PSD@GIA: Collection of Exercises (some solutions)

Eduard Ayguadé and Josep Lluis Berral
Departament d'Arquitectura de Computadors

Course 2023-24 (Fall semester)

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Unit 3

1. The energy efficiency metric is defined as the ratio between the peak FLOP/s and the power required to achieve them (the TDP). **We ask you to** compute the energy efficiency for an Intel socket Skylake Platinum 8360Y with 36 cores running at 2.4 GHz when the 2 AVX-512 units are used to perform double precision computation; the TDP for the socket is 250 Watts. Is the energy efficiency better for its companion Skylake Gold 6354 with half the number of cores running at 3.0 GHz with a TDP of 205 Watts?

   Solution:

   Skylake Platinum 8360Y:

   - Core: $2\,units \times (512 \div 64)\,ops\_per\_cycle \times 2\,flop\_per\_op \times 2.4\,Gcycles\_per\_second = 76.8\,GFLOP/s$
   - Socket: $36\ cores \times 76.8\ GFLOP/s\_per\_core = 2764.8\ GFLOP/s$
   - $TDP = 250\ Watts$
   - $E_{eff} = 2764.8 \div 250 = 11.06\ GFLOP/s/Watt$

   Skylake Gold 6354:

   - Core: $2\,units \times (512 \div 64)\,ops\_per\_cycle \times 2\,flop\_per\_op \times 3.0\,Gcycles\_per\_second = 96\,GFLOP/s$
   - Socket: $18\ cores \times 96\ GFLOP/s\_per\_core = 1728\ GFLOP/s$
   - $TDP = 205\ Watts$
   - $E_{eff} = 1728 \div 205 = 8.42\ GFLOP/s/Watt$

2.

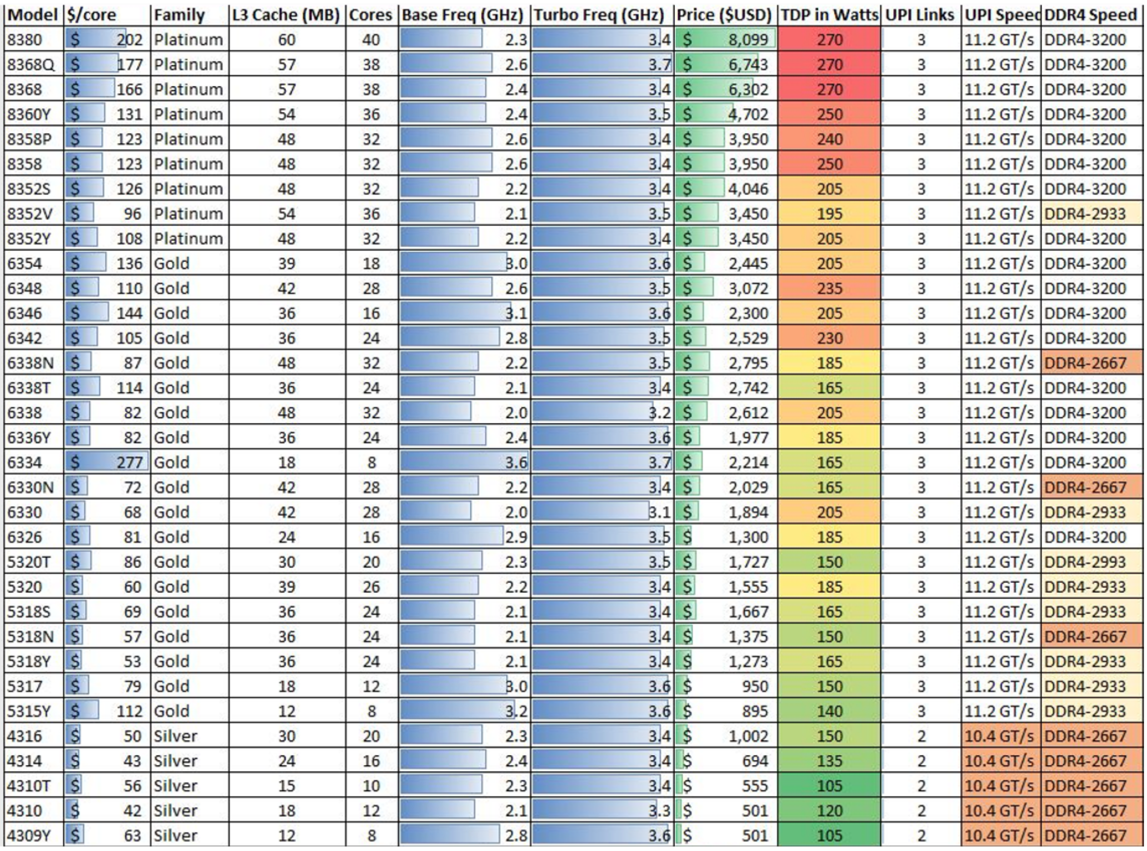
3. The energy efficiency metric for a complete node in the system is defined as the ratio between the peak FLOP/s and the power required to achieve them; the power consumption should include all components in the node (sockets, DIMMs, motherboard and other components), **We ask you to** compute the energy efficiency for a node that includes two Intel sockets Skylake Platinum 8160 with 24 cores running at 2.0 GHz when the 2 AVX-512 units are used to perform double precision computation. Each socket has all its 6 memory channels populated with one DIMM per channel consuming 5 W each. The TDP for the socket is 150 Watts and the rest of components in the motherboard consume 40 Watts.

   Solution:

   The peak performance are obtained from all cores running at maximum capabilities. For the power we need to consider all components in the node:

   - Core: $2\,units \times (512 \div 64)\,ops\_per\_cycle \times 2\,flop\_per\_op \times 2.0\,Gcycles\_per\_second = 64\,GFLOP/s$
   - Socket: $24\ cores \times 64\ GFLOP/s\_per\_core = 1536\ GFLOP/s$
   - $Power = 2 \times 150 + 12 \times 5 + 40 = 400\ Watts$
   - $E_{eff} = (2 \times 1536) \div 400 = 7.68\ GFLOP/s/Watt$

4. Consider the following sockets available in the 3rd generation of Intel Xeon Scalable product line. Which socket would you choose to build a node with a total of 32 cores (which could include 1, 2 or 4 sockets per node, all identical in case of multisocket)? Your decision would depend on a lot of parameters, right?



Model	$/core	Family	L3 Cache (MB)	Cores	Base Freq (GHz)	Turbo Freq (GHz)	Price ($USD)	TDP in Watts	UPI Links	UPI Speed	DDR4 Speed
8380	$ 202	Platinum	60	40	2.3	3.4	$ 8,099	270	3	11.2 GT/s	DDR4-3200
8368Q	$ 177	Platinum	57	38	2.6	3.7	$ 6,743	270	3	11.2 GT/s	DDR4-3200
8368	$ 166	Platinum	57	38	2.4	3.4	$ 6,302	270	3	11.2 GT/s	DDR4-3200
8360Y	$ 131	Platinum	54	36	2.4	3.5	$ 4,702	250	3	11.2 GT/s	DDR4-3200
8358P	$ 123	Platinum	48	32	2.6	3.4	$ 3,950	240	3	11.2 GT/s	DDR4-3200
8358	$ 123	Platinum	48	32	2.6	3.4	$ 3,950	250	3	11.2 GT/s	DDR4-3200
8352S	$ 126	Platinum	48	32	2.2	3.4	$ 4,046	205	3	11.2 GT/s	DDR4-3200
8352V	$ 96	Platinum	54	36	2.1	3.5	$ 3,450	195	3	11.2 GT/s	DDR4-2933
8352Y	$ 108	Platinum	48	32	2.2	3.4	$ 3,450	205	3	11.2 GT/s	DDR4-3200
6354	$ 136	Gold	39	18	3.0	3.6	$ 2,445	205	3	11.2 GT/s	DDR4-3200
6348	$ 110	Gold	42	28	2.6	3.5	$ 3,072	235	3	11.2 GT/s	DDR4-3200
6346	$ 144	Gold	36	16	3.1	3.6	$ 2,300	205	3	11.2 GT/s	DDR4-3200
6342	$ 105	Gold	36	24	2.8	3.5	$ 2,529	230	3	11.2 GT/s	DDR4-3200
6338N	$ 87	Gold	48	32	2.2	3.5	$ 2,795	185	3	11.2 GT/s	DDR4-2667
6338T	$ 114	Gold	36	24	2.1	3.4	$ 2,742	165	3	11.2 GT/s	DDR4-3200
6338	$ 82	Gold	48	32	2.0	3.2	$ 2,612	205	3	11.2 GT/s	DDR4-3200
6336Y	$ 82	Gold	36	24	2.4	3.6	$ 1,977	185	3	11.2 GT/s	DDR4-3200
6334	$ 277	Gold	18	8	3.6	3.7	$ 2,214	165	3	11.2 GT/s	DDR4-3200
6330N	$ 72	Gold	42	28	2.2	3.4	$ 2,029	165	3	11.2 GT/s	DDR4-2667
6330	$ 68	Gold	42	28	2.0	3.1	$ 1,894	205	3	11.2 GT/s	DDR4-2933
6326	$ 81	Gold	24	16	2.9	3.5	$ 1,300	185	3	11.2 GT/s	DDR4-3200
5320T	$ 86	Gold	30	20	2.3	3.5	$ 1,727	150	3	11.2 GT/s	DDR4-2993
5320	$ 60	Gold	39	26	2.2	3.4	$ 1,555	185	3	11.2 GT/s	DDR4-2933
5318S	$ 69	Gold	36	24	2.1	3.4	$ 1,667	165	3	11.2 GT/s	DDR4-2933
5318N	$ 57	Gold	36	24	2.1	3.4	$ 1,375	150	3	11.2 GT/s	DDR4-2667
5318Y	$ 53	Gold	36	24	2.1	3.4	$ 1,273	165	3	11.2 GT/s	DDR4-2933
5317	$ 79	Gold	18	12	3.0	3.6	$ 950	150	3	11.2 GT/s	DDR4-2933
5315Y	$ 112	Gold	12	8	3.2	3.6	$ 895	140	3	11.2 GT/s	DDR4-2933
4316	$ 50	Silver	30	20	2.3	3.4	$ 1,002	150	2	10.4 GT/s	DDR4-2667
4314	$ 43	Silver	24	16	2.4	3.4	$ 694	135	2	10.4 GT/s	DDR4-2667
4310T	$ 56	Silver	15	10	2.3	3.4	$ 555	105	2	10.4 GT/s	DDR4-2667
4310	$ 42	Silver	18	12	2.1	3.3	$ 501	120	2	10.4 GT/s	DDR4-2667
4309Y	$ 63	Silver	12	8	2.8	3.6	$ 501	105	2	10.4 GT/s	DDR4-2667

So for example, imagine that your system needs to ensure an average bandwidth to local memory (NOT remote through UPI) per core of 12 GB/s with maximum GFLOP/s.

- Which solution would you select?
- Which total bandwidth and GFLOP/s would you achieve at the node level with the solution selected?
- Draw its roofline model at the socket level and predict the performance of an application with AI=8 (compute or memory bound?).
- If each memory DIMM is 5W and node boards consume 40W, which would be the energy efficiency achieved by the solution selected?

Solution: We should look at the maximum memory frequency accepted by the sockets with 8, 16 and 32 cores. Depending on the socket we should then go for designs that are single socket, dual socket (2 UPI minimum) and quad socket (3 UPI minimum). Therefore, possible sockets are:

- single socket: 8358 (DDR-3200), 8352S or Y (DDR-3200), 6338 or N (DDR-3200 or 2667)
- dual socket: 6346 (DDR-3200), 6326 (DDR-3200), 4314 (DDR-2667)
- quad socket: 5315Y (DDR-2933), 6334 (DDR-3200)

All sockets have 6 memory channels. In the single socket designs with 32 cores per socket the bandwidth per core is 128/32 (for DDR-2667) and 153,6/32 (for DDR-3200), that is 4 and 4.8 GB/s, respectively. In the dual socket designs with 16 cores per socket the bandwidth per core is 128/16 (for DDR-2667) and 153,6/16 (for DDR-3200), that is 8 and 9.6 GB/s, respectively. In the quad socket designs with 8 cores per socket the bandwidth per core is 140,8/8 (for DDR-2933) and 153,6/8 (for DDR-3200), that is 17,6 and 19,2 GB/s, respectively. Therefor, if our design needs to ensure a minimum bandwidth per core of 12 GB/s the only viable solutions are 5315Y and 6334 mounted in a quad-socket board. The first one is more than half the price of the second and better in terms of energy efficiency. However, frequency is a bit different: 3.2 vs. 3.6 GHz, which result in overall FLOP/s value for the whole node of 3276 and 3686 GFLOP/s. Then the chosen solution is 6334 (bandwidth 19,2 GB/s per core, or equivalently 615 GB/s per node and close to 3686 GFLOP/s per node). Overall the design achieves FLOP/s/Byte ratio close to 6, so with AI=4 the application would be memory bound with a predicted performance of 2460 GFLOP/s. In terms of energy efficiency, the design is consuming $165 \times 4 + 5 \times 24 + 40 = 820 Watts$ resulting in 4.5 GFLOP/s/Watt, approx. Of course much worse than the single socket design (2662,4 GFLOP/s for 8358 with total power consumption of $240 + 5 \times 6 + 40 = 310$ resulting in 8,58 GFLOP/s/Watt.

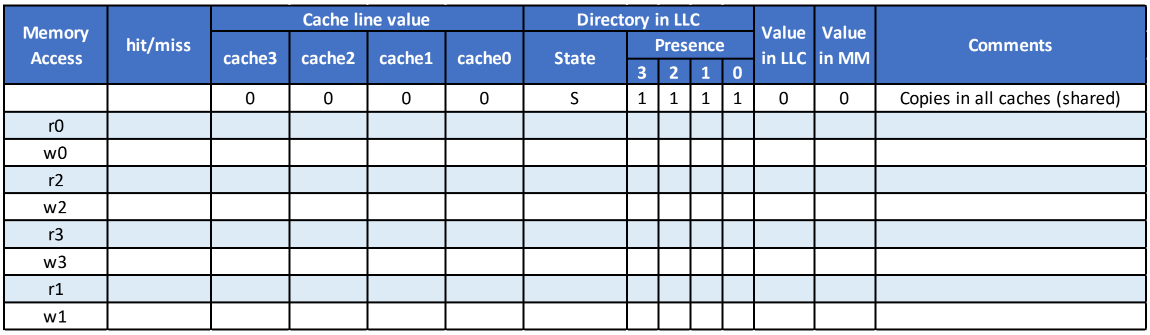
5.

6.

7. Consider the following program fragment executed on a node with a single socket with four cores using 4 *OpenMP* threads in total. Cores in the socket have their local *L1-L2* cache and share the access to a *Last–Level Cache LLC*; the socket has its own *Main Memory MM*. Data coherence is maintained using a *Write-Invalidate* protocol with a directory attached to the *LLC*.

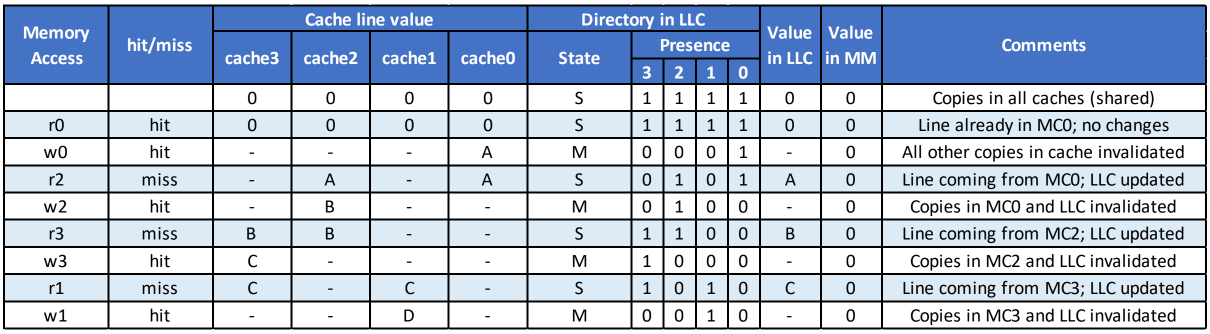
```
double result[4];
void dot_product(double *A, double *B, int n) {
   #pragma omp parallel for
   for (int i=0; i< n; i++)
       #pragma omp atomic
       result[0] += A[i] * B[i];
}
```

Fill in the following table with the evolution of the coherence state across cores in the socket for the first iteration of the `i` loop executed in each processor, only for the accesses to variable `result[0]`; the table shows the order in which the read (`rx`) and write (`wx`) accesses to that variable by core `x` happen. The initial state for the line containing it is *Shared* with copies in all cores.



Memory Access	hit/miss	Cache line value				Directory in LLC					Value in LLC	Value in MM	Comments
		cache3	cache2	cache1	cache0	State	Presence						
							3	2	1	0			
		0	0	0	0	S	1	1	1	1	0	0	Copies in all caches (shared)
r0													
w0													
r2													
w2													
r3													
w3													
r1													
w1													

Solution:



Memory Access	hit/miss	Cache line value				Directory in LLC					Value in LLC	Value in MM	Comments
		cache3	cache2	cache1	cache0	State	Presence						
							3	2	1	0			
		0	0	0	0	S	1	1	1	1	0	0	Copies in all caches (shared)
r0	hit	0	0	0	0	S	1	1	1	1	0	0	Line already in MC0; no changes
w0	hit	-	-	-	A	M	0	0	0	1	-	0	All other copies in cache invalidated
r2	miss	-	A	-	A	S	0	1	0	1	A	0	Line coming from MC0; LLC updated
w2	hit	-	B	-	-	M	0	1	0	0	-	0	Copies in MC0 and LLC invalidated
r3	miss	B	B	-	-	S	1	1	0	0	B	0	Line coming from MC2; LLC updated
w3	hit	C	-	-	-	M	1	0	0	0	-	0	Copies in MC2 and LLC invalidated
r1	miss	C	-	C	-	S	1	0	1	0	C	0	Line coming from MC3; LLC updated
w1	hit	-	-	D	-	M	0	0	1	0	-	0	Copies in MC3 and LLC invalidated

Would the contents of the table be different if the program fragment executed is the one shown below? To answer the question you can assume that 1) cache lines are 32 bytes wide; 2) each double occupies 8 bytes; and 3) vector `result` is aligned with the beginning of a memory line.

```
double result[4];
void dot_product(double *A, double *B, int n) {
   #pragma omp parallel for
   for (int i=0; i< n; i++)
       result[omp_get_thread_num()] += A[i] * B[i];
}
```

Solution:

The contents of the table would be the same since the 4 elements of the vector are in the same cache line. Coherence is kept at the cache line level.

8. **(new)** Following the previous problem, let's consider two new scenarios:

- **Scenario 1:** Dual socket, each socket with two cores. Cores in each socket have their local L1-L2 cache and share the access to the Last Level Cache LLC inside the socket; each socket has its own Main Memory MM, shared between both sockets. Remote caching for lines in MM is not enabled (i.e. a line of MM can only be cached in the LLC of the socket associated to it). Data coherence is maintained using a Write-Invalidate protocol with a directory attached to the LLC. Memory line containing result[0] assigned to MM of socket 1 (ownership).
- **Scenario 2:** As the previous one but now remote caching for lines in MM is enabled (i.e. a line of MM can be cached in the LLC of both sockets, simultaneously if needed). Data coherence is maintained using a Write-Invalidate protocol with a directory attached to each LLC to keep coherence inside its socket and with a directory attached to each MM to keep coherence between sockets. As before, memory line containing result[0] assigned to MM of socket 1 (ownership).

We ask you to fill in the following two tables for each one of the two new node configurations:

(a) **Scenario 1:**

Memory Access	hit/miss	Value in		Directory in LLC0					Value in LLC0	Comments	Value in		Directory in LLC1					Value in LLC1	Value in MM1
		MC1	MC0	State	Presence						MC3	MC2	State	Presence					
					3	2	1	0						3	2	1	2		
		0	0							Initial state for line containing result	0	0	S	1	1	1	1	0	0
r0																			
w0																			
r2																			
w2																			
r3																			
w3																			
r1																			
w1																			

(b) **Scenario 2:**

Memory Access	hit/miss	Value in		Directory in LLC0			Value in LLC0	Comments	Value in		Directory in LLC1			Value in LLC1	Directory in MM1			Value in MM1
		MC1	MC0	State	Presence				MC3	MC2	State	Presence			State	Presence		
					1	0						3	2			1	0	
		0	0	S	1	1	0	Initial state for line containing result	0	0	S	1	1	0	S	1	1	0
r0																		
w0																		
r2																		
w2																		
r3																		
w3																		
r1																		
w1																		

Solution:

(a) Scenario 1:

Memory Access	hit/miss	Value in		Directory in LLC0					Value in LLC0	Comments	Value in		Directory in LLC1					Value in LLC1	Value in MM1
		MC1	MC0	State	Presence						MC3	MC2	State	Presence					
					3	2	1	0						3	2	1	0		
		0	0							Initial state for line containing result	0	0	S	1	1	1	1	0	0
r0	hit	0	0							Line already in MC0; no changes	0	0	S	1	1	1	1	0	0
w0	hit	-	A							All other copies in cache invalidated	-	-	M	0	0	0	1	-	0
r2	miss	-	A							Line coming from MC0 LLC1 updated	-	A	S	0	1	0	1	A	0
w2	hit	-	-							All other copies in cache invalidated	-	B	M	0	1	0	0	-	0
r3	miss	-	-							Line coming fom MC2 LLC1 updated	B	B	S	1	1	0	0	B	0
w3	hit	-	-							All other copies in cache invalidated	C	-	M	1	0	0	0	-	0
r1	miss	C	-							Line coming from MC3 LLC1 updated	C	-	S	1	0	1	0	C	0
w1	hit	D	-							All other copies in cache invalidated	-	-	M	0	0	1	0	-	0

(b) Scenario 2:

Memory Access	hit/miss	Value in		Directory in LLC0			Value in LLC0	Comments	Value in		Directory in LLC1			Value in LLC1	Directory in MM1			Value in MM1
		MC1	MC0	State	Presence				MC3	MC2	State	Presence			State	Presence		
					1	0						3	2			1	0	
		0	0	S	1	1	0	Initial state for line containing result	0	0	S	1	1	0	S	1	1	0
r0	hit	0	0	S	1	1	0	Line already in MC0; no changes	0	0	S	1	1	0	S	1	1	0
w0	hit	-	A	M	0	1	-	All other copies in cache invalidated	-	-	-	-	-	-	M	0	1	0
r2	miss	-	A	S	0	1	A	Line coming from MC0 LLC0, LLC1 and MM1 updated	-	A	S	0	1	A	S	1	1	A
w2	hit	-	-	-	-	-	-	All other copies in cache invalidated	-	B	M	0	1	-	M	1	0	A
r3	miss	-	-	-	-	-	-	Line coming fom MC2 LLC1 updated (not MM1)	B	B	S	1	1	B	M	1	0	A
w3	hit	-	-	-	-	-	-	All other copies in cache invalidated	C	-	M	1	0	-	M	1	0	A
r1	miss	C	-	S	1	0	C	Line coming from MC3 LLC1, LLC0 and MM1 updated	C	-	S	1	0	C	S	1	1	C
w1	hit	D	-	M	1	0	-	All other copies in cache invalidated	-	-	-	-	-	-	M	0	1	C

Unit 4

1. The energy efficiency metric for a complete system is defined as the ratio between the peak FLOP/s and the power required to achieve them; the power consumption should include all components in the node (sockets, DIMMs, NIC, motherboard) and all the components in the interconnection network that allow to connect the desired number of nodes, **We ask you to** compute the energy efficiency for cluster with 2048 nodes interconnected with 100 Gbps EDR Infiniband. Each node includes two Intel sockets 8358 with 32 cores each running at 2.6 GHz when the 2 AVX-512 units are used to perform double precision computation. Each socket has all its 8 memory channels populated with one DIMM per channel consuming 5 Watts each. The TDP for the socket is 250 Watts and the motherboard consume 50 Watts. Regarding the interconnection network Nvidia/Mellanox provides the following typical power consumption for its switches and adapters: 6.4 KWatts for the 648-port switch, 4.9 KWatts for the 324-port switch, 3.8 KWatts for the 216-port switch, 136 Watts for the elementary 36-port switch and 22 Watts for the NIC. Compare the energy efficiency obtained for the whole cluster with the efficiency for a single socket and a single node. To build the interconnection network you can visit http://www.mellanox.com/clusterconfig/, looking for the configuration that minimizes power consumption in the network.

   Solution:

   To interconnect the 2048 we have to use a two–level IB network, with a total of $2048/18 = 114$ Leaf switches. For this number of nodes the topology requires spine switches of more than 36 ports, with different options: 18 of 216 port, 9 of 324 ports or 6 of 648 ports, according to the network configurator page from Mellanox. This means $18 \times 3.8 = 68.4\ KWatts$, $9 \times 4.9 = 44.1\ KWatts$ or $6 \times 6.4 = 38.4\ KWatts$. Therefore, we choose the 648 ports option. In total, the 2048 node cluster interconnect would consume $6 \times 6.4 + 114 \times 136 + 2048 \times 22 = 98.96\ KW$.

   The power consumption of the interconnection network just computed should be added to the $2048\ nodes \times (2\ sockets \times (250 + 8 \times 5) + 50\ (board)) = 2048 \times (2 \times 290 + 50) = 2048 \times 630 = 1290.24\ KW$ for the computing part. The total number of FLOP/s is $2048 \times (2 \times (32 \times 512 \div 64 \times 2 \times 2.6\ GHz)) = 2048 \times 2 \times 2662 = 2048 \times 5324 = 10905190\ GFLOP/s$. So the energy efficiency would be: $10905190\ GFLOP/s/(1290.24 + 98.96)\ KW = 7.85\ GFLOP/s/Watt$. So it is reduced from $2662/250 = 10.65$ GFLOP/s/Watt for the socket → 8.16 GFLOP/s/W for the node → 7.85 GFLOP/s/W for the whole system.

2. Consider a single node of the cluster described in the previous problem. **We ask you:**

   - Draw its roofline model at the level of the complete node, considering for the computation roof the capabilities of the two sockets and for the data access roof the capabilities of the NIC indicated.
   - Indicate the performance in GFLOP/s that theoretically would be achieved for three applications that have AI equal to 64, 128 and 256, indicating if they are network or compute bound.
   - Indicate which proposal would you do to ensure that the three applications fully benefit from the computational power offered by the two sockets when running the three applications mentioned above.

Solution:

The total number of FLOP/s is $2 \times (32 \times 512 \div 64 \times 2 \times 2.6\ GHz) = 2662 GFLOP/s$. For the network interface, the bandwidth provided is $100 Gbps \div 8 bits/Byte = 12.5 GB/s$. Therefore the intersection point is found at an $AI = 2662/12.5 = 213$.

3.

4. **We ask you** to answer the following questions:

(a) If we want to build a cluster using Nvidia/Mellanox 200 Gb Infiniband HDR technologies we can use 40-port and 800-port switches. Which would be the largest 2–level switch cluster configuration we would be able to build, assuming either the possibility of using 40-port or 800-port switches at the spine level? In both cases, computing nodes are connected to the network using a 200 Gb HDR NICs.

Solution:

- 800 when using 40-port switches at the spine level (40 leaf and 20 spine)
- 16000 when using 800-port switches at the spine level (800 leaf and 20 spine)

(b) Based on an analysis of the application domains to be executed, we observed that HDR 100 Gb would be sufficient for the NICs. Would the number of nodes that could be connected be different?

Solution:

Twice the previous numbers (1600 and 32000, respectively).

(c) Draw the two different choices for building a cluster with 800 nodes using 200 Gb Infiniband HDR at all levels. Compute their power consumption and space occupied in a rack considering that each 40-port leaf switch (QM8700) consumes 274 Watts and occupies 1U; and each 800-port spine (director) switch (CS8500) consumes 13 KWatts and occupies 29U. Would your selected choice change if we tell you that their unit cost is $15000 and $250000, respectively?

Solution:

- **Solution 1:** Use a single 800-port switch. Space: 29U. Power: 13 KWatts. Cost: $250000.
- **Solution 2:** Use 40 40-port switches at the Leaf level and 20 40-port switches at the Spine level. Space: 60x1U=60U. Power: 60x274W=16,44 KWatts. Cost: 60x$15000=$900000.

In all aspects it is a much better option to use the monolithic 800-port switch.

(d) Repeat the previous question but now considering the design of two systems with 400 nodes or 1600 nodes. How many choices do I have for each one of the two systems? Are switches occupied at their maximum capacity in terms of number of used ports?

Solution:

For 400 nodes again we have two options:

- **Option 1:** Use a single 800-port switch. Space: 29U. Power: 13 KWatts. Cost: $250000.
- **Option 2:** Use 20 40-port switches at the Leaf level and 10 40-port switches at the Spine level. Space: 30x1U=30U. Power: 30x274Watts=8,22 KWatts. Cost: 30x$15000=$450000.

Without considering the price, option 2 appears to be better than option 1, although occupying 1U more. However, the power given of 13KWatts is the maximum when all 800 ports are used; probably the actual power when using 400 ports is much less, close to half. In terms of price, option 2 is still better.

For 1600 nodes we need to go to a configuration using 80 Leaf switches and 2 Spine switches. This would be the only possible configuration occupying 138U, dissipating 47,9 KWatts and with a cost of 1.7 million $.

Unit 5

1. Repeat the first problem in Unit 4 but now assuming that the node includes 2 GPUs Ampere A100 per socket, each one connected through a PCIe Gen4 x16 port. In total 4 GPUs per node. As shown in slides 17 and 18 of Unit 5, each GPU contributes with 9.75 TFLOP/s for double precision and 40 GB of HBM2 memory able to provide 1550 GB/s, overall with a TDP of 400 Watts. In addition to that, we ask you to draw the roofline model at the node level (i.e. the roofline relating the peak computational performance of the whole node with the network bandwidth provided by the NIC) for the two configurations, without and with GPUs.

   Solution:

   In terms of the interconnection network, there are no changes in the power consumption since the number of nodes is still the same: $6 \times 6.4 + 114 \times 136 + 2048 \times 22 = 98.96\ KW$. This should be added to the 2048 $nodes \times (2\ sockets \times (250 + 2\ GPUs \times 400 + 8 \times 5) + 50\ (board)) = 2048 \times (2 \times 1050 + 50) = 2048 \times 2150 = 4403,2\ KWatts$ for the computing part. The total number of FLOP/s is 2048 $nodes \times (2\ sockets \times (32 \times 512 \div 64 \times 2 \times 2.6\ GHz + 2\ GPUs \times 9750)) = 2048 \times 2 \times 22162 = 2048 \times 44324 = 90775552\ GFLOP/s = 90,77\ PFLOP/s$. So the energy efficiency would be: $90775552\ GFLOP/s/(4567 + 98.96)\ KWatt = 19,45\ GFLOP/s/Watt$, much higher than the 7.85 GFLOP/s/W achieved for the non–accelerated cluster.

2. Assume that we want to re-train a language model such as GPT-3 every day, using the minimum possible number of nodes of the supercomputer available. Each one of the nodes consists of 4 GPUs H100, each GPU able to perform 33.5 TFLOP/s for double precision, 67 TFLOP/s for single precision and 989.4 and 1978.9 TFLOP/s when using the Tensor cores with half precision (FP16) and minifloat (FP8), respectively. In order to tune the already pre-trained model we are going to use a dataset with 1 billion (10^9) items per day; processing each one of these items in our GPT-3 model needs, on an average, 1 trillion (10^{12}) FP16 FLOP. **We ask you** to compute the number of nodes that should be allocated in the supercomputer. If the AI framework used is able to also use the FLOP/s provided by each one of the two general-purpose sockets available in each node, based on the Intel 8468H Sapphire Rapids processor with 48 cores per socket, each core with 2 AVX-512 units running at 3.8 GHz, does the number of nodes that need to be allocated change?

   Solution: On one side, the total number of FLOP that need to be performed are $10^9\ items \times 10^{12}\ FLOP/item = 10^{21}\ FLOP$. The machine available has a computational power of $N\ nodes \times 4\ GPU/node \times 989,9\ TFLOP/s/GPU = 4 \times N\ PFLOP/s$, approximately. The problem wants to be solved in 1 days, which translated into seconds is $1day \times 24hours/day \times 60minutes/hour \times 60seconds/minute = 86400seconds$. Therefore $4 \times N \times 10^{15} = 10^{21}/86400$, which results in $N = 11,57$, that is 12 GPUs. If the FLOP/s provided by the sockets are also usable, then the computational power would be increased by $x \times 2sockets \times 48cores/socket \times 128FLOP/core \times 3.8GHz = 46,7 \times NTFLOP/s$. In total 4046,7 TFLOP/s per node. Therefore, the number of nodes would sligthly decrease to $N = 2,86$, i.e. 3 nodes.