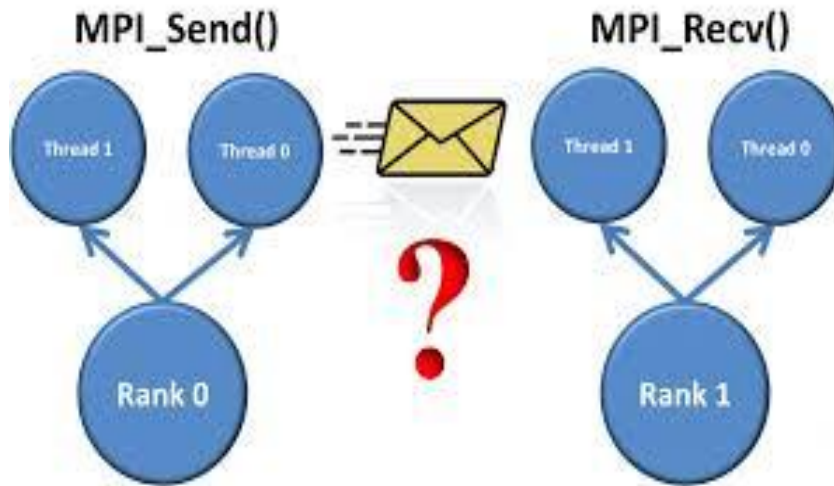


# Παράλληλα και Διανεμημένα Συστήματα

---

2η εργασία



Χρήση του Message Passing Interface.

Στερζεντσένκο Βλαδίμηρος

AEM:7777 email:vladster@auth.gr

Θεσσαλονίκη 13/02/2015

Στην εργασία αυτή καλούμαστε να υλοποιήσουμε παράλληλο αλγόριθμο ταξινόμησης με χρήση του αλγόριθμου ταξινόμησης bitonic sort.

Στο πρόβλημά μας, η κάθε διεργασία (ή διεργασίες το σύνολο) έχει έναν δικό της πίνακα με μήκος που είναι δύναμη του 2 (έστω  $N$ ). Στόχος μας είναι να κατασκευάσουμε μια bitonic ακολουθία, δηλαδή οι πρώτες  $n/2$  διεργασίες να είναι σε αύξουσα σειρά και τοπικά (δηλαδή κάθε πίνακας ταξινομημένος) και συνολικά (δηλαδή η πρώτη διεργασία να έχει μικρότερα στοιχεία από την δεύτερη κ.ο.κ.).

- Καταρχάς χρησιμοποιούμε μια global (group) μεταβλητή η οποία μας δείχνει ανα πόσες διεργασίες είναι ταξινομημένες μονότονα.
- Για επικοινωνία μεταξύ μηχανημάτων χρησιμοποιούμε μια μεταβλητή (handshake) η οποία προκύπτει από την πράξη  $\text{rank}^{(1 \ll (\log_2(\text{group}) - 1))}$ , η οποία αλλάζει το  $1 \ll \log_2(\text{group}) - 1$  bit του rank (στο rank 00 η μεταβλητή handshake με group 2 είναι 01 και vice versa).
- Το κάθε μηχάνημα γνωρίζει την φορά της ταξινόμησης από τον τύπο  $!(1 \ll \log_2(\text{group}))$  (τα rank 00 και 01 ταξινομούνται κατα αύξουσα σειρά και τα 10 11 κατα φθίνουσα όταν το group είναι 2 και τα αύξουσα όταν group είναι 4).

- Η κάθε διαδικασία στέλνει χωρίς blocking διαδικασία στο ζευγάρι του, το κάτω μισό του πίνακα αν ( $rank < handshake$ ) και το πάνω μισό αν ( $rank > handshake$ ) και λαμβάνει με blocking διαδικασία σε έναν τοπικό buffer κάνει συγκρίσεις αλλαγές με το εκάστοτε μισό του τοπικού πίνακα (οι αλλαγές εξαρτώνται από το αν είναι το κάτω μισό του πίνακα ή το πάνω και από την φορά της ταξινόμησης) και στέλνει πίσω τον επεξεργασμένο buffer.

### **Διαδικασία ταξινόμησης**

Σε κάθε βήμα του προγράμματος μας προσπαθούμε να κατασκευάσουμε bitonic ακολουθία. Ξεκινάμε με  $group=2$  και καλούμε την συνάρτηση `data_transfer()` η οποία χειρίζεται την επικοινωνία και επεξεργασία των στοιχείων όπως περιγράφεται παραπάνω. Η `data_transfer()` καλεί αναδρομικά τον εαυτό της μέχρι το  $group=2$ . Στην τελευταία κλήση της `data_transfer()` για το κάθε  $group$  ταξινομούμε τοπικά με χρήση `qsort()` με φορά που υπολογίζεται όπως περιγράφεται παραπάνω. Έτσι πετυχαίνουμε την ταξινόμηση με την επιθυμητή φορά ανα  $group$  διαδικασίες. Όταν φτάσουμε να καλέσουμε την `data_transfer()` για  $group=n$ , δηλαδή θέλουμε να ταξινομήσουμε όλα κατά αύξουσα.

## Διαδικασία ελέγχου

Στέλνουμε από κάθε διαδικασία έναν πίνακα temp στην διαδικασία με rank 0 με,

*ελάχιστο του τοπικού πίνακα*  
temp[3]= *μέγιστο του τοπικού πίνακα*  
*1 αν είναι ταξινομημένος, αλλιώς 0*

Αφού τα μαζέψουμε,ελέγχουμε το μέγιστο κάθε τοπικού πίνακα με το ελάχιστο του επόμενου και για τον αν είναι ταξινομημένος.

## Πίνακες μετρήσεων

Μήκος  
πίνακα  
(2<sup>^</sup>)

/nodes,ppn

2n1p

4n2p

15	2,676433	13	4,090772
16	3,388296	14	5,559624
17	6,049308	15	5,599362
18	6,225308	16	7,739058
19	6,465270	17	7,788186
20	6,624982	18	7,861965

4n4p

4n8p

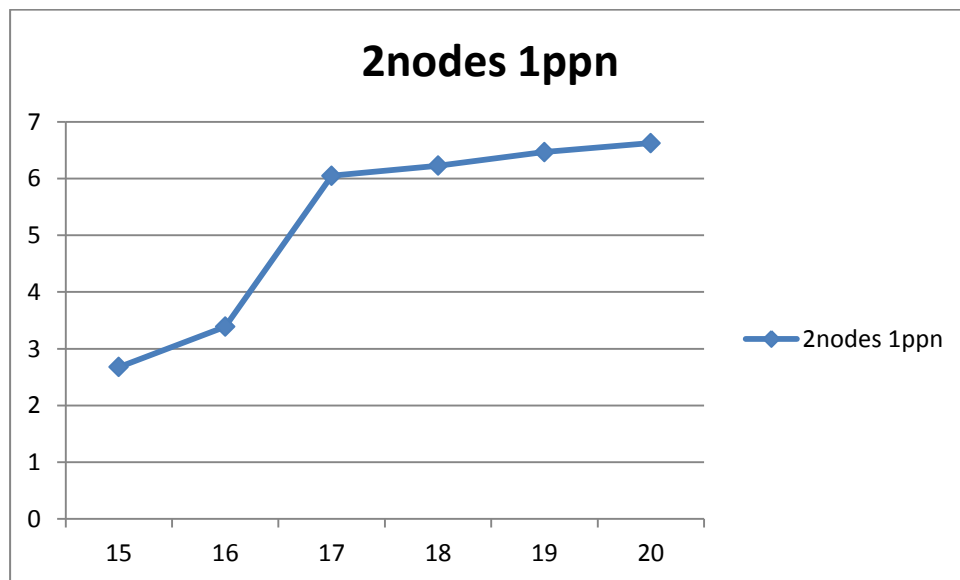
4n32p

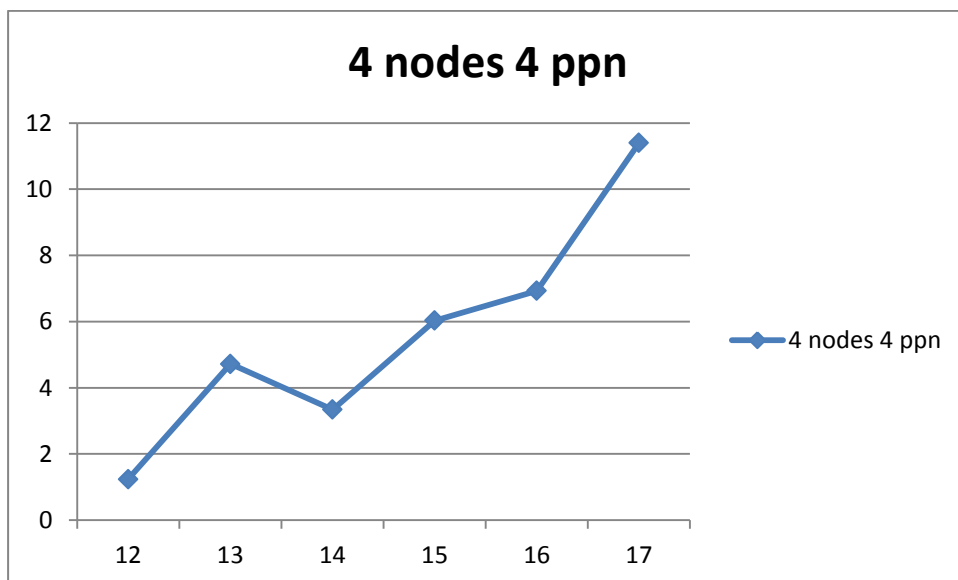
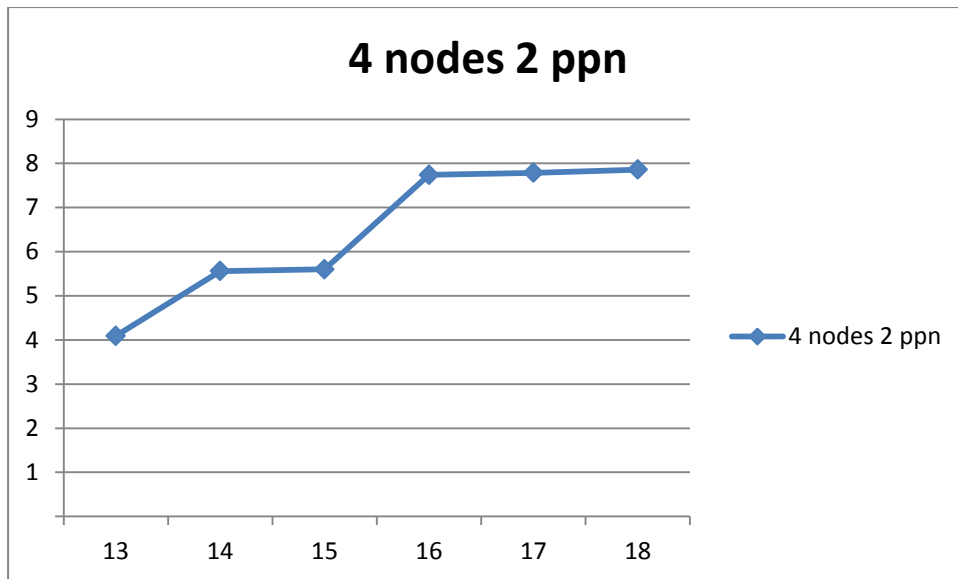
12	1,239105	11	3,411866	10	1,670264
13	4,721058	12	5,337195	11	2,785124
14	3,342839	13	7,138086	12	7,037779
15	6,028495	14	10,092425	13	6,964464
16	6,932330	15	12,759436	14	7,838230
17	11,399983	16	13,015762	15	7,052976

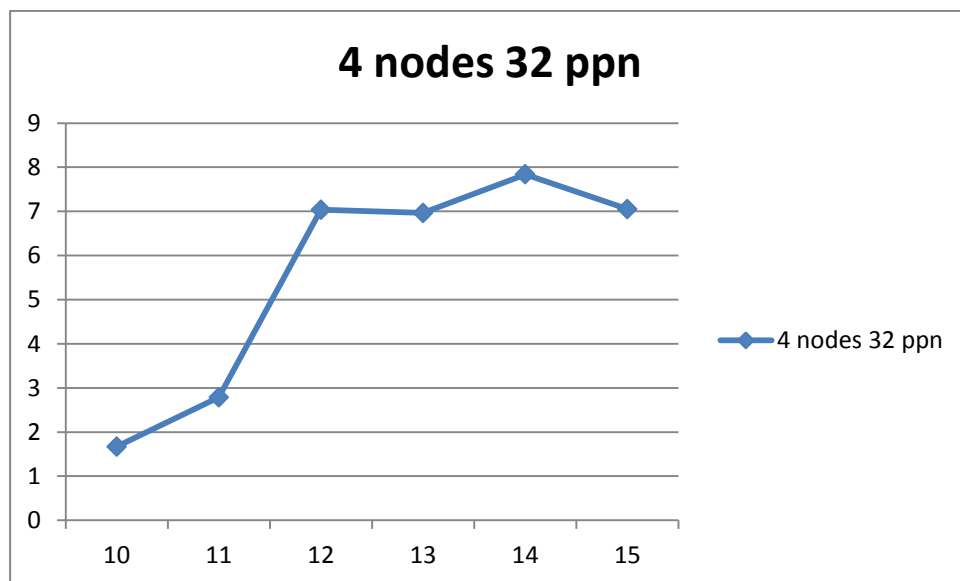
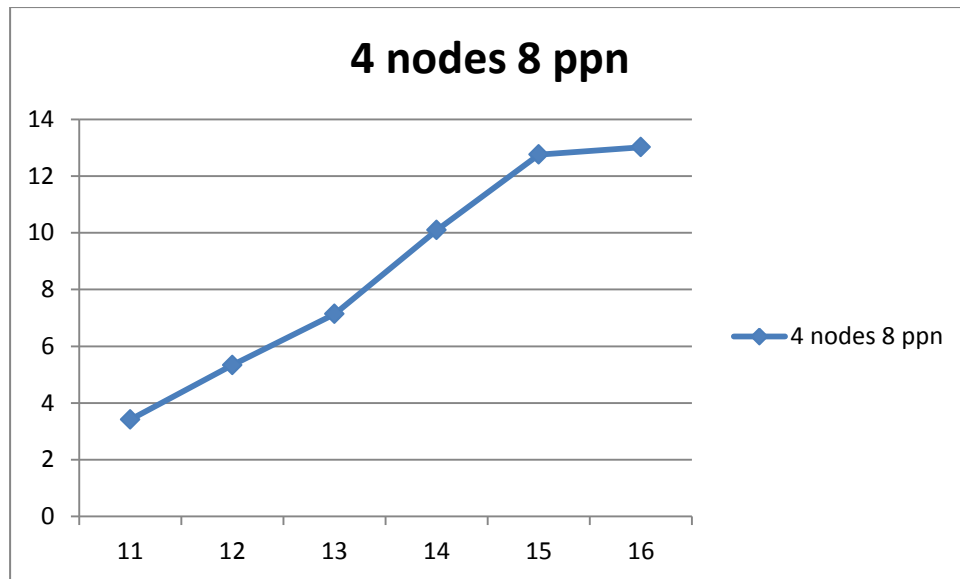
## Διαγράμματα

Στον κατακόρυφο άξονα έχουμε το λόγο, χρόνος εκτέλεσης παράλληλου κώδικα προς το αντίστοιχο χρόνο του σειριακού.

Στον οριζόντιο άξονα έχουμε το μήκος του πίνακα προς ταξινόμηση (δύναμη του 2).







## Αποτίμηση

Ο παράλληλος κώδικας δείχνει σημαντική βελτίωση στην διαδικασία της ταξινόμησης. Ωστόσο αυξάνοντας τον αριθμό των processes και μειώνοντας το μήκος του πίνακα δεν βλέπουμε σημαντική βελτίωση, καθώς στα πολλά processes έχουμε καθυστερήσεις λόγω των πολλών μεταβιβάσεων των στοιχείων των πινάκων μεταξύ των processes.

Για πολύ μεγαλύτερα προβλήματα, ο κώδικας δείχνει πάρα πολύ σημαντική βελτίωση, καθώς ο χρόνος επικοινωνίας γίνεται αμελητέος σε σχέση με τον χρόνο που απαιτείται για την ταξινόμηση.