# Towards Memory Disaggregation via NVLink C2C: Benchmarking CPU-Requested GPU Memory Access

Felix Werner, Marcel Weisgut, Tilmann Rabl

Hasso Plattner Institute, University of Potsdam, Germany





#### **Separation of Resources**

- Database systems have widely adapted separation of compute & storage
  - Elastic scaling of storage on demand
  - Avoid resource over-provisioning  $\rightarrow$  cost reduction



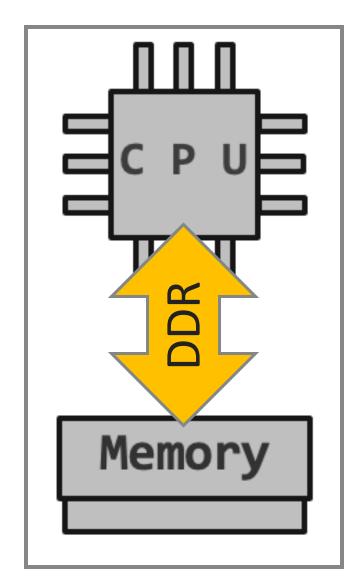




#### **Separation of Resources**

- Database systems have widely adapted separation of compute & storage
  - Elastic scaling of storage on demand
  - Avoid resource over-provisioning  $\rightarrow$  cost reduction
- Compute & memory tightly coupled
  - Result: stranded memory









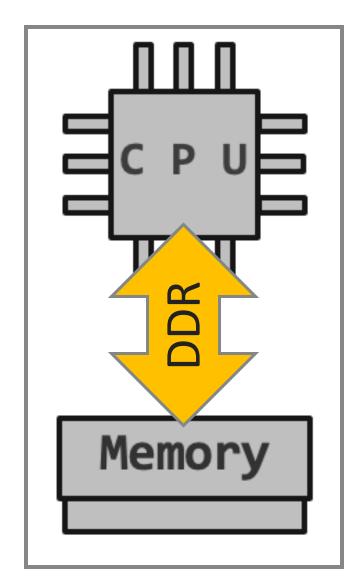
#### **Separation of Resources**

- Database systems have widely adapted separation of compute & storage
  - Elastic scaling of storage on demand
  - Avoid resource over-provisioning  $\rightarrow$  cost reduction
- Compute & memory tightly coupled
  - Result: stranded memory

#### **Memory Disaggregation**

- Initiatives towards memory disaggregation to separate memory from compute resources
  - Compute Express Link (CXL) as a recent option







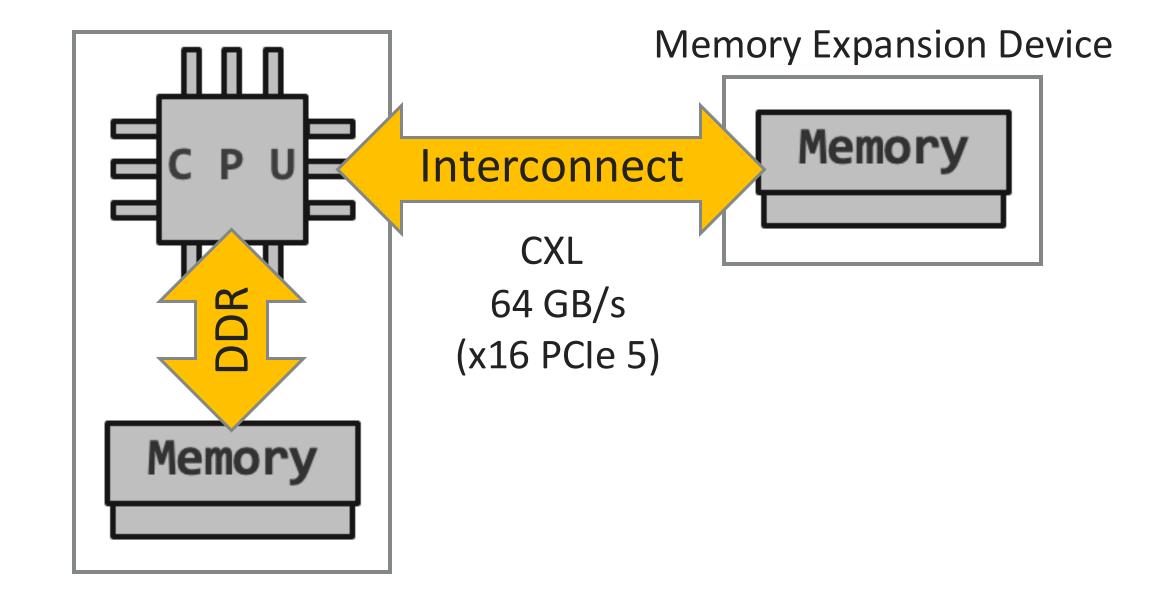
#### **Separation of Resources**

- Database systems have widely adapted separation of compute & storage
  - Elastic scaling of storage on demand
  - Avoid resource over-provisioning  $\rightarrow$  cost reduction
- Compute & memory tightly coupled
  - Result: stranded memory

#### **Memory Disaggregation**

- Initiatives towards memory disaggregation to separate memory from compute resources
  - Compute Express Link (CXL) as a recent option







NAAtivatian								
Motivation	Versio							
Separation of Resources	1							
Database systems have widely adapted separation of compute & storage	2 3 4 / C20							
Elastic scaling of storage on demand								
Avoid resource over-provisioning $\rightarrow$ cost reduction								
Compute & memory tightly coupled								

Result: stranded memory

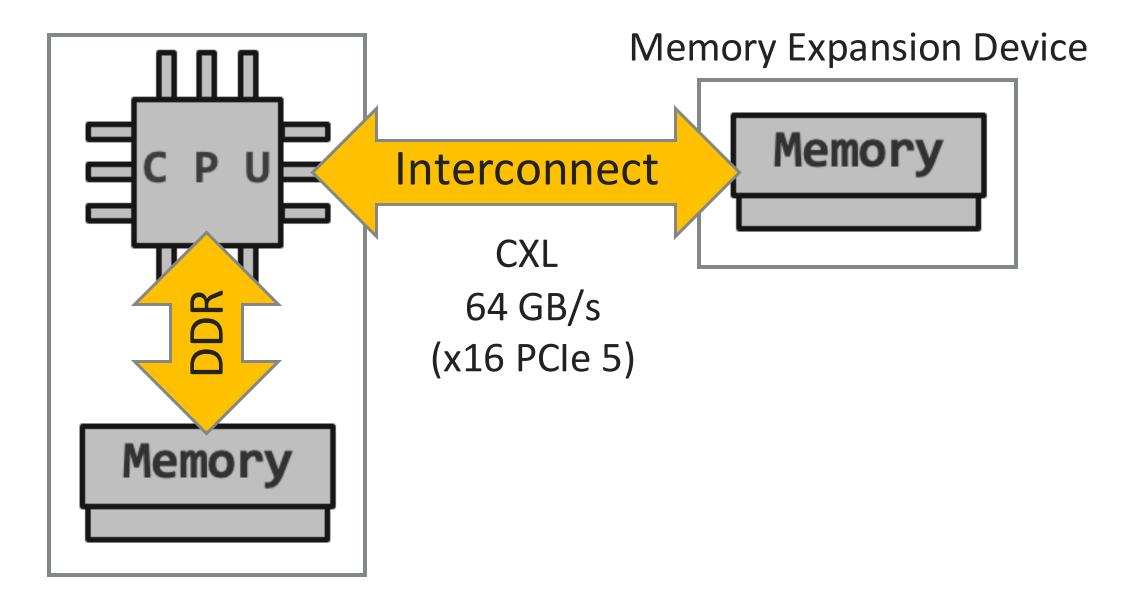
#### **Memory Disaggregation**

- Initiatives towards memory disaggregation to separate memory from compute resources
  - Compute Express Link (CXL) as a recent option



#### nk versions

sion	Data rate per link [GB/s]	# Lanes per link	# Links	Theoretical bandwidth	Architecture
1	20	8	4	80	Pascal
2	25	8	6	150	Volta
3	25	4	12	300	Ampere
C2C	25	4	18	450	Hopper







NAAtivatian								
Motivation	Versio							
Separation of Resources	1							
Database systems have widely adapted separation of compute & storage	2 3 4 / C20							
Elastic scaling of storage on demand								
Avoid resource over-provisioning $\rightarrow$ cost reduction								
Compute & memory tightly coupled								

Result: stranded memory

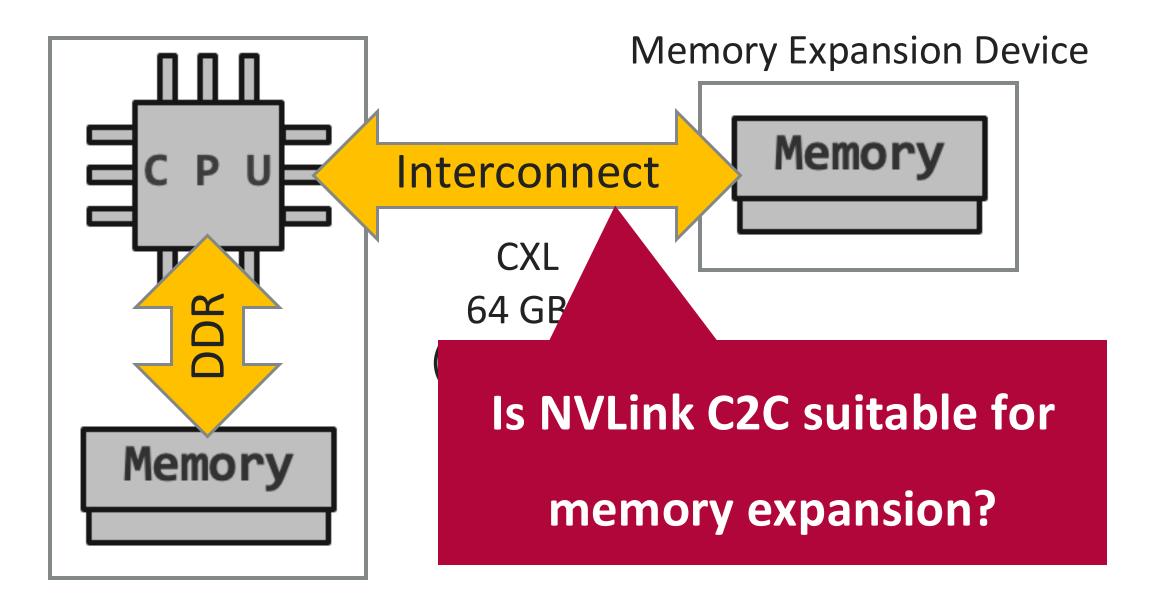
#### **Memory Disaggregation**

- Initiatives towards memory disaggregation to separate memory from compute resources
  - Compute Express Link (CXL) as a recent option



#### nk versions

sion	Data rate per link [GB/s]	# Lanes per link	# Links	Theoretical bandwidth	Architecture
1	20	8	4	80	Pascal
2	25	8	6	150	Volta
3	25	4	12	300	Ampere
C2C	25	4	18	450	Hopper



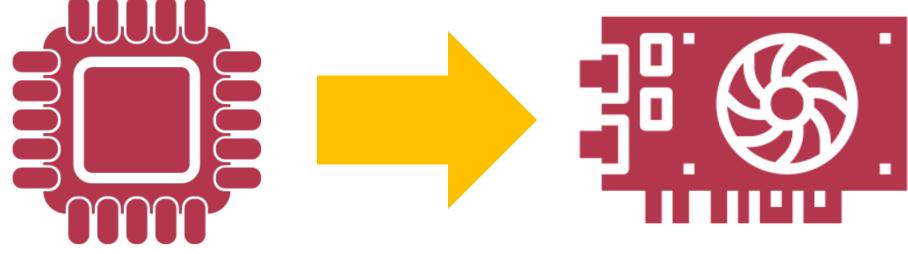


## Contributions

- Performance evaluation of memory access performance over NVLink C2C
  - $\blacktriangleright$  CPU $\rightarrow$ GPU memory: throughput, latency
  - $\blacktriangleright$  CPU $\rightarrow$ CPU & GPU memory: throughput expansion
- Impact of storing data in GPU memory on database operations
- Discuss suitability of NVLink-attached memory for memory expansion

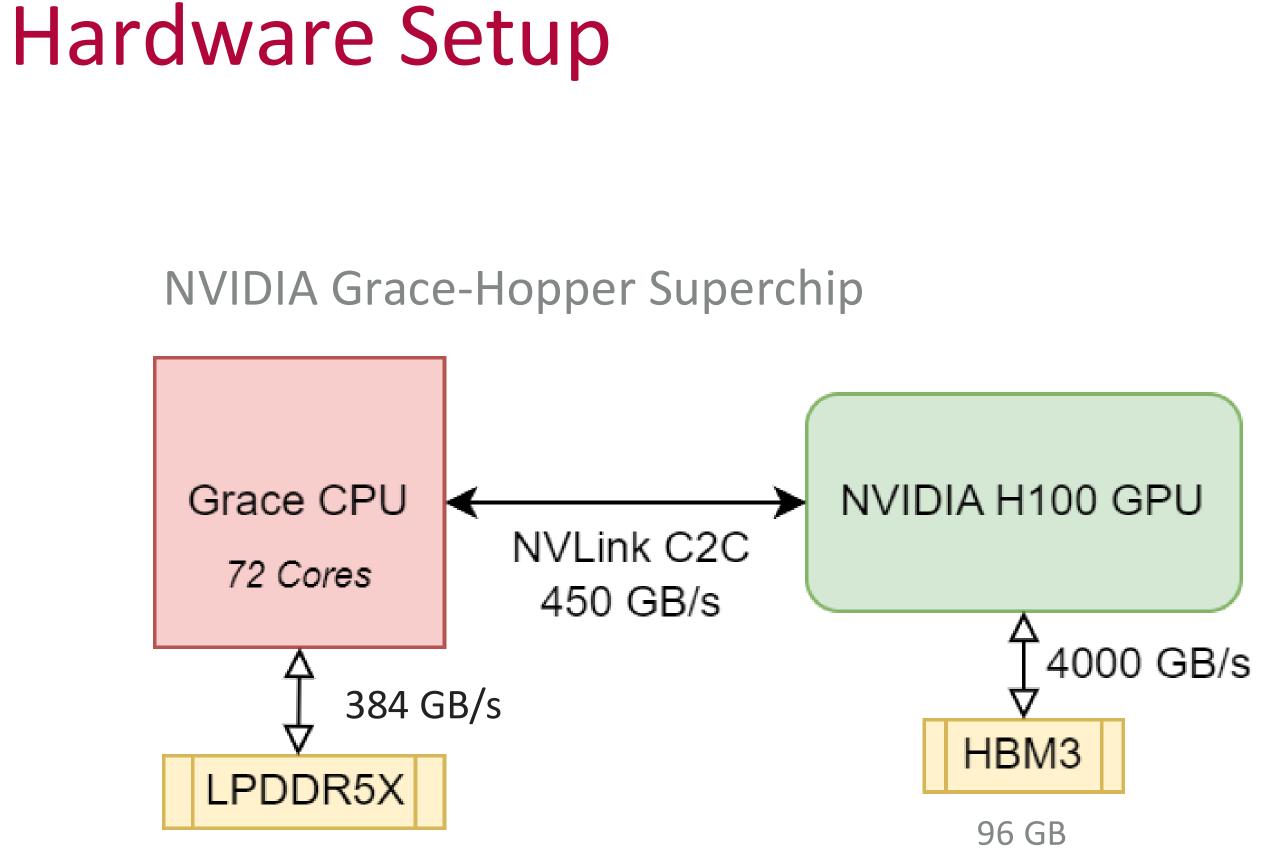












<sup>480</sup> GB







### Bandwidth & Latency

#### CPU→CPU

#### Sustained Bandwidth [GB/s]



Seq Read 362

Rnd Read 345



U Memory	CPU→GPU	J Memory	
Seq Write 370	Seq Read 130 36% CPU→CPU	Seq Write 163	
Rnd Write 370	Rnd Read 128	Rnd Write 168 45% CPU→CPU	



## Bandwidth & Latency

#### CPU→CPU

### Sustained Bandwidth [GB/s]

Seq Read 362

Rnd Read 345

### Latency [ns] (random reads)



Idle 22

Load

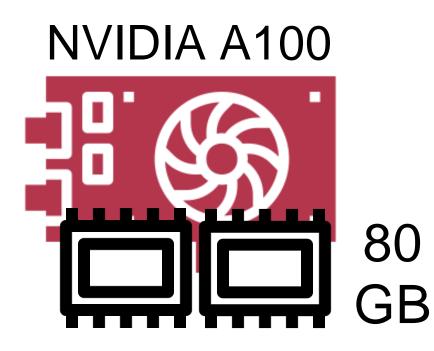
44

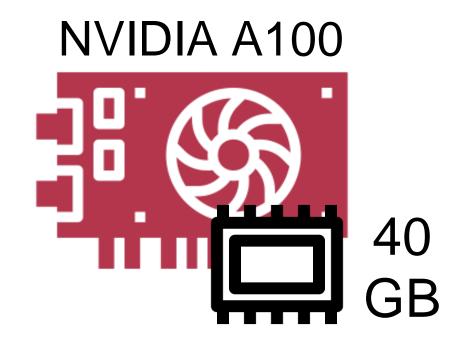


U Memory CPU→GPU Mem	ory
Seq Write Seq Read Seq V   370 130 16   36% CPU→CPU 10	
Rnd Write Rnd Read Rnd V   370 128 16   459 459 128	<b>8</b> %
le ldle 20 810 (3.7x) ded Loaded 1020 (2.3x)	



- HBM is on-package memory  $\rightarrow$  not modular like DRAM DIMMs
- Price estimation attempt: compare GPU models with different memory capacities





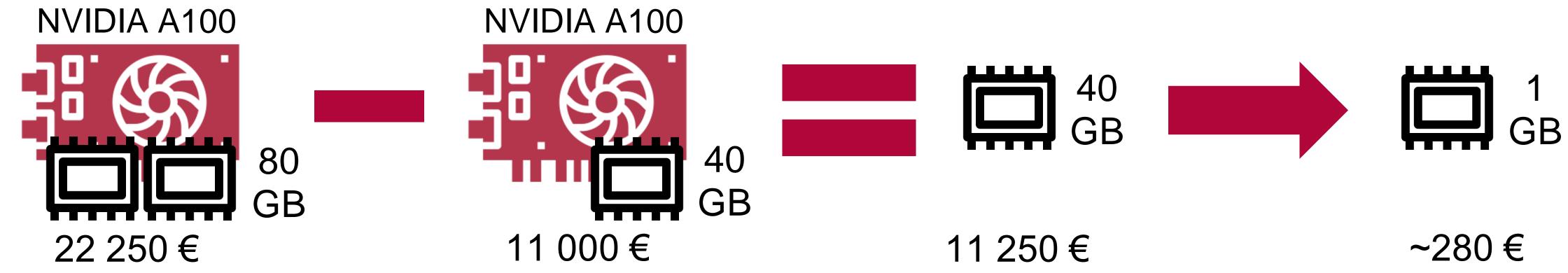
Prices retrieved from <u>geizhals.eu</u> in February 2025.



RAM DIMMs ith different memory capacities



- HBM is on-package memory  $\rightarrow$  not modular like DRAM DIMMs
- Price estimation attempt: compare GPU models with different memory capacities

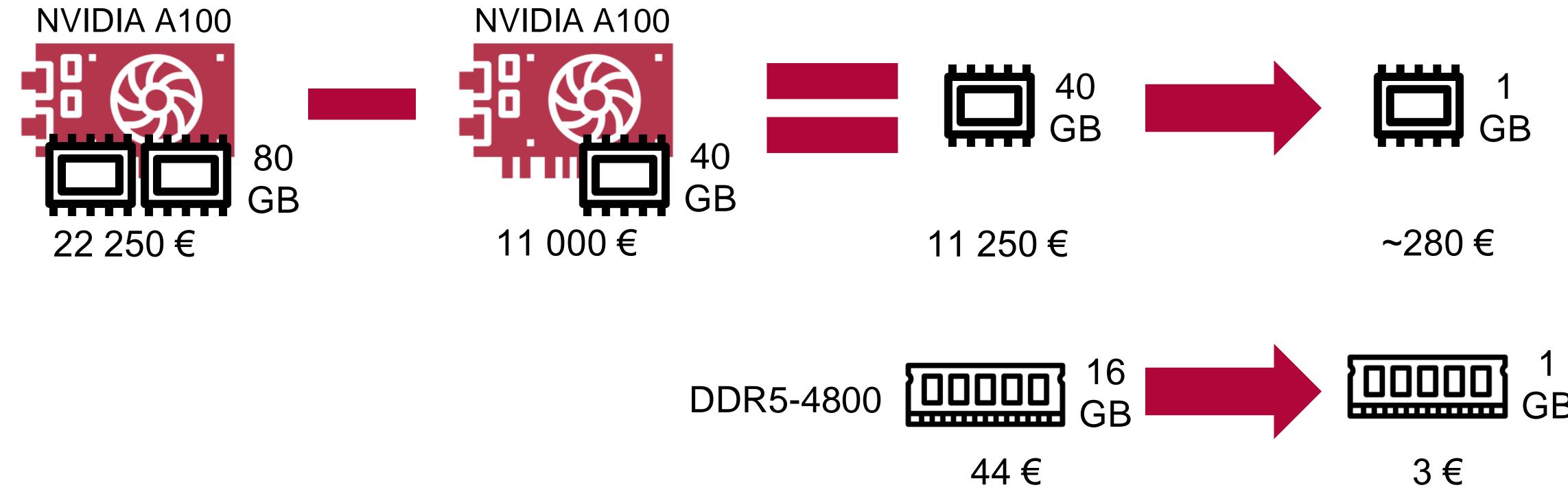


Prices retrieved from geizhals.eu in February 2025.





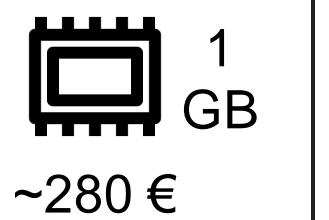
- HBM is on-package memory  $\rightarrow$  not modular like DRAM DIMMs
- Price estimation attempt: compare GPU models with different memory capacities



Prices retrieved from <u>geizhals.eu</u> in February 2025.









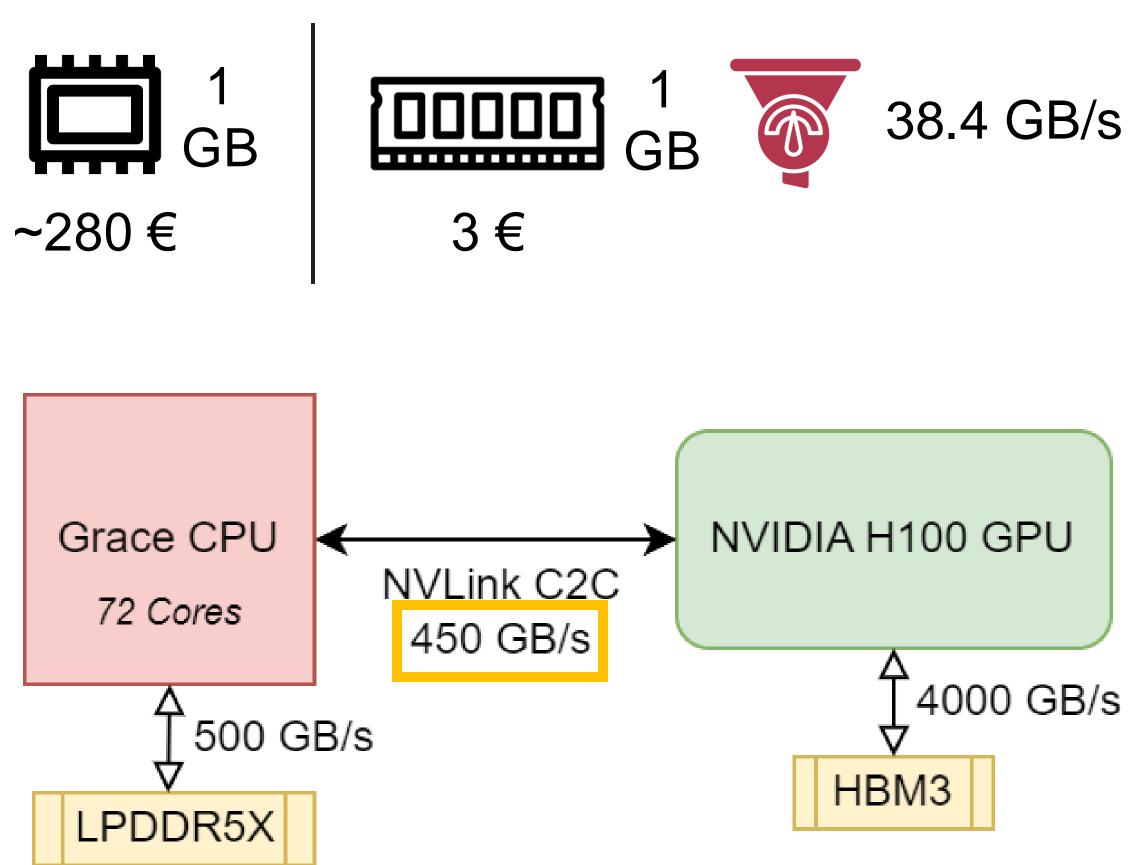
3€

Prices retrieved from <u>geizhals.eu</u> in February 2025.







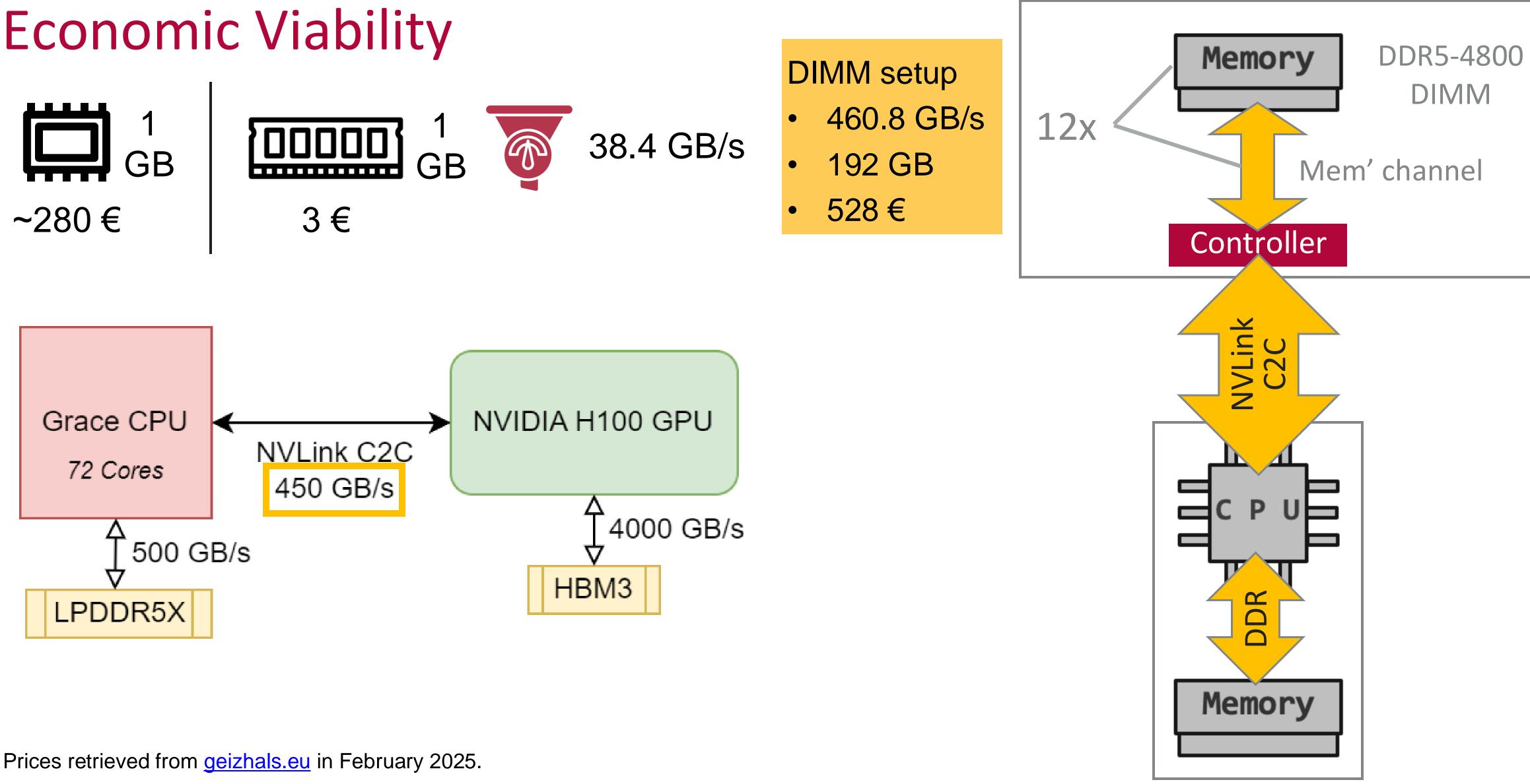


Prices retrieved from <u>geizhals.eu</u> in February 2025.











#### Memory Expansion Device





## Summary

- ► CPU $\rightarrow$ GPU achieves 36% and 45% of the CPU $\rightarrow$ CPU read/write throughput
- ▶ CPU $\rightarrow$ GPU has access latencies of between ~800 ns to ~1000 ns (3.7x / 2.3x of CPU $\rightarrow$ CPU)
- Bandwidth expansion study shows improvements between of 1.3x for reads and 1.7x for writes
- Multiple DDR DIMMs more cost-efficient for pure memory expansion than GPU HBM





## Data Engineering Systems Group @ HPI

A hpi.de/rabl



















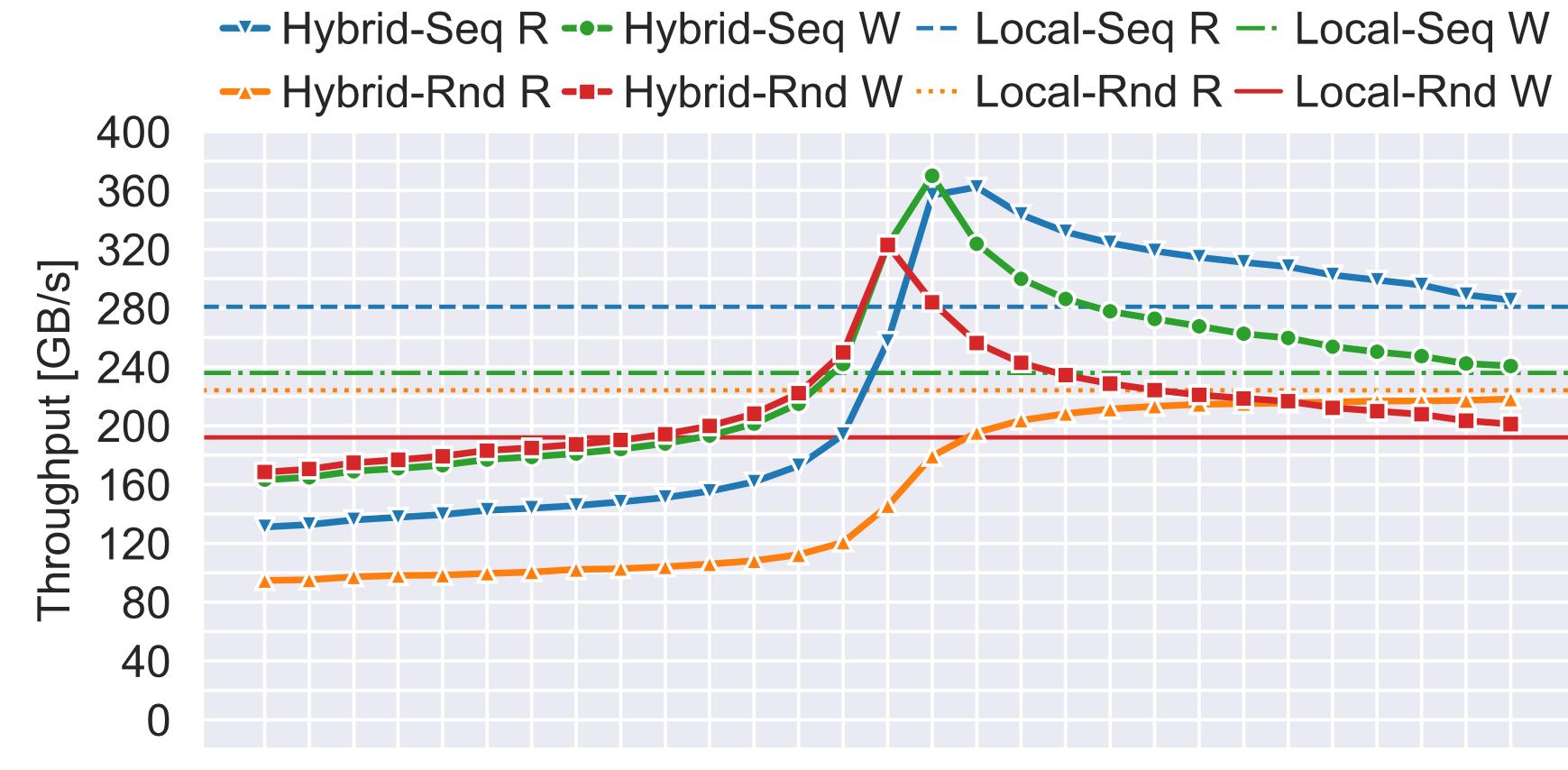








## **Bandwidth Expansion**



 $\begin{array}{c} & & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & & \\ & & \\ & & & \\ & & \\ & & & \\ & & & \\ & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\$ 040----040 $\overline{\phantom{a}}$ Interleaving Ratio [pages in CPU memory:pages in GPU memory]

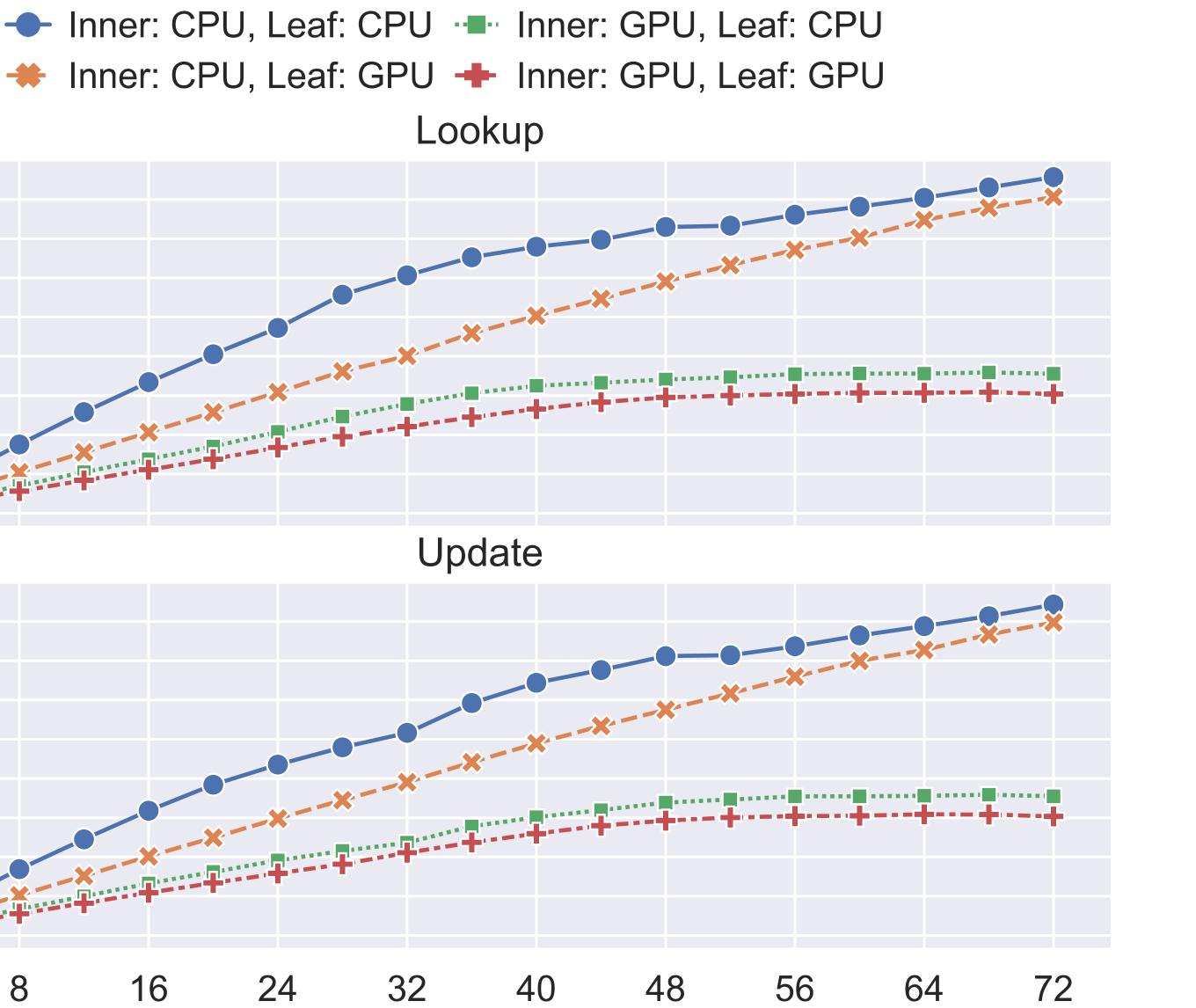


Rnd writes: 1.7x Seq writes: 1.6x Seq reads: 1.3x

Access size: 64 B Region size: 90 GiB

Reads: Id1 Writes: vst1q\_u8\_x4









# Threads



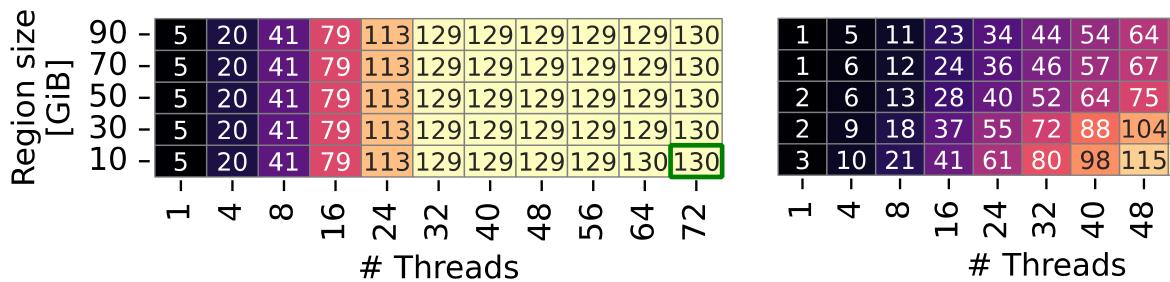
## **CXL-Bench**

- Support for fundamental and chains of memory access operations
- Memory region preparation
  - Pin memory region to NUMA node(s) (mbind) such as CXL and (modern) GPU memory
  - Multiple regions, partitions, and NUMA nodes supported
  - Round-robin & weighted page interleaving supported
- Task execution
  - Thread pinning to set of cores (pthread setaffinity np)
  - Memory access via scalar & SIMD instructions (using compiler vector intrinsics)
- Configurable with YAML files











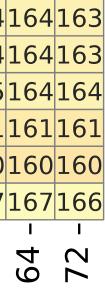
Rnd Writes

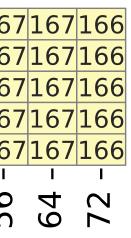
#### Seq Writes

122	122	9	42	74	137	163	163	162	162	162	162	162		9	37	74	138	163	164	164	164	164	1
101	101	9	37	74	137	163	163	162	162	162	162	162		9	37	74	138	163	164	164	164	164	1
80	80	9	37	74	137	163	163	162	162	162	162	161		9	37	73	138	165	165	165	165	165	1
105	109	9	37	73	137	162	162	162	162	162	161	161		9	37	73	136	160	162	162	162	161	1
94	94	9	37	73	136	162	162	162	161	161	161	161		9	36	71	134	159	161	161	161	160	1
127	128	9	35	70	132	162	162	162	161	161	161	161		7	29	57	114	158	168	168	168	167	1
Î.	I	Ι	I	I	I	Ī	I	Ī	I	I	Ī	Ī		I	I	I	I	I	I	I	Ι	I	
64	72	1	4	$\infty$	16	24	32	40	48	56	64	72		Н	4	$\infty$	16	24	32	40	48	56	1
		# Threads															-	# T	hre	ads	5		

				_																					
7	'4	83	92		9	35	70	132	162	162	162	162	161	161	161		6	22	42	84	128	164	167	167	167
7	7	87	96		9	35	70	132	162	162	162	162	161	161	161		6	22	43	85	134	165	168	167	167
8	37	97	107	1	9	35	70	132	162	162	162	162	161	161	161		6	22	43	91	139	166	168	167	167
11	16	119	118		9	35	70	132	162	162	162	161	161	161	161		6	23	47	100	145	167	168	167	167
512	25	127	128		9	35	70	132	162	162	162	161	161	161	161		7	28	56	110	155	168	168	167	167
	I	Ι	I		Ι	I	I	I	I	I	I	I	I	I	Ι	_	I	I	I	I	I	I	I	I	I
U U	0	64	72		Ч	4	$\infty$	16	24	32	40	48	56	64	72		-	4	$\infty$	16	24	32	40	48	56
					# Threads															-	# T	hre	ads	5	









## BenchmarDing CRU-Requested GAU Menory Actes DiRcer Waisgut ESS THROUGHPUT

Sequential Reads																					
6553	36 - 20	80	152	212	261	300									359	358	357	357	357	-	350
3276	58 - 21	82								362											300 4
ં 1638	34 - 21	83	157	218	266	302	328	346	356	362	361	360	360	359	359	358	358	358	358		500 3
(Byte) 818 819	92 - 21	83	158	219	266	303	328	345	356	361	361	360	360	359	359	358	358	358	358	-	250
405	96 - 21	83	157	218	266	301	327	344	354	360	360	360	359	358	358	358	357	357	357		
204 204 204	8 - 21	83	156	218	266	302	328	345	355	360	360	359	359	358	358	357	357	357	357		200
101	24 - 21	83	157	218	267	301	326	344	354	358	358	358	357	357	356	356	356	355	356		150 र
402 Access	2 - 21	81	154	214	263	299	324	341	351	355	355	354	354	353	353	352	352	351	351		
Ŭ 25	6 - 20	80	152	212	260	296	321	336	345	349	349	349	348	347	347	347	346	346	346	ŀ	<mark>ہ</mark> 100 ک
12	28 - 20	78	148	206	254	289	312	327	335	339	338	338	337	337	336	336	335	335	335		F0 F
e	54 - 19	74	139	194	238	272	295	309	316	319	319	318	318	317	316	316	316	316	316		50
	i	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72		
	1	т	U	12	10	20	24			ad C			10	52	50	00	04	00	, 2		

Ranc	lom	Reads

	65536 - 19	75	143	200	242	070	204	221	221	226	337	220	240	242	244	244	245	245	244		
																					200
	32768 - 18	70	134	187	228	259	281	297	306	311	314	318	321	324	327	330	332	334	335		- 300 س
te	16384 - 16	62	118	165	202	230	249	262	269	273	278	284	291	296	302	307	312	316	320		- 250 🗑
(Byte)	8192 - 13	50	97	136	167	190	205	215	221	224	233	243	252	261	269	277	284	290	296		
) 0	4096 - 10	39	75	105	128	144	154	162	166	179	196	213	225	234	242	247	253	259	265		- 200 -
size	2048 - 8	30	59	85	108	125	138	147	153	161	173	182	190	199	209	219	226	231	236		put
	1024 - 6	25	50	75	99	122	144	164	183	200	216	230	241	252	260	267	273	277	282		- 150 ਦੂ
Access	512 - 6	25	50	74	98	120	142	161	179	195	209	221	231	239	245	250	253	255	257		100 Bho
ΔC	256 - 8	31	63	92	121	147	170	190	205	216	225	230	235	237	239	240	241	242	244		- 100 은
	128 – 9	37	75	109	142	172	199	222	239	252	260	266	270	272	274	275	275	275	275		- 50 ⊢
	64 - 8	33	66	92	119	139	164	184	203	220	235	248	259	267	276	282	288	293	297		
	1	I	I	I	I	I	1	I.	I	I	I	I.	I	-	-	I	I.	I	-		•
1 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72																					
								Т	hrea	ad (	Cour	nt									



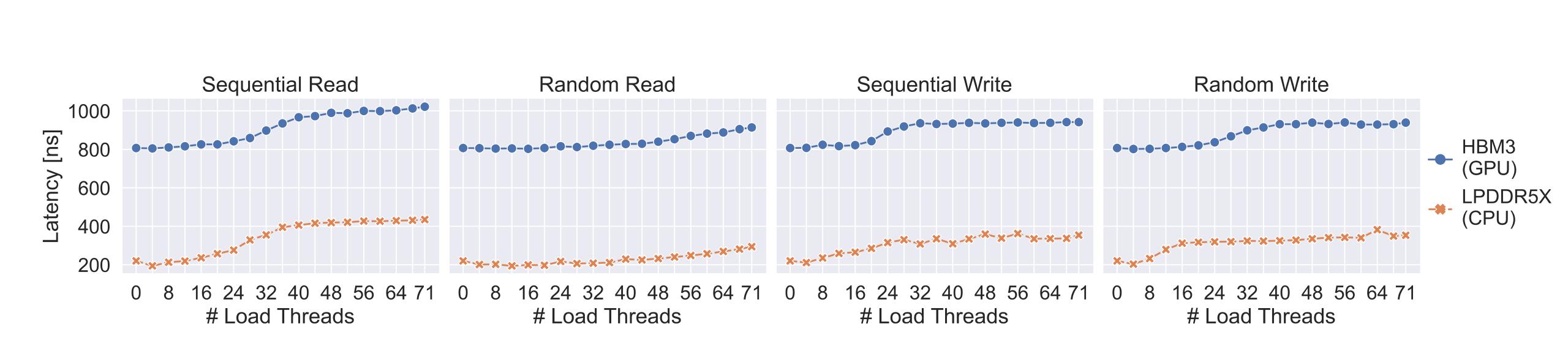


	65536 -	44	171	317	370	367	365	365	364	364	364	364	364	363	363	363	363	363	363	363	- 350	)
	32768 -	44	171	317	370	367	365	365	364	364	364	364	363	363	363	363	363	363	363	363		Ś
te)	16384 -	43	171	316	370	367	365	364	364	364	364	364	363	363	363	363	363	363	363	363	- 300	B
(Byt	8192 -	43	171	316	369	367	365	364	364	364	364	363	363	363	363	363	363	363	363	362	- 250	0
	4096 -	43	170	315	366	365	363	363	363	363	363	363	363	363	362	362	362	362	362	362		
size	2048 -	43	169	313	357	357	356	356	355	355	355	355	355	355	356	356	356	356	357	357	- 200	bu
	1024 -	43	167	310	342	342	341	341	340	339	339	338	338	338	338	337	337	337	337	337		_
Access	512 -	42	165	316	337	330	324	321	319	318	317	316	316	315	315	314	313	313	313	312	- 150	no
Ă	256 -	41	160	296	327	317	309	303	299	296	294	292	290	289	288	287	286	285	284	283	- 100	ام ا
	128 -	39	151	280	310	297	287	281	275	271	267	264	261	257	256	253	252	252	250	249	100	
	64 -	35	139	253	291	280	272	267	262	258	254	251	248	245	243	239	238	236	234	232	- 50	
		i	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72		
									Т	hrea	ad C	Cour	nt									



	65536 - 43	171	327	369	367	365	365	364	364	364	364	364	363	363	363	363	363	363	363	
	32768 - 43	171	327	369	367	366	366	365	365	364	364	364	364	364	363	363	363	363	363	200 10
te)	16384 - 43	170	325	370	370	368	368	367	366	366	366	366	365	365	365	365	364	364	364	- 300 s/ GB/s
(Byt	8192 - 43	169	322	344	348	350	352	353	353	353	353	354	354	354	354	354	355	355	355	0
	4096 - 43	168	314	338	343	346	347	348	349	349	349	350	350	350	350	350	350	350	350	rt.
size	2048 - 42	165	307	330	336	338	340	340	341	341	341	342	342	342	342	342	342	342	342	- 200 ja
	1024 - 41	159	295	320	325	326	327	327	327	327	327	327	327	327	327	327	327	328	328	dybn
Access	512 - 39	154	288	315	313	311	310	309	308	307	307	307	306	306	306	305	305	305	305	no
AC	256 - 38	141	269	319	312	305	300	300	295	289	288	287	283	284	281	281	277	276	276	لم 100 -
	128 - 30	113	183	290	282	272	267	266	257	247	243	250	238	239	243	232	231	229	232	
	64 - 16	71	140	208	217	215	212	210	206	201	200	201	197	195	197	194	190	190	188	
	i	4	8	'12	16	20	24	28	3 <sup>'</sup> 2	36	40	4 <sup>'</sup> 4	4 <sup>'</sup> 8	52	56	60	64	68	, 72	-
Thread Count																				

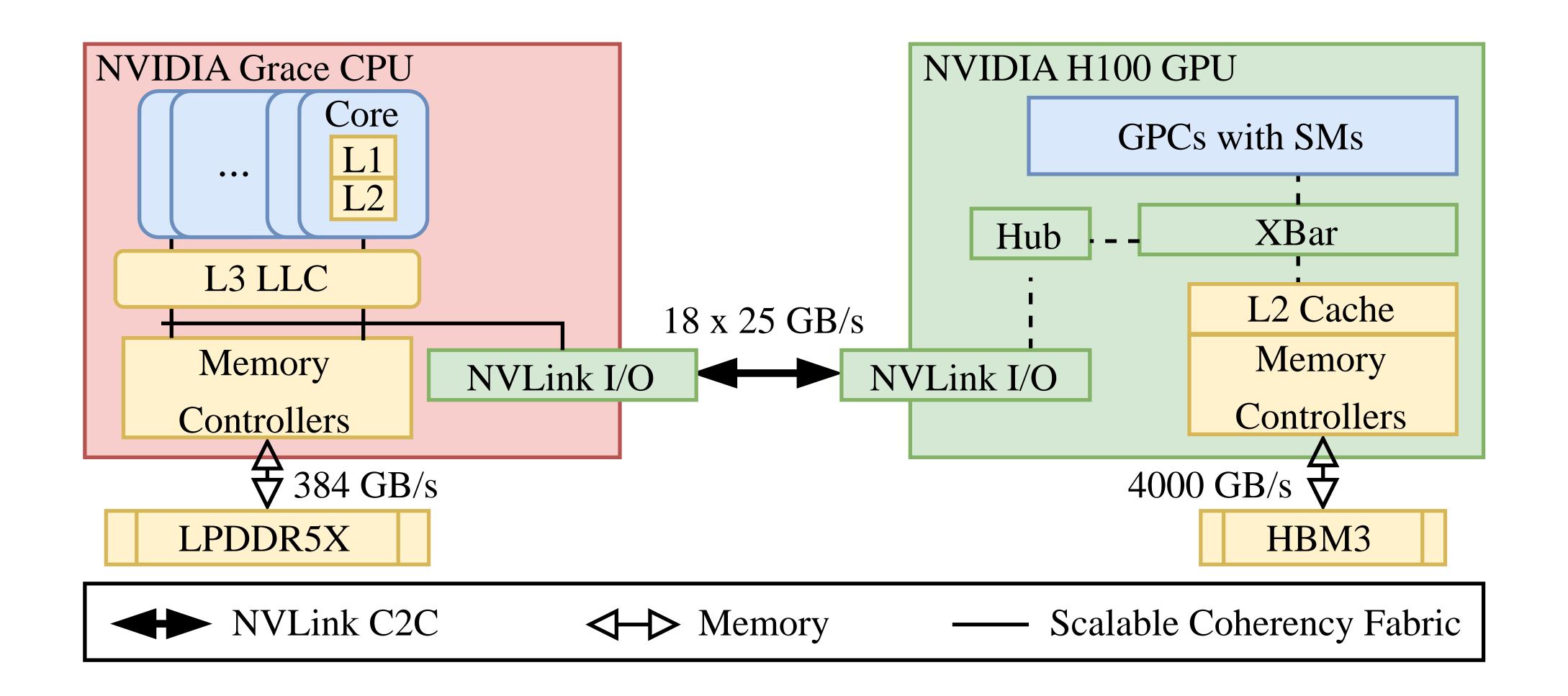
Throughput in GB/s















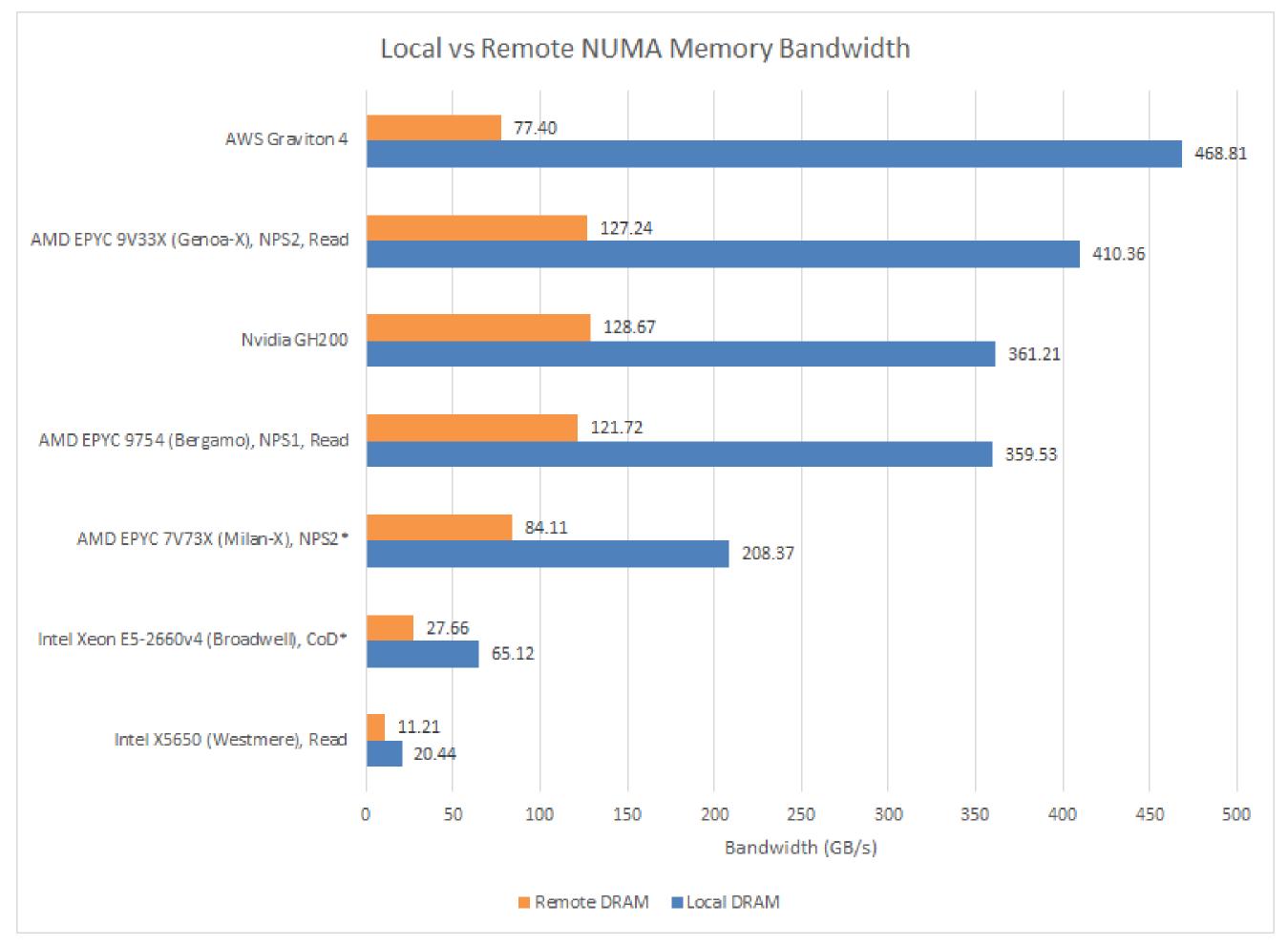
## **NVLink Evolution**

Version	Data rate per link [GB/s]	# Lanes per link	# Links	Theoretical bandwidth	Architecture
1	20	8	4	80	Pascal
2	25	8	6	150	Volta
3	25	4	12	300	Ampere
4 / C2C	25	4	18	<b>450</b>	Hopper





## **NVLINK C2C VS CPU INTERCONNECTS**



https://chipsandcheese.com/2024/07/31/grace-hopper-nvidias-halfway-apu/