Στυλιανή Καλαϊτζάκη dai19081
Χρήστος Κεχαγιάς dai18213

Περιγραφή Αλγορίθμου

Γλώσσα Προγραμματισμού: Java

Ο αλγόριθμος του προγράμματος περιλαμβάνει δύο κύκλους map-reduce.

Πρόγραμμα UserDocParser

Είσοδος: δέχεται το όνομα του αρχείου δεδομένων ή καταλόγου που περιέχει τα αρχεία δεδομένων, ένα όνομα για το output directory που θα δημιουργήσει και προεραιτικά τον αριθμό των tasks για τους reducers (η default τιμή είναι 1).

Έξοδος: το αποτέλεσμα αποτελείται από ζέυγη <χρήστης, αρχείο> για τους χρήστες που προσπέλασαν ένα αρχείο πάνω από μία φορά σε διαφορετικές ημερομηνίες. Βρίσκεται μέσα στον κατάλογο /final μέσα στο output direcoty που δόθηκε ως είσοδος.

Κλάση **FirstMapper**:
Περιλαμβάνει το πρώτο map των δεδομένων. Λαμβάνει ως είσοδο ένα αρχείο και με τη χρήση του Tokenizer παίρνει μία μία τις γραμμές, δηλαδή τις εγγραφές. Χρησιμοποιώντας το split χωρίζει τις γραμμές στα κόμματα και αποθηκεύει το κάθε πεδίο μιας γραμμής σε έναν προσωρινό πίνακα. Το output που γράφει στο context έχει ως key τα πεδία <ip,doc,date> και ως value την τιμή null.

Κλάση **FirstReducer**:
Λαμβάνει το output του πρώτου mapper και για κάθε μοναδικό key γράφει στο output <key,null>.

Εν ολίγης, ο πρώτος κύκλος map-reduce κάνει ένα group by με βάση τα ip,doc,date. Αποθηκεύει το αποτέλεσμα σε έναν /temp κατάλογο που διαγράφεται αφού τελειώσει και ο δεύτερος κύκλος map-reduce.

Κλάση **SecondMapper**:
Ο δεύτερος mapper παίρνει ως είσοδο το αποτέλεσμα του πρώτου κύκλου map-reduce και αφού αφαιρέσει το πεδίο date γράφει στο context ως key <ip,doc> και ως value <”1”>.

Κλάση **SecondReducer**:
Για κάθε τιμή του key προσθέτει τα values που του αντιστοιχούν και αν το άθροισμά τους είναι μεγαλύτερο του 1, γράφει στο output <ip,doc>.

Στην ουσία ο δεύτερος κύκλος map-reduce διενεργεί ένα count.

Σχολιασμός χρόνων

Όσον αφορά το ***Elapsed Time***

- **1 κόμβος**: Με task=1 σημείωνει χρόνο 7,50min και κάθως αυξάνονται οι διεργασίες από 1 σε 2 tasks παρατηρούμε μία μικρή μείωση στον χρόνο 7,50-7,426=0,08 seconds ενώ μόλις χρησιμοποιήσουμε 3 διεργασίες βλέπουμε ότι ο χρόνος αυξήθηκε 7,663-7,50=0,157sec σε σχέση με την χρήση μίας διεργασίας.

- **2 κόμβοι**: Πιο αισθητή είναι η διαφορά μεταξύ των διεργασών όταν χρησιμοποιούμε 2 κόμβους. Παρατηρούμε πως στην πρώτη διεργασία ο χρόνος είναι 4,8min ενώ όταν αρχίσουμε να αυξάνουμε τις διεργασίες κερδίζουμε 4,8-4,093=0,71sec στον χρόνο εκτέλεσης χρησιμοποιώντας 2 διεργασίες και 4,8-3,92=0,88sec για 4 διεργασίες

- **Μεταξύ 1 κόμβου & 2 κόμβων**: Αρχικά παρατηρούμε από το task 1 ύπαρχει μεγάλο gap και κάθως αυξάνονται οι διεργασίες αυτό το gap όλο και μεγαλώνει. Συγκεκριμένα όσον αφορά την χρήση της μίας διεργασίας η διαφορά στον χρόνο είναι 7,506-4,8=2,706min ενώ αντίστοιχα στην χρήση 2 διεργασίων η διαφορά είναι 7,42-4,09=3,327min και στις 4 διεργασίες είναι 7,663-3,92=3,743min. Και έτσι καταλήγουμε στο συμπέρασμα πως χρησιμοποιώντας περισσότερους κόμβους ο χρόνος μειώνεται δραματικά και σε αυτήν τη μείωση μπορεί να συνεισφέρει ακόμη περισσότερο η χρήση διεργασιών.

Όσον αφορά το ***Average Map Time***

- **1 κόμβος**: Με task=1 έχουμε 0,37sec ενώ για task=2 ο χρόνος είναι 0,393sec και για task=4 ο χρόνος είναι 0,403sec δηλαδή παρατηρούμε μια μικρή αύξηση καθώς αυξάνονται οι διεργασίες και ο λόγος που μπορεί να συμβαίνει αυτό είναι διότι αυξάνεται η πολυπλοκότητα μιας και θα πρέπει αφού τελειώσουν οι διεργασίες να ενωθούν τα αποτελέσματα και αυτό οδηγεί σε περισσότερη καθυστέρηση

- **2 κόμβοι**: Το ίδιο που συνέβαινε στον ένα κόμβο συμβάινει και στην χρήση 2 κόμβων.

- **Μεταξύ 1 κόμβου & 2 κόμβων:** Παρατηρούμε πως όταν χρησιμοποιούμε έναν κόμβο ο μέσος χρόνος του map είναι καλύτερος σε σχέση με την χρήση δύο κόμβων και αυτό συνεχίζει να συμβαίνει καθώς αυξάνονται οι διεργασίες πράγμα που μας δίνει να καταλάβουμε πως παρόλο που χρησιμοποιούμε διεργασίες για να μείωσουμε τον χρόνο εκτέλεσης αυξάνουμε τον χρόνο εκτέλεσης του map εξαιτίας όπως είπαμε της πολυπλοκότητας που προσδίδει η προσθήκη επιπλέον διεργασιών στον συντονισμό και την συγκέντρωση των αποτελεσμάτων.

Όσον αφορά το ***Average Reduce Time***

- **1 κόμβος**: Με task=1 έχουμε 0,68sec ενώ με task=2 έχουμε μια δραματική μείωση στον χρόνο και συγκεριμένα 0,68-0,38=0,3sec και για task=4 ο χρόνος είναι 0,21sec δηλαδή παρατηρούμε πως η χρήση διεργασίων έχει τεράστια επίδραση στον μέσο χρόνο του reduce time.

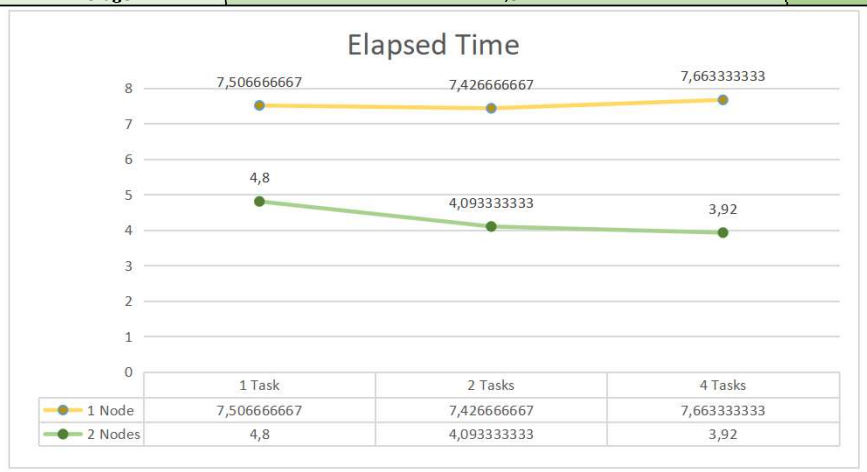
- **2 κόμβοι**: Το ίδιο που συνέβαινε στον ένα κόμβο συμβαίνει ακριβώς και στην χρήση 2 κόμβων δηλαδή υπάρχει μια μείωση όσο περισσότερες διεργασίες χρησιμοποιούμε.

- **Μεταξύ 1 κόμβου & 2 κόμβων**: Παρατηρούμε πως η χρήση ενός κόμβου σε σχέση με την χρήση δύο κόμβων έχουν περίπου τα ίδια αποτελέσματα στον μέσο χρόνο του reduce time.

Elapsed Time

1 Node	1 Task , 1 execution	1 Task, 2 execution	1 Task, 4 execution	2 Tasks, 1 execution	2 Tasks, 3 execution	2 Tasks, 4 execution	4 Tasks, 1 execution	4 Tasks, 3 execution	4 Tasks, 4 execution
First Mapper & Reducer	6,36	6,25	6,23	6,14	6,2	6,13	6,44	6,45	6,46
Second Mapper & Reducer	1,22	1,23	1,23	1,27	1,28	1,26	1,23	1,24	1,17
Total(Map1+Map2)	7,58	7,48	7,46	7,41	7,48	7,39	7,67	7,69	7,63
Average	7,506666667			7,426666667			7,663333333		

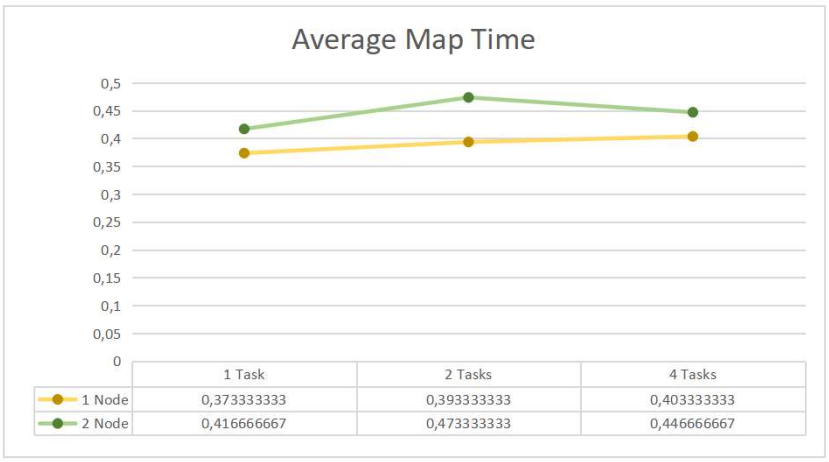
2 Nodes	1 Task , 2 execution	1 Task, 4 execution	1 Task, 5 execution	2 Tasks, 1 execution	2 Tasks, 2 execution	2 Tasks, 4 execution	4 Tasks, 2 execution	4 Tasks, 4 execution	4 Tasks, 5 execution
First Mapper & Reducer	3,53	3,57	4,08	3,4	3,36	3,44	3,39	3,41	3,4
Second Mapper & Reducer	1,09	1,05	1,08	0,52	1,02	0,54	0,53	0,53	0,5
Total(Map1+Map2)	4,62	4,62	5,16	3,92	4,38	3,98	3,92	3,94	3,9
Average	4,8			4,093333333			3,92		

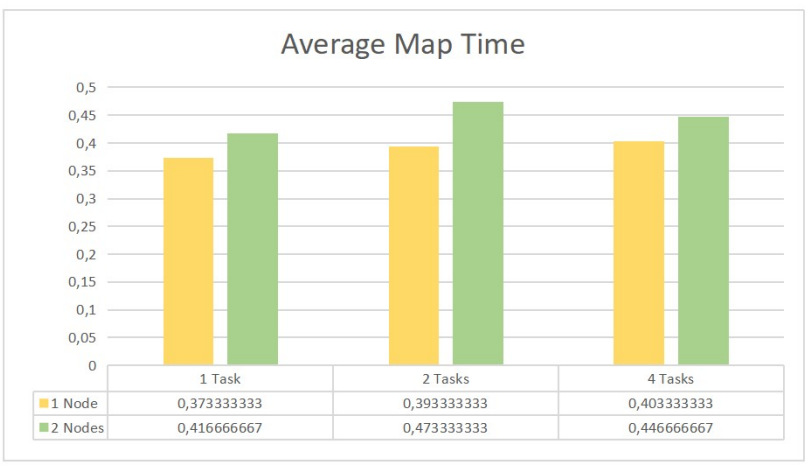


Average Map Time

1 Node	1 Task, 1 execution	1 Task, 2 execution	1 Task, 3 execution	2 Tasks, 1 execution	2 Tasks, 2 execution	2 Tasks, 3 execution	4 Tasks, 2 execution	4 Tasks, 3 execution	4 Tasks, 4 execution
First Mapper & Reducer	0,15	0,15	0,15	0,14	0,14	0,14	0,13	0,13	0,13
Second Mapper & Reducer	0,23	0,22	0,22	0,26	0,24	0,26	0,28	0,26	0,28
Total(Map1+Map2)	0,38	0,37	0,37	0,4	0,38	0,4	0,41	0,39	0,41
Average	0,373333333			0,393333333			0,403333333		

2 Nodes	1 Task, 2 execution	1 Task, 4 execution	1 Task, 5 execution	2 Tasks, 1 execution	2 Tasks, 2 execution	2 Tasks, 4 execution	4 Tasks, 2 execution	4 Tasks, 4 execution	4 Tasks, 5 execution
First Mapper & Reducer	0,19	0,19	0,2	0,19	0,19	0,19	0,18	0,18	0,17
Second Mapper & Reducer	0,22	0,22	0,23	0,28	0,29	0,28	0,27	0,27	0,27
Total(Map1+Map2)	0,41	0,41	0,43	0,47	0,48	0,47	0,45	0,45	0,44
Average	0,416666667			0,473333333			0,446666667		





Average Reduce Time

1 Node	1 Task, 2 execution	1 Task, 3 execution	1 Task, 4 execution	2 Tasks, 1 execution	2 Tasks, 2 execution	2 Tasks, 3 execution	4 Tasks, 2 execution	4 Tasks, 3 execution	4 Tasks, 4 execution
First Mapper & Reducer	0,46	0,45	0,43	0,25	0,25	0,24	0,14	0,15	0,15
Second Mapper & Reducer	0,23	0,23	0,24	0,13	0,13	0,14	0,07	0,07	0,07
Total(Map1+Map2)	0,69	0,68	0,67	0,38	0,38	0,38	0,21	0,22	0,22
Average	0,68			0,38			0,216666667		

2 Nodes	1 Task, 2 execution	1 Task, 4 execution	1 Task, 5 execution	2 Tasks, 1 execution	2 Tasks, 2 execution	2 Tasks, 4 execution	4 Tasks, 2 execution	4 Tasks, 4 execution	4 Tasks, 5 execution
First Mapper & Reducer	0,38	0,38	0,43	0,21	0,23	0,22	0,13	0,13	0,14
Second Mapper & Reducer	0,24	0,25	0,24	0,12	0,13	0,12	0,07	0,08	0,07
Total(Map1+Map2)	0,62	0,63	0,67	0,33	0,36	0,34	0,2	0,21	0,21
Average	0,64			0,343333333			0,206666667