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
Multicore
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

(*) 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

- Memory Controller (MC)
- DRAM DIMM
- Core1, Core2, Core3, Core4
- L3
- Bank 1, Bank 2, Bank 3, Bank 4

Fast

- Peak = 10.6 GB/s
  - DDR3 1333Mhz
Best-case

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

Fast
Most-cases

Mess

• Performance = ??
Worst-case

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

Slow
Background: DRAM Operation

- 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

1 bank & always row misses $\rightarrow$ $\sim 1/10$ 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

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

Flexible allocation policy

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

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

**Eliminate Inter-core bank conflicts**
Identifying Memory Mapping

- 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
/* 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)

```plaintext
# 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
```
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

- Setup: HRT $\rightarrow$ Core0, X-server $\rightarrow$ Core1
- Buddy: no bank control (use all Bank 0-15)
- Diffbank: Core0 $\rightarrow$ Bank0-7, Core1 $\rightarrow$ Bank8-15
### SPEC2006

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

- Use 15 high-medium memory intensive benchmarks
Performance Impact on Unicore

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

• Motivation
• Background & Problems
• PALLOC: DRAM Bank Allocation Mgmt.
• Evaluation
• Conclusion
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](https://github.com/heechul/palloc)
Thank you.

Questions?
Data Intensive Applications

- Multimedia processing, object tracking, game, big data\(^\ast\), ...
- More stress on the memory hierarchy

Samebank vs. Diffbank on P4080
Impact of Cache Partitioning

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