Αλγόριθμος Pagerank με παράλληλη υλοποίηση της μεθόδου Gauss-Seidel

Νικόλαος Κατωμέρης ΑΕΜ: 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 $\boldsymbol{A}$, ο $N \times N$ πίνακας μεταβάσεων της αλυσίδας για N σελίδες του διαδικτύου. Αν $j \to i$ σημαίνει ότι η σελίδα j περιέχει σύνδεσμο προς τη σελίδα i, ο $\boldsymbol{A}$ ορίζεται αρχικά ως εξής:

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

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 $\boldsymbol{A}_s$:

$$\boldsymbol{A}_s(i,j) = \begin{cases} \frac{1}{L_j}, & \text{αν } j \to i \\ 0, & \text{αλλιώς} \end{cases}$$

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 σελίδα 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:

$$\boldsymbol{A}_{final} = \frac{c}{N} \cdot \mathbf{1}_{N\times N} + (1-c)\boldsymbol{A}_s$$

Στην παραπάνω σχέση, η σταθερά c διαιρείται με τον αριθμό των γραμμών N έτσι ώστε οι στήλες και του τελικού πίνακα $\boldsymbol{A}_{final}$ 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 $\boldsymbol{A}$ για την ιδιοτιμή 1 και αποτελεί το ζητούμενο διάνυσμα με την αξιολόγηση της κάθε σελίδας (κόμβου).

Για να συγκλίνει με βεβαιότητα το διάνυσμα απαιτείται ακόμη ένα βήμα, θα πρέπει, αν ο περιπατητής βρεθεί σε κάποια σελίδα που δεν περιέχει καμία σύνδεση, να τηλεμεταφερθεί σε μία τυχαία σελίδα με βάση ένα διάνυσμα πιθανοτήτων. Στην υλοποίηση αυτή, ο περιπατητής έχει την ίδια πιθανότητα να τηλεμεταφερθεί σε οποιαδήποτε σελίδα.

Με όλα τα παραπάνω δεδομένα, αποδεικνύεται ότι ο τελικός πίνακας θα έχει μία ιδιοτιμή ένα και το πρόβλημα πλέον έγκειται στην εύρεση του ιδιοδιανύσματός 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:

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

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 σύγκλισής 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 $\boldsymbol{A}$ για την ιδιοτιμή 1. Επομένως, ισχύει ότι:

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

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:

$$(\boldsymbol{I} - (1-c)\boldsymbol{A}_s)\vec{x} = \frac{c}{N}\mathbf{1}_{N\times 1}$$

που μπορεί να λυθεί αναδρομικά με τη μέθοδο 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

Για την υλοποίηση του προγράμματος υπολογισμού της pagerank ενός συνόλου ιστοσελίδων (κόμβων) θα πρέπει τα δεδομένα των συνδέσμων μεταξύ των κόμβων, δηλαδή ο πίνακας $\boldsymbol{A}$, να αποθηκευτούν με κάποιο τρόπο στη μνήμη.

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

Η δομή αυτή επιτρέπει την αποθήκευση πινάκων απαιτώντας 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) καθώς των σταθερών τηλεμεταφοράς (c) και σύγκλισης (E) από τον χρήστη ή χρήση των προεπιλεγμένων τιμών: $b_{def} = 0.15, E_{def} = 1e - 12$

2. Φόρτωση στη μνήμη των δεδομένων εισόδου σε δομή CRS. Τα δεδομένα αυτά αποτελούν τον αρχικό πίνακα $\boldsymbol{A}$.

3. Μετατροπή του $\boldsymbol{A}$ στον $\boldsymbol{A}_s$ και πολλαπλασιασμός του με την σταθερά $-(1-c)$. Έστω a_ij τα στοιχεία του πίνακα που προκύπτει.

4. Δημιουργία των διανυσμάτων $\vec{b} = \frac{b}{N}\mathbf{1}_{N\times 1}$ και του $\vec{x}_0 = \frac{1}{N}\mathbf{1}_{N\times 1}$

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 $k_{max} = 150$. Σε κάθε βήμα του αλγορίθμου:
   - Υπολογίζεται το x_{k+1}:
     $x_i(k+1) = \dfrac{1}{1-a_{ii}}\left(b_i - \sum_{j=1}^{i-1} A_{ij}x_j(k+1) - \sum_{j=i+1}^{N} A_{ij}x_j(k)\right)$
   - Κάθε στοιχείο x_{k+1} διαιρείται με το άθροισμα όλων των στοιχείων του, ώστε το νέο x_{k+1} να ισχύει η προϋπόθεση ότι το άθροισμα των στοιχείων του x σε κάθε βήμα πρέπει να ισούται με 1.
   - Υπολογίζεται το $\|\vec{x}_{k+1} - \vec{x}_k\|^2$ και αυξάνεται το k κατά 1.

Παραλληλοποίηση

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

Για την εύρεση των ομάδων αυτών με τα μη αλληλοεξαρτώμενα στοιχεία του x υλοποιήθηκε ο εξής απλός 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 Λ+1 ομάδα όπου Λ η μεγαλύτερη ομάδα των κόμβων που συνδέονται μ' αυτούς.

Αφού, γίνει ο 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 σύγκριση του αποτελέσματος με τον σειριακό αλγόριθμο.

Συνθήκες και περιορισμοί

- Και οι δύο υλοποιήσεις έχουν μέγιστο όριο κόμβων και συνδέσμων με βάση τη διαθέσιμη μνήμη του συστήματος στο οποίο τρέχουν.

- Τα αρχεία συνδέσμων που δέχονται πρέπει να είναι συγκεκριμένης μορφής. Δηλαδή, θα πρέπει να είναι δυαδικά αρχεία που να περιέχουν με σειρά:

  1. Τον αριθμό κόμβων που περιέχουν.
  2. Τον αριθμό συνδέσμων μεταξύ κόμβων που περιέχουν.
  3. Όλους τους συνδέσμους με μορφή δύο συνεχόμενων αριθμών που αναπαριστούν τους δύο κόμβους του κάθε συνδέσμου.

  Όλοι οι παραπάνω αριθμοί θα πρέπει να έχουν τη μορφή «4-byte uint» για να διαβαστούν σωστά από το αρχείο.

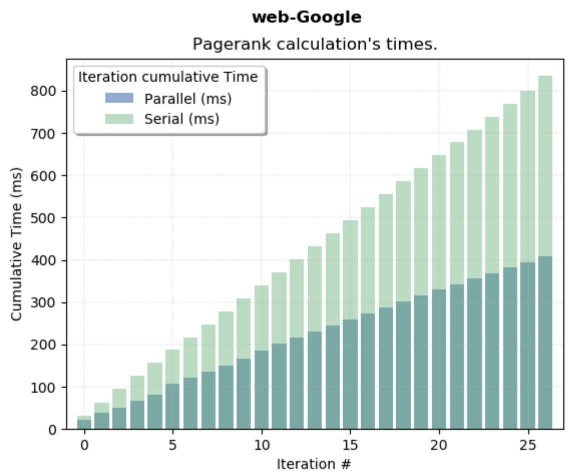
  Στα παραδοτέα, συμπεριλαμβάνεται ειδικό πρόγραμμα που μετατρέπει αρχεία κειμένου με κατευθυνόμενους συνδέσμους σε αρχεία της παραπάνω μορφής.

- 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.

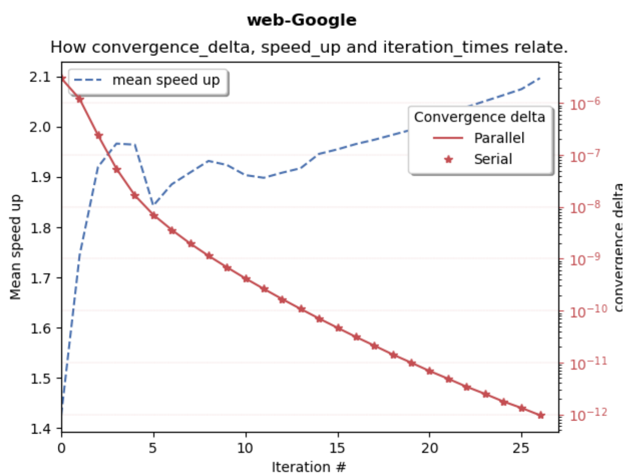
- Σε ορισμένα web-graphs, στα οποία ο αριθμός των ομάδων που προκύπτει με τον greedy αλγόριθμό είναι αρκετά μεγάλος, ο σειριακός αλγόριθμος θα είναι πιθανότατα ταχύτερος. Ένας διαφορετικός αλγόριθμος edge-coloring θα δώσει διαφορετικά αποτελέσματα.

- Τα προγράμματα έχουν υλοποιηθεί για σταθερό web graph. Προσθήκη νέων σελίδων ή αλλαγές σε συνδέσμους, κατά τη διάρκεια του υπολογισμού της pagerank, δεν υποστηρίζονται.

- Το παράλληλο πρόγραμμα, όπως δίνεται, ενδεχομένως να χρειαστεί περισσότερο χρόνο να τρέξει από το σειριακό καθώς -κάθε φορά που τρέχει- κάνει edge coloring και δύο ανακατατάξεις στον πίνακα και στο τελικό διάνυσμα της pagerank. Αυτοί οι έξτρα χρόνοι δεν είναι απαραίτητοι καθώς οι κόμβοι θα μπορούσαν να είναι εξ' αρχής έτσι στο αρχείο. Παρόλα αυτά μετρούνται χωριστά οι χρόνοι όλων των διαδικασιών και των δύο υλοποιήσεων. Έτσι, τα διαγράμματα που παρουσιάζονται στην επόμενη ενότητα συγκρίνουν τους καθαρούς χρόνους εκτέλεσης του αλγορίθμου Pagerank των δυο υλοποιήσεων.



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



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

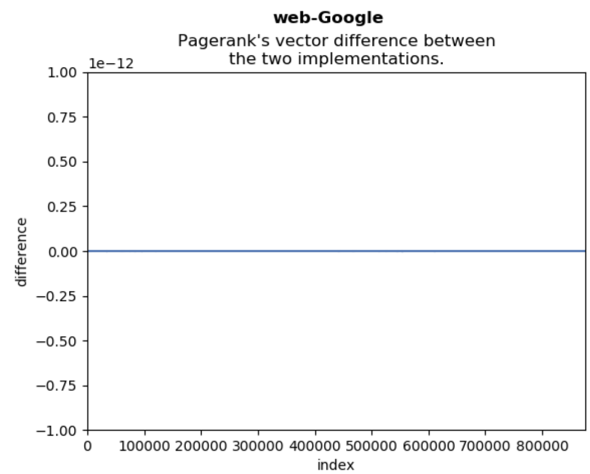
Ενδεικτικά αποτελέσματα

Έγινε εκτέλεση των δύο υλοποιήσεων στα web graphs των Leskovec και Krevl [3].

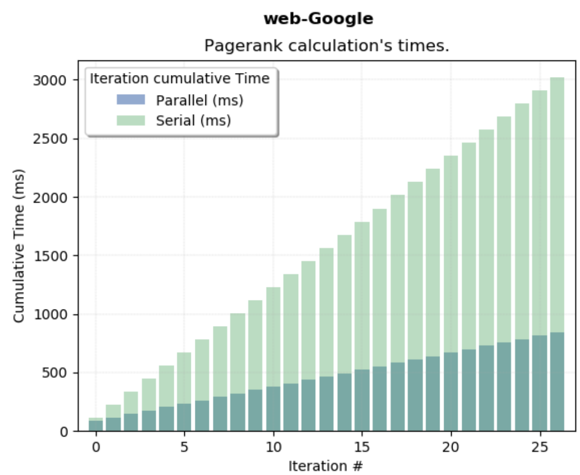
Παρακάτω παρουσιάζονται διαγράμματα τα διάφορα χαρακτηριστικά των εκτελέσεων.

Τα web graph που χρησιμοποιήθηκε περιείχε 875.713 κόμβους με 4.563.235 συνδέσμους ενώ δύο συστήματα έχουν τα εξής χαρακτηριστικά:

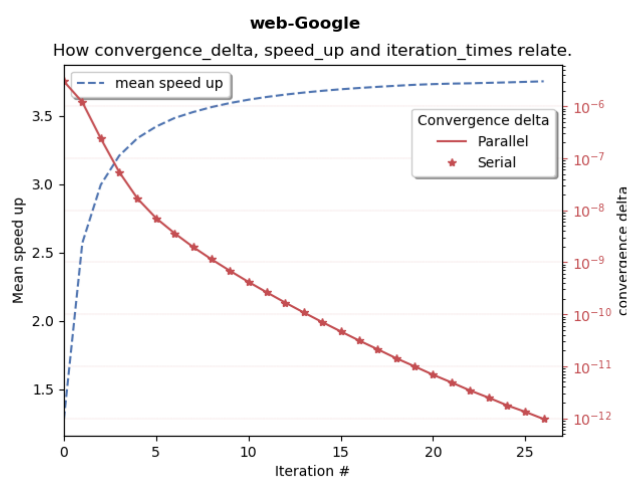
1. 4 πύρηνο σύστημα



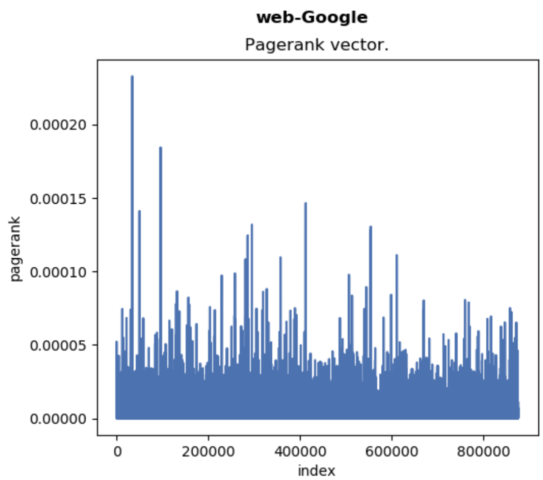
Σχήμα 3: Διαφορά των δύο υλοποιήσεων στο τελικό διάνυσμα.



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



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



Σχήμα 6: Το αποτέλεσμα για κάθε κόμβο.

- 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

Στα παραδοτέα, περιέχονται έτοιμα τα αποτελέσματα και για τα υπόλοιπα web-graphs. Γενικώς, παρατηρήθηκε ικανοποιητική επιτάχυνση με ταχύτητες έως και 240% στο τετραπύρηνο ή/και έως 360% στο οχταπύρηνο σύστημα σε σχέση με τη σειριακή υλοποίηση, για όλα τα δεδομένα με εξαίρεση το web-BerkStan graph όπου ο edge-coloring αλγόριθμος δημιούργησε ιδιαίτερα αυξημένο αριθμό ομάδων. Σε εκείνη την περίπτωση, ο πιο σύγχρονος και γρήγορος επεξεργαστής του 4-πύρηνου συστήματος, εμφάνισε επιτάχυνση έως 120% περίπου.

Στην παράλληλη υλοποίηση δίνεται και η δυνατότητα επιλογής σταθερού αριθμού threads που θα δημιουργήσει το openMP, η επιλογή αυτή, όμως, οδηγεί συνήθως σε χειρότερη επίδοση του προγράμματος.

Επαλήθευση ορθότητας αποτελεσμάτων

Ο σειριακός αλγόριθμος που υλοποιήθηκε υποστηρίζεται μαθηματικά σε κάθε του βήμα πράγμα που αρκεί για να θεωρηθεί σωστός. Σε κάθε περίπτωση, για να αποκλειστεί η πιθανότητα προγραμματιστικού λάθους, έγινε επαλήθευση αποτελεσμάτων για μικρής κλίμακας web-Graphs με προϋπάρχουσες υλοποιήσεις που βρέθηκαν στο διαδίκτυο 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 από την ταύτιση των αποτελεσμάτων της με αυτά του σειριακού. Συγκεκριμένα, έχουν πάντα ακριβώς ίδιο αριθμό από iterations, ίδια ακολουθία σύγκλισης και πρακτικά ίδιο αποτέλεσμα. Όλα τα παραπάνω μπορούν να επιβεβαιωθούν χρησιμοποιώντας το ειδικά δημιουργημένο *python script* που περιλαμβάνεται στα παραδοτέα. Αυτό δημιουργεί γραφικές παραστάσεις από τα δεδομένα καταγραφής και τα αποτελέσματα των προγραμμάτων σαν αυτές που παρουσιάστηκαν στην παρούσα αναφορά.

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 όλας επαναλήψεις, με το τετράγωνο του μέτρου της διαφοράς των τελευταίων προσεγγίσεων να είναι μικρότερο -σε όλες τις περιπτώσεις- του 10^{-7} από την 7η επανάληψη.

Αυτός ο ρυθμός σύγκλισης την καθιστά σαφώς ταχύτερη τόσο από την κλασσική για το πρόβλημα power method[4] 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[5], κρίνεται κι αυτή από ικανοποιητική έως πάρα πολύ καλή για τα περισσότερα web-graphs.

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] J. Leskovec και A. Krevl, *SNAP Datasets: Stanford Large Network Dataset Collection*, http://snap.stanford.edu/data, Ιούν. 2014.

[4] D. Silvestre, J. Hespanha και C. Silvestre, «A PageRank Algorithm based on Asynchronous Gauss-Seidel Iterations,»

[5] W. C. Hasenplaugh, «Parallel algorithms for scheduling data-graph computations,» Διδακτορική διατρ., Massachusetts Institute of Technology, 2016.