Algoritmul Cannon

Al doilea algoritm foloseste o multime de *m x m* procesoare dar fiecare din aceste task-uri calculeaza individual rezultatul inmultirii a unui rand de blocuri corespondente cu o coloana de blocuri pentru a obtine drept rezultat blocul $C_{i,j}$. Operatia este practic o abstractizare a operatitei de inmultire, adica in locul valorilor propriu zise se folosesc blocuri. Ideea a pornit de la implementarea inmultirii a doua matrici de dimensiune *n* in paralel folosind un grid de unitati de procesare. In cazul in care fiecare unitate de procesare ar fi calculat o valoare rezultat a matricii *C* s-ar fi ridicat doua probleme majore. Prima ar fi numarul mare de unitati de procesare necesare in cazul in care matricile ar fi fost de dimensiuni considerabile, iar a doua ar fi numarul variabil de unitati de procesare, precum si o limita superioara. Din acest motiv s-a abordat o solutie de inmultire pe blocuri, astfel avand chiar si un numar fix de procesoare (*m x m*) matricile se vor imparti in blocuri.

In acest caz fiecare unitate de procesare are nevoie de randul carespunzator de blocuri ale matricii *A* respectiv coloana corespunzatoare de blocuri ale matricii *B*. Considerand ca avem un numar de *m x m* procesoare iar matricile au dimensiunea *n*, dimensiunea blocului fiind $s = n/m$ se discuta in cele ce urmeaza problemele legate de eficienta din punctul de vedere al comunicarii.
Fiecare din cele m^2 procesoare trebuie sa primeasca un rand si o coloana de cate s^2 elemente, ceea ce ar aduce la un numar considerabil de date trimise pentru fiecare proces in parte. Am adoptat solutia trimiterii sub forma de broadcast a intregii matrici (atat *A* cat si *B*) din care fiecare proces este capabil sa-si extraga blocurile de care are nevoie, iar recompunerea matricii *C* se face cu o operatie de tip *gather*. In acest fel pentru matrici de dimensiuni rezonabile se economiseste efortul de transmitere in dauna excesului de memorie la nivelul fiecarui procesor care va trebui sa aiba cate o copie proprie a matricelor sursa (matricea rezultat este doar de dimensiunea unui bloc).

Pseudocodul algoritmului este prezentat in cele ce urmeaza:

```
[procesul parinte creeaza restul proceselor]
[procesul parinte trimite dimensiunile celorlalte procese]
[procesul parinte trimite broadcast ambii termeni matriciali
        ai inmultirii]
[procesele asteapta sa se sincronizeze - sa ajunga in acest
        punct toate]
[fiecare proces isi calculeaza blocul rezultat al inmultirii in
        felul urmator:
                - isi extrage randul de matrici blocuri
                - isi extrage coloana de matrici blocuri
                - efectueaza inmultirea
]
[se efectueaza operatia de gather pentru a cumula rezultatul
        in procesul parinte]
[procesul parinte rearanjeaza rezultatul]
```