Design Space Analysis for Heterogeneous Systems

Wenhao Jia (Princeton), Tae Jun Ham (Princeton), Kelly A. Shaw (Univ of Richmond), Margaret Martonosi (Princeton)

The Emergence of Heterogeneous Systems

- Increasingly demanding power/performance goals require designers to utilize heterogeneous components
  - GPUs offer high performance-per-watt, but they are difficult to design
  - Accelerators can substantially improve **streaming** application performance
- The problem: Heterogeneous systems are difficult to design and optimize
  - Must account for computation AND communication
  - Must account for performance AND power
  - Existing automated design space exploration approaches often cannot handle real-system variance and subspace-induced nonlinearity

Our Work

- Analyze Existing Systems
  - Power/performance analysis of heterogeneous systems
  - Real-system measurements and simulation
- Optimize Mappings onto GPUs
  - Statistical and machine learning-based design space analysis techniques
  - Also, compile-time analysis to prune program design parameters
- [Newer] Analyze and Design Mappings onto Accelerators
  - Key focus: Plan *Communication Accelerators* to pair with *Computation Accelerators* in a balanced manner

Approach 1: Starchart

(Publication) Starchart: Hardware and Software Optimization Using Recursive Partitioning Regression Trees, Wenhao Jia et al., Parallel Architectures and Compilation Techniques (PACT) 2013

Motivation

- GPU design spaces contain complex "*performance cliffs*" and "*subspaces*"
- Existing design space exploration approaches are insufficient

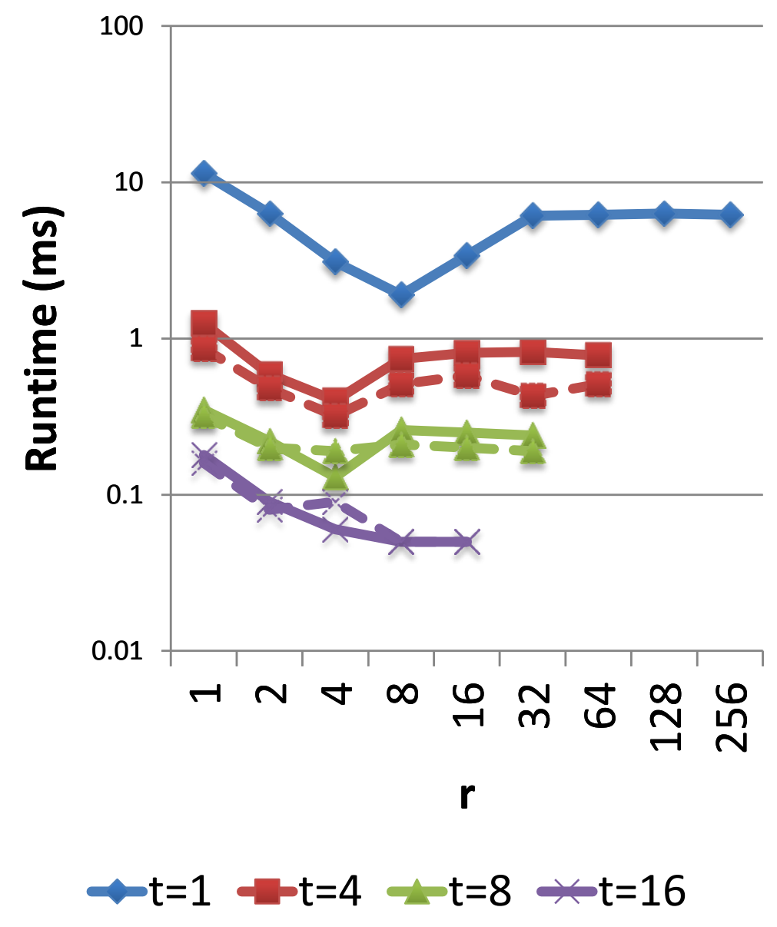
Our Work: Automated Design Space Partitioning

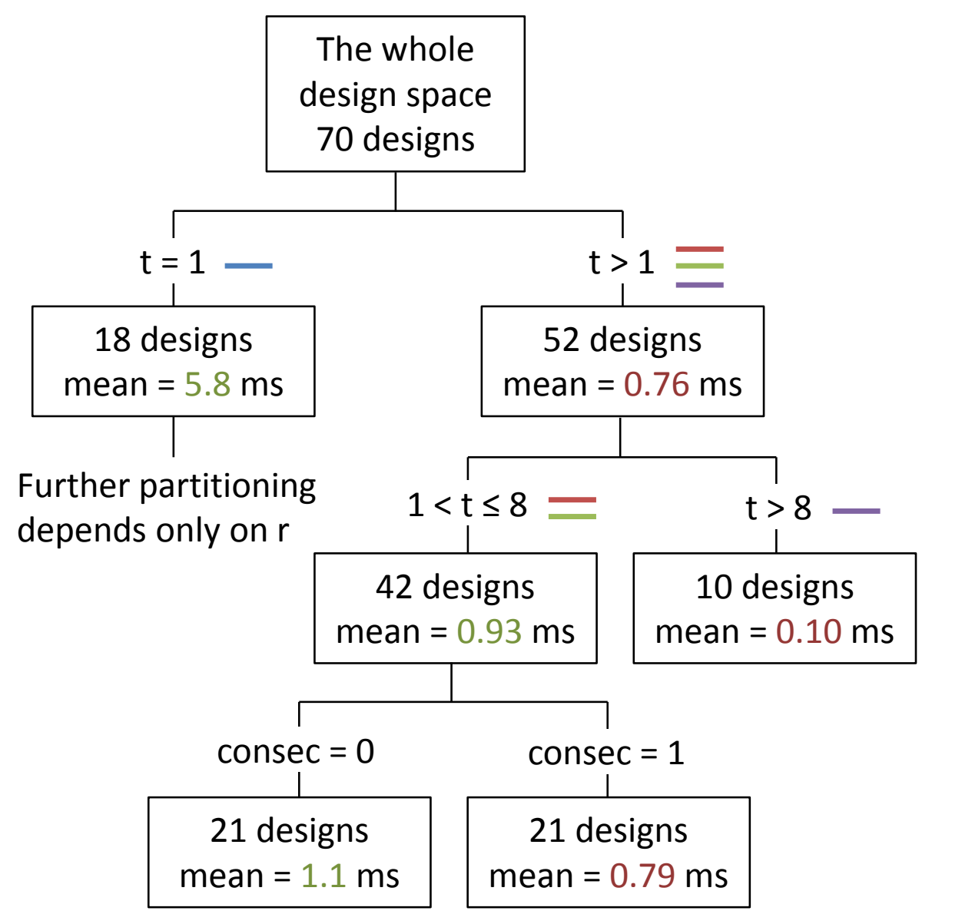
- Partition-based **regression tree** approach is powerful and robust
  - Handles real-system measurement variance
  - Handles "performance cliffs" and "subspaces" common for GPU systems
  - Applicable to multiple metrics and CPUs
  - Tree visualizations are intuitive
- For GPU users, tool builders and hardware designers
  - Optimize designs within or across different platforms
  - Reveal power/performance trade-offs
  - Measure a program's input sensitivity
  - > 300X speed-up in design space exploration

matrix transpose

param	meaning	value
r	# rows / thread block	1–256
t	# threads / row	1–16
consec	threads work on consecutive elements?	0 / 1
	# total designs	70

consec = 0 (—) or 1 (···)




The whole design space 70 designs
t = 1
t > 1
18 designs mean = 5.8 ms
52 designs mean = 0.76 ms
Further partitioning depends only on r
1 < t ≤ 8
t > 8
42 designs mean = 0.93 ms
10 designs mean = 0.10 ms
consec = 0
consec = 1
21 designs mean = 1.1 ms
21 designs mean = 0.79 ms

Starchart Method

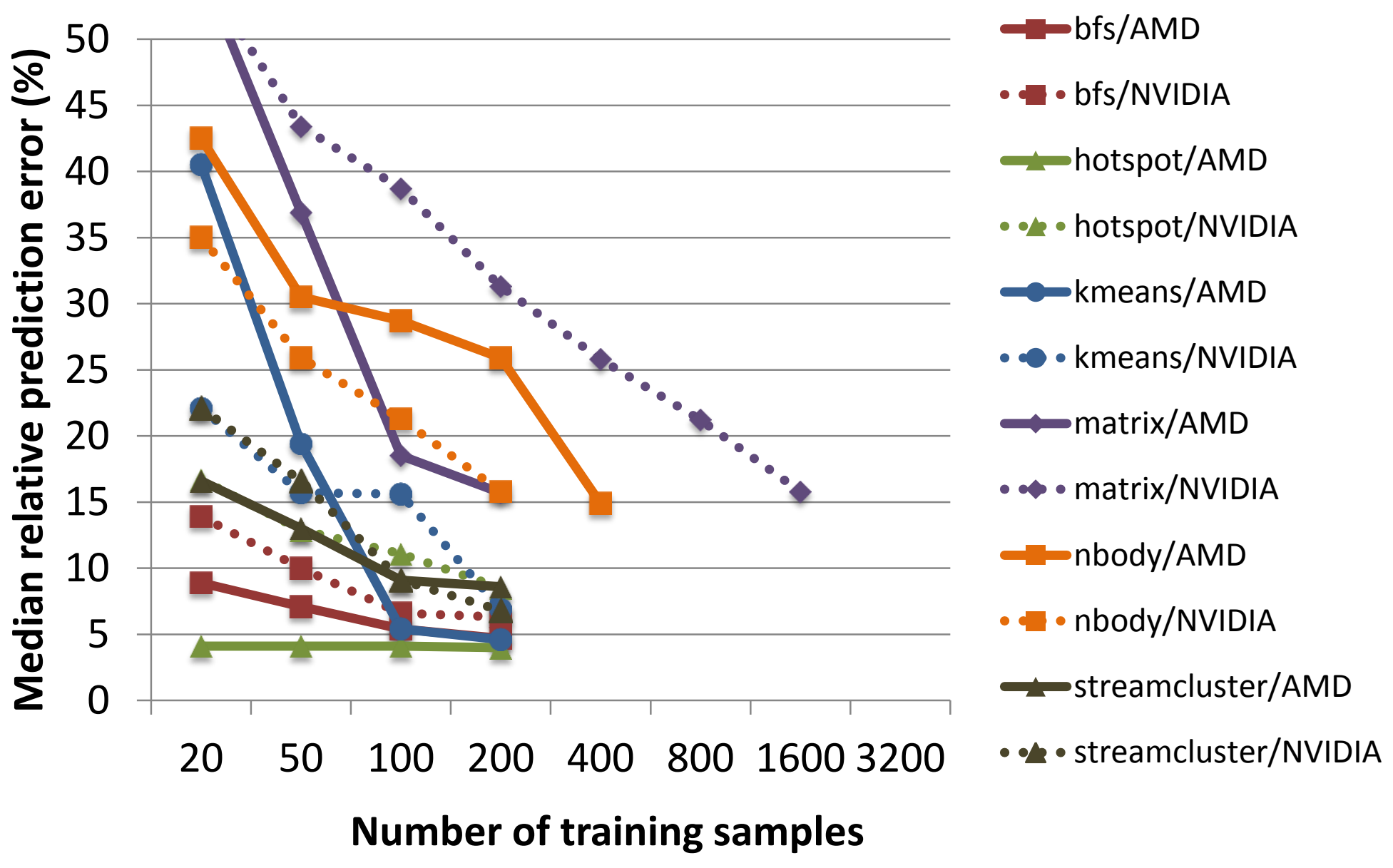
- Step 1: Uniformly and randomly sample N designs from the whole space
- Step 2: Apply an iterative algorithm to recursively partition the samples
- Step 3: Use resulting tree representations to solve subspace-based problems

recursive partitioning

try all tentativ e splits

stopping criteria

```
do
  for each current partition $C$
    for each parameter $d_s$
      for each possible value $v_t$ of $d_s$
      split $C$ into $C_1$ and $C_2$ based on $(d_s, v_t)$
        $SSE_{st} = \Sigma_i(P_{1i} - P_1) + \Sigma_j(P_{2j} - P_2)$
    find the smallest $SSE_{mn}$ among all SSE$_{st}$
    if $SSE_{mn} > 0$
      split $C$ using $(d_m, v_n)$ and add to tree
while at least one partition was split
output the generated partition tree
```



Approach 2: Designing Communication Accelerators

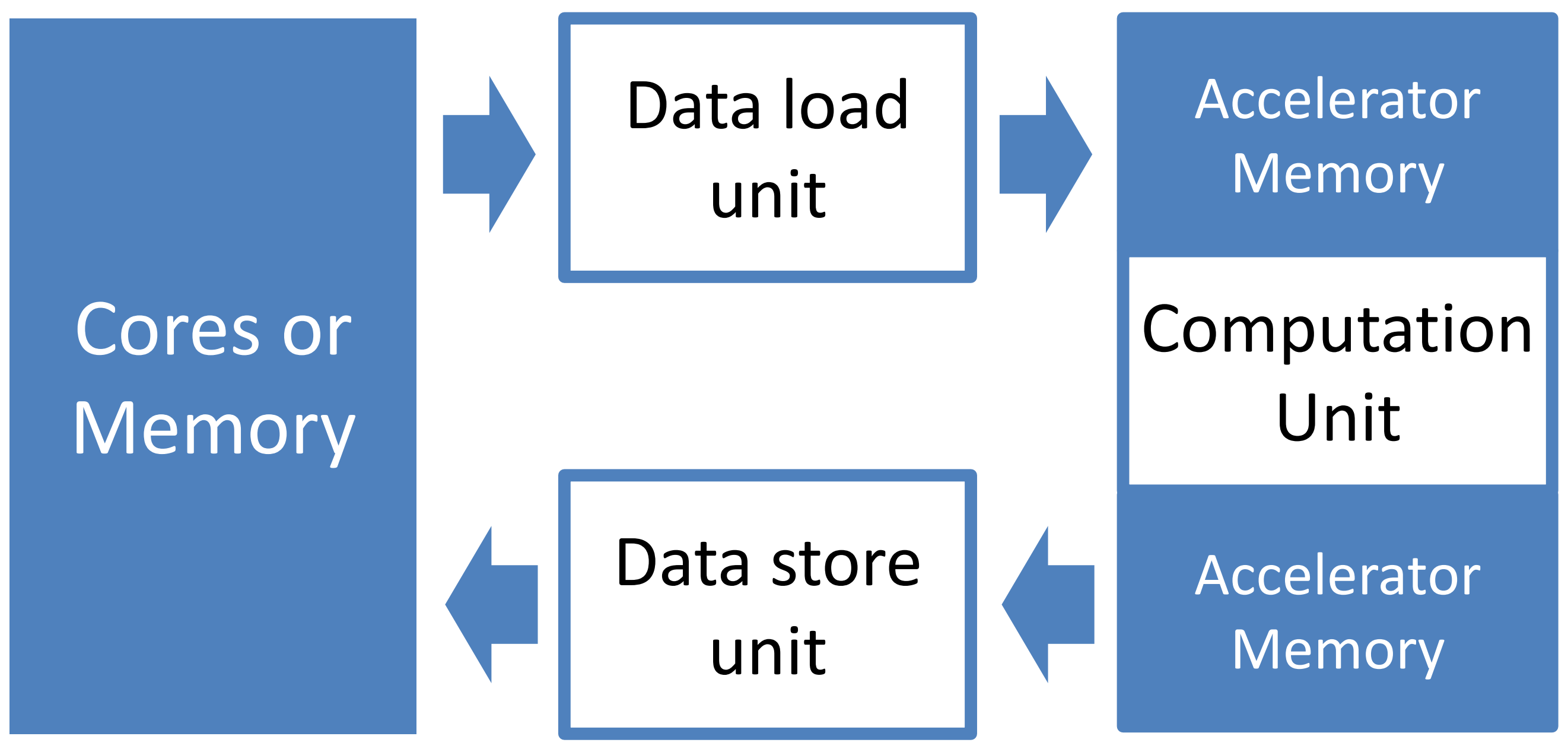
(Work in Progress)

Motivation

- Accelerator design is not just about computation!
- Moving data to/from the accelerator from/to the cores or memory can consume substantial amount of time and energy
- It is necessary to think about both **communication** and **computation** when utilizing an accelerator

Communication-Aware Accelerator Architecture

- An accelerator consists of three components : data load unit, computation unit, data store unit



- On this design, we consider data load and data store each as a single stage of the pipeline (computation can have multiple stages)
- To balance this pipeline, we perform DVFS or similar techniques on either data load/store stage or computation stage to minimize energy consumption
- Our goal is to automate this optimization process

Conclusion

- Heterogeneity calls for **systematic** and **novel** design space analysis techniques
- Automated regression tree methods can solve real-system power/performance optimization problems with > 300X productivity speedup
- Communication-aware accelerators balance communication with computation to significantly reduce wasted energy consumption

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · PRINCETON UNIVERSITY · UC Berkeley · UNIVERSITY of WASHINGTON