Inria

Using Bandwidth Throttling to Quantify Application Sensitivity to Heterogeneous Memory

Clément Foyer, Brice Goglin
Bordeaux, France

Outline

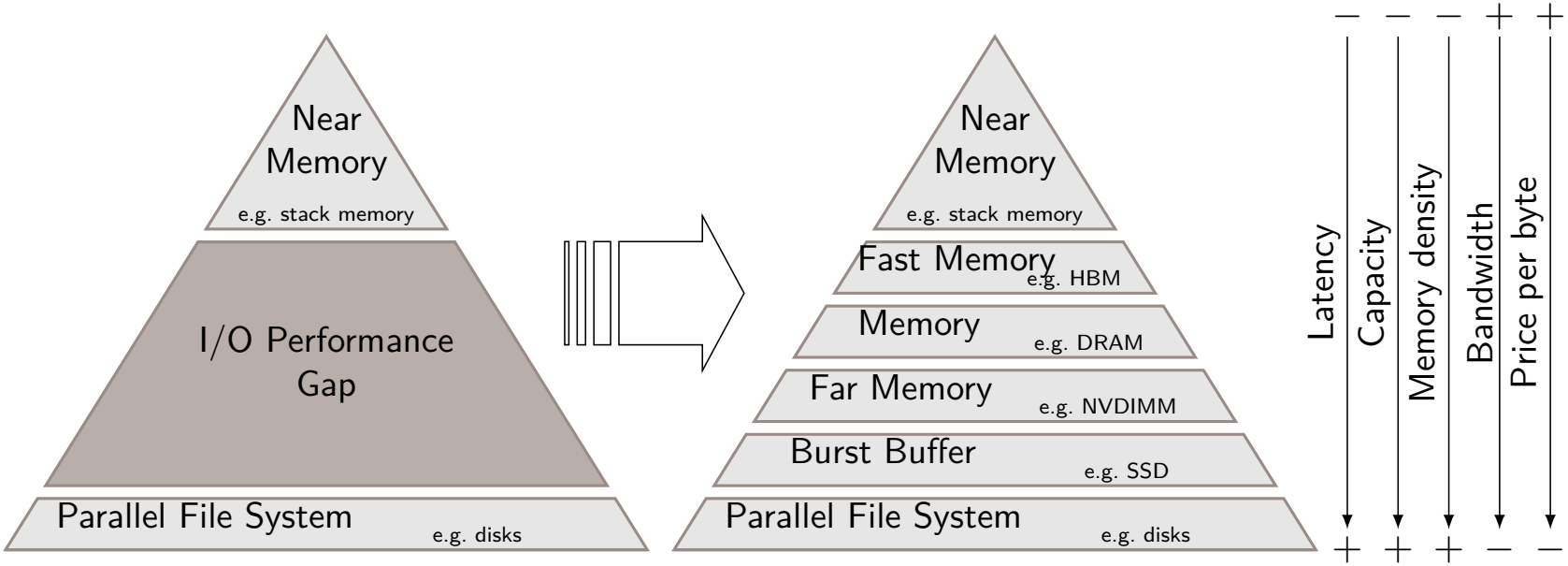
Inria

01

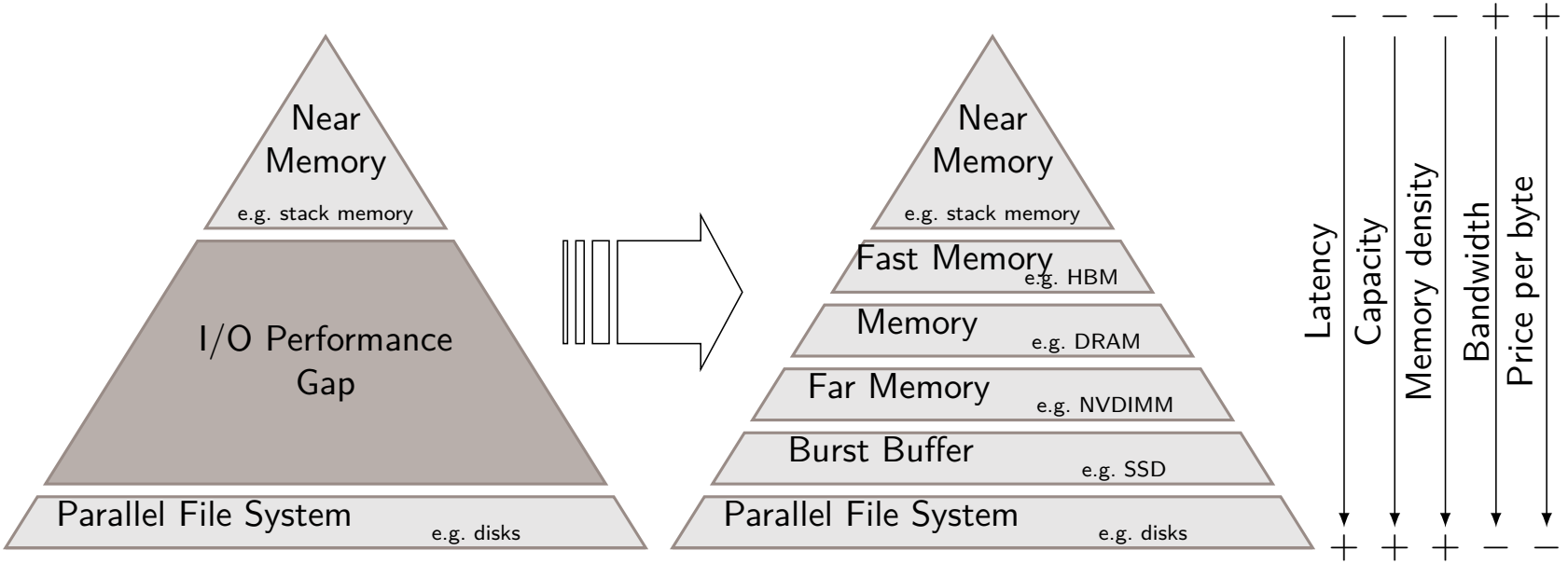
Problem Specification

Inria

Complex memory hierarchy



Complex memory hierarchy



Buffers with different access patterns

- Streamed accesses (Predictable, bandwidth bound)
- Pointer chasing (Unpredictable, latency bound)
- Random accesses (Unpredictable, latency bound)

How and where to allocate memory?

- It depends on the needs
  - Where is the computation happening?
  - What are the memory systems available?
  - How is the data accessed?

How and where to allocate memory?

- It depends on the needs
  - Where is the computation happening?
  - What are the memory systems available?
  - How is the data accessed?

- How portable is the solution?
  - Relying on vendors' solutions (manually)
  - Relying on heterogeneous aware libraries (Umpire, SICM, etc.)
  - Relying on the system allocations' policy to use the correct NUMA node

Inria

How and where to allocate memory?

- It depends on the needs
  - > Where is the computation happening?
  - > What are the memory systems available?
  - > How is the data accessed?

- How portable is the solution?
  - > Relying on vendors' solutions (manually)
  - > Relying on heterogeneous aware libraries (Umpire, SICM, etc.)
  - > Relying on the system allocations' policy to use the correct NUMA node

Last approach privileged

Problem: the NUMA distance doesn't enclose all related information.

Inria

Problem Statement

- Data placement is essential for achieving optimal performance
- We need a quantitative evaluation of data “needs”
- We need a quantitative evaluation of memory characteristics

- Data placement is essential for achieving optimal performance
- We need a quantitative evaluation of data "needs"
- We need a quantitative evaluation of memory characteristics

The general problem becomes

How to evaluate the fit between complex data accesses and complex memory architecture?

Inria

02

Creating Heterogeneity

Inria

- New technologies are required to test the placement algorithms
  - Either we buy expensive new hardware
    - very few different platforms available yet
  - Or we find a way to create heterogeneity within the existing systems

- New technologies are required to test the placement algorithms
  - Either we buy expensive new hardware
    - very few different platforms available yet
  - Or we find a way to create heterogeneity within the existing systems
- Bandit (Bandwidth or Latency)
- Resource Control
  - Added in Linux kernel 4.10
  - Expose control over L3 cache
  - Cache partitioning
  - Bandwidth throttling

Inria

- New technologies are required to test the placement algorithms
  - Either we buy expensive new hardware
    - very few different platforms available yet
  - Or we find a way to create heterogeneity within the existing systems
- Bandit (Bandwidth or Latency)
- Resource Control
  - Added in Linux kernel 4.10
  - Expose control over L3 cache
  - Cache partitioning
  - Bandwidth throttling
- Bandwidth throttling affect the L3 cache $\leftrightarrow$ DRAM I/O bus
- Granularity can be in percentage (Intel) of the total BW or arbitrary (AMD)

Inria

Platform evaluation: Intel

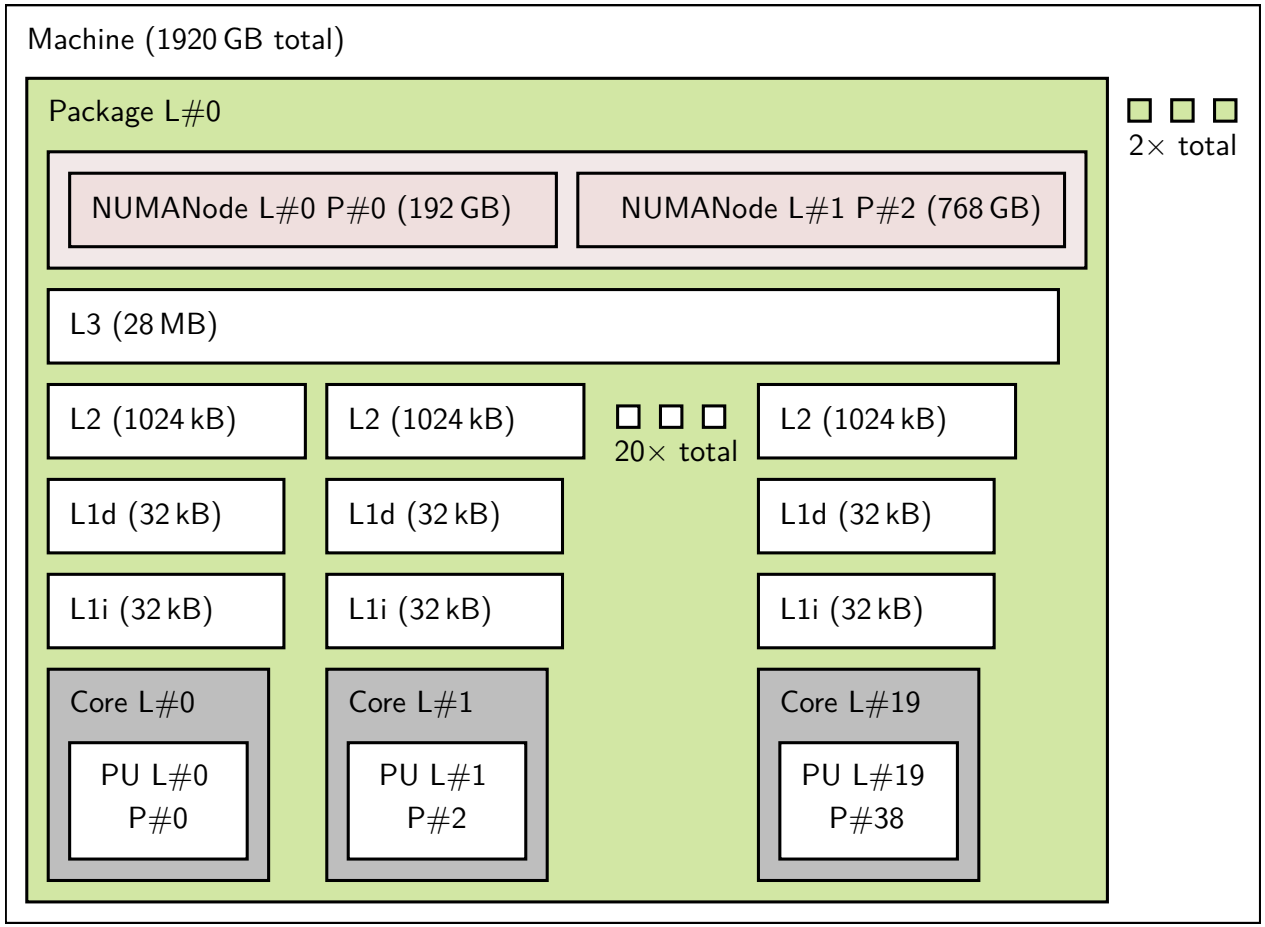
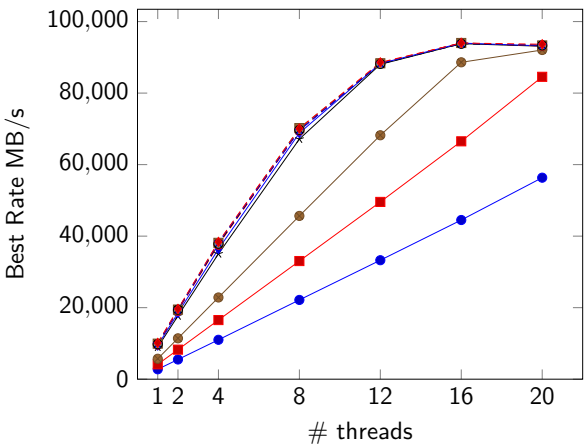
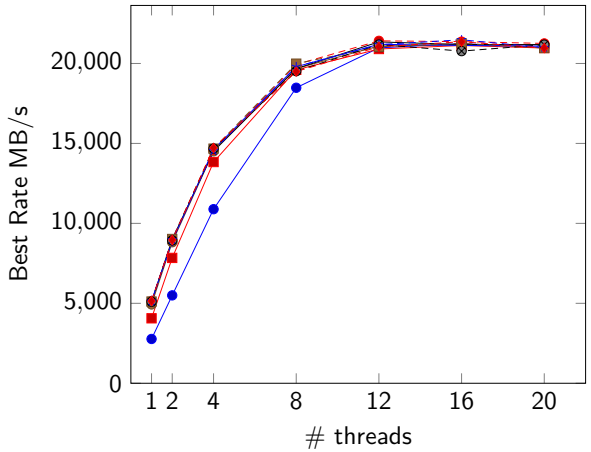


Figure: Topology of the dual Intel *Cascade Lake* Xeon Gold 6230 platform as reported by `lstopo` (factorized version), with 2 Optane PMM DIMMs (dax mode).

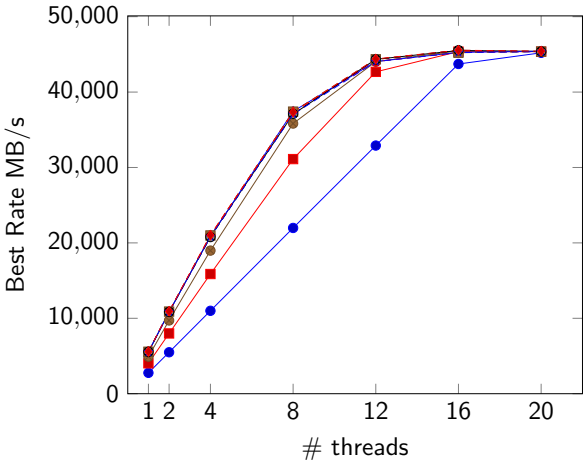
Platform evaluation: Intel



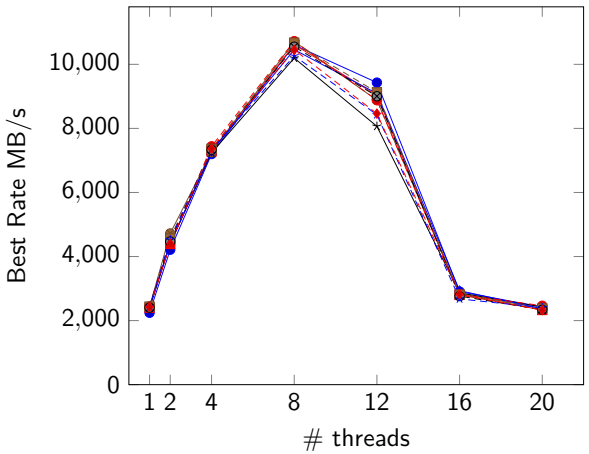
(a) DRAM (NUMA#0)



(b) NVDIMM (NUMA#1)



(c) cross-NUMA DRAM (NUMA#2)



(d) cross-NUMA NVDIMM (NUMA#3)

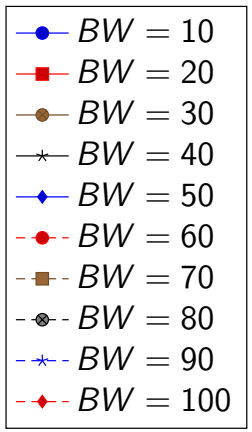


Figure: STREAM benchmark

Platform evaluation: Intel

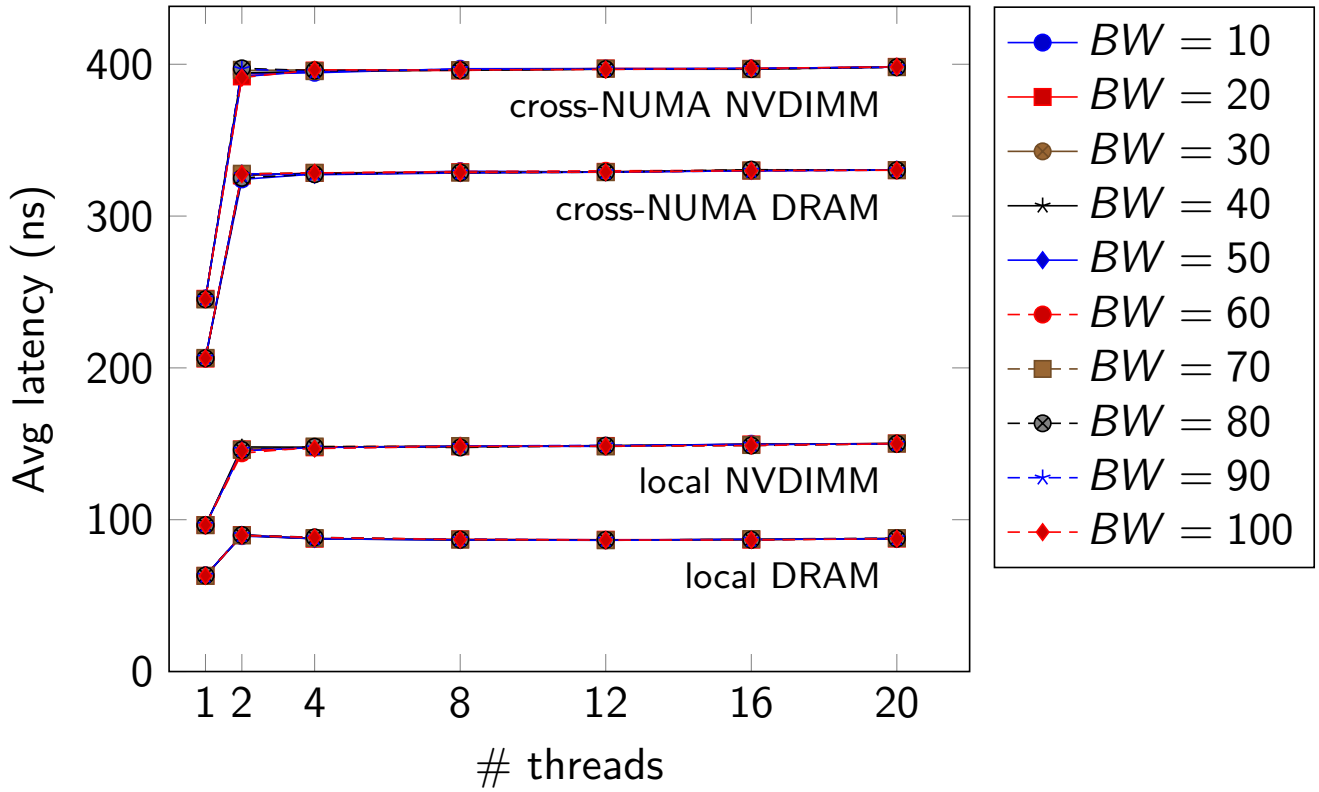


Figure: Latency depending on the number of threads and `resctrl` setting

- Big granularity when allocating the bandwidth
  - > 10 % steps
- Strong NUMA effect on bandwidth
- Little effect of the bandwidth throttling on application
  - > Need to use a very restrictive setting to observe an effect
- Bandwidth limitation shows no effect on latency

Platform evaluation: AMD

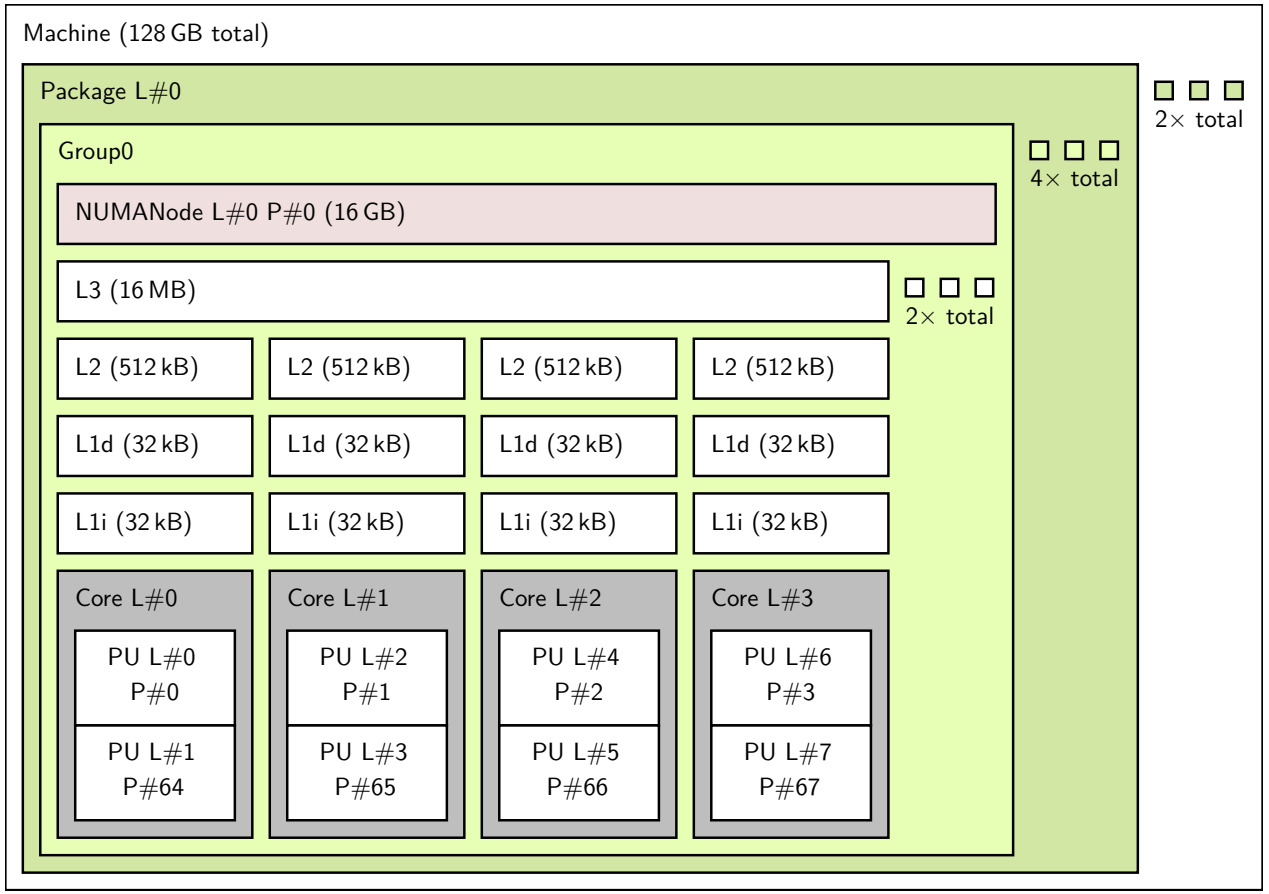
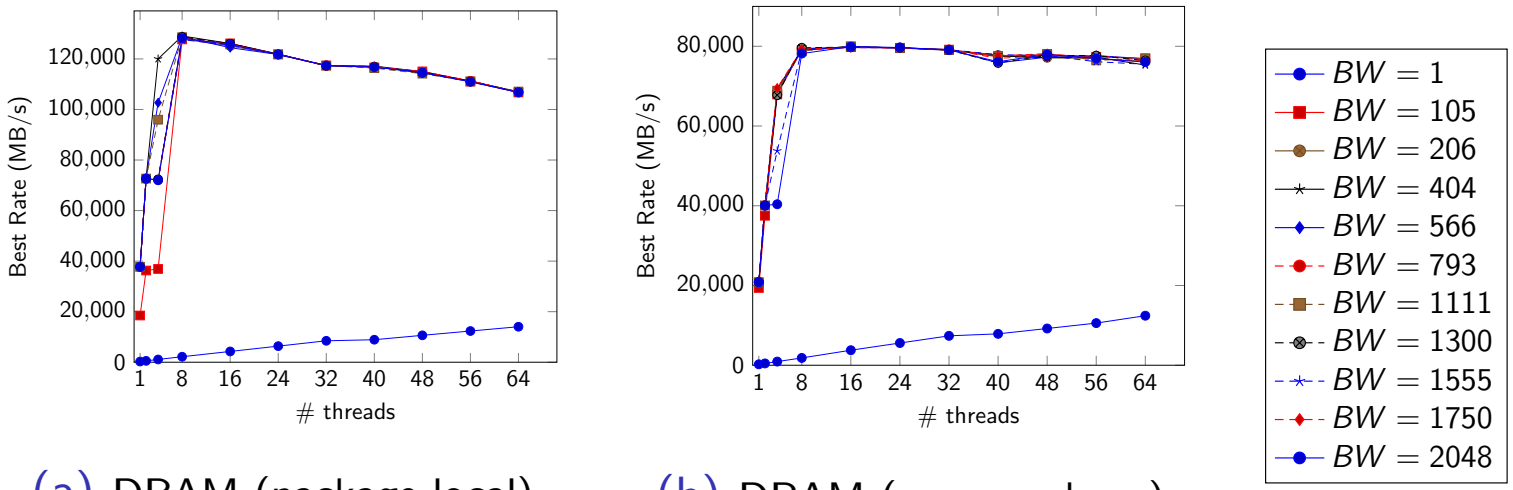


Figure: Topology of the dual AMD *Zen2 Rome* EPYC 7502 platform, as reported by `lstopo` (factorized version).

Inria

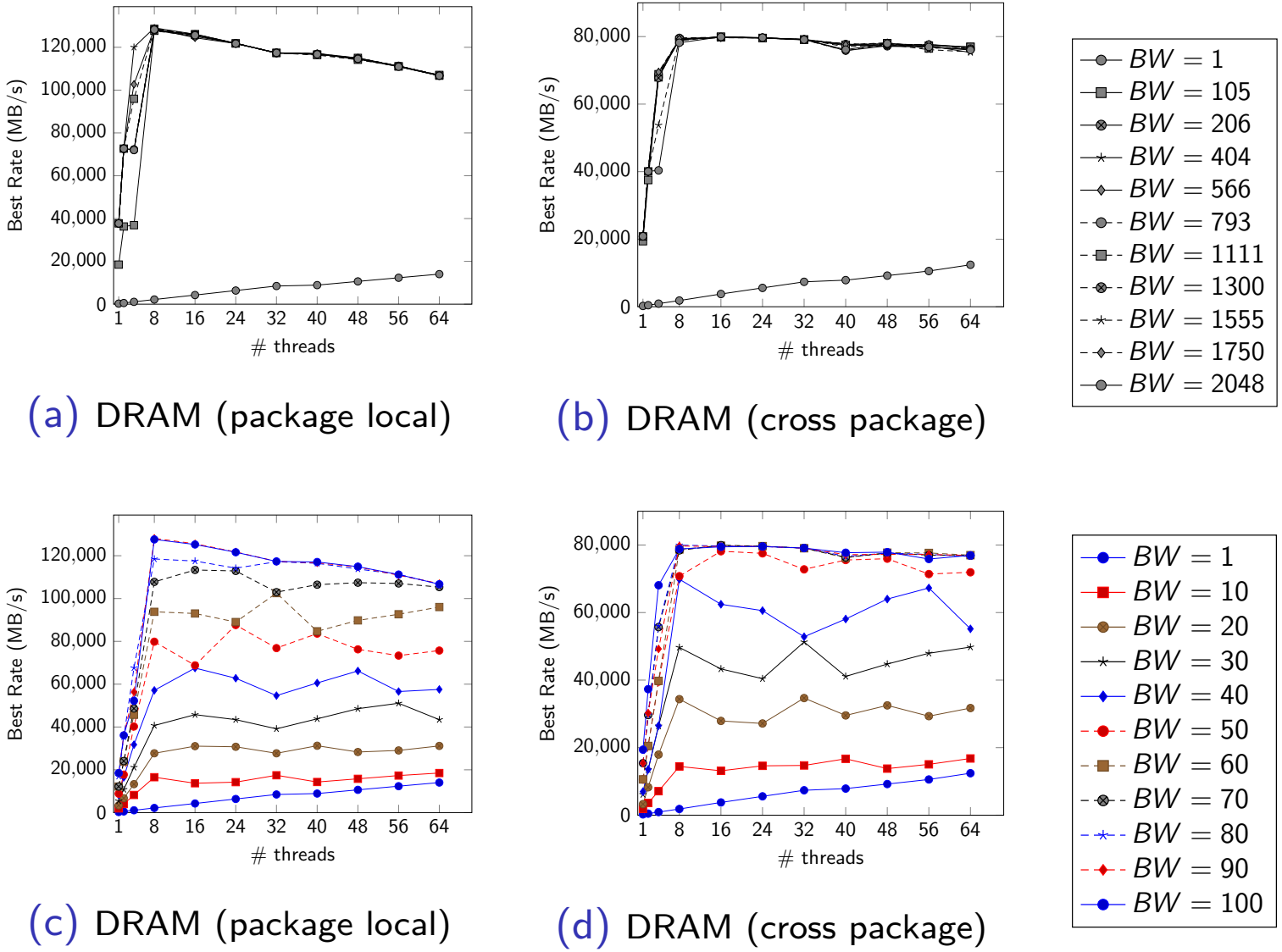
Platform evaluation: AMD



(a) DRAM (package local)

(b) DRAM (cross package)

Platform evaluation: AMD



(a) DRAM (package local)

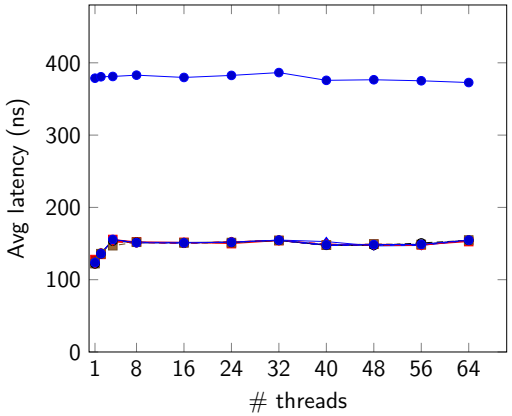
(b) DRAM (cross package)

(c) DRAM (package local)

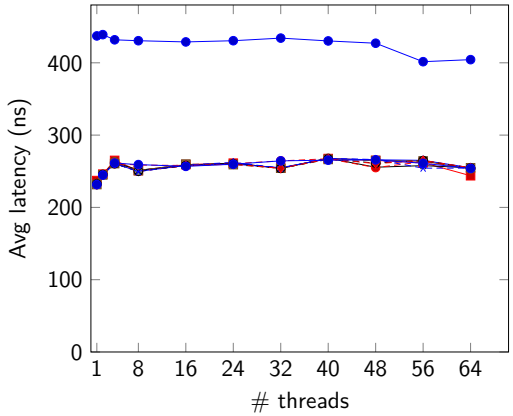
(d) DRAM (cross package)

Figure: STREAM benchmark

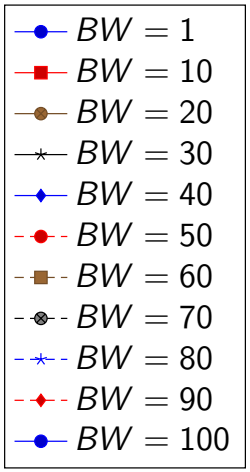
Platform evaluation: AMD

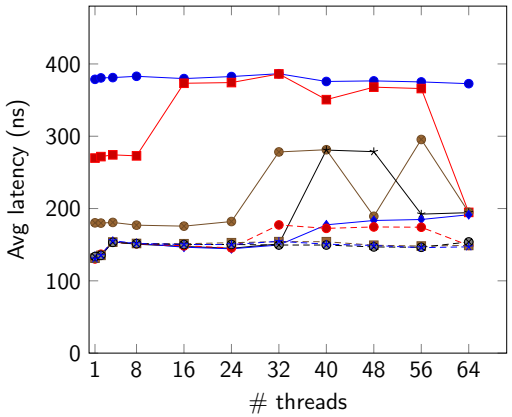


(a) DRAM (package local)

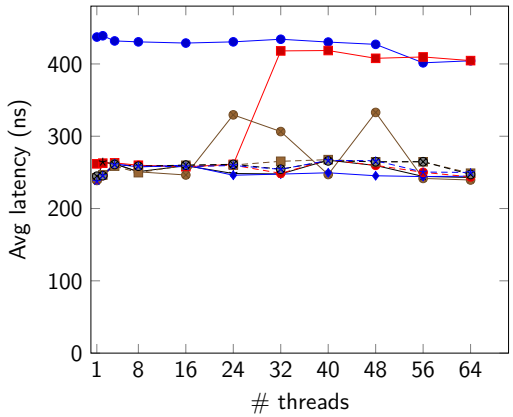


(b) DRAM (cross package)





(c) DRAM (package local)



(d) DRAM (cross package)

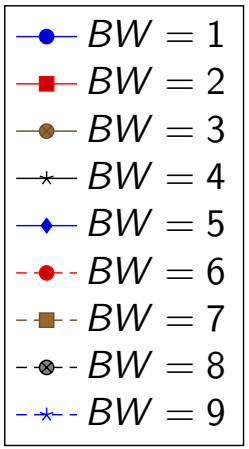


Figure: Latency depending on the number of threads and `resctrl`

Platform evaluation: AMD

- Small granularity when allocating the bandwidth
- No NUMA effect on bandwidth (with interleaved memory allocation)
- Wide range for the throttling settings
  - > Only range 1–100 is useful
  - > Fine grain setting of the bandwidth
- Strong effect of the bandwidth throttling on application (for values < 100)
- Bandwidth limitation may show effect on latency for too restrictive settings.

Inria

Platform evaluation: AMD

- Small granularity when allocating the bandwidth
- No NUMA effect on bandwidth (with interleaved memory allocation)
- Wide range for the throttling settings
  - > Only range 1–100 is useful
  - > Fine grain setting of the bandwidth
- Strong effect of the bandwidth throttling on application (for values < 100)
- Bandwidth limitation may show effect on latency for too restrictive settings.

Our experimental settings

- We tested on AMD, in the range 10–100
- We kept three values higher than 100 to ensure the coherency with our platform characterization.
- We used a step size of 10

Inria

03

Evaluating Bandwidth Sensibility

Inria

Set of Evaluated Applications

A wide panel of applications have been selected:

Pointer-Chasing Randomised accesses in an array.
FoM: *# elements accesed per second*

XSBench key computational kernel of the Monte Carlo neutron transport.
FoM: *lookups per second*

BT Tri-diagonal solver from the NASA parallel benchmark.
FoM: *FLOPS*

Lulesh LLNL Unstructured Lagrangian Explicit Shock Hydrodynamics application.
FoM: *elements solved per microsecond*

miniFE Finite element based proxy application.
FoM: *MFLOPS of the CG*

STREAM Reference benchmark for bandwidth applications.
FoM: *Maximum achieved bandwidth (in* MB/s*)*

Inria

A New Metric Defined

- Based on a progressive throttling of the bandwidth offered to the L3 cache
- Progressive decline of the FoM characterizes the sensibility to BW throttling
- We define three thresholds to evaluate quantitatively the sensibility
  - > 90 % of max FoM
  - > 75 % of max FoM
  - > 50 % of max FoM

Inria

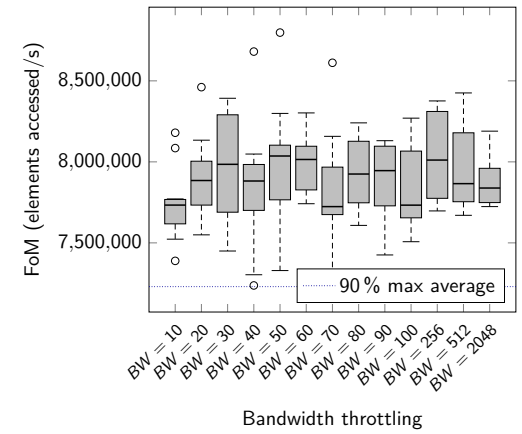
A New Metric Defined

- Based on a progressive throttling of the bandwidth offered to the L3 cache
- Progressive decline of the FoM characterizes the sensibility to BW throttling
- We define three thresholds to evaluate quantitatively the sensibility
  - > 90 % of max FoM
  - > 75 % of max FoM
  - > 50 % of max FoM

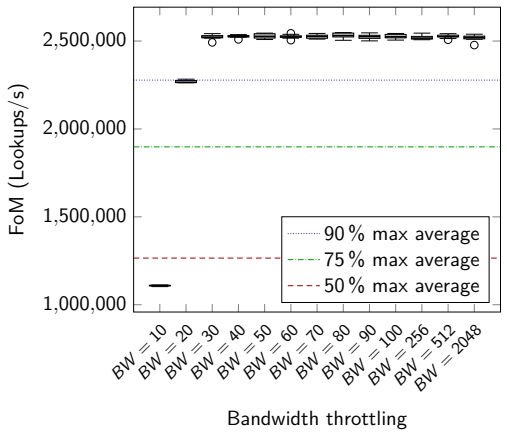
- AMD platform
- 10 runs per application, per throttling level
- Maximum average of FoM as baseline
- 8 threads (one per L3 cache)
- memory bound on all 4 NUMA nodes, with interleaving of pages

Inria

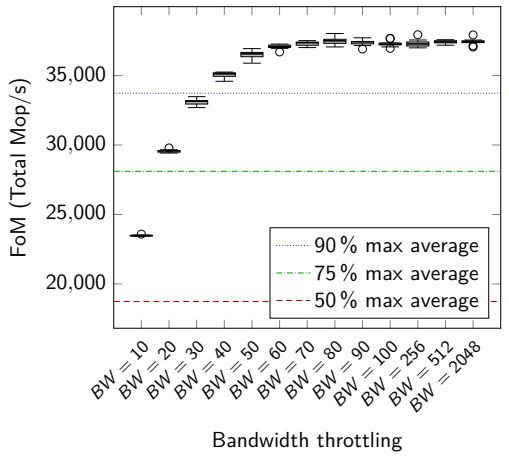
Results



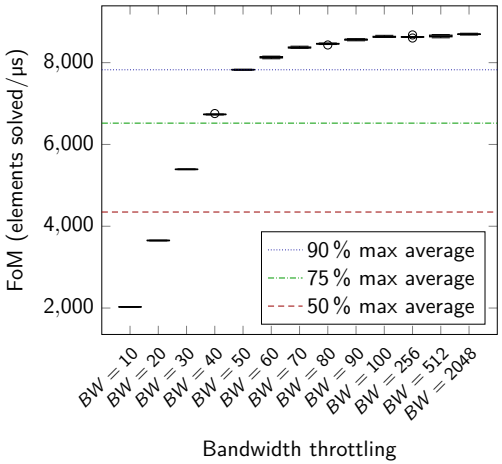
(a) Pointer-Chasing



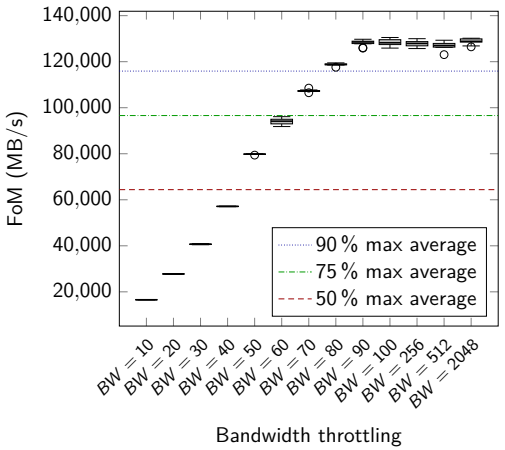
(b) XSBench



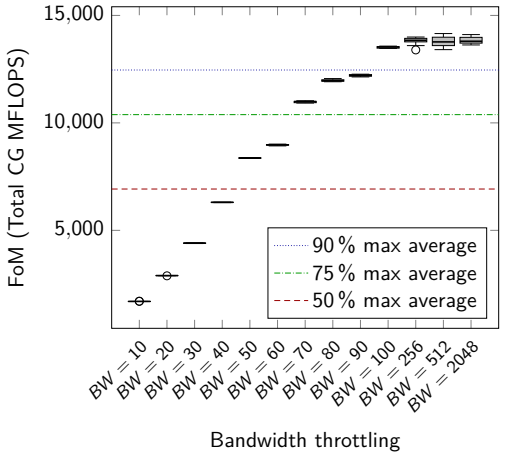
(c) NPB-BT



(d) Lulesh



(e) STREAM



(f) miniFE

Figure: Bandwidth sensitivity metric results

04

Conclusions

Inria

Conclusions

- x86 *Resource Control* is a viable way to alter the performance of a platform and generate extra heterogeneity
- Fine grained control is possible
  - > Add phases in the execution of the application instead of changing parameters for the whole execution
  - > Determining the sensibility of a specific buffer requires more control over the cache preloading
- The heterogeneity can be tuned to reflect different platform configurations

- Our metric shows promising results, and such quantitative approach may help sorting application and evaluation the expected benefit for a given technology
- Future work may investigate supporting other platforms (ARM support not yet available)

Inria

Thank you for your attention

Clément Foyer (clement.foyer@inria.fr)

This work was supported in part by the French National Research Agency (ANR) in the frame of the ANR-DFG H2M project (ANR-20-CE92-0022-01). Some experiments presented in this paper were carried out using the PlaFRIM experimental testbed, supported by Inria, CNRS (LaBRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d'Aquitaine (see `https://www.plafrim.fr/`).