DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

P&S Processing-in-Memory
16.03.2023

Geraldo F. Oliveira
Juan Gómez-Luna        Lois Orosa           Saugata Ghose
Nandita Vijaykumar     Ivan Fernandez       Mohammad Sadrosadati
Onur Mutlu

SAFARI

ETH Zürich  \hspace{1cm}  UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN  \hspace{1cm}  UNIVERSITY OF TORONTO  \hspace{1cm}  UNIVERSIDAD DE MÁLAGA
Executive Summary

• **Problem:** Data movement is a major bottleneck in modern systems. However, it is **unclear** how to identify:
  - different sources of data movement bottlenecks
  - the most suitable mitigation technique (e.g., caching, prefetching, near-data processing) for a given data movement bottleneck

• **Goals:**
  1. Design a methodology to **identify** sources of data movement bottlenecks
  2. Compare compute- and memory-centric data movement mitigation techniques

• **Key Approach:** Perform a large-scale application characterization to identify **key metrics** that reveal the sources to data movement bottlenecks

• **Key Contributions:**
  - Experimental characterization of 77K functions across 345 applications
  - A methodology to characterize applications based on data movement bottlenecks and their relation with different data movement mitigation techniques
  - DAMOV: a **benchmark suite** with 144 functions for data movement studies
  - Four case-studies to highlight DAMOV’s applicability to open research problems
Outline

1. Data Movement Bottlenecks
2. Methodology Overview
3. Application Profiling
4. Locality-Based Clustering
5. Memory Bottleneck Analysis
6. Case Studies
Data movement bottlenecks happen because of:
- Not enough data **locality** → ineffective use of the cache hierarchy
- Not enough **memory bandwidth**
- High average **memory access time**
Data Movement Bottlenecks (2/2)

- **Compute-Centric Architecture**
- **Memory-Centric Architecture**
- **Near-Data Processing (NDP)**

- Abundant DRAM bandwidth
- Shorter average memory access time

SAFARI
Near-Data Processing (1/2)

The goal of Near-Data Processing (NDP) is to mitigate data movement.

- Abundant DRAM bandwidth
- Shorter average memory access time

SAFARI
Identifying Memory Bottlenecks

• Multiple approaches to identify applications that:
  - suffer from data movement bottlenecks
  - take advantage of NDP

• Existing approaches are not comprehensive enough
The Problem

• Multiple approaches to identify applications that:
  - suffer from data movement bottlenecks
  - take advantage of NDP

No available methodology can comprehensively:

  - **identify** data movement bottlenecks
  - **correlate** them with the **most suitable** data movement mitigation mechanism
Our Goal

- **Our Goal:** develop a methodology to:
  - methodically identify sources of data movement bottlenecks
  - comprehensively compare compute- and memory-centric data movement mitigation techniques
Key Approach

• New **workload characterization methodology** to analyze:
  - data movement bottlenecks
  - suitability of different data movement mitigation mechanisms

• Two main profiling strategies:

  **Architecture-independent profiling:**
  characterizes the memory behavior independently of the underlying hardware

  **Architecture-dependent profiling:**
  evaluates the **impact of the system configuration** on the memory behavior
Methodology Overview

Step 1: Application Profiling
- User Input: Source Code, Target Application
- Step 1: Application Profiling
  - Profiler
  - roi_begin, roi_end

DAMOV-SIM Simulator
- Memory Traces
- Scalability Analysis
- # Cores

Step 2: Locality-based Clustering
- DRAM Bandwidth
- DRAM Latency
- L1/L2 Cache Capacity
- L3 Cache Contention
- Compute-Bound

Step 3: Memory Bottleneck Class
- Memory Bottleneck Classes
- Methodology Output
- SAFARI

Methodology Output
- Step 2: Locality-based Clustering
- Step 3: Memory Bottleneck Class
Step 1: Application Profiling

**Goal:** Identify *application functions* that suffer from *data movement bottlenecks*

Hardware Profiling Tool: Intel VTune

**MemoryBound:** CPU is stalled due to load_Store
Step 2: Locality-Based Clustering

- **Goal:** analyze application’s memory characteristics

**Spatial Locality**

Memory Trace

```
0 1 2 3 4 5
```

stride profile(1)+= 1

- **Low spatial locality**

- **High spatial locality**

[7] Weinberg+, “Quantifying Locality in the Memory Access Patterns of HPC Applications,” SC, 2005
Step 2: Locality-Based Clustering

• **Goal:** analyze application’s memory characteristics

**Spatial Locality**

Memory Trace

![Stride Profile Histogram](image)

- Low spatial locality
- High spatial locality

**Temporal Locality**

Memory Trace

![Reuse Profile Histogram](image)

- Low temporal locality
- High temporal locality

---

[7] Weinberg+, “Quantifying Locality in the Memory Access Patterns of HPC Applications,” SC, 2005
**Step 3: Memory Bottleneck Classification (1/2)**

**Arithmetic Intensity (AI)**
- floating-point/arithmetic operations per L1 cache lines accessed
  → shows **computational intensity** per memory request

**LLC Misses-per-Kilo-Instructions (MPKI)**
- LLC misses per one thousand instructions
  → shows **memory intensity**

**Last-to-First Miss Ratio (LFMR)**
- LLC misses per L1 misses
  → shows if an application **benefits from L2/L3 caches**
Step 3: Memory Bottleneck Classification (2/2)

- **Goal:** identify the specific sources of data movement bottlenecks

  DAMOV-SIM Simulator

  Scalability Analysis

  Integrated ZSim and Ramulator

  DAMOV-SIM: https://github.com/CMU-SAFARI/DAMOV

- **Scalability Analysis:**
  - 1, 4, 16, 64, and 256 out-of-order/in-order host and NDP CPU cores
  - 3D-stacked memory as main memory
Step 3: Memory Bottleneck Analysis

Temporal Locality

- **High**
  - LFMR
  - **Low**
    - Decreasing
      - MPKI
      - **Low**
        - AI
        - **Low**
          - 1a: DRAM Bandwidth
          - 1b: DRAM Latency
          - 1c: L1/L2 Cache Capacity
  - **High**
    - Increasing
      - MPKI
      - **Low**
        - AI
        - **Low**
          - 2a: L3 Cache Contention
          - 2b: L1 Cache Capacity
          - 2c: Compute-Bound

SAFARI
Step 3: Memory Bottleneck Analysis

Six classes of data movement bottlenecks:

- 1a: DRAM Bandwidth
- 1b: DRAM Latency
- 1c: L1/L2 Cache Capacity
- 2a: L3 Cache Contention
- 2b: L1 Cache Capacity
- 2c: Compute-Bound

Each class ↔ data movement mitigation mechanism
Class 1a: DRAM Bandwidth Bound (1/2)

- High MPKI → high memory pressure
- Host scales well until bandwidth saturates
- NDP scales **without saturating** alongside attained bandwidth

**Host**

| Number of Cores | Performance | Bandwidth (GB/s) |
|-----------------|-------------|------------------|
| 1               |             |                  |
| 4               |             |                  |
| 16              |             |                  |
| 64              |             |                  |
| 256             |             |                  |

**NDP**

| Number of Cores | Performance | Bandwidth (GB/s) |
|-----------------|-------------|------------------|
| 1               |             |                  |
| 4               |             |                  |
| 16              |             |                  |
| 64              |             |                  |
| 256             |             |                  |

**DRAM bandwidth bound applications:**
NDP does better because of the higher internal DRAM bandwidth
Class 1a: DRAM Bandwidth Bound (2/2)

- High LFMR → **L2 and L3 caches are inefficient**
- Host’s energy consumption is dominated by cache look-ups and off-chip data transfers
- NDP provides **large system energy reduction** since it does not access L2, L3, and off-chip links

**DRAM bandwidth bound applications:**
NDP does better because it eliminates off-chip I/O traffic
Step 3: Memory Bottleneck Analysis

Temporal Locality

- High
- Decreasing MPKI
- Low
- Low LFMR

- Low
- Increasing MPKI
- High
- High LFMR

1a: DRAM Bandwidth
1b: DRAM Latency
1c: L1/L2 Cache Capacity

2a: L3 Cache Contention
2b: L1 Cache Capacity
2c: Compute-Bound
Step 3: Memory Bottleneck Analysis

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

GERALDO F. OLIVEIRA¹, JUAN GÓMEZ-LUNA¹, LOIS OROSA¹, SAUGATA GHOSE², NANDITA VIJAYKUMAR³, IVAN FERNANDEZ¹,⁴, MOHAMMAD SADROSADATI¹, and ONUR MUTLU¹

¹ETH Zürich, Switzerland
²University of Illinois Urbana-Champaign, USA
³University of Toronto, Canada
⁴University of Malaga, Spain

Corresponding author: Geraldo F. Oliveira (e-mail: geraldod@inf.ethz.ch).
More in the Paper

• Methodology validation

• Effect of the last-level cache size
  - Large L3 cache size (e.g., 512 MB) can mitigate some cache contention issues

• Summary of our workload characterization methodology
  - Including workload characterization using in-order host/NDP cores

• Limitations of our methodology

• Benchmark diversity
1. Data Movement Bottlenecks
2. Methodology Overview
3. Application Profiling
4. Locality-Based Clustering
5. Memory Bottleneck Analysis
6. Case Studies
Case Studies

• Many open questions related to NDP system designs⁸:
  - Interconnects
  - Data mapping and allocation
  - NDP core design (accelerators, general-purpose cores)
  - Offloading granularity
  - Programmability
  - Coherence
  - System integration
  - ...

• Goal: demonstrate how DAMOV is useful to study NDP system designs

[⁸] Mutlu+, “A Modern Primer on Processing in Memory,” Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann, 2021
Load Balance and Inter-Vault Communication on NDP

portion of the memory requests an NDP core issues go to remote vaults → increases the memory access latency for the NDP core

NDP Accelerators and Our Methodology

NDP accelerator is faster than compute-centric accelerator for Class 1a and 1b applications; slower for Class 2c → key observations hold for other NDP architectures

Different Core Models on NDP Architectures

using in-order cores limits performance of some applications → static instruction scheduling cannot exploit memory parallelism

Fine-Grained NDP Offloading

few basic blocks are responsible for most of LLC misses → offloading such basic blocks to NDP are enough to improve performance
Case Studies

Load Balance and Inter-Vault Communication on NDP
portion of the memory requests an NDP core issues go to remote vaults → increases the memory access latency for the NDP core.

NDP Accelerators and Our Methodology
NDP accelerator is faster than compute-centric accelerator for Class 1a and 1b applications; slower for Class 2c → key observations hold for other NDP architectures.

Different Core Models on NDP Architectures
using in-order cores limits performance of some applications → static instruction scheduling cannot exploit memory parallelism.

Fine-Grained NDP Offloading
few basic blocks are responsible for most of LLC misses → offloading such basic blocks to NDP are enough to improve performance.

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

Geraldo F. Oliveira¹, Juan Gómez-Luna¹, Lois Orosa¹, Saugata Ghose², Nandita Vijaykumar³, Ivan Fernandez¹, ⁴, Mohammad Sadrosadati¹, and Onur Mutlu¹

¹ETH Zürich, Switzerland
²University of Illinois Urbana-Champaign, USA
³University of Toronto, Canada
⁴University of Malaga, Spain

Corresponding author: Geraldo F. Oliveira (e-mail: geraldo@inf.ethz.ch).
DAMOV is Open-Source

• We open-source our benchmark suite and our toolchain

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing. Described by Oliveira et al. (preliminary version at https://arxiv.org/pdf/2105.03725.pdf)

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing.

The DAMOV benchmark suite is the first open-source benchmark suite for main memory data movement-related studies, based on our systematic characterization methodology. This suite consists of 144 functions representing different sources of data movement bottlenecks and can be used as a baseline benchmark set for future data-movement mitigation research. The applications in the DAMOV benchmark suite belong to popular benchmark suites, including BWA, Chai, Darknet, GASE, Hardware Effects, Hashjoin, HPCC, HPCG, Ligra, PARSEC, Parboil, PolyBench, Phoenix, Rodinia, SPLASH-2, STREAM.
DAMOV is Open-Source

• We open-source our benchmark suite and our toolchain

Get DAMOV at:

https://github.com/CMU-SAFARI/DAMOV

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

DAMOV is a benchmark suite and a methodical framework targeting the study of data movement bottlenecks in modern applications. It is intended to study new architectures, such as near-data processing.

The DAMOV benchmark suite is the first open-source benchmark suite for main memory data movement-related studies, based on our systematic characterization methodology. This suite consists of 144 functions representing different sources of data movement bottlenecks and can be used as a baseline benchmark set for future data-movement mitigation research. The applications in the DAMOV benchmark suite belong to popular benchmark suites, including BWA, Chai, Darknet, GASE, Hardware Effects, Hashjoin, HPCC, HPCG, Ligra, PARSEC, Parboil, PolyBench, Phoenix, Rodinia, SPLASH-2, STREAM.
• **Problem:** Data movement is a major bottleneck in modern systems. However, it is **unclear** how to identify:
  - **different sources** of data movement bottlenecks
  - the **most suitable** mitigation technique (e.g., caching, prefetching, near-data processing) for a given data movement bottleneck

• **Goals:**
  1. Design a methodology to **identify** sources of data movement bottlenecks
  2. Compare compute- and memory-centric data movement mitigation techniques

• **Key Approach:** Perform a large-scale application characterization to identify **key metrics** that reveal the sources to data movement bottlenecks

• **Key Contributions:**
  - Experimental characterization of 77K functions across 345 applications
  - A methodology to characterize applications based on data movement bottlenecks and their relation with different data movement mitigation techniques
  - **DAMOV:** a benchmark suite with **144 functions** for data movement studies
  - Four case-studies to highlight DAMOV’s applicability to open research problems

**DAMOV:** [https://github.com/CMU-SAFARI/DAMOV](https://github.com/CMU-SAFARI/DAMOV)
Longer Lecture

DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

P&S Processing-in-Memory
18.10.2022

Geraldo F. Oliveira
Juan Gómez-Luna  Lois Orosa  Saugata Ghose
Nandita Vijaykumar  Ivan Fernandez  Mohammad Sadrosadati
Onur Mutlu

PIM Course: Lecture 2: How to Evaluate Data Movement Bottlenecks - Fall 2022

SaRAF

https://youtu.be/An2lACOASdo
Executive Summary

• **Problem:** Data movement is a major bottleneck in modern systems. However, it is **unclear** how to identify:
  - different sources of data movement bottlenecks
  - the **most suitable** mitigation technique (e.g., caching, prefetching, near-data processing) for a given data movement bottleneck

• **Goals:**
  1. Design a methodology to **identify** sources of data movement bottlenecks
  2. Compare compute- and memory-centric data movement mitigation techniques

• **Key Approach:** Perform a large-scale application characterization to identify **key metrics** that reveal the sources to data movement bottlenecks

• **Key Contributions:**
  - Experimental characterization of 77K functions across 345 applications
  - A **methodology** to characterize applications based on data movement bottlenecks and their relation with different data movement mitigation techniques

[SAFARI Live Seminar: DAMOV: A New Methodology & Benchmark Suite for Data Movement Bottlenecks](https://www.youtube.com/live/GWideVyo0nM?feature=share)
DAMOV: A New Methodology and Benchmark Suite for Evaluating Data Movement Bottlenecks

P&S Processing-in-Memory
16.03.2023

Geraldo F. Oliveira
Juan Gómez-Luna     Lois Orosa     Saugata Ghose
Nandita Vijaykumar     Ivan Fernandez     Mohammad Sadrosadati
Onur Mutlu

SAFARI

ETH Zürich    UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN    UNIVERSITY OF TORONTO