Jedna realizacija distribuiranog Map-Reduce sistema

Ilija Bašičević, Miroslav Popović, Sabina Jovanović, Branislav Drapšin, Boris Mlikota, Pavle Kuzevski

***Sadržaj* — Map - Reduce je afirmisana tehnika za obradu podataka koja se često koristi. Cilj ovog rada je razvoj sopstvene izvedbe distribuiranog Map - Reduce sistema. Prikazana je izvedba realizovana korišćenjem programskog jezika C++. Objašnjena je arhitektura sistema i neka korišćena rešenja. Dve konkretne primene ovog sistema su množenje matrica i brojanje reči. Data je procena performanse na osnovu ta dva eksperimenta.**

***Ključne reči* — Distribuirani sistemi, Map-Reduce, paralelna obrada, performansa paralelizacije.**

I. UVOD

PARALELIZACIJA obrade podataka na skupu povezanih računara je često vrlo složen posao koji sa sobom nosi probleme vezane za raspoređivanje zadataka, prenos podataka između računara kao i rukovanje otkazima. Kao jedno od rešenja pojavljuje se stvaranje jednostavnog okruženja koje bi od korisnika (programera) skrivalo sve detalje vezane za paralelizaciju. U ovom radu je opisano jedno rešenje okruženja koje se zasniva paradigmi preslikavanje-svođenje (map reduce) preuzetoj iz funkcionalnog programiranja (Lisp programski jezik).

Korisniku okruženja se pruža mogućnost da razvije funkcije za preslikavanje (map) i svođenje (reduce) koje izvode obradu nad podacima definisanim u obliku parova ključa i vrednosti (key-value pairs). Prva funkcija kao rezultat ima skup podataka koji su takođe u obliku parova ključa i vrednosti. Ovi međurezultati se prosleđuju drugoj funkciji koja izvodi obradu nad njima i daje konačan rezultat.

Glavna prednost ovakvog modela ogleda se u tome što se paralelizacija obrade automatizuje [1], kao i što se pomoću jednostavne sprege prema korisniku od njega sakrivaju kompletni detalji vezani za prethodno pomenute probleme pri paralelizaciji. Ovakvim pristupom može se obuhvatiti veliki broj obrada, međutim, na korisniku je da prilikom pisanja funkcija vodi računa o njihovoj ispravnosti.

Zahvalnica: Ovaj rad je delom finansiran od Ministarstva za nauku, Republike Srbije, projekat 12004 od 2008. god.

I. Bašičević, e-mail: ilija.basicevic@krt.neobee.net
M. Popović, e-mail: miroslav.popovic@krt.neobee.net
S. Jovanović, e-mail: sabina.jovanovic@dmsgroup.rs
B. Drapšin, e-mail:branislav.drapsin@dmsgroup.rs
B. Mlikota, e-mail: boris.mlikota@yahoo.com
P. Kuzevski, e-mail: pavle.kuzevski@rt-rk.com

Fakultet Tehničkih Nauka u Novom Sadu, Trg Dositeja Obradovića 6, Srbija (telefon: 381-21-4801100; Faks: 381-21-450721).

II. SPECIFIKACIJA PROGRAMSKOG MODELA

Distribuirani sistem čine upravljački, proces preslikavanja i svođenja čiji je broj statički određen i ne menja se tokom izvršavanja. Realizovana infrastruktura obezbeđuje automatsko pokretanje procesa preslikavanja i svođenja nakon što korisnik pokrene aplikaciju na računaru na kome je pokrenut upravljački proces.

Za realizaciju je korišćen ANSI C++ programski jezik u okviru Microsoft Visual 6.0 razvojnog okruženja. Za komunikaciju između procesa se koristi TCP/IP Winsock [2] biblioteka. Obavljeno je ispitivanje korišćenjem CPP Unit radnog okruženja [3].

Realizacija u velikoj meri ispunjava sledeće zahteve:

1. Prenosivost koda.
2. Robusnost.
3. Dinamička raspodela zadataka.
4. Performanse:
   - Upotreba indeksne strukture umesto standardne liste (brojanje reči).
   - Pretvaranje iz ASCII u celobrojni tip podataka (množenje matrica).
5. Aplikacija izveštava korisnika putem Hypertext Markup Language (HTML) izveštaja.

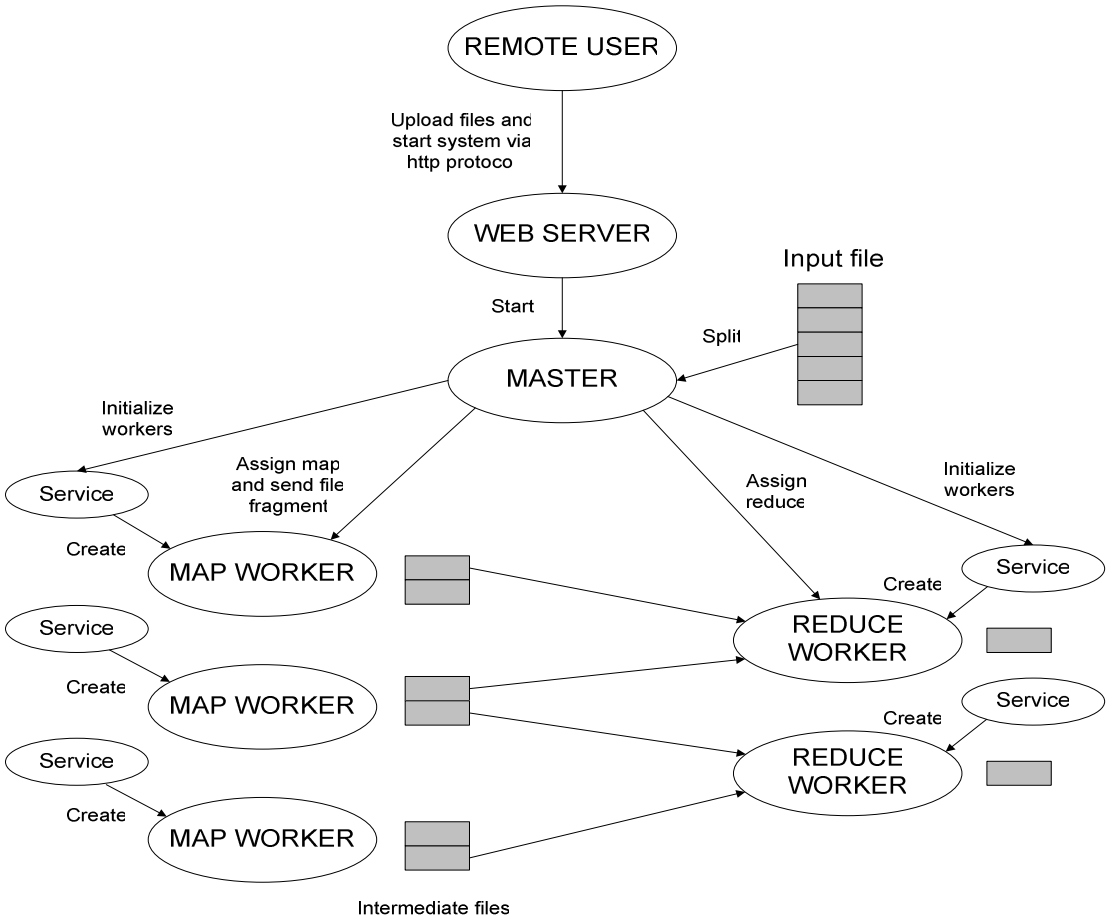
III. OPIS REŠENJA

U sistemu postoje tri fizički odvojene celine:

1. Upravljački proces.
2. Procesi preslikavanja.
3. Procesi svođenja.

Pri projektovanju sve tri celine se koristi pristup kojim se odvaja komunikacioni podsistem od jezgra koje sadrži logiku te celine. Komunikacioni podsistem je zajednički za sve tri celine. Infrastrukturni podsistem za podršku automatskog pokretanja takođe koristi komunikacioni podsistem.

Sva tri tipa procesa (upravljački, proces preslikavanja i svođenja) realizovani su u okviru zasebnih klasa. Svaki objekat tih klasa je realizovan kao automat stanja.



Sl. 1. Model sistema

Komunikacija između procesa realizovana je pomoću komunikacione sprege (interface) koji sakriva od korisnika konkretan način realizacije mrežnog sloja. Pri prelasku na drugi operativni sistem (koji ne podržava Winsock biblioteku) potrebno je izmeniti postojeći komunikacioni sloj.

Upravljački proces obavlja podelu ulazne datoteke i aktivira procese preslikavanja. Procesi preslikavanja, nakon prijema poruke od upravljačkog procesa za početak rada, prenose deo ulazne datoteke u lokalnu memoriju svog računara, obavljaju korisničku funkciju preslikavanja i obaveštavaju upravljački proces da su završili obradu porukom u kojoj se nalaze adrese lokacija u kojoj su zapisani međurezultati (Intermediate data). Proces preslikavanja zatim prelazi u stanje čekanja na zahtev za datotekom sa međurezultatima od strane procesa svođenja. Kada svi procesi preslikavanja završe rad, prelazi se u drugu fazu u kojoj se obavlja proces svođenja. Upravljački proces aktivira procese svođenja slanjem poruke koja sadrži celu mrežnu putanju do datoteke sa međurezultatima. Po prijemu poruke proces svođenja šalje zahteve svakom procesu preslikavanja ponaosob i od njih dobija datoteke sa ulaznim podacima. Obavlja se obrada pristiglih datoteka, razultat obrade se smešta u izlaznu datoteku i slanjem poruke se obavestava upravljački proces o završetku rada procesa svođenja.

A. Dinamička raspodela zadataka

U sistemu je realizovana dinamička raspodela zadataka procesima preslikavanja, tako da je svaki u mogućnosti da obavlja proizvoljan broj operacija preslikavanja, a upravljački proces ima ulogu da podeli ulaznu datoteku na proizvoljan broj delova, koje dostavlja pojedinim procesima preslikavanja radi obrade. Osim toga svaki od tih procesa vrši segmentaciju svojih izlaznih podataka. Kriterijum za segmentaciju izlaznih podataka zavisi od tipa obrade koja se izvršava, a razlog za segmentaciju izlaznih podataka je smanjivanje mrežnog saobraćaja.

B.Automatsko pokretanje procesa

Kao podrška infrastrukturi realizovana je aplikacija za automatsko pokretanje procesa preslikavanja i svođenja. Aplikacija se automatski pokreće pri podizanju operativnog sistema i po zahtevu upravljačkog procesa pokreće zahtevani proces. Za realizaciju udaljenog pokretanja sistema iskorišćeno je rešenje jednostavnog *web* usluživača (server), koji je razvijen na odseku i korišćen u svrhu računarskih vežbi. U okviru ovog rešenja realizovana je HTML stranica sa jednostavnom formom

koja omogućava unos željenih podataka kao i njihovo prosleđivanje na udaljeni računar. Takođe je realizovan i servlet koji obrađuje zahtev klijenta, pokreće sistem (upravljački proces) i klijentu šalje HTML stranicu sa vezom na izveštaj.

Kada korisnik koji pristupa *web* usluživaču pošalje svoj zahtev, pokreće se upravljački proces kome se prosleđuju ulazni podatci, a nakon toga korisniku se šalje stranica sa vezom do izveštaja. Sistem je takođe moguće pokrenuti direktnim pokretanjem upravljačkog procesa.

Na početku svog rada upravljački proces iščitava konfiguracionu datoteku i šalje poruke modulima infrastrukture koji vraćaju odgovor kao potvrdu spremnosti za početak rada. Nakon prijema odgovora upravljački proces šalje datoteku sa podacima vezanim za infrastrukturu mreže (IP adrese i portovi računara koji učestvuju u obradi). Po prijemu ovih datoteka infrastrukturni modul stvara zahtevani tip procesa (preslikavanja ili svođenja) kome se prepušta dalja komunikacija sa upravljačkim procesom. Dalja komunikacija upravljačkog i procesa preslikavanja ili svođenja se odvija bez učešća infrastrukturnog modula.

C.Množenje matrica

Realizacija algoritma za množenje matrica zasniva se na podeli posla između procesa preslikavanja i svođenja, tako da procesi preslikavanja obavljaju množenje elemenata odgovarajuće vrste i kolone, a potom sabiranje rezultata množenja, dok procesi svođenja treba da sortiraju redove u rezultujuću matricu. Jedna matrica se šalje cela svim procesima preslikavanja, a druga se deli na zadatke koji se dinamički raspoređuju.

Upravljački proces na početku rada određuje koliko će biti zadataka na osnovu broja linija (gde linija pretstavlja jednu vrstu matrice) u zadatku i dimenzije matrice. Upravljački proces obavlja i podelu posla između procesa svođenja tako što odredi koliko zadataka će svaki od njih sortirati. U okviru jednog zadatka procesa svođenja redovi rezultujuće matrice su sortirani, ali zadaci međusobno nisu sortirani. Procesi svođenja obavljaju sortiranje zadataka. Nakon što obavi podelu posla između procesa preslikavanja i svođenja, upravljački proces šalje poruku za početak rada svim procesima preslikavanja i obrada se dalje odvija na način prethodno opisan u prethodnom poglavlju.

Procesi preslikavanja primaju od upravljačkog i informacije o tome koji zadaci su namenjeni kom procesu svođenja. Svaka linija u zadatku množi se sa svakom kolonom matrice i rezultat se upisuje kao jedan element u niz veličine linije zadatka, odnosno veličine jedne vrste rezultujuće matrice. Emitovanje rezultata obavlja se na nivou linije zadatka. Svaki proces preslikavanja stvara onoliko datoteka sa međurezultatima koliko ima procesa svođenja, a na osnovu rednog broja zadatka i informacija koje je dobio od upravljačkog procesa zna u koju datoteku upisuje rezultat obrađivanog zadatka. Pored datoteke sa međurezultatima, za svaki proces svođenja proces preslikavanja stvara još jednu, konfiguracionu datoteku u koju upisuje redni broj zadatka koji je obradio i broj linija u zadatku čije rezultate obrade upisuje u datoteku sa međurezultatima. Konfiguraciona datoteka je potrebna jer se usled dinamičke raspodele zadataka ne zna unapred koji zadatak će proces preslikavanja dobiti, a bez te informacije ni proces svođenja ne može da obavi sortiranje. Proces preslikavanja šalje odgovarajuću kofiguracionu datoteku procesu svođenja koji je poslao zahtev za datotekom, a zatim i datoteku sa međurezultatom. Na osnovu ovih informacija proces svođenja obavlja sortiranje u rezultujuću matricu.

Matrice se nalaze u odvojenim datotekama, tako da jedna linija sadrži elemente jedne vrste matrice. Kako bi se izbegle konverzije iz ASCII u celobrojni tip podataka prilikom množenja matrice, koristi se binarni zapis. Matrica koja se šalje cela, na strani procesa preslikavanja se po prijemu pretvara u binarnu i tek tada se obavlja množenje.

D.Brojanje reči

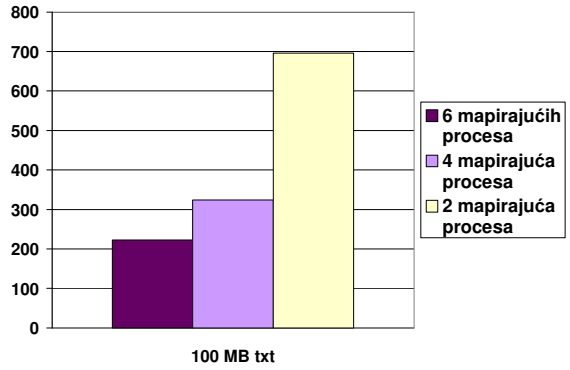
Upravljački proces na osnovu broja procesa svođenja određuje koliko će datoteka sa međurezultatima praviti svaki proces preslikavanja. Dinamička raspodela zadataka realizovana je tako da procesi preslikavanja u svakoj iteraciji dobavljaju deo ulazne datoteke nad kojim obavljaju preslikavanje. Preslikavanje se sastoji u brojanju istih pojava reči i sortiranju reči u datoteku sa međurezultatima. Svaki proces preslikavanja pravi onoliko datoteka sa međurezultatom koliko ima procesa svođenja. Kada se završi proces preslikavanja, procesi svođenja dovlače od svih procesa preslikavanja datoteke sa međurezultatima, sumiraju iste pojave reči u svim datotekama, sortiraju reči i upisuju u izlaznu datoteku.

E. Funkcija izveštavanja

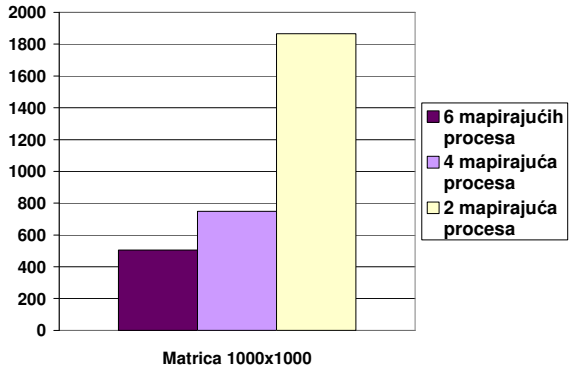
Osnovna uloga HTML izveštaja je da korisniku omogući uvid u trenutno stanje zadatka koji se izvršava na datom skupu povezanih računara. Sam izveštaj se generiše na računaru na kojem je pokrenut upravljački proces, a sastoji se od jedne datoteke (*.html* ekstenzije) koja može biti otvorena bilo kojim *Web browser*-om. U samoj datoteci se nalaze sve korisniku bitne informacije o napretku zadatka koji se trenutno izvršava uključujući i razne statističke podatke (npr. prosečna vremena obrade segmenta podataka na svakom pojedinom računaru itd.) kao i procena uspešnosti paralelizacije same obrade podataka koja je dostupna na kraju izvršavanja.

F.Rezultati obrade

Kao ilustracija poslužiće dva slučaja ispitivanja. U prvom slučaju (Slika 2) sistem vrši pretragu i sortiranje reči u tekstualnoj datoteci od 100 MB i pri tom se sistem konfiguriše da koristi 2, 4 i 6 računara kao procese preslikavanja. Kod drugog slučaja (Slika 3) sistem množi dve matrice dimenzija 1000x1000 i pri tom se kao i u prethodnom slučaju konfiguriše za 2, 4 i 6 procesa preslikavanja. U oba slučaja u obradi učestvuju dva procesa svođenja. Povećanjem broja računara ukupno vreme obrade se smanjuje(vertikalna osa je ukupno vreme obrade u sekundama).



Sl. 2. Prikaz vremena brojanja reči.



Sl. 3. Prikaz vremena množenja matrica.

IV. ZAKLJUČAK

U radu je prikazana jedna realizacija distribuiranog sistema zasnovanog na paradigmi preslikavanje-svođenje (map-reduce). Ta paradigma se koristi za paralelizaciju obrade kod nekih zadataka. Ovde su kao slučajevi ispitivanja realizovani množenje matrica i brojanje reči. Rezultati ukazuju na uspešnost primene paradigme preslikavanje-svođenje u tim slučajevima.

Dalji pravci razvoja su:

- Poboljšanje komunkacionog sloja radi postizanja boljih performansi pri prenosu podataka.
- Poboljšanje *web* usluživača, i omogućavanje pristupa sistemu od strane više korisnika u isto vreme.
- Implementacija dodatnih korisničkih funkcija (npr. *Combine* [1],[4]).
- Povećanje robusnosti sistema, tj. rukovanje otkazima računara i mrežne infrastrukture (*fault tolerance* [5]).

LITERATURA

[1] MapReduce: Simplified data processing on large clusters, Jeffrey Dean and Sanjay Ghemawat, Google Inc.
[2] http://msdn.microsoft.com/en-us/library/ms740673.aspx.
[3] CPP unit: http://cppunit.sourceforge.net/
[4] Google's MapReduce Programming Model—Revisited, Ralf L¨ammel, Data Programmability Team, Microsoft Corp., Redmond, WA, USA
[5] Evaluating MapReduce for Multi-core and Multiprocessor Systems, Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, Christos Kozyrakis Computer Systems Laboratory Stanford University

ABSTRACT

Map - Reduce is a well known technique for data processing which is widely used on large clusters. The purpose of this work is development of our implementation of distributed Map-Reduce system. This paper describes one developed in C++ language. System architecture and some design concepts are presented. Two typical applications of Map-Reduce system are multiplication of matrices and word counting. Performance evaluation based on executed experiments is also presented in the paper.

A REALIZATION OF DISTRIBUTED MAP REDUCE SYSTEM

Ilija Bašičević, Miroslav Popović, Sabina Jovanović, Branislav Drapšin, Boris Mlikota, Pavle Kuzevski.