UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA

Izveštaj

Kandidat: Vedran Bajić

Broj indeksa: SV10/2023

Predmet: Paralelno Programiranje

Tema rada: Paralelni web sakupljač

Mentor rada: dr Dušan Kenjić

Novi Sad, Septembar 2025

Uvod

U okviru projekta implementiran je paralelni web sakupljač podataka (web scraper), čiji je cilj efikasno preuzimanje i analiza sadržaja web stranica.

Projekat koristi:

- Intel Threading Building Blocks (TBB) biblioteku za paralelizaciju,
- CPR biblioteku za HTTP zahteve,
- Gumbo biblioteku za parsiranje html dokumenta

Analizirani su podaci sa sajta https://books.toscrape.com/index.html, koji je namenjen za testiranje web sakupljača i slobodno dozvoljava ovakve eksperimente.

Glavni cilj bio je da se kroz paralelizaciju ubrza proces obrade: od preuzimanja HTML stranica, preko parsiranja sadržaja, do analize i skladištenja rezultata.

Tok algoritma

Možemo podeliti projekat na 3 faze.

1. Prva faza predstavlja skidanje html dokumenta sa web stranica. To podrazumeva uzimanje url-ova koje je korisnik uneo, slanje get zahteva na svaku stranicu, i skladištenje dobijenih podataka

2. Druga faza je parsiranje html dokumenta i izdvajanje objekata book iz stranica. Skladištenje objekata, i obezbeđivanje robusnosti i ispravnosti algoritma.

3. Treća faza predstavlja analizu. Analiziranje niza knjiga, i prikaz rezultata, ispisivanje u izlaznu datoteku.

Struktura projekta

Materijal je podeljen u četiri direktorijuma.

- Include folder: Sadrži sve potrebne header fajlove klasa koje naš program koristi
- Src folder: Ovde se nalaze implementacije klasa
- Result folder: U ovaj folder, u file result.txt se smeštaju rezultati analize knjiga
- Input folder: U ovom folderu je file sa zadatim url stranica koje želimo da obradimo

Spisak klasa

Book

Predstavlja objekat knjige koja se parsira. Pošto se nad knjigama rade analize, ova klasa sadrži potrebne atribute kao što su ocena knjige, naslov, cena knjige i dostupnost.

Downloader

Ova klasa služi da bi preuzela sve stranice sa interneta. Ima vector stringova, koji predstavljaju url, koji se inicijalizuje u konstruktoru.

Metoda get_pages(), prolazi kroz linkove i koristi cpr biblioteku za skidanje html stranice. Ukoliko je odgovor skidanja pogrešan, postoji retry logika.

Metoda fetch_with_retry vraća odgovor zahteva. Radi tako što ukoliko je zahtev odbijen, sačeka 500 milisekundi, i pošalje ga ponovo. To radi nekoliko puta, a zatim odustaje, i stranica nije uspešno fetchovana.

Metoda get_pages() koristi parallel_for za prolazak kroz sve linkove, i paralelno šalje zahtev svakom linku. Rezultat vraća u konkurentni vector, koji čuva stringove, i oni predstavljaju sirov html stranice.

Parser

Ovo je klasa koja služi da iz čistog html stringa, izdvoji nama potrebne podatke o knjigama, i čuva ih sve u konkurentnu strukturu podataka. U našem slučaju vector, koji čuva objekte Book.

Kao i Downloader, Parser sadrži listu stringova, i oni predstavljaju stranice.

Metoda parse_pages koristi parallel_for da bi paralelno parsirala stranicu po stranicu. Za parsiranje stranice poziva se metoda parse_book koja prima string kao input, a vraća objekte book u konkurentni vector result.

Metoda search_book koristi gumbo biblioteku za parsiranje stringa. Pošto html elementi ugnježdavanjem jedan u drugi predstavljaju neku vrstu stabla, za pretragu dokumenta ćemo koristiti poznat algoritam pretrage stabla DFS. Kada iz nekog čvora, obilazimo njegovu decu, to ćemo paralelizovati, jer se svako dete nalazi u posebnom podstablu, i možemo paralelno izvršavati pretragu podstabala.

Metoda parse_book prima pokazivač na čvor u kojem se nalaze podaci o našoj knjizi, i ekstrakuje te podatke, i pravi objekat Book, i zatim ubacuje taj objekat u konkurentni vektor result. Ona uglavnom koristi metoda i atribute klasa GumboNode, GumboElement, i pronalazi vrednosti odgovarajućih tagova.

Analyzer

Analiza knjiga se radi po 5 kriterijuma

- Broj knjiga ocenjen sa 5 zvezdica
- Prosečna cena svih knjiga
- Broj knjiga kojima naslov ima jednu reč
- Broj knjiga sa cenom manjom od 20
- Spisak 10 najjeftinijih knjiga

Ova metoda služi za konačnu obradu i analiziranje knjiga koje smo preuzeli. Ima dve metode, i konstruktor koji prima vector knjiga, i string koji predstavlja izlazni direktorijum.

Metoda analyze_books prolazi kroz sve knjige i vrši obradu.

Prve četiri analize su vrlo slične, kroz parallel reduce prolazimo kroz sve knjige i akumuliramo konačnu sumu.

Poslednja analiza koristi algoritam sličan merge sortu. Koristimo task group ponovo radi paralelizma, i rekurzivno pronalazimo 10 najjeftinijih knjiga u datom intervalu. Kasnije taj interval spajamo, i od 2 strane, leve i desne, spajamo u jednu.

Metoda write_to_file Ispisuje rezultate analize u izlazni fajl result/result.txt.

Zaključak

Projekat pokazuje kako se pomoću Intel TBB biblioteke može efikasno implementirati paralelni web sakupljač podataka.
Kombinacijom paralelnog preuzimanja, konkurentnog parsiranja i thread-safe skladištenja postignuto je znatno bolje iskorišćenje resursa i smanjeno vreme izvršavanja u poređenju sa sekvencijalnim pristupom.

Slanje zahteva i preuzimanje stranica se otprilike uradio za 0.1 sekundu, odnosno oko 10tak stranica po sekundi.

Parsiranje svih stranica i izdvajanje objekata knjige se izvrši brže, za oko 1-2 sekunde.