Indexarea documentelor folosind paradigma Map-Reduce

Responsabil de temă: **Elena Apostol**

Data publicării: **11/09/2012**

Termenul de predare: ~~**11/23/2012**~~ **11/25/2012**

Observatie: Metoda **main** va trebui sa se afle intr-o clasa cu numele **Main** si **nu** va fi inclusa in vreun pachet.

1. Introducere. Cerinţele temei

Pentru a facilita obţinerea rapidă si precisă a informaţiilor existente in paginile web, motoarele de căutare indexează (colectează, parsează si stochează) datele înainte să efectueze căutări. Procesul se mai numeşte si Web indexing atunci cand se referă la indexarea paginilor existente in Internet.

Se poate face o căutare simplă, după un cuvînt cheie intr-un document neindexat, se poate face indexarea unui document in timp ce se face o căutare in el pentru prima dată, sau se pot indexa in prealabil toate documentele. Pentru optimizarea timpului de căutare, motoarele de căutare cele mai populare fac indexarea totală a textului existent online.

Cerinţele temei

In aceasta temă se cere scrierea unui program paralel in Java care să facă indexarea unui set de documente primit ca input si apoi căutare după cuvinte cheie in documentele indexate. In urma procesului de indexare se determină numarul de aparitii al fiecărui cuvant existent intr-o pagină web sau document, obtinandu-se o listă de perechi (cuvant, număr de apariţii). Programul trebuie sa permită efectuarea de căutări după cuvinte cheie intr-un set de documente si afişarea documentelor cu gradul maxim de relevanţă. Căutările se pot face după unul, două sau mai multe cuvinte cheie.

Pentru paralelizarea indexării sa va folosi paradigma replicated workers (vezi laborator 5) si modelul MapReduce, ideea generală fiind: fiecare document se fragmentează in părţi de dimensiune fixă, care vor fi procesate in paralel, pentru fiecare parte rezultand cate un vector partial continand termenii si numarul de aparii ale acestora. Apoi vectorii sunt combinati si va rezulta un vector ce caracterizeaza intregul document. Notam lungimea finala a vectorului cu N (adica pastram cele mai frecvente N cuvinte). Pentru

ca un document sa fie returnat ca relevant intr-o anumita cautare, cuvantul/cuvintele cheie dupa care se face cautarea trebuie sa se afle in vectorul care contine termenii cu frecventele cele mai mari de aparitie.

2. Implementare

2.1 Paradigma Map-Reduce - Prezentare generala

Pentru rezolvarea acestei probleme se va folosi un model replicated-workers, asemanator cu modelul MapReduce folosit de inginerii de la Google pentru procesarea unor seturi mari de documente in sisteme paralele si distribuite. Acest articol prezinta modelul MapReduce folosit de Google si o parte dintre aplicatiile lui (mai importante pentru intelegerea modelului sunt primele 4 pagini).

MapReduce este un model de programare paralela (si implementarea asociata) pentru procesarea si generarea unor seturi imense de date folosind sute sau mii de procesoare. Modelul permite paralelizarea si distributia automata a taskurilor. Paradigma MapReduce se bazeaza pe existenta a doua functii: map si reduce. Map primeste ca input o functie f si o lista si returneaza o noua lista aplicand functia f fiecarui element din lista. Reduce combina rezultatele obtinute anterior.

Mecanismul MapReduce functioneaza in modul urmator:

- Utilizatorul cere procesarea unui set de documente; aceasta cerere este adresata unui proces (fir de executie) master.

- Master-ul imparte documentele in fragmente de dimensiuni fixe care vor fi asignate unor procese (fire de executie) worker; un worker va executa pentru un fragment de fisier o operatie numita “map”, care va genera niste rezultate partiale avand forma unor perechi de tip (cheie, valoare).
- Dupa ce operatiile “map” au fost executate, master-ul asigneaza worker-ilor task-uri de tip “reduce”, care combina rezultatele partiale.

2.2 Map-Reduce - Cerinte pentru problema propusa

Dandu-se un set de ND documente si un set de NC cuvinte de cautat in documentele respective, sa se determine primele X documente cu relevanta maxima pentru cautare.

Prin document cu relevanta maxima intr-o cautare dupa un set de cuvinte cheie se intelege documentul in care cuvintele date au frecventa cea mai mare de aparitie

Frecventa de aparitie a unui cuvant la nivel de document:

frecventa = (nr_aparitii_cuvant / nr_total_cuvinte) *100 [%]

Pentru cerinta descrisa mai sus se considera:

- **[MAP]:** Impartirea fisierelor se va face in fragmente de cate aproximativ D octeti (cu exceptia ultimului fragment, care poate fi mai scurt).

Observatie: Poate aparea problema ca fragmentul prelucrat de un worker sa se termine sau sa inceapa in mijlocul unui cuvant. Se va adopta urmatoarea conventie: daca fragmentul incepe in mijlocul unui cuvant, worker-ul va "sari peste" acel cuvant; daca fragmentul se termina in mijlocul unui cuvant, worker-ul va prelucra si acel cuvant. In acest mod, cuvintele care sunt "la granita" dintre doua fragmente vor fi prelucrate tot timpul de worker-ul ce se ocupa de fragmentul care este in fisier inaintea cuvantului respectiv.

Astfel un task de tip “map”:

- poate primi ca input : numele fisierului, pozitia de inceput din fisier, si pozitia de sfarsit
- intoarce ca output: perechi de tip (cheie, valoare), in cazul problemei de fata: (nume document, vector partial). Vectorul partial poate contine setul de cuvinte impreuna cu ~~frecventele acestora~~ numarul de aparitii ale acestora.

- **[REDUCE]:** In cazul problemei propuse, operatia "reduce" se face in 2 etape:
  - In **prima** operatie de "reduce" : se vor aduna numarul de aparitii ale cuvintelor din vectorii obtinuti pentru fiecare parte dintr-un document, si se pastreaza primele N cuvinte cu cele mai mari frecvente.

Observatie: Daca sunt mai multe cuvinte care au aceeasi frecventa cu a ultimului cuvant din cele N relevante, se vor memora toate

  - **A doua** operatie de tip "reduce": se determina documentele care contin in vectorul de cuvinte cu frecvente maxime de aparitie, toate cele NC cuvinte cheie primite (cuvintele dupa care se face cautarea). Este necesar ca toate cuvintele cheie sa apara in vectorul de cuvinte cu frecv maxime pentru toate cele X (sau mai putine - in cazul in care nu exista cel putin X documente cu propritetatea de mai sus) documente ce vor fi afisate ca rezultat in final.

2.3 Observatii Generale:

- Avand la dispozitie memoria partajata intre thread-uri, rezultatele operatiilor "map" vor fi tinute in memorie; in mod normal ele s-ar fi scris si pe disc.

- Ca mod de executie, puteti folosi (desi nu este obligatoriu) obiectele de tip "thread pool" care sunt deja implementate in Java (vezi interfata*ExecutorService*); astfel, un thread worker poate prelucra mai multe task-uri.

- Pentru simplificare puteti utiliza mai multe thread pool-uri – de ex. unul pentru operatiile de tip "map", si unul pentru operatiile de tip "reduce".
- Cuvintele pot fi de orice dimensiune si pot contine doar litere

- Compararea intre cuvinte nu va fi case sensitive (veti transforma toate cuvintele in cuvinte cu litere mici inainte de a le prelucra).

- Afisati frecventele cu 2 zecimale, obtinute prin trunchiere (nu prin rotunjire).

3. Formatul datelor de intrare/ieşire.

Programul va avea ca argumente in linia de comanda: NT (numarul de thread-uri worker), numele unui fisier de intrare si numele unui fisier de iesire (in aceasta ordine).

Observatie: Se vor porni NT thread-uri de tip Map, respectiv NT thread-uri de tip Reduce.

Fisierul ce contine datele de intrare are urmatorul format:

- *pe linia I*: numarul NC de cuvinte cheie de cautat
- *pe linia II*: cele NC cuvinte cheie despartite prin spatiu
- *pe linia III*: dimensiunea D (in octeti) a fragmentelor in care se vor imparti fisierele
- *pe linia IV:* numarul N (cele mai frecvente N cuvinte retinute pentru fiecare document)
- *pe linia V:* numarul X reprezentand cate documente vreau sa primesc ca raspuns (ex. vreau sa mi se returneze primele X documente cele mai relevante pentru cautare)
- *pe linia VI:* numarul ND de documente de tip text de indexat si in care se va face cautarea
- *pe urmatoarele ND linii:* numele celor ND documente (cate unul pe linie)

Observatie: Toate fisierele de intrare vor contine doar caractere ASCII.

In **fisierul de iesire**, pentru setul de cuvinte cheie de cautat se vor afisa numele primelor X documente cu relevanta maxima pentru cautarea facuta -- fiecare nume pe cate un rand, in ordinea aparitiei lor in fisierul de intrare -- impreuna cu frecventele cuvintelor cautate.

Formatul fisierului de iesire este urmatorul:

Rezultate pentru: (cuvant_1, cuvant_2, …, cuvant_NC)

DOCUMENT_1 (frecventa_1, frecventa_2, …, frecventa_NC)
…
DOCUMENT_X (frecventa_1, frecventa_2, …, frecventa_NC)

, unde frecventa_i este frecventa cuvantului cuvant_i

4. Teste.

Pentru a verifica functionalitatea temei puteti folosi aceste teste .

5. Resurse (opţional)

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Alt articol introductiv despre MapReduce

MapReduce on Wikipedia