ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

Τμήμα Πληροφορικής

ΕΠΛ 660 – Ανάκτηση Πληροφοριών και Μηχανές Αναζήτησης

ΑΣΚΗΣΗ 2 – Ανάπτυξη Κατανεμημένου και Κλιμακώσιμου Συστήματος Ανάκτησης Πληροφοριών «DSPythia»

Διδάσκων: Γιώργος Πάλλης

Υπεύθυνος Εργασίας: Παύλος Αντωνίου

Ημερομηνία Ανάθεσης: Δευτέρα 15/10/18

Ημερομηνία Παράδοσης: Δευτέρα 05/11/18 και ώρα 12:00 μεσημέρι (21 μέρες)

(η λύση να υποβληθεί σε zip μέσω του Moodle)

http://www.cs.ucy.ac.cy/courses/EPL660

1. Στόχος

Ο στόχος της παρούσας εργασίας είναι η εξοικείωσή σας με διάφορα εργαλεία επεξεργασίας και ανάλυσης δεδομένων όπως για παράδειγμα τη μηχανή αναζήτησης ElasticSearch, το περιβάλλον κατανεμημένης επεξεργασίας μεγάλων δεδομένων Apache Hadoop (που υλοποιεί το μοντέλο Map-Reduce), τη βιβλιοθήκη δημιουργίας ευρετηρίων και αναζήτησης Apache Lucene, την πλατφόρμα αναζήτησης Apache Solr, και το μοντέλο επερωτήσεων Vector Space.

2. Ζητούμενα

Στην πρώτη εργασία αναπτύξατε ένα σύστημα επεξεργασίας, αποθήκευσης, αναζήτησης και ανάκτησης πληροφοριών με την ονομασία **Dionysos**. Το σύστημα υποστήριζε δημιουργία αντεστραμμένων ευρετηρίων όρων (inverted index) από συλλογές κριτικών για κρασιά και μια βασική τεχνική αναζήτησης πληροφοριών (Boolean model) μέσα στα ευρετήρια.

Η επεξεργασία των δεδομένων προς δημιουργία του αντεστραμμένου ευρετηρίου και η αποθήκευσή του γινόταν στη μνήμη και το δίσκο μιας μόνο μηχανής. Καθώς ο όγκος δεδομένων μεγαλώνει, η ανάγκη για κατανεμημένη (distributed) και κλιμακώσιμη (scalable) επεξεργασία δεδομένων και αποθήκευση αντεστραμμένου ευρετηρίου γίνεται ολοένα και πιο επιτακτική.

Σε αυτή την εργασία καλείστε να αναπτύξετε ένα κατανεμημένο και κλιμακώσιμο σύστημα επεξεργασίας, αποθήκευσης, αναζήτησης και ανάκτησης πληροφοριών με το όνομα **DSPythia (Distributed Scalable Pythia)**. Στην καρδιά του συστήματος αυτού θα βρίσκεται ένα κατανεμημένο και κλιμακώσιμο **αντεστραμμένο ευρετήριο (full scalable and distributed inverted index)**. Η αναζήτηση και ανάκτηση δεδομένων στο ευρετήριο αυτό θα γίνεται με βάση το **Vector Space model**.

Να υλοποιήσετε το λογισμικό σύστημα διαχείρισης συλλογών εγγράφων DSPythia, που περιλαμβάνει τα ακόλουθα υποσυστήματα:

- Υποσύστημα επεξεργασίας συλλογών αρχείων κειμένου, δημιουργίας και διαχείρισης αντεστραμμένου ευρετηρίου. Οι ενέργειες που μπορούν να εκτελούνται πάνω στη κάθε συλλογή (π.χ. δημιουργία, άνοιγμα, κλείσιμο κτλ) **είναι οι ίδιες με αυτές τις άσκησης 1**.
- Υποσύστημα αναζήτησης όρων στο αντεστραμμένο ευρετήριο.
    - Υποστήριξη επερωτήσεων χρησιμοποιώντας το Vector Space Model.
    - Υποστηρίζει τη δυνατότητα αναζήτησης ενός ή περισσοτέρων όρων. Σε κάθε αναζήτηση, το πρόγραμμα να επιστρέφει, εκτός από τα αρχεία στα οποία περιλαμβάνεται ο αναζητούμενος όρος, και το χρόνο που χρειάσθηκε η πραγματοποίηση της αναζήτησης.

Για την υλοποίηση των πιο πάνω υποσυστημάτων μπορείτε να χρησιμοποιήσετε ότι κρίνετε εσείς καταλληλότερο: ElasticSearch ή/και, Apache Hadoop ή/και Apache Lucene ή/και Apache Solr κτλ.

Να αξιολογήσετε την υλοποίηση που κάνατε (χρησιμοποιώντας τις κατάλληλες μετρικές) και **να ερμηνεύσετε τα αποτελέσματα**. Για την αξιολόγηση σας θα βρείτε ιδιαιτέρως χρήσιμο να χρησιμοποιήσετε επερωτήματα (queries) που δίνονται μαζί με τη συλλογή δεδομένων (dataset) που παρουσιάζεται πιο κάτω (παράγραφος 5) και να συγκρίνετε τα αποτελέσματα που θα λάβετε από το σύστημά σας μαζί με τα αποτελέσματα της αξιολόγησης που περιέχεται στο dataset.

3. Παραδοτέα

Θα παραδώσετε (μέσω moodle) ένα αρχείο .zip με όνομα epl660_id_ex2.zip που να περιλαμβάνει:

- πηγαίο κώδικα του συστήματός σας με σχόλια (αν υλοποιηθεί σε Eclipse να παραδοθεί ολόκληρο το project) που να συμπεριλαμβάνει τις επιπλέον βιβλιοθήκες (αν υπάρχουν),

- τεκμηρίωση όπου να φαίνεται ο τρόπος σκέψης σας για την ανάπτυξη του συστήματος, αλγόριθμοι και τεχνικές που χρησιμοποιήθηκαν, τα αποτελέσματα της αξιολόγησης σας, κλπ. Η αναφορά θα πρέπει να ονομάζεται με τον εξής τρόπο: epl660_id_ex2.[doc|pdf]. Η τεκμηρίωση είναι ΥΠΟΧΡΕΩΤΙΚΗ και πολύ σημαντική για την βαθμολόγηση της εργασίας.

4. Συλλογή Αρχείων (Dataset)

Η συλλογή αρχείων τα οποία θα εντάξετε στη συλλογή του DSPythia ονομάζεται Cranfield. Η συλλογή αυτή περιέχει 1399 περιλήψεις (abstracts) από άρθρα που δημοσιεύθηκαν σε περιοδικά αεροδυναμικής (aerodynamics journal articles) στα τέλη της δεκαετίας του 1950. Είναι σχετικά μικρής κλίμακας συλλογή που είναι ευρέως διαδεδομένη στον τομέα της ανάκτησης πληροφορίας (IR) για τη διενέργεια πιλοτικών πειραματικών εφαρμογών, όπως η παρούσα άσκηση. Η συλλογή διατίθεται για στο ακόλουθο URL: http://www.cs.ucy.ac.cy/courses/EPL660/exercises/cran.tar.gz και περιέχει τα πιο κάτω αρχεία:

- cran.all – Όλα τα άρθρα σε ένα αρχείο απλού κειμένου (txt). Για κάθε άρθρο δίνονται τα εξής χαρακτηριστικά: I (index), T (title), A (author), B (bibliography) και W (words). Το τελευταίο αφορά στις λέξεις που υπάρχουν στο abstract κάθε άρθρου.
- cran.qry – Ένα σύνολο από 225 επερωτήσεις (queries). Για κάθε query δίνονται το I (index)

και το W (words) που είναι οι όροι του query.

- cranqrel – Αξιολόγηση της σχετικότητας των επερωτήσεων για όλα τα ζεύγη (επερώτηση, άρθρο). Πιο συγκεκριμένα, σε κάθε γραμμή του αρχείου περιέχονται 3 ακέραιοι αριθμοί που αντιστοιχούν στον αριθμό της επερώτησης (query number), τον αριθμό του άρθρου (article number) και τον κωδικό της σχετικότητας (1-5). Για τους κωδικούς αυτούς διαβάστε τι υπάρχει μέσα στο πιο κάτω αρχείο.
- cranqrel.readme – Επεξήγηση αρχείου αξιολογήσεων.

5. Χρήσιμες αναφορές

- Apache Hadoop. http://hadoop.apache.org/
- Apache Lucene. http://lucene.apache.org/
- Apache Solr – Distributed Search. http://wiki.apache.org/solr/DistributedSearch
- Apache Solr – Running Solr on HDFS. https://cwiki.apache.org/confluence/display/solr/Running+Solr+on+HDFS
- Apache Solr – MapReduceIndexer Tool [link1] [link2].
- Solrj – Java Client to access Solr. https://wiki.apache.org/solr/Solrj, http://www.solrtutorial.com/solrj-tutorial.html
- ElasticSearch - https://www.elastic.co/
- ElastixSearch – Java API
- R. McCreadie , C. Macdonald, I. Ounis. MapReduce indexing strategies: Studying scalability and efficiency, Information Processing and Management 48 (2012) 873-888.
- D. Jeff and G. Sanjay. MapReduce: Simplified Data Processing on Large Clusters. In Sixth Symposium on Operating System Design and Implementation (OSDI'04), San Fransisco, CA, December 2004.
- D. Jeff and G. Sanjay. MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM, 51(1), 2008.
- J. Zobel. Inverted Files for Text Search Engines. ACM Computing Surveys, 38(2), 2006.