

Elastic Cooperative Caching:

An Autonomous Dynamically Adaptive Memory
Hierarchy for Chip Multiprocessors

Enric Herrero¹, José González², Ramon Canal¹

¹Universitat Politècnica de Catalunya

²Intel Barcelona



UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Outline

- Motivation
- Related Work
- Elastic Cooperative Caching
- Evaluation
- Conclusions

Motivation

- Find optimal cache organization for tiled microarchitectures
 - Desired behavior
 - Scalable
 - Minimize access latency
 - Minimize inter-thread interference
 - Minimize off-chip misses
- ↗ Avoid centralized structures.
- ↗ Data placement based on proximity.
- Private cache partitions.
- Dynamic cache allocation.

Motivation



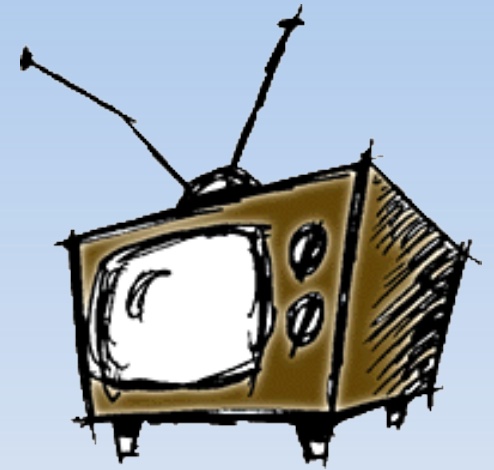
- Application Taxonomy
 - Saturating Utility
 - Low Utility
 - Shared High Utility
 - Private High Utility

Extended classification from Qureshi et al. [MICRO'06]

Related Work

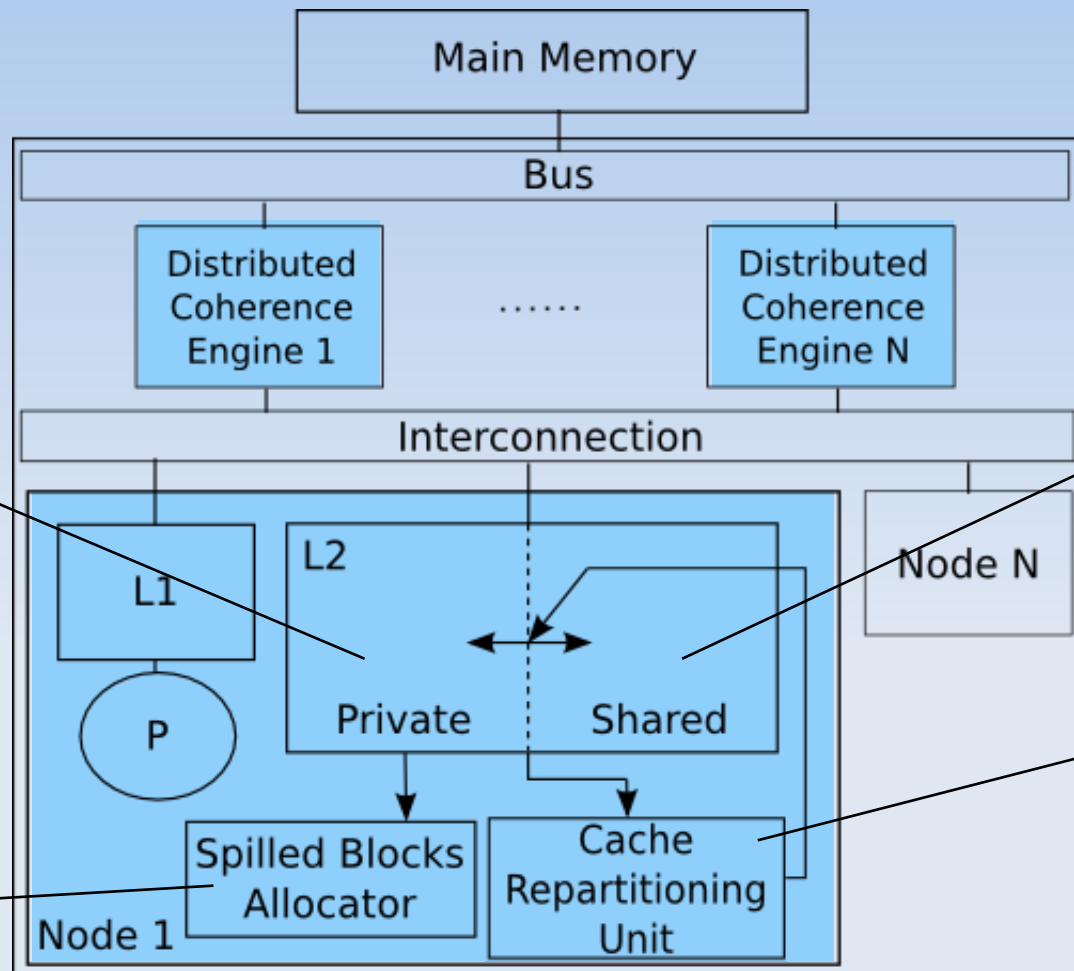
- Reactive NUCA [ISCA'09]
- Adaptive Selective Replication [MICRO'06]
- Adaptive Shared/Private NUCA [HPCA'07]

- OS-page granularity.
- Software based.
- Common shared cache space.
- Adjusts replication but not amount of cache per node.
- Centralized structures.



More: Athena
Award Lecture
Mary Jane Irwin

Elastic Cooperative Caching – Structure



Herrero et al.
[PACT'08]

Only local core
can allocate

Distributes
evicted blocks
from private
partition among
nodes.

Allocates
evicted blocks
from all private
regions

Every N cycles
repartitions
cache based on
LRU hits in S&P
partitions.

Elastic Cooperative Caching – Adaptive Spilling

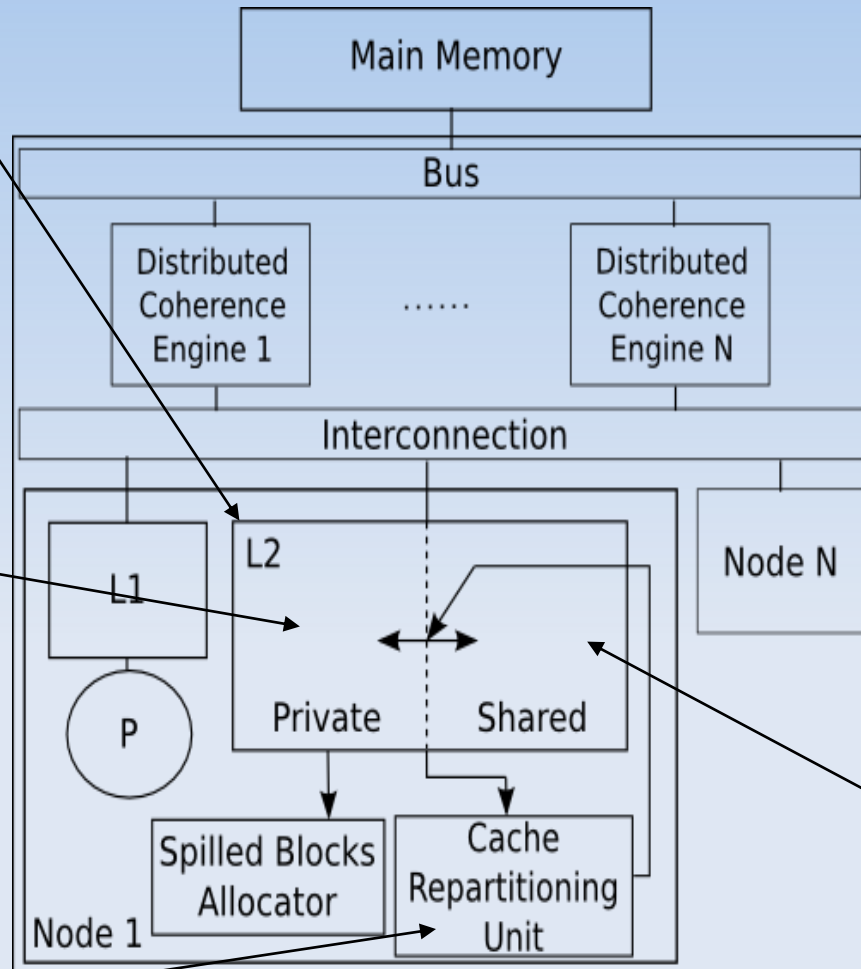
- **ElasticCC opportunity:** Not only repartition but also decide which nodes can use shared partitions.

Type	Working Set Size	Sharing	Local Reuse	Private Cache Size	Spilling
Saturating Utility	Small/Medium	H/L	H/L	Small/Medium	No
Low Utility	Big	Low	Low	Small	No
Shared High Utility	Big	High	H/L	Small	Yes
Private High Utility	Big	Low	High	Big	Yes

Spill shared blocks or blocks from caches with 75% or more private cache space

Elastic Cooperative Caching – Structure

Distributed cache among nodes.
Local allocation.



Private Regions.

Independent local repartitioning units.

- Desired behavior
 - Scalable
 - Minimize access latency
 - Minimize inter-thread interference
 - Minimize off-chip misses

Cache Partitioning.
Dynamic Cache Allocation.

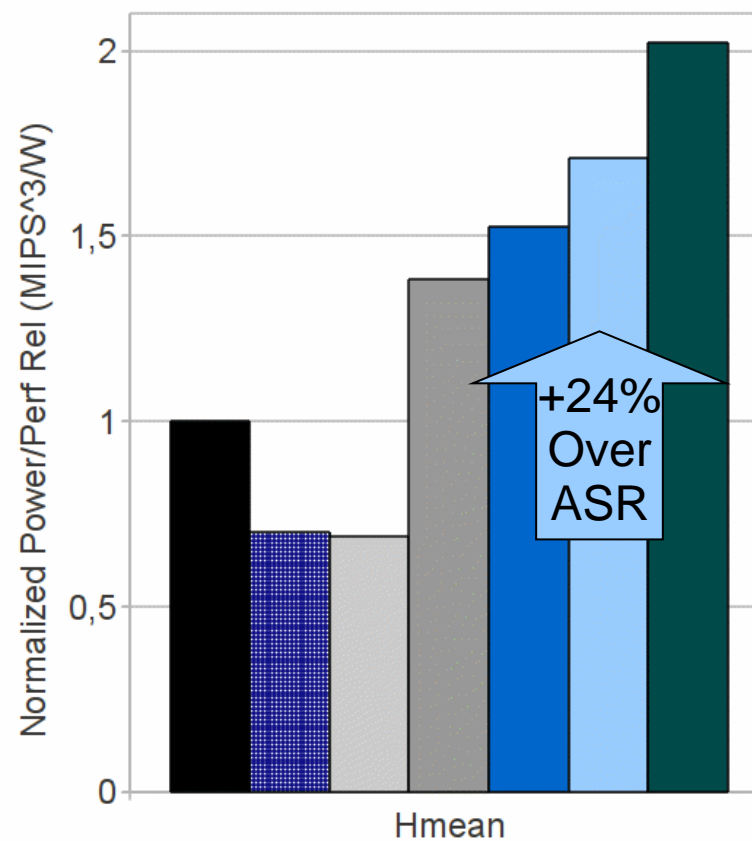
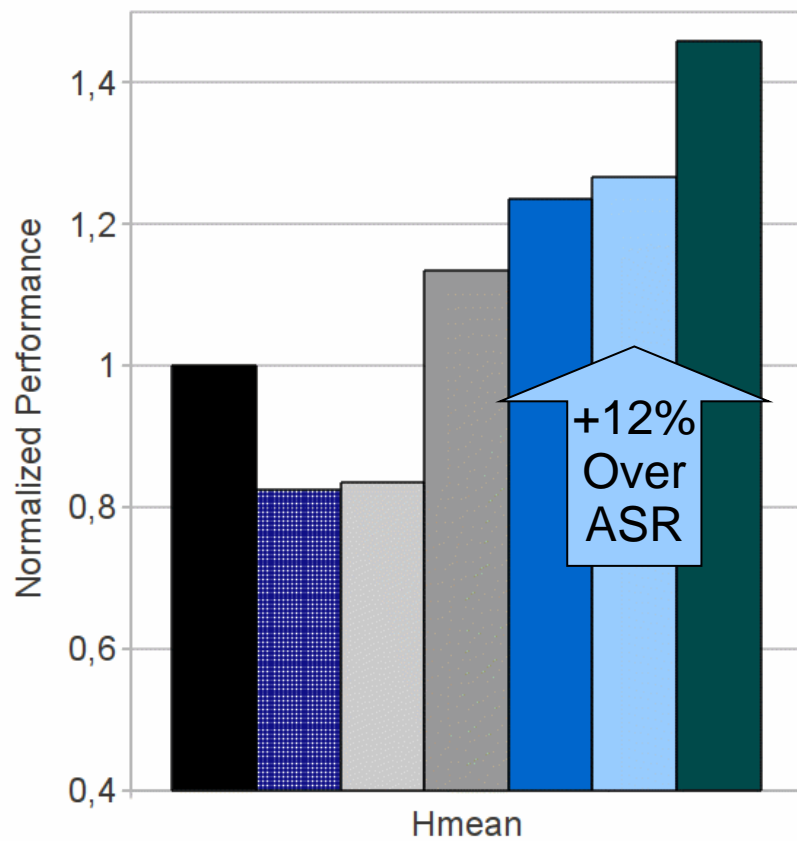


Evaluation – Studied Configurations

- **16 Processors**
- **Pairs of SPEC OMP'01 benchmarks of each of previous categories.**
- **Configurations**
 - **Shared Memory**
 - **Private Memory**
 - **Distributed Cooperative Caching (DCC)**
 - **Adaptive Selective Replication (ASR)**
 - **Elastic Cooperative Caching**
 - **ElasticCC + Adaptive Spilling**
 - **Ideal: Fixed Half Private/Half Shared 2xL2**

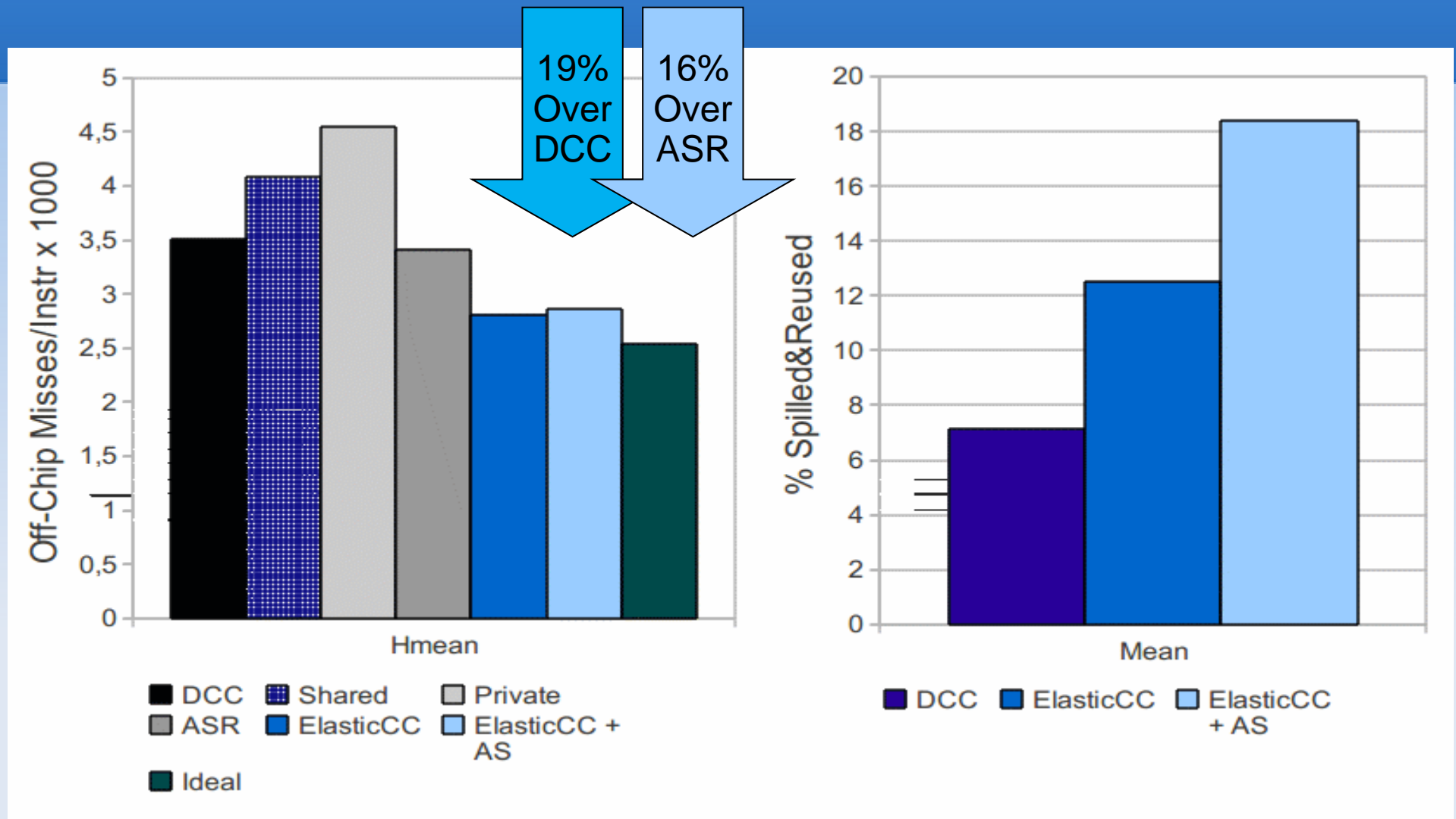


Evaluation – Performance & Efficiency



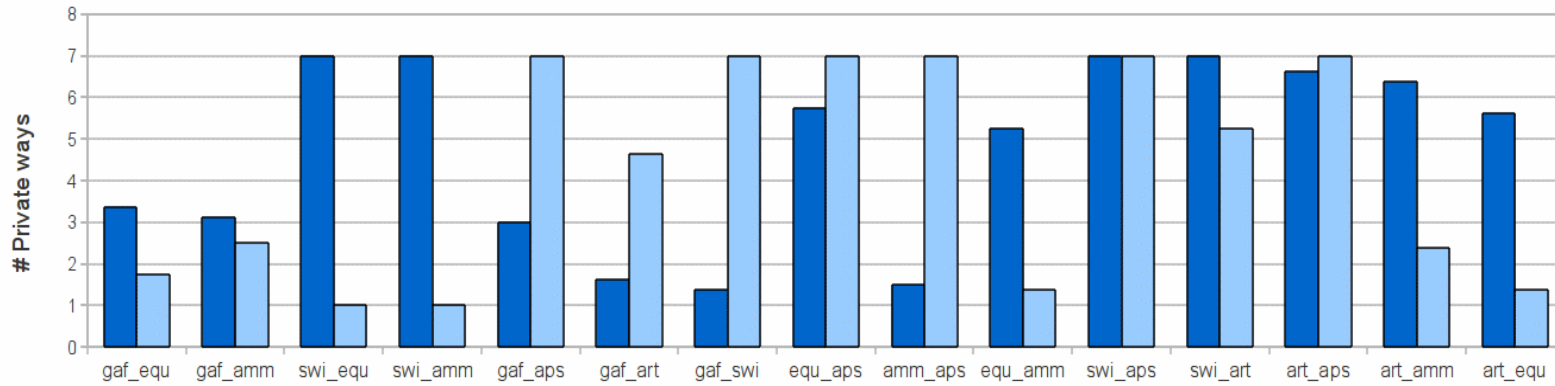
■ DCC ■ Shared ■ Private ■ ASR ■ ElasticCC ■ ElasticCC + ASR ■ Ideal

Evaluation – Off-Chip Misses & Reuse

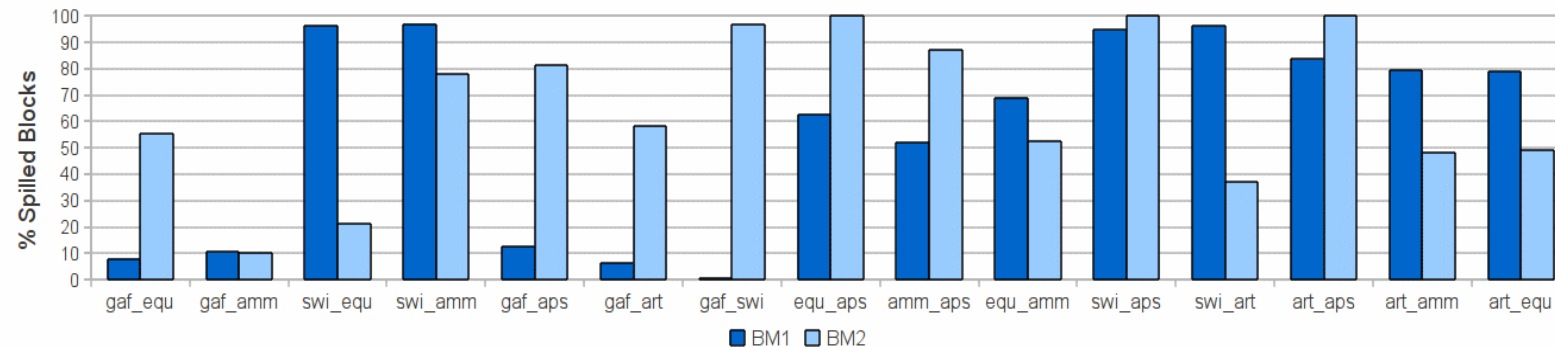
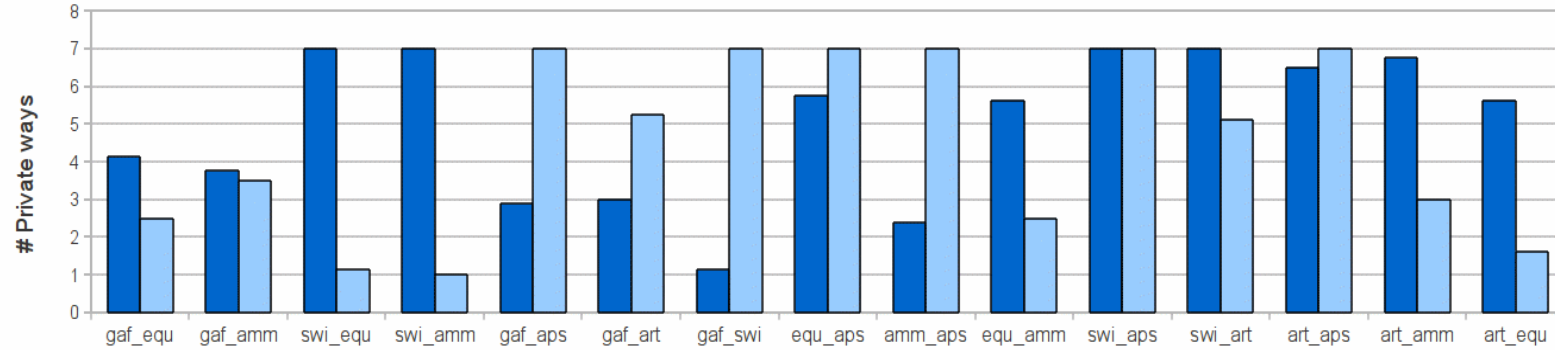


Evaluation – Cache Behavior

ElasticCC



ElasticCC + Adaptive Spilling



■ BM1 ■ BM2

Gafort –
Low Utility

**Apsi, Art,
Equake** –
Saturating
Utility

Amp –
Shared High
Utility

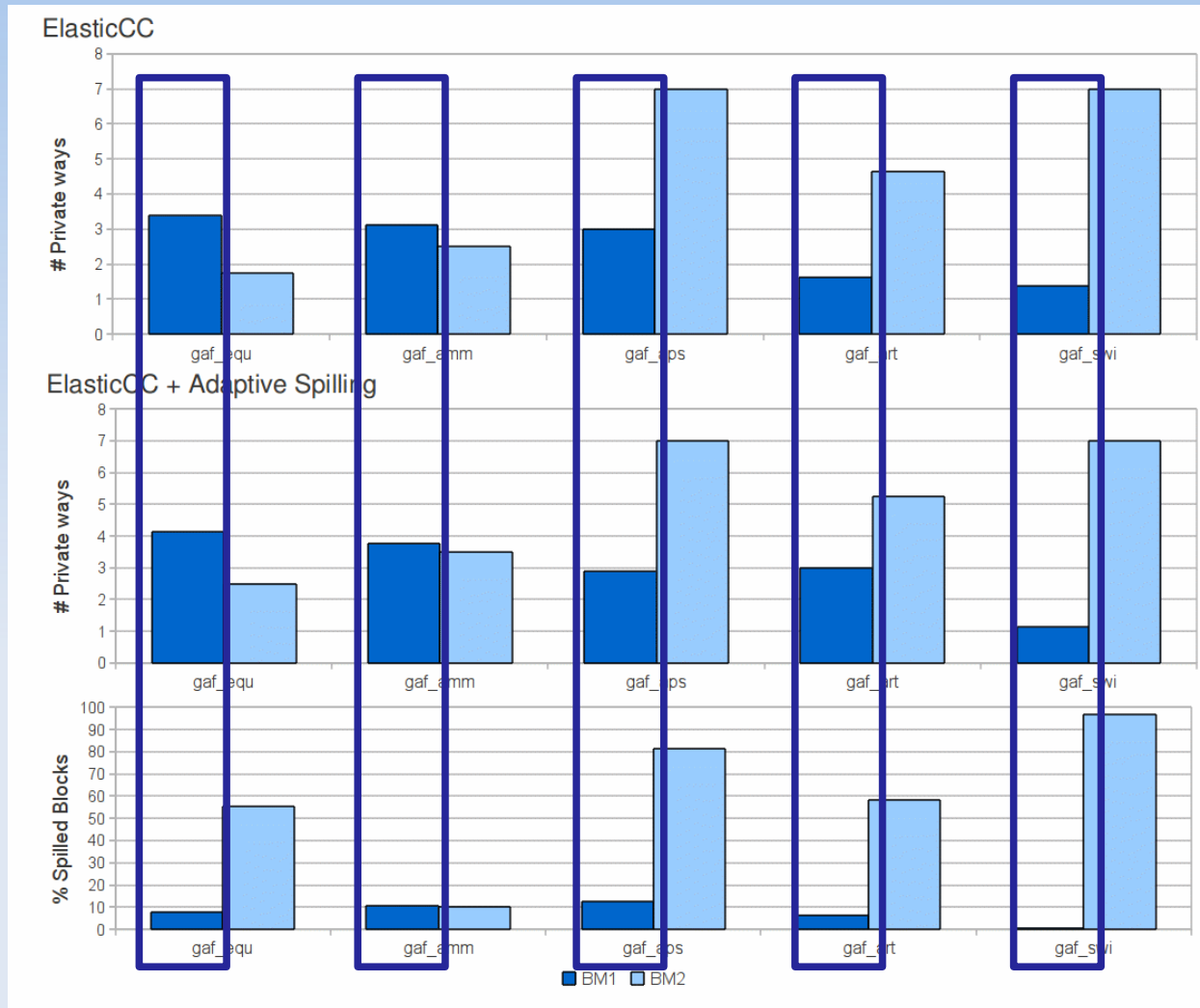
Swim –
Private High
Utility



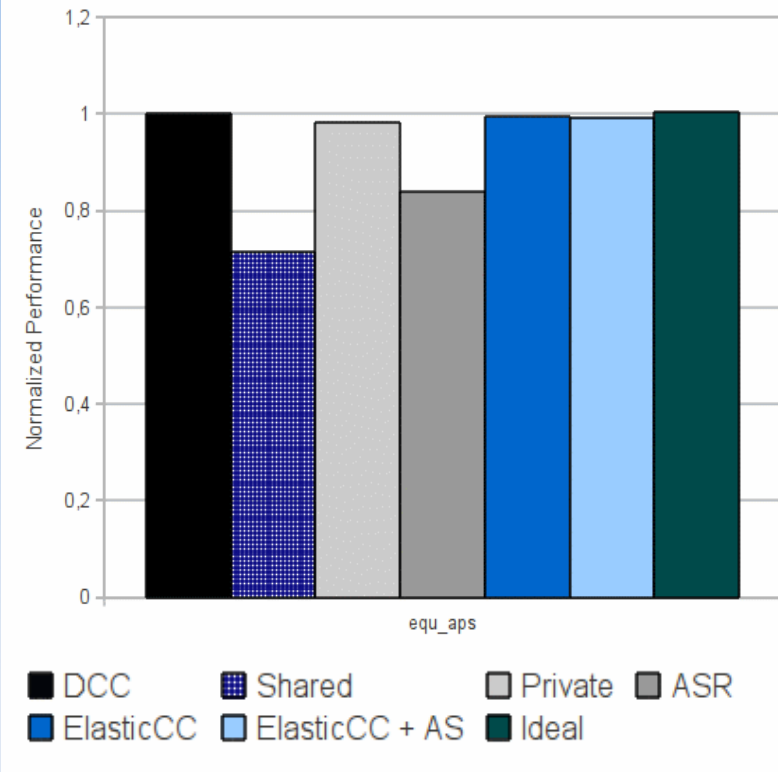
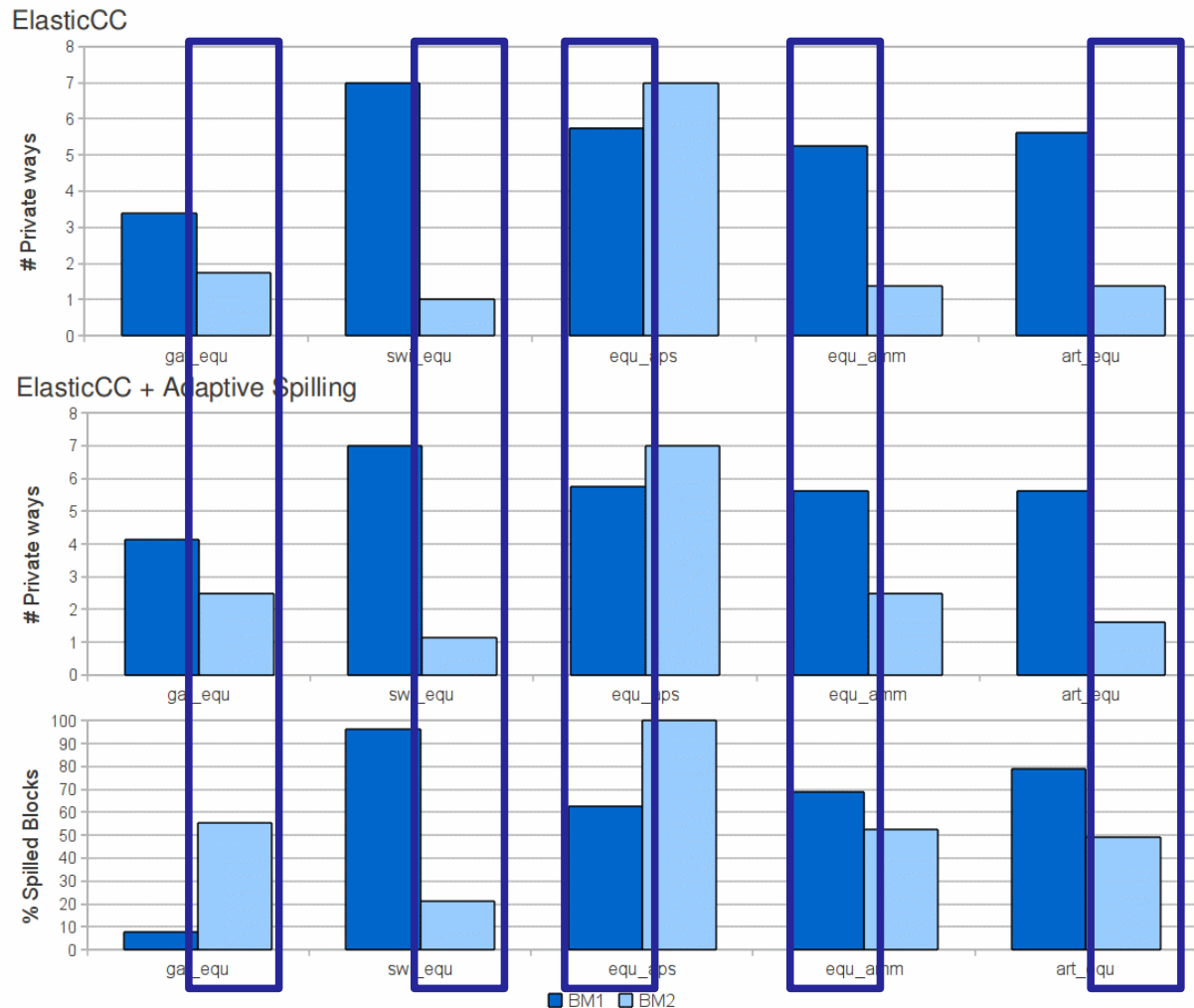
Evaluation – Cache Behavior

Gafort –
Low Utility

No reuse, does not
benefit from caches.



Evaluation – Cache Behavior

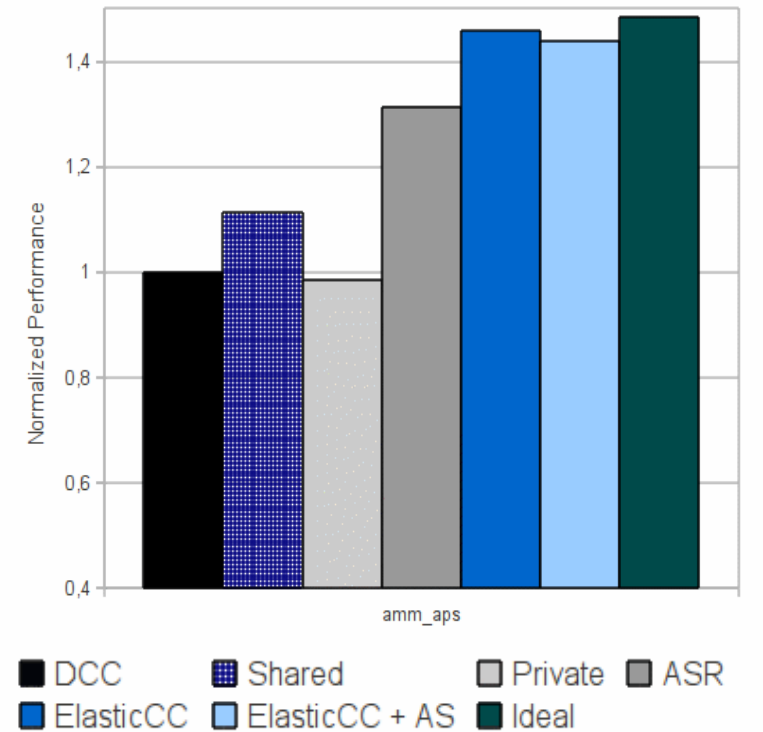
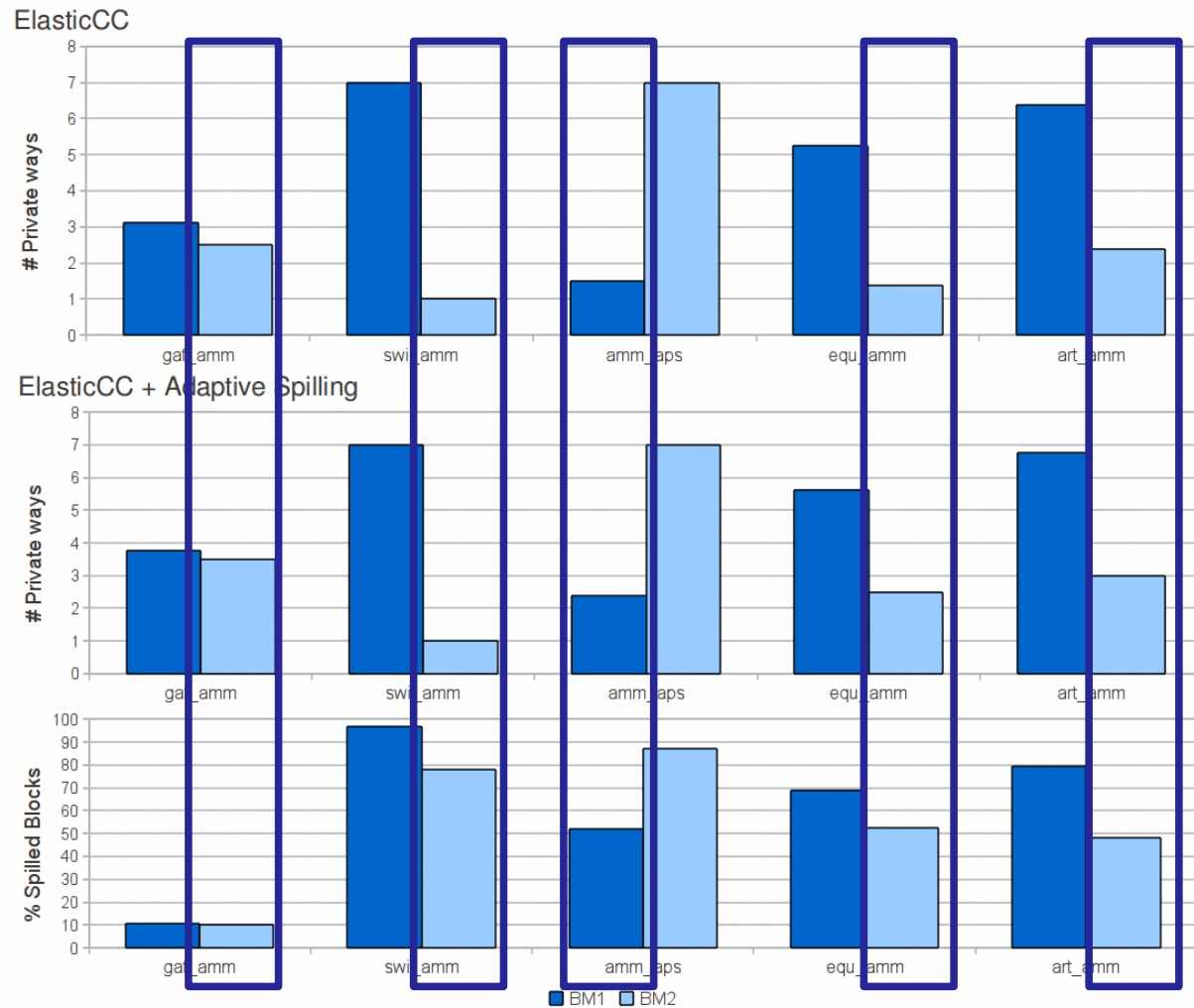


Apsi, Art, Equake –
Saturating Utility

Benefits from a given
amount of extra
cache



Evaluation – Cache Behavior

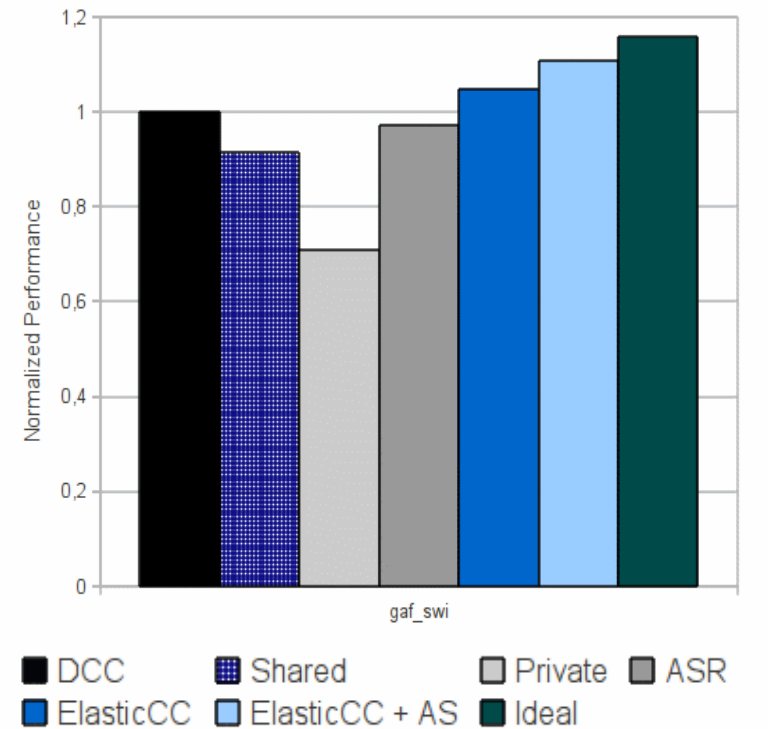
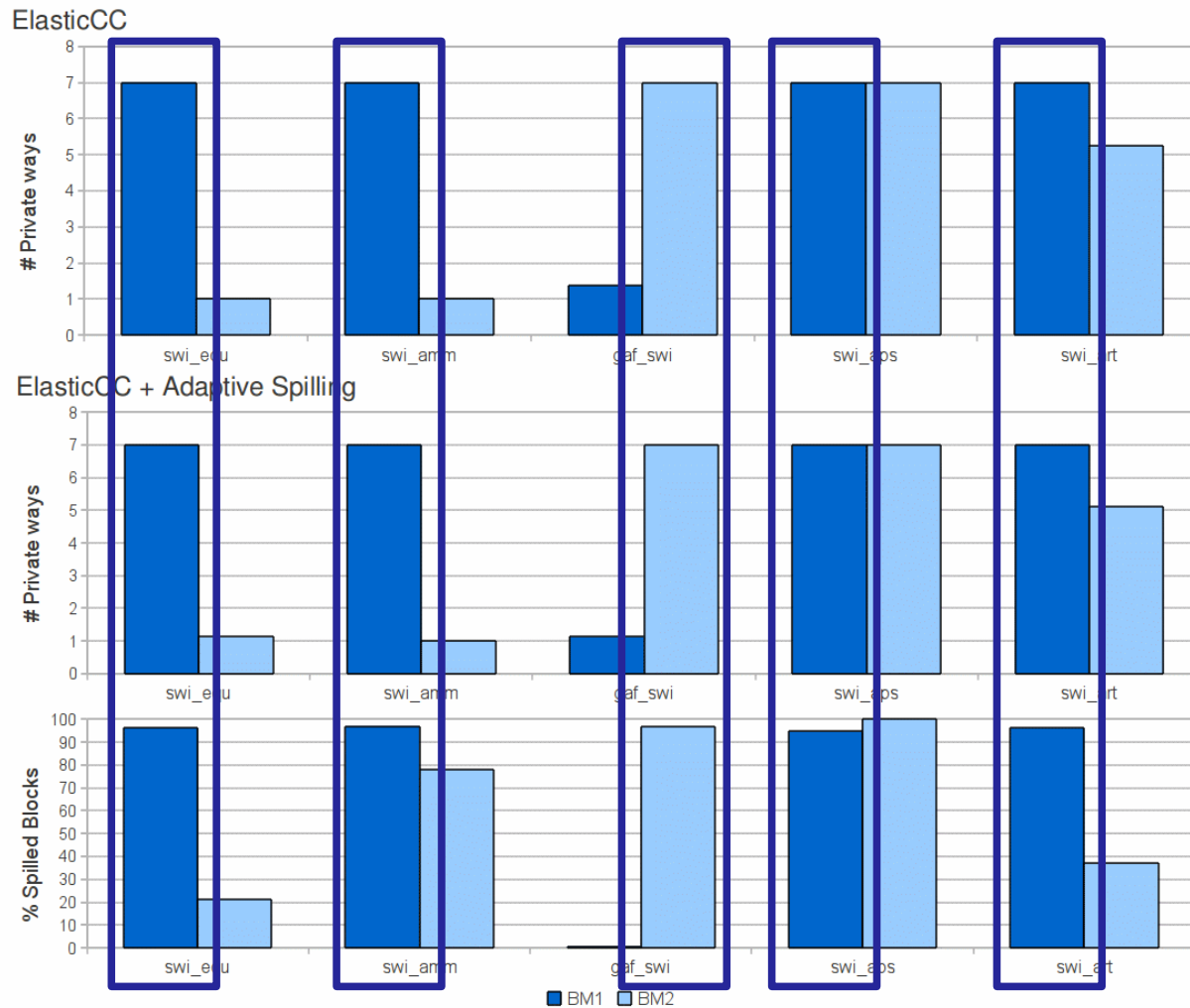


Ammp –
Shared High Utility

Benefits from shared
cache space.



Evaluation – Cache Behavior

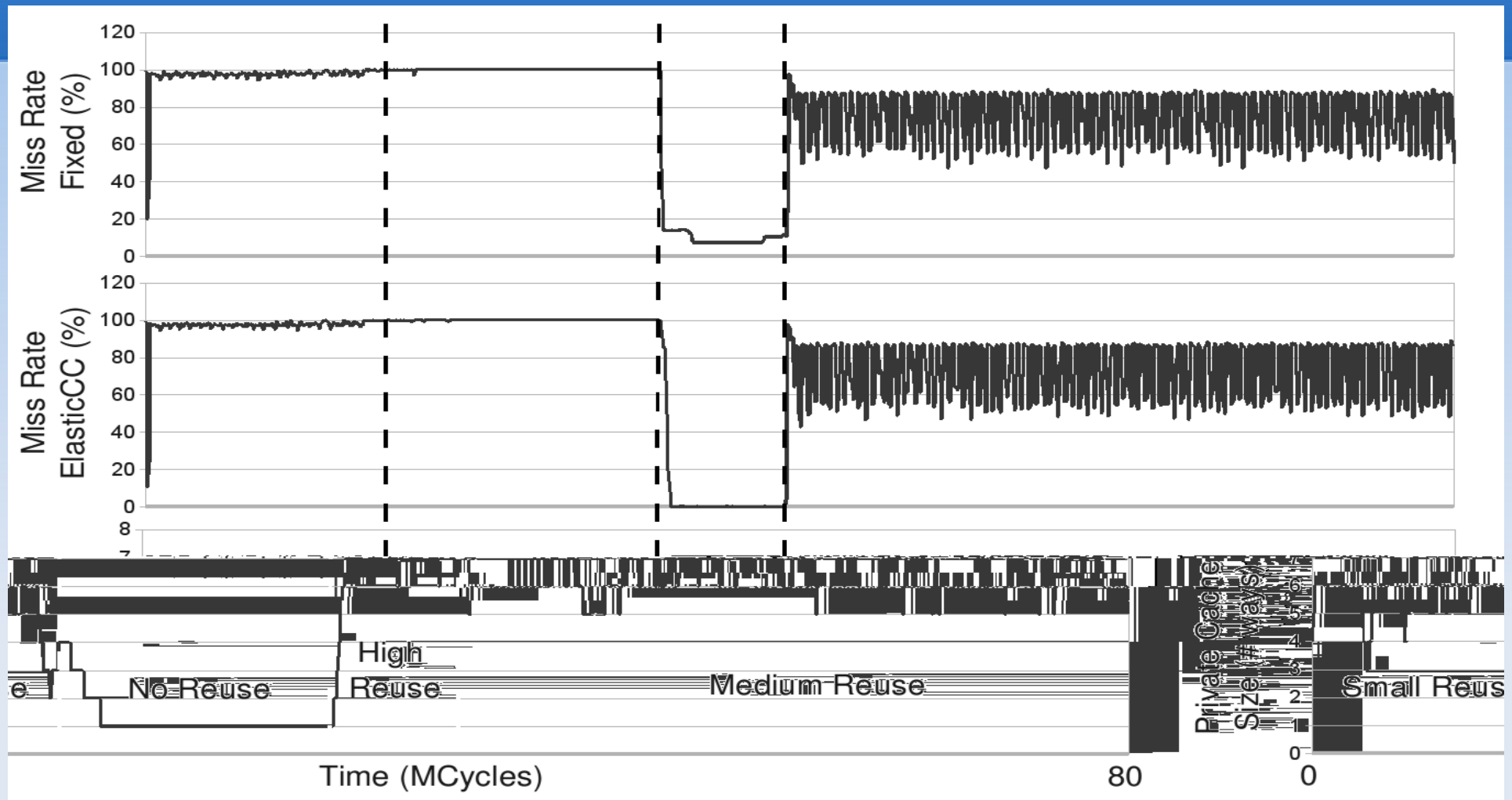


Swim –
Private High Utility

Always benefits from
extra cache



Evaluation - Temporal Cache Behavior

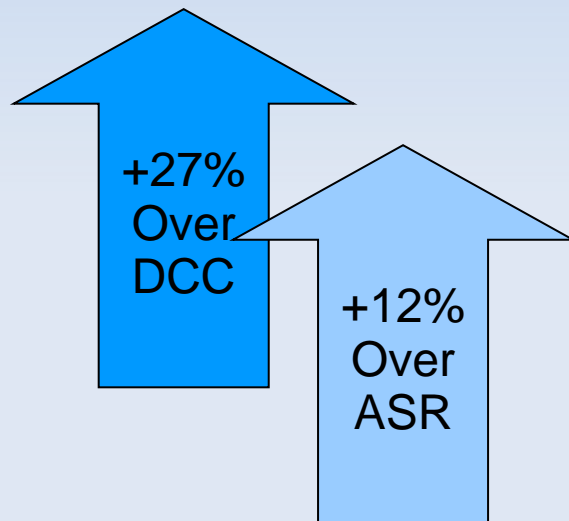


Gafort-Equake execution, Equake Thread 1

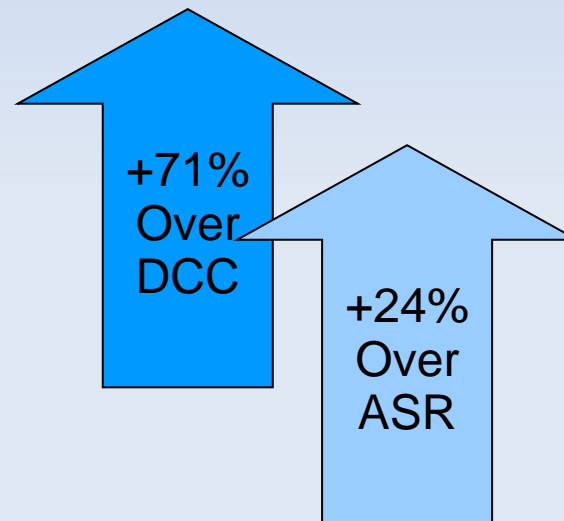
Conclusions

- Elastic Cooperative Caching
 - Distributed organization
 - Adaptive behavior to application requirements

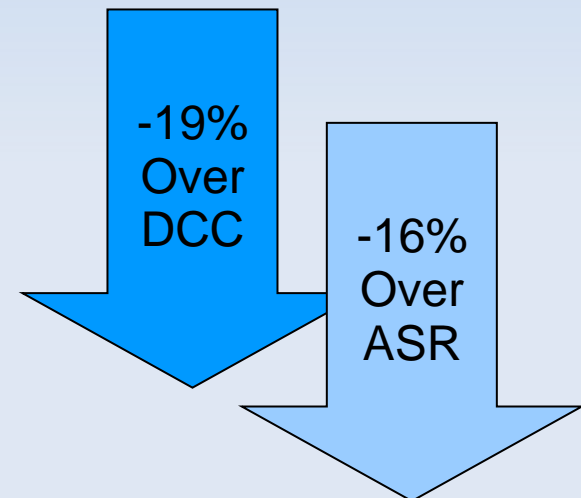
Performance



Energy-Efficiency



Off-Chip Misses



Elastic Cooperative Caching:

An Autonomous Dynamically Adaptive Memory
Hierarchy for Chip Multiprocessors

Enric Herrero¹, José González², Ramon Canal¹

¹Universitat Politècnica de Catalunya

²Intel Barcelona

eherrero@ac.upc.edu



UNIVERSITAT POLITÈCNICA
DE CATALUNYA