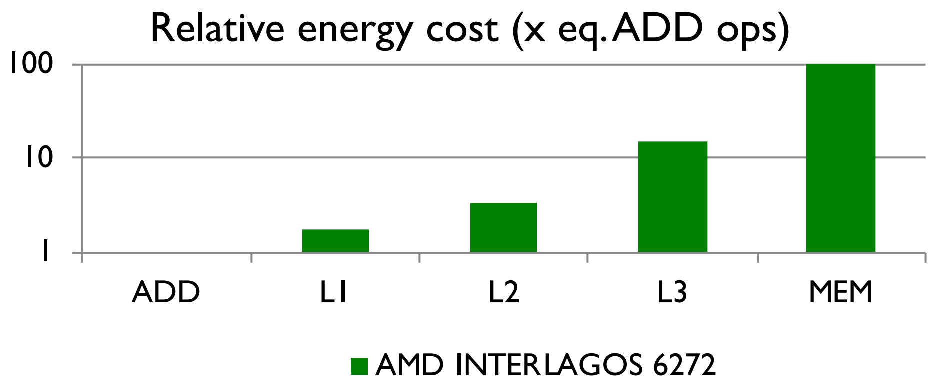
DataProf: Exposing Data Movements in the Memory Hierarchy

William Wang, Chris Emmons, Nigel Paver

June 13, 2014

Data Movements Dominate

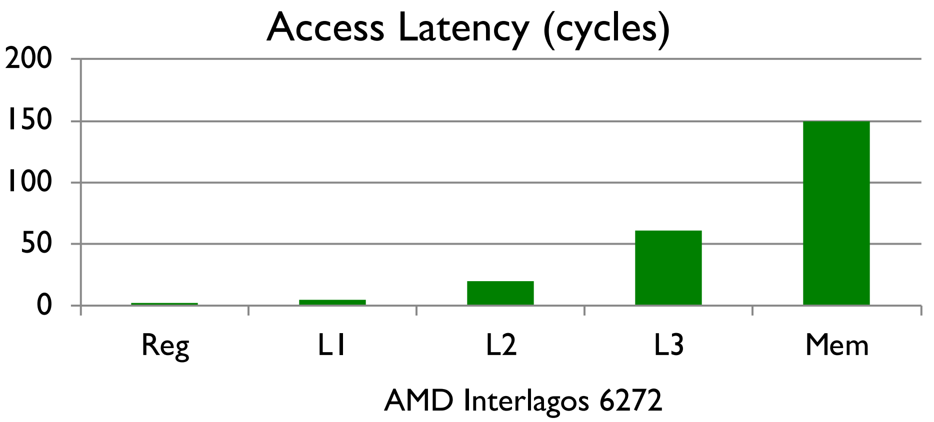
- Data movements cost 2x ~100x more energy than computations, and getting worse with shrinking nodes



technology node	130nm CMOS (2006)	45nm CMOS (2008)
transfer 32b across-chip	20 computations	57 computations
transfer 32b off-chip	260 computations	1300 computations

Source: Simon Moore, Communication: the next resource war

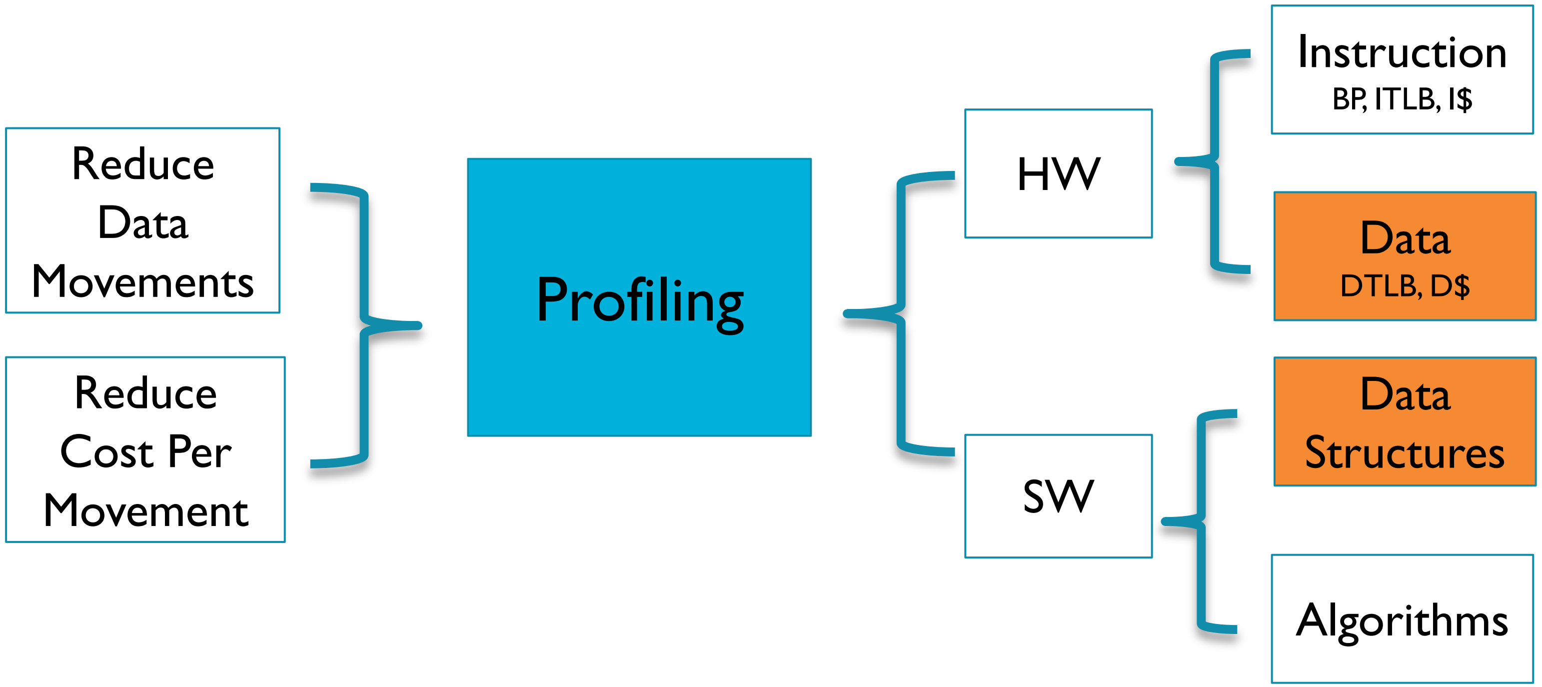
- Plus, it takes more cycles to move data to registers than the actual computation



Source: Kestor, Gokcen, et al. "Quantifying the energy cost of data movement in scientific applications."

ARM

Optimize Data Movements for Energy Efficiency



ARM

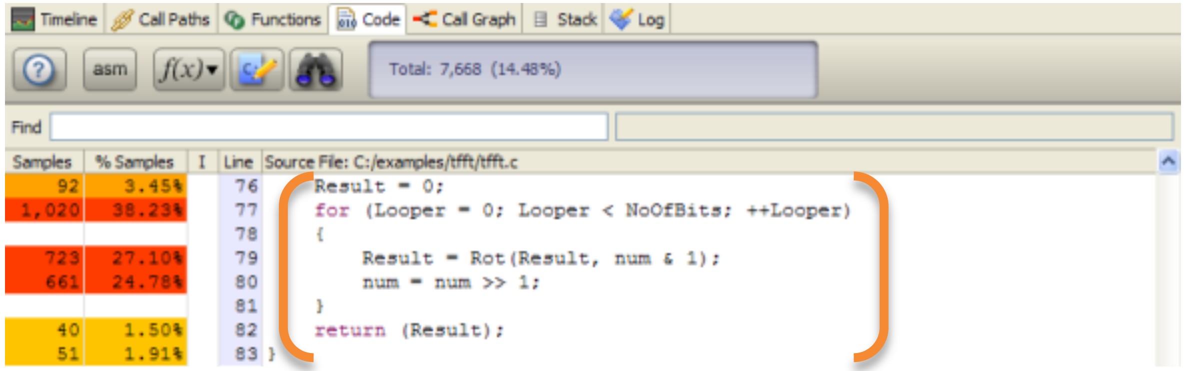
Data Profiling Helps Measure Data Movements

"You can't optimize what you can't measure"

"To measure is to know." – Lord Kelvin

- Code profile helps detect code hotspots
  - DS-5
  - gprof
  - OProfile

Code Profile



- Data profile helps detect data hotspots
  - MemSpy
  - CProf
  - DProf

Data Profile

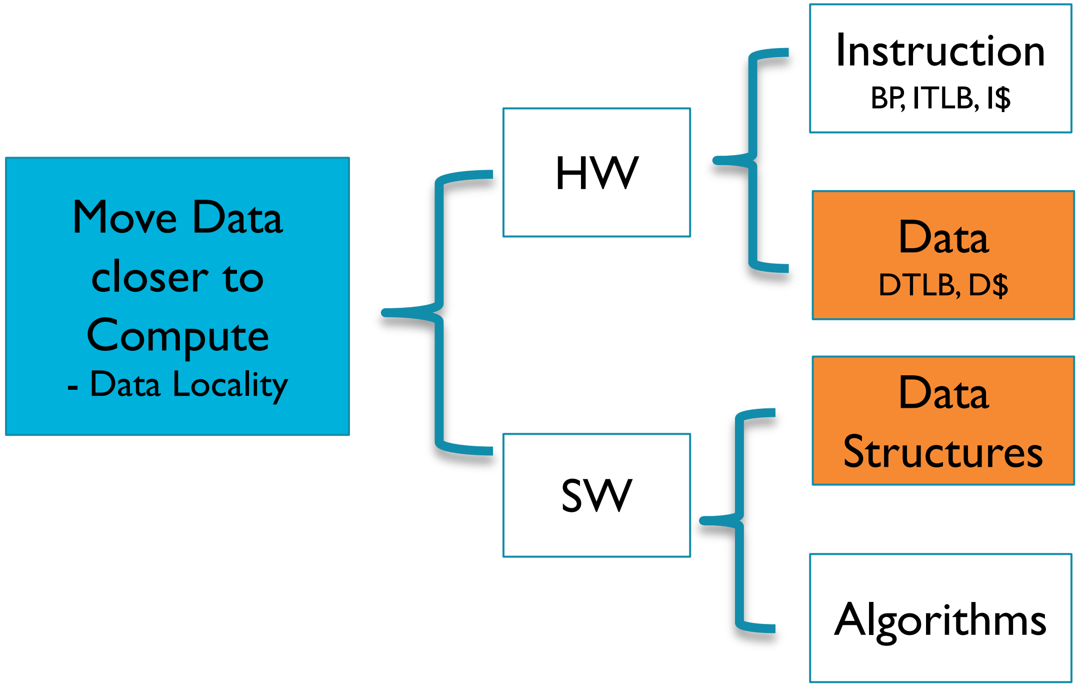
Type Name	Description	Working Set View	Data Profile View	
		Size	% of all L3 misses	Bounce
slab	SLAB bookkeeping structure	2.5MB	32%	yes
udp_sock	UDP socket structure	11KB	23%	yes
size-1024	packet payload	20MB	14%	yes
net_device	network device structure	5KB	12%	yes
skbuff	packet bookkeeping structure	34MB	12%	yes
ixgbe_tx_ring	IXGBE TX ring	1.6KB	1.7%	no
socket_alloc	socket inode	2.3KB	1.7%	yes
Qdisc	packet schedule policy	3KB	0.8%	yes
array_cache	SLAB per-core bookkeeping	3KB	0.4%	yes
	Total	57MB	98%	—

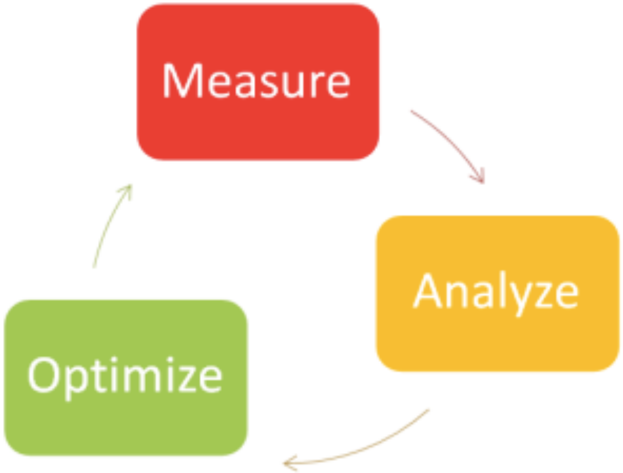
Source: Pesterev et.al, Locating Cache Performance Bottlenecks Using Data Profiling

ARM

Data Profiling and Heterogeneous Memory

- Goals: Address rising cost of communication
  - **<u>Expose data flows in real software</u>**
  - Optimize software data structures and access patterns
  - Optimize system memory hierarchies
    - Optimize data storage onto heterogeneous memories

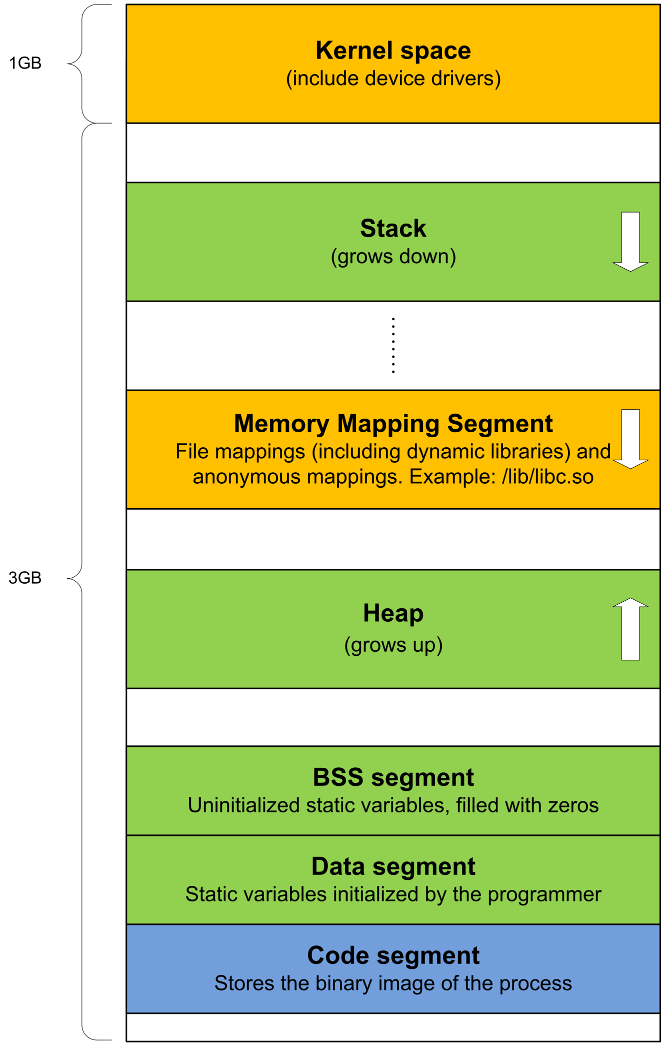




ARM

DataProf Features

- Data Access Hotspots
  - All data variables in the user space
    - Dynamic data on the heap and local variables on the stack
    - Static data in the .bss and .data sections
  - Data members in C structures and arrays
    - Structure layout reorganization and access pattern optimization
- Cache Miss Types
  - Non-sharing misses: compulsory, capacity and conflicts
  - Sharing misses: false and true sharing
- Data View Linked to Code View in Streamline Analyzer®
  - Dwarf information
- Data Access Call Paths
  - Dwarf debug frame information for stack backtrace



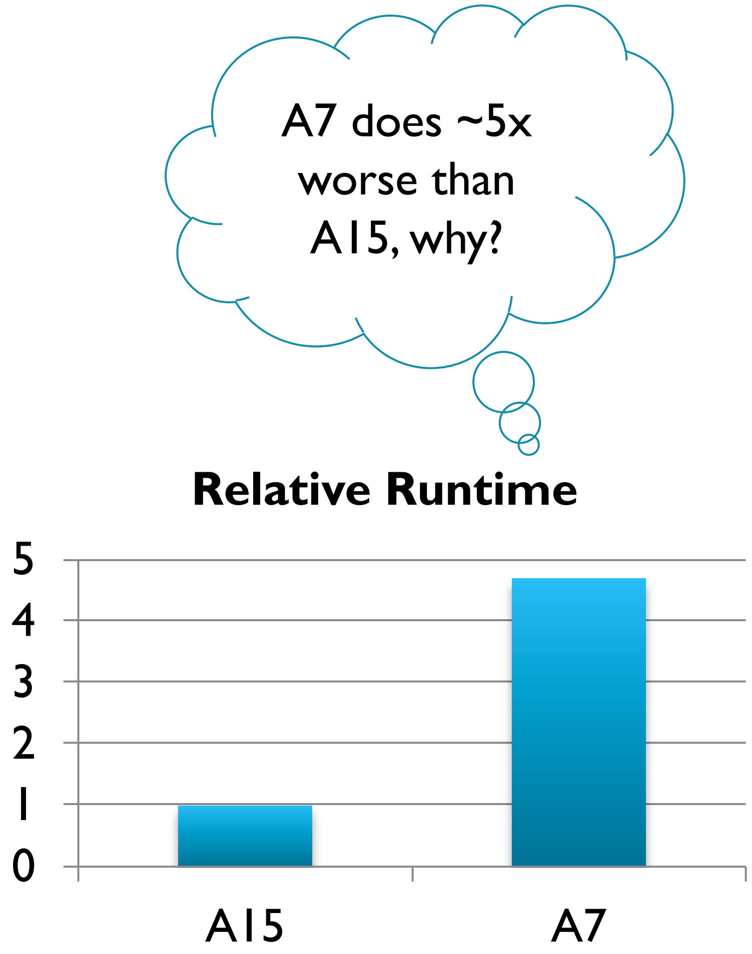
ARM

Example Program

```
#define M = 2048;    // stride distance
#define N = 64;      // number of elements
#define IREP = 200;  // iterations

double x[M*N], y[M*N];

for (int j = 0; j < IREP; ++j) {
  for (int i = 0; i < N*M; i += M) {
    y[i] += x[i];
}}
```



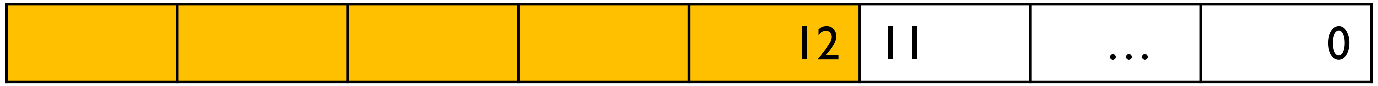
Linux with Normal Page Size 4KB on TC2 platform, IREP=2,000,000

TC2 Platform A15 and A7 Cache Configurations

- Configure the platform in gem5 simulator
- Run the program in gem5 with DataProf enabled
- Visualize the results in Streamline Analyzer

	L1D$			L2$		
	Size (KB)	**Way**	**Replacement**	**Size (KB)**	**Way**	**Replacement**
A15	32	2	LRU	1024	16	Random
A7	32	4	Pseudo Random	512	8	Pseudo Random

Normal Page 4KB



Reference: Gutierrez, et al. "Sources of Error in Full-System Simulation."

Data Profiling – Streamline Data View Shows Cache Misses

All Reads miss in L1

Mostly due to conflict

Write no miss in A15, mostly miss in A7

L1 D Cache

A15

Data variable	Size	Accesses	Read	L1D$Misses	Compulsory	Conflict	Capacity	True sharing	False sharing	Write	L1D$Misses
y[131072]	8	25600	12800	12800	64	12736	0	0	0	12800	0
x[131072]	8	12800	12800	12800	64	12736	0	0	0	0	0

A7

Data variable	Size	Accesses	Read	L1D$Misses	Compulsory	Conflict	Capacity	True sharing	False sharing	Write	L1D$Misses
y[131072]	8	25600	12800	12800	64	12736	0	0	0	12800	12799
x[131072]	8	12800	12800	12800	64	12736	0	0	0	0	0

L2 Cache

A15

L2ReadMisses	Compulsory	Conflict	Capacity	L2WriteMisses	Compulsory	Conflict	Capacity
10333	64	10269	0	0	0	0	0
10426	64	10362	0	0	0	0	0

A7

L2ReadMisses	Compulsory	Conflict	Capacity	L2WriteMisses	Compulsory	Conflict	Capacity
19215	64	19151	0	0	0	0	0
12795	64	12731	0	0	0	0	0

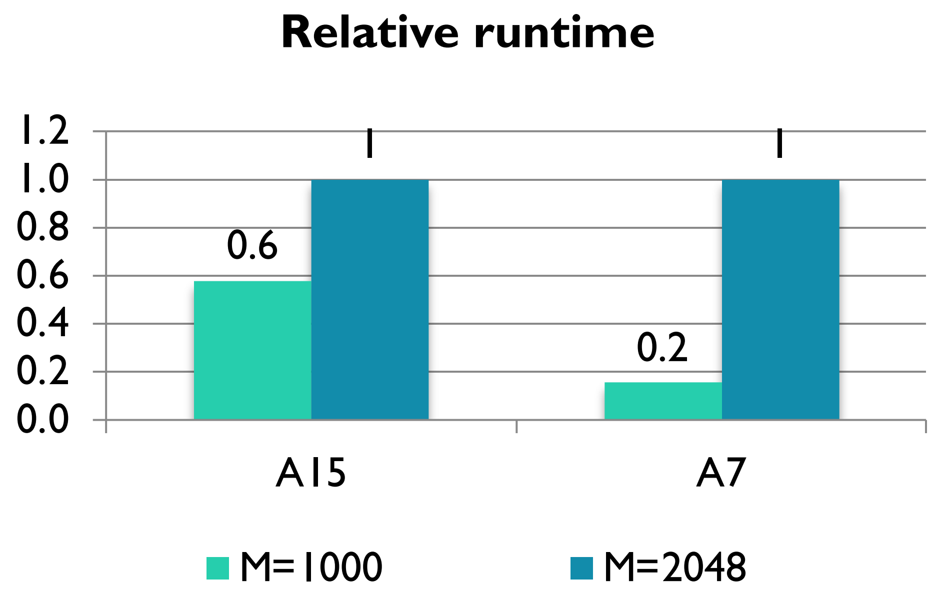
L2 accesses hit more in A15 than in A7

ARM

Optimizations in Software and Hardware

Software optimizations

- Don't stride at the D$ set size
- Reorganize array elements – gather/scatter



Hardware optimizations

- Hashed cache indexing
- Increase A7 L2 associativity

Summary

- Overview of Data Profiling
- DataProf Features
- Data Profile, Analyze and Optimize with an Example Program