Reevaluating Memory Systems for Energy Efficiency: Bringing Computation and Data Closer

Grant Ayers, Mingyu Gao, Christos Kozyrakis
Stanford University

Motivation

System View

Memory systems must be redesigned because:

- Voltage scaling has ceased, making systems energy-bound.
- Energy to move data dominates energy for computation.
- Main memory can constitute 45% of total system power consumption.

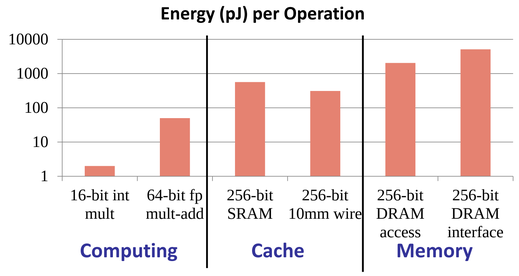

Figure 1: Compute and Transfer Energy

Application View

High energy and low performance are characteristic for emerging applications with low locality.

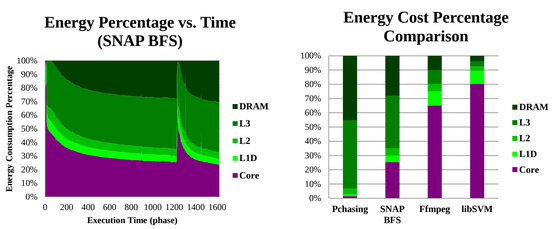

Figure 2: Workload Memory Behaviors

Reducing data movement can help improve both energy and performance.

- Less inter-chip and long wire transfers
- Lower latency for data accesses

Technical Approach

Bring Computation and Data Closer

1. Move computation to the memory when beneficial.
2. Maximize data transfer efficiency for everything else.

Opportunities

- 3D CPU & memory integration.
- Logic layer with DRAM stack (HMC).
- Small logic chips bonded on top of DRAM.

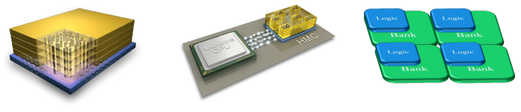

Figure 3: Logic Layer Stacking, HMC, and Bonding

Challenges

1. When does it make sense to move computation to memory?
2. What type of logic is required and how should it be integrated?
3. System issues: Must address DRAM structure, ISA, virtual memory, OS, and user space.
4. Management: What parts of the system oversee this?

Methodology

1. Identify common and important workloads
   - Especially workloads which are memory-bound or show low locality
2. Characterize workload performance
   - Use source analysis, performance counters, and simulation
3. Model and build the new architecture

Workloads & Constraints

Characterize Workloads

Name	Processing Complexity	Parallel Capable	Dataset Size per LU	Data Structure	Locality	Read/Write
MapReduce	Variable	Yes	GB	Varied	Varied	Varied
VC Graph	Low	Yes	MB	Nodes	No	Read
Graph BFS	Low	Yes	MB	Nodes	No	Read
Deduplication	Low	Yes	GB	No	Spatial	Read
K-means	Medium	Yes	MB~GB	Vectors	Temporal	Read
Collaborative Filtering	High	Yes	10~100 MB	Matrices	Spatial	R & W
Convolutional DBN	High	Difficult	MB	Arrays	Spatial	R & W
Merge Sort	Low	Yes	kB~MB	Varied	Spatial	R & W
Inverted File Indexing	Low	Yes	GB	No	Spatial	Read

Table 1: Workloads Characteristics

- Data-centric operations
  - Address Translate/Remap
    - ✓ Move, copy, remap
  - Find/Gather
    - ✓ Traverse, pointer chase
  - Communicate
- Computation-centric operations
  - General arithmetic and logical operations
  - Matrix and vector computation

Power and Area Budgets

Current technique can support

- Power: at least 11W or even 50W
- Area: a few hundreds of mm^2

That means

- 1x high-performance Ivy Bridge core
- 22x ARM A6 2-core
- 343x ARM Cortex-M3

Combine Software and Hardware

1. Most workloads ask for parallel capability
2. High-performance CPU is neither available nor necessary

Data Processing Unit (DPU)

- *Designed for data-centric operations*
- *Small and replicable for high parallelism*

Modeling

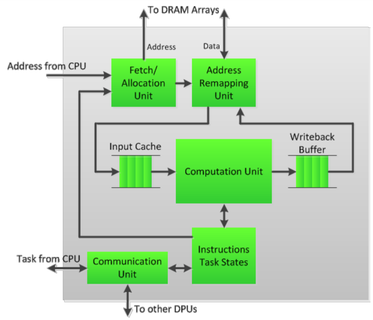

Figure 4: Data Processing Unit

Design for Data-centric Modules

Remap Unit

- Virtual address translation
- Smart address remapping

Fetch Unit

- Independent: sequential, strided, hash
- Dependent: pointer chase, inverted index

Communication Unit

- Task exchange and Data exchange
- MapReduce and Gather-Apply-Scatter patterns

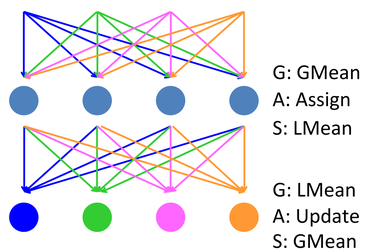

Figure 5: G-A-S Pattern for K-means Algorithm

Future Work

1. Improve the model with a wider range of workloads, including image processing, big data, and machine learning
2. Design DPU in detail according to the workload-derived model
3. Evaluate the DPU architecture