Energy-Efficient In-Memory Data Stores on Hybrid Memory Hierarchies

Eleventh International Workshop on Data Management on New Hardware

June 2015

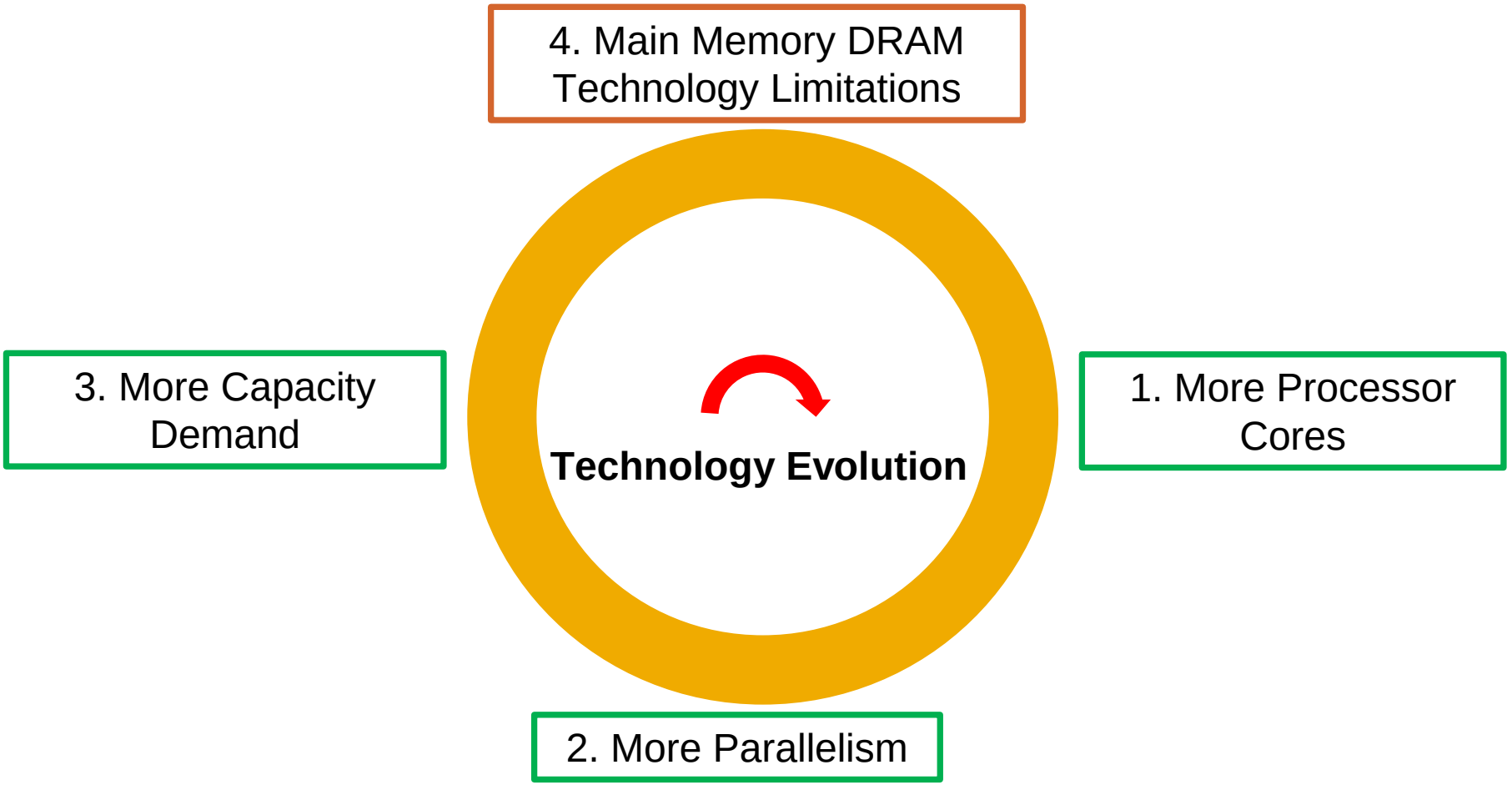
Ahmad Hassan (SAP), Hans Vandierendonck (QUB) and Dimitrios S. Nikolopoulos (QUB)

Presentation structure

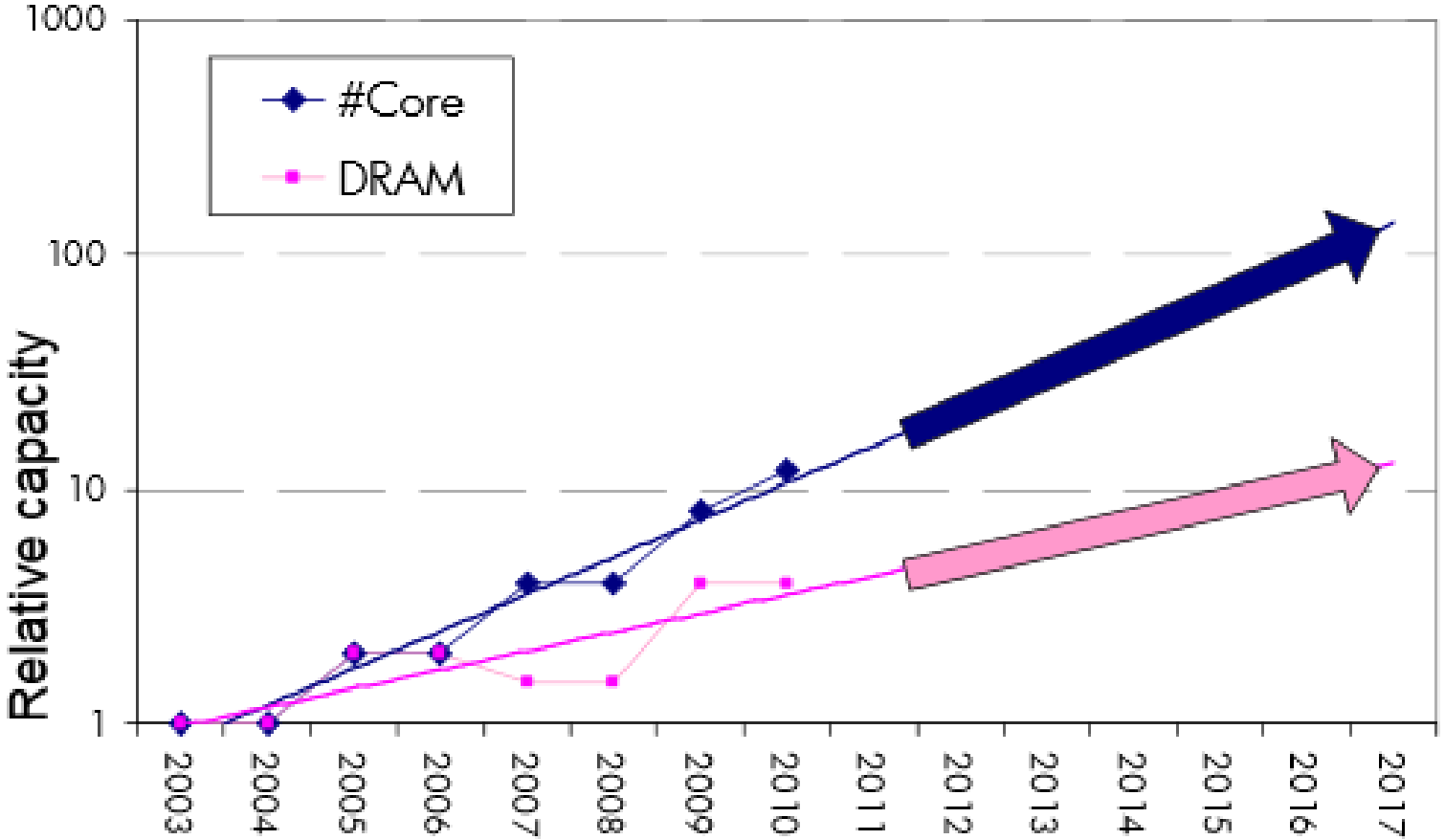
- Research Problem
- Proposed Solution
- Methodology
- Evaluation
- Conclusion

Research Problem

Processor technology has evolved faster than Main Memory



Every 2 years, there is a 30% relative decrease in Main Memory DRAM capacity per processor core



ISCA 2009: web.eecs.umich.edu/~twenisch/papers/isca09-disaggregate.pdf

DRAM has technology limitations – physical scalability limits and inefficient power consumption

Scalability

Technology Scaling for Large Memory Capacity:
DRAM has hit scaling limit (Hard to scale below 40 nm) [ITRS. International Technology Roadmap for Semiconductors, 2011]

Power-inefficiency

Main memory subsystem energy:
DRAM-based main memory consumes 30-40% of the total server power [L. A. Barroso et al. Synthesis Lectures on Computer Arch. 2009]

Different Main Memory Technologies

Feature	DRAM	RRAM	STTRAM	PCM
Cell Size	$6 – 8F^2$	$> 5F^2$	$37F^2$	$8 – 16\ F^2$
Read Latency	~30ns	~116ns	~105ns	~151ns
Write Latency	~30ns	~145ns	~77ns	~396ns
Read Energy*	5.90	4.81	16.60	80.41
Write Energy*	12.70	13.80	21.05	418.6
Static Energy	**YES**	Negligible	Negligible	Negligible
Byte-Addressable	YES	YES	YES	YES
Write Endurance	$> 10^{15}$	$> 10^{5}$	$> 10^{15}$	$> 10^{8}$

***Read/write Energy is presented in nanojoule per 32 byte access**
http://www3.pucrs.br/pucrs/files/uni/poa/facin/pos/relatoriostec/tr060.pdf
http://dl.acm.org/citation.cfm?id=2742854.2742886

All this means is that,

DRAM is not a viable choice for applications that demand large memory

And our research problem becomes….

DRAM is not a viable choice for applications that demand large memory

Can Non-Volatile Memories (NVM) present a better alternative?

Proposed Solution

Before we dive down further, let’s quickly re-cap what an NVM is

NVM (Non-volatile memory) is an emerging main memory technology that is byte-addressable like DRAM

Using NVM over DRAM has key advantages – such as power efficiency

Advantage

Lower leakage power than DRAM

Using NVM over DRAM has key advantages – such as power efficiency and better scalability

Advantage

Lower leakage power than DRAM

Advantage

Large capacity and better scalability than DRAM

However it has its downsides too – NVM has higher latency than DRAM

Advantage	Lower leakage power than DRAM
Advantage	Large capacity and better scalability than DRAM
Disadvantage	**Higher latency and dynamic energy than DRAM**

So we gather a pure NVM-based approach is not viable either

Pure NVM-based solution

Because of the higher latency, and

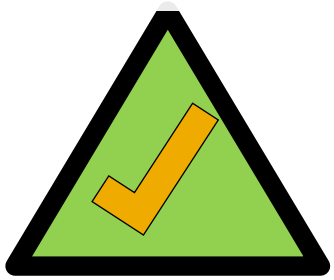
Pure NVM-based solution

Challenge! How to use NVM as main memory technology without hitting NVM low latency bottleneck and reducing main memory subsystem's energy?

So instead a hybrid NVM/DRAM approach could be the answer we are looking for...

Pure NVM-based solution

Challenge! How to use NVM as main memory technology without hitting NVM low latency bottleneck and reducing main memory subsystem's energy?



Proposed Solution: Hybrid NVM/DRAM main memory system…and we'll explain how…

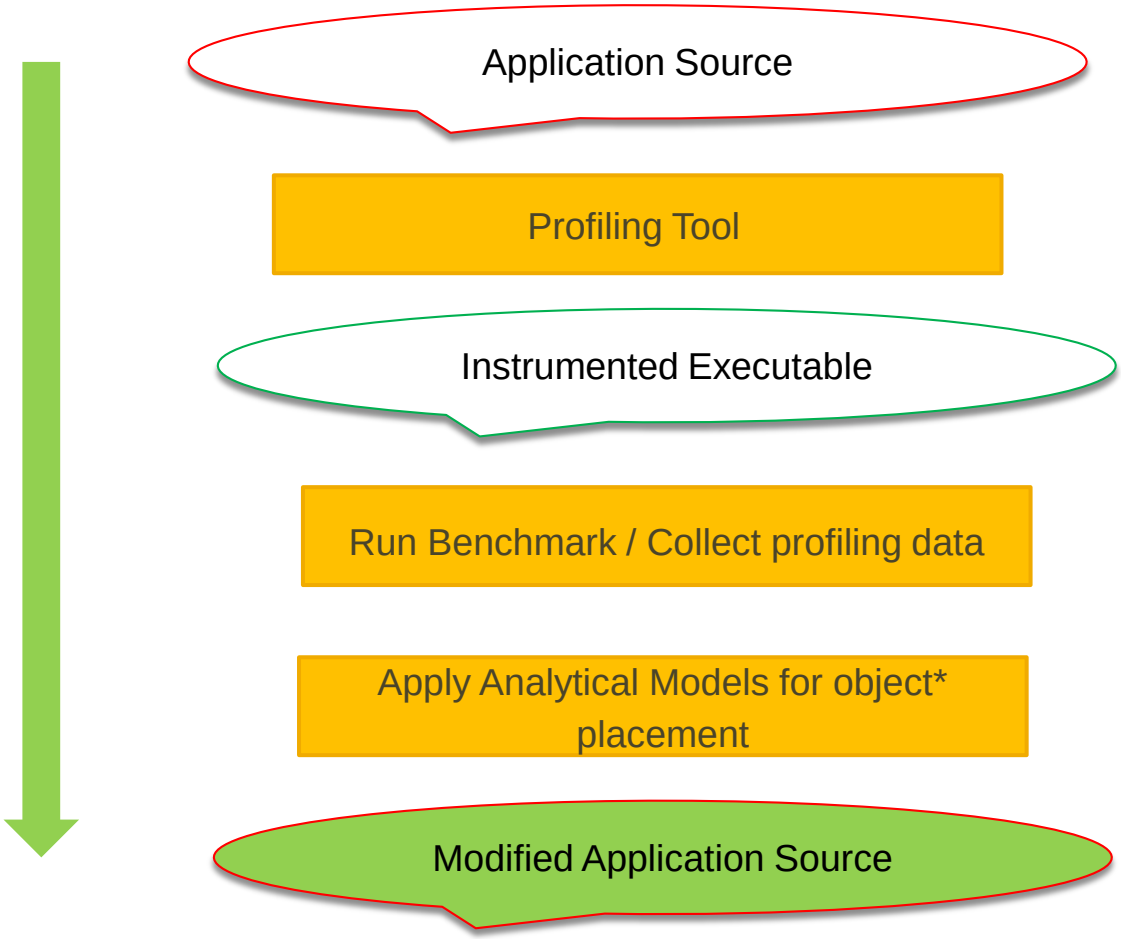
For such hybrid memory schemes, Application-level data management is useful – because it provides a hardware-independent way to manage data

- Data management on Hybrid memory at:
  1. Application Level ★★★
  2. Operating System Level
  3. Hardware Level

One key finding was that, objects presented more accurate granularity of data than pages

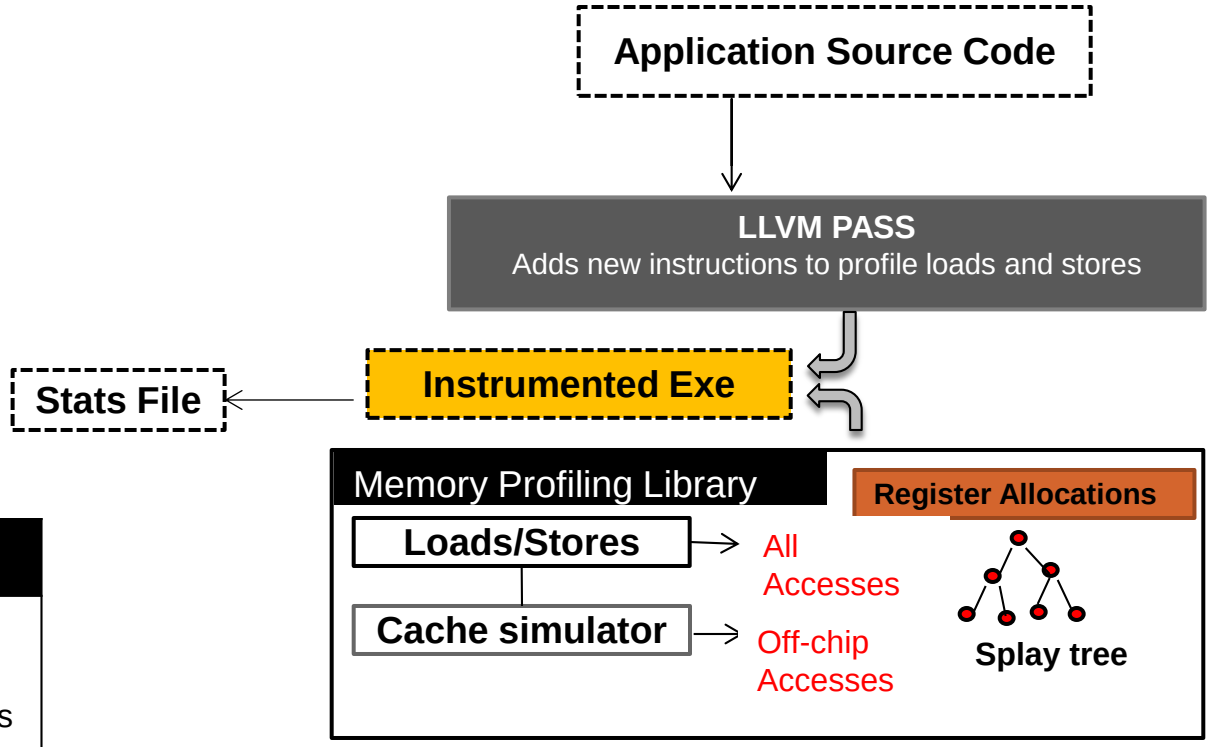
Methodology

Application Instrumentation



* Objects are individual program variables and memory allocations.

Profiling Tool



Collected Metric

Memory Loads
Memory Stores
Off-chip Memory accesses
Memory Allocations
Allocation sizes
Callpath
Lifetime

Performance and Energy Models

- Performance Model

$$AMAT_{DRAM} = \mu_r L_r + \mu_w L_w + (1 - \mu_r)\, L_{LLC}$$

μ_r and μ_w are number of main memory read and write accesses respectively, L_r and L_w are DRAM read and write latencies respectively and L_{LLC} is last level cache latency

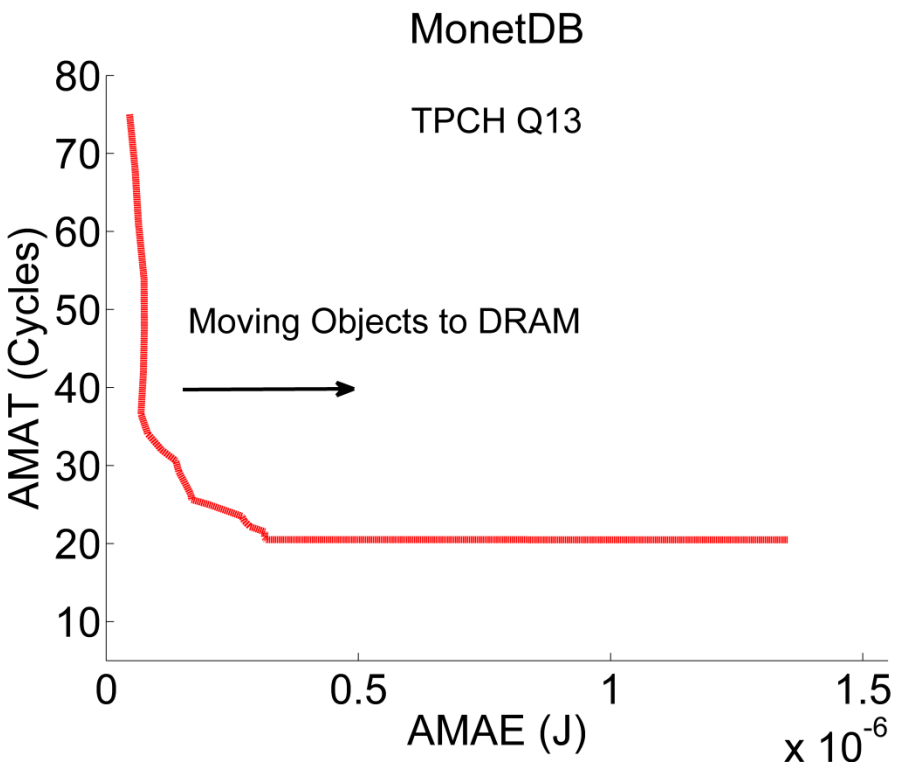
- Energy Model

$$AMAE_{DRAM} = \mu_r E_r + \mu_w E_w + S\, P_{DRAM} T$$

μ_r and μ_w are DRAM read and write access respectively. E_r and E_w are read and write energies respectively.

Object Placement Algorithm

1. $\Delta AMAE = AMAE_{DRAM} - AMAE_{NVM}$
2. $\Delta AMAT = AMAT_{DRAM} - AMAT_{NVM}$
3. Sort total objects on $\Delta AMAT$
4. $\sum_{i=s+1}^{N} \Delta AMAT \leq \lambda \sum_{i=1}^{N} AMAT_{DRAM}$



Where λ is a user-configurable parameter

Evaluation

Benchmarks and Simulation

Benchmarks

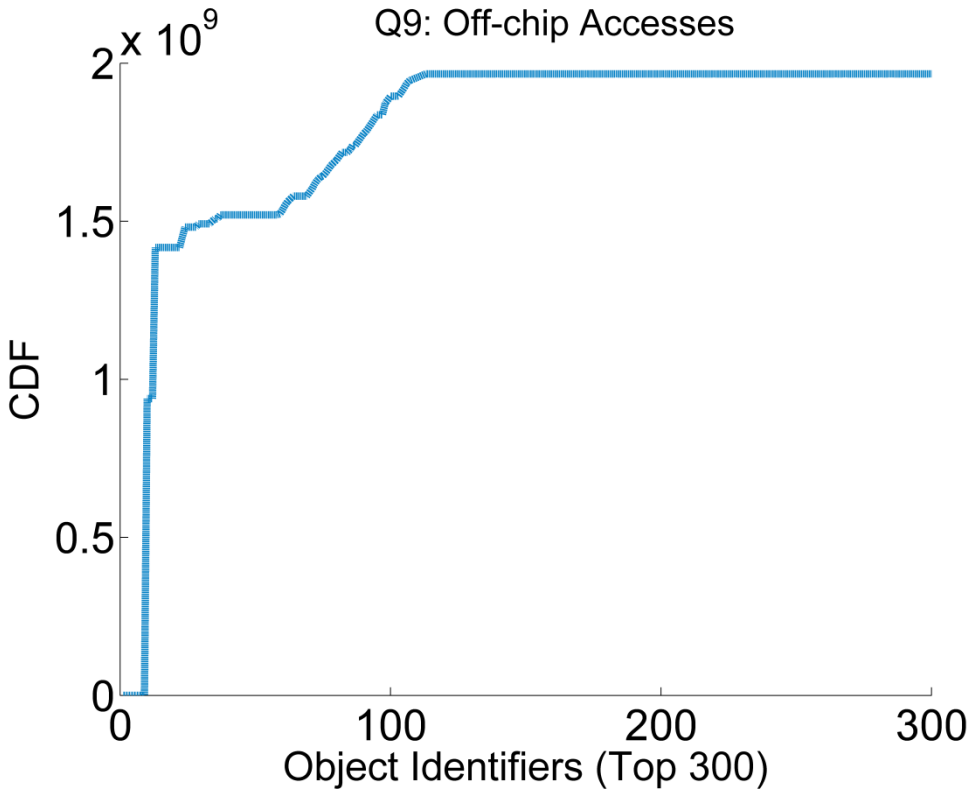
- MonetDB – In-memory column store
  - TPCH analytical queries
- Memcached – In-memory key-value store
  - Twitter and Yahoo Cloud Serving Benchmark

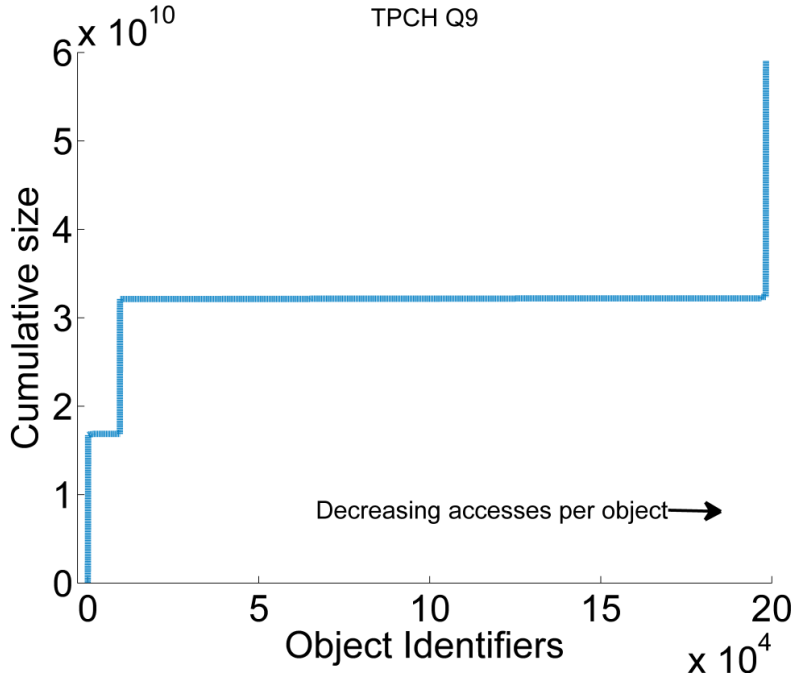
Simulation

- GEM5 Syscall emulation. 512 MB DRAM, 8GB RRAM
- Custom application-level memory allocators for DRAM and RRAM

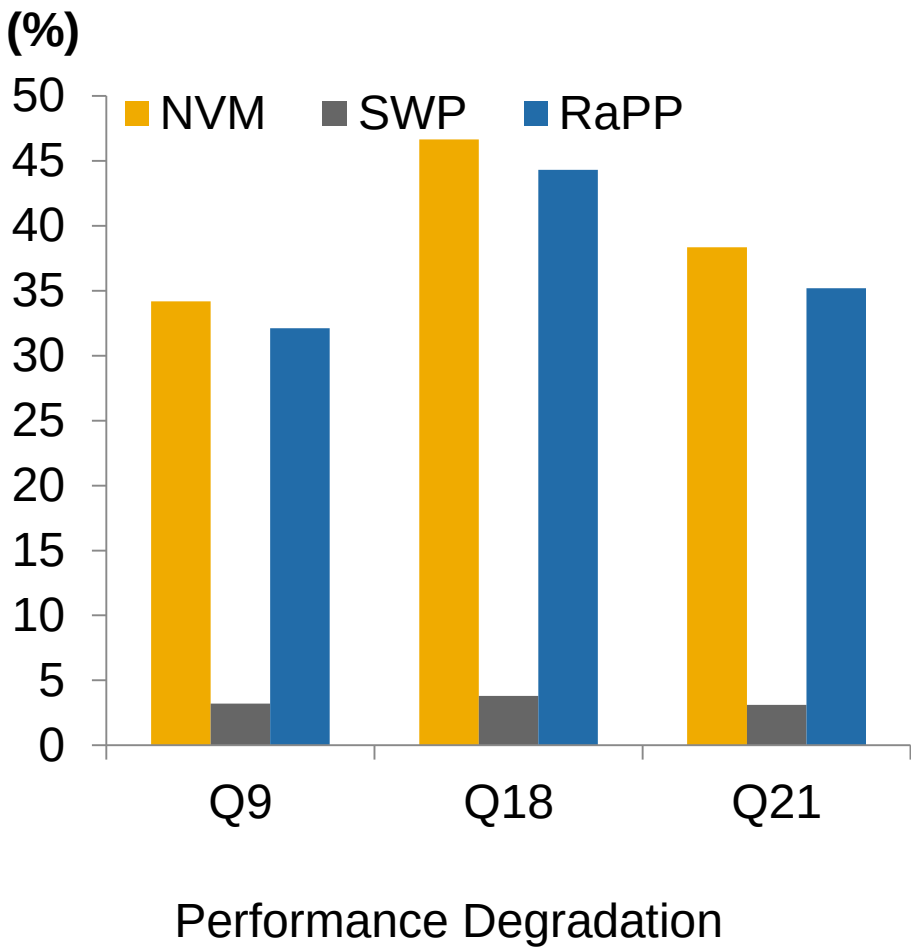
MonetDB Analysis

1.

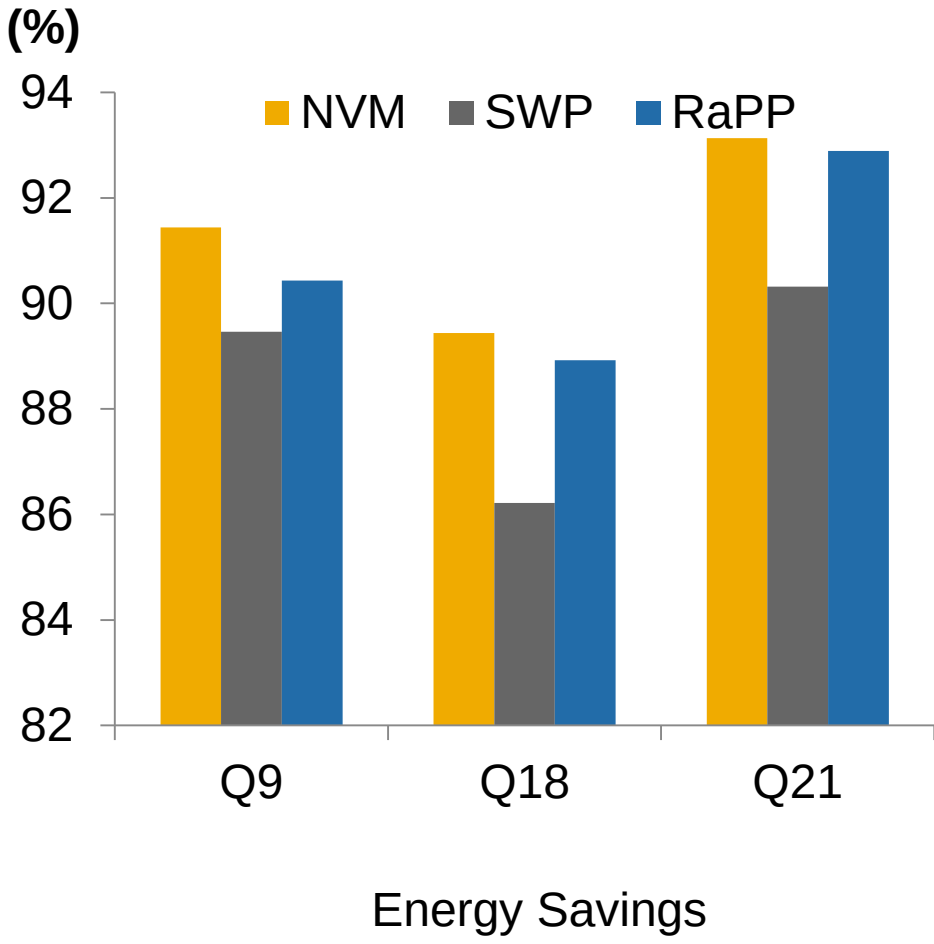




MonetDB: Performance Degradation vs Energy Savings



Performance Degradation



Energy Savings

Conclusion

- Use of NVM as main memory is inevitable for meeting main memory capacity demands.
- Application-level data management provides a hardware independent way to manage data on hybrid memories.
- For the workloads we studied, objects provide better granularity than pages for data management on hybrid memory.
- Hybrid DRAM / NVM main memory found promising for in-memory data stores.
- Future work on dynamic data placement techniques through operator level rules.

Acknowledgements

- Nanostreams Project (http://www.nanostreams.eu)
- NovoSoft Project (http://www.qub.ac.uk/research-centres/HPDC/Articles/EUMarieCurieFellowshipNovosoft/)

Thank you!

Contact information:

Ahmad Hassan
ahmad.hassan@sap.com