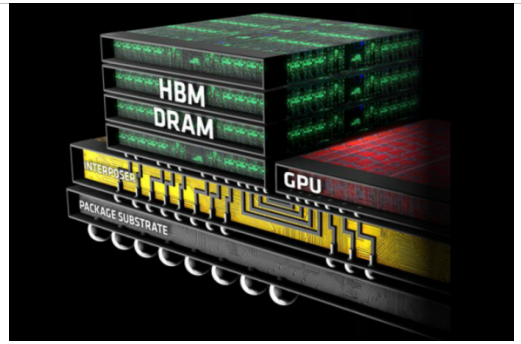
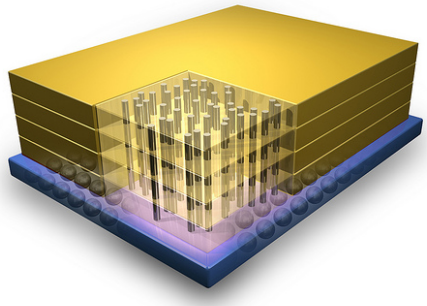
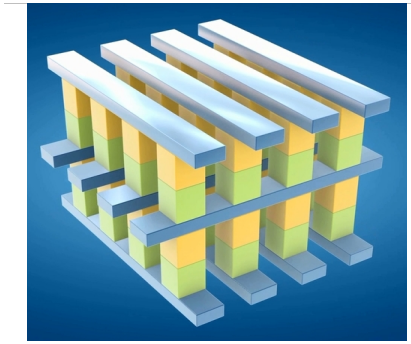
Exceptional service in the national interest

Sandia National Laboratories







The Impact of Increasing Memory System Diversity on Applications

Gwen Voskuilen

Arun Rodrigues, Mike Frank, Si Hammond

4/25/2017

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. SAND2017-4325 C

Diversity in memory hierarchy

Sandia National Laboratories

- New memory technologies – stacked DRAM, non-volatile
  - Advantages: higher bandwidth, persistent, etc.
  - Disadvantages: expensive, higher latencies
- Solution: ***multi-level memory (MLM)***
  - Mix memories to expose advantages, hide disadvantages
  - Really hard in practice
    - Must carefully locate data based on data characteristics
    - Which data goes into which memory? Who decides?

	Automatic	**Manual**
Hardware	Hardware decides	
Software	OS/runtime decides	Application decides

Managing MLM

Sandia National Laboratories

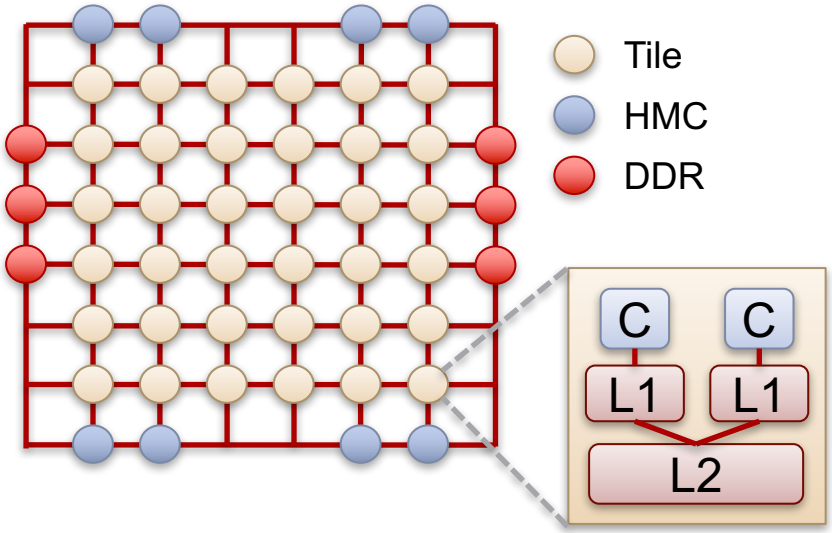
- Trinity KNL: Small stacked DRAM + large DDR DRAM
  - Application data footprint >> stacked DRAM capacity
- Management requires *identifying* and *selectively allocating* data needing bandwidth into stacked DRAM
- Study 1: Software management policies
  - Ranging from simple to complex
  - Developed analysis tool, *MemSieve,* to evaluate memory behavior
- Study 2: Hardware caching
  - Insertion/eviction policies

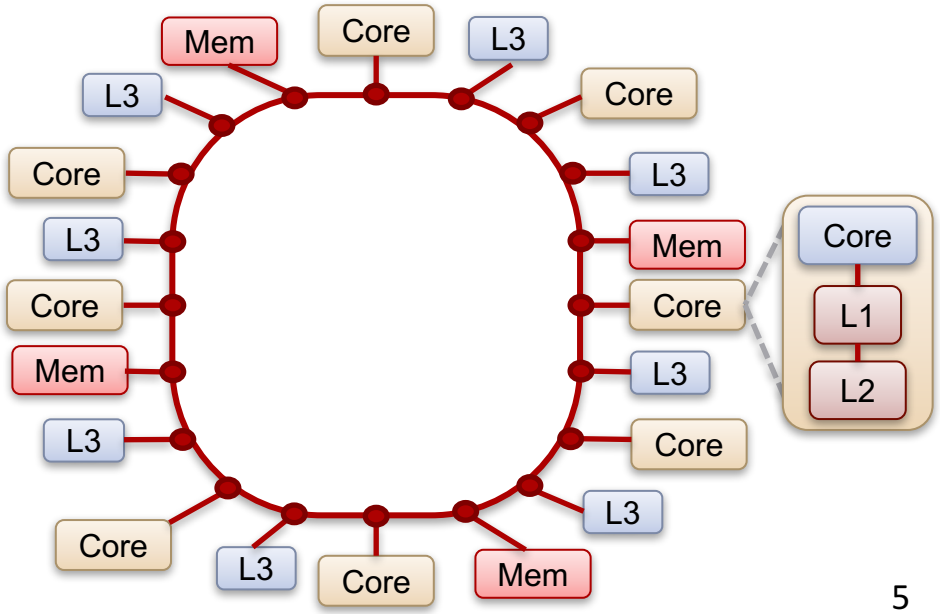
Outline

- Methodology
- Software / manually managed MLM (malloc() based)
- Hardware / automatic managed MLM
- Conclusions

Simulation Methodology

- MiniApps: HPCG, PENNANT, SNAP
  - Memory footprints of 1-8GB → sampling required
- Simulated on two architectures using SST
  - Lightweight: 72 small cores, mesh, private L1, semi-private L2
  - Heavyweight: 8 big cores, ring, private L1 + L2, shared L3
  - Both: DDR and HMC memories



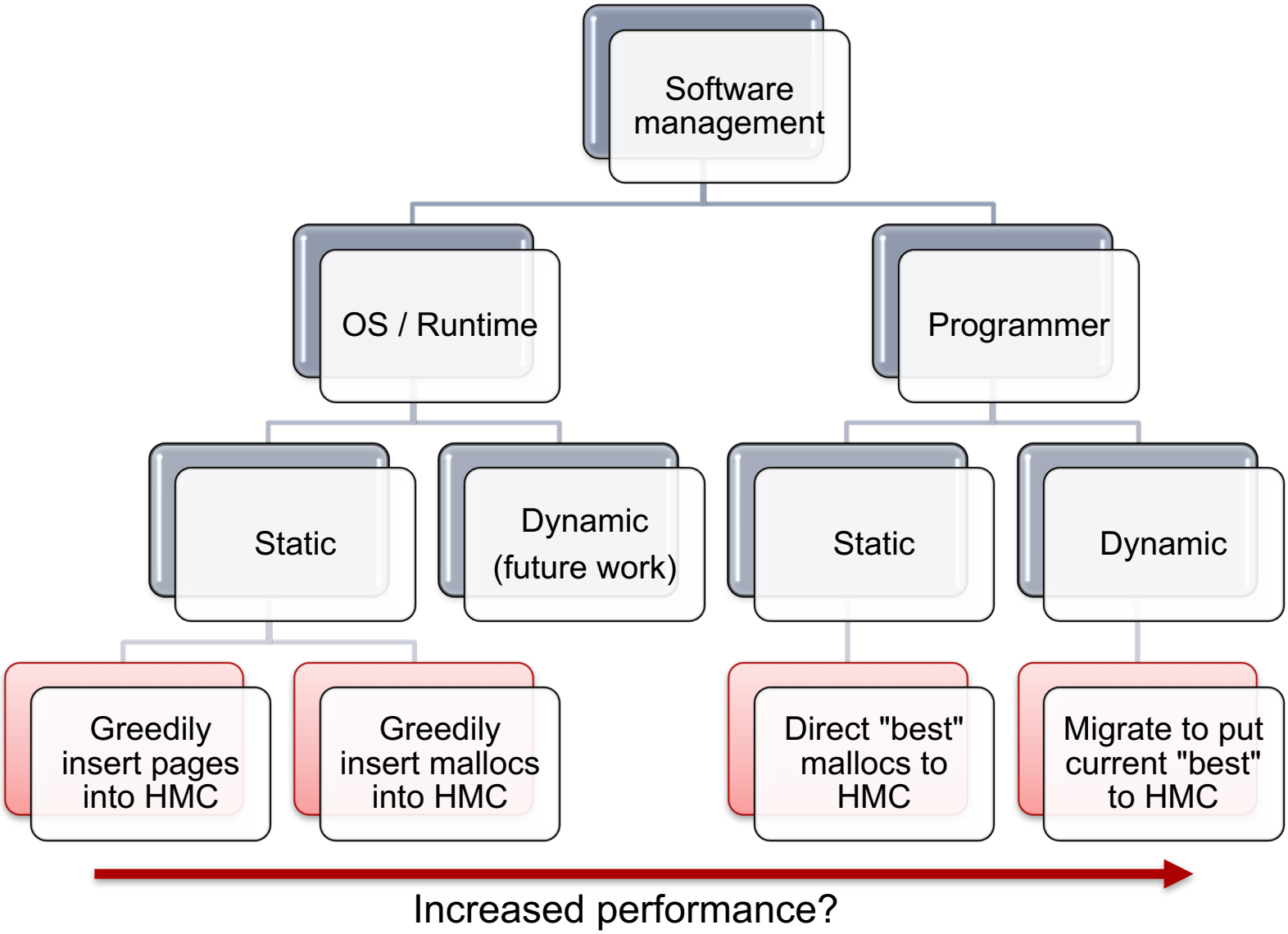


Outline

Sandia National Laboratories

- Methodology
- Software / manually managed MLM
- Hardware / automatic managed MLM
- Conclusions

Software approaches



Tradeoff

Sandia National Laboratories

- Automatic (OS)
  - Easier for programmer
  - Able to capture allocations not under programmer control
    - Library, pre-program start, etc.
  - Page-table complexity; potentially expensive re-mapping
  - No program knowledge → worse performance?
    - Could use programmer hints or runtime profiling but more work
- Manual (programmer)
  - More work for programmer, pervasive (?) changes
  - Not able to handle all allocations
  - Possible conflicts between application and library allocations
    - What if libraries decide to manage allocation for internal structures too?
  - Knowledge of program behavior → better performance?

Analysis tool: MemSieve

Sandia National Laboratories

- Captures an application's memory accesses and correlates to the application's memory allocations
  - Filters out cache hits
  - Without simulating full memory hierarchy → 2.5X + faster
- Key measurement: **malloc density**
  - # accesses / size
- Hypothesis: dense mallocs should be put in HMC
  - Assuming similar latencies between HMC and DDR

Malloc analysis

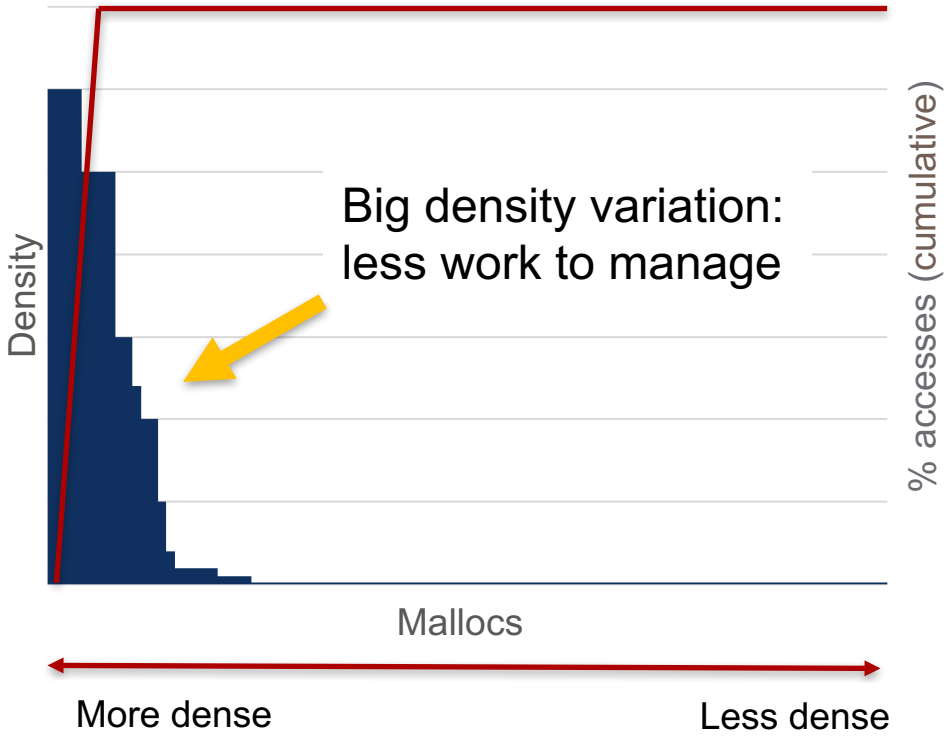
	Pennant	HPCG	Snap *	MiniPIC
Malloc count	8B	23M	1B	438K
Malloc size	32.1 TB	7.43 GB	30GB	7.9GB
Distinct traces	248	612	323	39043
Accessed traces	140	146	90	10794
Size of accessed traces as % total	89.7%	99.987%	89.6%	84%

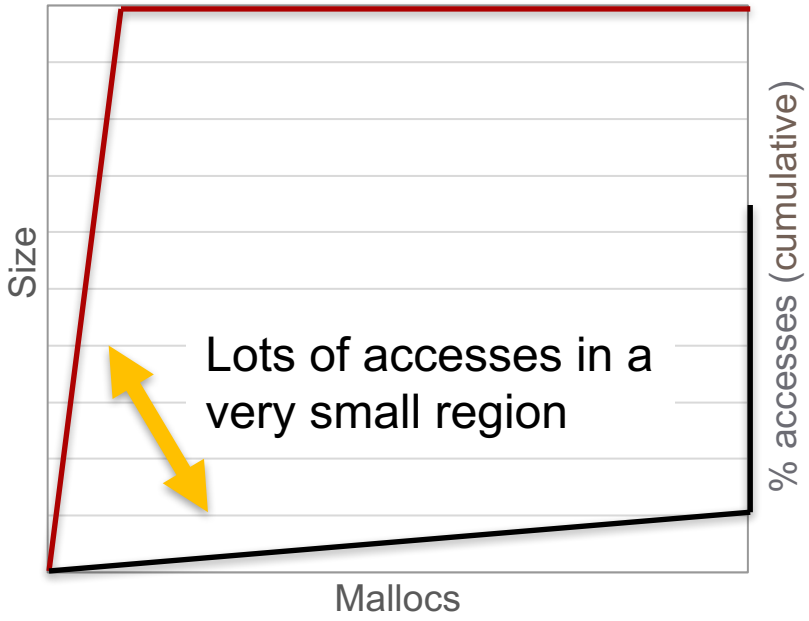
*Iterations from beginning & middle only

- Many mallocs but few distinct malloc call traces (locations)
- Reasons mallocs are not accessed
  - Same address malloc'd repeatedly → cache-resident
  - Malloc was not accessed in profiled section of application

Ideal malloc behavior

- **Good:** A few, small, very dense mallocs
- **Bad:** Many, equally dense mallocs; densest are big

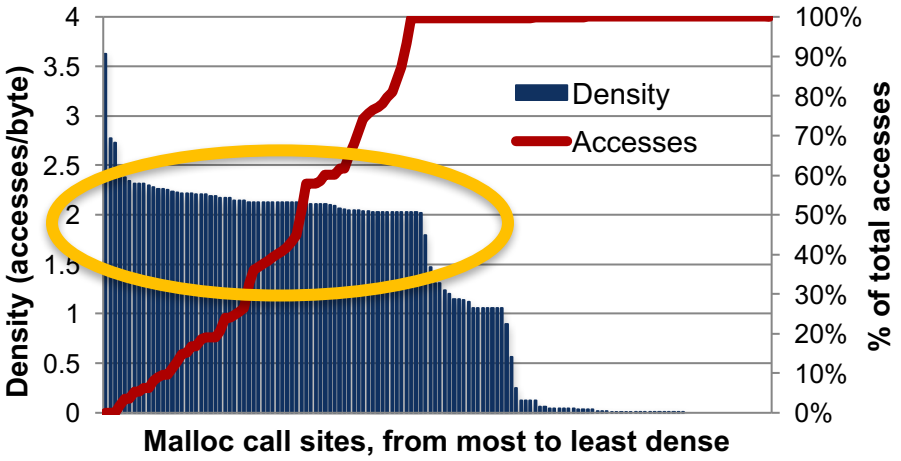




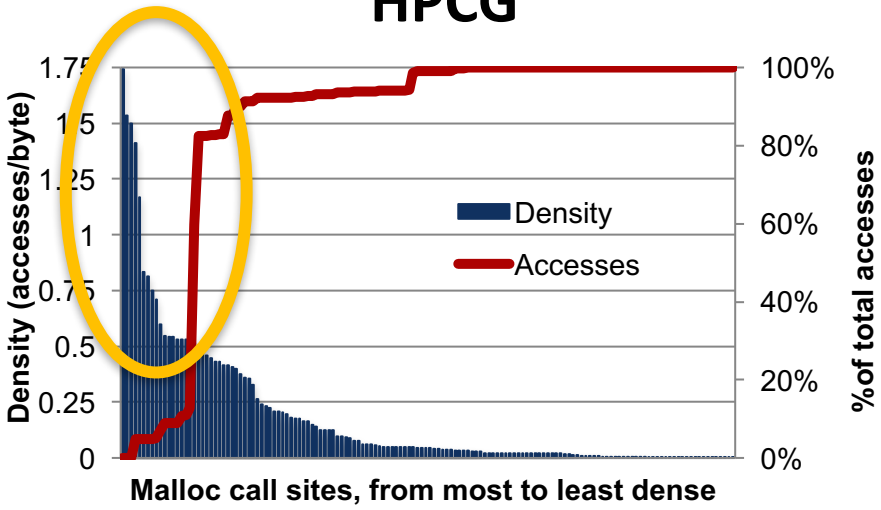
Malloc density

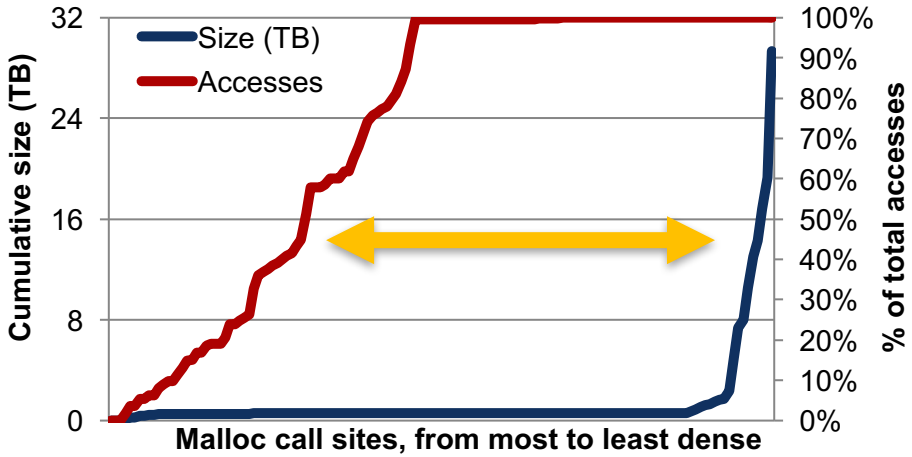
Sandia National Laboratories

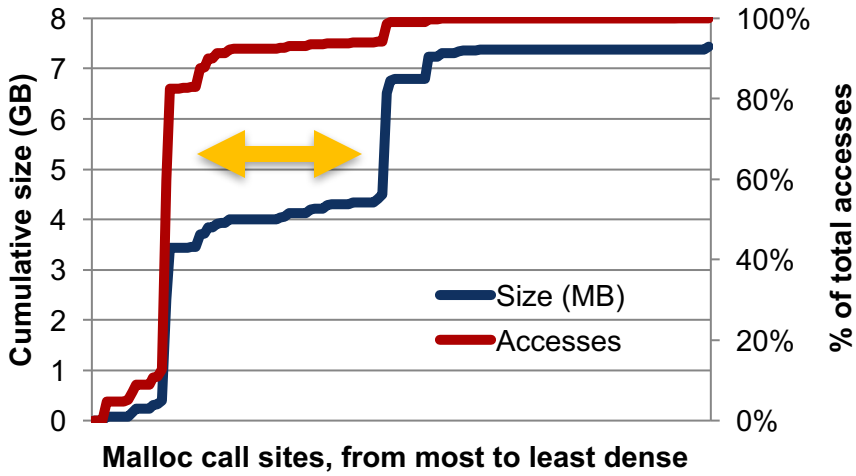
PENNANT



HPCG

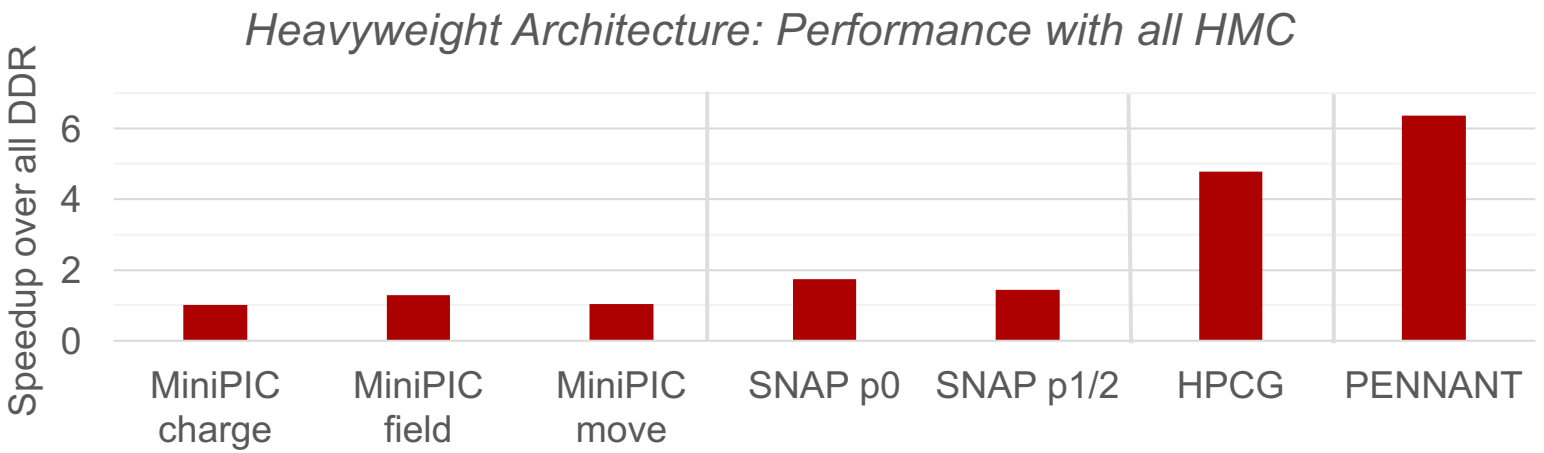


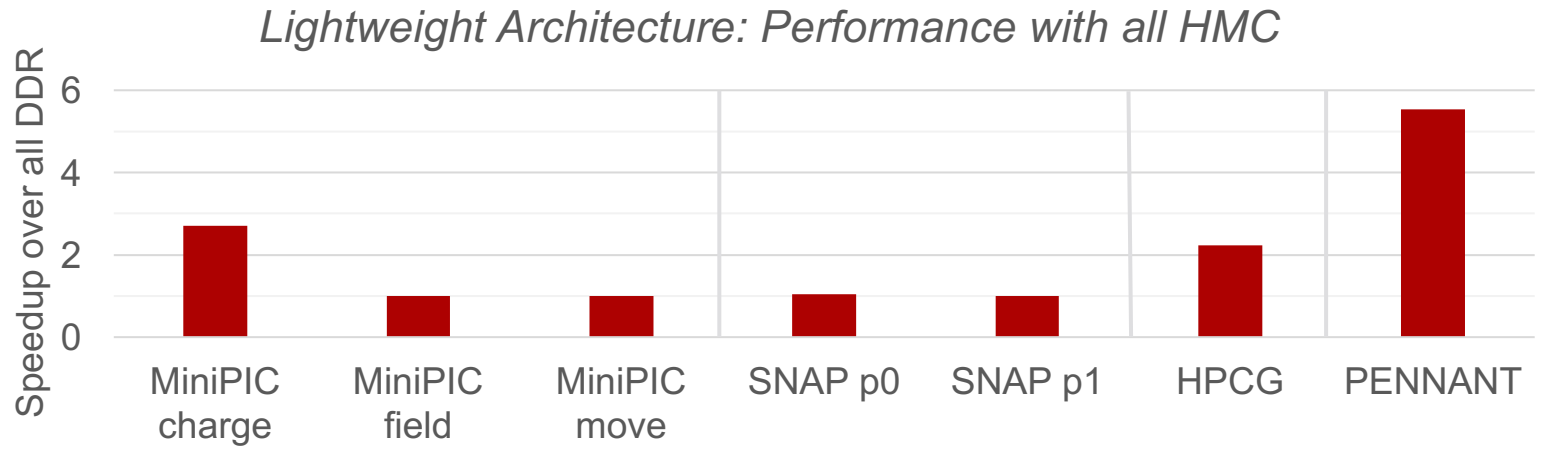




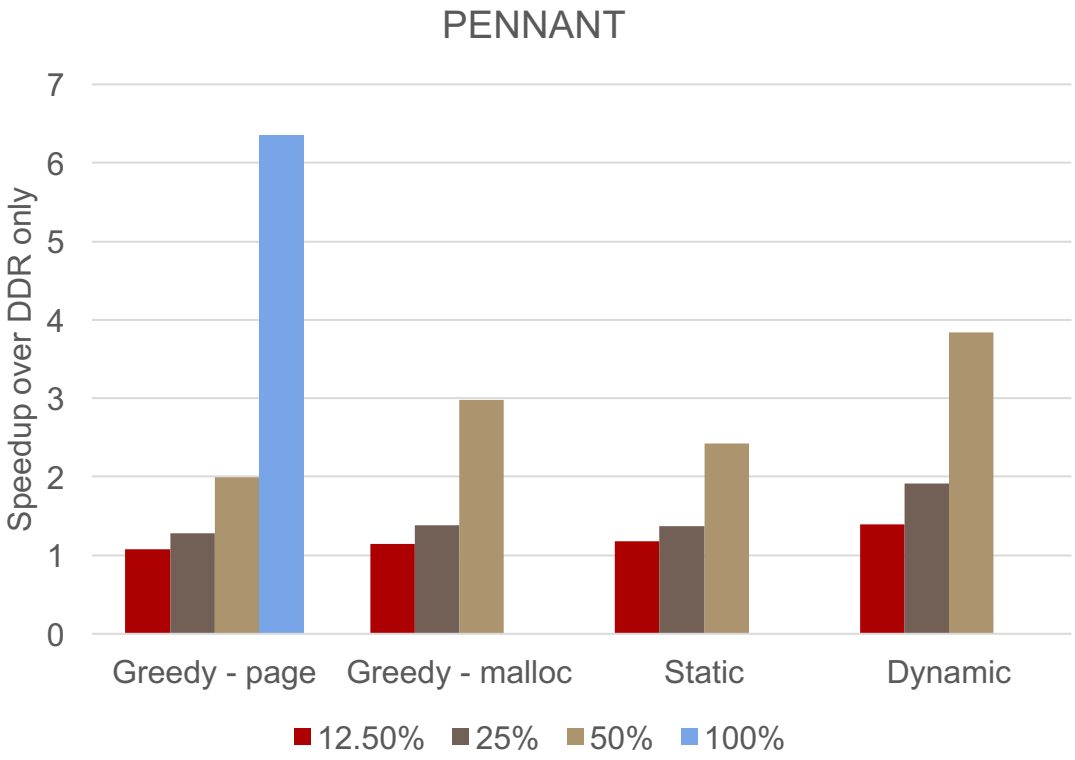
HMC Potential Performance

- **Max: 8X**
- Trends do not change with data set size (1-8GB)



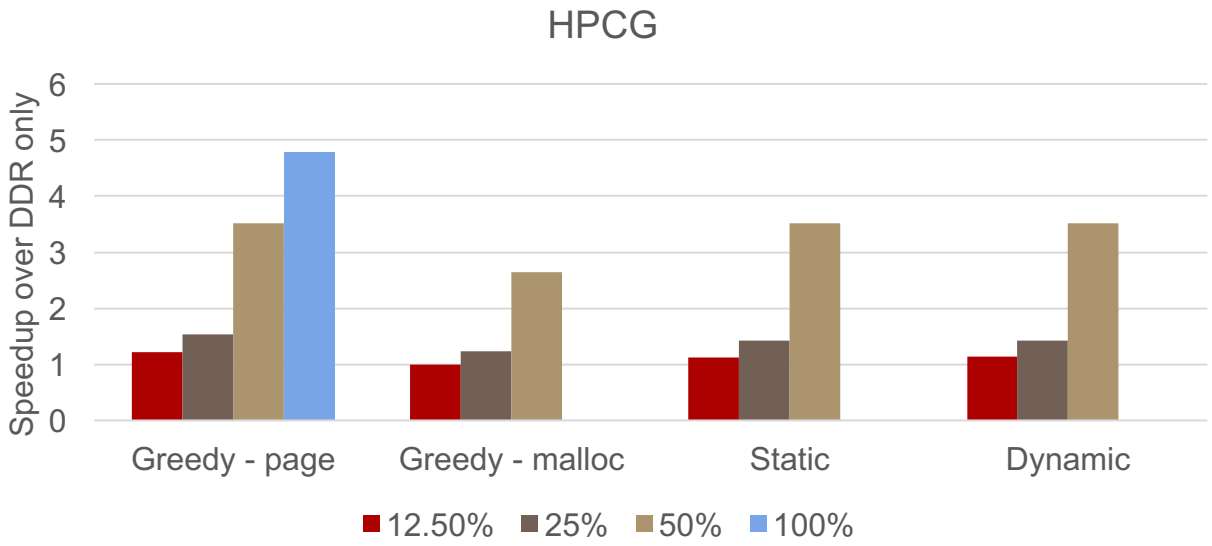


Manual allocation: PENNANT



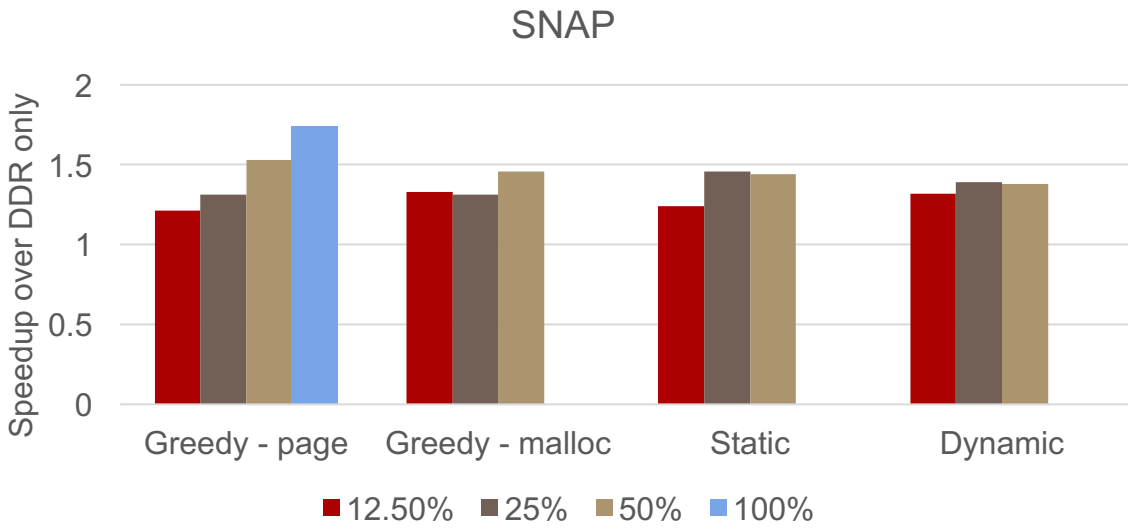
- Large performance jump from 25% to 50% HMC
- Dynamic migration necessary

Manual allocation: HPCG



- Again, large jump from 25% to 50% HMC
- Page-based & static perform similarly
- Dynamic not better
  - But granularity of migration is large

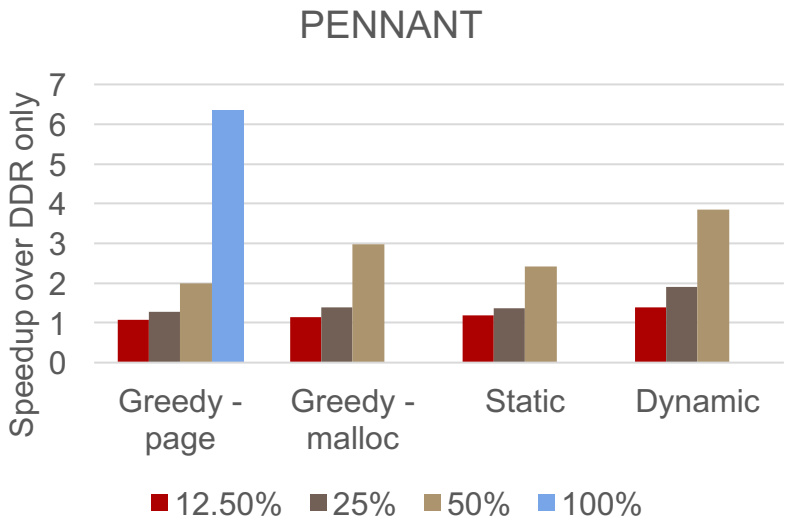
Manual allocation: SNAP



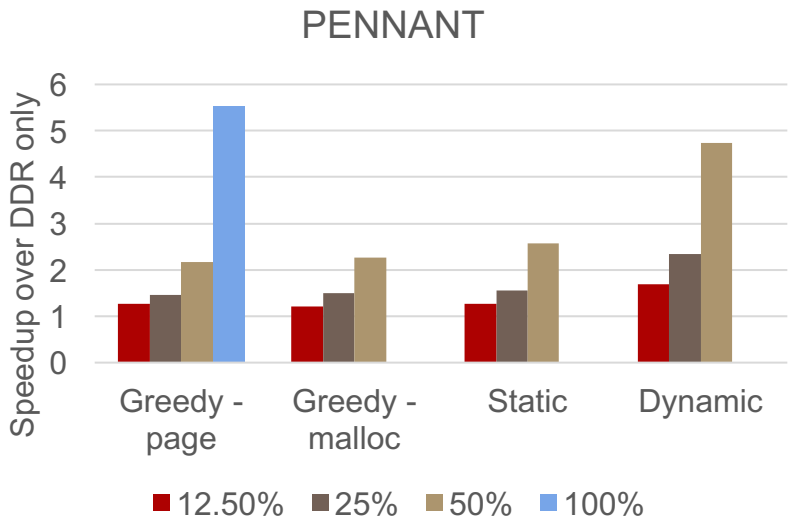
- Greedy-page performs the best
  - Two large mallocs in SNAP, each 42% of total
    - Medium/low density
  - Once they don't fit, HMC size / malloc strategy doesn't matter
- Suggested code change
  - Break up large mallocs to improve HMC utilization

Topology comparison

- Similar trend



Heavyweight



Lightweight

Outline

- Methodology
- Software / manually managed MLM
- Hardware / automatic managed MLM
- Conclusions

Hardware management

Sandia National Laboratories

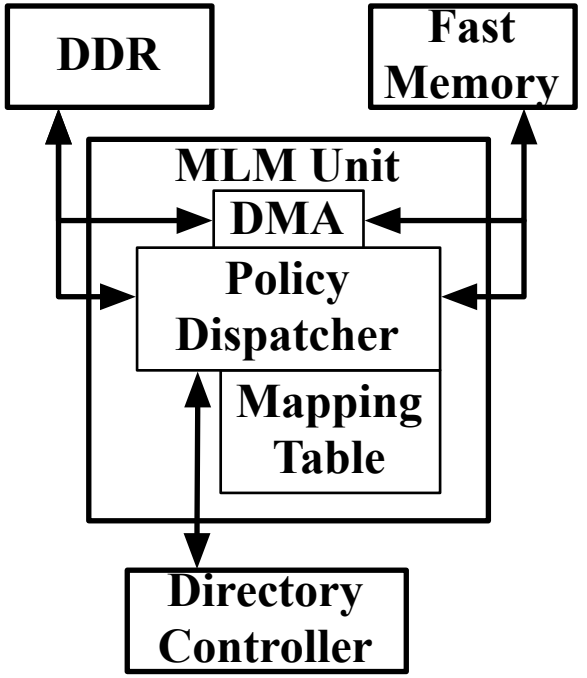
- Hardware management of MLM at the page level
  - Cache pages in HMC, page still resides in DDR
  - Compared to block level: lower tracking overhead but higher add/remove overhead

- Focus was hardware caching
  - But, also possible to do caching via OS
  - Usually, less information (hits, misses, etc.)

Automatic Page-Level Swapping

Sandia National Laboratories

- Addition policies
- Replacement policies

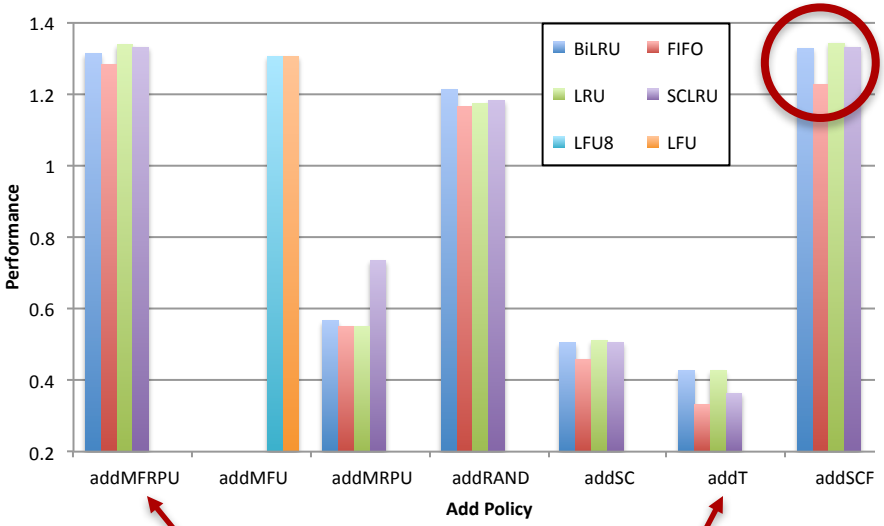


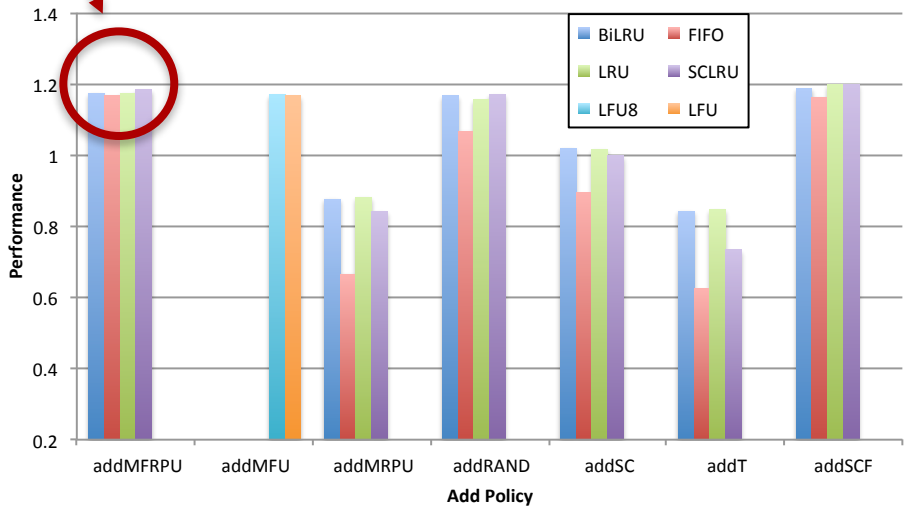
Addition Policies	Replacement Policies
• addT: Simple Threshold • addMFU: Most Frequently Used • addRAND: 1:8192 chance • addMRPU: More Recent Previous Use • addMFRPU: More Frequent + More Recent Previous Use • addSC: Deprioritize streams • addSCF: as addSC + More Frequent	• FIFO: First-in, First-out • LRU: Least Recently Used • LFU: Least Frequently Used • LFU8: LFU w/ 8-bit counter • BiLRU: BiModal LRU • SCLRU: Deprioritize streams

Performance vs. Policy

Sandia National Laboratories

Replacement policy: little variation



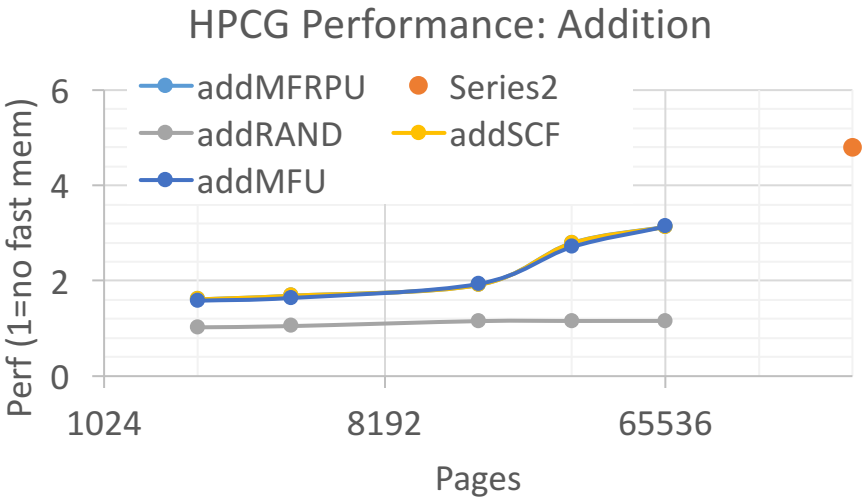


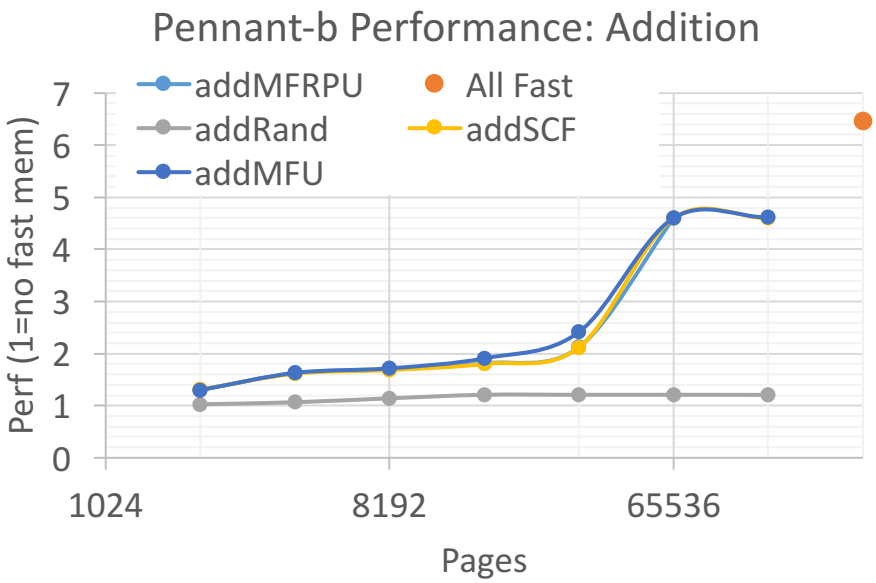
Addition policy: big variation

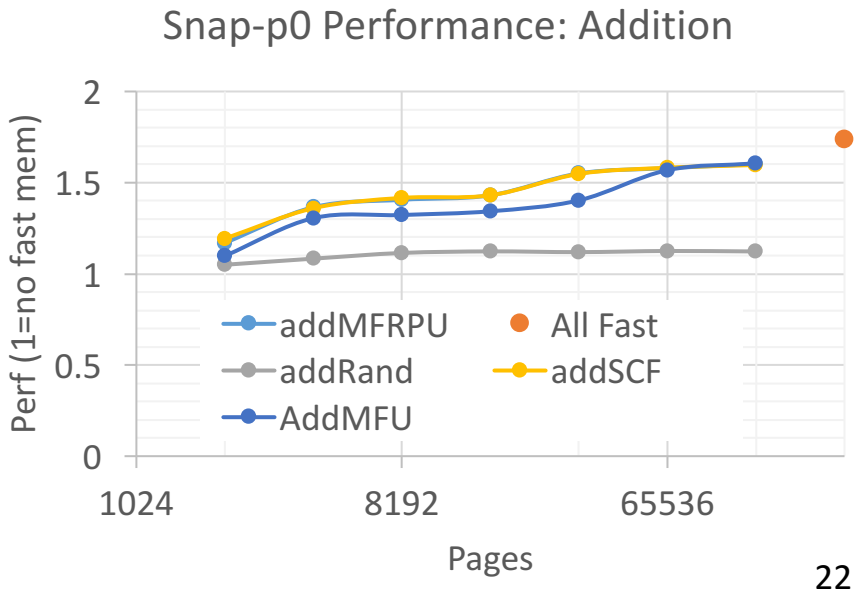
"What you put in matters more than what you take out"

Larger data sets

- Looked at highest performing addition policies
  - Variants of most-frequently used
  - Baseline: random
  - LRU replacement

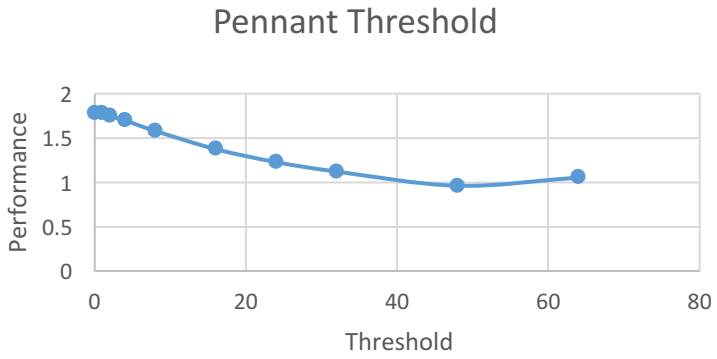


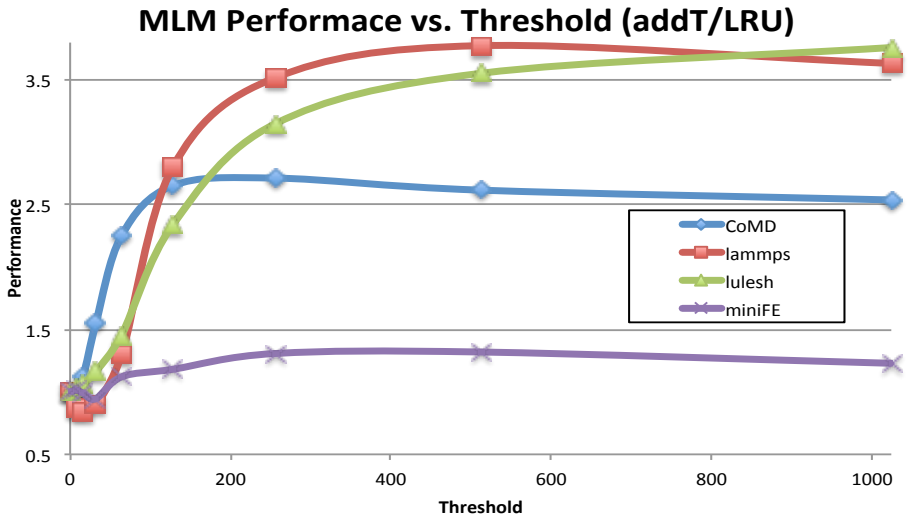




Fine Tuning

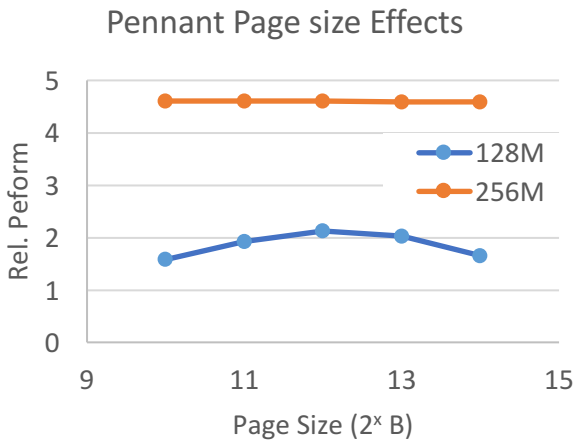
1. Thresholds

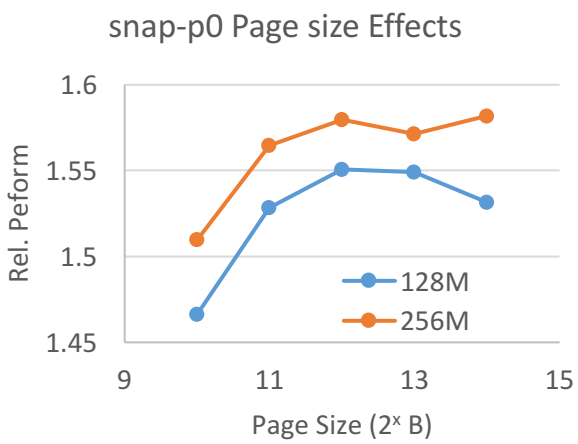


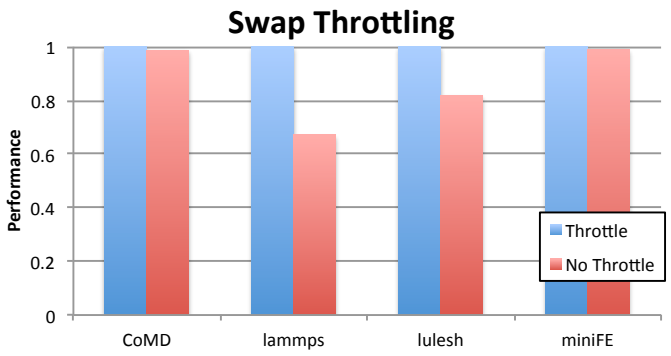


2. Page size

3. Throttling







Conclusions

Sandia National Laboratories

- Application behavior varies significantly
- Software management is feasible
  - Small to moderate number of *dense* call sites
  - Static allocation sufficient in some cases, dynamic necessary in others
- For hardware management, addition policy matters most
- For automatic & manual, profiling is instrumental
  - Application managed – helps identify high-bandwidth data
  - Automatically managed – helps identify places where application changes will improve performance

Sandia
National
Laboratories

Exceptional service in the national interest