Παράλληλα και Διανεμημένα Συστήματα
Εργασία 4
PageRank

Ζαμπόκας Γεώργιος
ΑΕΜ: 7173
Ημερομηνία: 29/1/2014

Λογική Υλοποίησης

Ο αλγόριθμος υπολογισμού της σημαντικότητας των κόμων του διαδικτύου PageRank γίνεται αρκετά πολύπλοκος όσο αυξάνεται ο αριθμός των κόμβων τον οποίο καλούμαστε να αναλύσουμε και όσο αυξάνεται α αριθμός των συνδέσεων μεταξύ τους.
Για μια ενδεικτική υλοποίηση υποθέσαμε ότι έχουμε το μέγιστο 1 εκατομμύριο κόμβους, όπου ο καθένας όμως δεν μπορεί να έχει πάνω απο 15 συνδέσεις με άλλους κόμβους. Τα δεδομένα αυτα βρίσκονται στο αρχείο G, στην εξής μορφή. Κάθε κόμβος περιέχει τον αύξοντα αριθμό των κόμβων που συνδέεται και όπου υπάρχει 0 σημαίνει ότι δεν έχουμε σύνδεση.

Ο σειριακός τρόπος υλοποίησης χρησιμοποιεί τον τύπο που δίνεται για τον υπολογισμό των πιθανοτήτων και δεν υπάρχει λόγος να αναλυθεί παιρεταίρω.

Στον παράλληλο τρόπο το σύνολο των κόμβων διαιρείται με τον αριθμό των threads που επιλέγεται κάθε φορά. Εφαρμόζεται πάλι ο ίδιος τύπος απλά η διαδικασία γίνεται παράλληλα και σαρώνονται τόσα σημεία κάθε στιγμή, όσα και τα threads. Τα threads συγχρονίζονται καθώς δεν γίνεται να εκτελεστούν ανεξάρτητα αφού στις πράξεις που εκετελούν εμπεριέχονται μεταβλητές που επηρεάζονται από άλλα threads στο ίδιο βήμα.

Ανάλυση Κώδικα

dataset: Προκύπτει στην μορφή που περιγράφηκε ο κόμβος, καθώς επίσης και τα διανύσματα P, E με απλό τυχαίο τρόπο. Περιέχεται και το script από MATLAB.

pagerankp.c: Ο κώδικας παράλληλης υλοποίησης. Αρχικά δηλώνονται οι μεταβλητές των δεδομένων που επρόκειτο να χρησιμοποιηθούν ως global ώστε να έχουν πρόσβαση τα threads. Επίσης κατασκευάζεται η δομή που θα περιέχει τα δεδομένα που θα έχουν τα threads.
Πέρα από την δήλωση μνήμης, γίνεται καταμερισμός εργασίας στα threads, δηλαδή πόσα σημεία θα αναλάβει το καθένα για να υπολογίσει την πιθανότητά τους.
Αξίζει να σημειωθεί ότι δεν χρησιμοποιήθηκαν barriers αλλά ο γνωστός και από την προηγούμενη εργασία τρόπος με την συνάρτηση pthread_join(). Σε κάθε βήμα προς την σύγκλιση δημιουργούνται threads, τόσα όσα επιλέχθηκαν και εκτελούν ένα βήμα υπολογισμών. Όταν και το τελευταίο από αυτά φτάσει στην σύγκλιση της πιθανότητας τότε

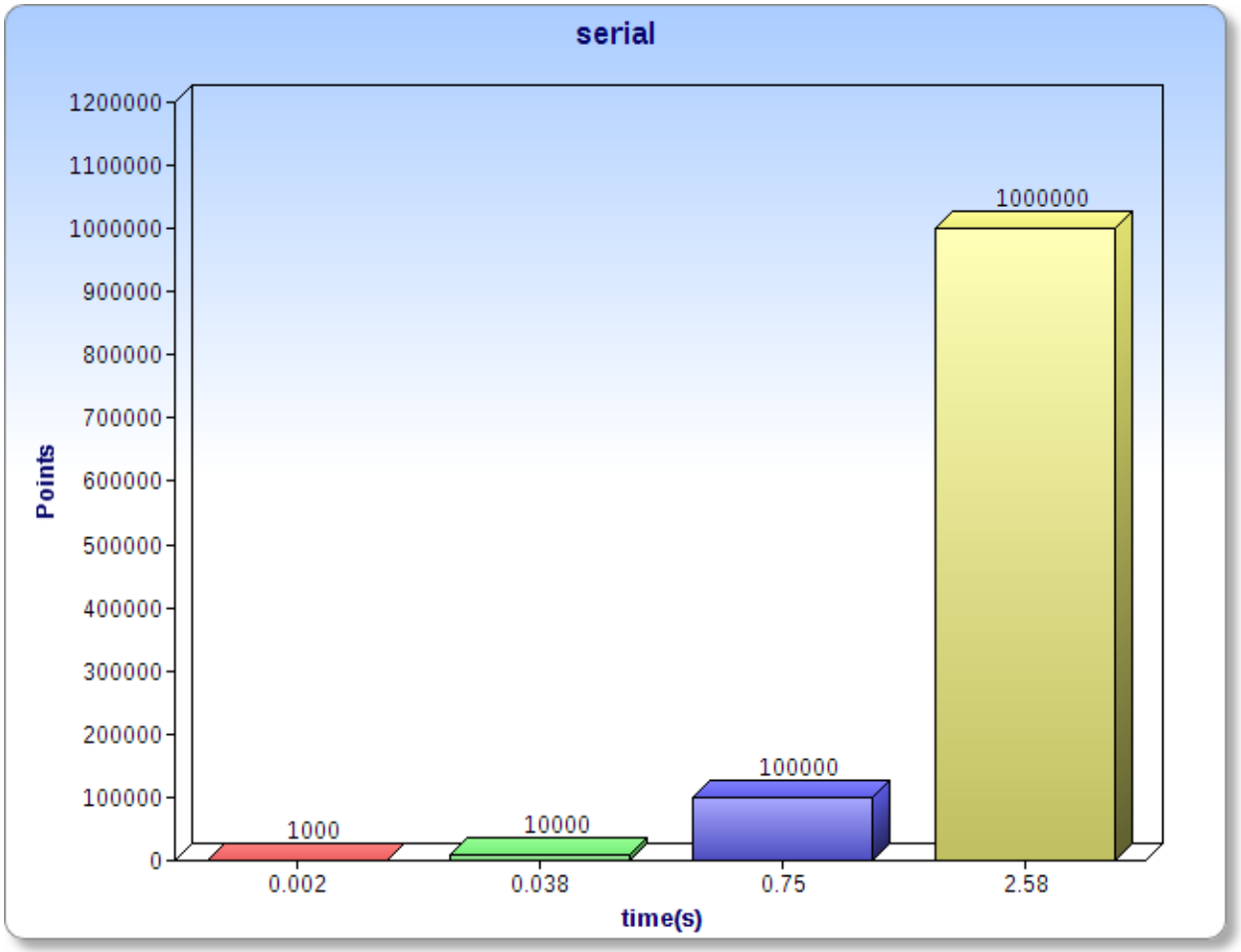
σταματάει ο αλγόριθμος και έχουμε καταλήξει στις τελικές τιμές. Η pthread_join() καλεί την παρακάτω συνάρτηση για τον υπολογισμό των πιθανοτήτων.

calcPageRank: ¨Οπως αναφέραμε και πριν αυτή η συνάρτηση είναι υπεύθυνη για την εκτέλεση των πράξεων. Έχοντας χωρίσει το dataset σε ισάριθμα μέρη με τα threads υπολογίζει τρέχοντας σε κάθε thread παράλληλα τις τιμές P(t+1) για κάθε σημείο του κόμβου. Εάν έχουμε σύγκλιση τότε σταματάει τους υπολογισμούς και επιστρέφει.

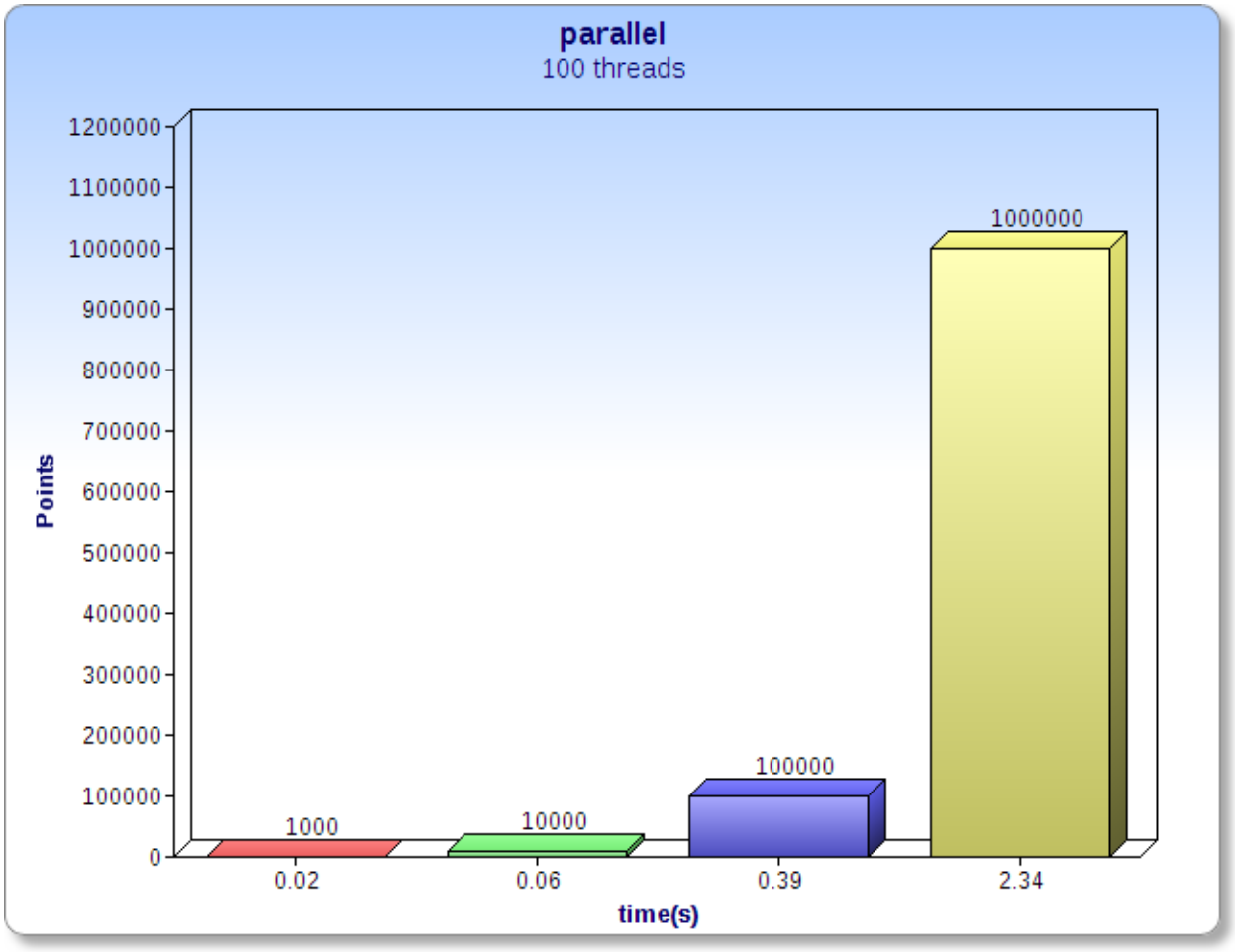
Χρόνοι εκτέλεσης

Εδώ παρουσιάζουμε συνοπτικά τα διαγράμματα με κάποιους ενδεικτικούς χρόνους εκτέλεσης για διάφορα μεγέθη δεδομένων.

Σειριακά:



και παράλληλα:



Συμπέρασμα: Η παράλληλη υλοποίηση παρ' ότι φαίνεται να υστερεί σε ταχύτητα για μικρό όγκο δεδομένων, προσφέρει μια ικανοποιητική επιτάχυνση για δεδομένα μεγάλου όγκου, η οποία είναι γύρω στο 10%.