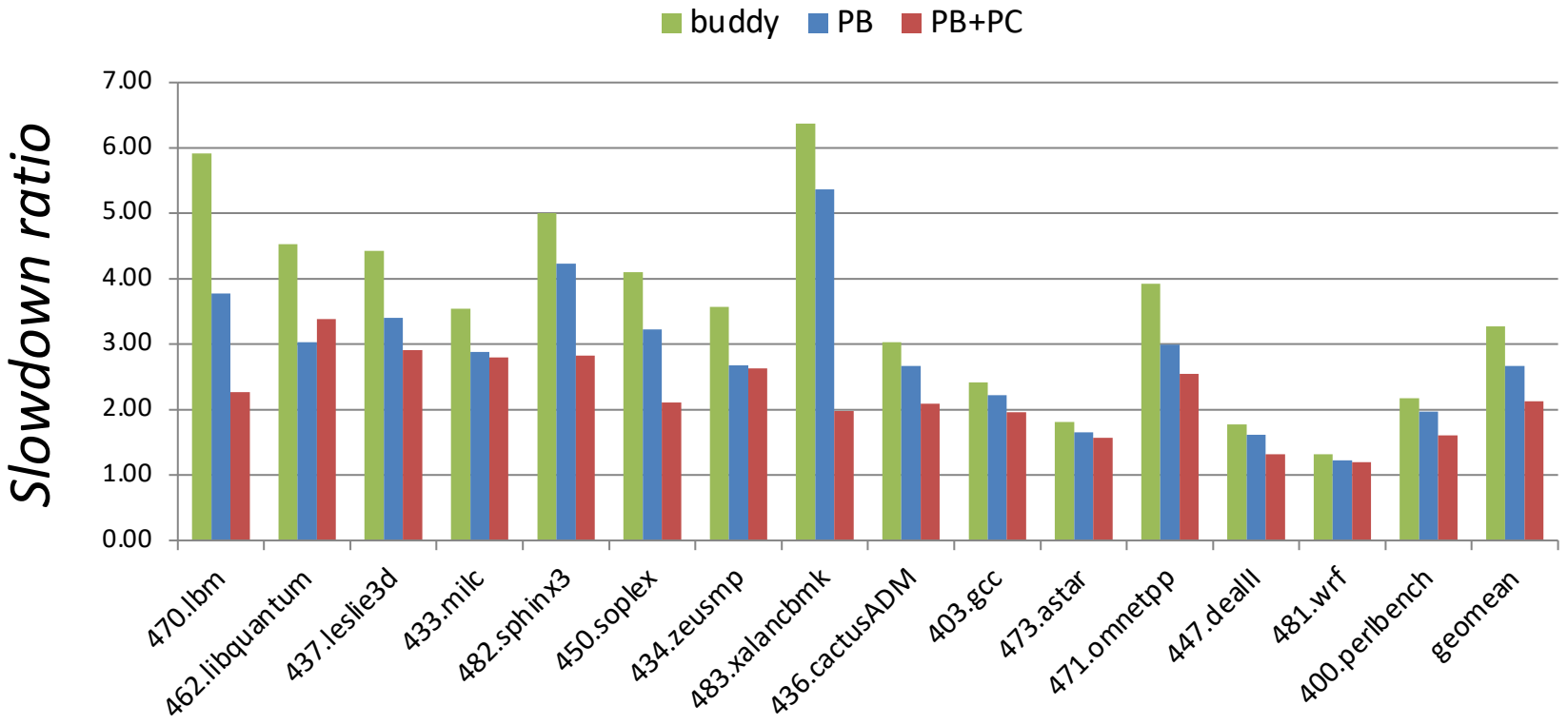
Performance Impact on Unicore



- #of bank vs. performance on a single core
- Finding: bank partitioning negatively affect performance due to reduced MLP, but not significant for most benchmarks

KU THE UNIVERSITY OF KANSAS

Performance Isolation on 4 Cores



- Setup: Core0: X-axis, Core1-3: 470.lbm x 3 (interference)
- PB: DRAM bank partitioning only;
- PB+PC: DRAM bank and Cache partitioning
- Finding: bank (and cache) partitioning improves isolation, but far from ideal

KU THE UNIVERSITY OF KANSAS

Outline

- Motivation
- Background & Problems
- PALLOC: DRAM Bank Allocation Mgmt.
- Evaluation
- **Conclusion**

KU THE UNIVERSITY OF KANSAS

Conclusion

- PALLOC
  - DRAM bank aware kernel level memory allocator
  - **Can eliminate inter-core bank conflicts**
- Findings
  - Private banking improves performance isolation
  - But, far from ideal isolation: *memory bus bottleneck*
- Future work
  - Integration with memory bandwidth control (MemGuard [RTAS'13])
  - Multichannel, NUMA systems.

https://github.com/heechul/palloc

KU THE UNIVERSITY OF KANSAS

Thank you.

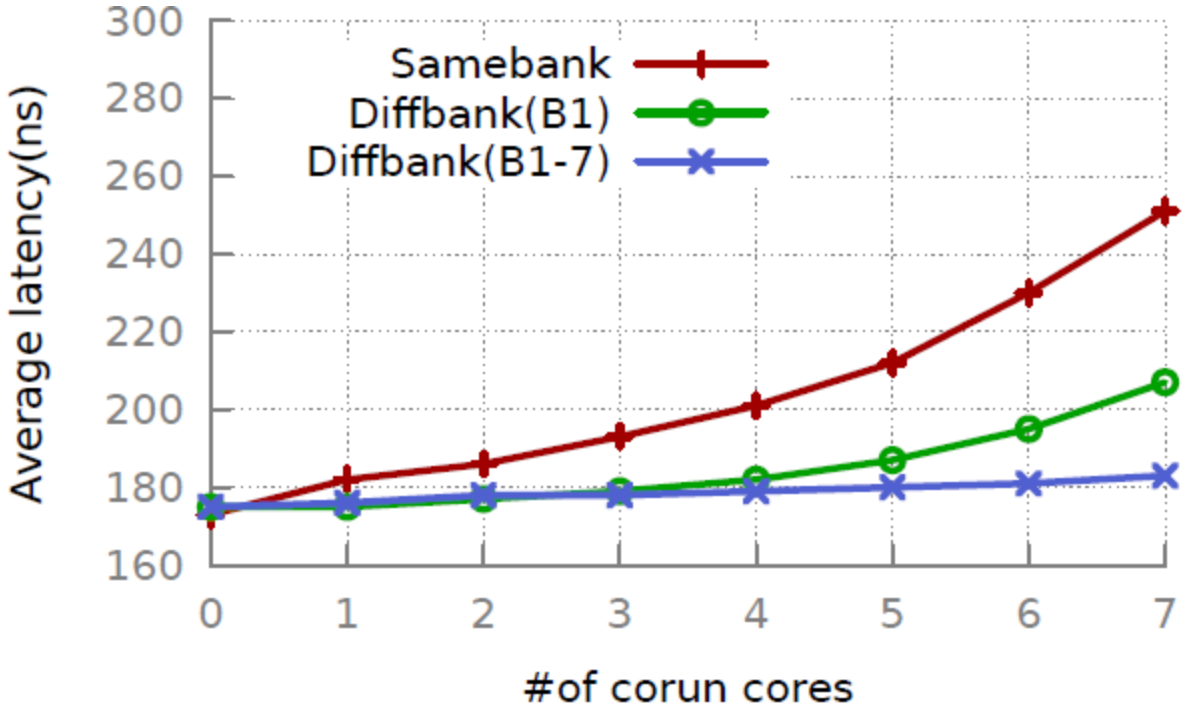
Questions?

Data Intensive Applications

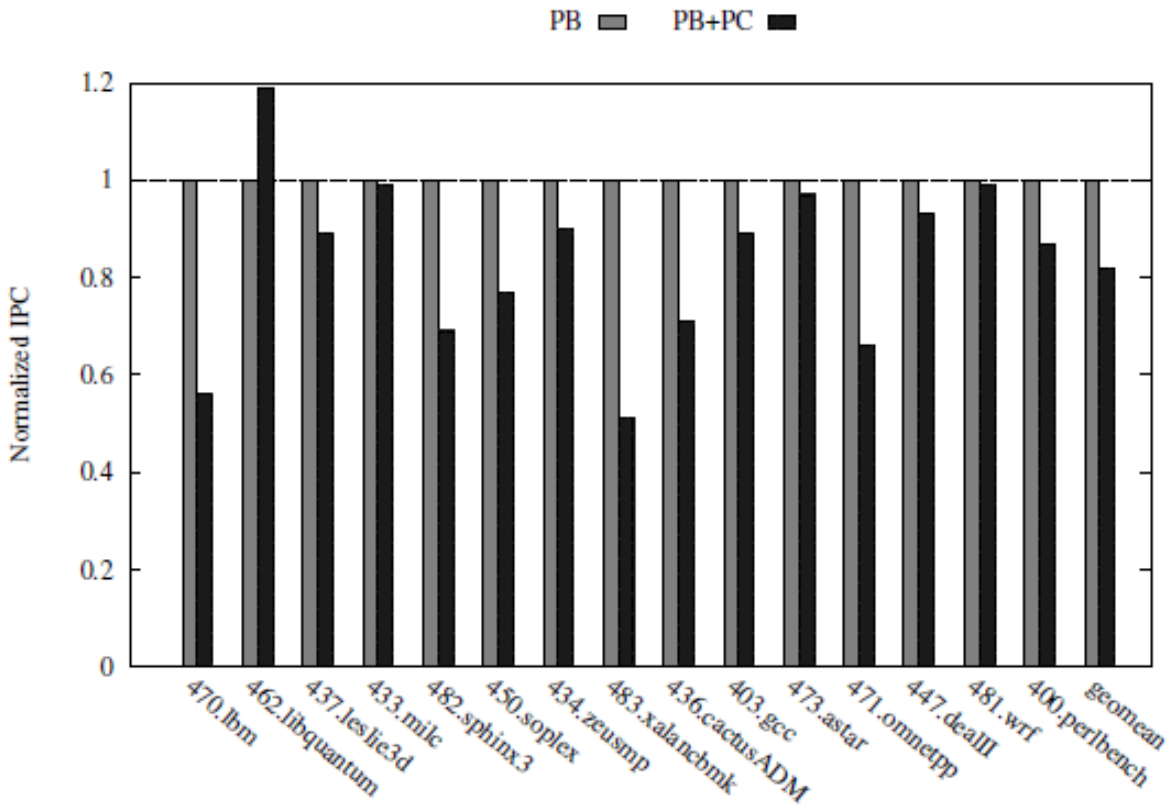
- Multimedia processing, object tracking, game, big data(*), …
- More stress on the memory hierarchy

(*) Source: Intel, **The Growing Importance of Big Data and Real-Time Analytics**, 2012

Samebank vs. Diffbank on P4080



Impact of Cache Partitioning



- PB: private DRAM banking
- PB+PC: private DRAM banking & private cache partitioning