Dynamic Spatially Isolated Secure zones for NoC-based Multi and Many-core Accelerators

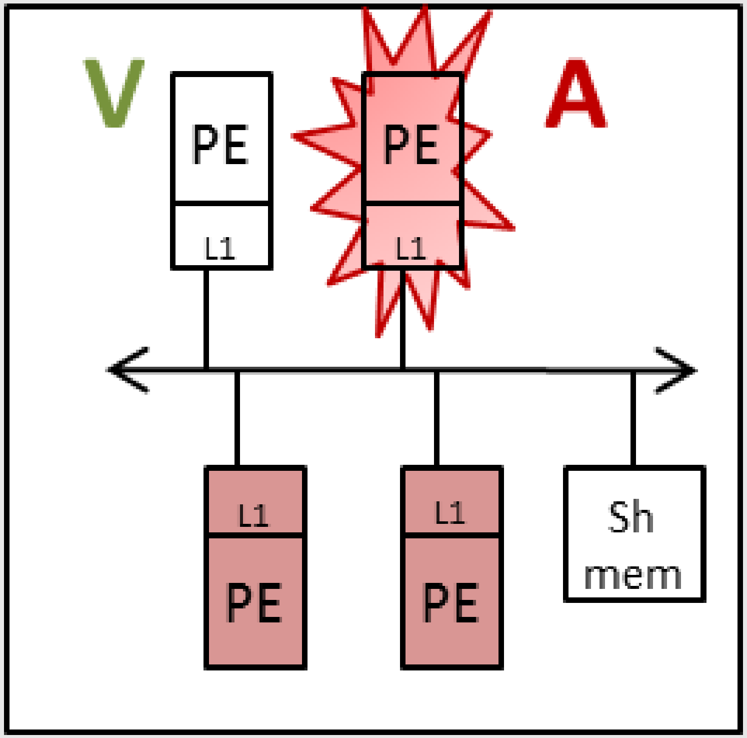
Maria Méndez Real, Vincent Migliore,
Vianney Lapotre, Guy Gogniat
Lab-STICC Université de Bretagne-Sud Lorient FRANCE

Philipp Wehner, Diana Göhringer
Ruhr-University Bochum, GERMANY

Context: Attacking a multicore system

Different SW attacks due to resource sharing:

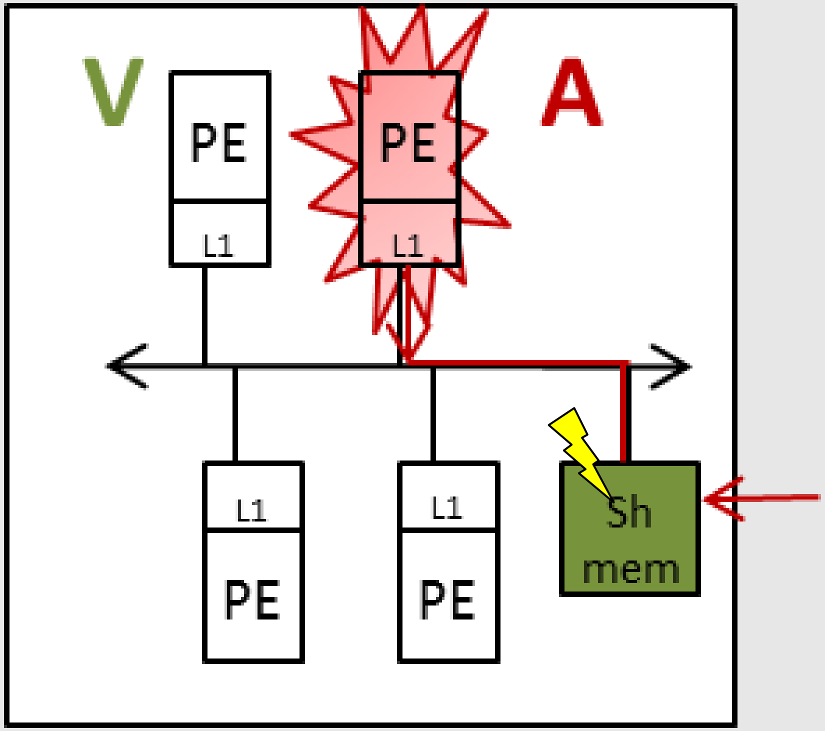
1. Denial of services: preventing other applications from using the shared resources (computing, memory and communication infrastructure resources)



V: Victim
A: Attacker

Context: Attacking a multicore system

2. Confidentiality and integrity attacks: illegal direct access to data



Context: Attacking a multicore system

3. Confidentiality : illegal indirect access to data

Cache based Side-Channel Attacks

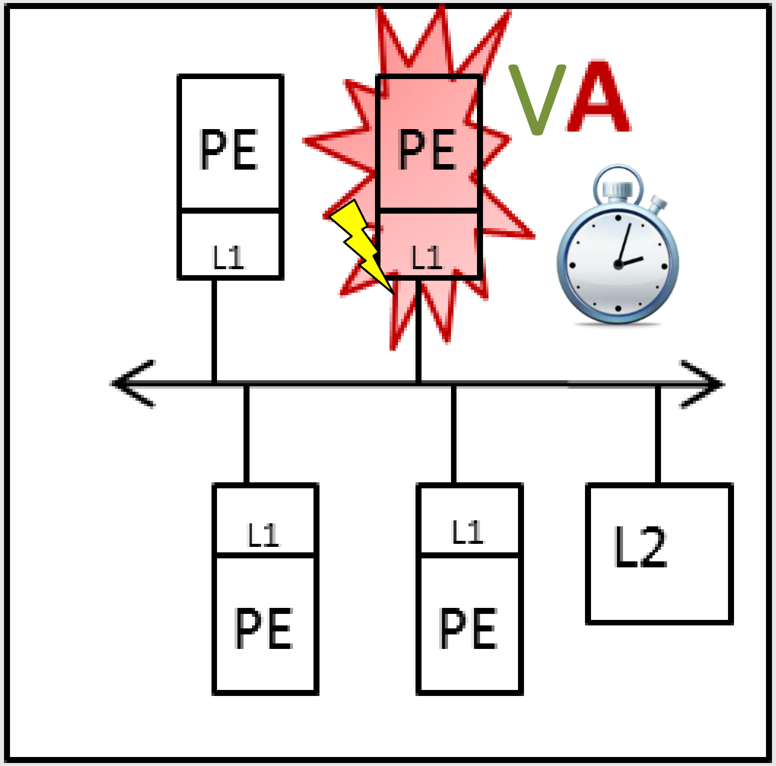
Principle:

Analyzing the leakage of memory access and communications in order to deduce sensitive information about the victim

Context: Attacking a multicore system

1. Access-driven Side-Channel Attacks due to cache sharing

   - Sharing the same core

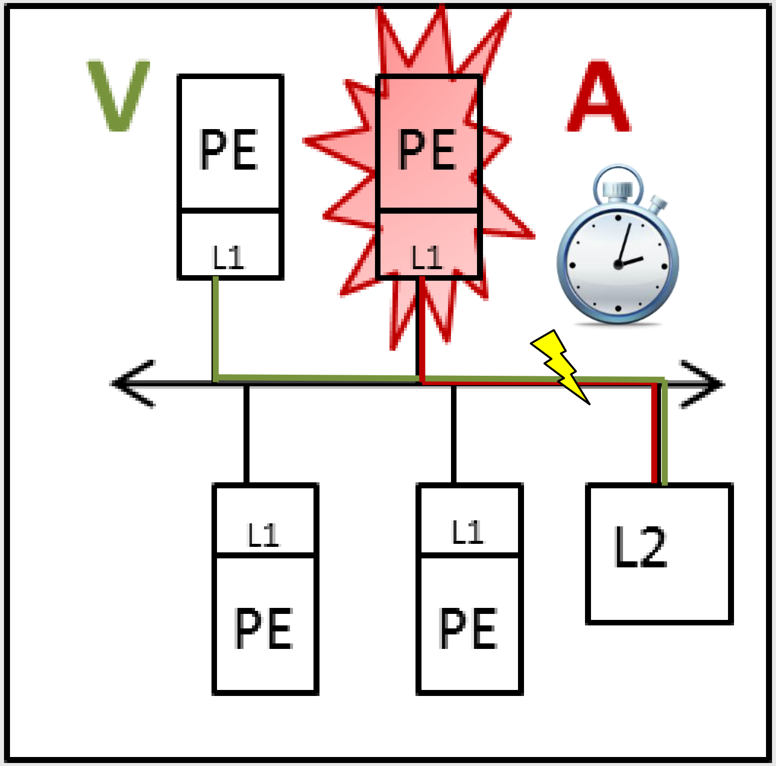


[1] "D. Gullasch, et al., "Cache Games – Bringing Access-Based Cache Attacks on AES to Practice".

Context: Attacking a multicore system

Access-driven Side-Channel Attacks due to cache sharing

- Sharing the same core
- Across cores



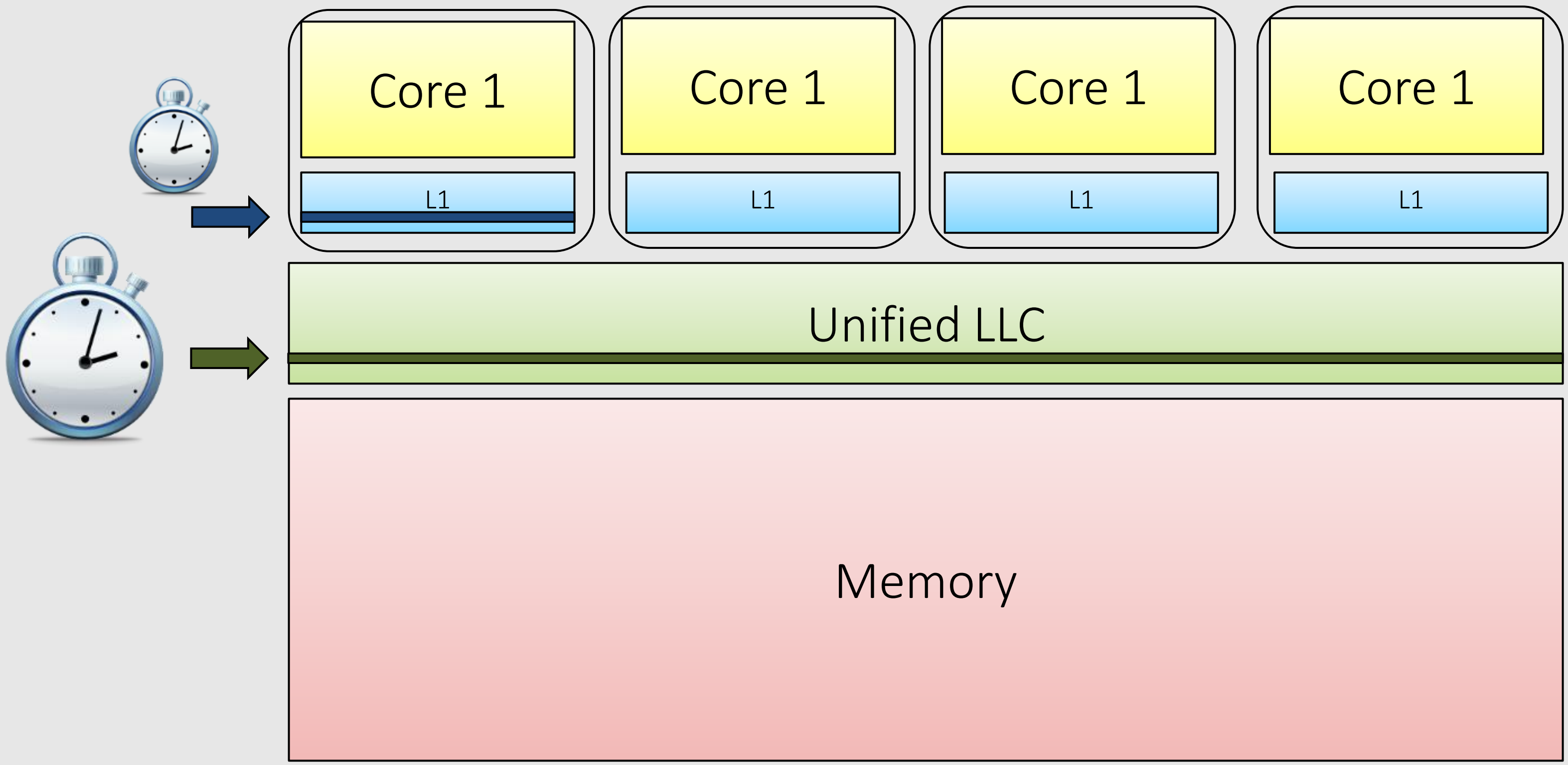
Principle:

Analyzing its own performance -> information about the memory accesses of the victim or the communication flow -> deducing sensitive information

Context: Attacking a multicore system

Background: Cache properties

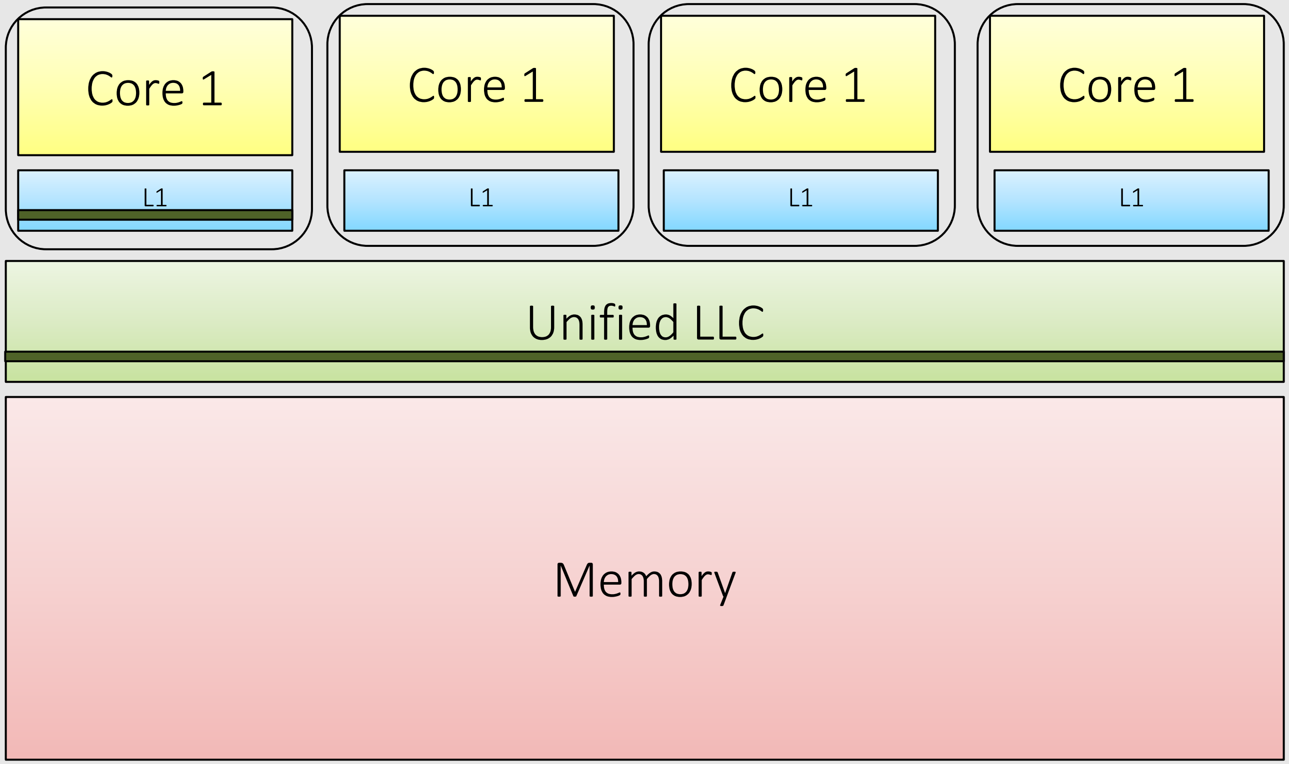
- Retrieving data from cache level closer to memory takes longer than cache levels closer to the core



Context: Attacking a multicore system

Background: Cache properties

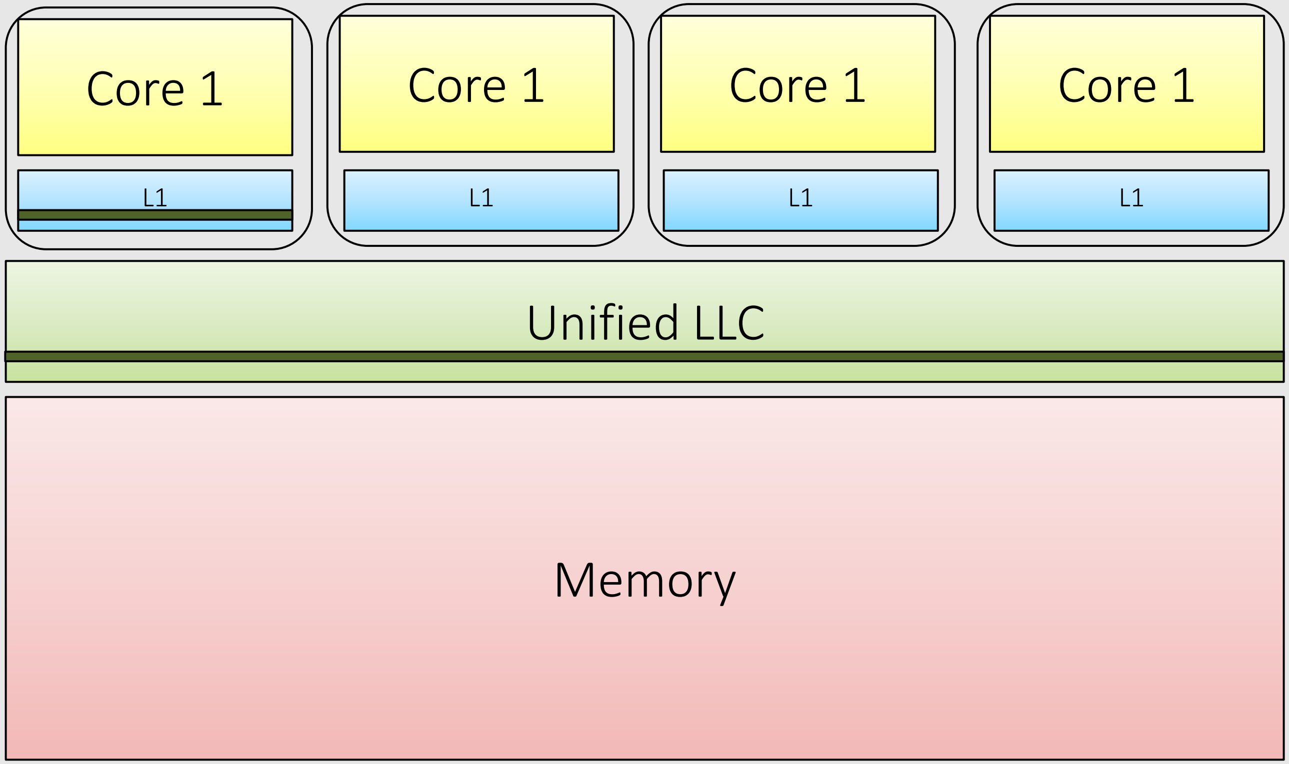
- Retrieving data from cache level closer to memory takes longer than cache levels closer to the core
- LLC caches are inclusive



Context: Attacking a multicore system

Background: Cache properties

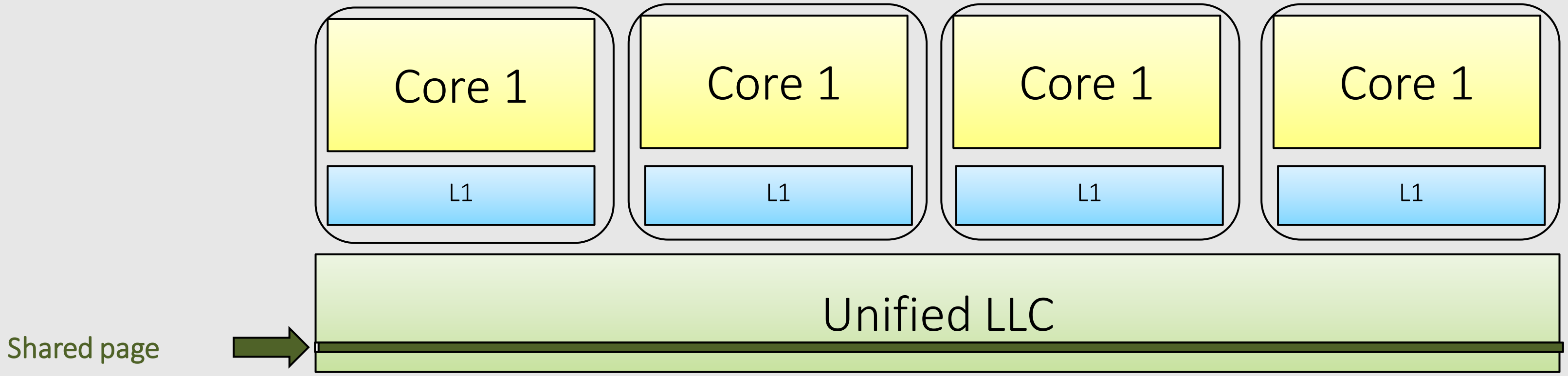
- Retrieving data from cache level closer to memory takes longer than cache levels closer to the core
- LLC caches are inclusive
- Evicting data from LLC evicts also the data on the lower caches



Context: Attacking a multicore system

Background: Access-driven Side Channel Attacks across cores

- Flushing + Reload [1] example



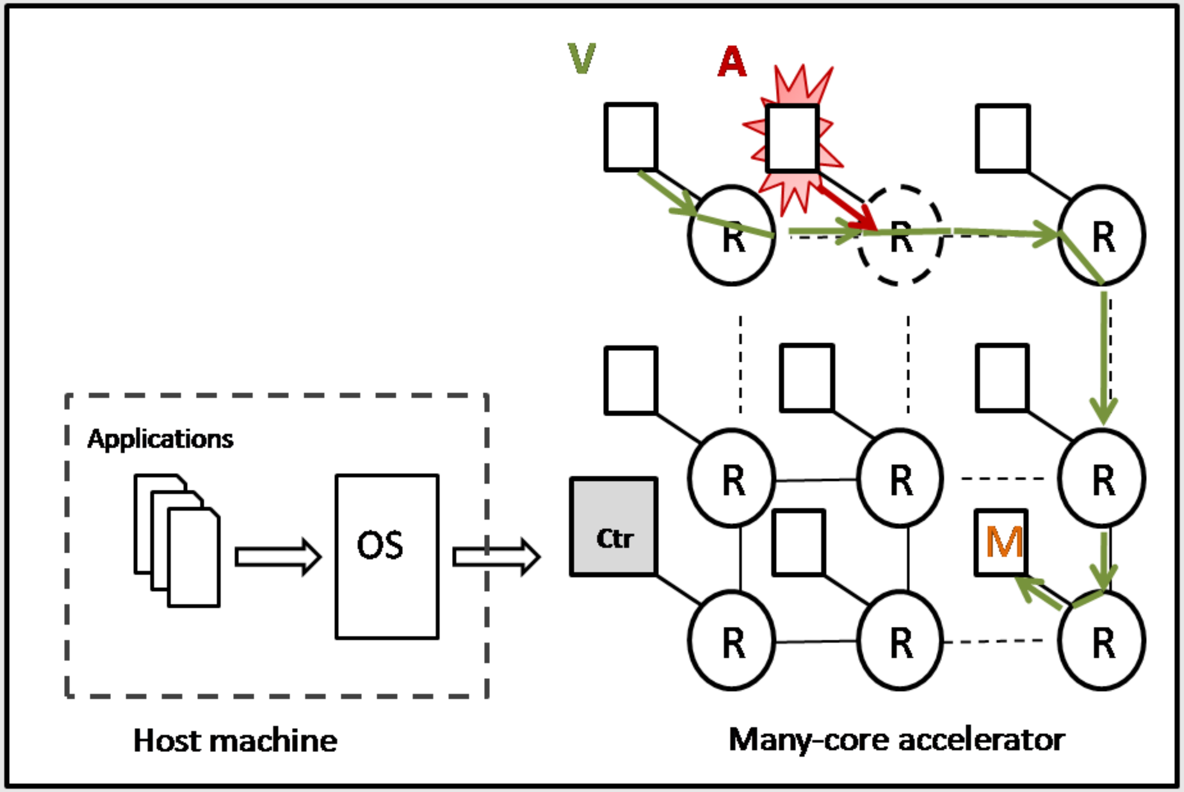
A round of attack:

1. The attacker flushes the monitored memory page
2. Wait until the victim potentially access the line
3. The attacker reload the memory line measuring the time to load it

[2] "Y. Yarom, et al., "FLUSH+RELOAD: A High Resolution, Low Noise, L3 Cache Side-Channel Attack", in the 23th USENIX Security Simposium, 2014.

Context: Attacking a NoC-based system

Time-driven Side-Channel attacks on NoC-based multi and many-core architectures



Principle: Analyzing its own performance to deduce the communication flow

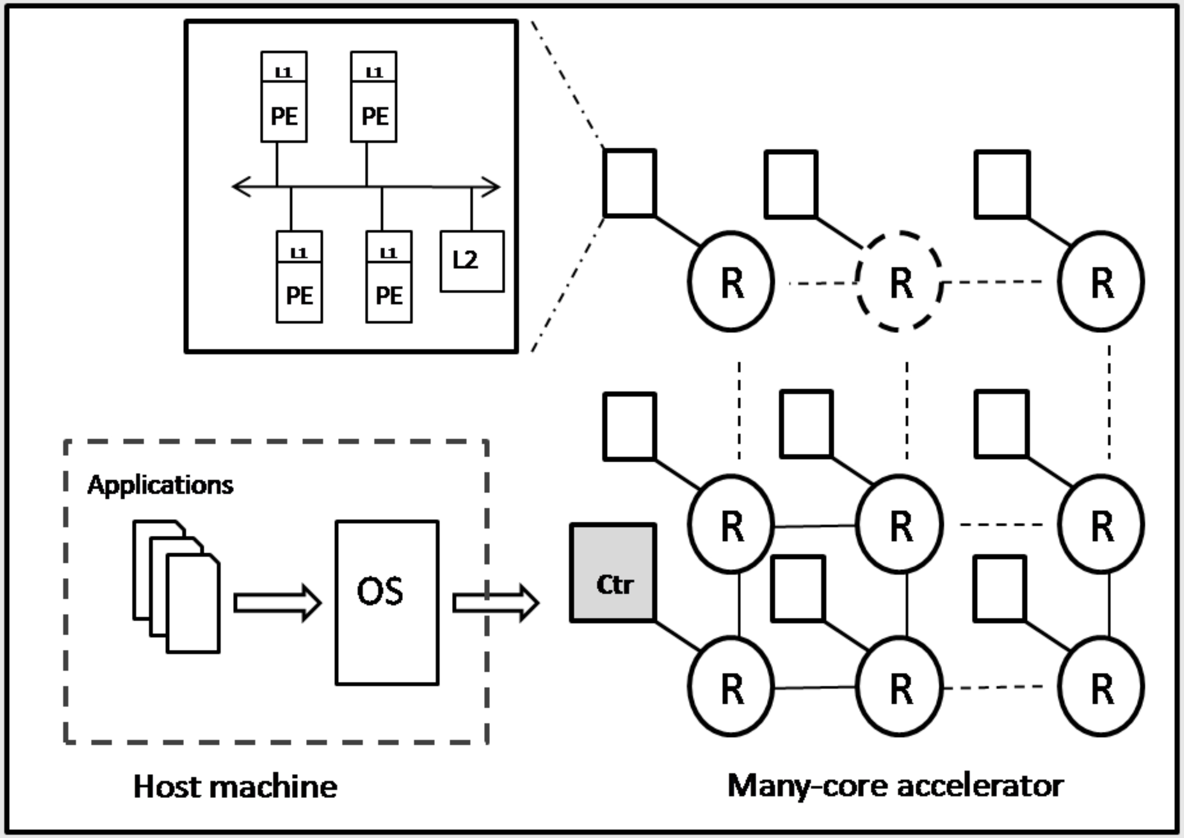
[3] J. Sepulveda, et al., “Noc-based protection for soc time-driven attacks”, Embedded Systems Letters, IEEE, vol. 7, no. 1, pp. 7–10, March 2015.

Context: Attacking a NoC-based system

Sharing resources

Security vulnerabilities

Context: NoC-based many-core accelerator



A host machine delegates part of the computation to a many-core accelerator

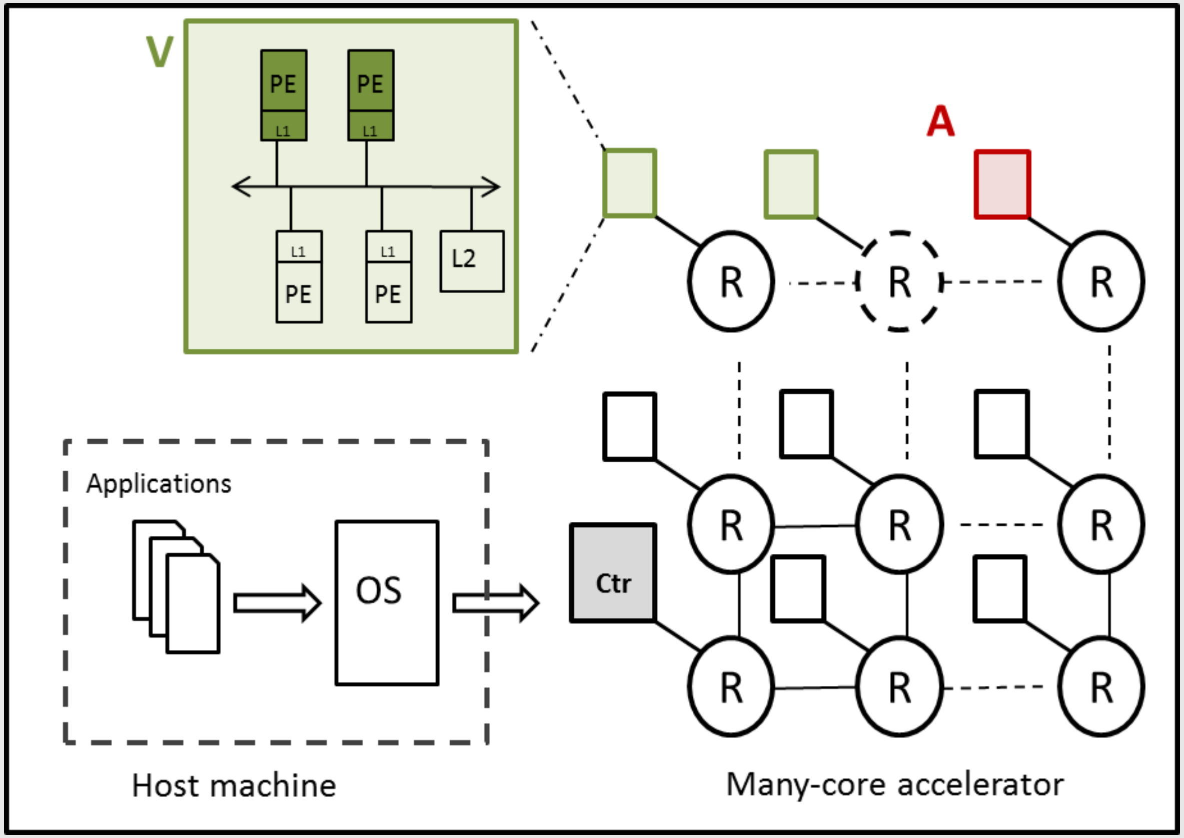
A controller is in charge of the deployment of the applications on the accelerator

Background

Countermeasure	Direct illegal memory access	Access-driven attacks across cores	Time-Driven attacks on the NoC	DoS
Bi partitioning the processor [4]	✔	✘	✘	✘
Logical isolation (MMU, MPU, NoC MMU [5][6])	✔	✘	✘	✘
Monitoring mechanisms [7]	✘	✘	✘	✔
NoC protection [3]	✘	✘	✔	✘

[3] J. Sepulveda, et al., “Noc-based protection for soc time-driven attacks”, Embedded Systems Letters, IEEE, vol. 7, no. 1, pp. 7–10, March 2015.
[4] www.arm.com/products/processors/technologies/trustzone/
[5] R. Masti, et al., “Isolated execution in many-core architectures”, in Proc. of Network and Distributed System Security Simposium (NDSS), 2014.
[6] G. Kornaros, et al., “Hardware Support for Cost-Effective System-level Protection in Multi-Core SoCs”, in Proc. of Digital System esign (DSD), 2015.
[7] L. Fiorin, et al., “A security monitoring service for nocs”, in Proc. of Hardware/Software codesign and system synthesis (CODES+ISSS), 2008.

Spatial isolation for sensitive applications



- How can this be achieved?
- Expected under utilization of resources, how can the performance overhead be evaluated?

Implementation

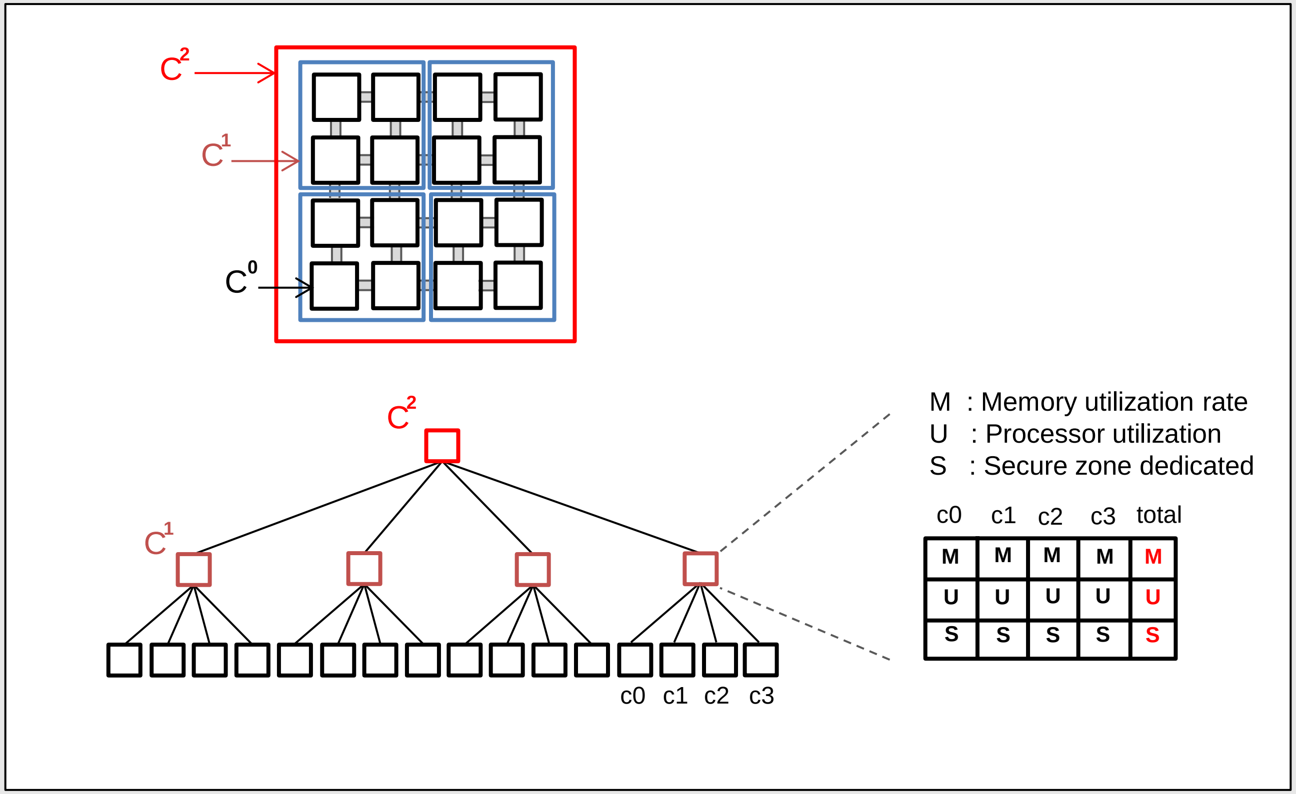
The ARM as a controller of the platform

Evaluation of different deployment strategies running on the ARM

- Monitoring of the platform state
- Resource allocation algorithms
- Secure zones creation strategies

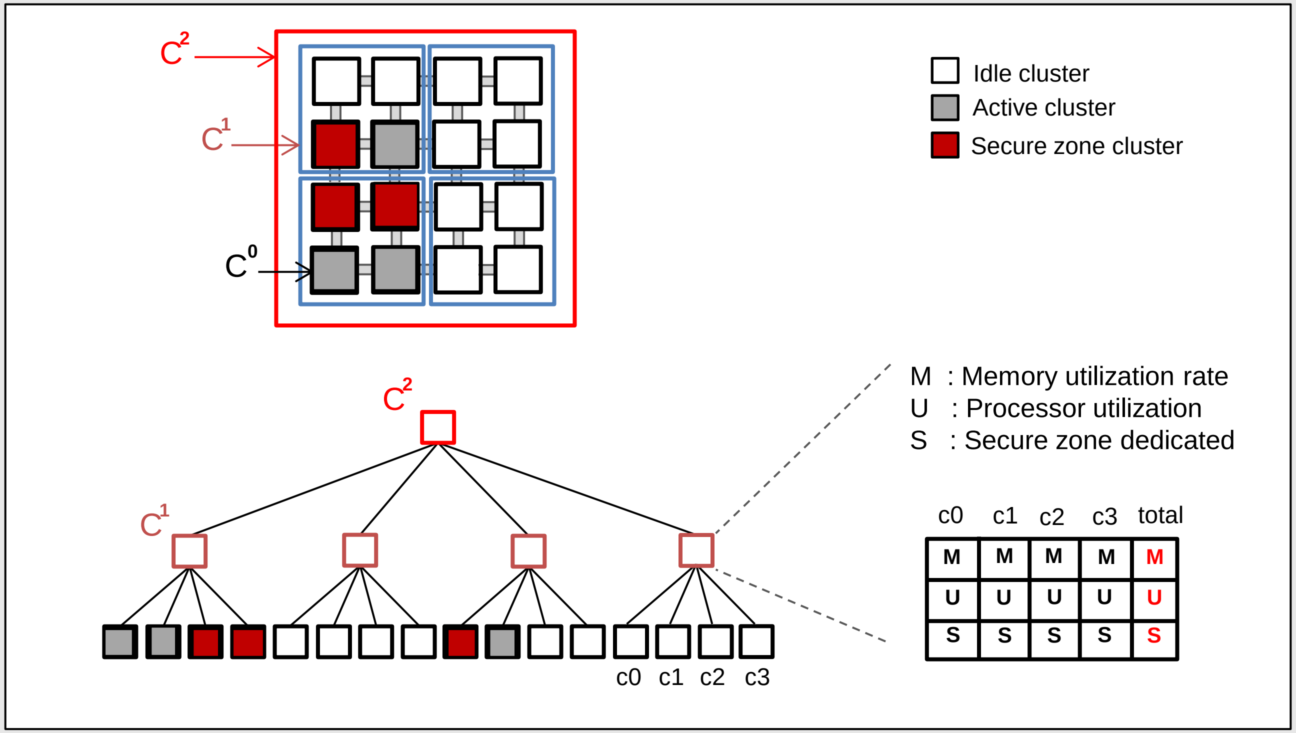
Secure-enable mechanisms

- **Scheduling:** Round Robin
- **Monitoring:** Distributed quaternary decision tree



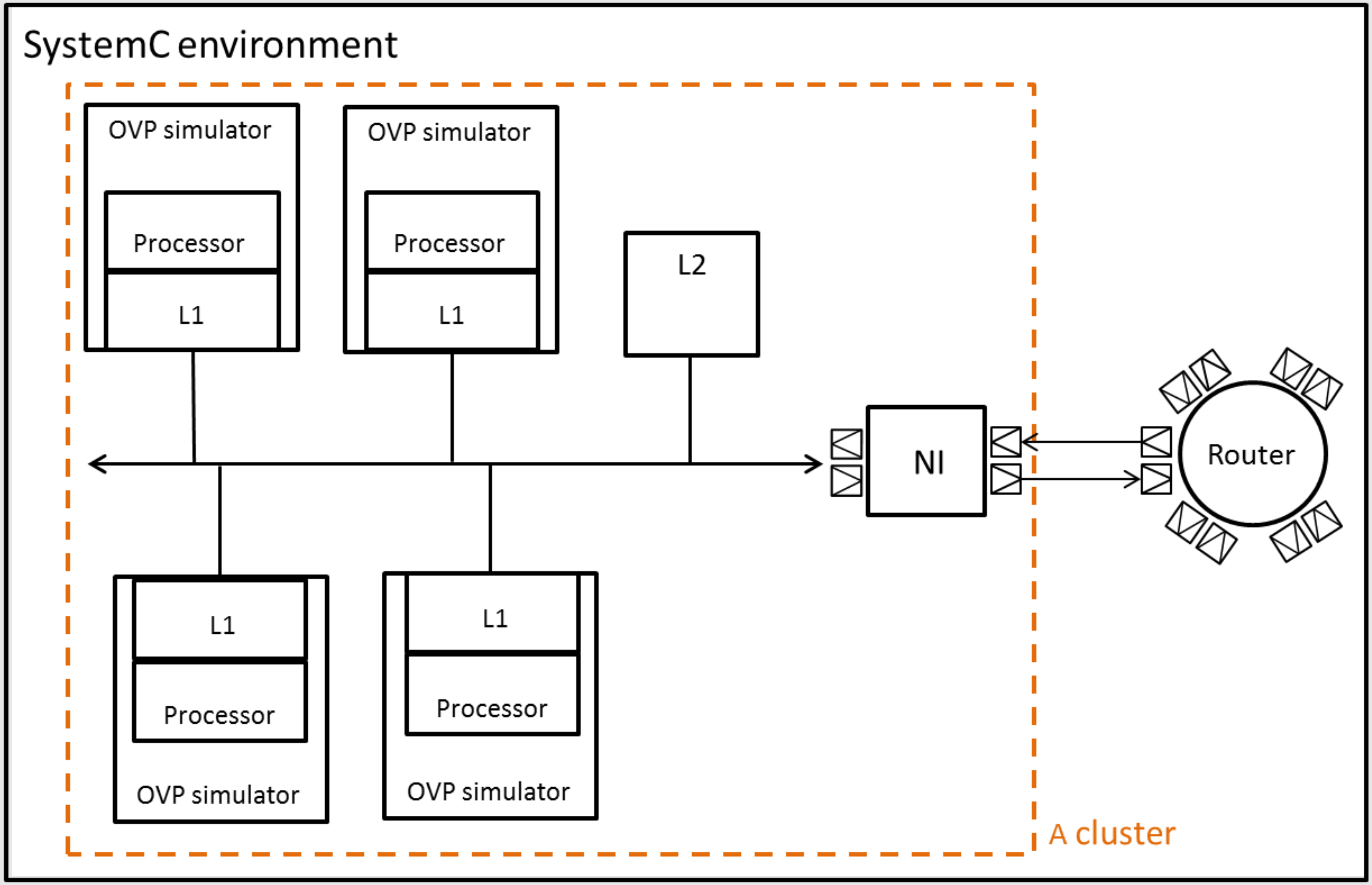
Secure-enable mechanisms

- **Task mapping :** Leverage the data accesses locality
- **Isolated application -> secure zone creation:**
  - Static SZ size
  - Dynamic SZ size



Evaluation environment

OVP-based MPSoCSim



[8] M.Méndez Real et al., "MPSoCSim extension: An OVP Simulator for the Evaluation of Cluster-based Multicore and Many-core architectures," in Proc. Workshop on Virtual Prototyping of Parallel and Embedded Systems (ViPES) as part of the International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XV), 2016.

Experimental setup

Experimental protocol:

- 4x4 clusters architecture (60 MB + 1 ARM)
- Matrix multiplications, 17 parallel tasks, unfavorable case => 5 clusters needed, only 17 used. 5 applications meaning 85 tasks running in parallel

Different deployment strategies:

1. Secure zones of fixed size
2. Secure zones with dynamic size

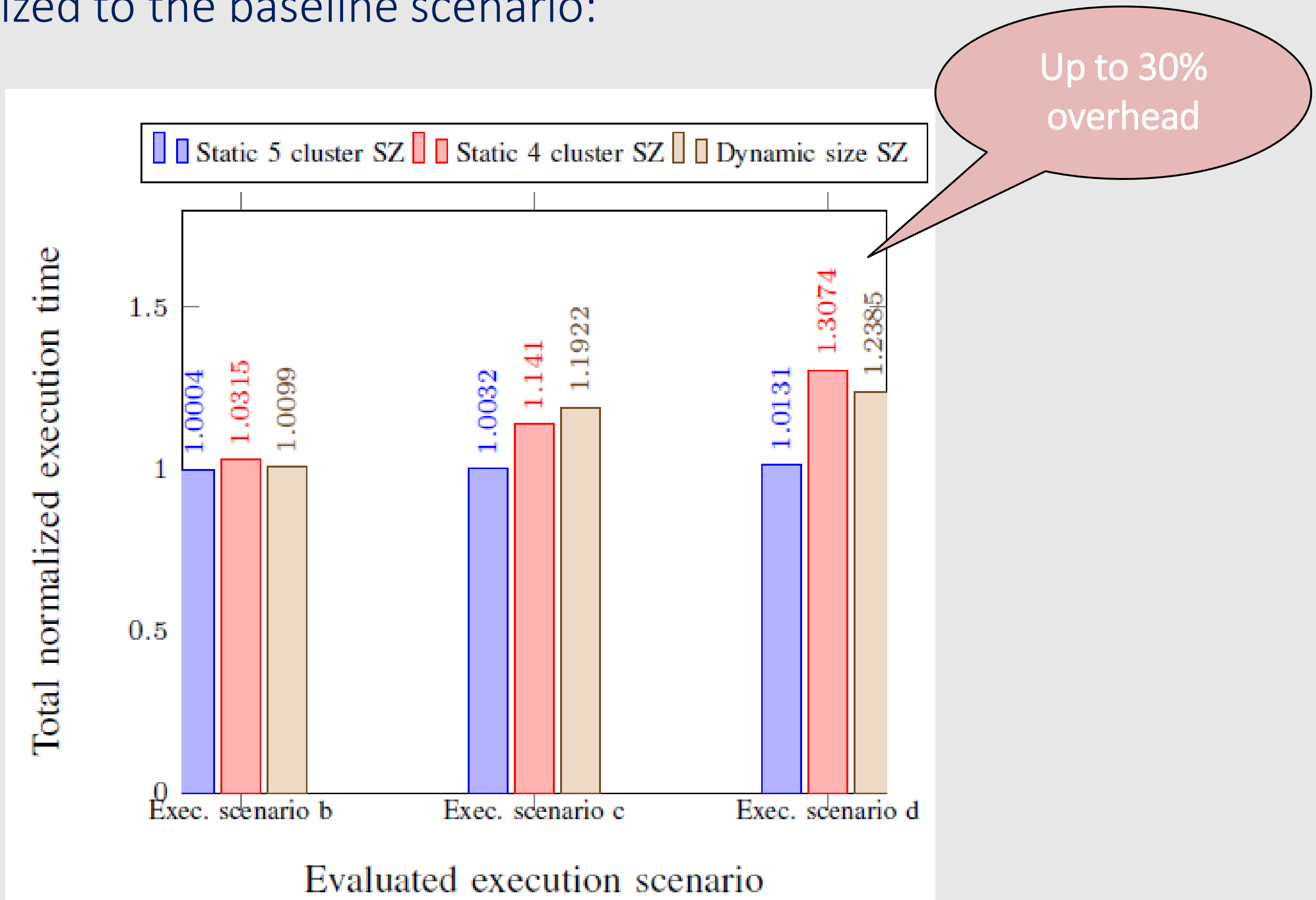
Different execution scenarios:

a. Baseline scenario
b. One priority isolated application
c. One treated at the middle of the execution
d. Three isolated applications

ARM frequency	667 MHz
MB frequency	100 MHz
Nom MIPS	100
realFlitTime(ARM)	850 ns
realFlitTime(MB)	40 ns
Network frequency	100 MHz
Network size	**4*4**
Processors per cluster	**4**
Clock delay pass through	1 cycle

Results

Total execution time normalized to the baseline scenario:

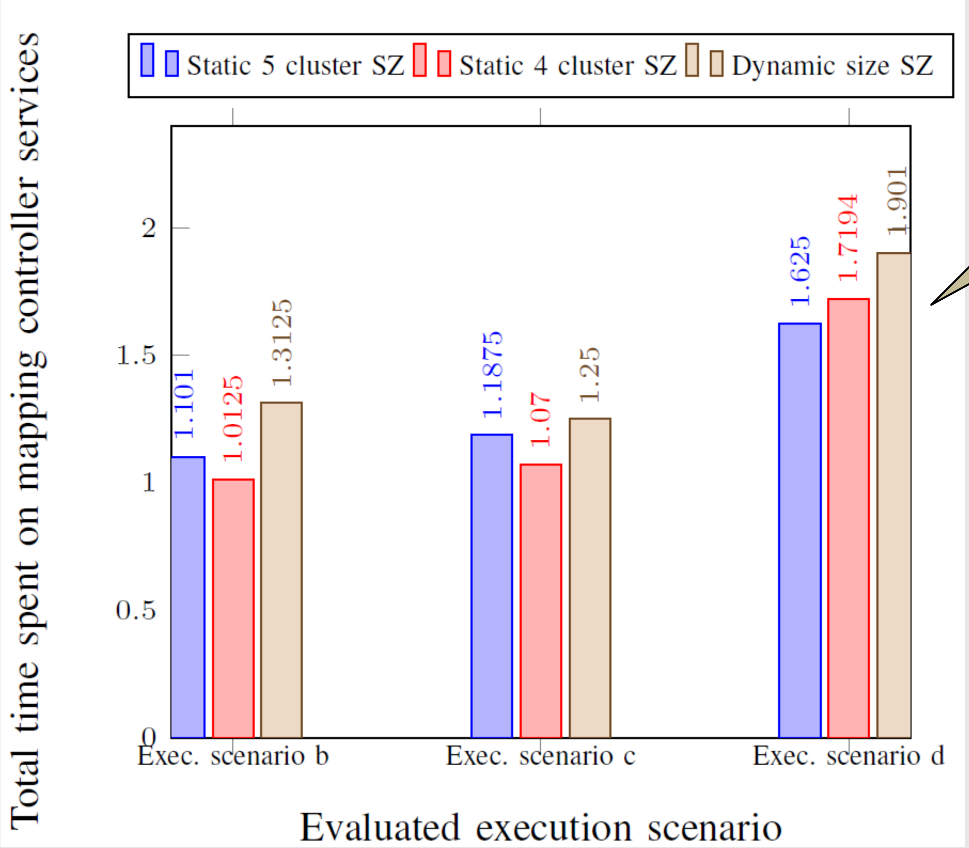


b. One isolated application with the highest priority
c. One isolated application with a medium priority
d. Three isolated applications

[9] M. Méndez Real et al., "Dynamic Spatially Isolated Secure Zones for NoC-based Many-core Accelerators", 11th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2016.

Results

Time spent on the mapping related controller services:



The dynamic strategy entails the highest overhead on the controller services

b. One isolated application with the highest priority
c. One isolated application with a medium priority
d. Three isolated applications

Results

Exec. time of non isolated, isolated application (in msec.) and in average:

	Evaluated deployment strategy and execution scenario	Total exec. time	Exec. time in average for non-isolated applications	Exec. time in average for isolated applications	Exec. time in average
	baseline	**363.35**	**202**	**-**	**202**
1. Static SZ size (5 clusters)	(1.a)	363.49	206	163	197
	(1.b)	364.51	338	214	313
	(1.c)	368.10	242	181	205
	Average for 1.		262 (+29%)	186 (-8%)	237 (+17%)
2. Static SZ size (4 clusters)	(2.a)	374.79	261	198	248
	(2.b)	414.58	264	269	265
	(2.c)	374.51	272	270	271
	Average for 2.		265 (+31%)	245 (+21%)	261 (+29%)
3. Dynamic SZ size	(3.a)	366.94	223	159	210
	(3.b)	433.18	261	400	253
	(3.c)	450.00	214	376	264
	Average for 3.		232 (+14%)	311 (+53%)	242 (+19%)

The dynamic strategy leverages the performance of non isolated applications

b. One isolated application with the highest priority
c. One isolated application with a medium priority
d. Three isolated applications

Results

Resources utilization rate:

Evaluated deployment strategy and execution scenario	SZ resources utilization rate during the time of the SZ	Total resource utilization rate in average
a. (baseline)	-	**77%**
1. b.	85%	68.5%
1. c.	85%	71%
1. d.	85%	61.6%
2. b.	65%	64%
2. c.	65%	69%
2. d.	65%	55%
3. b.	85%	72%
3. c.	89%	69%
3. d.	92%	67%

1. Static SZ size
5 clusters

2. Static SZ size
4 clusters

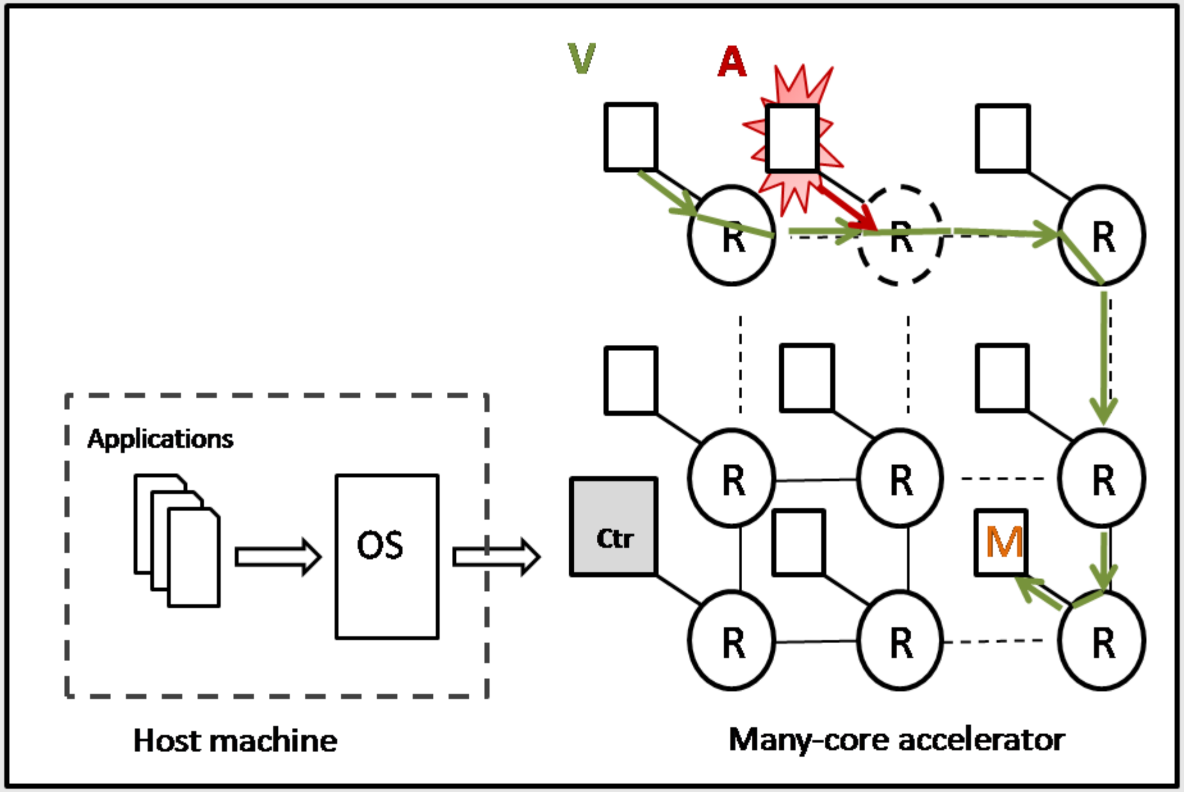
3. Dynamic SZ size

The dynamic strategy achieves the highest resource utilization rate

Conclusion and future work

- Spatial isolation of sensitive applications
- Evaluation on different execution scenarios with different strategies through virtual prototyping
- A global performance overhead up to 30%

Future work: NoC protection



Thank you for your attention!