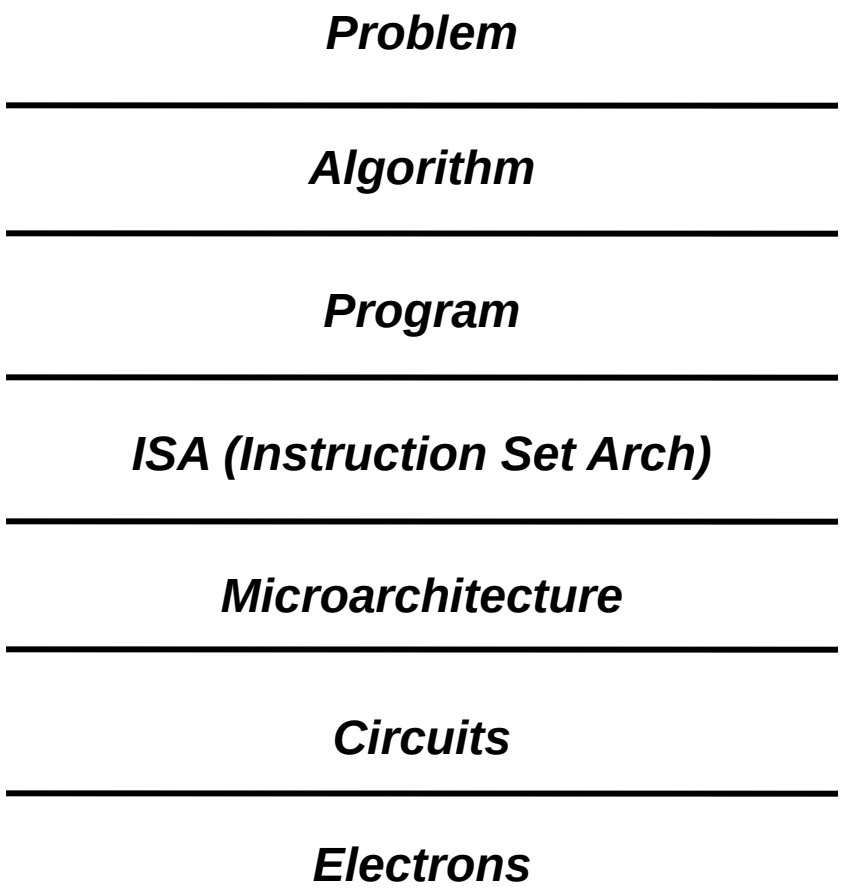
Rethinking Computer Architecture

Wen-mei Hwu
University of Illinois, Urbana-Champaign

Celebrating Yale@75
September 19, 2014

What Yale and I debate about in Samos and other places.



Application developers should not deal with variations in HW

Patt and Patel, Introduction to Computer Systems: from Bits and Gates to C and Beyond

The HPS Vision - 1985

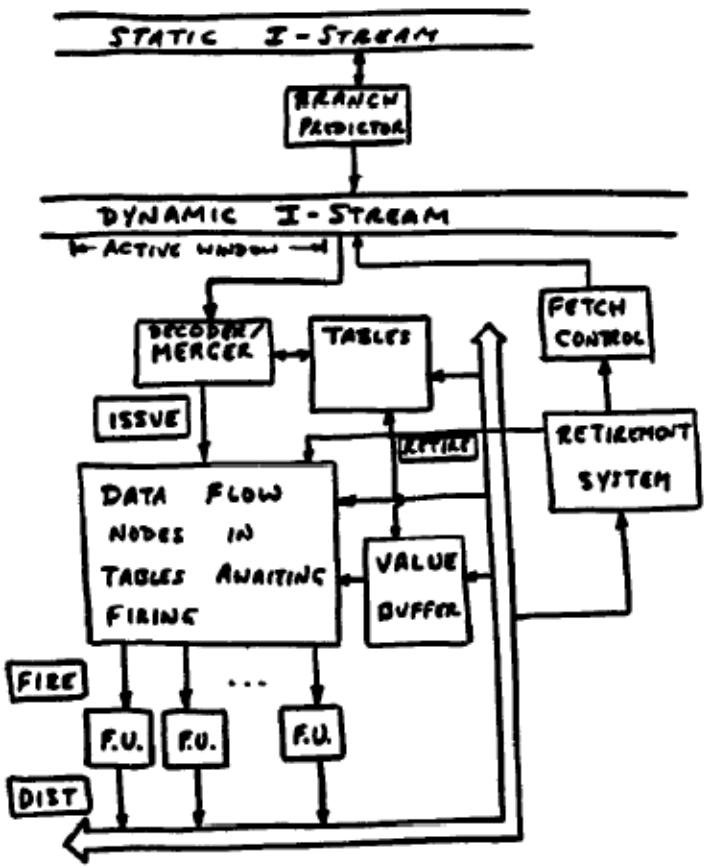


FIGURE 1.

- One static program (algorithm)
- Many execution resource configurations
  - Types of Function Units
  - Number of Function Units
  - I-Fetch bandwidth
  - Memory Latencies
- Key enablers
  - Branch prediction
  - Resource mapping
  - Restricted data flow execution
  - Sequential retirement

data flow graph for the entire program is in the machi
at one time. We define the active window as the set
ISP instructions whose corresponding data flow nodes

Patt, Hwu, Shebanow, “HPS, A New Microarchitecture: Rationale and Initial Results

Some Lessons Learned

- Parallelism and communication costs motivate algorithm changes
  - Locality vs. parallelism tradeoffs in libraries
- Performance and efficiency pressure breaks abstraction
  - Java is great for abstraction portability but insufficient for performance and efficiency
  - MPI, OpenMP apps often explicitly handle hardware-centric details

Productivity and Performance

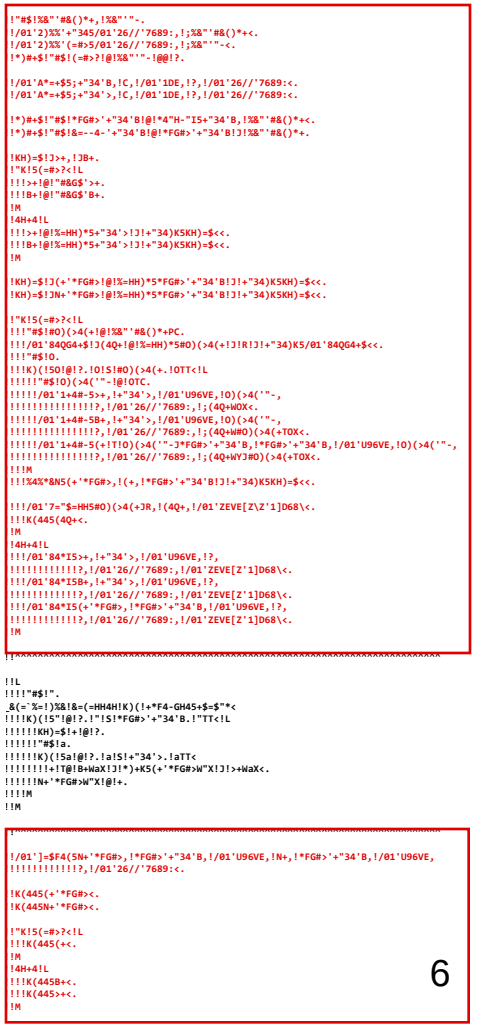
Triolet

```
ys = [sum(x * cos(r*k) for (x, k) in zip(xs, ks))
      for r in par(rs)]
```

- Library functions factor out data decomposition, parallelism, and communication

128-way Speedup (*16 cores × 8 nodes*)	**Triolet**	**C with MPI+OpenMP**
	99	115

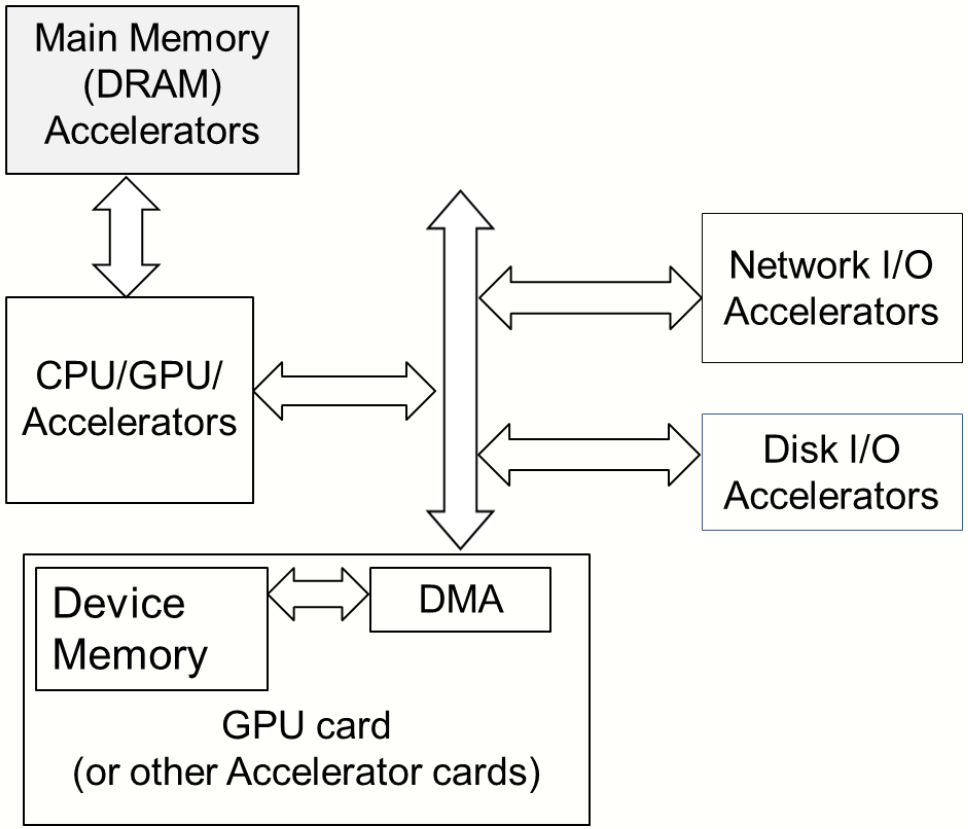
C with MPI+OpenMP



Setup

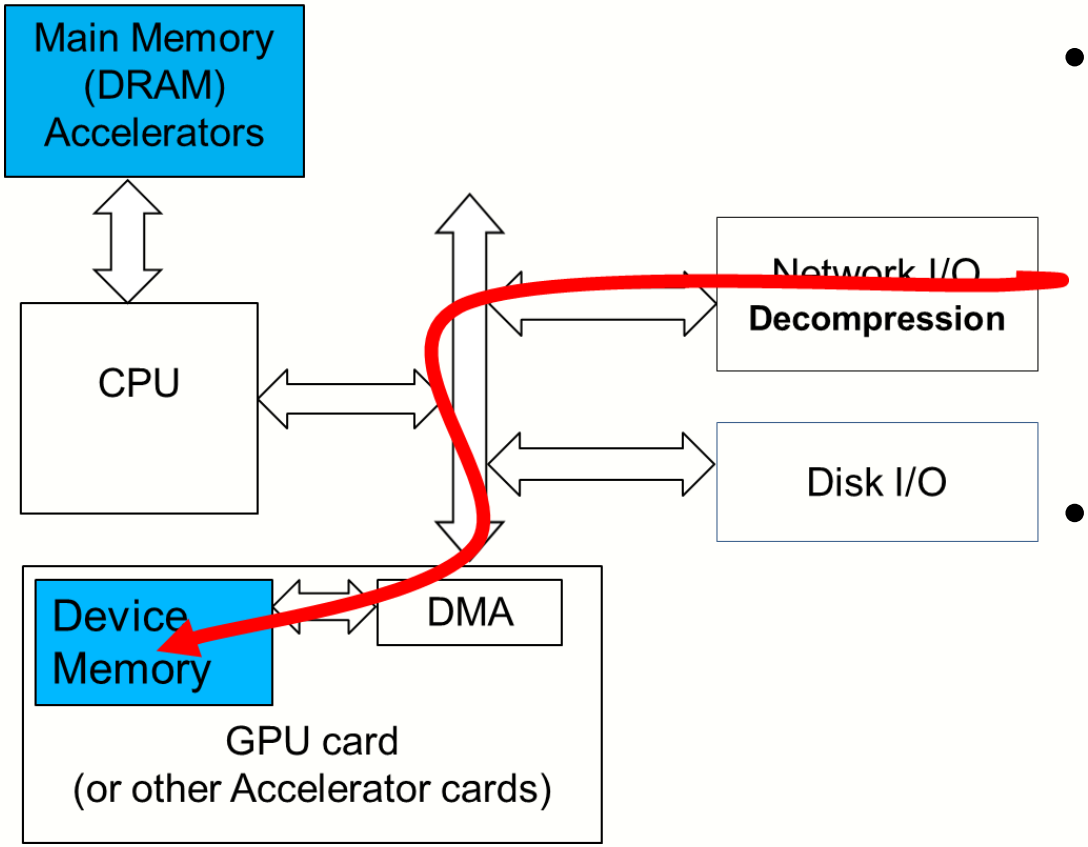
Cleanup

Trends in System Design – 2014



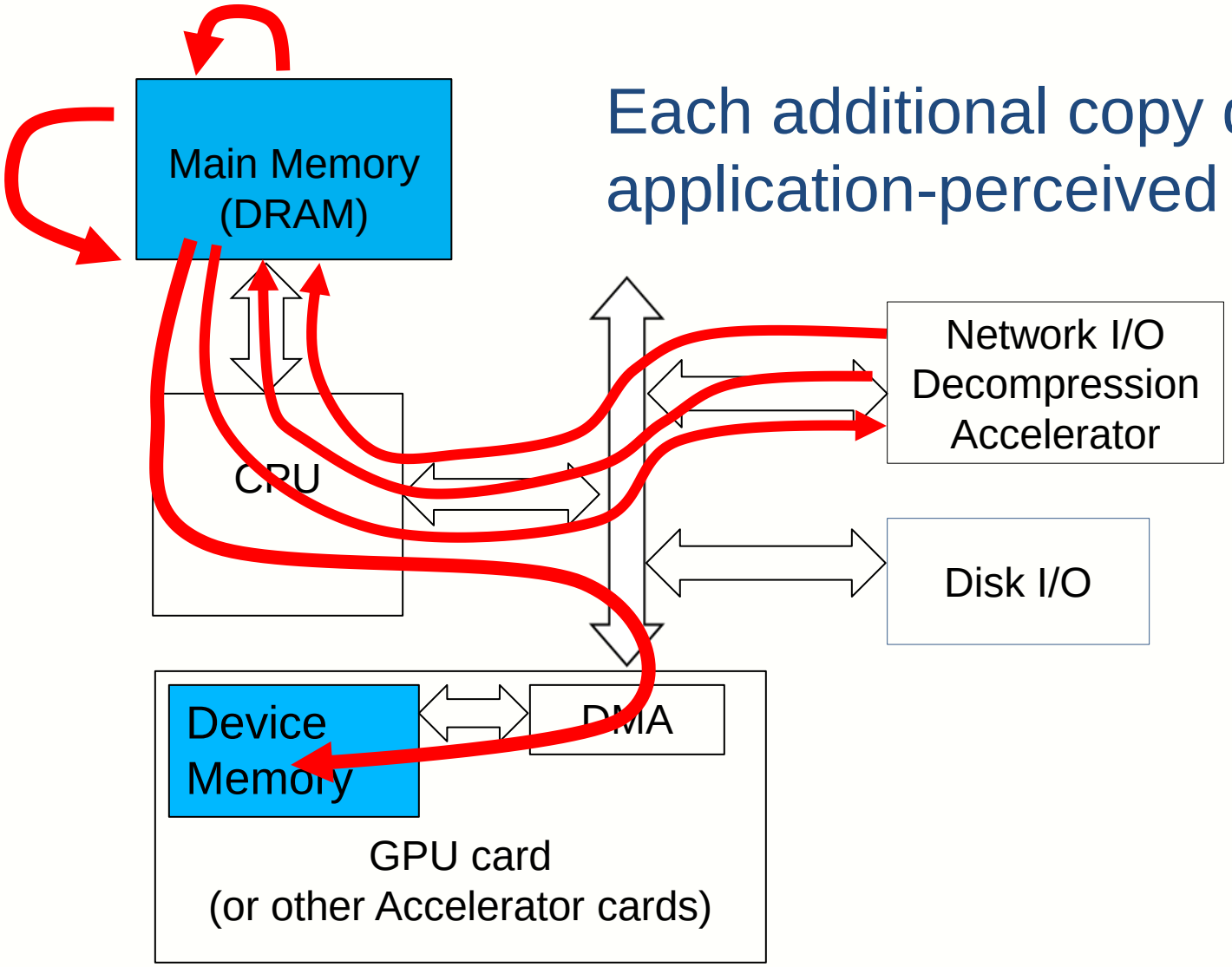
- CPUs/GPUs/Accelerators or entire nodes are the new function units
- Compute functions are the new instructions
- Distributed execution of functions to avoid data movement
  - Accelerators in/near Network I/O, Disk I/O, DRAM
  - Some come with own DRAM/SRAM for bandwidth

Example - Desirable Data Transfer and Compute Behavior



- Runtime/OS should map buffers and compute functions
  - I/O buffer to any major DRAM/SRAM
- Compute functions (decompression) to any CPU/GPU/accelerators

Example -Today's Data Transfer and Compute Behavior



Each additional copy diminishes application-perceived bandwidth

A Call to Action

- Redefine system architecture
  - HAS/CUDA 6.0 a step in the right direction
- Redefine ISA binary standard
  - SPIR/HSAIL/PTX with finalizers a step in the right direction
- Redesign OS/Runtime for data and compute mapping
  - UNIX/Linux overdue for redesign
- Provide performance portable domain libraries to sustain abstraction
  - High-level mechanisms such as Triolet and Tangram to fuse and tune library code into apps

Congratulations, Yale!