# Performance Analysis of Static and Dynamic NUCA Caches

**Exploring the Relationship between Architectures and Management Policies in the design of NUCA-based Chip Multicore Systems**

Sandro Bartolini, Pierfrancesco Foglia, Cosimo Antonio Prete

**Analysis of Static and Dynamic Energy Consumption in NUCA Caches**

Alessandro Bardine, Pierfrancesco Foglia, Giacomo Gabrielli, Cosimo Antonio Prete

1

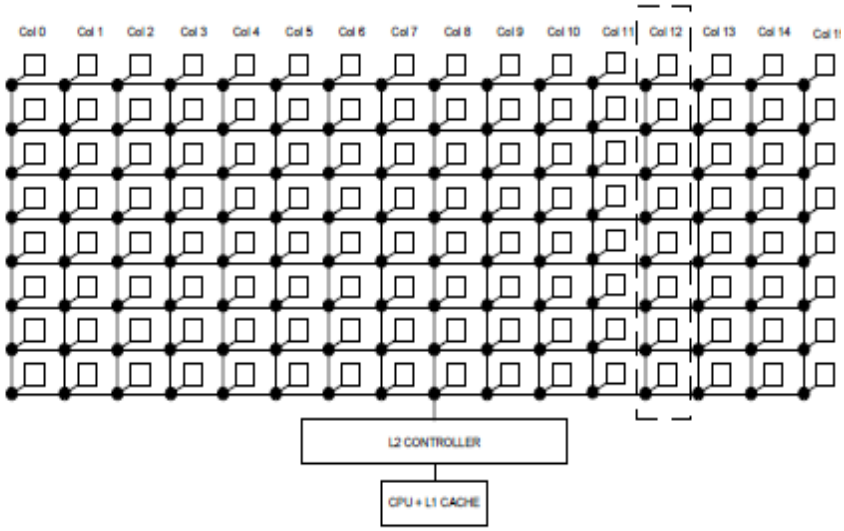
## Non Uniform Cache Architectures (NUCA)

- Technology trends are leading to the use of large, on-chip, level-2 (L2) and level-3 (L3) cache memories.
- For high frequency systems, the latencies of such caches are dominated by wire delay.
- **In a NUCA architecture, the cache is partitioned into many independent banks usually interconnected by a switched network; in this model the access latency is proportional to the physical distance of the banks from the cache controller.**
- NUCA caches reduce the effects of the consequent high access latencies in big caches.

2

# Non Uniform Cache Architectures (NUCA)

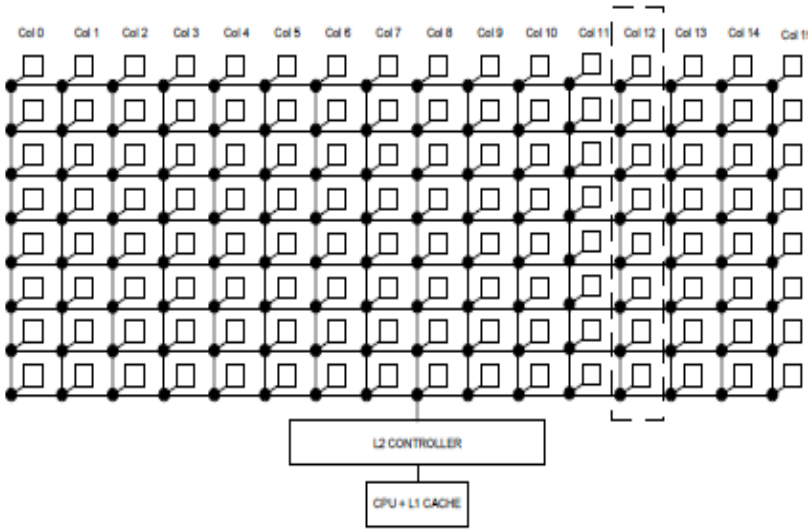
**In a NUCA architecture, the cache is partitioned into many independent banks usually interconnected by a switched network; in this model the access latency is proportional to the physical distance of the banks from the cache controller.**




3

# S-NUCA and D-NUCA

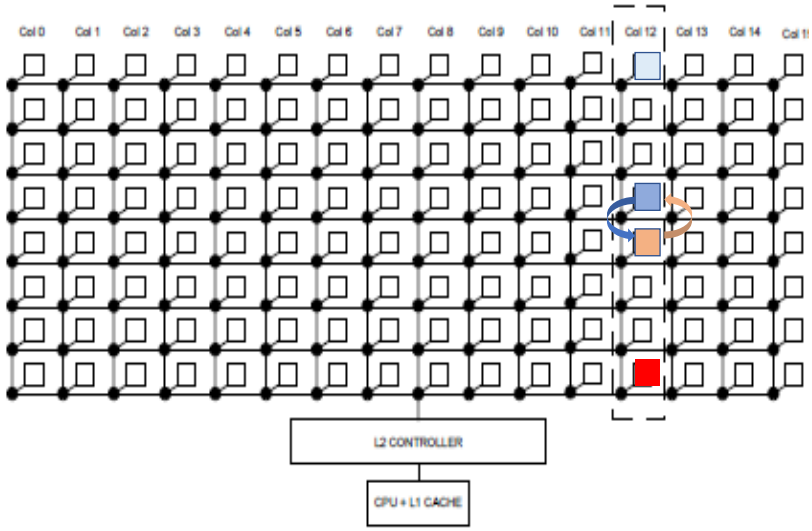
- The mapping between cache lines and banks can be either static or dynamic (namely S-NUCA and D-NUCA); in the former, each line can exclusively reside in a single predetermined bank, while in the latter a line can dynamically migrate from one bank to another.
- Both S-NUCA and D- NUCA caches have proven to outperform traditional UCA (Uniform Cache Architecture) caches in large size, wire dominated designs.




4

# D-NUCA

- In D-NUCA, a block can dynamically migrate from one bank to another.
- The most frequently accessed data are likely to be located in the closest banks to the cache controller.




5

# Configuration parameters

| | UCA | S-NUCA | D-NUCA |
|---|---|---|---|
| Size | 8 MB | 8 MB | 8 MB |
| Line size | 64 B | 64 B | 64 B |
| N. of banks | 1 | 32 | 128 |
| N. of sub-banks | 2 | 1 | 1 |
| N. of bank rows | 1 | 8 | 16 |
| N. of bank columns | 1 | 4 | 8 |
| Bank size | 8 MB | 256 KB | 64 KB |
| Bank associativity | 4-way | 4-way | direct mapped |
| Bank latency (cycles) | 18 | 5 | 3 |
| Link latency (cycles) | - | 2 | 1 |
| Link width (bits) | - | 2x128 | 2x128 |

We considered a single processor system with microarchitectural parameters matching those of the Alpha 21264 processor.

We selected a clock frequency equal to 5 GHz operating frequency and 70 nm technology.

The processor is backed by a 64 KB 2-way setassociative L1 I-cache with single cycle access latency and by a 64 KB 2-way set associative L1 D-cache with 3 cycles latency.

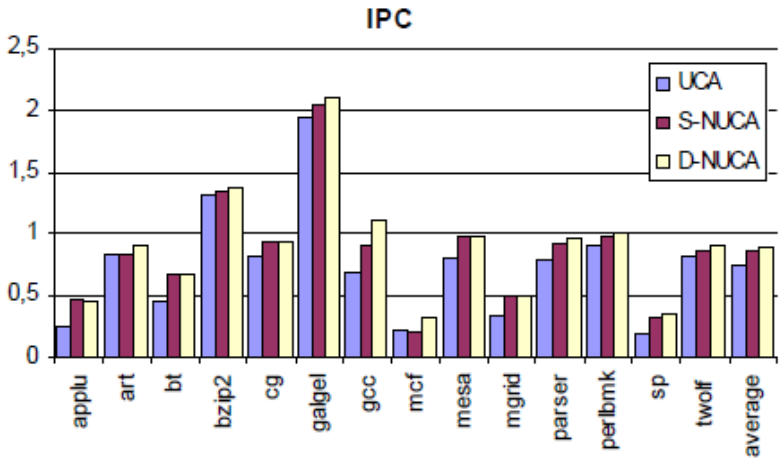
The cache line size is fixed at 64 bytes.

L2 caches were assumed to have 8 Mbytes capacity with block size fixed at 64 bytes.

For each typology (UCA, S-NUCA, D-NUCA) the best performing configuration was selected.

6

# Comparison of the IPC (instructions per clock)

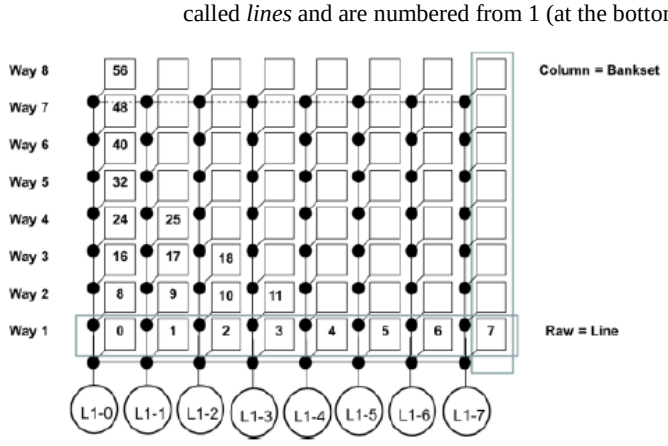


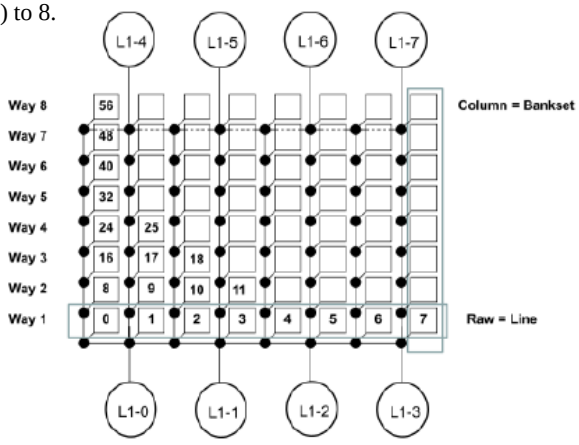

**D-NUCA outperforms S-NUCA by 4.98% and UCA by 20.77% (average values).**

7

# "one-side" and "two-sides design in CMP

Squares are NUCA banks, black circles are NoC switches, white circles represent cpu+L1 caches. In the case of a DNUCA, bank columns correspond to banksets. Rows of the bank matrix are called *lines* and are numbered from 1 (at the bottom) to 8.




8

# Two basic layouts

In the “one-side”, all cores are placed on the same side of the shared LLC.

- Adopted in the Intel Core Gulftown microarchitecture (e.g. Core i7 980X - 6 cores), in the Intel Core Sandy Bridge or Ivy Bridge microarchitecture (e.g. i7 2600K or i7-3770K - 4 cores+ GPU)), in the Xeon Broadwell microarchitecture and in the AMD Opteron Magny Cours;

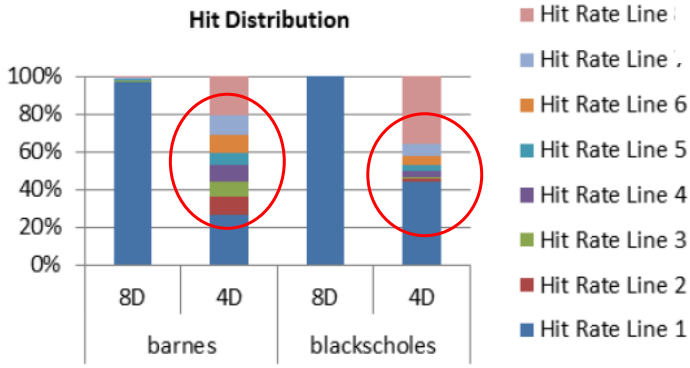
In the “two-sides”, half cores are positioned on one side and the other half on the other.

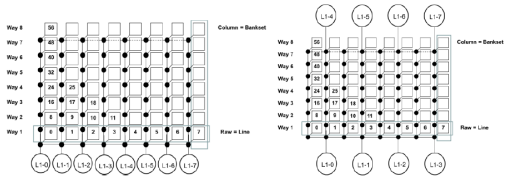
- The scheme “two-sides” is exemplified by the Intel core Sandy Bridge-E (i7 3960 - 6 cores) and Ivy Bridge-EP microarchitecture, IBM Power 7 and Power 8 multicore families,, Oracle SPARC M7 Processor.

Power 7 and 8 CMP families utilize the banked L3 cache as a victim for the L2 cache, putting evicted L2 data in banks “near” to the L2 cache originating the data, similarly to the Sparc M7 processor, de facto all leveraging NUCA features. Furthermore, some i7 processor sub-families expose NUCA organizations for their L3 cache.

9

# Hit distribution in D-Nuca schema for “one-side” and “two-sides design



8D is a DNUCA scheme with an 8P layout, while 4D is the same scheme on a 44P layout.

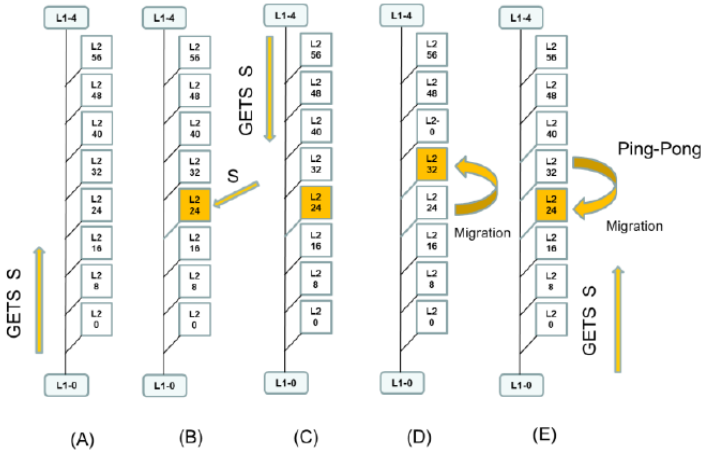
In the 8P layout, both the applications exhibit more than 96% of the accesses to the fastest way (line 1).

In the 44P, accesses are spread throughout the lines in different ways for the two applications.

It is expected that performance difference when moving from 8P to 44P is higher in Barnes, as accesses in the 44P layouts are almost to all the lines.

10

# Conflict hits and the ping-pong phenomenon in “two-sides” layout




**B** L1-0 requests data S (A), that is taken from main memory and put in the L2-24 block.

**C** Then L1-4, located at the opposite side of the cache with respect to L1-0, requests the same data.

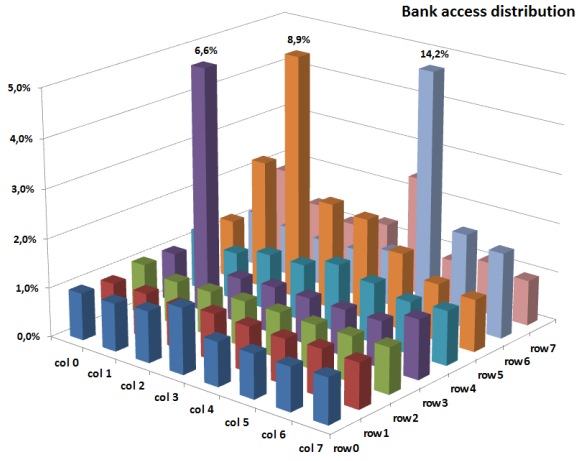
**D** The data now hits in the cache, and migrates towards L1-4 (the requestor) in the bank L2-32.

**E** If L1-0 requests again the data S, it migrates towards L1-0, but it does not improve the access latency with respect to (B), nullifying the benefit of migration.

11

# Program memory layout optimizations

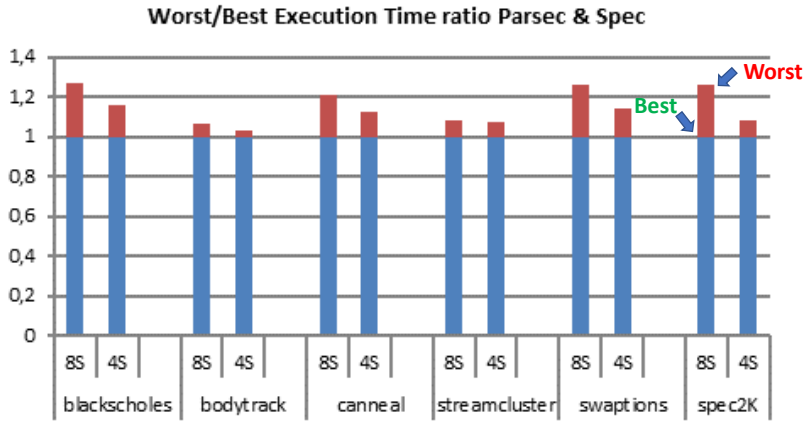
- The general observation that performance is significantly affected by the access distribution to the LLC banks suggests that specific memory layout optimizations of the application can the average SNUCA access latency.
- In principle, it is possible to improve the performance of the cache, by mapping the mostly accessed data to the LLC banks near to the requesting cores.
- For instance, this is what a DNUCA cache tries to achieve in hardware at run-time, promoting the cache lines close to the requesting cores upon a hit event.
- A similar approach can also be implemented via software, at compile time, by utilizing a profiling phase which collects information on how accesses are distributed in the LLC, and then using such information to optimize the mapping of data and code to the LLC banks in order to improve overall cache access time or other performance metric.




12

# Execution Time

the best and the worst performing mapping layout for the Parsec & Spec

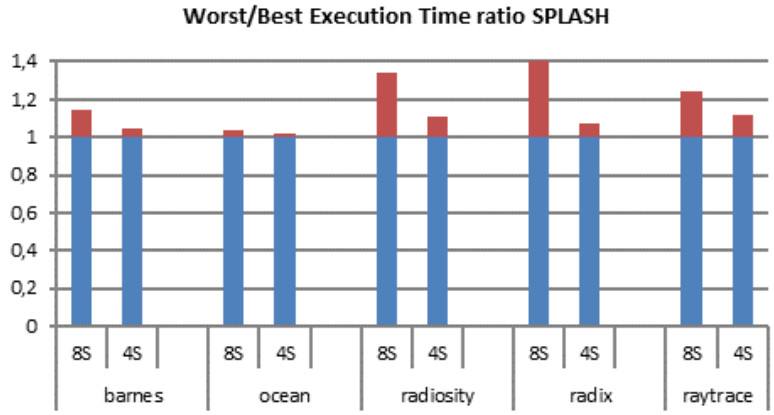



The 44P layout is less sensitive to data layouts, although the best performance improvements can be achieved with the 8P layout

13

# Execution Time

the best and the worst performing mapping layout for the SPLASH




The 44P layout is less sensitive to data layouts, although the best performance improvements can be achieved with the 8P layout

14

# Summary of the experiments performed and related configurations

| | |
|---|---|
| *Processors* | *8 cores (64-bit), connected all to one side (8P) or four to each of two opposite sides (44P)* |
| *Clock Frequency/tech.* | *4 GHz / 32 nm* |
| *Coherence Scheme* | *Directory based MESI* |
| *L1 cache* | *Private 16 kBytes I + 16 kBytes D, 2 way set associative, 1 cycle to TAG, 2 cycles to TAG+Data* |
| *L2 cache* | *16 MBytes, 64 banks, 64 bytes block* |
| *L2 cache bank* | *256 kBytes, 4 way set associative, sequential, 2 cycles TAG, 5 Cycles TAG+Data.* |
| *NoC configuration* | *Partial 2D Mesh Network; 256 bit flit; NoC switch latency: 1 cycle; NoC link latency: 1 cycle* |
| *Main Memory* | *2 GBytes, 240 cycles latency* |

| *Name* | *Layout* | *NUCA Organizations* | *Optimizations* |
|---|---|---|---|
| 8S | 8 cores at one side of the cache (8P) | SNUCA | None |
| 8S_rem | 8 cores at one side of the cache (8P) | SNUCA | Software mapping of the mostly accessed memory blocks near to the cores using them (Section 4.3) |
| 8D | 8 cores at one side of the cache (8P) | DNUCA | None |
| 4S | 8 cores, 4 at one side of the cache, 4 at the opposite side (44P) | SNUCA | None |
| 4S_rem | 8 cores, 4 at one side of the cache, 4 at the opposite side (44P) | SNUCA | Software mapping of the mostly accessed memory blocks near to the cores using them (Section 4.3) |
| 4D | 8 cores, 4 at one side of the cache, 4 at the opposite side (44P) | DNUCA | None |
| 4D_RE | 8 cores, 4 at one side of the cache, 4 at the opposite side (44P) | DNUCA | Replication of a copy once accessed by cores at the opposite sides of the LLC (section 4.2) |

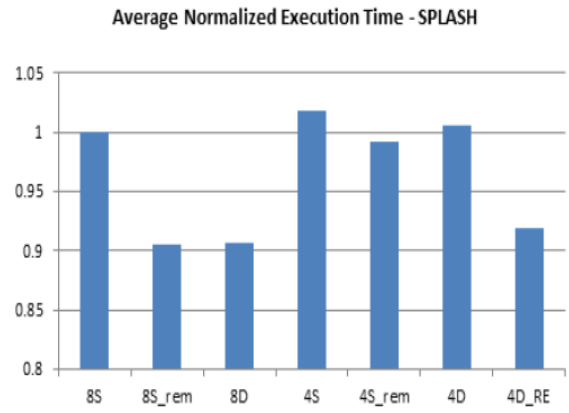
15

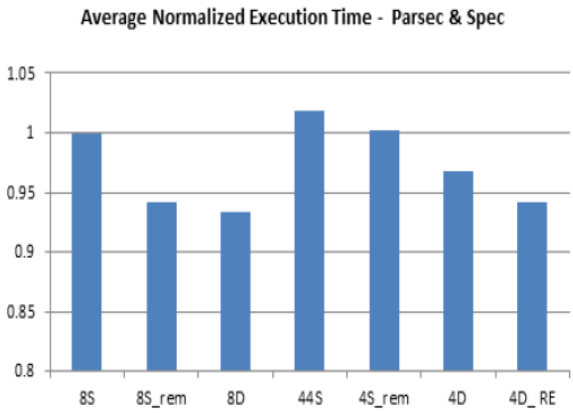
# *Splash-2 and Parsec workloads*

| SPLASH-2 | |
|---|---|
| Barnes | N-body Simulation |
| Ocean | Ocean Current Simulation |
| radiosity | Graphics |
| Radix | Integer Sort |
| raytrace | 3D Rendering |
| | |
| PARSEC 2.0 | |
| blackscholes | Financial Analysis |
| bodytrack | Computer Vision |
| canneal | Engineering |
| streamcluster | Data Mining |
| swaptions | Financial Analysis |
| | |
| SPEC2K | |
| Mcf | Combinatorial Optimization |
| bzip2 | Compression Algorithm |
| Art | Image Recognition / Neural Networks |
| Parser | Word Processing |

16

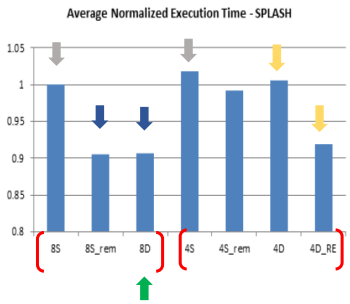
# Average Normalized Execution Time for the set of SPLASH and Parsec & Spec applications

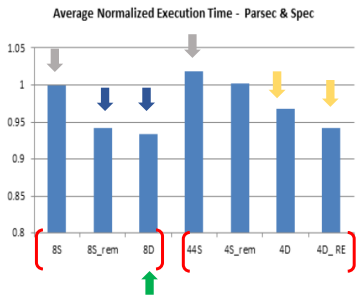



17

# Average Normalized Execution Time for the set of SPLASH and Parsec & Spec applications

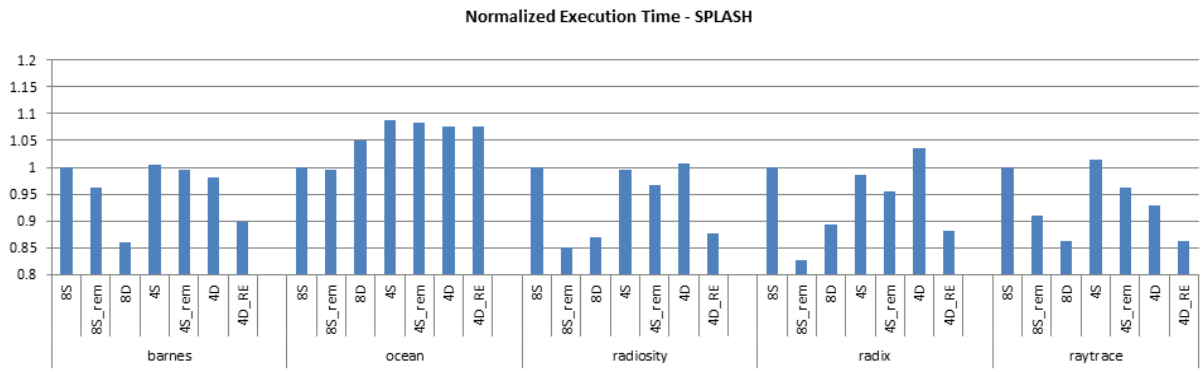



- **The DNUCA cache on the 8P layout (8D) is the most performing solution for all the benchmarks.**
- **In the 44P layout, DNUCA caches get lower gains than in the 8P one (on average, 1% for SPLASH and 4% for Parsec & Spec).**
- **In the 8P configuration, performance like a cache DNUCA can be achieved with a simpler SNUCA architecture by adopting the data layout optimization**
- **In a 44P layout, significant performance improvements in DNUCA can be achieved by adopting, together with migration, a replication policy that limits the effect of ping-pong (conflict hits).**
- **In the case of SNUCA cache, there are no significant differences in performance, on average, when the applications are executed in a 44P layout or in a 8P layout.**

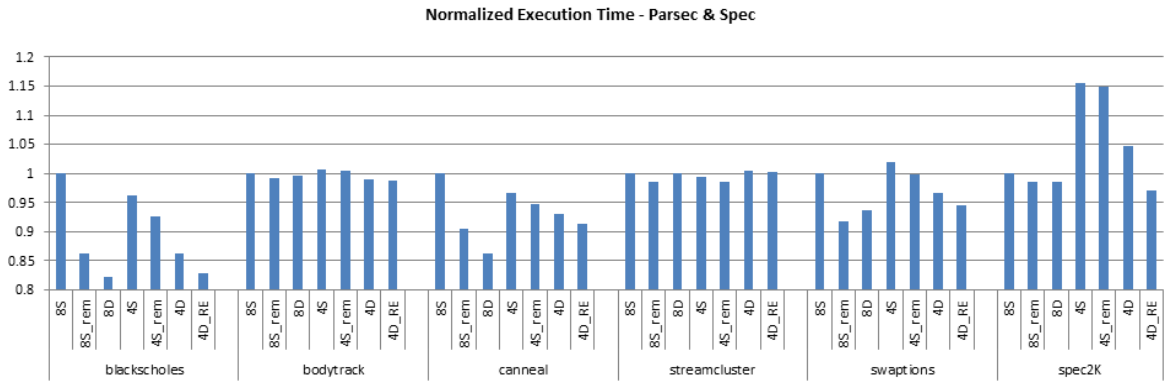
18

# Normalized Execution Time for the SPLASH applications



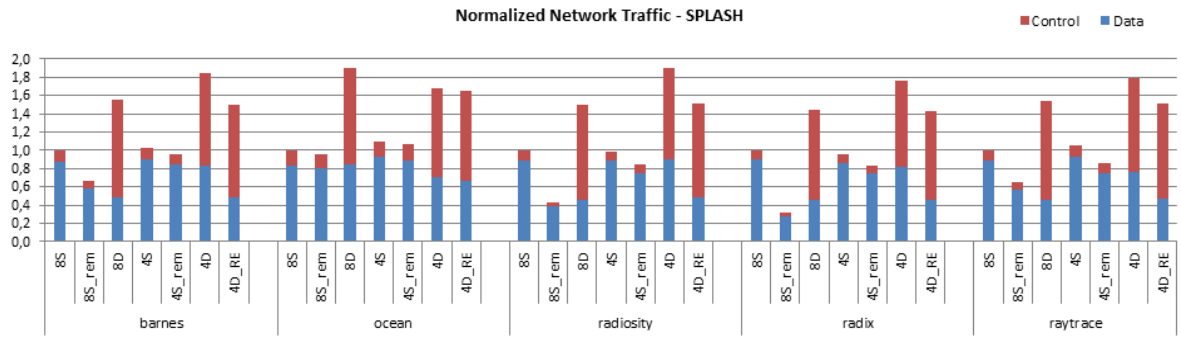

19

# Normalized Execution Time for Parsec & Spec



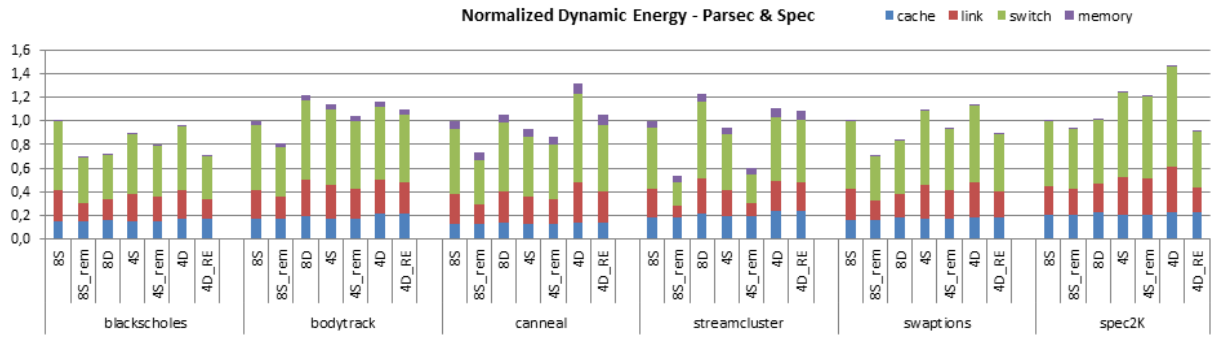

20

# Normalized NoC traffic for the SPLASH applications




21

# Energy consumption

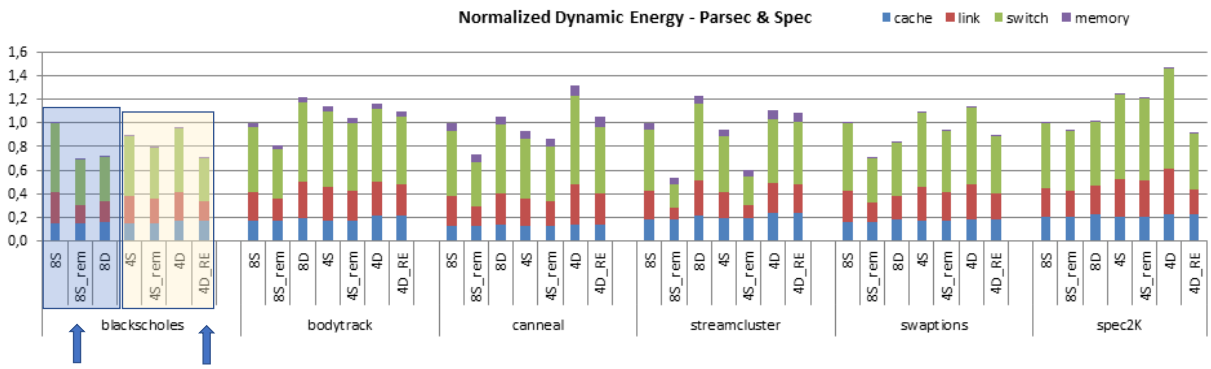



22

# Energy consumption

**8S_rem solution delivers the lowest dynamic energy consumption for all the benchmarks.**

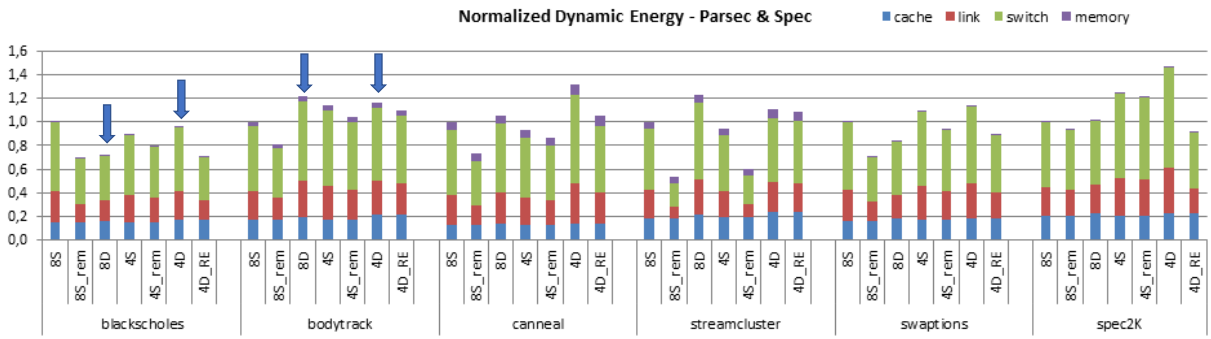
**8S_rem and 4D_RE solutions induce the lowest overall energy consumption.**




23

# Energy consumption

**Dynamic energy consumption for DNUCA solutions is highly sensitive to the layout and the behavior of the applications**.




24