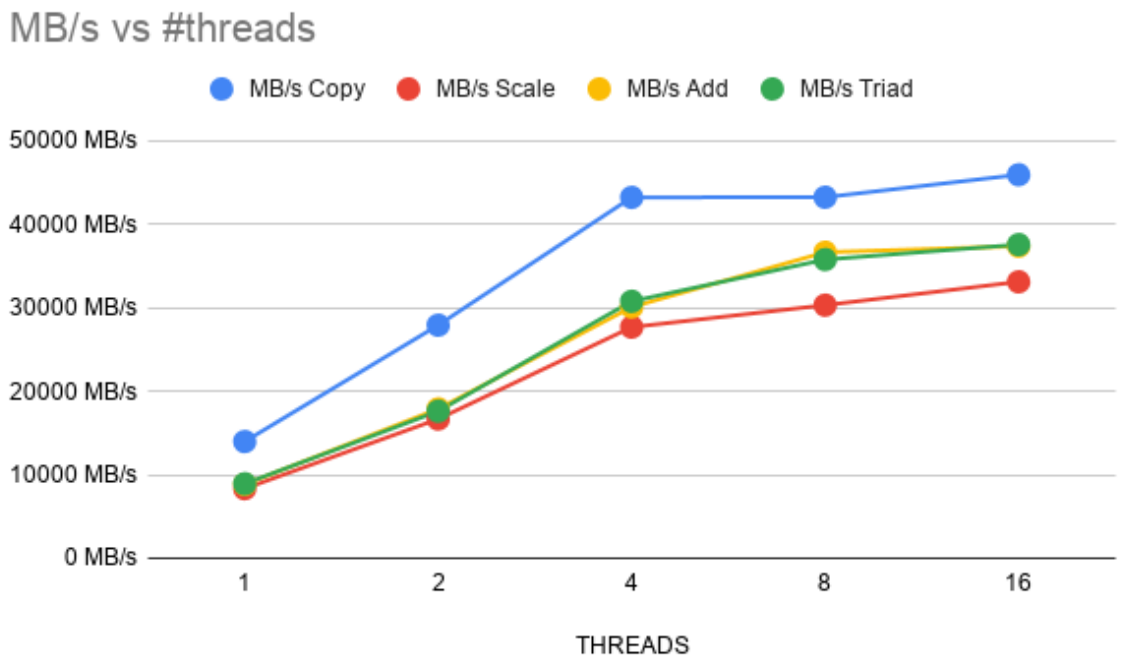
LAB 3: Performance characteritzation of HPC clusters

En el document s'inclouen els resultats de diferents programes de prova. Els resultats es comparen amb la màquina BOADA. Cada programa de proves està dissenyat per un supòsit.

Stream

Aquest joc de proves comprova operacions d'alt ús de memória. L'ample de banda obtingut de les diferents operacions es representa en el gràfic.



Boada utilitza les memories DDR4-2400 i DDR4-2133 que tenen sobre el paper un ample de banda de pic de 19.200 i 17.066 MB/s.

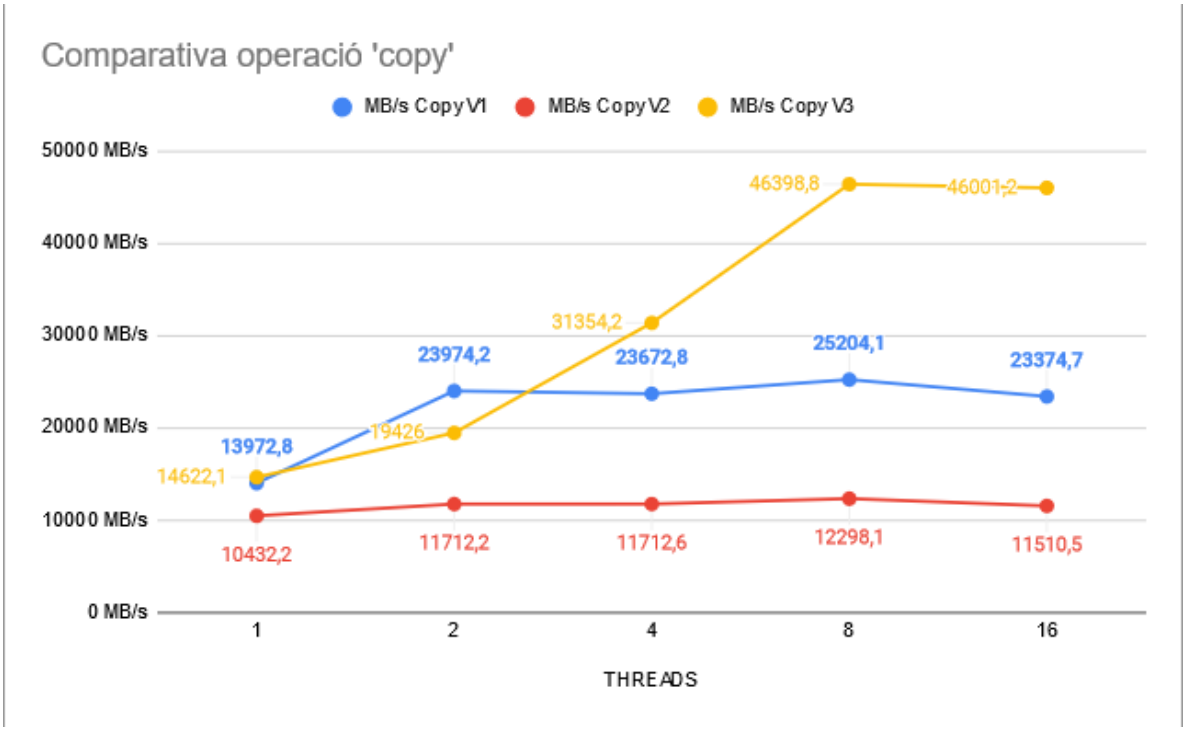
Per mirar el rendiment concret de la memòria hauríem de centrar-nos en els resultats de l'operació de còpia, ja que les altres operacions veuen limitat el seu ample pel temps en realitzar les operacions.

Mirant els resultats, podem comprobar que existeixen diverses memòries que s'accedeixen en paral·lel; donant lloc a una lantecia major a la l'especificada pel tipus de memòria.

Pel que fa a l'ús de les directives numactl, tenim que aquestes controlen les polítiques de planificació i ubicació de memòria. L'opció `--membind` fa que només s'ubiqui memoria als nodes; d'altra banda que l'opció `--cpunodebind` fa que s'executin les comandes als nodes seleccionats.

S'han aplicat els següents paràmetres en els diferents test proposats:

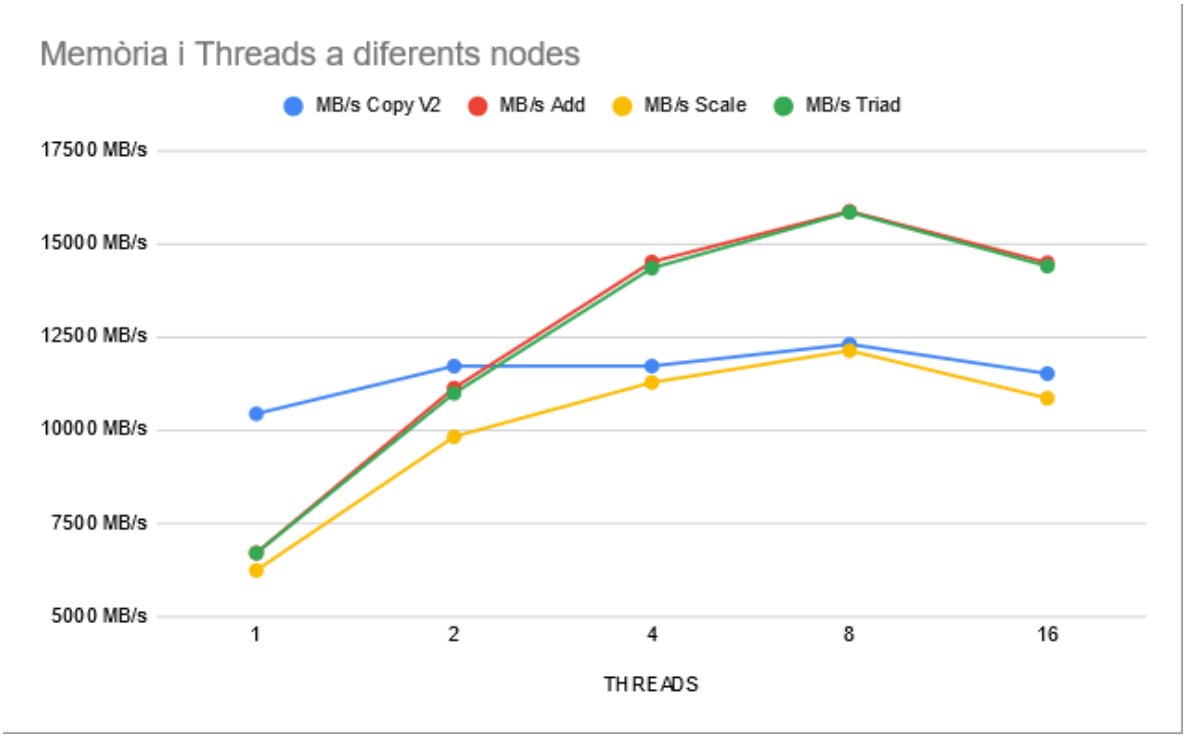
- Ubicar la memória i els threads a un NUMA node.(V1 al gràfic)`numactl -m 0 -N 0`
- Ubicar la memória a un node i els threads a un altre node.(V2 al gràfic)`numactl -m 0 -N 1`
- Ubicar memória i threads a dos nodes.(V3 al gràfic)`numactl -m 0,1 -N 0,1`



En el gràfic, que compara l'ample de banda en les tres versions podem veure:

- La segona versió (memoria i thread en diferent) node es la que pitjor rendiment obté, el rendiment ve condicionat per la comunicació de dades entre els threads i la memória(que estan a diferent node).
- La primera versió no escala al augmentar el numero de threads, ja que no s'aprofita tot el paral·lelisme possible (respecte a la versió 3) al mantenir l'execució en un únic node.
- La tercera versió es la que dóna el millor rendiment, molt similar al de la segona versió amb pocs threads pero ràpidament escala el seu rendiment amb els threads.
- El fet que amb dos threads la tercera versió estigui tant (relativament) lluny de la segona versió pot ser degut a un cas extrem en la justa execucció del test de proves.

Respecte aquest segon gràfic, podem veure que les operacions que requereixen càlcul (add/triad) tenen un millor ample de banda que la operació de copia, això es dóna pel temps de latència de l'operacio, aquest temps emmascara la latencia de memoria (que en aquest cas esta a un altre node).



En resum, mentres es fa l'operació s'aprofita aquest temps per transmetre a l'altre node els resultats. Aquesta reflexió és interesant pero tot i això el rendiment d'aquesta versió es molt menor a les demés

Linpack

Aquest *benchmark* està destinat a mesurar el rendiment en operacions de coma flotant. S'han mesurat el programa sobre 1 i 2 nodes amb diferents configuracions i nombre de processos. Als gràfics es veuen en verd les millors configuracions per cada configuració i l'*speed-up* respecte la pitjor configuració(també s'adjunta el full de cálcul).

- Per 1 node, s'han trobat els següents resultats:

1	1x1		
NB's	Time 1x1	GFlops	Speed-UP
32	66,73	5,11	0
64	56,06	6,09	19,03
128	53,85	6,34	23,92
256	53,16	6,42	25,53

2	2x1			1x2		
NB's	Time 2x1	GFlops	Speed-UP	Time 1x2	GFlops	Speed-UP
32	34,22	9,97	0	33,6	10,14	1,85
64	29,83	11,44	14,72	28,93	11,80	18,29
128	28,51	11,97	20,03	28,33	12,05	20,79
256	28,90	11,81	18,41	28,63	11,92	19,52

4	4x1			2x2			1x4		
NB's	Time 4x1	GFlops	Speed-UP	Time 2x2	GFlops	Speed-UP	Time 1x4	GFlops	Speed-UP
32	17,95	19,01	0	17,31	19,72	3,70	17,42	19,60	3,04
64	15,75	21,67	13,97	15,19	22,47	18,17	15,28	22,34	17,47
128	15,53	21,98	15,58	14,85	22,99	20,88	15,14	22,55	18,56
256	16,41	20,81	9,38	15,44	22,11	16,26	16,01	21,32	12,12

6	6x1			2x3			3x2			1x6		
NB's	Time 6x1	GFlops	Speed-UP	Time 2x3	GFlops	Speed-UP	Time 3x2	GFlops	Speed-UP	Time 1x6	GFlops	Speed-UP
32	12,48	27,36	0	11,68	29,24	6,85	11,79	28,96	5,85	11,89	28,70	4,96
64	11,14	30,65	12,03	10,49	32,53	18,97	10,49	32,54	18,97	10,68	31,97	16,85
128	11,22	30,42	11,23	10,43	32,72	19,65	10,38	32,89	20,23	10,79	31,65	15,66
256	12,22	27,94	2,13	11,08	30,82	12,64	11,1	30,77	12,43	11,76	29,02	6,12

8	8x1			2x4			4x2			1x8		
NB's	Time 8x1	GFlops	Speed-Up	Time 2x4	GFlops	Scalability	Time 4x2	GFlops	Scalability	Time 1x8	GFlops	Scalability
32	10,15	33,65	1,08	9,24	36,95	11,04	9,32	36,64	10,09	9,40	36,30	9,15
64	8,91	38,3	13,92	8,14	41,97	26,04	8,07	42,32	27,14	8,42	40,53	21,85
128	9,15	37,3	10,93	8,19	41,71	25,27	8,13	42,00	26,20	8,64	39,49	18,75
256	10,26	33,26	0,00	8,87	38,50	15,67	8,86	38,54	15,80	9,77	34,96	5,02

Per 1 node, el millor rendiment en trobem amb 8 processos i una graella de 4x2.

- Per 2 nodes, s'han trobat els següents resultats:

NB's	2x1			1x2		
	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up
32	32,91	10,37	0,00	31,66	10,78	3,95
64	29,6	11,53	11,18	28,68	11,90	14,75
128	25,57	11,95	28,71	27,90	12,23	17,96
256	28,89	11,81	13,91	28,28	12,07	16,37

NB's	4x1			2x2			1x4		
	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up
32	17,63	19,37	0	17,42	19,59	1,21	17,24	19,8	2,26
64	15,69	21,76	12,36	15,19	22,47	16,06	15,18	22,49	16,14
128	15,48	22,05	13,89	14,85	22,99	18,72	15,09	22,62	16,83
256	16,37	20,85	7,70	15,45	22,09	14,11	15,98	21,36	10,33

NB's	8x1			2x4			4x2			1x8		
	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up
32	10,02	34,06	2,40	9,13	37,85	12,38	9,17	37,22	11,89	9,22	37,01	11,28
64	8,93	38,22	14,89	8,10	42,16	26,67	8,08	42,25	26,98	8,38	40,73	22,43
128	9,37	36,45	9,50	8,15	41,87	25,89	8,14	41,93	26,04	8,63	39,55	18,89
256	10,26	33,26	0,00	8,85	38,57	15,93	8,85	38,57	15,93	9,75	35,00	5,23

NB's	12x1			2x6			4x3			3x4			6x2			1x12		
	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up
32	7,35	46,47	13,74	6,32	54,08	32,28	6,32	54,06	32,28	6,34	53,89	31,86	6,49	52,58	28,81	6,62	51,55	26,28
64	6,70	50,98	24,78	5,73	59,57	45,90	5,82	60,77	46,75	5,64	60,54	48,23	5,82	58,65	43,64	6,13	55,66	36,38
128	7,08	48,24	18,08	5,86	58,24	42,66	5,75	59,42	45,39	5,9	57,85	41,69	5,97	57,23	40,03	6,47	52,76	29,21
256	8,36	40,84	0	6,54	52,20	27,83	6,38	52,25	31,03	6,54	52,93	27,83	6,69	51,03	24,96	7,78	43,91	7,46

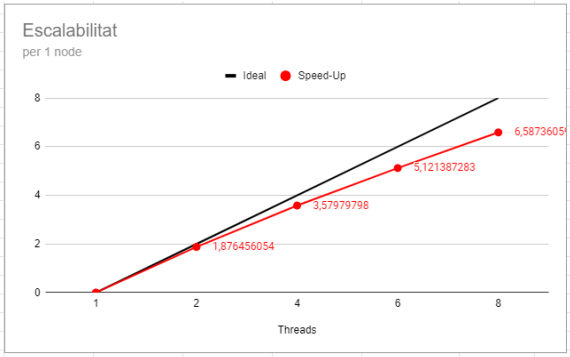
NB's	16x1			2x8			4x4			8x2			1x16		
	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up	Time	GFlops	Speed-Up
32	6,23	54,76	18,30	4,93	69,24	49,49	4,94	69,11	49,19	5,25	65,06	40,38	5,32	64,23	38,53
64	5,66	60,27	30,21	4,53	75,34	62,69	4,37	78,09	68,65	4,68	72,93	57,48	5,02	68,00	46,81
128	6,19	55,14	19,06	4,71	72,49	56,48	4,51	75,63	63,41	4,88	69,92	51,02	5,41	63,05	36,23
256	7,37	46,35	0	5,45	62,70	35,23	5,12	66,74	43,95	5,68	60,12	29,75	6,57	51,95	12,18

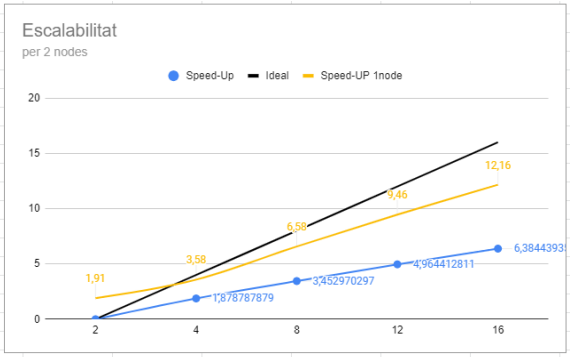
Podem comprovar que les configuracions que millor funcionen són aquells on la graella de processos està "equilibrada", ja que així la càrrega de treball es distribueix millor i obtenim més rendiment.

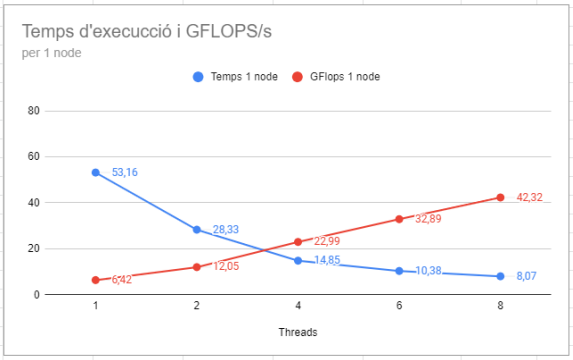
També podem veure que la mida òptima està entre 64 i 128, podem fer les següents hipòtesis:

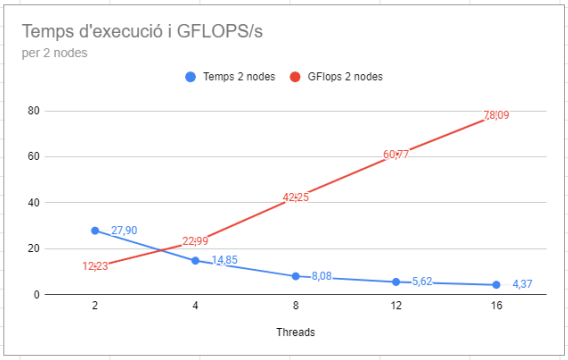
- En el cas d'un procés i 1 node, obtenim més rendiment si fem menys divisions.
- Per sota de 8 processos, la mida òptima és 128, degut possiblement a què cada procés pot reservar aquest espai de memòria sense tenir conflictes per l'espai.
- Amb 8 processos la mida baixa a 64, seguint amb el raonament anterior, es possible que els diferents processos en reservar el mida de 128 tinguin conflictes i es perdi rendiment per culpa de la distribució de la memòria.

Obtenim els següents gràfics per un i dos nodes on podem veure l'escalabilitat i el rendiment:









Cada node Boada té 2 Intel Xeon E5-2609 v4 @ 1,70GHz, amb un rendiment de pic de 217.6 GFLOPS.

HPCG