Performance Prediction of Graph Analytics on Persistent Memory

Diego Moura\textsuperscript{a}, Daniel Mossé\textsuperscript{b}, Vinicius Petrucci\textsuperscript{a,b}

\textsuperscript{a}Universidade Federal da Bahia, Brazil
\textsuperscript{b}University of Pittsburgh, USA
Research Context

• For 20 years (1982-2002), DRAM capacity improved by 60% each year *
  ○ After that, rate dropped to 40%
  ○ Now we are around 25%
  ○ Is DRAM technology scaling ending? (not yet, but there is a downward trend)

• Emerging applications are increasingly demanding more memory (Big Data!)
  ○ Even if DRAM could scale, the technology would be prohibitively expensive in cost and power

• How to keep up with such demands in face of the DRAM scaling problem?
  ○ Accept DRAM limitations and wait for some huge improvement? (unlikely!)
  ○ Emerging memory technologies
  ○ Both! Because there is no single technology that can combine cost, capacity and speed

* Computer Organization and Design, David Patterson, John Hennessy, 2018 (Page 86)
Emerging Non-Volatile Memories (NVMs)

- Intel Optane (2019+)
  - high capacity, low cost/GB, low energy

- Not a drop-in replacement for DRAM
  - Compared to DRAM *:
    - 4x more latency for read
    - 10x more latency for store


Figure extracted from "A Close Look at the Intel Micro 3D XPoint Memory, Objective Analysis (2015)"
Intel Optane Technology: Operation Modes

Our focus

Legacy (DRAM)

DRAM

DRAM 1st Level Memory

DRAM + App Direct (AD)

DCPMM
Persistent

DRAM + AD

App Direct (AD)
Persistent programming model with no caching

Memory Mode (MM)

Memory Mode
(2nd Level Memory)
Volatile with DRAM caching

Image from the presentation "Performance Tools for Intel Optane DC Memory" - Kevin O'Leary
Should we allocate the application data to DRAM or PMEM?

**Server**

- **DRAM**
  - Low latency
  - Low capacity
  - High cost

- **PMEM/NVDIMM**
  - High latency
  - High capacity
  - Low cost

**Applications**

- Streaming analytics
- Sentiment analysis
- Machine learning
- Medical imaging processing

Graph Apps on PMEM

dmoura@pitt.edu

March 2021

NVM Workshop
Applications

● Data-intensive Graph Analytics (Berkeley GAP)

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breadth-First Search (BFS)</td>
<td>road (806MB)</td>
</tr>
<tr>
<td>Single-Source Shortest Paths (SSSP)</td>
<td>twitter (12GB)</td>
</tr>
<tr>
<td>PageRank (PR)</td>
<td>web (16GB)</td>
</tr>
<tr>
<td>Connected Components (CC)</td>
<td>kron (17GB)</td>
</tr>
<tr>
<td>Betweenness Centrality (BC)</td>
<td>urand (18GB)</td>
</tr>
</tbody>
</table>

● Need to process large-scale graphs

● Irregular memory access patterns
  ○ Makes it challenging for traditional cache systems and policies
A single PMC is not sufficient to characterize the applications
A single PMC is not sufficient to characterize the applications
A single PMC is not sufficient to characterize the applications.
Multiple PMCs can better estimate performance (DRAM $\rightarrow$ PMEM)

Move to DRAM those with HIGHER perf. slowdown

Move to PMEM those with LOWER perf. slowdown
Usage of the Prediction Model

Performance Estimation for Capacity Planning

Enable Runtime Allocators
Our Proposal

1) Collect PMCs
2) Apply Model

Run on DRAM to collect PMCs

Applications

Sorted List

Lowest Degradation

CPU0
CPU1
CPU2
CPU3

iMc
iMc

DRAM
PMEM

Highest Degradation

N
...
4
3
2
1
Preliminary Results

Performance gains of **22% (avg)** and **37% (max)**

Reduction of **120% (avg)** and **244% (max)** in performance degradation (as experienced by the applications)
Results: Detailed Performance Degradation

- FIFO simulation of **1000 runs** with varying seeds
- FIFO maps **3-16 apps** to PMEM (avg 9.44 apps)
- FIFO degrades **2.2x (avg)** more than our approach
- FIFO had **13X (avg)** more degradation than DRAM execution
- Our approach maps **10 apps** to PMEM (same sorted queue)
Next Steps

- Scheduling of applications evaluated using simulations (though prediction model based on real experiments)
  
  **Make the experiment completely real**

- Assumption of knowledge of the applications
  
  **No prior-knowledge assumption and migration allowance**

- Beyond application-level memory allocation
  
  **Remove whole-application assumption and allow for smaller granularity, such as objects or pages**
Acknowledgements

- Thanks Intel for partially supporting this research

*Intel, the Intel logo, and Intel Optane are trademarks of Intel Corporation or its subsidiaries*
Performance Prediction of Graph Analytics on Persistent Memory

Diego Moura
Ph.D. Student
diegobnh@gmail.com