



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Συστήματα Παράλληλης Επεξεργασίας

Άσκηση 4: Παραλληλοποίηση και βελτιστοποίηση αλγορίθμων σε αρχιτεκτονικές κατανομημένης μνήμης

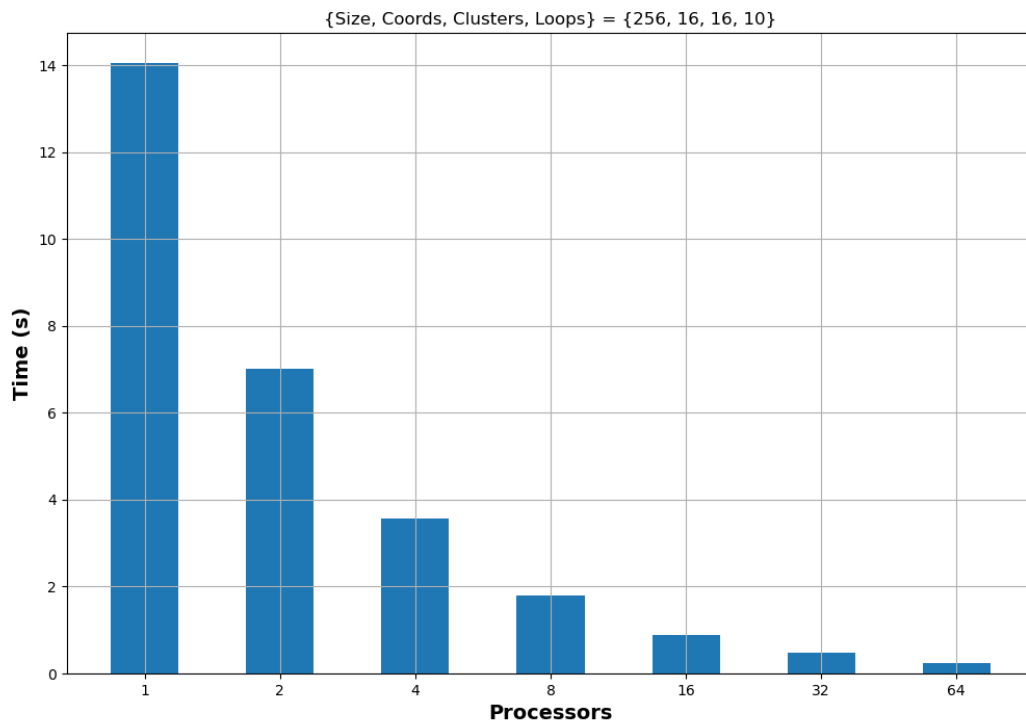
Όνομα	Επώνυμο	A.M.
Αλτάν	Αβτζή	03119241
Τζόναταν	Λουκάι	03119230
Σταύρος	Λαζάρου	03112642

Αλγόριθμος K-means

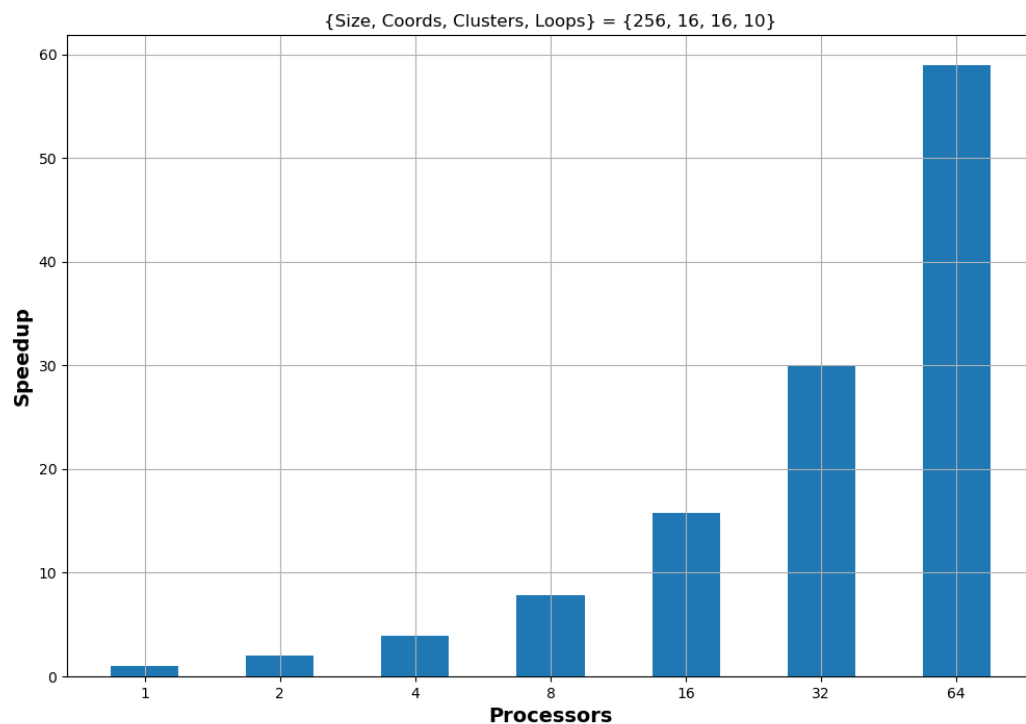
Μετρήσεις και αξιολόγηση

- Γραφικές Παραστάσεις

KMeans MPI Benchmark



KMeans MPI Speedup



Από τα αποτελέσματα αυτά παρατηρούμε ότι επιτυγχάνουμε σχεδόν γραμμική επιτάχυνση, όσο αυξάνονται οι επεξεργαστές, ειδικά μέχρι τους 16. Από τους 32 επεξεργαστές και μετά, οι αποδοτικότητα μειώνεται ελάχιστα αλλά παραμένει ικανοποιητικά ψηλά, υποδεικνύοντας ότι το σύστημα κατανεμημένης μνήμης αξιοποιεί αποτελεσματικά τους πόρους του.

Διάδοση θερμότητας σε δύο διαστάσεις

Υλοποίηση υπολογιστικού πυρήνα Jacobi

- Μοιρασμός των δεδομένων από τον καθολικό στους τοπικούς πίνακες

```
MPI_Scatterv(U[0], scattercounts, scatteroffset, global_block, &u_current[1][1],  
zero2d(u_previous, local[0]+2, local[1]+2);
```

- Ορισμός τύπου δεδομένων γραμμών-στηλών

```
MPI_Datatype column;  
MPI_Type_vector( local[0], 1, local[1]+2, MPI_DOUBLE, &dummy);  
MPI_Type_create_resized(dummy, 0, sizeof(double), &column);  
MPI_Type_commit(&column);  
  
MPI_Datatype row;  
MPI_Type_contiguous( local[1], MPI_DOUBLE, &row);  
MPI_Type_commit(&row);
```

- Εύρεση γειτονικών διεργασιών

```
MPI_Cart_shift(CART_COMM, 0, 1, &north, &south);  
MPI_Cart_shift(CART_COMM, 1, 1, &west, &east);
```

- Ορισμός ορίων επανάληψης

```
i_min = 1;  
i_max = local[0];  
j_min = 1;  
j_max = local[1];  
  
//north  
if (north == MPI_PROC_NULL) {  
    i_min = 2;  
}  
  
//south  
if (south == MPI_PROC_NULL) {  
    int pad_i = global_padded[0] - global[0];  
    i_max = local[0]-1-pad_i;  
}  
  
//east  
if (east == MPI_PROC_NULL) {
```

```

    int pad_j = global_padded[1] - global[1];
    j_max = local[1]-1-pad_j;
}

//west
if (west == MPI_PROC_NULL) {
    j_min = 2;
}

```

- Κύριο loop. Αποστολή/παραλαβή δεδομένων προς/από γειτονικούς επεξεργαστές, και υπολογισμός

```

swap=u_previous;
u_previous=u_current;
u_current=swap;

MPI_Request reqs[8];
MPI_Status stats[8];
int reqs_count = 0;

if (north != MPI_PROC_NULL) {
    MPI_Isend(&u_previous[1][1], 1, row, north, 0, CART_COMM, &reqs[reqs_count++]);
    MPI_Irecv(&u_previous[0][1], 1, row, north, 0, CART_COMM, &reqs[reqs_count++]);
}

if (south != MPI_PROC_NULL) {
    MPI_Isend(&u_previous[local[0]][1], 1, row, south, 0, CART_COMM, &reqs[reqs_count++]);
    MPI_Irecv(&u_previous[local[0]+1][1], 1, row, south, 0, CART_COMM, &reqs[reqs_count++]);
}

if (east != MPI_PROC_NULL) {
    MPI_Isend(&u_previous[1][local[1]], 1, column, east, 0, CART_COMM, &reqs[reqs_count++]);
    MPI_Irecv(&u_previous[1][local[1]+1], 1, column, east, 0, CART_COMM, &reqs[reqs_count++]);
}

if (west != MPI_PROC_NULL) {
    MPI_Isend(&u_previous[1][1], 1, column, west, 0, CART_COMM, &reqs[reqs_count++]);
    MPI_Irecv(&u_previous[1][0], 1, column, west, 0, CART_COMM, &reqs[reqs_count++]);
}

MPI_Waitall(reqs_count, reqs, stats);

gettimeofday(&tcs, NULL);

for (i=i_min; i<=i_max; i++) {
    for (j=j_min; j<=j_max; j++) {
        u_current[i][j]=(u_previous[i-1][j]+u_previous[i+1][j]+u_previous[i][j-1]+u_previous[i][j+1])/4;
    }
}

gettimeofday(&tcf, NULL);

```

```
tcomp+=(tcf.tv_sec-tcs.tv_sec)+(tcf.tv_usec-tcs.tv_usec)*0.000001;

#ifdef TEST_CONV
if (t%C==0) {
    converged = converge(u_previous,u_current,i_min,i_max,j_min,j_max);
    MPI_Allreduce(&converged,&global_converged,1,MPI_INT,MPI_LAND,MPI_COMM_W

}
#endif
```

- Συγκέντρωση των δεδομένων πίσω στον καθολικό πίνακα

```
MPI_Gatherv(&u_current[1][1],1,local_block,U[0],scattercounts,scatteroffset
```

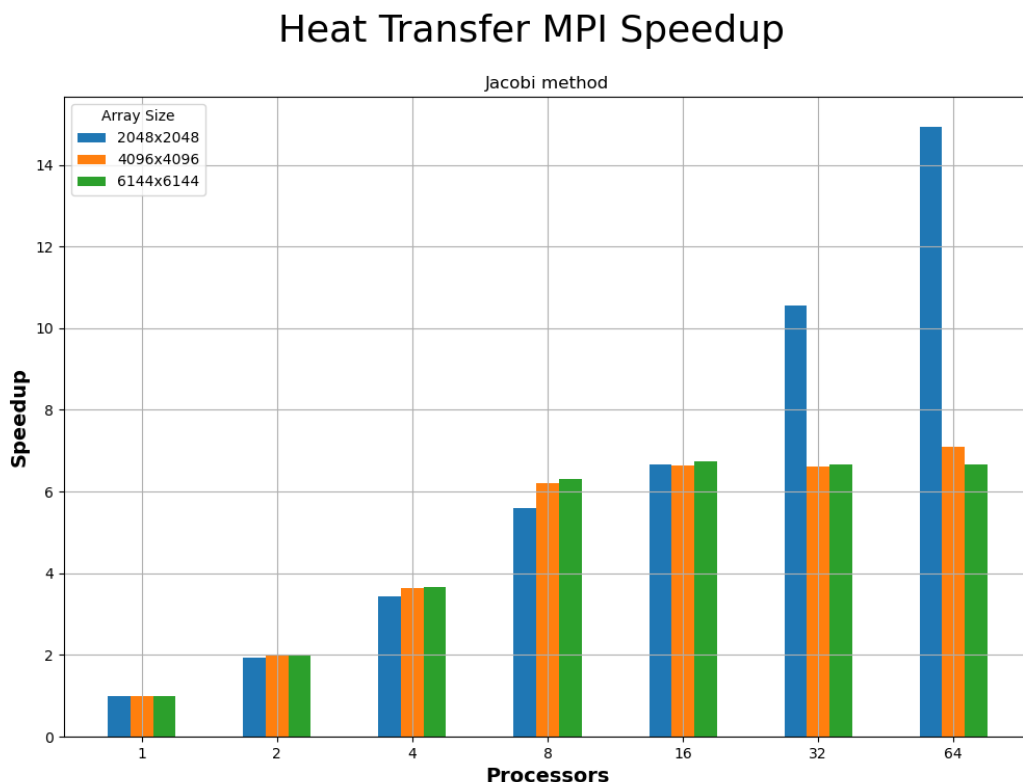
Μετρήσεις με έλεγχο σύγκλισης

Ο χρόνος εκτέλεσης του προγράμματος για τη μέθοδο Jacobi ήταν κατά μέσο όρο:

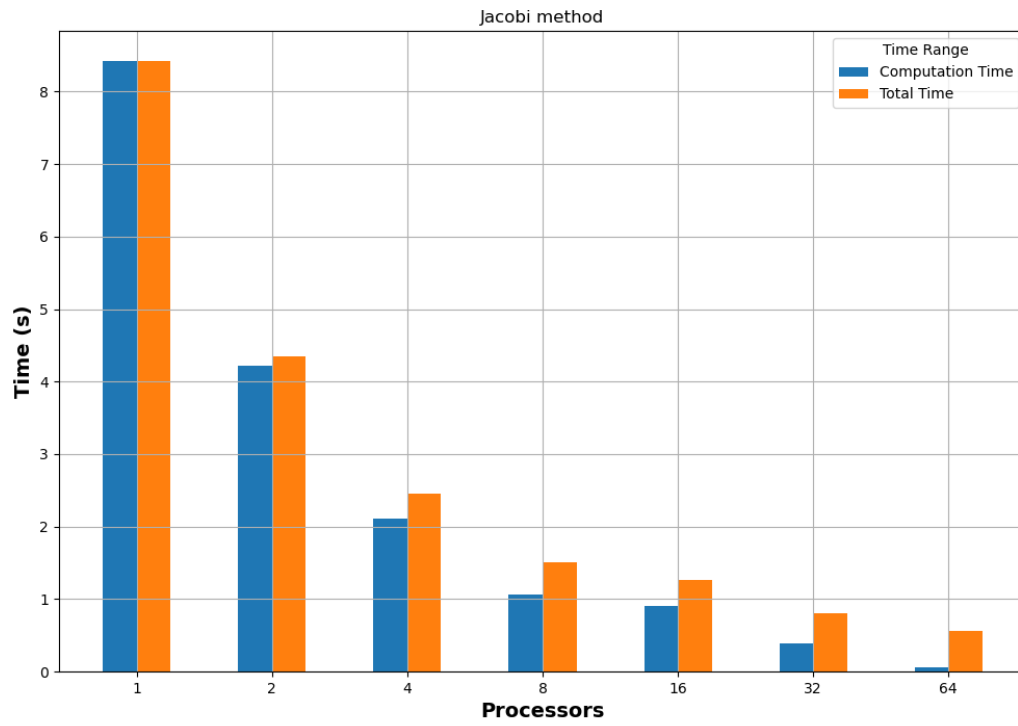
- Total time: 517s
- Computation time: 40s

Μετρήσεις χωρίς έλεγχο σύγκλισης

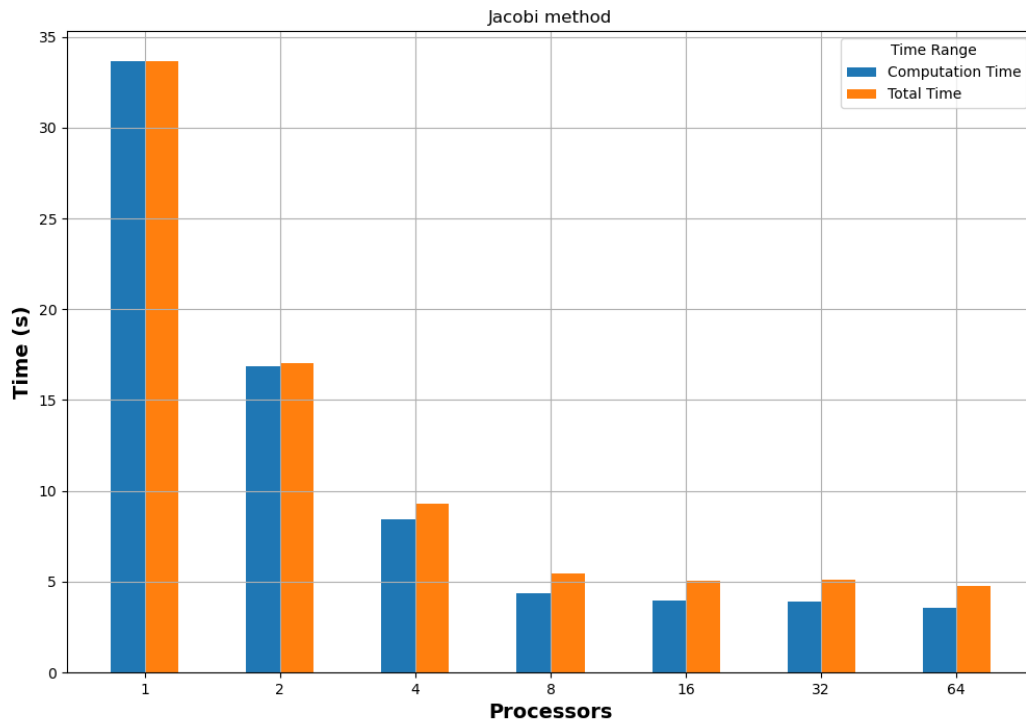
- Γραφικές Παραστάσεις



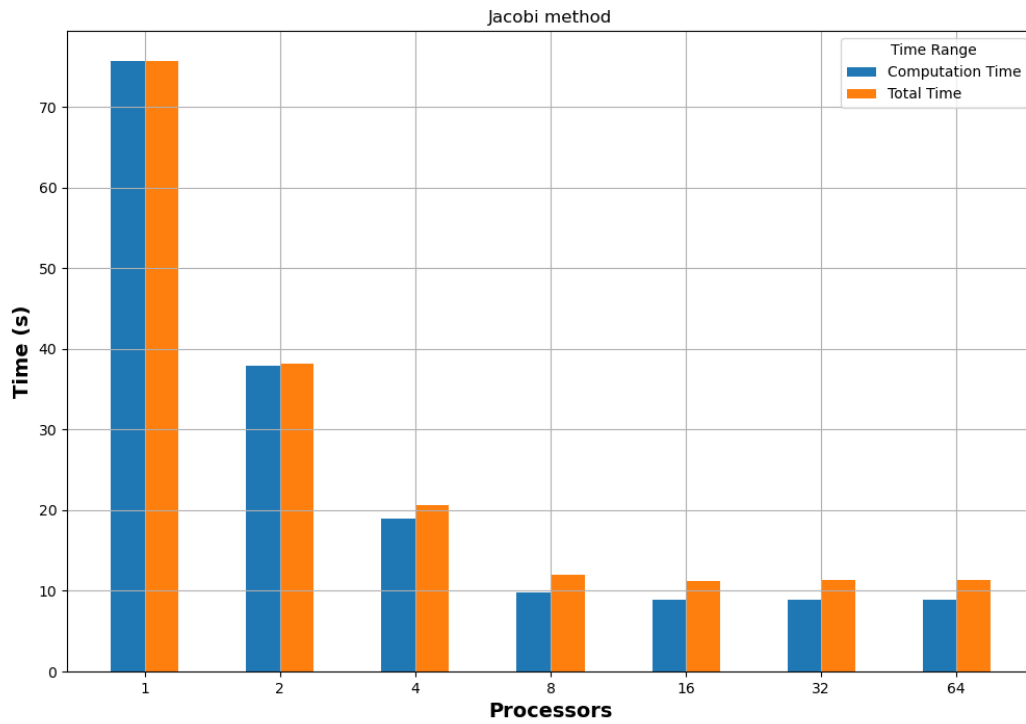
Heat Transfer MPI Time 2048x2048



Heat Transfer MPI Time 4096x4096



Heat Transfer MPI Time 6144x6144



Παρατηρούμε εν γένει μια βελτίωση στους χρόνους εκτέλεσης με την αύξηση των επεξεργαστών, ωστόσο η επίδοση παραμένει σταθερή μετά τους 8 επεξεργαστές για τα μεγαλύτερα μεγέθη πινάκων.