Coordonatele punctelor întregi într-o hyper-sfera

Ciprian-Mihai Stafie

Inginerie software

Programare paralela si concurenta

Anul I, Semestrul I

Profesori

Gheorghe Ștefănescu și Ciprian Păduraru

Grupa 406

Cerință

Calculează numărul de puncte întregi dintr-o hyper-sfera care are dimensiunea N si raza R.

Descrierea problemei

Pentru că cerința proiectului a fost să folosim MPI sau familiarizarea cu tehnica folosită. Soluția nu este rudimentară, ci una bazată pe o arhitectura care imită efectul unui thread, sau mai bine zis îi maschează/deghizează comportamentul și aspectul, acelea fiind de a rula o funcție în paralel cu aceleași, sau alte, date de intrare pe mai multe nuclee, în cazul de față fiind înlocuite cu alte procese.

Soluție

Ideea a fost următoarea: de a calcula un număr de intervale egal cu numărul de taskuri cerute. Iar pentru fiecare sa fie calculat numărul întregi corespunzător.

S-a creat o funcție de backtracking care calculează pentru fiecare interval, de forma [ai, bi], numărul de elemente care pot fi incluse în sfera N-dimensionala după formula distantei euclidiene astfel încât să fie acoperite toate numerele care reușesc să nu treacă de pragul perimetrului acesteia.

Formula folosita: SUM(ai, i de la 1 la n) <= r*r, unde ai reprezintă coordonata pe fiecare axă în parte, indicele i aparținând/fiind în relație cu axa i.

Lungimea unui interval este calculata după formula:

(2 * t + 1)/[N_Tasks], unde t = partea întreagă a lui R, iar [N_Tasks] numărul de procese MPI folosite

Intervalele sunt calculate începând de la –[R] și sfârșind la [R].

[R] este partea întreagă a razei.

Astfel, pentru fiecare interval în parte sunt luate toate posibilitățile care reușesc sa nu facă inegalitatea prezentată mai sus să fie încălcată. În momentul în care se găsește un astfel de element se înaintează în nivel. Adică se ia pe rând, de la dreapta la stânga, toate valorile din intervalul posibil(acesta fiind [-[R], [R]]) astfel încât să respecte regula de mai sus. În momentul în care se găsește un element necorespunzător se resetează toate elemente dinaintea indicelui curent pentru a nu afecta verificarea(i.e.: se setează toate elemente la 0, aceasta fiind neutre, pentru că centrul/originea sistemului de coordonate este 0, iar adunarea pe Z formează un grup). Pentru eficiență se începe de la 0 și se îndreaptă către margine, in cazul de față fiind distanța de la centru la modulul razei pe fiecare axa și direcție, astfel fiind luate mai ușor în calcul variantele corecte.

Pentru fiecare încercare se verifică inegalitatea și în momentul în care este adevărată se adună la numărul de puncte găsite. În caz contrar se procedează ca mai sus.

În final se transmite primului proces numărul de puncte calculate și se însumează în acel proces, după care se afișează.

Complexitate temporala

Pentru fiecare interval acesta e: [lungimea intervalului]*[numărul de posibilități pentru celelalte n-1 dimensiuni]

Fie r = 2t+1, numărul de elemente de la [-R, R]

p = r/[N_Tasks], lungimea unui interval

Pentru o dimensiune avem r posibilități, deci pentru toate axele disponibile, conform teoriei probabilităților o sa fie r^n-1.

Complexitatea temporala o sa fie obținută după formula p*(r^(n-1)).

Ea aparține O(R^N/[N_Tasks])

Corectitudinea datelor de intrare

Pentru a primi rezultate, datele de intrare trebuie să fie corecte. Pentru a nu primi eroare va trebui să avem un număr de procese mai mic sau egal decât 2 * [N] + 1.

[N_Tasks] <= 2 * [N] + 1,   [N] <- Partea întreagă a lui N

[N_Tasks] <- Numărul de procese

Rezultate din teste

Pentru exemplele din carte:

N = 1

R = 25.5

51 puncte

Pentru 2 taskuri o sa avem intervalele [-25, 0), [0, 25], iar pentru 3: [-25, -8), [-8, 9), [9, 25].

N = 2

R = 2.05

13 puncte

Pentru 2 taskuri o sa avem intervalele [-2, 0), [0, 2], iar pentru 3: [-2, -1), [-1, 0), [0, 2].

N = 3

R = 1.5

19 puncte

Pentru 2 taskuri o sa avem intervalele [-1, 0), [0, 1], iar pentru 3: [-1, -0), [0, 1), [1, 1].