

# Αλγόριθμος Pagerank με παράλληλη υλοποίηση της μεθόδου Gauss-Seidel

Νικόλαος Κατωμέρης

AEM: 8551

ngkatomer@auth.gr

Παράλληλα και διανεμημένα συστήματα.

Τμήμα ηλεκτρολόγων μηχανικών και μηχανικών υπολογιστών, ΑΠΘ

## Περίληψη

Στην παρούσα αναφορά περιγράφεται η διαδικασία υλοποίησης και παραλληλοποίησης του αλγορίθμου Gauss-Seidel για τον υπολογισμό του διανύσματος Pagerank ενός γράφου. Επίσης, παρουσιάζονται αποτελέσματα της εκτέλεσης των υλοποιήσεων σε έναν ενδεικτικό γράφο. Για τον κώδικα που αναπτύχθηκε ανατρέξτε στην τελευταία ενότητα.

## Pagerank

Η βασική ιδέα του αλγορίθμου Pagerank είναι η αξιολόγηση της κάθε σελίδας του διαδικτύου με βάση τον αριθμό των συνδέσμων προς αυτήν και την ποιότητα αυτών, η οποία καθορίζεται από την αξιολόγηση των σελίδων που τους περιέχουν.

Το αποτέλεσμα της παραπάνω αξιολόγησης ταυτίζεται με το ποσοστό του χρόνου στον οποίο ένας τυχαίος περιπατητής θα βρισκόταν στην εκάστοτε σελίδα, αν ξεκινώντας από κάποια τυχαία σελίδα, ακολουθούσε τυχαία κάποιον από τους συνδέσμους της, μετά το ίδιο για την επόμενη κι ούτω καθ' εξής.

Η παραπάνω διαδικασία μπορεί να προσομοιωθεί με μία αλυσίδα Markov της οποίας ο πίνακας μεταβάσεων αποτελείται από στοιχεία που δείχνουν την πιθανότητα μετάβασης από μία σελίδα σε μία άλλη.

Έστω  $A$ , ο  $N \times N$  πίνακας μεταβάσεων της αλυσίδας για  $N$  σελίδες του διαδικτύου. Αν  $j \rightarrow i$  σημαίνει ότι η σελίδα  $j$  περιέχει σύνδεσμο προς τη σελίδα  $i$ , ο  $A$  ορίζεται αρχικά ως εξής:

$$A(i, j) = \begin{cases} 1, & \text{αν } j \rightarrow i \\ 0, & \text{αλλιώς} \end{cases}$$

Επειδή όμως ο παραπάνω πίνακας θέλουμε να εκφράζει την πιθανότητα ο περιπατητής να πάει από τη σελίδα  $j$  στη σελίδα  $i$  σε ένα βήμα, είναι απαραίτητο το άθροισμα των στοιχείων κάθε στήλης του πίνακα να ισούται με 1.

Οπότε, αφού λάβουμε τον πίνακα με τους συνδέσμους των σελίδων, διαιρούμε όλα του τα στοιχεία με το άθροισμα των στοιχείων της εκάστοτε στήλης  $j$ , έστω  $L_j$ , και παίρνουμε τον «στοχαστικοποιημένο» πίνακα  $A_s$ :

$$A_s(i, j) = \begin{cases} \frac{1}{L_j}, & \text{αν } j \rightarrow i \\ 0, & \text{αλλιώς} \end{cases}$$

Επιπλέον, καθώς ο περιπατητής ξεκινά από μία σελίδα, είναι πολύ πιθανό, ακολουθώντας συνεχώς συνδέσμους να είναι αδύνατο να μεταβεί σε όλες τις σελίδες του γράφου. Το γεγονός αυτό, συν το γεγονός ότι ένας χρήστης του διαδικτύου δεν ακολουθεί μόνο συνδέσμους, αλλά «τηλεμεταφέρεται» κιόλας σε άλλες σελίδες, καθιστά τον παραπάνω πίνακα ανεπαρκή για την προσομοίωση της τυχαίας διαδρομής ενός χρήστη του διαδικτύου.

Έτσι, εισάγεται στον αλγόριθμο μια σταθερά «τηλεμεταφοράς» που δηλώνει την πιθανότητα ο περιπατητής να μεταβεί σε κάποια τυχαία σε-

λίδα του διαδικτύου χωρίς να υπάρχει κάποιος σύνδεσμος προς αυτήν στη σελίδα που βρίσκεται.

Η σταθερά αυτή, έστω  $c$ , μπορεί να επιλεγεί ως όρισμα στο πρόγραμμα υπολογισμού του διανύσματος pagerank που υλοποιήθηκε. Στην περίπτωση που δεν δοθεί, θα έχει την τιμή 0.15 που έχει δειχθεί ότι αποτελεί καλή επιλογή [1].

Με την εισαγωγή της παραπάνω σταθεράς, ο πίνακας μεταβάσεων πλέον θα είναι:

$$A_{final} = \frac{c}{N} \cdot \mathbf{1}_{N \times N} + (1 - c)A_n$$

Στην παραπάνω σχέση, η σταθερά  $c$  διαιρείται με τον αριθμό των γραμμών  $N$  έτσι ώστε οι στήλες και του τελικού πίνακα  $A_{final}$  να έχουν στοιχεία με άθροισμα 1.

Με τη χρήση του παραπάνω πίνακα είναι δυνατό να υπολογιστεί ένα διάνυσμα  $N \times 1$  πολλαπλασιάζοντάς τον πίνακα με ένα αρχικό διάνυσμα -πχ ένα ομοιόμορφο- αρκετές φορές, έως ότου το αποτέλεσμα συγκλίνει. Το διάνυσμα αυτό αποτελεί ιδιοδιάνυσμα του πίνακα  $A$  για την ιδιοτιμή 1 και αποτελεί το ζητούμενο διάνυσμα με την αξιολόγηση της κάθε σελίδας (κόμβου).

Τα παραπάνω δεν θα ισχύουν αν ο πίνακας  $A_s$  περιέχει σελίδες χωρίς καμία σύνδεση. Σε αυτή την περίπτωση, επιθυμούμε ο περιπατητής να τηλεμεταφερθεί σε μία τυχαία σελίδα. Οπότε, προτού υπολογιστεί ο  $A_{final}$ , είναι απαραίτητο οι μηδενικές στήλες του  $A_s$  να αντικατασταθούν με νέες στήλες που αποτελούνται από στοιχεία  $\frac{1}{N}$ . Έστω  $A_n$  ο πίνακας που δημιουργείται.

Με όλα τα παραπάνω δεδομένα, αποδεικνύεται ότι ο τελικός πίνακας θα έχει μοναδική ιδιοτιμή ένα και το πρόβλημα πλέον έγκειται στην εύρεση του ιδιοδιανύσματος της.

Για την εύρεση του ιδιοδιανύσματος, όπως ζητείται, χρησιμοποιείται η μέθοδος Gauss-Seidel.

## Gauss Seidel

Η μέθοδος Gauss-Seidel είναι μια αναδρομική μέθοδος επίλυσης γραμμικών συστημάτων. Ο αναδρομικός αλγόριθμός της μεθόδου για την επίλυση ενός γραμμικού συστήματος της μορφής:

$$A\vec{x} = \vec{b}$$

είναι για το  $i$ -στο στοιχείο στο  $k + 1$  βήμα:

$$x_i(k+1) = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij}x_j(k+1) - \sum_{j=i+1}^N A_{ij}x_j(k) \right)$$

Όπως φαίνεται, ο αλγόριθμος αυτός μοιάζει καθαρά σειριακός αφού ο υπολογισμός του κάθε στοιχείου σε κάθε βήμα εξαρτάται από τον υπολογισμό των νέων τιμών των στοιχείων πριν απ' αυτό. Για περισσότερες λεπτομέρειες σχετικά με τη μέθοδο αυτή και τα κριτήρια σύγκλισης της ανατρέξτε στον Saad [2].

Στην περίπτωσή μας, για την εύρεση της pagerank με τη μέθοδο Gauss-Seidel, θα πρέπει πρώτα να το εκφράσουμε σαν σύστημα γραμμικών εξισώσεων. Όπως προαναφέρθηκε το διάνυσμα Pagerank αποτελεί ιδιοδιάνυσμα του πίνακα  $A$  για την ιδιοτιμή 1. Επομένως, ισχύει ότι:

$$A_{final}\vec{x} = \vec{x} \Rightarrow (I - A_{final})\vec{x} = 0 \Rightarrow \left( I - \frac{c}{N} \cdot \mathbf{1}_{N \times N} - (1 - c)A_n \right) \vec{x} = 0$$

Όμως, το  $\mathbf{1}_{N \times N} \cdot \vec{x}$  είναι πάντα ίσο με  $\mathbf{1}_{N \times 1}$  επειδή το άθροισμα των στοιχείων του  $\vec{x}$ , ως άθροισμα όλων των πιθανοτήτων, εκφράζει την πιθανότητα ο περιπατητής να βρίσκεται σε μια οποιαδήποτε σελίδα. Έτσι, το σύστημά μας παίρνει τη μορφή:

$$(I - (1 - c)A_n)\vec{x} = \frac{c}{N}\mathbf{1}_{N \times 1} \quad (1)$$

που μπορεί να λυθεί αναδρομικά με τη μέ-

θοδο Gauss-Seidel, αρκεί διασφαλιστεί ότι σε κάθε βήμα  $\mathbf{1}_{N \times N} \cdot \vec{x}_k = \mathbf{1}_{N \times 1}$ , δηλαδή ότι το άθροισμα των στοιχείων του  $\vec{x}_k$  παραμένει ένα.

## Δεδομένα στη μνήμη

Το πρόβλημα με την Εξ. (1) είναι ότι ο πίνακας  $(\mathbf{I} - (1 - c)\mathbf{A}_n)$  δεν είναι αραιός καθώς ο  $\mathbf{A}_n$ , όπως αναφέρθηκε νωρίτερα, έχει γεμάτες τις στήλες με τις σελίδες που στον  $\mathbf{A}_s$  δεν είχαν καμία σύνδεση. Επίσης, το σύνολο αυτών των σελίδων στο διαδίκτυο είναι πολύ μεγάλο[3], με αποτέλεσμα ο  $\mathbf{A}_n$  να απαιτεί πολύ περισσότερο χώρο στη μνήμη από τον  $\mathbf{A}_s$ .

Έχουν προταθεί διάφορες λύσεις για την αποφυγή αυτού του φαινομένου, όπως για παράδειγμα να προστεθεί σ' όλες αυτές τις σελίδες σύνδεσμος προς τον εαυτό τους. Μια άλλη πρόταση, η οποία και προτιμήθηκε, χρησιμοποιεί μόνο τον πίνακα  $\mathbf{A}_s$  ο οποίος είναι αραιός και καταλήγει στην ίδια ακριβώς λύση με την Εξ. (1). Όπως αποδεικνύεται [4], η λύση της Εξ. (1) ισοδυναμεί με  $\frac{\vec{y}}{\|\vec{y}\|_1}$ , όπου  $\vec{y}$  η λύση της:

$$(\mathbf{I} - (1 - c)\mathbf{A}_s)\vec{y} = \frac{1}{N} \cdot \mathbf{1}_{N \times 1} \quad (2)$$

Η οποία απαιτεί την αποθήκευση μόνο του πίνακα  $\mathbf{A}_s$ . Δεδομένου ότι ο πίνακας αυτός είναι πολύ αραιός, αλλά και λόγω του γεγονότος ότι η μέθοδος gauss-seidel έχει, σε κάθε της βήμα και για κάθε κόμβο, πολλαπλασιασμούς στοιχείων μίας γραμμής του  $\mathbf{A}_s$ , επιλέχθηκε η χρήση της δομής «Compressed Row Storage (CRS)».

Η δομή αυτή επιτρέπει την αποθήκευση πινάκων σε χώρο στη μνήμη ανάλογο με τα μη μηδενικά στοιχεία αυτών. Επιπλέον, διατρέχοντας τον πίνακα ανά γραμμή, όλα τα στοιχεία της γραμμής βρίσκονται συνεχόμενα στη μνήμη καθιστώντας τη δομή αυτή φιλική προς την μνήμη cache των επεξεργαστών.

## Σειριακός αλγόριθμος

Με βάση τα παραπάνω, υλοποιήθηκε το σειριακό πρόγραμμα υπολογισμού της pagerank. Συνοψίζεται ως εξής:

1. Επιλογή γράφου δεδομένων εισόδου (σύνδεσμοι) και προαιρετικά συγκεκριμένου αριθμού επαναλήψεων ή σταθερών τηλεμεταφοράς ( $c$ ) και σύγκλισης ( $E$ ). Οι προεπιλεγμένες τιμές είναι:

$$c_{def} = 0.15, E_{def} = 1e - 12$$

2. Φόρτωση στη μνήμη των δεδομένων εισόδου σε δομή CRS. Τα δεδομένα αυτά αποτελούν τον αρχικό πίνακα  $\mathbf{A}$ .
3. Μετατροπή του  $\mathbf{A}$  στον  $\mathbf{A}_s$  και πολλαπλασιασμός του με την σταθερά  $-(1 - c)$ . Έστω  $a_{ij}$  τα στοιχεία του πίνακα που προκύπτει.
4. Δημιουργία των  $\vec{b}, \vec{x}_0 = \frac{1}{N} \mathbf{1}_{N \times 1}$
5. Εκτέλεση αναδρομικού αλγορίθμου της μεθόδου Gauss-Seidel έως ότου  $\|\vec{x}_{k+1} - \vec{x}_k\|^2 < E$  ή  $k > k_{max}$ , όπου έχει προκαθοριστεί  $k_{max} = 150$  για ασφάλεια. Επίσης, αν έχει δοθεί συγκεκριμένος αριθμός επαναλήψεων, θα χρησιμοποιηθεί αυτός.

Σε κάθε βήμα του αλγορίθμου:

- Υπολογίζεται το  $x_{k+1}$ :

$$x_i(k+1) = \frac{1}{1 - a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j(k+1) - \sum_{j=i+1}^N a_{ij}x_j(k) \right)$$

- Υπολογίζεται το  $\|\vec{x}_{k+1} - \vec{x}_k\|^2$  και αυξάνεται το  $k$  κατά 1.

6. Διαίρεση των στοιχείων του  $\vec{x}$  με το άθροισμά τους, ώστε να προκύψει το  $\frac{\vec{x}}{\|\vec{x}\|_1}$  που είναι το διάνυσμα με την pagerank της κάθε σελίδας.

## Παραλληλοποίηση

Η μέθοδος gauss-seidel γενικώς είναι μια σειριακή μέθοδος. Σε περιπτώσεις όμως που ο πίνακας  $A$  είναι αραιός, όπως και στην προκειμένη, είναι δυνατός ο παράλληλος υπολογισμός των  $x_i$  που δεν αλληλοεξαρτώνται και εξαρτώνται μόνο από στοιχεία  $x_j$  με  $j < i$ , των οποίων η τιμή έχει ήδη ενημερωθεί, ή στοιχεία που δεν χρειάζεται να έχουν ενημερωθεί ( $j > i$ ).

Για την εύρεση των ομάδων αυτών με τη μη αλληλοεξαρτώμενα στοιχεία του  $x$  υλοποιήθηκε ο εξής άπληστος αλγόριθμος:

- Διατρέχουμε κατά αύξουσα σειρά τα στοιχεία (κόμβους) του γράφου.
- Τον 1ο κόμβο τον εισάγουμε στην 1η ομάδα.
- Τον 2ο κόμβο τον εισάγουμε κι αυτόν στην 1η ομάδα εκτός αν υπάρχει σύνδεσμος μεταξύ αυτού και του 1ου. Σε εκείνη την περίπτωση τον τοποθετούμε στην 2η ομάδα.
- Συνεχίζουμε με τους υπόλοιπους κόμβους, τοποθετώντας τους κάθε φορά στην  $\Lambda+1$  ομάδα όπου  $\Lambda$  η μεγαλύτερη ομάδα των κόμβων που συνδέονται μ' αυτούς.

Αφού, γίνει ο «χρωματισμός» των κόμβων σε ομάδες προχωράμε σε ανακατάταξη των κόμβων έτσι ώστε τα στοιχεία της κάθε ομάδας να βρίσκονται συνεχόμενα στη μνήμη και με αύξουσα σειρά ομάδας.

Ύστερα, η εύρεση του διανύσματος pagerank είναι όμοια με τον σειριακό αλγόριθμο με τη διαφορά ότι εδώ υπολογίζουμε παράλληλα τις τιμές  $\vec{x}_i(k+1)$  σε κάθε βήμα για τα  $i$  που βρίσκονται στην ίδια ομάδα. Επίσης, παράλληλα γίνονται και όλες οι πράξεις πινάκων που απαιτούνται για τον έλεγχο σύγκλισης.

Τέλος, οι κόμβοι επιστρέφουν στην αρχική τους αλληλουχία ώστε να γίνει εύκολα σύ-

γκριση του αποτελέσματος με τον σειριακό αλγόριθμο.

## Συνθήκες και περιορισμοί

- Η διαθέσιμη μνήμη του συστήματος περιορίζει το μέγιστο όριο κόμβων/συνδέσμων που μπορούν να εισαχθούν στο πρόβλημα. Αυτό το όριο περιορίζεται και από τη χρήση 32-bit μεταβλητών για τους κόμβους. Δηλαδή, δεν μπορούν να χρησιμοποιηθούν γράφοι με πάνω από περίπου 4.3δισ κόμβους.

- Τα αρχεία με τους συνδέσμους που επιλέγονται ως είσοδος στα προγράμματα υπολογισμού της pagerank, πρέπει να είναι συγκεκριμένης μορφής για να είναι συμβατά.

Στα παραδοτέα, συμπεριλαμβάνεται ειδικό πρόγραμμα που μετατρέπει αρχεία κειμένου σαν αυτά των Leskovec και Krevl [5] σε συμβατά αρχεία για τα προγράμματα υπολογισμού της pagerank. Το πρόγραμμα αυτό, διαβάζει στην έναρξη των αρχείων τον συνολικό αριθμό κόμβων και απορρίπτει κάθε κόμβο με id μεγαλύτερο αυτού του αριθμού.

- Η παραλληλοποίηση των iteration της μεθόδου gauss-seidel γίνεται μόνο στις ομάδες που περιέχουν περισσότερους από 50 κόμβους, καθώς το overhead της παραλληλοποίησης κάνει την παράλληλη εκτέλεση του αλγορίθμου μη αποδοτική για μικρότερες ομάδες. Αυτός ο αριθμός, μπορεί εύκολα να αλλάξει ώστε να ταιριάζει στο εκάστοτε σύστημα.

- Σε ορισμένα web-graphs, στα οποία ο αριθμός των ομάδων που προκύπτει με τον greedy αλγόριθμό είναι αρκετά μεγάλος, ο σειριακός αλγόριθμος θα είναι πιθανότατα ταχύτερος. Χρήση διαφορετικής, αλληλουχίας των κόμβων θα δώσει διαφορετικά αποτελέσματα τόσο στη σύγκλιση όσο και στον αριθμό των διαφορετικών ομάδων.

- Δεν επιχειρήθηκε αναζήτηση ακολουθίας κόμβων που θα οδηγούσε σε καλύτερη σύγκλιση ή βελτίωση της ταχύτητας των αλγορίθμων. Όλοι οι κόμβοι διαβάζονται όπως βρίσκονται στο αρχείο και η παραλληλοποίηση γίνεται με τρόπο που δεν επηρεάζει τη σύγκλιση.
- Τόσο η σειριακή όσο και η παράλληλη υλοποίηση, εκτός των αποτελεσμάτων, δημιουργούν σε κάθε τους εκτέλεση κι ένα αρχείο με δεδομένα χρόνων ή κι άλλα χρήσιμα δεδομένα. Τα διαγράμματα που παρουσιάζονται στην επόμενη ενότητα συγκρίνουν τους καθαρούς χρόνους εκτέλεσης του αλγορίθμου Pagerank των δυο υλοποιήσεων, δηλαδή δεν αφορούν τον απόλυτο χρόνο εκτέλεσης του κάθε προγράμματος.

## Ενδεικτικά αποτελέσματα

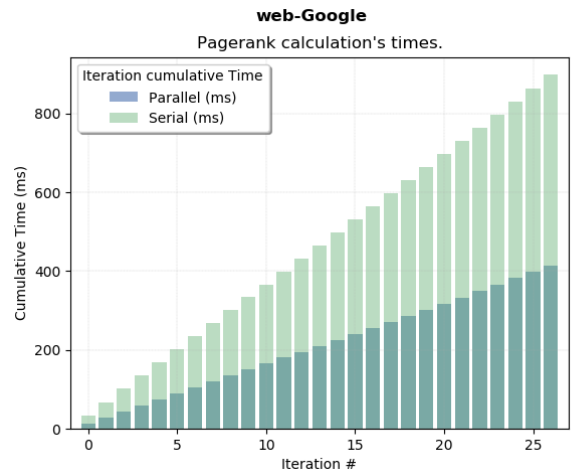
Έγινε εκτέλεση των δύο υλοποιήσεων στα web graphs των Leskovec και Krevl [5]. Εδώ παρουσιάζονται διαγράμματα με διάφορα χαρακτηριστικά των εκτελέσεων για web graph με 875.713 κόμβους με 4.563.235 συνδέσμους στα ακόλουθα συστήματα:

### 1. 4 πύρηνο σύστημα

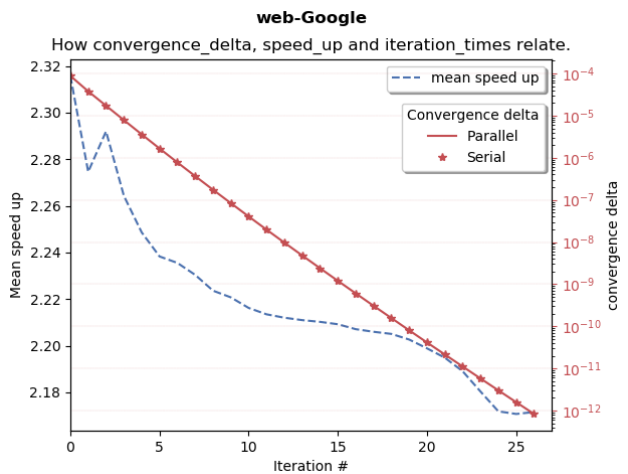
- Intel(R) Core(TM) i5-4670 CPU @ 3.40GHz.
- 8GB DDR3 RAM
- gcc version 7.3.0

### 2. 8 πύρηνο σύστημα (diades)

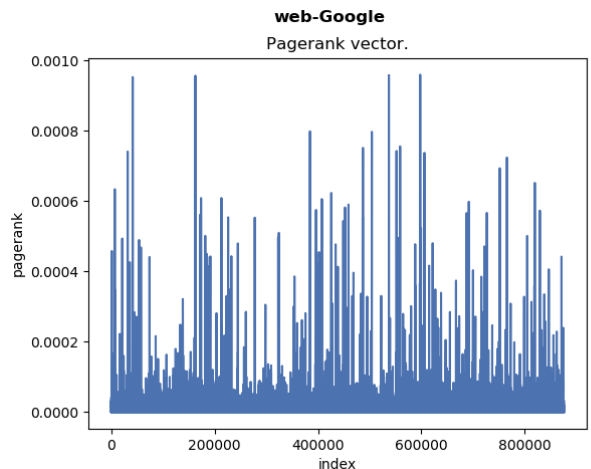
- Intel(R) Xeon(R) CPU E5420 @ 2.50GHz.
- 8GB DDR3 RAM
- gcc version 5.4.0



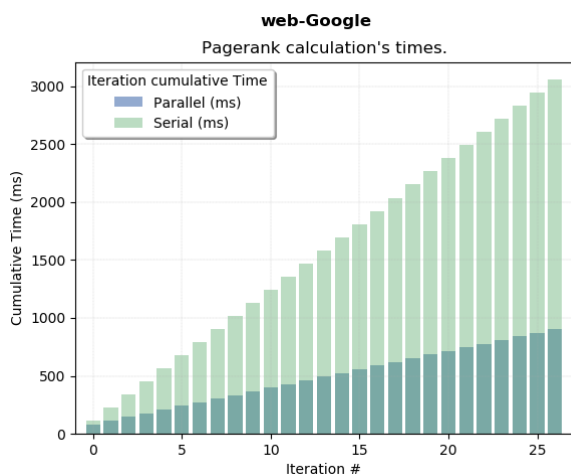
Σχήμα 1: Χρονική εξέλιξη των δύο υλοποιήσεων. 4-πύρηνο σύστημα.



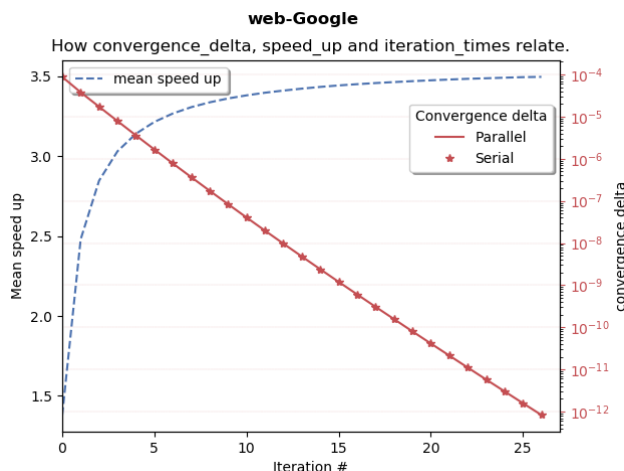
Σχήμα 2: Σχέση σταθεράς σύγκλισης, επιτάχυνσης και αριθμού επαναλήψεων. 4-πύρηνο σύστημα.



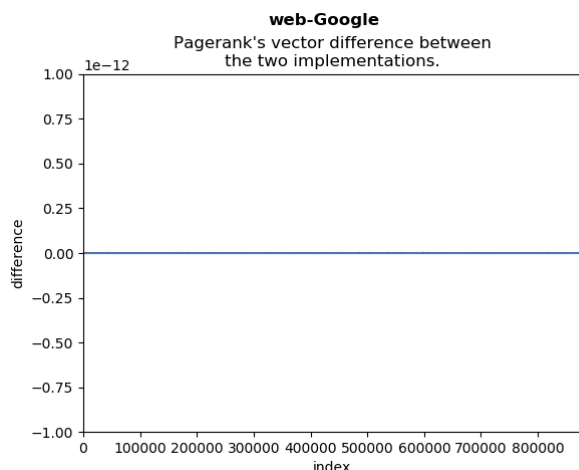
Σχήμα 3: Το αποτέλεσμα για κάθε κόμβο.



Σχήμα 4: Χρονική εξέλιξη των δύο υλοποιήσεων. 8-πύρηνο σύστημα.



Σχήμα 5: Σχέση σταθεράς σύγκλισης, επιτάχυνσης και αριθμού επαναλήψεων. 8-πύρηνο σύστημα.



Σχήμα 6: Διαφορά των δύο υλοποιήσεων στο τελικό διάνυσμα.

Αποτελέσματα και για τα υπόλοιπα web-graphs συμπεριλαμβάνονται στα παραδοτέα. Γενικώς, παρατηρήθηκε ικανοποιητική επιτάχυνση με ταχύτητα τουλάχιστον 2 φορές μεγαλύτερη για την παράλληλη υλοποίηση στο τετραπύρηνο σύστημα ή/και έως 3,5 φορές μεγαλύτερη στο οκταπύρηνο σύστημα. Αντίστοιχα καλές επιδόσεις παρατηρήθηκαν για όλα τα σετ δεδομένων με εξαίρεση το web-BerkStan graph όπου ο edge-coloring αλγόριθμος δημιούργησε ιδιαίτερα αυξημένο αριθμό ομάδων.

Σε όλες τις περιπτώσεις, η παράλληλη υλοποίηση του αλγορίθμου «στοχαστικοποίησης» των στηλών του γράφου για την δημιουργία του πίνακα  $A_s$  ήταν σαφώς ταχύτερη από τη σειριακή.

## Επαλήθευση ορθότητας αποτελεσμάτων

Αρχικά, ο αλγόριθμος υπολογισμού της pagerank που υλοποιήθηκε υποστηρίζεται μαθηματικά σε κάθε του βήμα. Σε κάθε περίπτωση, για να αποκλειστεί η πιθανότητα προγραμματιστικού λάθους, έγινε επαλήθευση αποτελεσμάτων για μικρής κλίμακας web-Graphs με προϋπάρχουσες υλοποιήσεις [6] σε γλώσσα προγραμματισμού *MATLAB*. Αυτές οι υλοποιήσεις συμπεριλαμβάνονται στα παραδοτέα, καθώς και τα script που χρησιμοποιήθηκαν για τη σύγκριση των αποτελεσμάτων.

Η ορθότητα του παράλληλου αλγορίθμου μπορεί να επιβεβαιωθεί κι από την ταύτιση των αποτελεσμάτων του με αυτά του σειριακού, όπως φαίνεται στο Σχ. 6. Επιπλέον, έχουν πάντα ακριβώς ίδιο αριθμό από iterations και ίδια ακολουθία σύγκλισης όπως φαίνεται στα Σχ. 2 και Σχ. 5.

Καθώς ο σκοπός της εργασίας ήταν η παραλληλοποίηση του αλγορίθμου, δεν έγινε αλλαγή στην αλληλουχία των κόμβων με τρόπο που θα διαφοροποιούσε το παράλληλο από το σει-

ριακό πρόγραμμα, καθώς η σύγκλιση της μεθόδου gauss-seidel εξαρτάται από αυτή την αλληλουχία. Έτσι, μπορεί να επιβεβαιωθεί η ταύτιση των αποτελεσμάτων των δύο υλοποιήσεων.

Επιπλέον, μπορεί να επιβεβαιωθεί και πειραματικά η απουσία race condition στην παράλληλη υλοποίηση από το γεγονός ότι συγκλίνει πάντα με τον ίδιο τρόπο που ταυτίζεται και μ' αυτόν της σειριακής.

Τέλος, έχει επιβεβαιωθεί (με το εργαλείο valgrind) ότι τα προγράμματα αυτά δεν παρουσιάζουν σφάλματα μνήμης.

## Ταχύτητα σύγκλισης κι εκτέλεσης

Η ταχύτητα σύγκλισης της μεθόδου gauss-seidel όπως φαίνεται στα αποτελέσματα είναι εντυπωσιακή. Το διάνυσμα της pagerank άρχισε να συγκλίνει από τις πρώτες κι όλες επαναλήψεις, με το τετράγωνο του μέτρου της διαφοράς των τελευταίων προσεγγίσεων να είναι μικρότερο -σε όλες τις περιπτώσεις- του  $10^{-5}$  πριν την 7η επανάληψη.

Αυτός ο ρυθμός σύγκλισης την καθιστά ταχύτερη τόσο από, την κλασσική για το πρόβλημα, power method[7] όσο και από την μέθοδο Jacobi που χρησιμοποιεί μόνο τις προηγούμενες τιμές κάθε  $x_i$  σε κάθε επανάληψη.

Η επιτάχυνση της παράλληλης υλοποίησης, λαμβάνοντας υπόψη τις συνθήκες του αλγορίθμου, τον αριθμό των διαφορετικών χρωμάτων και τις επιταχύνσεις που επιτεύχθηκαν στη βιβλιογραφία[8], κρίνεται κι αυτή από ικανοποιητική έως πάρα πολύ καλή για τα τρία από τα τέσσερα web-graphs στα οποία εξετάστηκε.

## Παραδοτέα

Το repository με όλα τα αρχεία της εργασίας βρίσκεται [εδώ](#).

Επιπλέον, το documentation του κώδικα σε C της εργασίας βρίσκεται [εδώ](#).

## Αναφορές

- [1] S. Brin και L. Page, «The anatomy of a large-scale hypertextual web search engine,» *Computer networks and ISDN systems*, τόμ. 30, αρθμ. 1-7, σσ. 107–117, 1998.
- [2] Y. Saad, *Iterative methods for sparse linear systems*. siam, 2003, τόμ. 82, σσ. 103–105.
- [3] N. Eiron, K. S. McCurley και J. A. Tomlin, «Ranking the web frontier,» στο *Proceedings of the 13th international conference on World Wide Web*, ACM, 2004, σσ. 309–318.
- [4] G. M. Del Corso, A. Gulli και F. Romani, «Fast PageRank computation via a sparse linear system,» *Internet Mathematics*, τόμ. 2, αρθμ. 3, σσ. 251–273, 2005.
- [5] J. Leskovec και A. Krevl, *SNAP Datasets: Stanford Large Network Dataset Collection*, [http : / / snap . stanford . edu/data](http://snap.stanford.edu/data), Ιούν. 2014.
- [6] R. den Besten, «The PageRank Problem,» Delft Institute for Applied Mathematics, Αύγ. 2010.
- [7] D. Silvestre, J. Hespanha και C. Silvestre, «A PageRank Algorithm based on Asynchronous Gauss-Seidel Iterations,»
- [8] W. C. Hasenplaugh, «Parallel algorithms for scheduling data-graph computations,» Διδακτορική διατρ., Massachusetts Institute of Technology, 2016.