DE LA RECHERCHE À L'INDUSTRIE

cea

www.cea.fr

Université de Bretagne-Sud

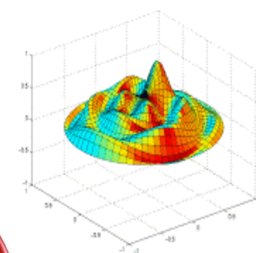
INTRODUCING A DATA SLIDING MECHANISM FOR COOPERATIVE CACHING IN MANY-CORES ARCHITECTURES

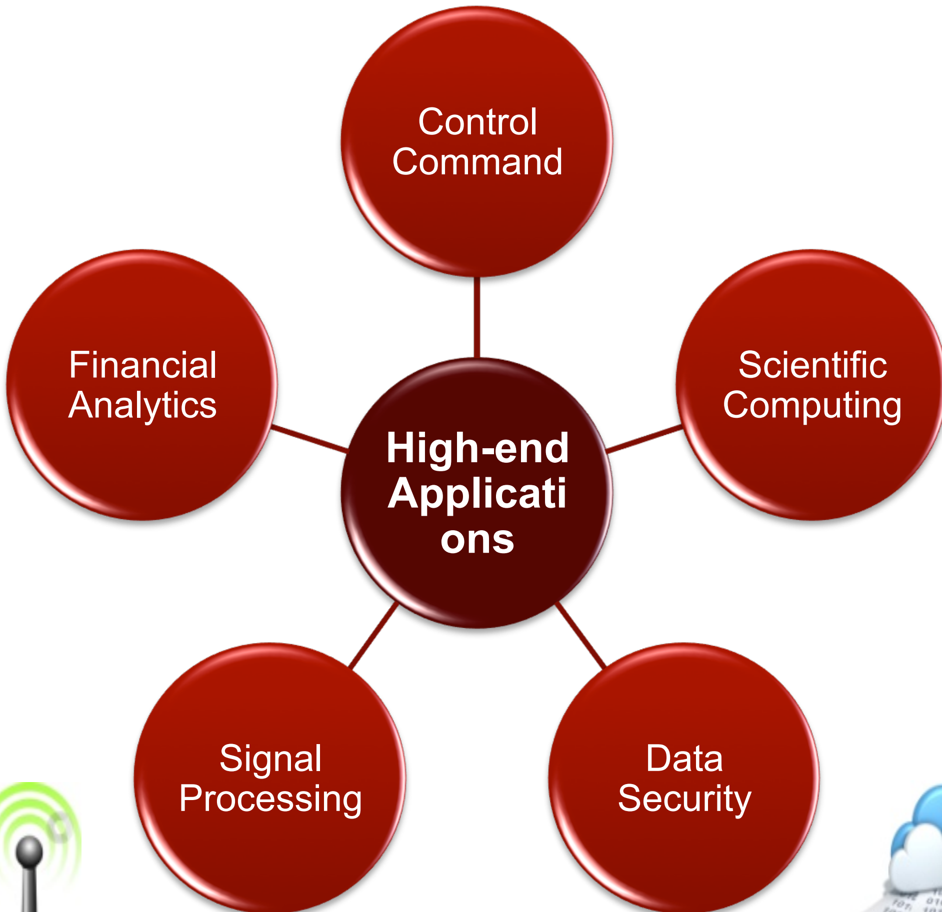
18TH INTERNATIOALWORKSHOP ON HIGH-LEVEL PARALLEL PROGRAMMING MODEL AND SUPPORTIVE ENVIRONMENTS |

Safae Dahmani, Loïc Cudennec, CEA LIST
Guy Gogniat, LabSTICC University of Bretagne Sud

Boston, USA | MAY 20th 2013

TOWARDS MANY-CORES





TOWARDS MANY-CORES

Multi-cores: Multiple complex cores

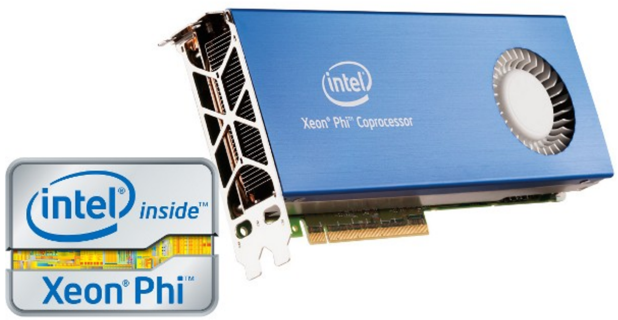
- **Power and heat limits**

Many-cores: hundreds of small cores

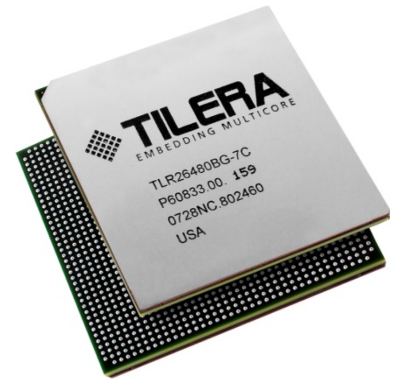
- **High compute throughput with low power consumption**

DE LA RECHERCHE À L'INDUSTRIE

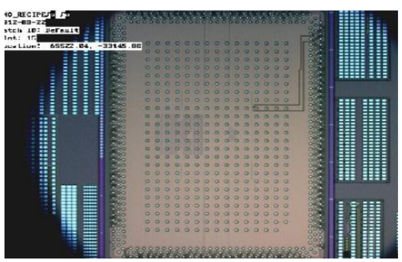
TOWARDS MANY-CORES



***INTEL** Xeon-Phi(**60 cores**)*



TILERA processors:
*TILE-GX (**72 cores**),*
*TILEPro (**64 cores**)*



STMicroelectronics, CEA
*STHORM (**69 cores**)*

***KALRAY** MPPA (**256 cores**)*

MANY-CORES CHALLENGES

- **The main architects issue is:** **Scalability**

- **Memory wall:**
  - There is an inherent cost of accessing distant memory (NUMA)
  - Speed gap between processor and memory

- **Heterogeneous set of workloads:**
  - Massively parallel applications with different memory requirements
  - Complex deployment on large NUMA platform

MEMORY HIERARCHY SINGLE CORE

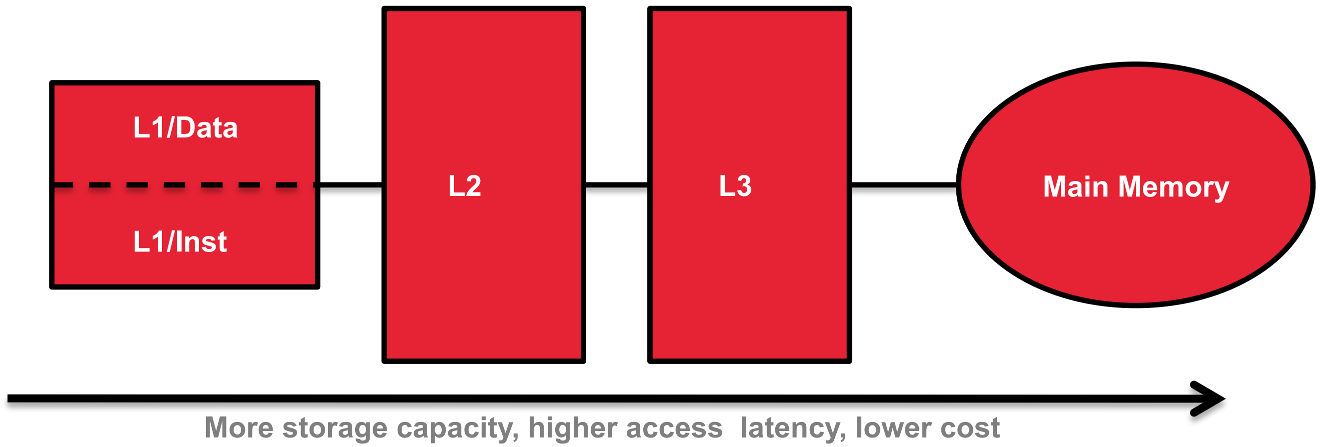


Figure 1: Cache hierarchy for distributed architectures

- **Cache hierarchy organization:**
  - Number of levels
  - Cache size
  - Private/shared
- **Caching strategy:**
  - Replacement policy
  - Data allocation
  - Cache coherency

MEMORY HIERARCHY MANY-CORE

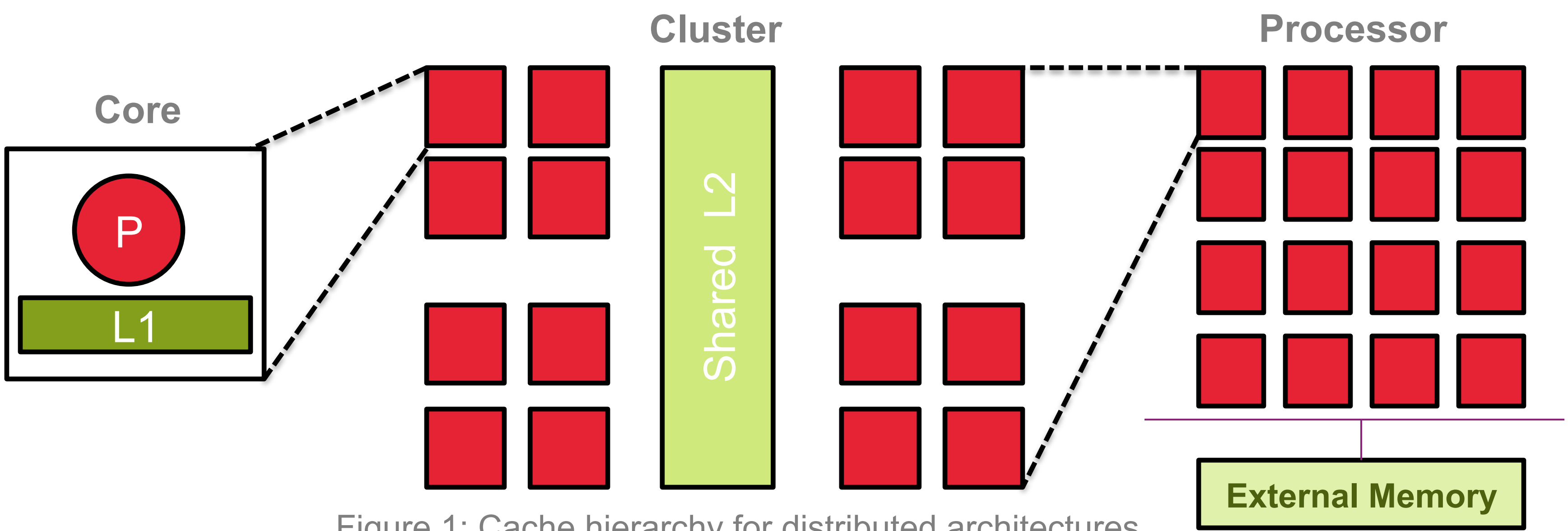


Figure 1: Cache hierarchy for distributed architectures

- **Cache hierarchy organization:**
  - Number of levels
  - Cache size
  - Private/shared
- **Caching strategy:**
  - Replacement policy
  - Data allocation
  - Cache coherency

PRIVATE VS SHARED

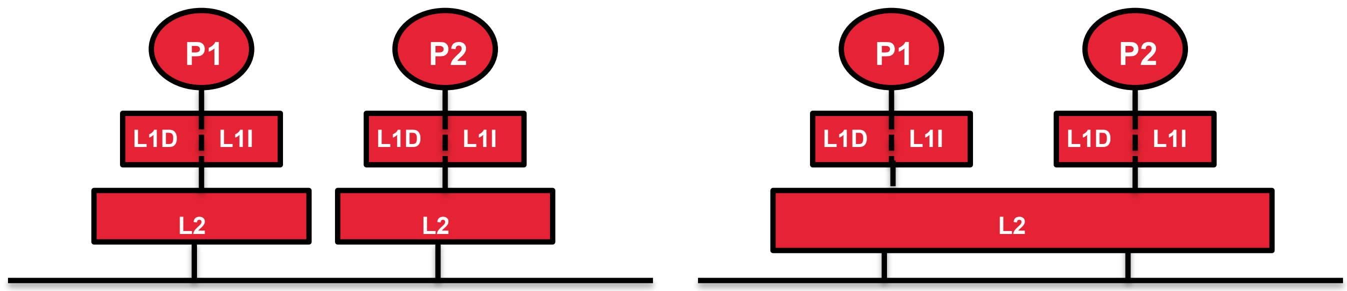


Figure 2: Private/Shared caching

Private	Shared
+ Scale easily - Unneeded data replication	+ Avoid some data transfers + Higher storage capacity - Contention

COOPERATIVE CACHING

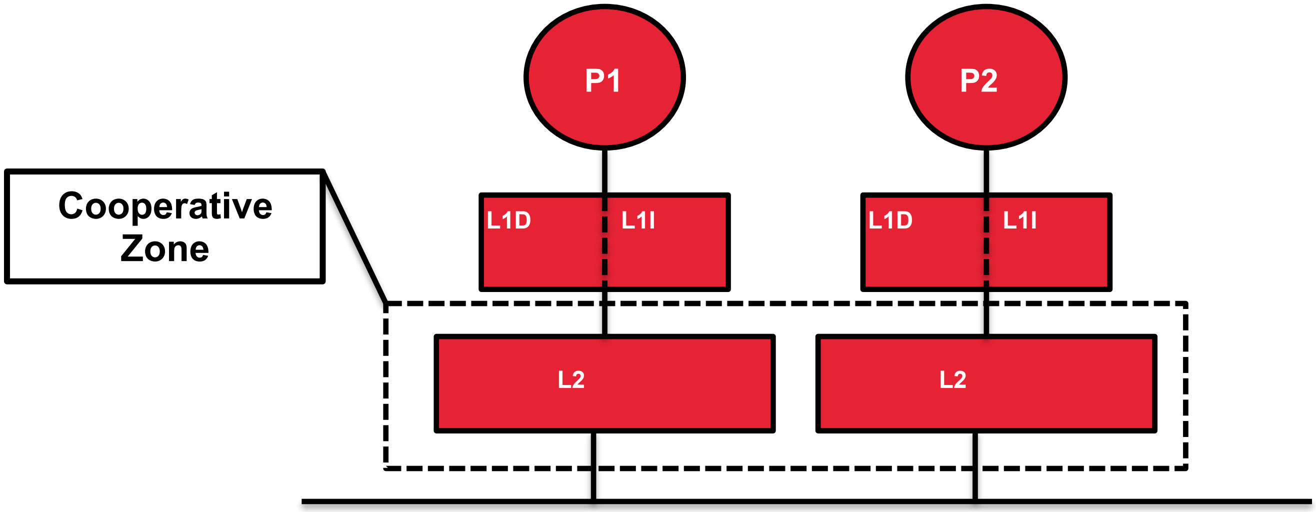


Figure 3: Cooperative caching

Aggregate {
- Low access latency of private caches
- High storage capacity of shared caches

❑**Cooperative caching** is mainly used in power-aware large scale systems: wireless networking (MANET), data storage systems(web servers).

COOPERATIVE CACHING TECHNIQUES

- Hybrid Caching:
  - Static cache **partitioning**

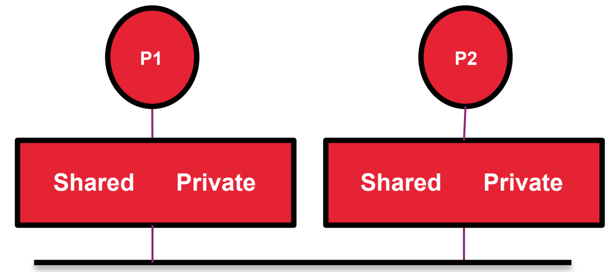


Figure 4: Hybrid Caching

- Adaptive Cache Partitioning
  - **Dynamic** cache partitioning

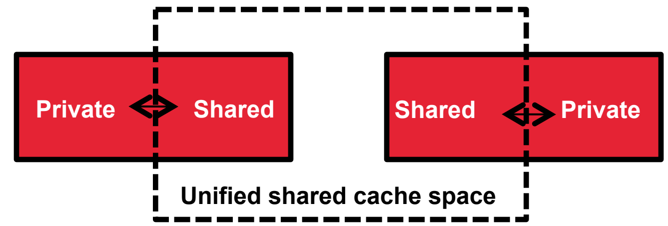


Figure 5: Adaptive Caching

- Elastic Cooperative Caching**
  - **Local** cache partitioning

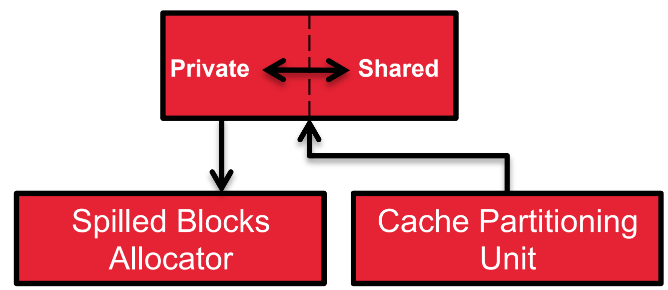
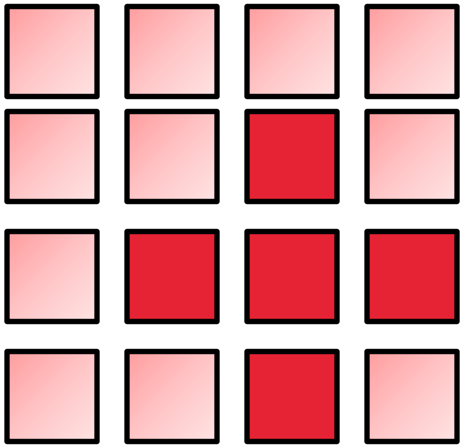


Figure 6: Elastic Cooperative Caching

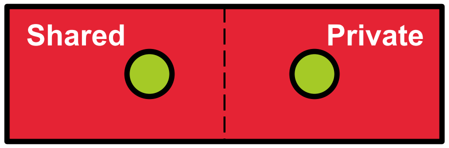
****E. Herrero, J. Gonzalez, and R. Canal, "Elastic cooperative a caching: an autonomous dynamically adaptive memory hierarchy for chip multiprocessors,"**

STRESSED NEIGHBORHOOD CONTEXT

- In the case of **highly stressed** neighborhood:
  - High data reuse amount
  - Concurrent accesses to cooperative zone



- Elastic Cooperative Caching:
  - Cyclic cache partitioning
  - Round Robin spilled blocks allocation



CONTRIBUTION: DATA SLIDING MECHANISM

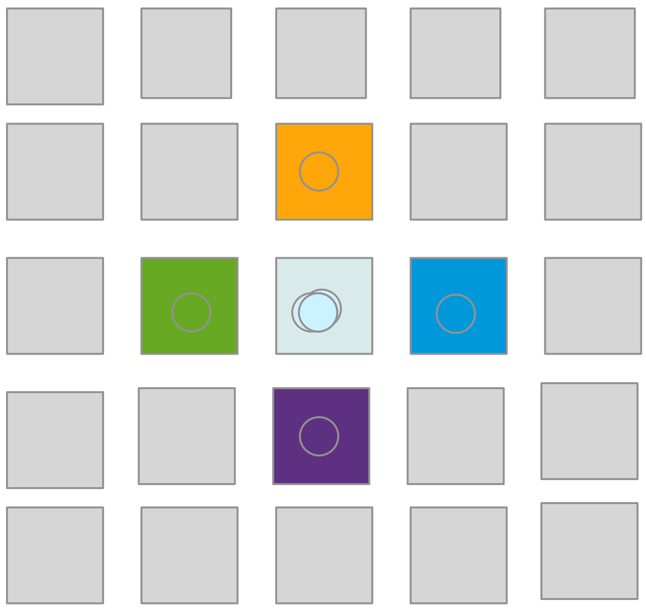
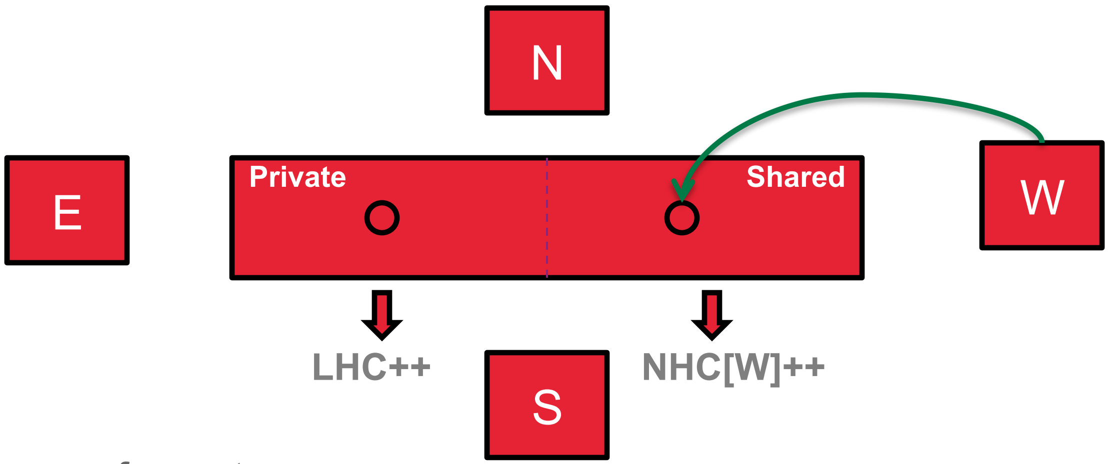


Figure 7: Data Sliding Approach

DATA REPLACEMENT POLICY

Replacement policy: Data replacement policy is based on access frequency to each data set (shared/private).



- Use two types of counters:
  - **LHC**: Local Hit Counter for private data access
  - **NHC[4]**: One Neighbor Hit Counter per neighbor for shared data access
- Priority based replacement:
  - **LHC> sum(NHC):** Shared LRU block
  - **LHC< sum(NHC):** Private LRU block
- Stressed Neighborhood: Distance(LHC, sum(NHC))< Threshold

BEST NEIGHBOR SELECTOR POLICY

Best Neighbor policy: Host neighbor is the least stressed one ⇔ Min(NHC[4]).

**NHC: Neighbor Hit Counter

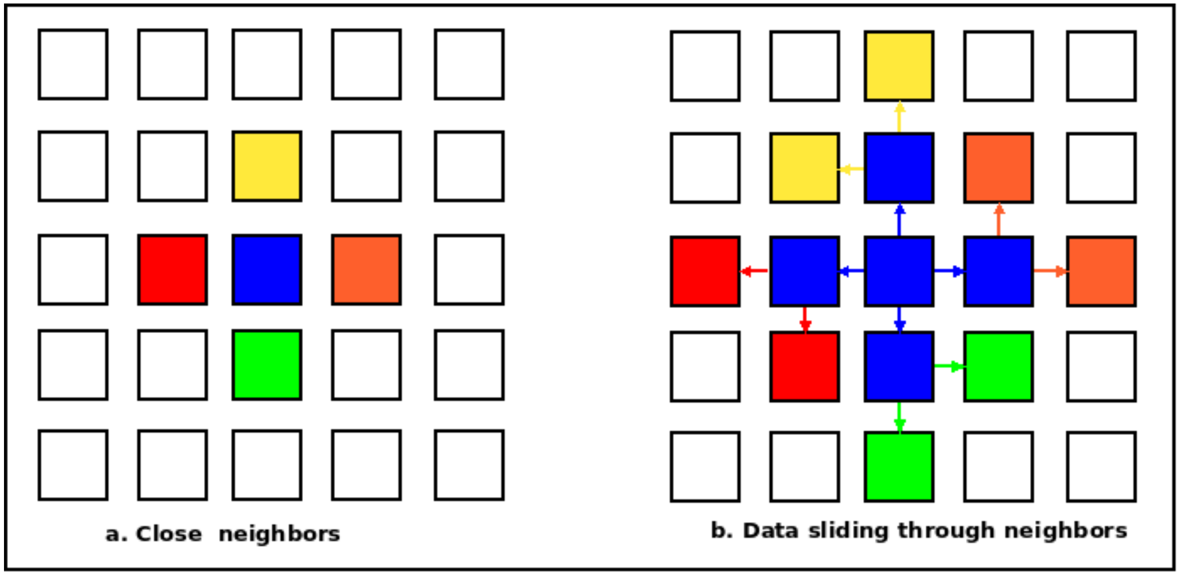


Figure 8: Data sliding through closest neighbors

➔ *Block's destination is selected according to cache stress amount of each neighbor.*

VALIDATION PROCESS: ANALYTICAL SIMULATION

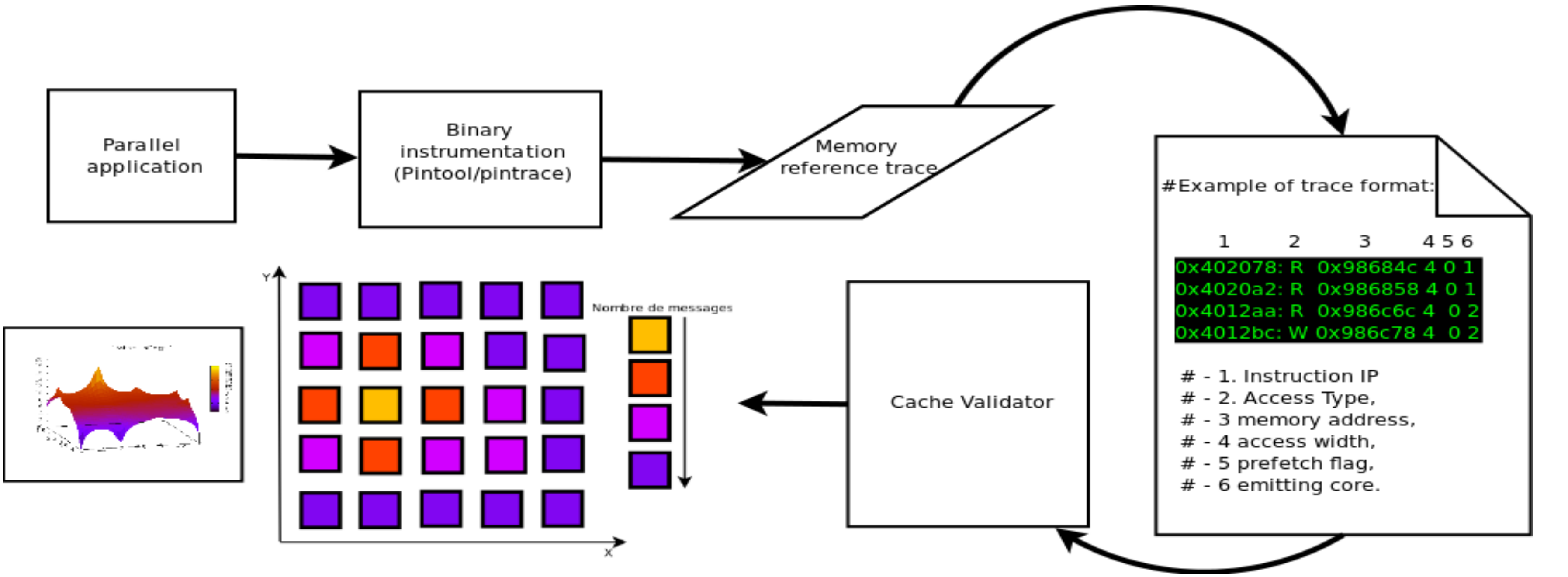


Figure 9: Validation Process Chain

- Simulation tool calculates messages/node induced by application access.
  - No timing is considered

GLOBAL TRAFFIC EVALUATION

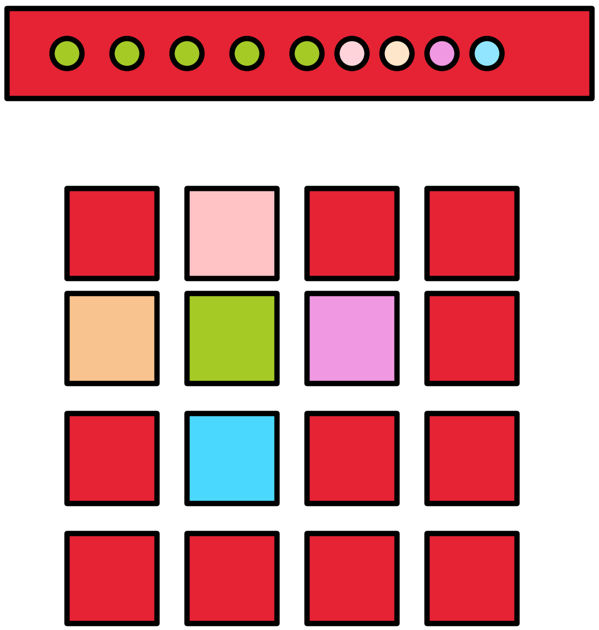
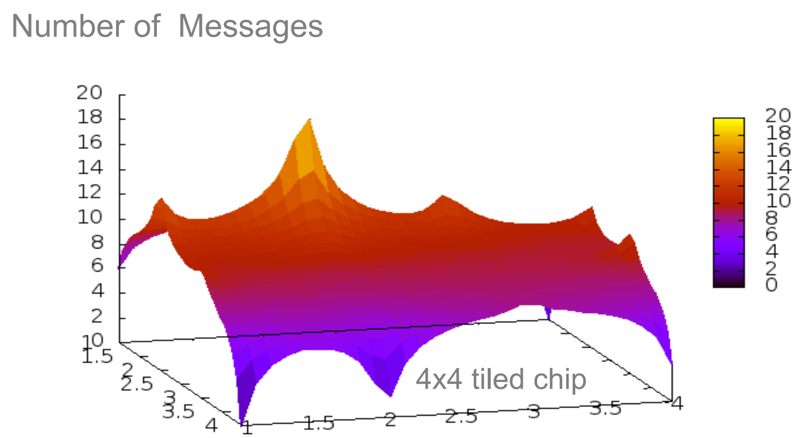
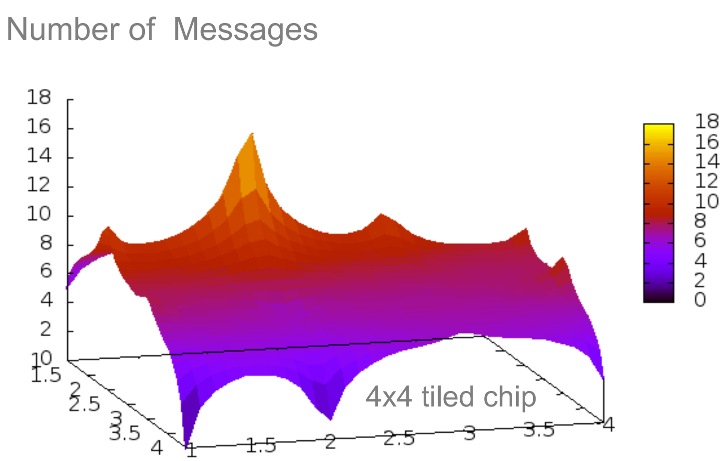


Figure 11: Stressed neighborhood

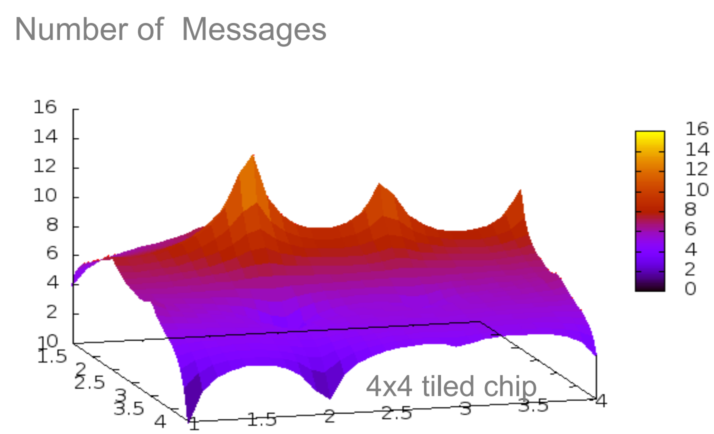
GLOBAL TRAFFIC EVALUATION



Baseline protocol

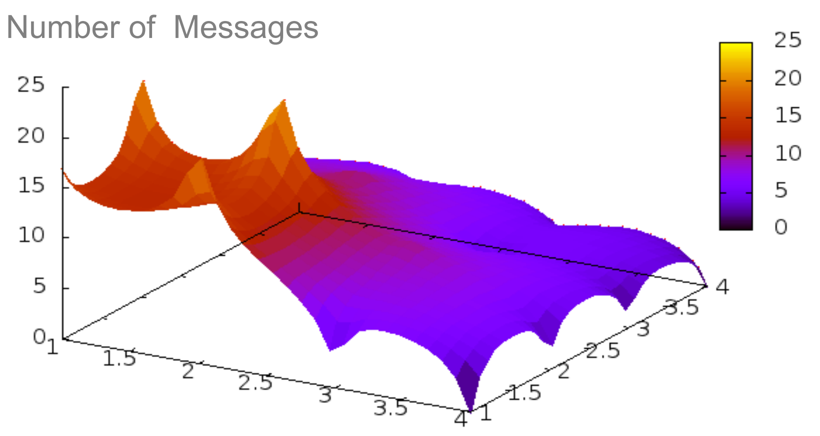


Elastic Cooperative Caching protocol

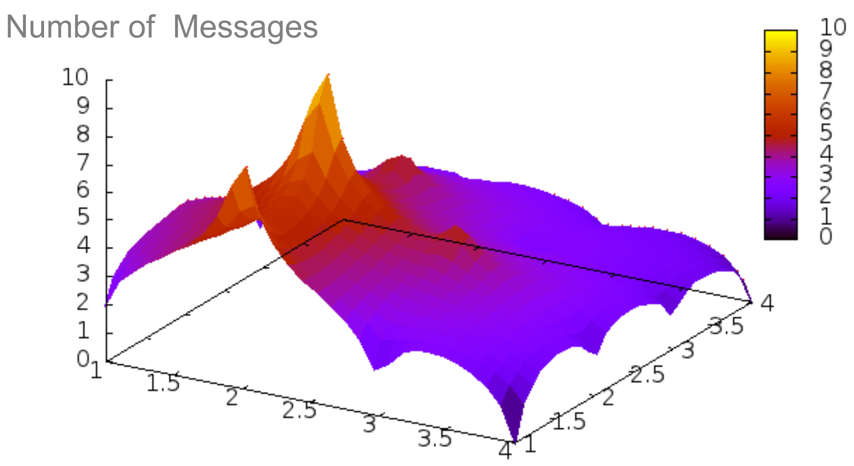


Cooperative Caching with Data Sliding

PRIORITY-BASED REPLACEMENT EVALUATION



Cyclic Replacement



Priority-based Replacement

- **Traffic is half reduced**
  - **Lower off-chip eviction rate**

CONCLUSION

- **Data Sliding Mechanism:** New approach of **Cooperative Caching** that allows data migration between neighbors in **highly stressed** context.

- Local/ Neighbor Counters (Local view built on local and remote access frequency):
  - **Data Replacement policy**
  - **Best Neighbor Selector**

- Resulting Improvements:
  - **Reduces by half on-chip communications**
  - **Improves cache miss rate (less data evictions)**

PERSPECTIVES

- Extension of data sliding from **1-Hop** to **N-Hop**.

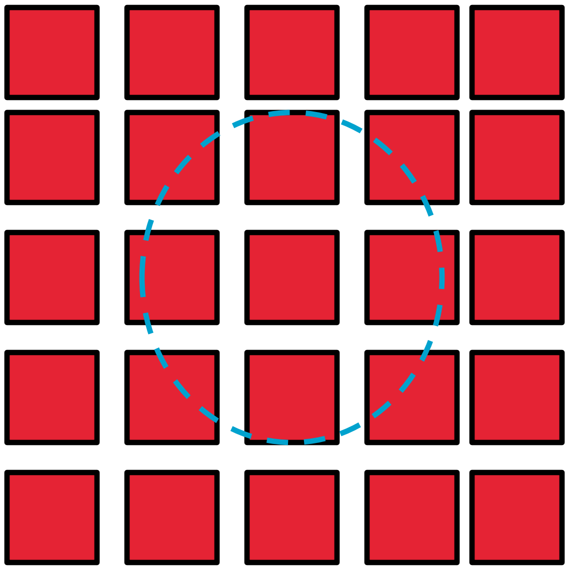


Figure 12: N-Hop Data Sliding

- Allowing access to **hosted blocks.**

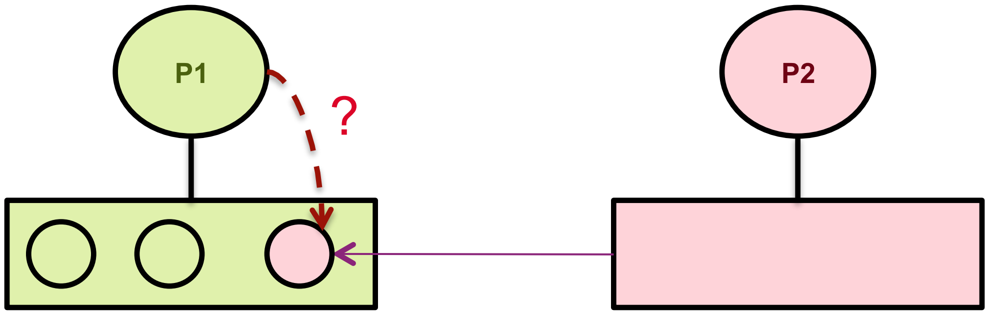


Figure 13: Shared blocks access

DE LA RECHERCHE À L'INDUSTRIE

cea

Université de Bretagne-Sud

INTRODUCING A DATA SLIDING MECHANISM FOR COOPERATIVE CACHING IN MANY-CORES ARCHITECTURES

18TH INTERNATIOALWORKSHOP ON HIGH-LEVEL PARALLEL PROGRAMMING MODEL AND SUPPORTIVE ENVIRONMENTS |

Safae Dahmani, Loïc Cudennec, CEA LIST

Guy Gogniat, LabSTICC University of Bretagne Sud

www.cea.fr

Boston, USA | MAY 20th 2013