Adaptive Execution on 3D Microprocessors

Koji Inoue

Kyushu University

Outline

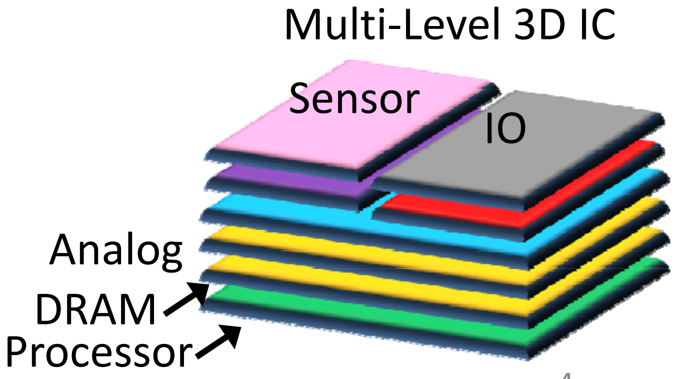
- Why 3D?
- Will 3D always work well?
- Support Adaptive Execution!
  - Memory Hierarchy Run-time Optimization
- Conclusions

Outline

- Why 3D?
- Will 3D always work well?
- Support Adaptive Execution!
  - Memory Hierarchy Run-time Optimization
- Conclusions

From 2D to 3D! (not only TV)

- Stack Multiple Dies
- Connect Dies with Through Silicon Vias



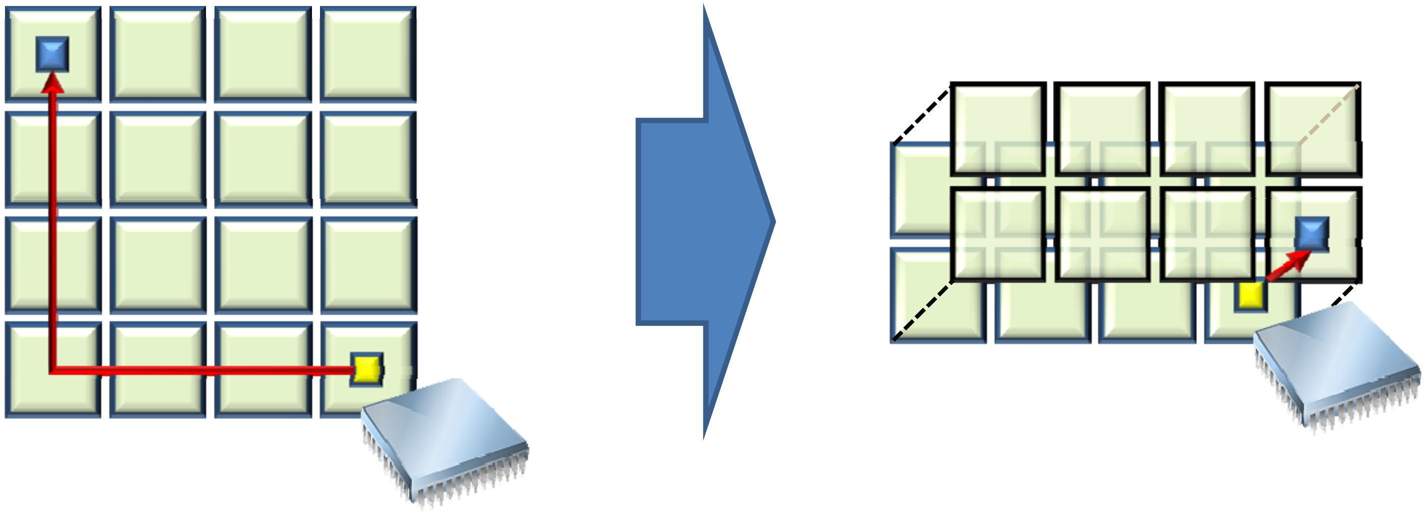
Chip Implementation Examples from ISSCC'09

- Image Sensors
- SRAM for SoCs
- DRAM
- Multi-core + SRAM connected with wireless TSVs

U. Kang et al., "8Gb DDR3 DRAM Using Through-Silicon-Via Technology," ISSCC'09.
H. Saito et al., "A Chip-Stacked Memory for On-Chip SRAM-Rich SoCs and Processors, " ISSCC'09.
V. Suntharalingam et al., "A 4-Side Tileable Back Illuminated 3D-Integrated Mpixel CMOS Image Sensor," ISSCC'09.
K. Niitsu et al., "An Inductive-Coupling Link for 3D Integration of a 90nm CMOS Processor and a 65nm CMOS SRAM," ISSCC'09.

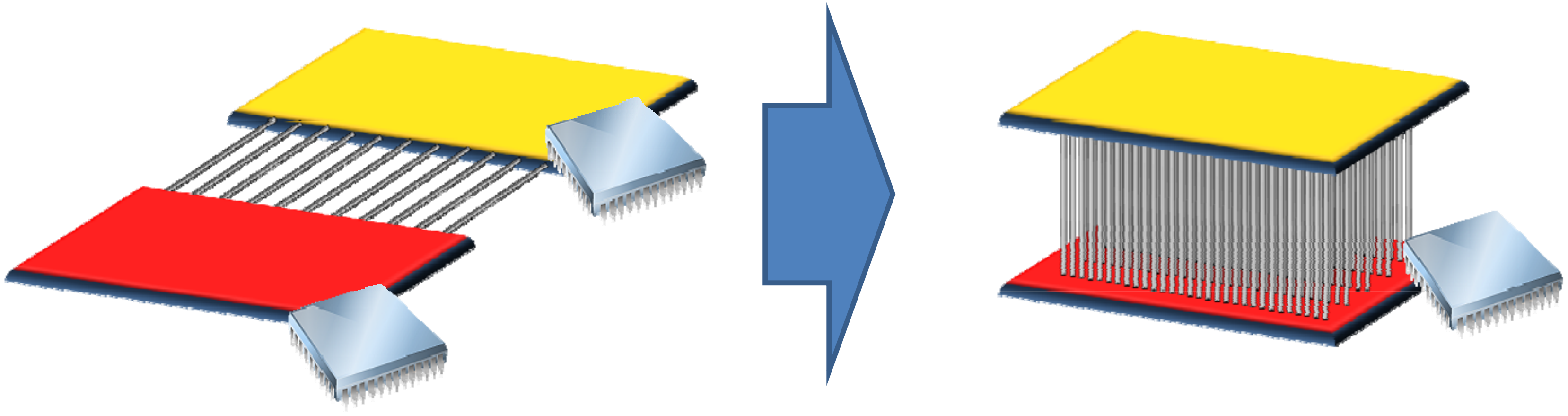
Why 3D? (1/2)

- Wire Length Reduction
  - Replace long, high capacitance wires by TSVs
  - Low Latency, Low Energy
- Small footprint



Why 3D? (2/2)

- Integration
  - From “Off-Chip” to “On-Chip”
  - Improved Communication
    - Low Latency, High Bandwidth, and Low Energy
  - Heterogeneous Integration
    - E.g. Emerging Devices



Outline

- Why 3D?
- Will 3D always work well?
- Support Adaptive Execution!
  - Memory Hierarchy Run-time Optimization
- Conclusions

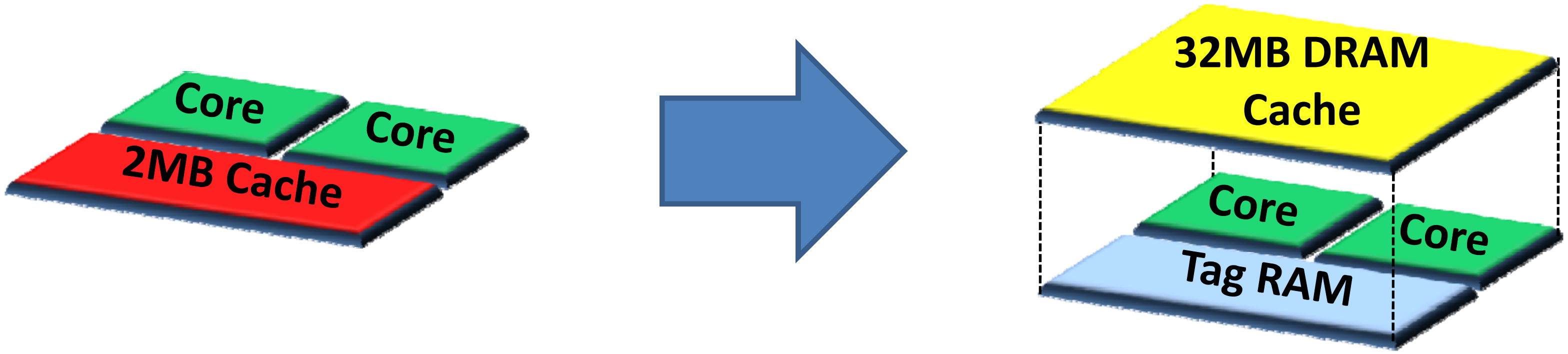
Importance of On-Chip Caches

- Memory-Wall Problem
  - Memory bandwidth does not scale with the # of cores
  - Growing speed gap between processor cores and DRAMs
  - So, Becomes more serious
- Let's increase on-chip cache capacity, but…
  - Requires large chip area

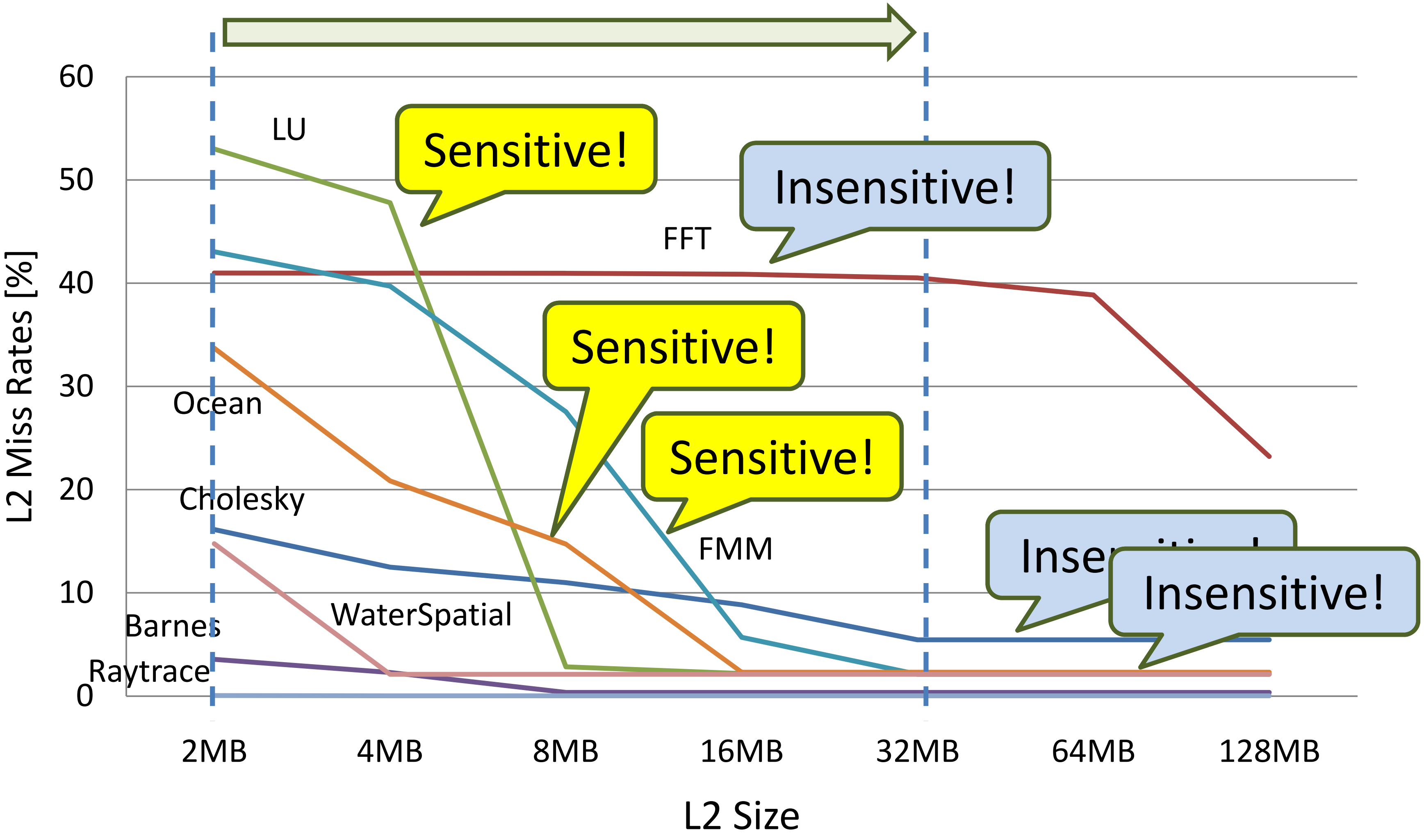
Will 3D always work well?
“Stacking a DRAM Cache”

$$\underset{\text{Ave. Memory Acc. Time}}{AMAT} = \overset{\text{L1 Hit Time}}{HT_{L1}} + \overset{\text{L1 Miss Rate}}{MR_{L1}} \times \left(\overset{\text{L2 Hit Time}}{HT_{L2}} + \overset{\text{L2 Miss Rate}}{MR_{L2}} \times \overset{\text{Main-Memory Access Time}}{MMAT} \right)$$

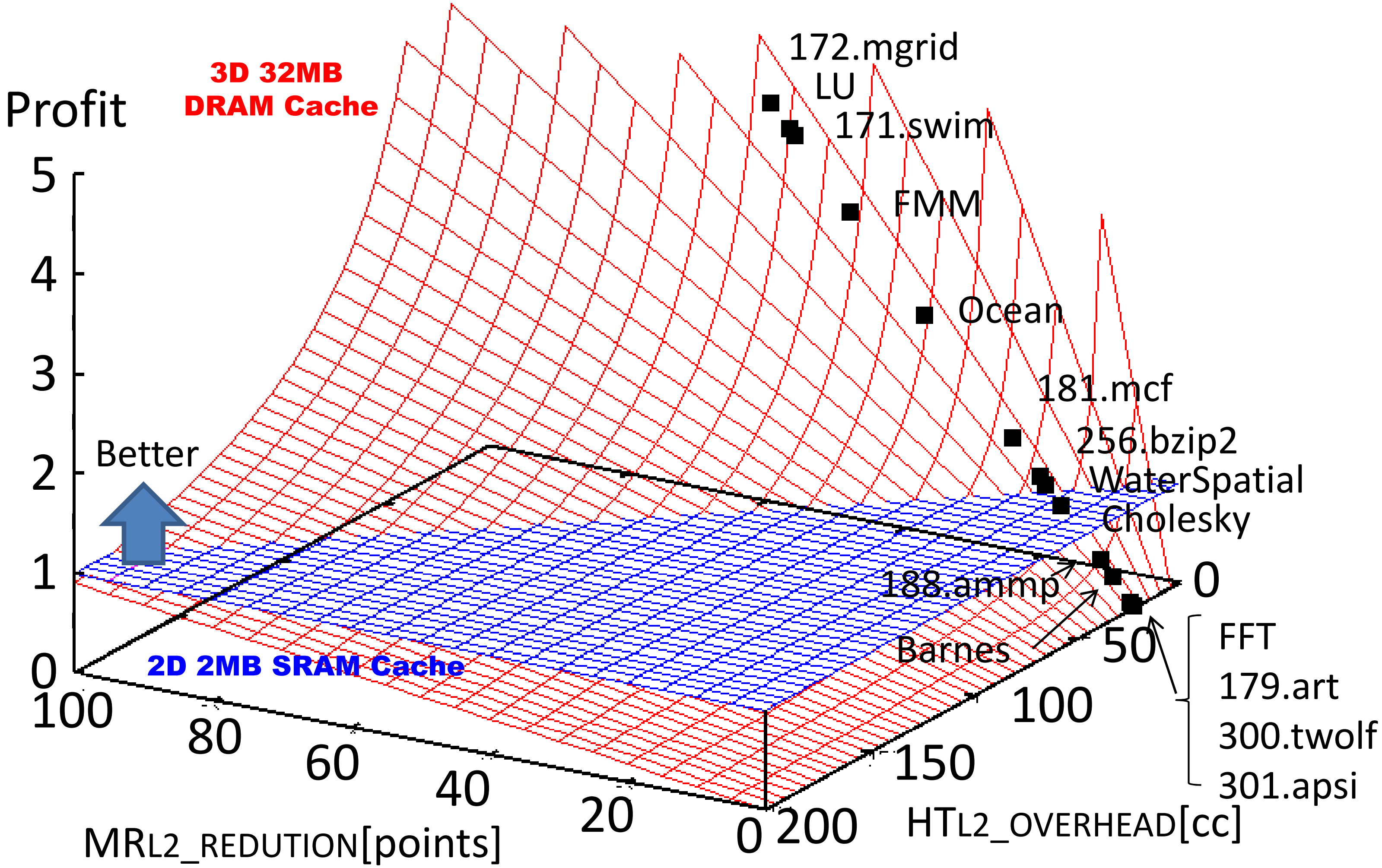
Impact of DRAM Stacking	→	→	↗	↘ **?**	→



Cache-Size Sensitivity Varies among Programs!

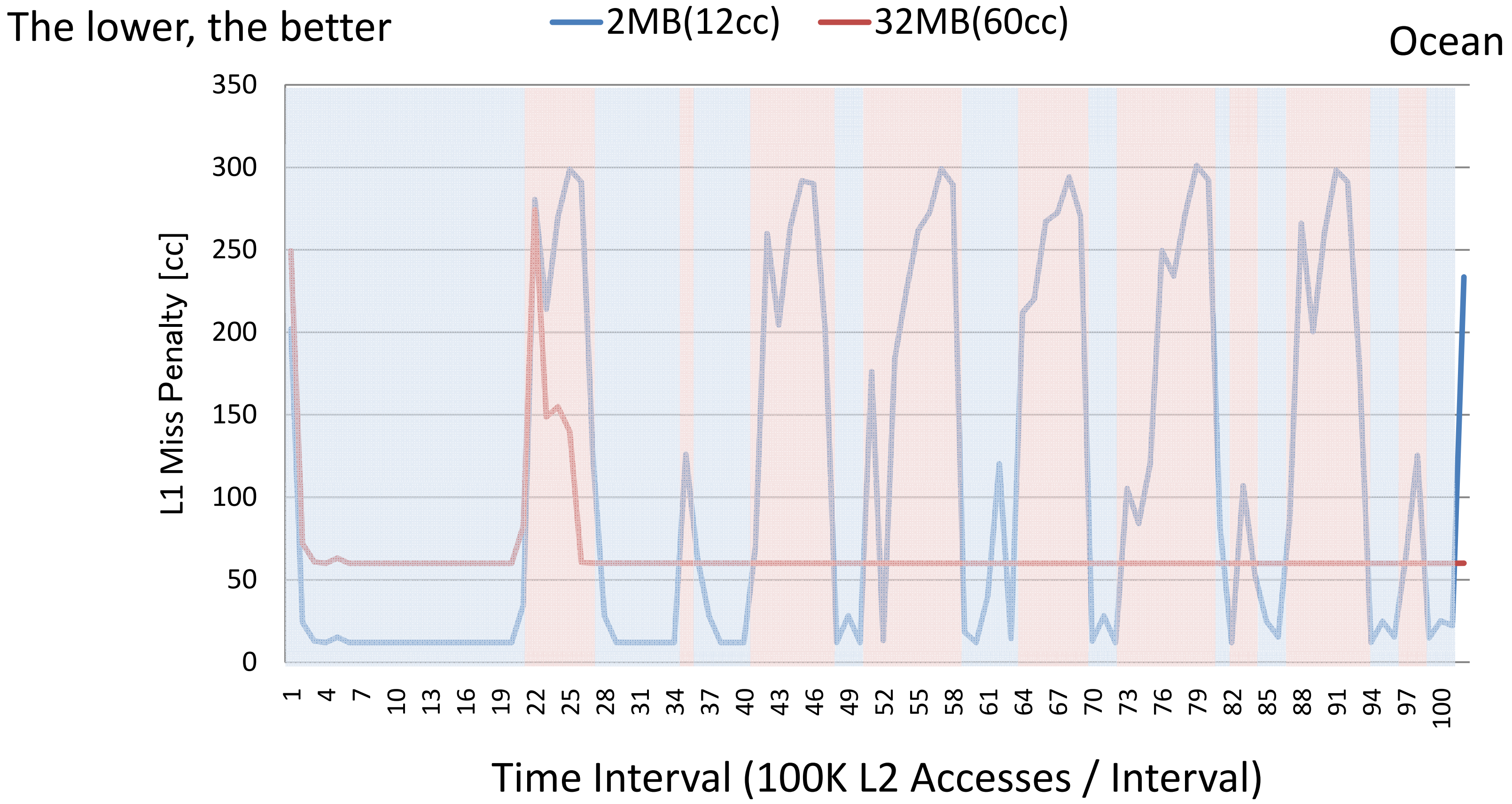


2D vs. 3D



$$Profit = \frac{MR_{L2\_REDUCTION} \times MMAT}{HT_{L2\_OVERHEAD}}$$

Appropriate Cache Size Varies within Programs!



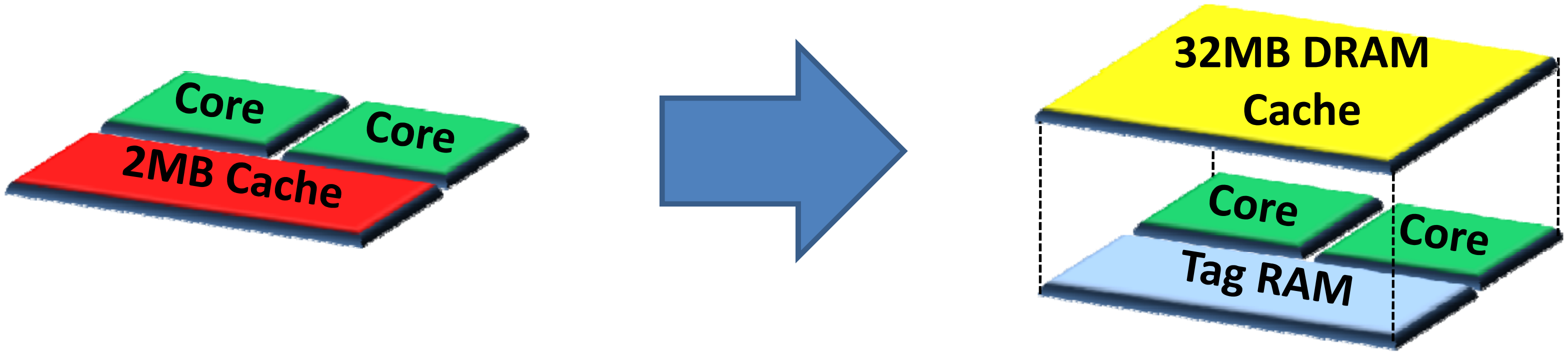
Outline

- Why 3D?
- Will 3D always work well?
- Adaptive Execution!
  - Memory Hierarchy Run-time Optimization
- Conclusions

Will 3D always work well?
"Stacking a DRAM Cache"

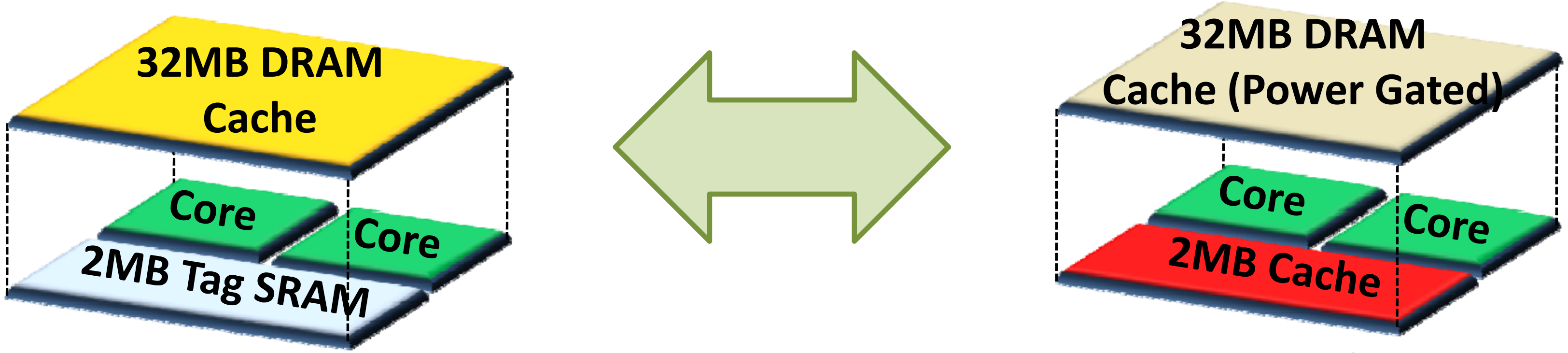
$$\underset{\text{Ave. Memory Acc. Time}}{AMAT} = \overset{\text{L1 Hit Time}}{HT_{L1}} + \overset{\text{L1 Miss Rate}}{MR_{L1}} \times (\overset{\text{L2 Hit Time}}{HT_{L2}} + \overset{\text{L2 Miss Rate}}{MR_{L2}} \times \overset{\text{Main-Memory Access Time}}{MMAT})$$

Impact of DRAM Stacking	→	→	↗	↘ ?	→



SRAM/DRAM Hybrid Cache Architecture

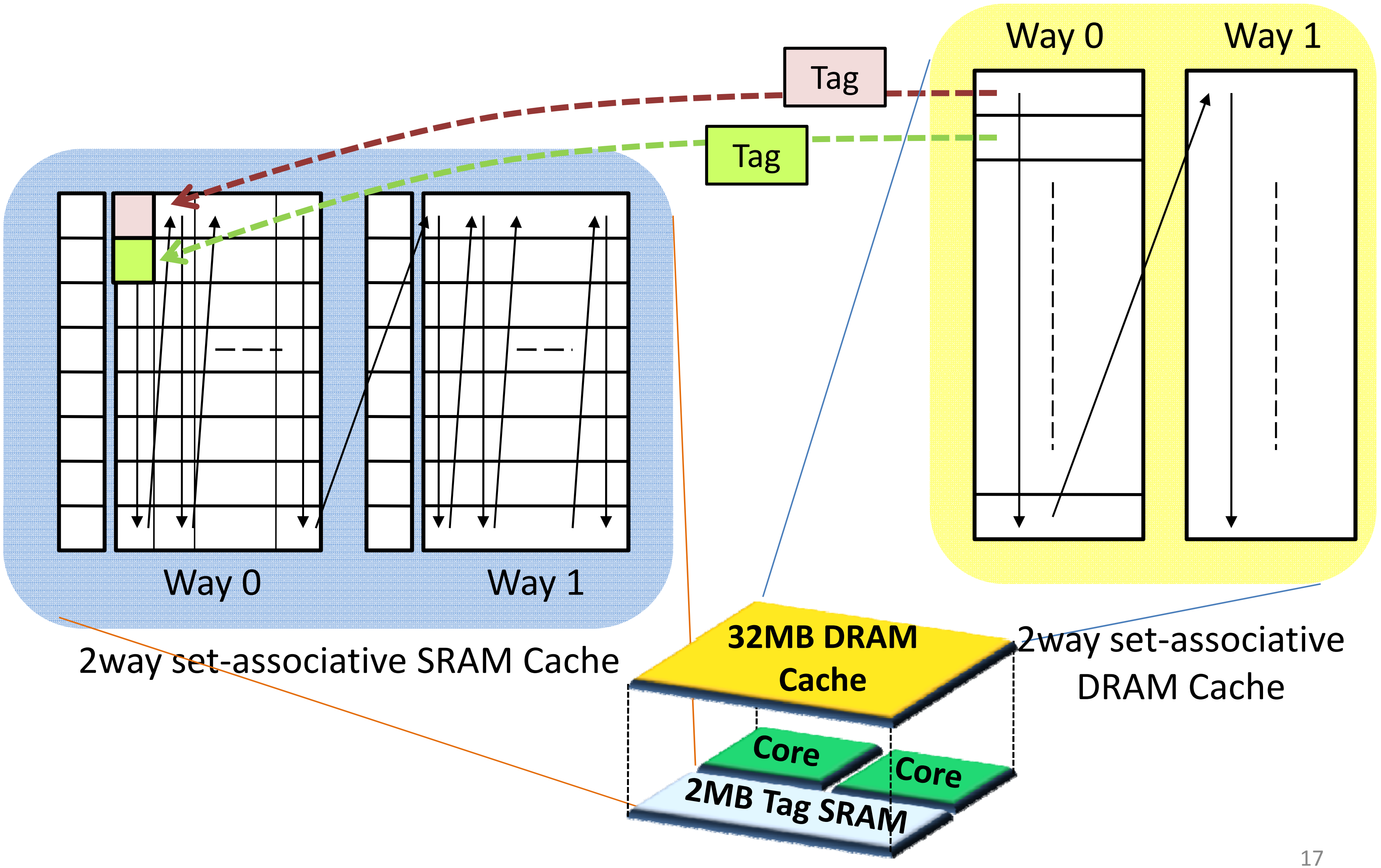
- Support Two Operation Modes
  - High-Speed, Small Cache Mode (or SRAM Cache Mode)
  - Low-Speed, Large Cache Mode (or DRAM Cache Mode)
- Adapt to variation of application behavior



DRAM Cache Mode

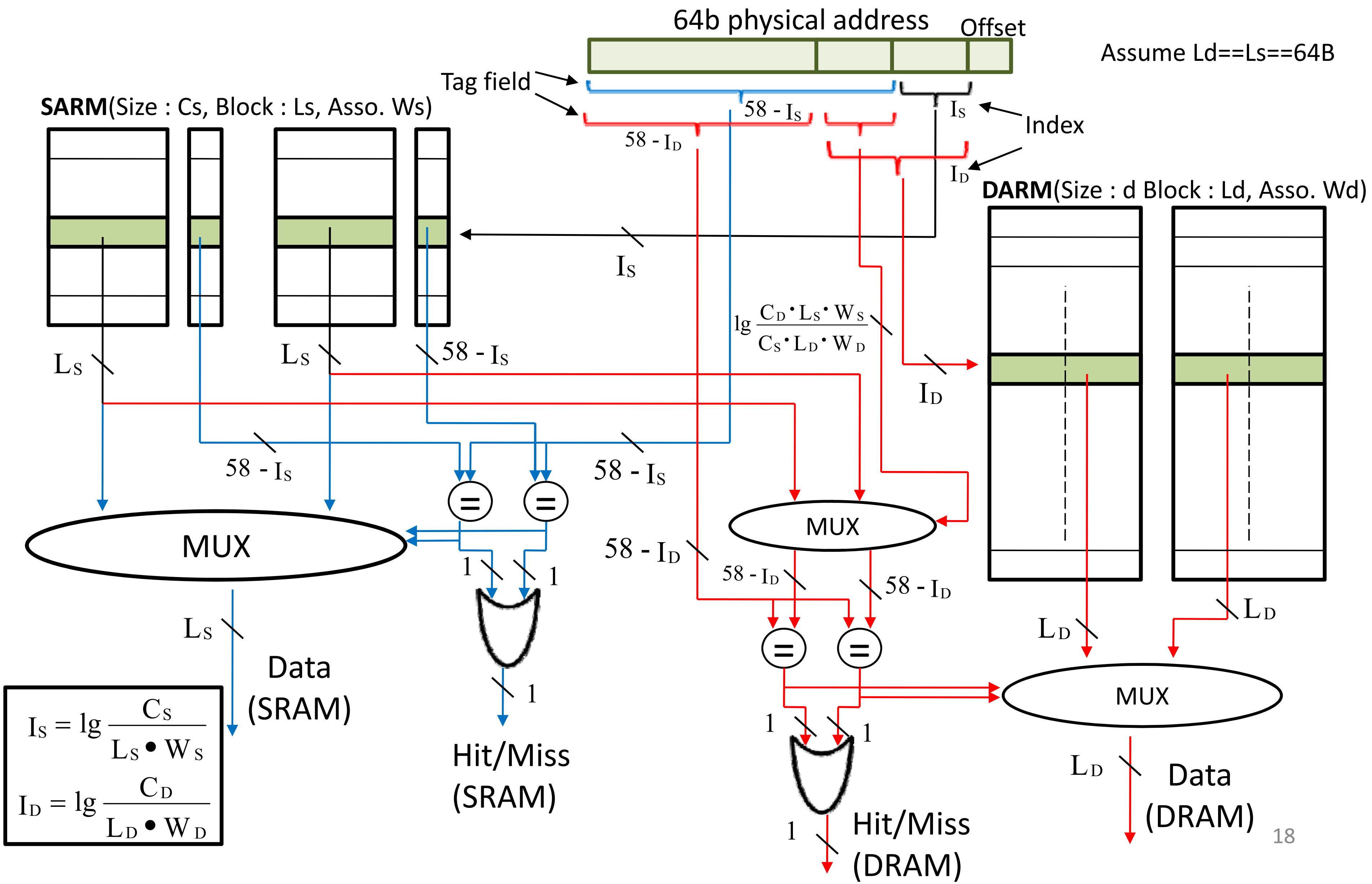
SRAM Cache Mode

Microarchitecture (1/2)


Tag
Tag
Way 0
Way 1
Way 0
Way 1
2way set-associative SRAM Cache
2way set-associative DRAM Cache
32MB DRAM Cache
Core
Core
2MB Tag SRAM

Microarchitecture (2/2)

Assume Ld==Ls==64B

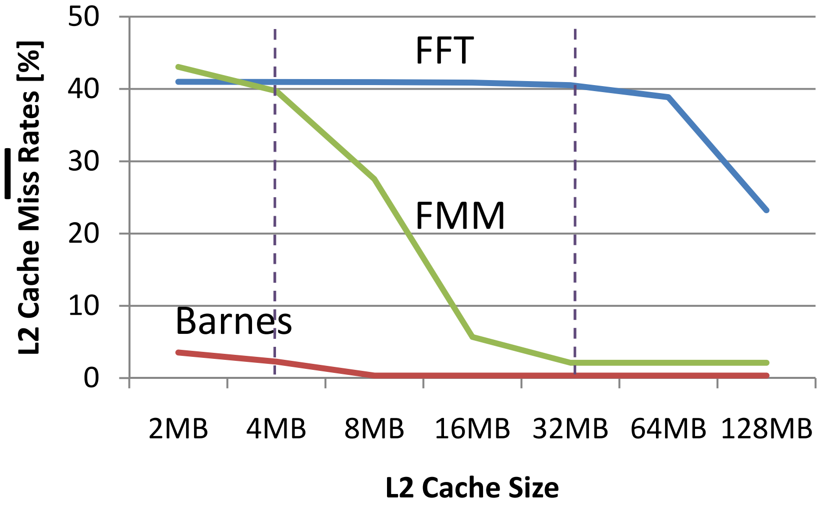


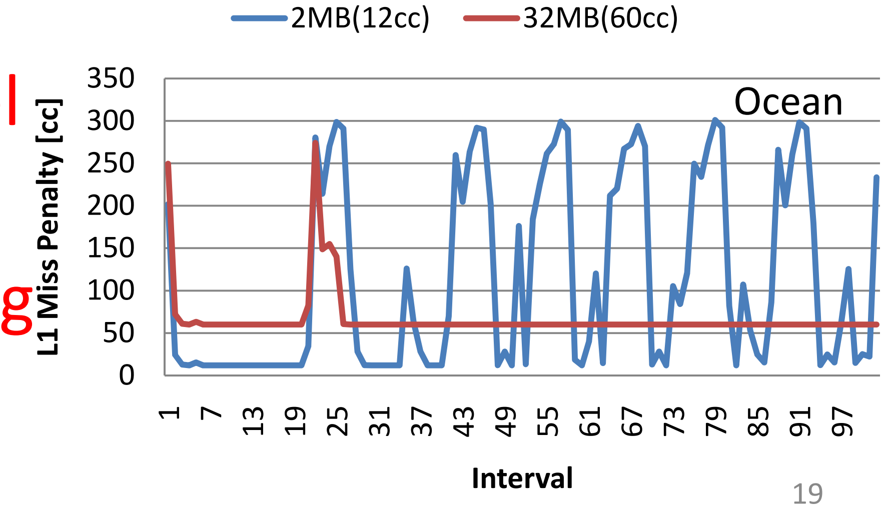
$$I_S = \lg \frac{C_S}{L_S \bullet W_S}$$

$$I_D = \lg \frac{C_D}{L_D \bullet W_D}$$

How to Adapt

- Static Approach
  - Optimizes at program level
  - Does not change it during execution
  - Needs a static analysis
- Dynamic Approach
  - Optimizes at interval level (or phase level)
  - Needs a run-time profiling





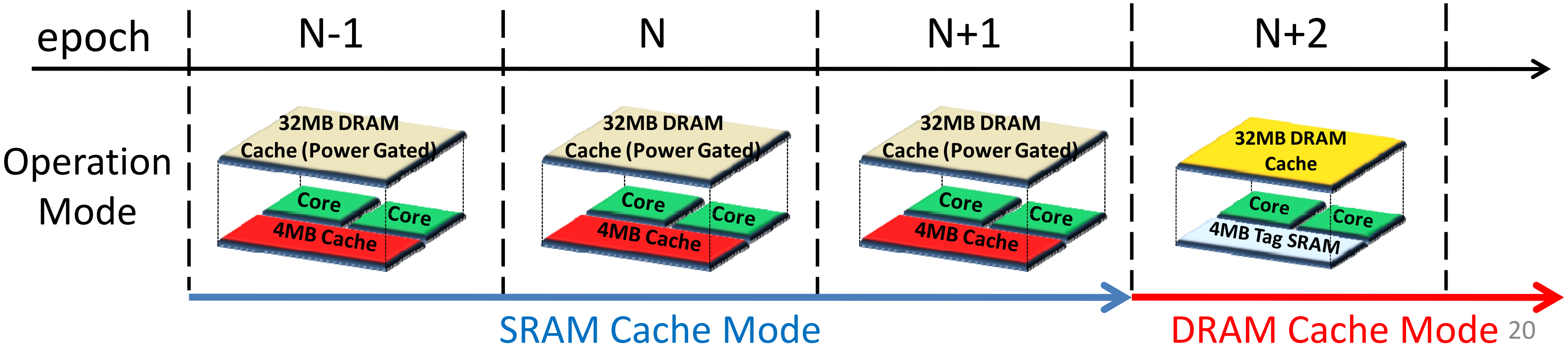
Run-Time Mode Selection

- Divide Program Execution into “epochs”, e.g. 200K L2 Misses
- Predict an Appropriate Operation Mode for Next Epoch
- On SRAM mode, a small tag RAM which stores sampled tags is used to predict DRAM mode miss rates

Hardware Support for Measurement

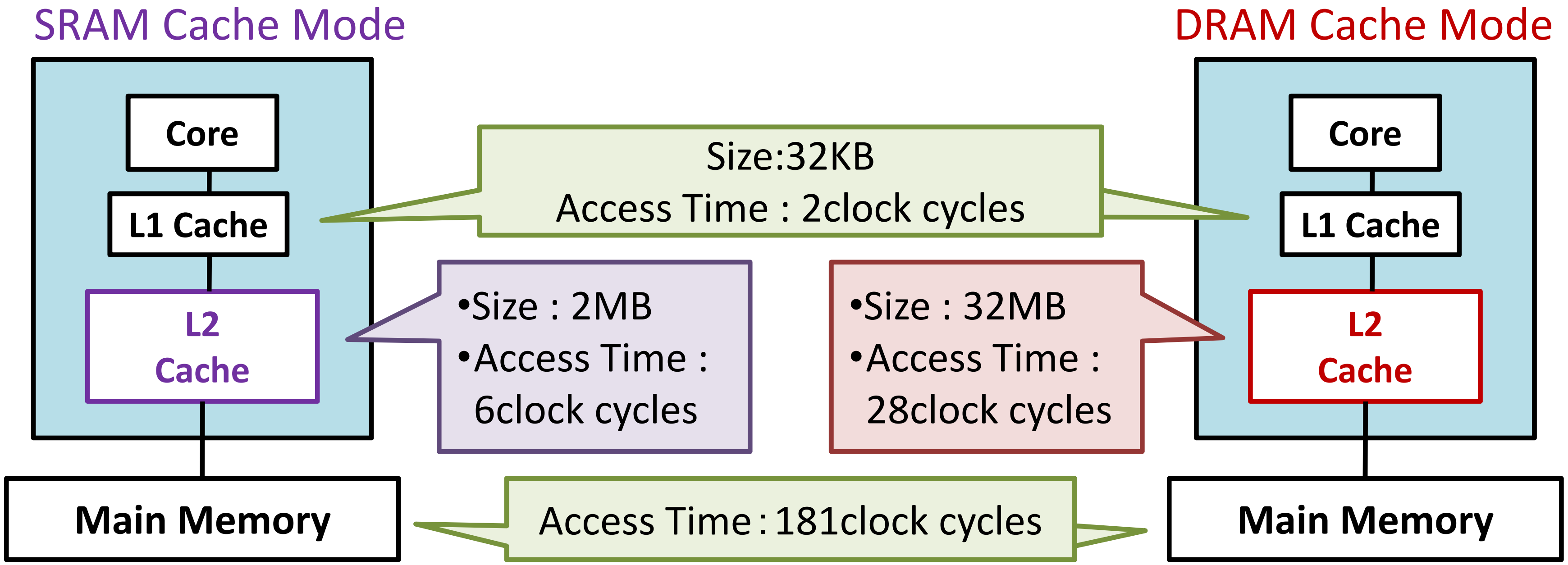
$$\mathit{if}\ \ MR_{L2SRAM} - MR_{L2DRAM} > \frac{HT_{L2DRAM} - HT_{L2SRAM} + AveOverhead}{MMAT}$$

then transit from SRAM mode to DRAM mode!

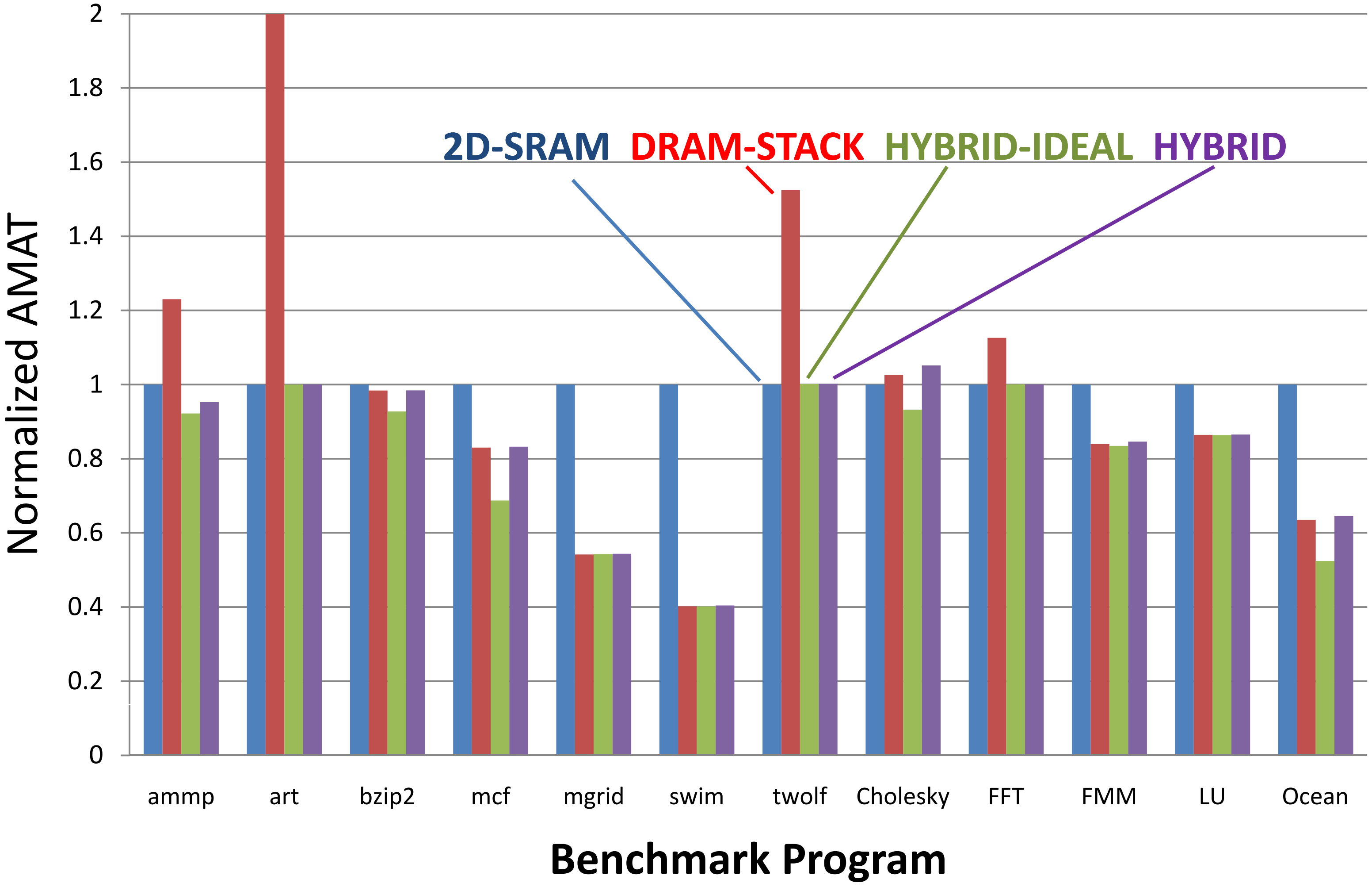


Experimental Set Up

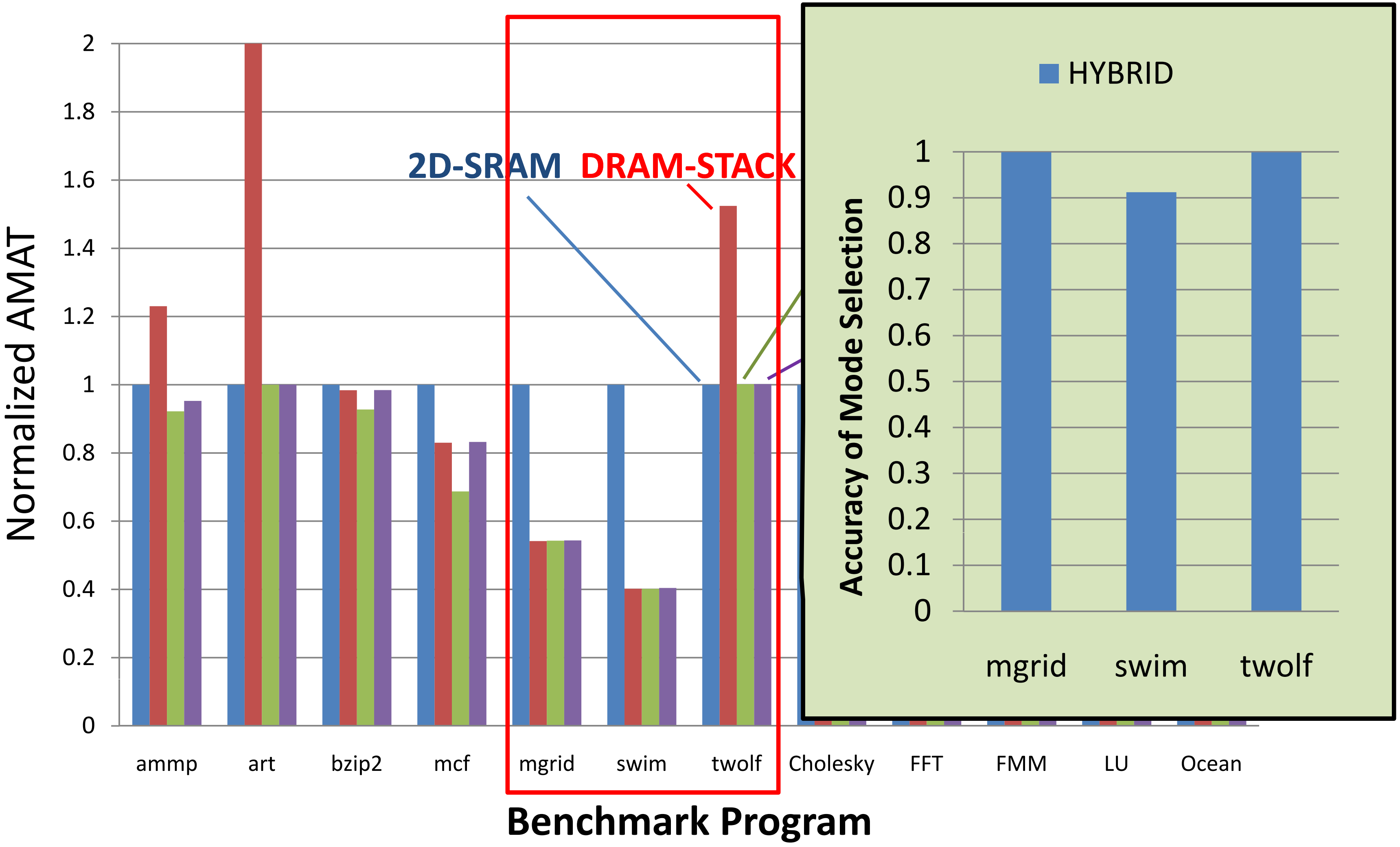
- Processor: In-Order
- Benchmarks: SPEC CPU 2000, Splash2



Results

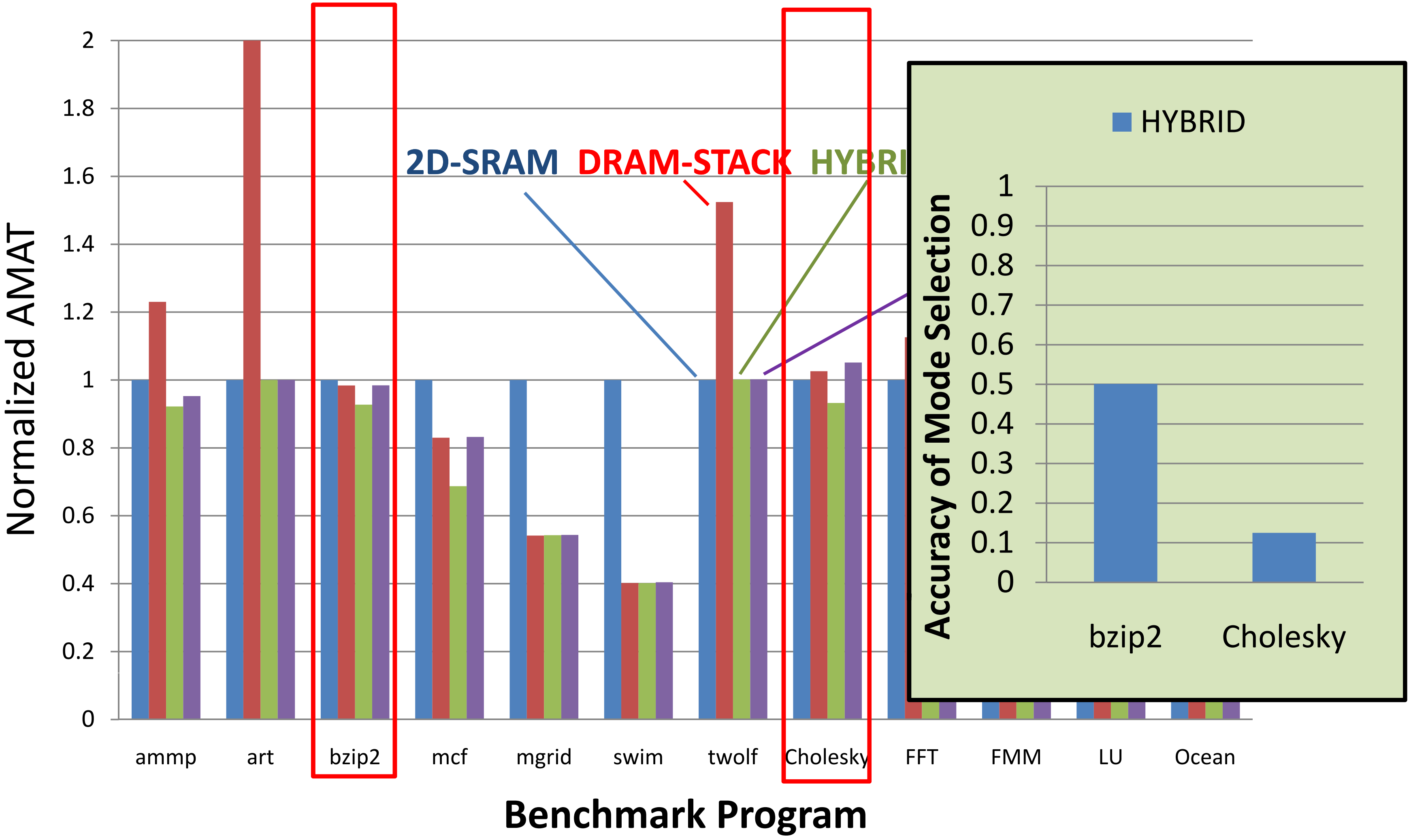

2D-SRAM
DRAM-STACK
HYBRID-IDEAL
HYBRID
Normalized AMAT
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
ammp
art
bzip2
mcf
mgrid
swim
twolf
Cholesky
FFT
FMM
LU
Ocean
Benchmark Program

Results



Normalized AMAT
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
2D-SRAM
DRAM-STACK
ammp
art
bzip2
mcf
mgrid
swim
twolf
Cholesky
FFT
FMM
LU
Ocean
Benchmark Program
HYBRID
Accuracy of Mode Selection
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
mgrid
swim
twolf

Results


2D-SRAM
DRAM-STACK
HYBRI
Normalized AMAT
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
ammp
art
bzip2
mcf
mgrid
swim
twolf
Cholesky
FFT
FMM
LU
Ocean
Benchmark Program
HYBRID
Accuracy of Mode Selection
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
bzip2
Cholesky

Conclusions

- The 3D solution is one of the most promising ways to achieve…
  - High performance
  - Low energy
- It does not ALWAYS work well!
- Run-time adaptive execution by considering memory access behavior

Acknowledgement

- This research was supported in part by New Energy and Industrial Technology Development Organization