Tema de casă #1 – Partide politice

Responsabili temă: Eliana Andreica si Catalin Leordeanu

Data publicării: 25.10.2013
Data ultimei modificări a enunţului: 26.10.2013 (adaugarea fisierelor de test in directorul **resurse_tema1** si actualizarea limitei de timp pentru varianta seriala optimizata)
Termenul de predare: 8.11.2013, ora 23:55

1. Introducere. Descrierea problemei

In urma unei revolutii, maretul dictator al republicii Latveria a fost alungat din tara si in locul lui s-a incercat instaurarea democratiei. S-au infiintat mai multe partide politice si fiecare si-a numit un numar de senatori. In fiecare zi senatul se intruneste intr-o sala in care senatorii sunt asezati intr-o matrice de dimensiune NxN, fiecare scaun din sala ocupand un element din matrice. Un senator are o pozitie fixa (i,j) in sala, adica se afla pe linia i si coloana j din matricea de scaune.

Din pacate senatorii nu erau multumiti de culoarea lor politica (partidul din care faceau parte) asa ca la inceput isi schimbau preferintele intr-un mod haotic. Din aceasta cauza, presedintele senatului a introdus un set de reguli. Culoarea politica a unui senator este notata cu Ck pentru 0<k<Nc, unde Nc reprezinta numarul maxim de partide (culori) politice. Se accepta schimbarea partidului politic, dar numai daca se respecta urmatoarele reguli:

R1 : Fiecare senator poate avea o singura culoare politica pe parcursul unei saptamani

R2 : In fiecare saptamana, un senator are dreptul (si obligatia) sa-si schimbe culoarea politica, dar trebuie sa respecte regula R3.

R3 : Noua culoare politica a unui senator (i,j) este acea culoare Ck a *altui senator* de pe pozitia (i',j'), cu proprietatea ca (i',j') este cel mai apropiat senator de culoarea Ck de (i,j), iar (i',j') se gaseste la distanta mai mare fata de (i,j) decat cel mai apropiat senator de orice alta culoare Cm pentru orice m, cu 0<m<Nc, iar m!=k. Senatorii (i, j) si (i', j') trebuie sa fie neaparat diferiti - dar e posibil ca Ck sa coincida cu vechea culoare a senatorului (i,j)) Daca la un pas nu exista senatori dintr-o anumita culoare, atunci acel partid este desfiintat si nu va mai fi luat in considerare in saptamana urmatoare.

Explicatie formala R3:

Pentru un anumit senator (i,j)

Fie dmin((i,j), Ck) = distanta minima in saptamana SAPT, de la senatorul (i,j) la orice senator de culoarea politica Ck (fie (i', j') acest senator; senatorii (i,j) si (i',j') trebuie sa fie diferiti), pentru 0<=k<Nc;

Se va calcula cate o distanta dmin pentru fiecare culoare politica Ck, disponibila in saptamana SAPT.

Daca o culoare politica Ck nu mai exista in saptamana SAPT, sau singurul senator de culoarea Ck este chiar senatorul (i,j), puteti considera, pentru simplitate, dmin((i,j), Ck)=0.

Fie Cmax, culoarea politica pentru care: dmin((i,j), Cmax) = max0<=k<Nc, (dmin(i,j), Ck) (daca sunt mai multe culori politice care indeplinesc conditia, se va considera culoarea cu indicele minim)

Cmax va fi culoarea senatorului (i,j) in saptamana SAPT+1.

R4: Distanta intre doi senatori (i,j) si (i',j') se considera max(|i-i'| , |j-j'|).

R5: Senatorii actioneaza simultan, astfel:

a) intai fiecare senator calculeaza noua sa culoare politica

b) apoi, fiecare senator se inscrie in partidul cu noua culoare politica pe care a calculat-o

(Pentru a simula acest comportament, folositi doua zone de memorie diferite pentru a modela configuratiile din doua saptamani consecutive)

Numim configuratie matricea care contine culorile politice ale senatorilor intr-o anumita saptamana. Configuratia initiala corespunde finalului saptamanii 0, respectiv inceputului saptamanii 1. In configuratia initiala, sunt disponibile Nc culori politice si vi se garanteaza ca fiecare culoare este intalnita la cel putin doi senatori (cu trecerea timpului, e posibil ca unele partide sa se desfiinteze, insa

in virtutea democratiei, vor exista mereu cel putin 2 partide/culori). Trecerea dintr-o configuratie in alta se face conform regulilor de mai sus. Pentru a tine configuratia curenta si configuratia urmatoare se vor folosi 2 zone de memorie(vezi si regula R5).

2. Cerintele temei

Veti simula un numar de S saptamani de schimbare a culorii politice a senatorilor (configuratia initiala este configuratia de la inceputul saptamanii 1, prima schimbare a culorilor/partidelor conform regulilor de mai sus este la finalul saptamanii 1 – voi veti simula transferurile intre partide de la saptamana 1 la finalul saptamanii S). Cum unele partide s-ar putea sa dispara, dupa fiecare saptamana veti retine(**in paralel, la punctul 2 de mai jos**) si afisa (intr-o regiune seriala) cati senatori fac parte din fiecare partid. La final afisati configuratia de la sfarsitul saptamanii S (vedeti si sectiunea cu formatul fisierelor de intrare si iesire).

Se cer:

1.Implementarea seriala (secventiala) a acestei probleme, in variantele neoptimizata **(2p)** si optimizata **(1p)**. Pentru varianta optimizata, calculul distantelor minime pentru fiecare culoare politica, se va face cautand in zone “concentrice” din ce in ce mai mari, in jurul unui senator dat. (Nu se va parcurge de fiecare data toata matricea, daca nu e cazul).

Se acorda punctul pentru optimizare daca programul secvential da rezultate corecte pe toate testele (publice si nu numai), iar rularea pe testul cu N=100 dureaza sub 30 de secunde pe un sistem din laboratorul de APD (ED 120). Puteti implementa si alte optimizari de ordin algoritmic decat cea sugerata, daca intra in timp, vor fi acceptate. Nu se vor puncta optimizarile de compilator (O2, O3, etc...).

2.Implementarea paralela in C/C++ utilizand OpenMP. **(3p pentru paralelizare corecta + 1p daca scaleaza).** Veti paraleliza atat calculul noilor culori politice, cat si “numararea”(veti fi depunctati daca numararea se face complet in regiunea seriala) senatorilor cu fiecare culoare, dupa fiecare saptamana.

O implementare scaleaza daca performanta sa creste o data cu cresterea numarului de threaduri . De asemenea, varianta paralela cu 2 threaduri trebuie sa fie mai rapida decat varianta seriala (conditie valabila si daca implementati o versiune seriala optimizata – vezi mai sus). Punctul pentru scalare se obtine pentru **speedup MINIM (pentru versiunea optimizata, 2 threaduri): 1.4**. In cazul in care nu implementati versiunea seriala optimizata: **speedup MINIM(pentru 2 threaduri): 1.6**. Testele de scalabilitate se vor face intotdeauna cu numarul de threaduri <= numarul de threaduri hardware disponibile

3.Analizati performantele problemei implementate/paralelizate pentru un numar fix (suficient de mare; >=100) de saptamani si o anumita dimensiune N a matricii de senatori (N >= 100), variind numarul de threaduri (1, 2, 4, eventual 6, 8 daca aveti acces la arhitecturi cu atatea threaduri hardware). Trebuie sa testati cu cel putin 4 threaduri, daca nu aveti un sistem personal care are 4 threaduri hardware, puteti testa pe sistemele din laboratorul de APD (ED 120). Testati cu mai multe tipuri de scheduling si chunk.

Va trebui sa scrieti in Readme dimensiunea testului(nr de saptamani si N), timpii pentru cea mai buna executie pentru fiecare numar de threaduri(cel putin pentru 1, 2, 4) si speedup-ul obtinut (vezi definitie laborator 3). Mentionati de asemenea cu ce tip de scheduling si valoare chunk le-ati obtinut. **(2p)** Testele de la acest punct se vor face intotdeauna cu numarul de threaduri <= numarul de threaduri hardware disponibile. Inainte de a testa, fie algoritmul secvential, fie pe cel paralel, scoateti instructiunile de afisare nenecesare; in nici un caz sa nu aveti astfel de instructiuni intr-o regiune paralela.

4.Va trebui sa aveti un fisier Readme(cu explicatii pentru surse – explicatii detaliate in cazul surselor optimizate si paralelizate, si cerintele de la punctul 4). Readme-ul trebuie sa contina si specificatiile hardware/software ale sistemului pe care ati rulat: arhitectura, numarul de procesoare/core-uri/threaduri per core, viteza procesoarelor (eventual dimensiuni memorie, cache...) sistemul de operare pe care ati testat (distributie, versiune) si compilatorul (versiune).

Trebuie sa aveti si Makefile cu reguli de compilare si clean (o tema fara makefile *functional* care creeaza executabile diferite pentru toate sursele seriale si paralele din arhiva, se va depuncta cu 1p, chiar daca exista Readme). **(1p)** Arhivare NEAPARAT in format **zip**; numele arhivei va contine si numele studentului si grupa.

3. Formatul datelor de intrare/ieşire. Exemple de fisiere de intrare/iesire

Exemplu de executie:
./serial nr_saptamani fisin fisout
./paralel nr_saptamani fisin fisout
Programul paralel primeste aceleasi argumente la rulare, dar se vor seta variabile de mediu, una pentru numarul de threaduri si alta pentru tipul de scheduling(si dimensiune chunk).
Numarul de procesoare va fi identificat de variabila de mediu OMP_NUM_THREADS (nu setati manual in program numarul de procesoare).
Tipul de scheduling poate fi setat, impreuna cu chunk-ul si in interiorul programului la static, dynamic, guided sau runtime. Daca il setati “runtime”, el va fi specificat (impreuna cu chunk-ul) de variabila de mediu OMP_SCHEDULE. Folosire:

```
export OMP_SCHEDULE="tip[,chunk_size]"
unde tip se refera la tipul de schedule, chunk_size la dimensiunea chunk-ului(care e optionala; nu puneti paranteze drepte)
```

Exemple de fisiere de intrare si iesire: **in resurse_tema1**
Formatul fisierului de intrare este:

- •pe prima linie numarul N, reprezentand latura salii senatului,
- •pe urmatoarea linie numarul Nc, reprezentand numarul de partide (culori politice) care se afla initial in matrice
- •incepand de pe a treia linie, matricea reprezentand configuratia initiala: N linii cu cate N elemente fiecare (i=0,...,N-1 pt linii si i=0,...,N-1 pe coloane)

Rezultatele (distributia senatorilor pe culori politice dupa fiecare iteratie + configuratia finala) se vor scrie intr-un fisier de iesire care are urmatorul format (cu N latura terenului pe care se afla senatorii iar X numarul de luni):
- S linii a cate Nc elemente, reprezentand numarul de senatori din fiecare culoare, de la finalul fiecarei saptamani
- o matrice cu N linii cu cate N elemente fiecare => configuratia la sfarsitul saptamanii S

4. Masurarea timpului de rulare
Cronometrarea se va face cu comanda (din consola) time:
time ./executabil [parametri]
Din outputul comenzii time ne intereseaza valoarea din dreptul cuvantului "real".

real 0m1.287s
user 0m1.270s
sys 0m0.010s

Resurse (opţional)
• LLNL OpenMP Tutorial, http://www.llnl.gov/computing/tutorials/openMP/