The AMD EPYC Rome processor

Björn Dick (HLRS), Thomas Bönisch (HLRS)

Node - Overview

- 2 x AMD EPYC 7742 (Zen2 aka “Rome”), each:
  - 64 cores @ 2.25GHz, AVX2
  -

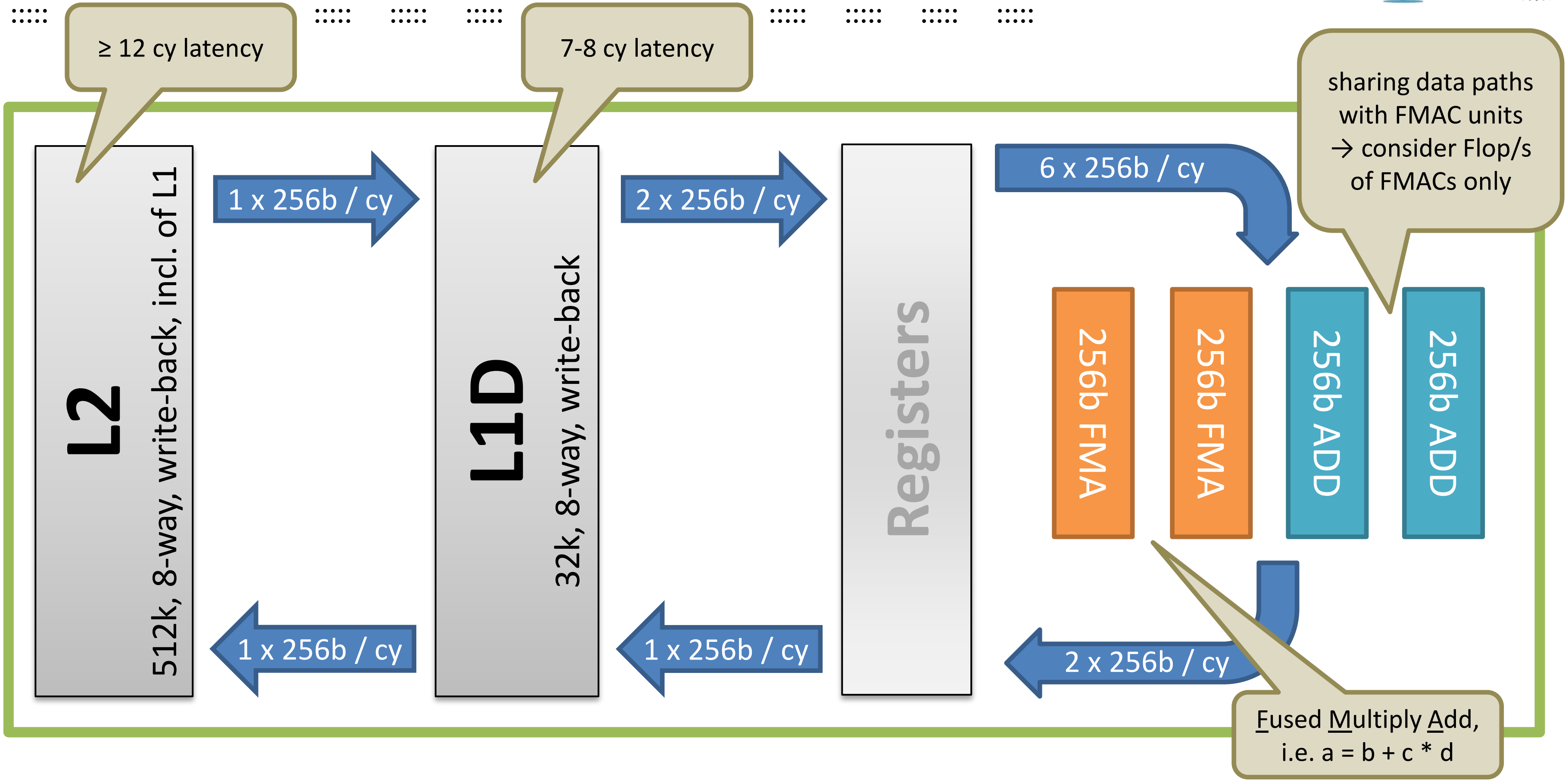
Cache	L1D	L1I	L2	L3
Size	32kB		512kB	16MB
Cache line size	64B			
Associativity	8-way			16-way
private/shared?	private			shared among 4 cores only!
Inclusion policy	inclusive			victim cache
Write policy	write-back			
Write-miss policy	write-allocate			*not applicable*

- DRAM:
  - 256GB @ 380GB/s

Hierarchical architecture (1)

- **EPYC Rome has a highly hierarchical architecture!**
- The basic building block of the processor is a core.
- A core can be used by up to 2 hyperthreads / hardware threads, hence sharing L1I, L1D and L2 caches.

Core

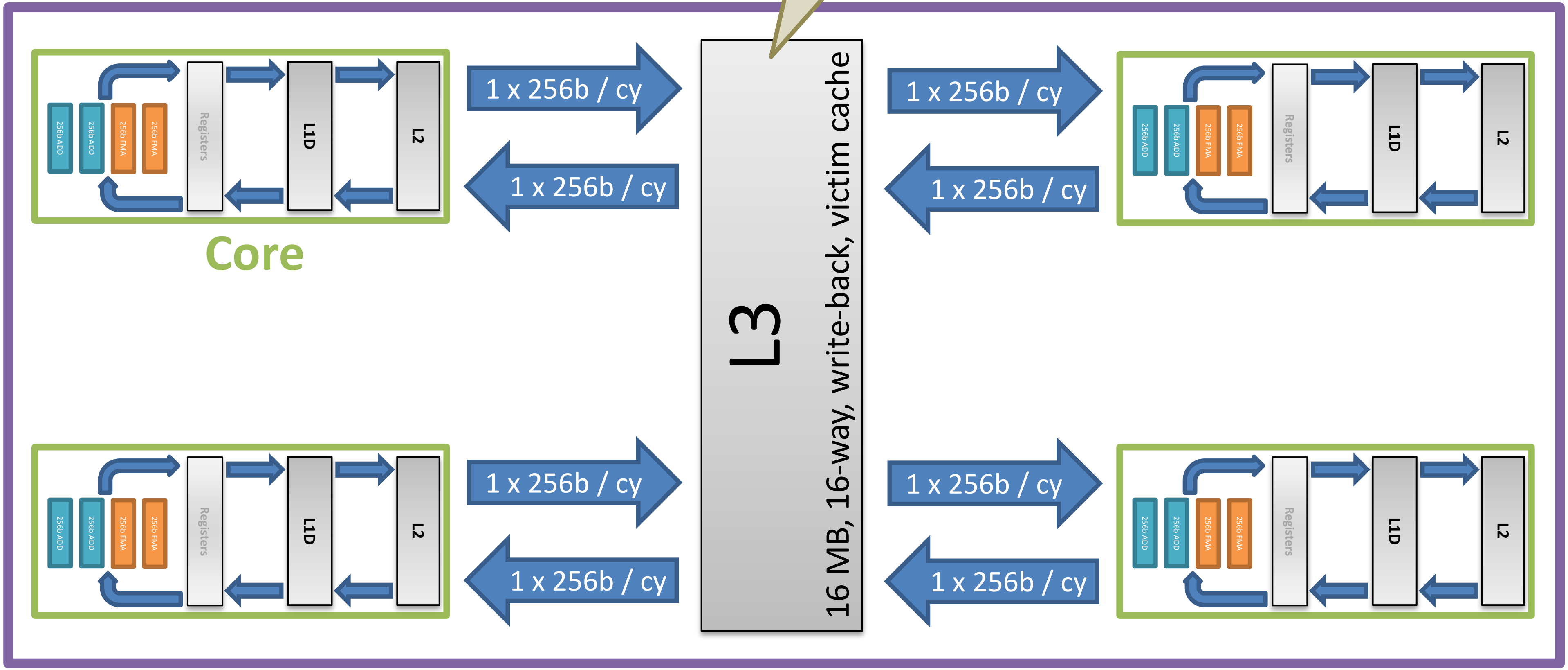

≥ 12 cy latency
7-8 cy latency
sharing data paths with FMAC units → consider Flop/s of FMACs only
L2
512k, 8-way, write-back, incl. of L1
1 x 256b / cy
L1D
32k, 8-way, write-back
2 x 256b / cy
Registers
6 x 256b / cy
256b FMA
256b FMA
256b ADD
256b ADD
1 x 256b / cy
1 x 256b / cy
2 x 256b / cy
Fused Multiply Add, i.e. a = b + c * d

Hierarchical Architecture (2)

- 4 cores are combined in a so called CCX (Core CompleX)
- together with a common L3 cache
- L3 can *not* be used by cores in other CCXs!

CCX

39 cy latency
on average



L3

16 MB, 16-way, write-back, victim cache

1 x 256b / cy

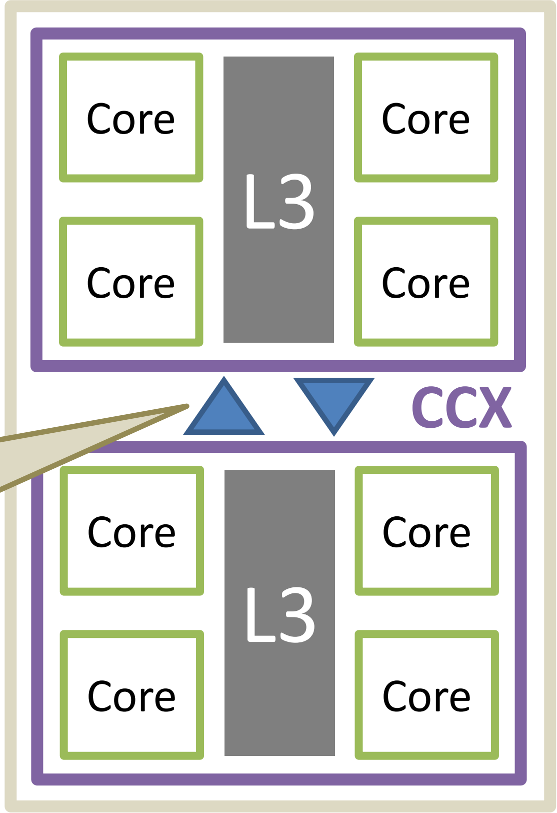
1 x 256b / cy

Core

Hierarchical Architecture (3)

- 2 CCXs are combined to a CCD (Cluster Complex Die)
- Sharing a common interface to the I/O die (cf. below)
- Each CCD is located on a separate silicon die
- → This hierarchy layer is relevant from a manufacturing point of view, but not that much from a user’s point of view.

CCD



CCXs within CCD *probably* connected with same type of link as CCD ↔ I/O die (2 x 51(?) GB/s)

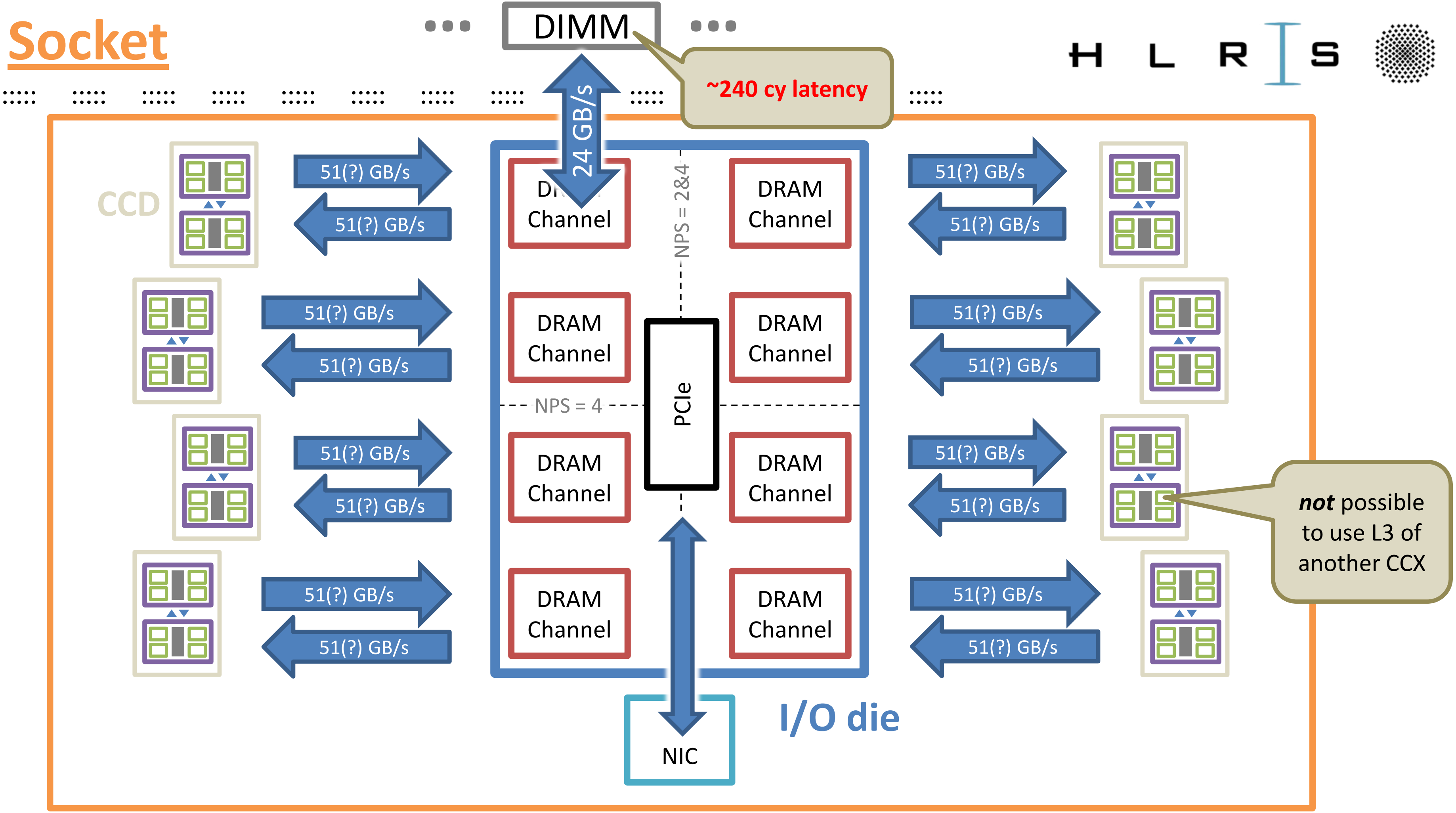
Hierarchical Architecture (4)

- 8 CCDs are attached to a common I/O die (holding memory controllers as well as PCIe) in order to form a socket.

- Memory channels as well as PCIe lanes can be split into 1, 2 or 4 NUMA nodes per socket (NPS)
  - pages distributed round robin among domains' DRAM channels
  - NPS=4 set by HLRS at system boot time → fixed!

- Every socket has a link to the NIC (aka "*Socket Direct®* ")

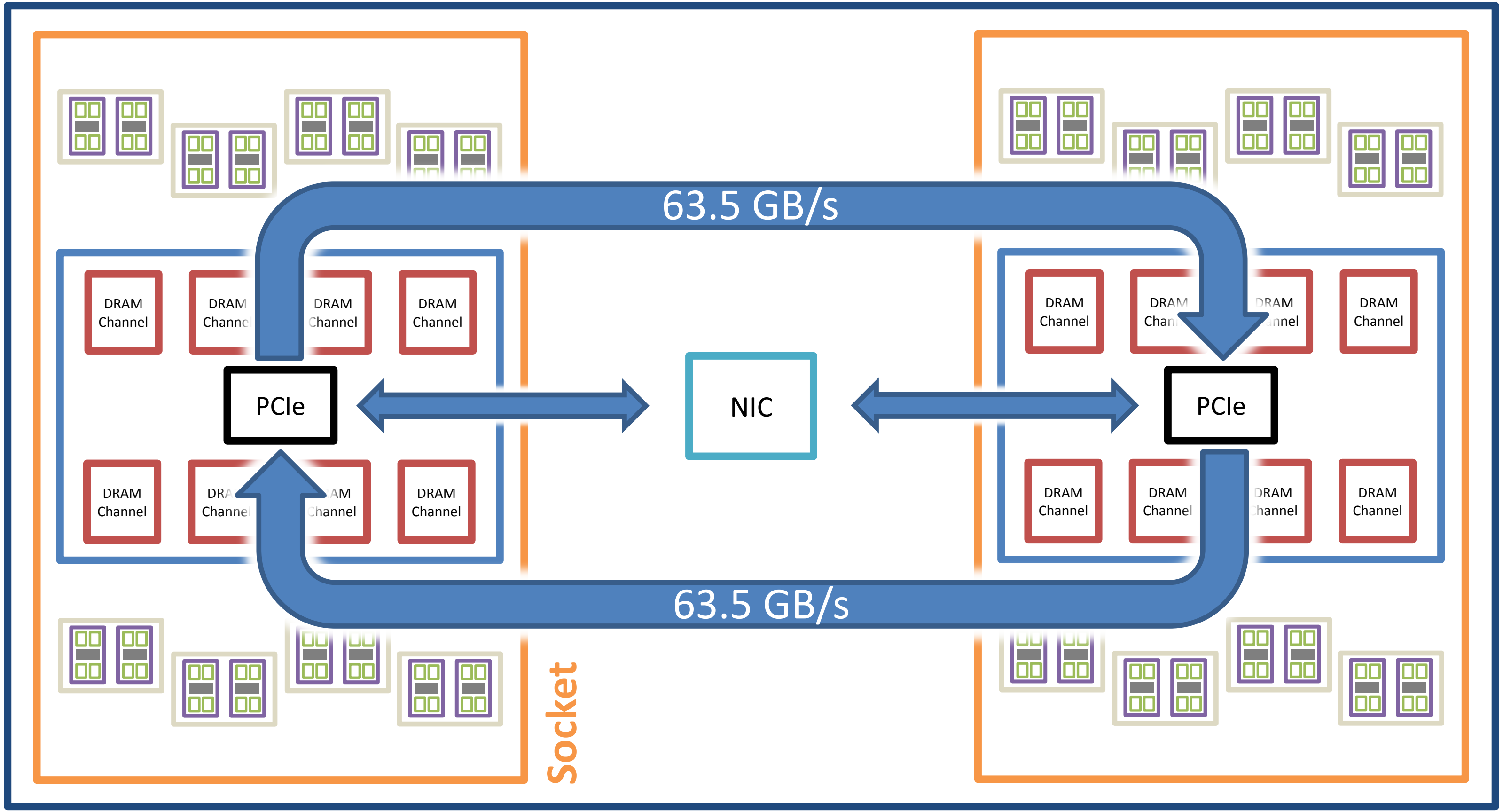
Socket


DIMM
24 GB/s
~240 cy latency
CCD
51(?) GB/s
DRAM Channel
NPS = 2&4
NPS = 4
PCIe
NIC
I/O die
not possible to use L3 of another CCX

Hierarchical Architecture (5)

- 2 sockets are combined to form a node.
- Every core in the node can access all memory DIMMs. However, cores have different distance to different DIMMs due to the hierarchical architecture!

Node


63.5 GB/s
DRAM Channel
DRAM Channel
DRAM Channel
DRAM Channel
DRAM Channel
DRAM Channel
DRAM Channel
DRAM Channel
PCIe
NIC
PCIe
DRAM Channel
DRAM Channel
DRAM Channel
DRAM Channel
DRAM Channel
DRAM Channel
DRAM Channel
DRAM Channel
63.5 GB/s
Socket

Hierarchical Architecture :: Summary

hierarchy layer	what	per	sharing	distance	data transfer	
					delay	B/W
1	2 Threads	Core	L1I, L1D and L2			
2	4 Cores	CCX	L3			
3	2 CCXs	CCD	link to I/O die			
4	2/4/8 CCDs	NUMA node	DRAM channels & PCIe lanes			
5	4/2/1 NUMA nodes	Socket	inter-socket link			
6	2 Sockets	Node	inter-node link			

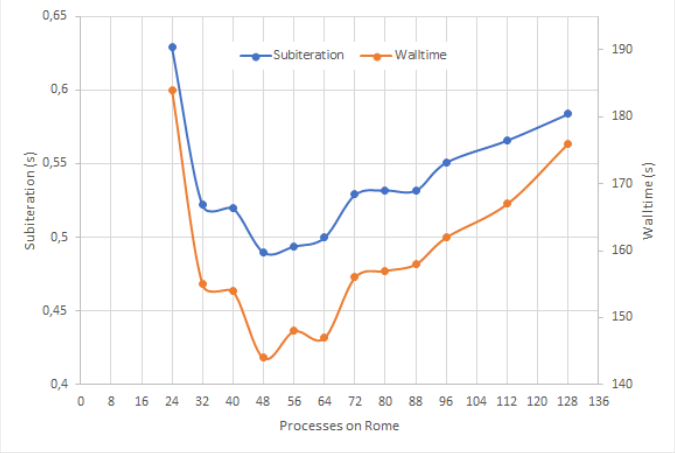
minor relevance

→ To achieve sufficient performance, partition problem & place groups of processes and threads in hierarchical manner so that distances of data transfers are minimized!

Remarks

- Rome core @ 2.25 GHz vs.
  Haswell core @ 2.5 GHz
- Haswell node: 2 x 70 GB/s
  Rome node: 2 x 190 GB/s
- → it’s ***not*** fair to compare
  128 Haswell cores
  (= 6 x 2 x 70 GB/s = 840 GB/s)
  to 128 Rome cores
  (= 2 x 190 GB/s = 380GB/s)
  if your code is bound by DRAM B/W
- → compare node vs. node instead

- It might be beneficial to use less than 128 cores per node!
  In particular if your code is bound by DRAM B/W:



Hardware Performance Counter

- Available via PAPI interface
- However, almost no *derived* PAPI metrics available
- Hence, use native events and do the math on your own:

Metric	PAPI event name	raw event	raw umask
IPC	RETIRED_INSTRUCTIONS	0xC0	0x00
	CYCLES_NOT_IN_HALT	0x76	0x00
DP Flop/s	RETIRED_SSE_AVX_FLOPS	0x03	0x0F
L1 misses / cy	*unfortunately not published by AMD*		
L2 misses / cy	CORE_TO_L2_CACHEABLE_REQUEST_ACCESS_STATUS:LS_RD_BLK_C	0x64	0x08
L3 misses / cy	*unfortunately not available (yet) due to security concerns*		
DRAM B/W	*Unfortunately not available (yet) due to security concerns*		

- cf. here (section 2.1.15.4) w.r.t. events/umasks description