A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory

Nandita Vijaykumar
Abhilasha Jain, Diptesh Majumdar, Kevin Hsieh, Gennady Pekhimenko
Eiman Ebrahimi, Nastaran Hajinazar, Phillip B. Gibbons, Onur Mutlu

Carnegie Mellon University
University of Toronto
SFU
NVIDIA
ETH Zürich
Today’s cross-layer abstractions are designed to primarily convey functionality, not to aid performance optimization.
The narrow abstractions lose key information

Data Structures

Code Optimizations

Access Patterns

Integer
Char
Float

Data Type

Instructions
Memory Addresses

100011111...
101010011...
Consequence 1: The Hardware Approach
We design hardware to infer and predict program behavior to optimize for performance
Consequence 2: The Software Approach
Software is tuned to the specifics of hardware architecture when optimizing for performance.

- Programmability
- Portability

Hardware optimizations?
Cache space available?
How many DRAM banks?
With growing HW/SW sophistication, traditional interfaces limit optimization effectiveness

Time for a richer interface between hardware and software?
A fresh approach to traditional optimizations

- Cache Management
- Data Placement in DRAM
- Data Compression
- Approximation
- DRAM Cache Management
- NVM Management
- NUCA/NUMA Optimizations

✓ Programmability
✓ Portability
✓ Resource Efficiency
A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory

Nandita Vijaykumar
Abhilasha Jain, Diptesh Majumdar, Kevin Hsieh, Gennady Pekhimenko
Eiman Ebrahimi, Nastaran Hajinazar, Phillip B. Gibbons, Onur Mutlu

Carnegie Mellon University
UNIVERSITY OF TORONTO
SFU
NVIDIA
ETH Zürich