



Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής & Τηλεπικοινωνιών
Παράλληλα Υπολογιστικά Συστήματα
Σεπτέμβριος 2019

Παραλληλοποίηση Αλγορίθμου Προσομοίωσης Μεταφοράς Θερμότητας

Γιώργος Κατσογιάννης-Μεϊμαράκης
sdi1400065@di.uoa.gr

Γιάννης Χήρας
sdi1400225@di.uoa.gr

Περιεχόμενα

1	Εισαγωγή	1
2	Μεταγλώττιση & Εκτέλεση	1
2.1	Μεταγλώττιση	1
2.2	Εκτέλεση	1
3	Εργαλεία Ανάπτυξης Project	1
3.1	Version Control (Git/Github)	1
4	Οργάνωση Αρχείων & Φακέλων	2
4.1	Οργάνωση Αρχείων	2
5	Βελτιωμένο Πρόγραμμα MPI	2
5.1	Γενικά	2
5.2	Ανάλυση Βελτιώσεων	2
5.2.1	Διαμοιρασμός σε 2 Διαστάσεις	2
5.2.2	Άλλω Εξωτερικών Σημείων	2
5.2.3	Workflow Επικοινωνιών - Υπολογισμός	2
5.2.4	Πρώτα Receive - Μετά Send	2
5.2.5	Datatypes Σειρών - Στηλών	2
5.2.6	Υπολογισμός Γειτόνων	2
5.2.7	Αποφυγή Ifs για Επικοινωνίες Γειτόνων	2
5.2.8	Γείτονες σε ίδιο κόμβο	2
5.2.9	Persisten Communication	2
5.2.10	Δυναμικοί Πίνακες	2
5.2.11	11 ???	3
5.2.12	Compile με -O3	3
5.3	Χρόνοι	3
5.4	Speed-up	3
5.5	Efficiency	3
6	Βελτιωμένο Πρόγραμμα MPI με Σύγκλιση	3
6.1	Γενικά	3
6.2	Ανάλυση Βελτιώσεων	4
6.3	Χρόνοι	4
7	Βελτιωμένο Υβριδικό Πρόγραμμα MPI & OpenMP με Σύγκλιση	4
7.1	Γενικά	4
7.2	Ανάλυση Βελτιώσεων	4
7.3	Χρόνοι	4

1 Εισαγωγή

HEAT2D is based on a simplified two-dimensional heat equation domain decomposition. The initial temperature is computed to be high in the middle of the domain and zero at the boundaries. The boundaries are held at zero throughout the simulation. During the time-stepping, an array containing two domains is used; these domains alternate between old data and new data. At each time step, worker processes must exchange border data with neighbors, because a grid point's current temperature depends upon its previous time step value plus the values of the neighboring grid points.

Για κάθε τετράγωνο του πλέγματος η τιμή της θερμοκρασίας στο επόμενο time-step υπολογίζεται από την εξής εξίσωση:

$$\begin{aligned} u'[x][y] = & u[x][y] \\ & + c_x * (u[x+1][y] + u[x-1][y] - 2 * u[x][y]) \\ & + c_y * (u[x][y+1] + u[x][y-1] - 2 * u[x][y]) \end{aligned}$$

Όπου $c_x = c_y = 0.1$ σταθερές.

2 Μεταγλώττιση & Εκτέλεση

2.1 Μεταγλώττιση

Τα προγράμματα μεταγλωττίζονται με τις παρακάτω εντολές και παράγουν ένα εκτελέσιμο το οποίο δίνεται προς εκτέλεση στην ουρά του συστήματος.

Για τη μεταγλώττιση προγραμμάτων που θα χρησιμοποιηθούν για μετρήσεις, δίνουμε στον compiler το flag -O3 για τη βελτιστοποίηση του κώδικα σε χαμηλό επίπεδο.

- `mpi_heat2Dn.c`: Βελτιωμένος κώδικας MPI χωρίς σύγκλιση

```
mpicc -O3 mpi_heat2Dn.c -o mpi_heat2Dn.x
```

- `mpi_heat2Dn_conv.c`: Βελτιωμένος κώδικας MPI με σύγκλιση

```
mpicc -O3 mpi_heat2Dn_conv.c -o mpi_heat2Dn.x
```

- `mpi_heat2Dn_hybrid.c`: Βελτιωμένος υβριδικός κώδικας MPI + OpenMP με σύγκλιση

```
mpicc -O3 -fopenmp mpi_heat2Dn_hybrid.c -o mpi_heat2Dn.x
```

2.2 Εκτέλεση

Για την εκτέλεση του προγράμματος στο cluster του hellasgrid.gr χρησιμοποιούμε το script `pbs_script.sh`

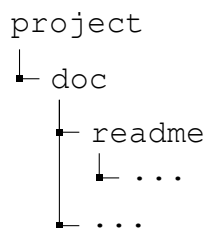
```
qsub pbs_script.sh
```

3 Εργαλεία Ανάπτυξης Project

3.1 Version Control (Git/Github)

Για την καλύτερη διαχείριση των εκδόσεων του κώδικα και των αλλαγών χρησιμοποιείται το πρόγραμμα `git` και η πλατφόρμα Github.

4 Οργάνωση Αρχείων & Φακέλων



Ο κώδικας οργανώνεται σε διαφορετικά αρχεία ανάλογα με το σκοπό και τη λειτουργικότητά του. Συγκεκριμένα στα εξής αρχεία:

4.1 Οργάνωση Αρχείων

5 Βελτιωμένο Πρόγραμμα MPI

5.1 Γενικά

5.2 Ανάλυση Βελτιώσεων

5.2.1 Διαμοιρασμός σε 2 Διαστάσεις

Χήρας

5.2.2 Άλλο Εξωτερικών Σημείων

Χήρας

5.2.3 Workflow Επικοινωνιών - Υπολογισμός

Χήρας

5.2.4 Πρώτα Receive - Μετά Send

Χήρας

5.2.5 Datatypes Σειρών - Στηλών

Κάτσο

5.2.6 Υπολογισμός Γειτόνων

Κάτσο

5.2.7 Αποφυγή Ifs για Επικοινωνίες Γειτόνων

Κάτσο

5.2.8 Γείτονες σε ίδιο κόμβο

Κάτσο

5.2.9 Persisten Communication

Χήρας

5.2.10 Δυναμικοί Πίνακες

Χήρας

5.2.11 11 ???

5.2.12 Compile με -O3

Χήρας Χρειάζεται να μπει εδώ; Το αναφέρω στη μετταγλώττιση

5.3 Χρόνοι

Nodes	Tasks	80×64	160×128	320×256	640×512	1280×1024	2560×2048	5120×4096
1	1	0.003476	0.013983	0.116129	0.464293	1.907506	7.492858	30.003448
1	4	0.001333	0.004159	0.014871	0.117549	0.472859	1.935913	7.563494
2	16	0.118759	0.153217	0.013015	0.012097	0.098547	0.053514	2.033478
8	64	0.281022	0.414862	0.061213	0.301717	0.326839	0.316296	0.527302
16	128	0.041406	0.131406	0.433416	0.321896	0.487963	0.296483	0.274705
20	160	0.071904	0.089528	0.596713	0.487695	0.521694	0.273331	0.205910

Table 1: Χρόνοι Εκτέλεσης mpi_heat2Dn

5.4 Speed-up

Nodes	Tasks	80×64	160×128	320×256	640×512	1280×1024	2560×2048	5120×4096
1	1	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1	4	2.61	3.36	7.81	3.95	4.03	3.87	3.97
2	16	0.03	0.09	8.92	38.38	19.36	140.02	14.75
8	64	0.01	0.03	1.90	1.54	5.84	23.69	56.90
16	128	0.08	0.11	0.27	1.44	3.91	25.27	109.22
20	160	0.05	0.16	0.19	0.95	3.66	27.41	145.71

Table 2: Speed-up για το mpi_heat2Dn

5.5 Efficiency

Nodes	Tasks	80×64	160×128	320×256	640×512	1280×1024	2560×2048	5120×4096
1	1	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1	4	2.61	3.36	7.81	3.95	4.03	3.87	3.97
2	16	0.01	0.05	4.46	19.19	9.68	70.01	7.38
8	64	0.00	0.00	0.24	0.19	0.73	2.96	7.11
16	128	0.01	0.01	0.02	0.09	0.24	1.58	6.83
20	160	0.00	0.01	0.01	0.05	0.18	1.37	7.29

Table 3: Efficiency για το mpi_heat2Dn

6 Βελτιωμένο Πρόγραμμα MPI με Σύγκλιση

6.1 Γενικά

Κάτσο

6.2 Ανάλυση Βελτιώσεων

Κάτσο

6.3 Χρόνοι

Κάτσο

7 Βελτιωμένο Υβριδικό Πρόγραμμα MPI & OpenMP με Σύγκλιση

7.1 Γενικά

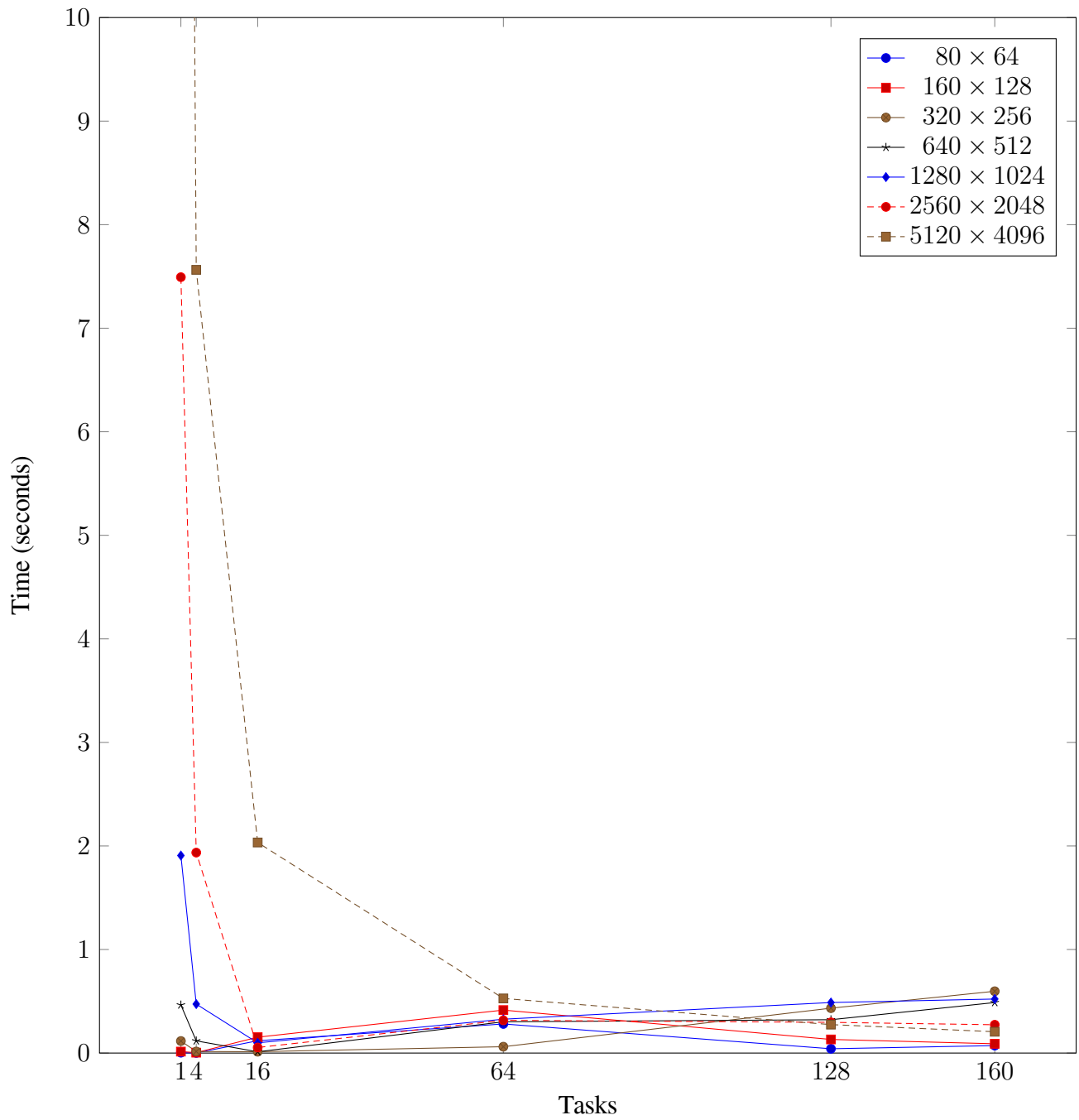
Κάτσο

7.2 Ανάλυση Βελτιώσεων

Κάτσο

7.3 Χρόνοι

Κάτσο

Figure 1: Χρόνοι Εκτέλεσης `mri_heat2Dn` προς πλήθος Tasks

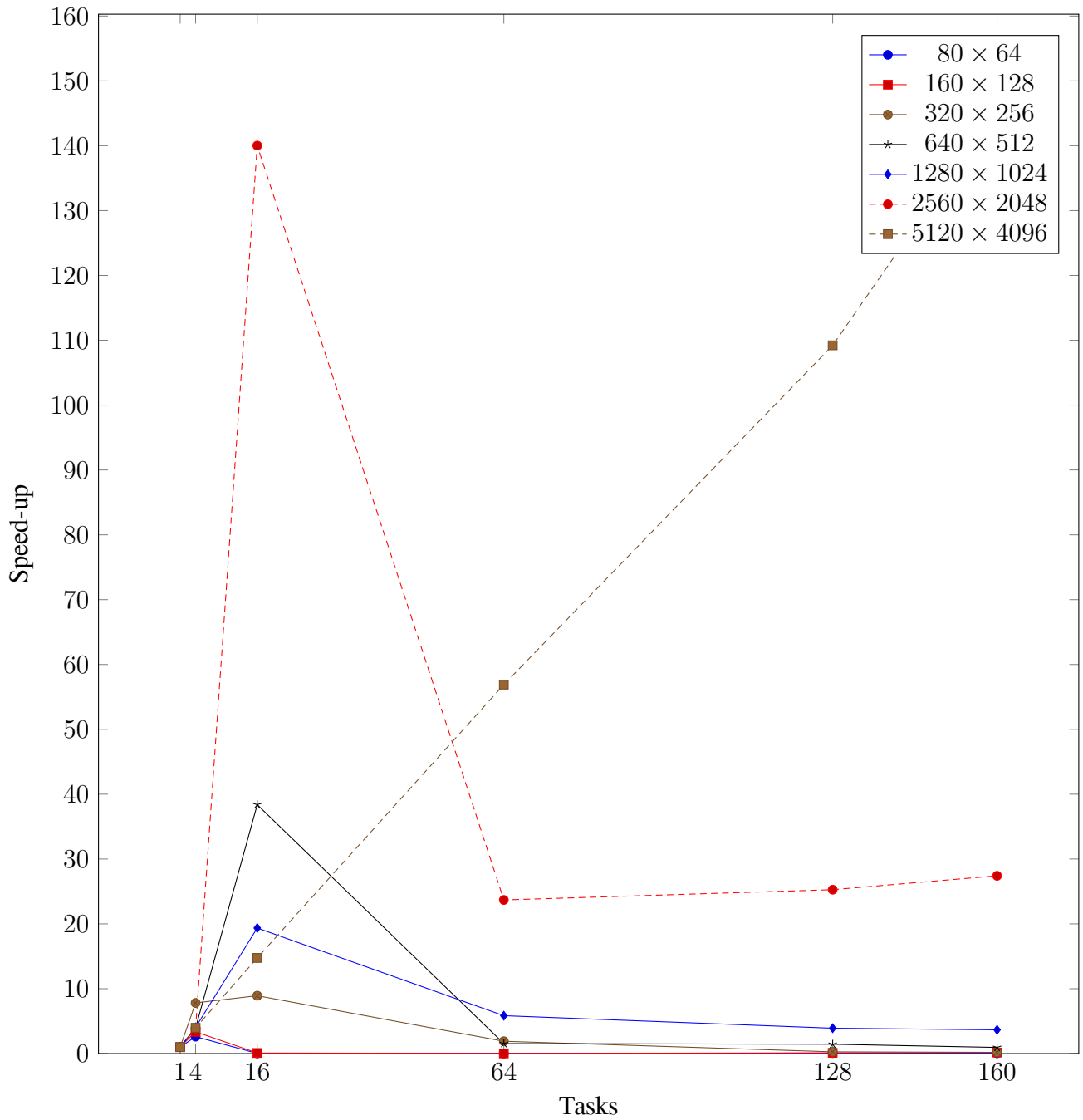


Figure 2: Speed-up του mpi_heat2Dn προς πλήθος Tasks

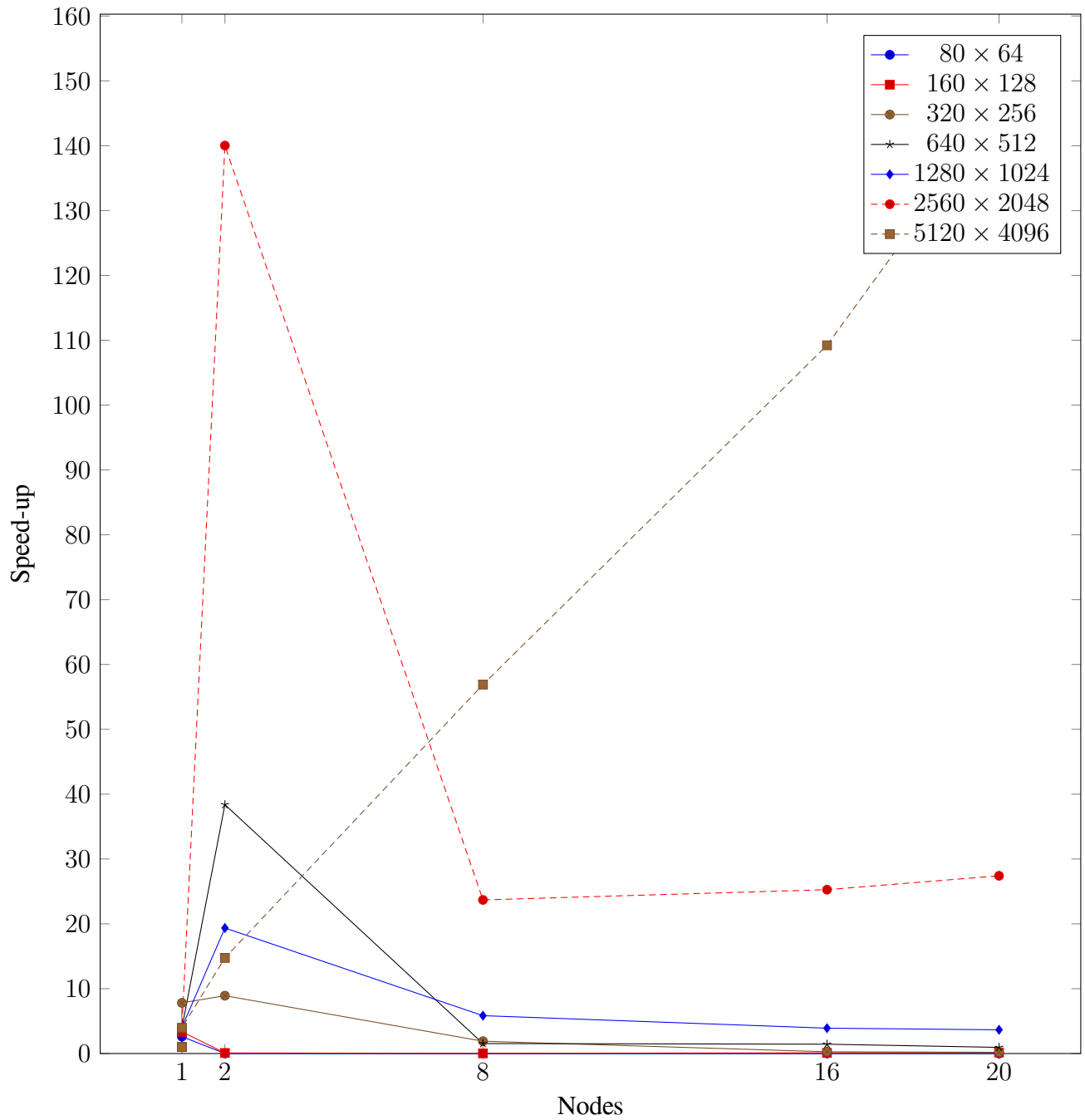


Figure 3: Efficiency του mpi_heat2Dn προς πλήθος Nodes