

SMP - PROBLEME DE EXAMEN

EVALUAREA PERFORMANTELOR

1) Un sistem multiprocesor generează rezultate de patru categorii la ratele $R_1=1$ MFLOPS, $R_2=5$ MFLOPS, $R_3=10$ MFLOPS și $R_4=100$ MFLOPS. Considerând că 1% reprezintă rezultate de tipul 1, 10% de tipul 2, 4% de tipul 3 și restul de tipul 4, să se calculeze rata medie armonică ponderată a mașinii.

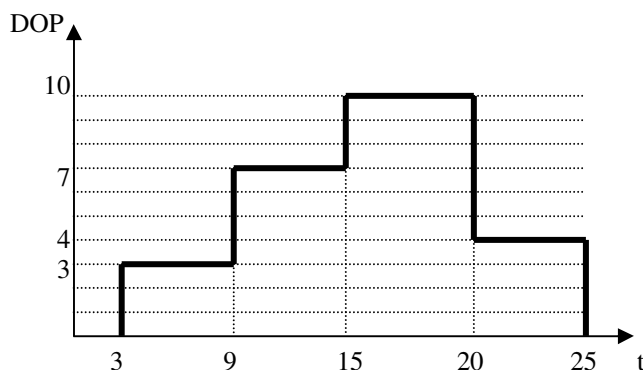
2) Într-un calculator paralel 30% din prelucrări sunt scalare cu rata de execuție de 5 MFLOPS, iar restul sunt vectoriale cu rata de execuție de 200 MFLOPS. Să se calculeze rata medie armonică ponderată a mașinii.

3) O problemă poate fi rezolvată pe un uniprocessor în timpul $T_1=1.5$ s, iar pe un sistem cu 8 procesoare în timpul $T_8=250$ ms. Să se calculeze accelerația și eficiența sistemului paralel.

4) Se consideră execuția a $N=1000$ de taskuri, fiecare necesitând $t=1$ ms pentru execuție pe un sistem uniprocessor. Pe un sistem cu $n=10$ procesoare execuția în paralel a acestor taskuri necesită un timp suplimentar de overhead $t_o=0.05$ ms și un timp de sincronizare (pentru trecerea la un grup nou de taskuri) $t_s=0.1$ ms. Să se calculeze factorul de accelerare și eficiența sistemului paralel. Să se determine valorile limită pentru S și E în cazul în care $n \rightarrow \infty$ (N fixat), respectiv $N \rightarrow \infty$ (n fixat).

5) a) Capacitatea de calcul pentru un procesor este aproximat cu rata MIPS. Să se calculeze rata MIPS pentru un program cu $4.8 \cdot 10^9$ instrucțiuni executat într-un timp de 500 ms. b) analog, pentru un procesor cu frecvența ceasului 1.6 GHz și 2 cicluri / instrucțiune.

6) Se consideră graficul DOP (grad de paralelism) pentru un sistem multiprocesor:



7) Se presupune $t_i = 1/i$ pentru $i = 1, 2, \dots, n$ (corespunde cazului ideal când un job unitate de timp este executat de i procesoare în timp minim). deci rata $R_i = i$, caci rata de execuție crește de i ori de la $R_1 = 1$ atunci când sunt utilizate i procesoare.

Se consideră distribuția de probabilitate în care

$$s = \sum_{i=1}^n i$$

și anume:

$$\pi_2 = (1/s, 2/s, \dots, n/s) \quad (\text{favorizează utilizarea mai multor procesoare})$$

Să se calculeze rata medie armonică ponderată.

8) Se consideră că pentru rezolvarea unei probleme într-un sistem uniprocessor sunt necesare $O(1)=2 \cdot 10^6$ operații unitare (atenție $T(1)=O(1)!!!$). Pentru aceeași problemă într-un sistem cu 100 de procesoare sunt necesare $O(n)=2.25 \cdot 10^6$ operații unitare care necesită un timp de execuție $T(n)=0.05 \cdot 10^6$ unități de timp. Să se calculeze factorul de accelerare S , eficiența E , redundanța R , utilizarea U și calitatea paralelismului Q .

RETELE DE INTERCONECTARE

1) Sa se proiecteze o retea pentru un multicalculator cu 64 de noduri, utilizand: a) tor 3D; b) hipercub 6D; c) retea cu cicluri cub conectate CCC cu un diametru minim.

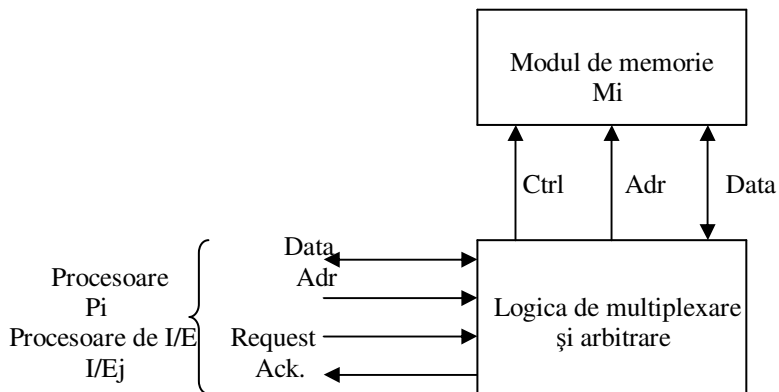
a) daca d =gradul de nod, D =diametrul, l =numarul total de legaturi, atunci se defineste calitatea retelei $q = l / (d \cdot D \cdot l)$. Sa se ordoneze solutiile dupa calitate.

b) distanta iternod medie este numarul mediu de legaturi strabatute de un mesaj intre oricare doua noduri (se considera toate perechile de noduri sursa si destinatie si se face media). Sa se ordoneze solutiile dupa distanta medie, daca se presupune ca un nod va trimite un mesaj la toate celelalte noduri la distanta i cu

probabilitatea $\frac{D-i+1}{\sum_{k=1}^D k}$.

2) Sa se proiecteze logica de acces (multiplexare, arbitrare, control) la o memorie multiport pentru patru procesoare P0-P3, avand prioritati diferite a) P0 este procesorul c.m.prioritar b) se utilizeaza algoritmul LRU (Least Recently Used).

3) Sa se proiecteze un punct de intersectie dintr-un comutator grila (crossbar switch) care conecteaza patru procesoare la un modul de memorie 256M*32 biti (procesorul P0 este cel mai prioritar). Se cere:



a) schema bloc a punctului de intersectie;

b) structura internă detaliată a punctului de intersectie (se vor utiliza magistrale);

c) organigrama unității de comandă (modulul de arbitrare).

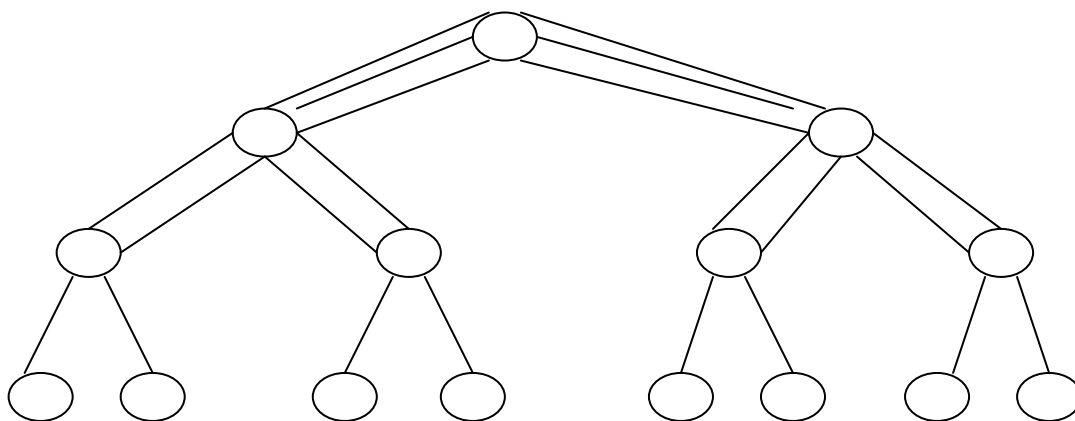
4) Sa se proiecteze detaliat un comutator elementar 2x2 (bloc de interschimb) care poate realiza patru tipuri de conexiuni, in functie de semnalele de control C0 si C1: directa (00), transm.inferioara (01), transm.superioara (10) si inversa (11).

5) Sa se proiecteze o retea baseline 16x16.

6) Sa se determine interconexiunile shuffle pentru un singur nivel 16x16. Sa se reprezinte retea.

7) Sa se determine interconexiunile butterfly de ordinul 2 (B_2) pentru un singur nivel 16x16. Sa se reprezinte retea.

8) Pentru arborele gros din figura sa se calculeze gradul de nod si diametrul.



9) Se considera un hipercub 4D. Sa se specifice intr-o tabela adresele (codurile) nodurilor hipercubului (cu ..., dar minim 8 noduri). Sa se listeze codurile tuturor nodurilor care pot fi accesate de la nodul (0000) in exact 2 pasi (la distanta 2). Sa se identifice un algoritm aplicabil si pentru un hipercub 10D si sa se denumeasca algoritmul.

10) Se considera un multicalculator cu 64 de noduri interconectate printr-o retea tor 3D. Se cere:

- Sa se furnizeze o solutie de organizare si de codificare (etichetare) a nodurilor.
- Sa se calculeze gradul de nod (d), diametrul (D) si numarul de legaturi (l). Argumentati.
- Sa se reprezinte (partial) reteaua cu nodurile etichetate.
- Dati trei exemple de perechi de noduri care comunica in timpul cel mai lung.

11) Se considera o retea ciclu cub conectata de dimensiune 4 (4D). Se cere:

- Sa se calculeze numarul de noduri (N), gradul de nod (d), diametrul (D) si numarul de legaturi (l).
- Sa se furnizeze o solutie de codificare (etichetare) a nodurilor.
- Sa se reprezinte reteaua (partial) cu nodurile codificate.

MULTICALCULATOARE

1) Sa se analizeze latentă de comunicare (variante simplă de analiză) în cazul mecanismelor de rutare „store-and-forward” și „whormhole” pentru lungimea pachetului $L=2$ Kbiti, lărgimea de bandă a canalului $w=1$ Mbit/sec, distanța între nodurile sursă și destinație $D=8$ (canale traversate), lungimea unui flit 8 biti.

2) Într-o rețea plasă 3D cu $8 \times 8 \times 8$ noduri să se determine ruta de la nodul $s=(0,1,7)$ la nodul $d=(3,3,5)$ pe baza algoritmului determinist de rutare xyz. Să se reprezinte traseul folosind două secțiuni plane prin rețea (notatii!). Să se calculeze traficul și latentă comunicatiei.

3) Într-un hipercub 4D se considera că nodul 1010 trimite un mesaj la toate celelalte noduri folosind algoritmul de broadcast bazat pe arbore de acoperire („spanning tree”). Să se reprezinte acest arbore și să se marcheze în cadrul hipercubului transmisia mesajului. Să se calculeze traficul și latentă comunicatiei.

4) Într-un hipercub 4D nodul 1001 trimite un mesaj la nodul 0010 pe baza unui algoritm de rutare E-cub. Să se determine toate nodurile intermediare și să se marcheze ruta pe un desen al rețelei. Să se calculeze traficul și latentă comunicatiei.

5) Să se realizeze un comutator cu 48 de canale (48 intrări și 48 ieșiri) utilizând comutatoare IMS C004 (32 canale) pentru un sistem cu transputere.

6) Sa se realizeze o comparatie intre cele doua categorii de baza de multicalculatoare (MPP si COW) sub forma unui tabel care sa cuprinda caracteristicile comparate ale acestora, sub forma:

| Caracteristica | MPP | COW |
|----------------|-----|-----|
| ... | ... | ... |
| ... | ... | ... |

PROGRAMAREA MULTICALCULATOARELOR

1) Sa se scrie o aplicatie paralela utilizand MPI pentru calcularea mediei aritmetice a elementelor unei matrici citite de la tastatura.

2) Sa se scrie o aplicatie paralela utilizand MPI pentru calcularea histogramei unei imagini cu niveluri de gri 256x256 pixeli, 8 biti/pixel, citita de la tastatura.

3) Sa se specifice un algoritm pentru matrici/vectori executat pe un hipercub de dimensiune d (cu 2^d procesoare).

MULTIPROCESOARE

1) Intr-un sisteme cu patru procesoare coerenta memoriilor cache este asigurata cu protocol Dragon. Sa se specifice starile la nivelul fiecărei memorii cache considerand succesiunea de operatii referitoare la acelasi bloc (initial blocul se gaseste numai in memoria principala): P3Rd, P0Rd, P1Rd, P2Wr, P2Rd, P0Rd, P0Wr, P1Rd.

2) Intr-un sisteme cu trei procesoare coerenta memoriilor cache este asigurata cu protocol Dragon. Sa se specifice starile la nivelul fiecărei memorii cache considerand succesiunea de operatii referitoare la acelasi bloc (initial blocul se gaseste numai in memoria principala): P0Rd, P1Rd, P0Wr, P2Rd, P1Rd, P0Wr, P2Wr.

3) Intr-un sisteme cu patru procesoare coerenta memoriilor cache este asigurata cu protocol MESI. Sa se specifice starile la nivelul fiecărei memorii cache considerand succesiunea de operatii referitoare la acelasi bloc (initial blocul se gaseste numai in memoria principala): P3Rd, P0Rd, P1Rd, P2Wr, P2Rd, P0Rd, P0Wr, P1Rd.

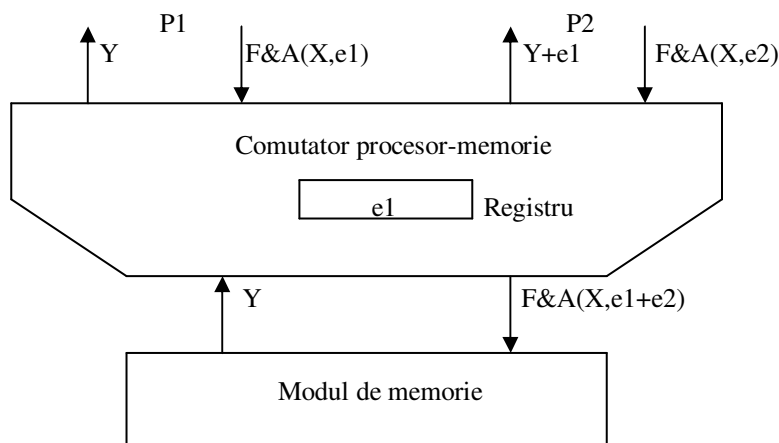
4) Sa se proiecteze unitatea de control a dispozitivului de acces la magistrala intr-un sistem multiprocesor cu patru procesoare cu magistrala partajata de tipul:

- a) cu inlantuire si rotirea prioritatilor;
- b) cu interogare in care unitatea de control utilizeaza un algoritm de tip LRU („Least Recently Used”);
- c) cu cereri independente in care unitatea de control utilizeaza un algoritm FIFO al cererilor (la cereri simultane P0 are prioritate maxima).

5) Sa se proiecteze interfata unei memorii multiport dintr-un sistem multiprocesor cu patru procesoare. Interfata arbitreaza cererile pe baza unui algoritm LRU. Se cere:

- a) schema bloc detaliata a interfetei;
- b) organigrama unitatii de control.

6) Sa se furnizeze o solutie pentru implementarea hardware a primitivei Fetch-and-Add (X,e) sau F&A(X,e) in cadrul unui switch procesor-memorie. X este o variabila partajata intreaga, iar e este o expresie intreaga. Se presupune ca exista doua cereri simultane, P_1 cu prioritate mai mare decat P_2 .



Se cere:

- schema bloc detaliata a switch-ului;
- organigrama UC.

7) Se considera un sistem multiprocesor cu patru procesoare. Intr-o variabila partajata procesoarele P0 si P1 lanseaza scrieri in ordinea:

- P0 scrie 0
- P0 scrie 1
- P1 scrie 2
- P1 scrie 3

Procesoarele P2 si P3 citesc aceasta locatie (patru citiri succesive de valori diferite fiecare procesor). Sa se indice toate variantele posibile de valori citite de cele doua procesoare in cazurile modelelor de consistenta a memoriei partajate: consistenta stricta, consistenta secventiala, consistenta de procesor.

8) Intr-un sistem multiprocesor 16 procesoare sunt conectate la 16 module de memorie printr-o retea omega. Sa se reprezinte schema. Adresa (codul) modulului de memorie se utilizeaza pentru routare prin retea. Sa se marcheze pe schema ruta pentru o operatie lansata de procesorul P11 de scriere in modulul de memorie M5 (explicatii)

9) Se considera un multiprocesor CC-NUMA cu 64 de procesoare cu cataloage solutie distribuita neierarhica bazata pe memorie. Memoria totala este de 64 GB, repartizata cate 1GB/nod, 64 octeti/bloc de cache. La un moment dat pot exista copii ale unui bloc la toate nodurile. Sa se reprezinte schema bloc (simplificata) a sistemului, sa se specifice structura cuvintului de adresa fizica si sa se faca un calcul pentru memoria necesara cataloagelor (in GB).

10) Se considera un multiprocesor CC-NUMA cu 512 de noduri (procesoare) cu cataloage solutie distribuita neierarhica bazata pe memorie. Blocul de cache este de 16 octeti/bloc de cache. La un moment dat pot exista copii ale unui bloc la toate nodurile. Sa se calculeze overhead-ul cauzat de memoria necesara cataloagelor. Varianta 1: dimensiune bloc 64 B / bloc cache. Varianta 2: dimensiune bloc 256 B / bloc cache.

11) Sa se proiecteze supraveghetorul de magistrala („bus snooper”) corespunzator unui procesor (unei memorii cache) in cadrul unui sistem multiprocesor. Se cere:

- schema bloc detaliata a supraveghetorului (inclusiv resursele corespunzatoare din memoria cache);
- explicarea functionarii;
- organigrama de functionare a supraveghetorului.

12) Sa se specifice o posibila solutie pentru arbitrarea magistralei cu patru agenti (procesoare) simetrici plus un agent prioritar, de tipul celei utilizate intr-un sistem multiprocesor cu microprocesoare Intel Pentium Pro. Se cere:

- structura detaliata a arbitrului la nivelul unui agent;
- explicarea functionarii schemei de arbitrare;
- organigrama functionarii arbitrului.

13) In cadrul modelului de consistenta secventiala a memoriei partajate sa se proiecteze switch-ul de acces la memoria uniport pentru patru procesoare de prioritati egale, avand in vedere numai operatiile de scriere:

- sa se defineasca dialogul intre procesoare si switch (schema bloc, semnale, specificarea semnalelor);
- structura switch-ului (schema bloc);
- organigrama UC a switch-ului.

PROGRAMAREA MULTIPROCESOARELOR

1) Utilizand operatorii P si V sa se specifice algoritmul prin care doua procese concurente A si B executa operatii curente intr-un cont bancar (depuneri si retrageri), contul fiind reprezentat printr-o variabila partajata. Fiecare proces ruleaza intr-o bucla in care citeste codul operatiei (0/1 depunere / retragere) si suma, executand apoi operatia respectiva.

2) Utilizand operatorii PE si VE sa se specifice algoritmul prin care patru procese P0-P3 utilizeaza mutual exclusiv cate doua resurse din patru (R0-R3), astfel:

P0: R0, R1;

P1: R1, R2;

P2: R2, R3;

P3: R3, R0.

Procesele au prioritati diferite, P0 fiind procesul cel mai prioritar.

3) Sa se scrie o aplicatie OpenMP in care se calculeaza histograma unei imagini binare (0,1) cu rezolutia 128x128 pixeli, imaginea fiind impartita in benzi de 16 linii si distribuite thread-urilor.

4) Sa se scrie o aplicatie paralela utilizand OpenMP pentru calcularea mediei aritmetice a elementelor unei matrici citite de la tastatura.

5) Sa se scrie o aplicatie cu thread-uri Windows in care se calculeaza histograma unei imagini binare (0,1) cu rezolutia 128x128 pixeli, imaginea fiind impartita in benzi de 16 linii si distribuite thread-urilor.

6) Sa se scrie o aplicatie paralela cu thread-uri Windows pentru calcularea mediei aritmetice a elementelor unei matrici citite de la tastatura.

7) Sa se construiasca graful de precedenta (de taskuri) pentru urmatorul program concurent:

```
begin
    S1; S2;
    cobegin
        S3; S4; S5
    coend;
    S6;
    cobegin
        S7;
        begin
            S8;
            cobegin
                S9; S10
```

```

coend;
S11
end;
S12
coend;
S13
end;

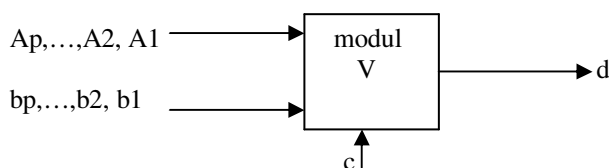
```

8) Sa se construiasca programul concurrent pentru urmatorul graf de precedenta (de taskuri).

SIMD

1) Sa se specifice o solutie de implementare a modulului de inmultire matrice-vector / matrice-matrice.

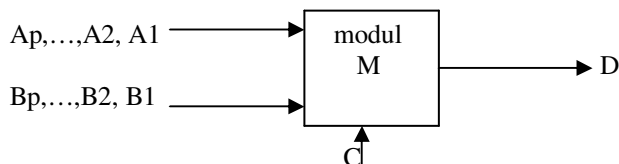
Modul de înmulțire matrice-vector:



$$d = c + \sum_{i=1}^p A_i b_i$$

unde $\{b_i \ i=1,2,\dots,p\}$, c și d sunt vectori coloană $m \times 1$, iar $\{A_i \ i=1,2,\dots,p\}$ sunt matrici $m \times m$.

Modul de înmulțire matrici:



$$D = C + \sum_{i=1}^p A_i B_i$$

unde $\{A_i, B_i \ i=1,2,\dots,p\}$, C și D sunt matrici $m \times m$.

In cazul utilizarii si altor tipuri de procesoare (decat cele de la curs) sa se specifice structura si functionarea acestora.

2) Sa se proiecteze o retea sistolica de procesoare pentru calcularea unei sume de matrici

$$D = c \cdot \sum_{i=1}^p A_i$$

unde A_i este o matrice $m \times m$, iar c este o constanta reala. Se cere:

- structura interna a unui procesor;
- structura si functionarea retelei.

3) Sa se specifice structura unui dispozitiv aritmetic sistolic (VLSI) care calculeaza convolutia a doi vectori a si b , fiecare element al vectorului rezultat fiind dat de:

$$c_i = \sum_{k=1}^i a_k b_{i+1-k}$$