

Παράλληλα Συστήματα

Χειμερινό Εξάμηνο 2020-2021

Αθανάσιος Αναγνωστόπουλος (Α.Μ. : 1115201800006)

Μιχάλης Βολάκης (Α.Μ. : 1115201800022)

1. Sequential

Matrix Size	Time(s)
840 * 840	0,797
1680 * 1680	3,18
3360 * 3360	12,709
6720*6720	50,81
13440 * 13440	203,328
26480 * 26480	812,3901

Βήματα που ακολουθήσαμε για την βελτιστοποίηση του ακολουθιακού προγράμματος:

1. Η συνάρτηση `one_jacobi_iteration` μεταφέρθηκε εσωτερικά της κύριας (`while`) επανάληψης για να αποφευχθούν οι κλήσεις σε αυτήν.
2. Μεταφέραμε δηλώσεις και αρχικοποιήσεις μεταβλητών εκτός της `while` εφόσον οι τιμές παρέμεναν οι ίδιες.

2. Parallel MPI

Matrix Size/ Processes	4	9	16	25	36	49	64
840 * 840	0,259239	0,122076	0,084338	0,074815	0,091304	0,082749	0,042310
1680 * 1680	0,959181	0,601706	0,452182	0,208230	0,168526	0,129883	0,089150
3360 * 3360	3,738130	2,368907	1,827639	0,975239	0,745386	0,600494	0,466219
6720*6720	14,824242	9,311772	7,109669	3,679720	2,814715	2,232871	1,825282
13440 * 13440	59,190298	36,979222	28,119858	14,533210	10,925871	8,660385	7,126181
26480 * 26480	238,326169	147,056559	112,010842	57,829337	43,433667	34,292015	28,221556

SPEEDUP

Matrix Size/ Processes	9	16	25	36	49	64
840 * 840	2,12	3,07	3,47	2,84	3,13	6,13
1680 * 1680	1,59	2,12	4,61	5,69	7,38	10,76
3360 * 3360	1,58	2,05	3,83	5,02	6,23	8,02
6720*6720	1,59	2,09	4,03	5,27	6,64	8,12
13440 * 13440	1,60	2,10	4,07	5,42	6,83	8,31
26480 * 26480	1,62	2,13	4,12	5,49	6,95	8,44

EFFICIENCY

Matrix Size/ Processes	9	16	25	36	49	64
840 * 840	0,53	0,34	0,22	0,11	0,09	0,13
1680 * 1680	0,18	0,13	0,18	0,16	0,15	0,17
3360 * 3360	0,18	0,13	0,15	0,14	0,13	0,13
6720*6720	0,18	0,13	0,16	0,15	0,14	0,13
13440 * 13440	0,18	0,13	0,16	0,15	0,14	0,13
26480 * 26480	0,18	0,13	0,16	0,15	0,14	0,13

Βήματα για την υλοποίηση του Parallel MPI:

1. Χρησιμοποιήσαμε την καρτεσιανή τοπολογία για την εύρεση του rank του κάθε process καθώς και των γειτόνων του μέσω του MPI_Cart (MPI_Cart_create, MPI_Cart_coords και MPI_Cart_Shift).
2. Τα Send και τα Receives του κάθε process είναι non-blocking έτσι ώστε να υπολογίζονται τα stencils των εσωτερικών λευκών κελιών όσο γίνεται η μεταφορά των halos από και προς τα γειτονικά processes.
3. Χρήση Datatypes , συγκεκριμένα vector και contiguous έτσι ώστε να είναι πιο αποδοτική η μεταφορά ολόκληρων στηλών και γραμμών αντί να μεταφέρεται κάθε κελί ως ξεχωριστό datatype.
4. Τα ranks υπολογίζονται μία φορά και έξω από την while.

Hybrid (MPI + OpenMP)

Matrix Size/ Processes	4	9	16	25	36	49	64
840 * 840	0,195358	0,400307	0,127908	0,157533	0,188834	0,252229	0,264927
1680 * 1680	2,396004	1,698811	0,373360	0,238620	0,263937	0,292318	0,305302
3360 * 3360	9,836861	8,969585	8,758309	5,540009	2,472626	0,825215	0,633309
6720*6720	40,101043	41,165249	45,403705	25,999140	19,957302	17,754605	13,512512
13440 * 13440	162,000152	176,959281	162,320637	119,564771	72,582951	78,000072	62,268803
26480 * 26480	657,427594	755,686787	690,893166	450,445526			

Matrix Size/ Processes	9	16	25	36	49	64
840 * 840	0,49	1,53	1,24	1,03	0,77	0,74
1680 * 1680	1,41	6,42	10,04	9,08	8,20	7,85
3360 * 3360	1,10	1,12	1,78	3,98	11,92	15,53
6720*6720	0,97	0,88	1,54	2,01	2,26	2,97
13440 * 13440	0,92	1,00	1,35	2,23	2,08	2,60
26480 * 26480	0,87	0,95				

SPEEDUP

Matrix Size/ Processes	9	16	25	36	49	64
840 * 840	0,05	0,10	0,05	0,03	0,02	0,01
1680 * 1680	0,16	0,40	0,40	0,25	0,17	0,12
3360 * 3360	0,12	0,07	0,07	0,11	0,24	0,24
6720*6720	0,11	0,06	0,06	0,06	0,05	0,05

Matrix Size/ Processes	9	16	25	36	49	64
13440 * 13440	0,10	0,06	0,05	0,06	0,04	0,04
26480 * 26480	0,10	0,06				

Βήματα για την υλοποίηση του Hybrid MPI + OpenMP:

Στο OpenMP έγινε παραλληλοποίηση της διπλής for για τον υπολογισμό των white boxes καθώς και των υπόλοιπων 4 επαναλήψεων. Στο OpenMP παρατηρήσαμε αύξηση του χρόνου εκτέλεσης σε σχέση με το καθαρό MPI. Ένας από τους λόγους είναι η αναλογία μεταξύ διεργασιών και νημάτων (threads) καθώς και το scheduling δεν συμβάλλει στο optimization.

Στο OpenMP χρησιμοποιούμε την αναλογία 2δ-4ν, δηλαδή τον διπλό αριθμό threads για κάθε διεργασία.

CUDA

Matrix Size/ Processes	1 GPU	2 GPUs
840 * 840	20ms	67ms
1680 * 1680	327ms	327ms
3360 * 3360	612ms	11ms
6720*6720	1sec 123ms	15ms
13440 * 13440	2sec 150ms	
26480 * 26480		

Στο Cuda παρατηρήσαμε μείωση του χρόνου εκτέλεσης λόγω της χρήσης της GPU για τον υπολογισμό των πράξεων. Υπάρχει σημαντική μείωση χρόνων μεταξύ 1 και 2 GPUs εφόσον στις 2 GPU ο πίνακας χωρίζεται στην μέση και εκτελούνται παράλληλα.