



# Arhitectura Sistemelor de Calcul



**Universitatea Politehnica Bucuresti**  
**Facultatea de Automatica si Calculatoare**

[cs.ncit.pub.ro](http://cs.ncit.pub.ro)  
[curs.cs.pub.ro](http://curs.cs.pub.ro)



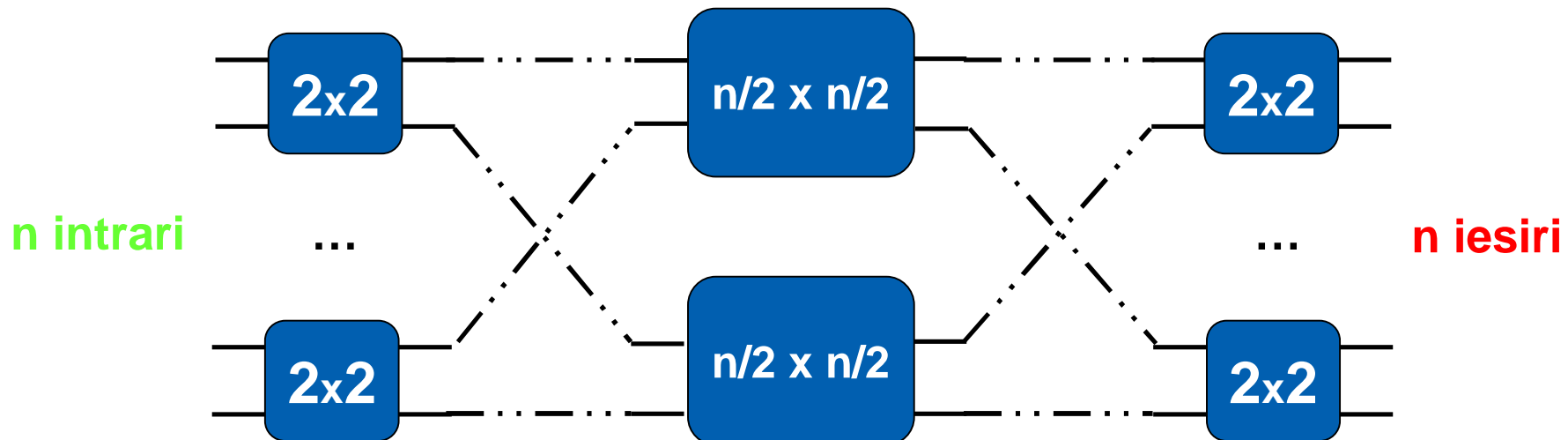
- Retele de Comutare Ierarhice
- Retele de Comutare de tip Delta
- Retele Bazate pe Rutare
- Performantele Retelelor de Comutare
  - Analiza Retelei de Tip CrossBar
  - Analiza Retelei de Tip Delta



# Retele de Comutare Ierarhice

3

- Realizate din CrossBar-uri de dimensiuni mici asezate pe mai multe nivele de interconectare
  - Crește timpul de întârziere datorită utilizării mai multor module – dar se menține  $\forall \text{intrare} \leftrightarrow \forall \text{iesire}$
- Exemplu – structura Beizer (Benes):
  - Retea  $n \times n$  realizată cu două module  $n/2 \times n/2$  (se pot împărți la rândul lor în subrețele mai mici) și  $4n$  module de tip  $2 \times 2$
  - Complexitatea este de  $(4n \log n - 2n)$





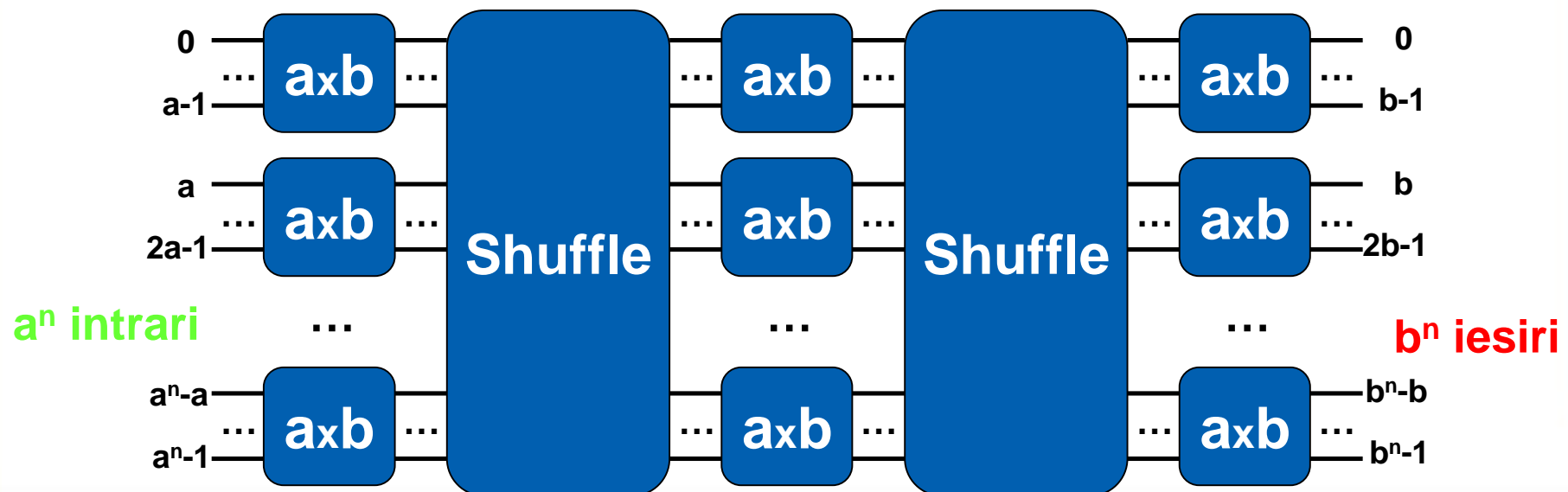
- Retele de Comutare Ierarhice
- Retele de Comutare de tip Delta
- Retele Bazate pe Rutare
- Performantele Retelelor de Comutare
  - Analiza Retelei de Tip CrossBar
  - Analiza Retelei de Tip Delta



# Retele de Comutare de tip Delta

5

- Sunt comutatoare ierarhice cu  $a^n$  intrari si  $b^n$  iesiri ce utilizeaza CrossBar-uri (CB)  $axb$  si retele de permutare de tip intercalare perfecta (Shuffle)
- Reteaua Delta are  $n$  nivele ierarhice:
  - Nivelul 1 contine  $a^{n-1}$  CB-uri de tip  $axb$ , pentru a conecta  $a^n$  intrari cu  $a^{n-1}xb$  iesiri
  - Nivelul 2 contine asadar  $a^{n-1}xb$  intrari conducand la  $a^{n-2}xb$  module CB





# Retele de Comutare de tip Delta

6

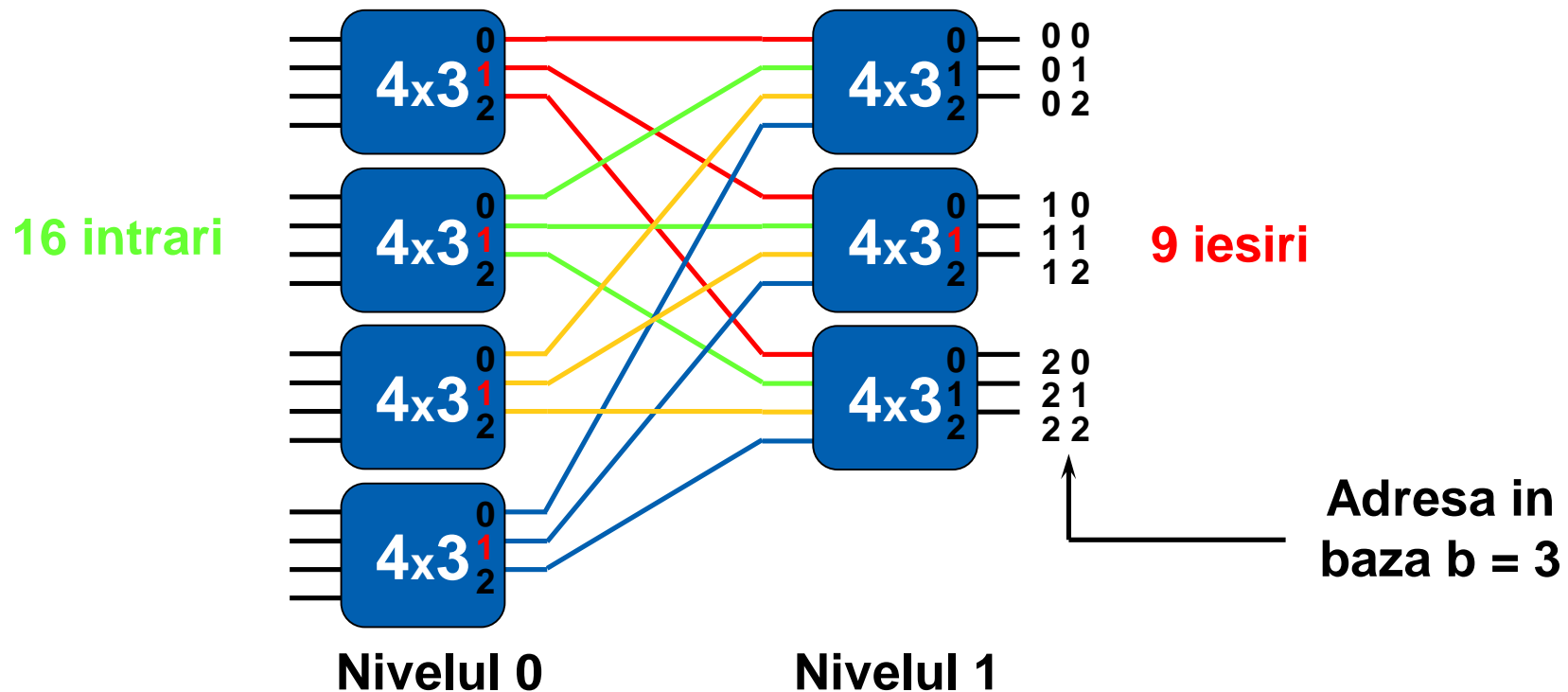
- Astfel, in general:
  - Nivelul  $i$  contine  $a^{n-i}b^{i-1}$  module CB de tip  $axb$
- Intreaga retea contine:
  - $(a^n - b^n)(a - b)$  CB-uri de tip  $axb$  cand  $a \neq b$
  - $nb^{n-1}$  CB-uri de tip  $axb$  cand  $a = b$
- Destinatia este  $D = (d_{n-1} \ d_{n-2} \ \dots \ d_1 \ d_0)$  cu  $0 \leq d_i \leq b$
- Cifrele de reprezentare in baza  $b$ , a adresei destinatiei vor controla modulul CB de pe nivelul respectiv ( $d_i$  controleaza CB-ul  $i$ )
- Functia Shuffle-ului este de a interconecta nivelele retelei Delta



## Exemplu – Delta 16x9

7

- Presupunem ca avem 16 procesoare si 9 memorii
- $16 = 4^2$  si  $9 = 3^2 \rightarrow$  folosim 2 nivele de CB 4x3
- Daca vrem acces la memoria 4 (11 in baza 3) atunci se merge pe 1 (niv 0) si 1 (niv 1)!

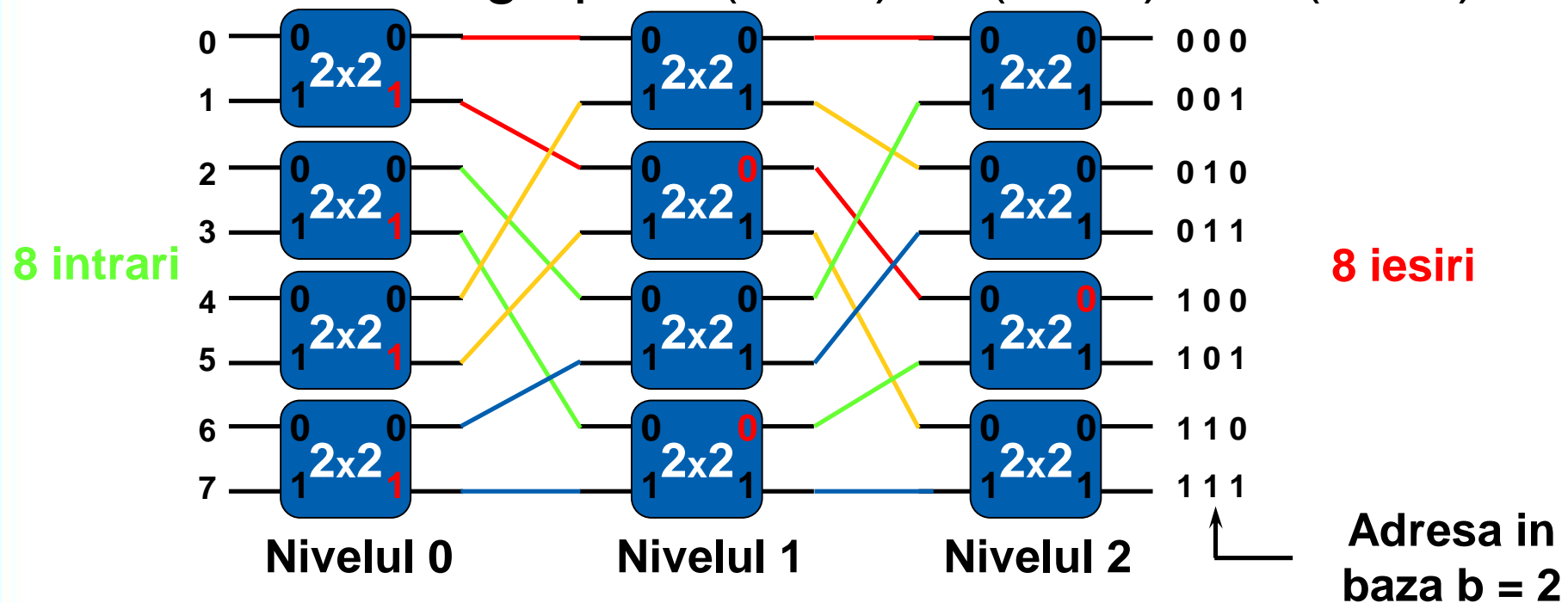




## Exemplu – Delta 8x8

8

- Presupunem ca avem 8 procesoare si 8 memorii
- $8 = 2^3 \rightarrow$  folosim 3 nivele de CB 2x2
- Daca vrem acces la memoria 4 (100 in baza 2) atunci se merge pe 1 (niv 0), 0 (niv 1) si 0 (niv 2)!



TA: Conexiunea unui Delta 8x8 cu corespondenta directa  $I_i \rightarrow O_i$





- Retele de Comutare Ierarhice
- Retele de Comutare de tip Delta
- Retele Bazate pe Rutare
- Performantele Retelelor de Comutare
  - Analiza Retelei de Tip CrossBar
  - Analiza Retelei de Tip Delta



# Retele Bazate pe Rutare

10

- Au ca nucleu legaturi directe intre o resursa si cele considerate vecine
- Resursele pot fi distincte dar toate au in comun un router ce se ocupa de comunicarea prin mesaje
- Traditional aceste retele sunt modelate prin grafuri cu urmatoarele proprietati de baza:
  - **Gradul** nodului (al resursei) – defineste numarul de canale de conexiune ale resursei la vecinii sai
  - **Diametrul** – distanta maxima intre doua resurse ale retelei
  - **Regularitatea** – nodurile/resursele care au/nu au acest grad
  - **Simetria** – o retea este simetrica daca “arata la fel” pentru orice nod
- Obs: acest tip de retea nu foloseste comutarea de circuite (CB), ci rutarea – legatura intre noduri/resurse e permisa doar cand este ceruta & permisa



## Retele Bazate pe Rutare – Caracteristici

11

- **Topologia** – definește modul în care nodurile sunt interconectate
- **Rutarea** – stabilește calea de date pe care un mesaj o urmează de la sursă la destinație
- **Comutarea** – mecanismul ce determină cum și când un canal de intrare este conectat la unul de ieșire
- Obs: trebuie avute în vedere
  - Alocarea de buffere
  - Controlul fluxului de date



# Retele Bazate pe Rutare – Topologii

12

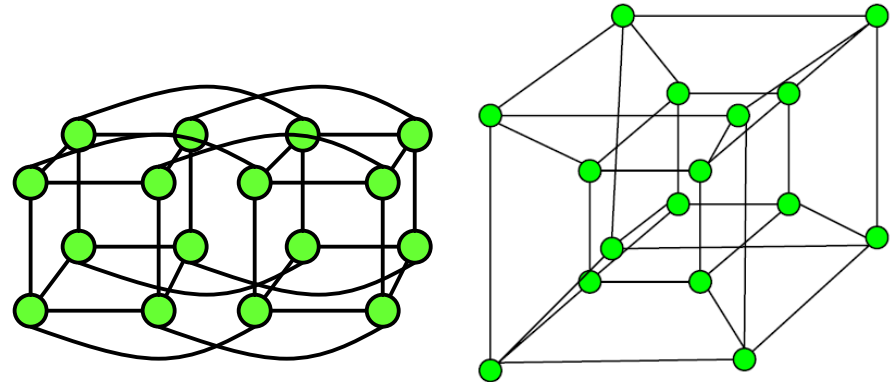
- Topologii ortogonale:
  - Nodurile sunt aranjate intr-un spatiu n-dimensional ortogonal
  - Fiecare legatura produce o deplasare intr-o singura dimensiune
- Topologii strict ortogonale (mai interesante):
  - Fiecare nod are cel putin o legatura in fiecare dimensiune
  - Trecerea intr-o noua dimensiune se poate face din orice nod
  - Rutarea e mai simpla si implementarea HW e mai eficienta
  - Nodurile pot fi numerotate folosind coordonatele lor in spatiul n-dimensional
  - Procesul de rutare se face pe baza diferentei intre coordonate
  - Obs: Distanța dintre noduri poate fi calculata direct folosind adresele nodurilor respective



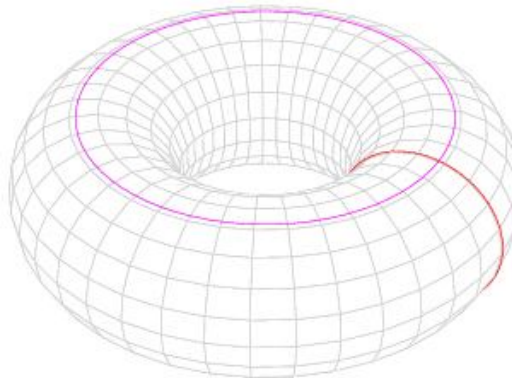
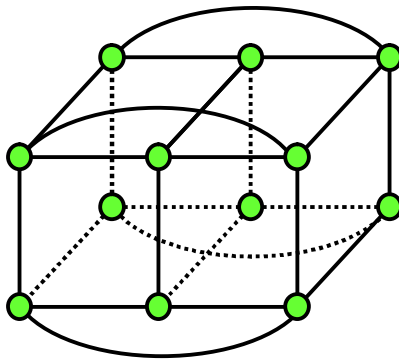
# Retele Bazate pe Rutare – Topologii

13

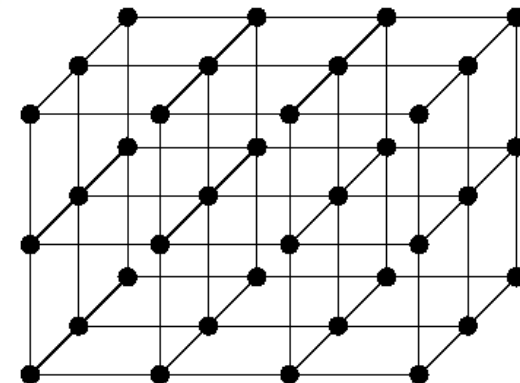
- Hiper cub:
  - 2 cuburi unul intr-altul
  - Din orice nod se poate ajunge pe orice nivel
  - Se poate optimiza drumul



- Tor:



- Plasa n-dimensională:

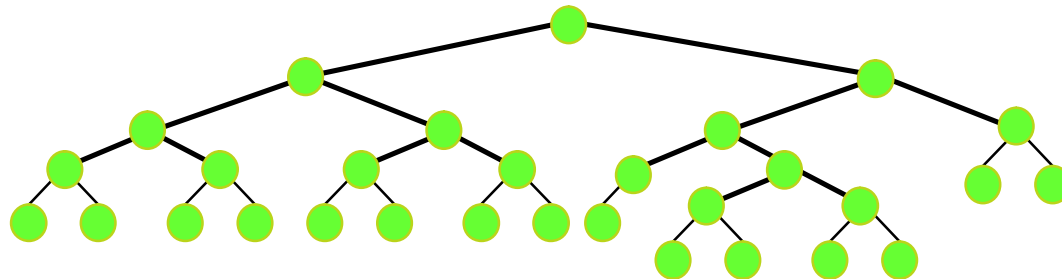




# Retele de Tip Arbore

14

- Sunt un alt tip de rețele de comutare



- Nu e neaparat necesar sa fie arbori binari!
- Fiecare nod al arborelui poate avea la randul sau un subarbore – subarbori multiplii
- Similar – la structurile cu hipercub, fiecare nod poate fi inlocuit cu un nou cub!
- Exista mai multe cai posibile pentru mesaje – totdeauna se cauta drumul de lungime minima



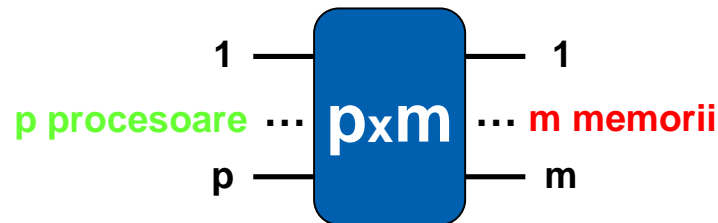
- Retele de Comutare Ierarhice
- Retele de Comutare de tip Delta
- Retele Bazate pe Rutare
- Performantele Retelelor de Comutare
  - Analiza Retelei de Tip CrossBar
  - Analiza Retelei de Tip Delta



# Performantele Retelelor de Comutare

16

- Analizăm performantele rețelelor ce asigură comutarea între  $p$  procesoare și  $m$  memorii



- Analiza va fi realizată pentru rețele de tip CB și Delta
- Rata de servire** – media cererilor de acces la memorii acceptate într-un ciclu (toate memoriile)
- Ciclu de servire** – timpul necesar pentru ca o cerere de acces la memorie să se propage prin logica de comutare + timpul efectiv de acces la memorie (propagarea inversă)





# Presupuneri/Restrictii

17

1. Nu se fac deosebiri intre ciclurile de citire si de scriere
2. Fiecare procesor va genera cereri independente pentru acces la un modul de memorie
3. Cererile sunt uniform distribuite in cadrul tuturor modulelor de memorie – fiecare procesor cu un modul de memorie
4. La inceputul fiecarui ciclu
  - Fiecare procesor genereaza o noua cerere de acces cu probabilitatea  $r$  (media cererilor generate de fiecare procesor intr-un ciclu)
  - Nu exista coada de asteptare → dialog direct
5. Cererile neacceptate intr-un ciclu sunt ignorate
  - Cererile din ciclul urmator sunt independente de cele din ciclul anterior
  - Cererile neacceptate trebuiesc relansate intr-un ciclu urmator



# Analiza Retelei de tip CrossBar

18

- Consideram un CB (p x m) si:
  - r = probabilitatea ca un procesor sa genereze o cerere intr-un ciclu
  - q(i) = probabilitatea ca "i" cereri sa soseasca in timpul unui ciclu:
$$q(i) = C_p^i r^i (1-r)^{p-i}$$
  - E(i) = numarul de cereri acceptate de retea p x m intr-un anumit ciclu
    - i cereri pot fi deservite in m<sup>i</sup> posibilitati
    - Daca un modul nu ar fi adresat atunci ar ramane (m-1)<sup>i</sup> posibilitati
    - Numarul de posibilitati in care un modul e mereu adresat este: m<sup>i</sup> – (m-1)<sup>i</sup>
    - Probabilitatea ca un modul oarecare sa fie cerut este: (m<sup>i</sup> – (m-1)<sup>i</sup>) / m<sup>i</sup>
    - Daca un modul de memorie e adresat de mai multe procesoare si cum RC accepta doar o singura cerere la un moment dat:

$$E(i) = \left( \frac{m^i - (m-1)^i}{m^i} \right) m = \left[ 1 - \left( \frac{m-1}{m} \right)^i \right] m$$

- B(p, m) = rata de transfer cu memoria este media cererilor acceptate:

- Simplificat:  $B(p, m) = m - m \left( 1 - \frac{r}{m} \right)^p$

$$B(p, m) = \sum_{i=0}^p E(i) q(i)$$



# Analiza Retelei de tip CrossBar (2)

19

- Consideram un CB ( $p \times m$ ) si:
  - $P_A$  = probabilitatea de acceptare a unei cereri oarecare
    - E data de rata de transfer  $B(p, m)$ , de  $r$  si de  $p$ :

$$P_A = \frac{B(p, m)}{r \cdot p} = \frac{m}{r \cdot p} - m \frac{\left(1 - \frac{r}{m}\right)^p}{r \cdot p}$$

- Obs: se considera ca cererile sunt independente; in realitate cererile rejectate se transfera ciclului urmator → marind rata de transfer

- Daca numarul de resurse interconectate creste:  $\lim_{m \rightarrow \infty} \left(1 - \frac{r}{m}\right)^p = e^{-r \frac{p}{m}}$

- Daca  $p$  si  $m$  sunt foarte mari atunci :

$$B(p, m) \cong m \left(1 - e^{-r \frac{p}{m}}\right) \quad P_A \cong \frac{m}{r \cdot p} \left(1 - e^{-r \frac{p}{m}}\right)$$

- Formele simplificate sunt acceptabile in urmatoarele limite:
  - Daca  $p$  si  $m \geq 30$  atunci ele sunt in limita unei erori de 1%
  - Daca  $p$  si  $m \geq 8$  atunci ele sunt in limita unei erori de 5%



## Comportarea unui Procesor intr-un CB

20

- Se va utiliza un graf Markov
- Un procesor are 2 stari posibile:
  - A = starea activa – cererea este acceptata
  - W = starea de asteptare – cererea este respinsa
- $q_A$  si  $q_W$  sunt probabilitatile ca procesorul sa se afle in starile A si W respectiv
- $P_A$  = probabilitatea ca cererea sa fie acceptata – adica de a trece din starea W in starea A
- Astfel, putem defini:

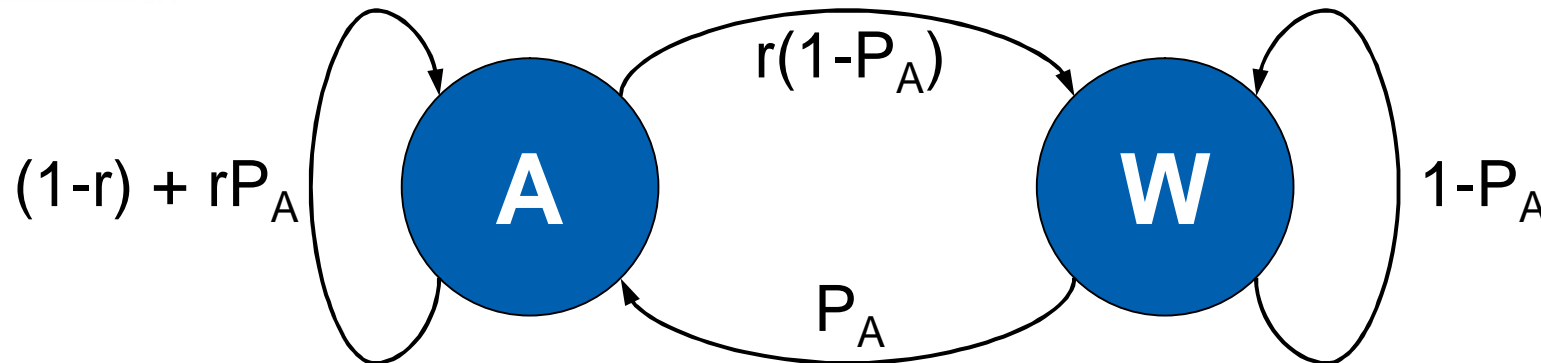
$$q_A = \frac{P_A}{P_A + r(1 - P_A)}$$

$$q_W = 1 - q_A$$



# Graful Markov al Procesorului

21



- Se considera rata de cereri  $r$  ca fiind statica
  - In realitate  $r$  e diferita luandu-se in considerare conflictele de acces
- Modulele de memorie primesc  $r' > r$  cereri de acces:

$$r' = rq_A + q_W = \frac{r}{r + (1-r)P_A} \quad P_A = \frac{m}{r'p} \left( 1 - \left( 1 - \frac{r'}{m} \right)^p \right)$$



## Analiza Retelei de tip CrossBar (3)

22

- Relatiile dinainte, definesc  $r'$  ca o functie de  $P_A$  si  $P_A$  ca o functie de  $r' \rightarrow$  definesc un proces iterativ:
  - Initiem rata de transfer  $r' = r$
  - $P_A$  poate fi considerat ca o masura a cererilor de acces nerezolvate
  - Daca  $P_A$  este mare atunci sunt putine stari de asteptare
- Fie  $\bar{w}$  media numarului de cicli de asteptare pentru o cerere
  - O cerere rejectata de  $i$  ori va astepta  $i$  cicluri
$$\bar{w} = \sum_{i=1}^p i(1 - P_A)^i = \frac{1 - P_A}{P_A}$$
- In practica, cererile ce produc conflicte de acces la memorii, nu sunt rejectate, ci introduse intr-o coada de asteptare si servite FIFO



# Analiza Retelei de tip Delta

23

- Consideram o retea Delta ( $a^n \times b^n$ ) formata din CB-uri elementare ( $a \times b$ ) –  $a^n$  procesoare si  $b^n$  memorii
- Fiecare nivel al retelei Delta e controlat de unul din bitii de reprezentare a adresei destinatie, in baza  $b$ !
- Orice modul CB  $a \times b$  e independent de celelalte module  $\rightarrow$  putem considera:
  - Cererile independente si uniform distribuite
- Daca  $r$  este rata de cereri la intrarea in  $a \times b$  atunci:

$$B(a, b) = b - b \left( 1 - \frac{r}{b} \right)^a$$

- Impartim la  $b \rightarrow$  rata de cereri la fiecare iesire din CB-ul curent:

$$\frac{B(a, b)}{b} = 1 - \left( 1 - \frac{r}{b} \right)^a$$



## Analiza Retelei de tip Delta (2)

24

- Pentru fiecare nivel intermediar al rețelei Delta:
  - $r_{out}$  (rata de cereri pentru fiecare iesire) este o funcție de  $r_{in}$  (rata de cereri de la intrare):

$$r_{out} = 1 - \left(1 - \frac{r_{in}}{b}\right)^a$$

- Rata de cereri de la iesirea unui nivel  $r_n$  este rata de cereri de intrare pentru nivelul adiacent
  - Rata de cereri de la ultimul nivel determina rata de transfer a rețelei de tip Delta

- Fie  $r_i$  rata de cereri de la iesirea unui nivel in rețeaua Delta → rata de cereri a unei RC Delta, generata de fiecare procesor este:

$$B(a^n, b^n) = b^n \cdot r_n \text{ unde } r_i = 1 - \left(1 - \frac{r_{i-1}}{b}\right)^a \text{ si } r_0 = r$$





## Analiza Retelei de tip Delta (3)

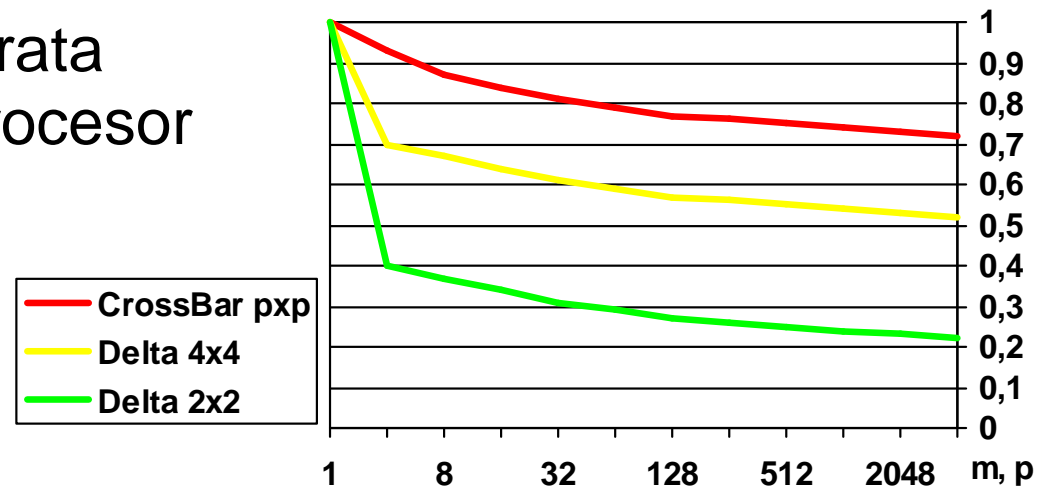
25

- Probabilitatea ca o cerere sa fie acceptata este:

$$P_A = \frac{b^n \cdot r_n}{a^n \cdot r}$$

- Ce influenta au m si p-ul unei retele?

- CrossBar pxp cu r=1 rata de cereri a fiecarui procesor
- Delta  $2^n \times 2^n$  cu  $2 \times 2$
- Delta  $4^n \times 4^n$  cu  $4 \times 4$



- Pentru m mare,  $P_A$ -ul scade puternic → multe rejectii si conflicte → limita la  $16^n \times 16^n$  este 0.6



# What Next?

26

- Q & A?
- Next time:
  - Benchmarking – nevoia de a compara sisteme de calcul
  - Benchmark-uri pentru masini seriale
  - HPCC: HPC Challenge Benchmark
    - Motivatie
    - Prezentarea componentelor software
    - Grafice & Date
    - Hands-on! Da, puteti chiar sa vedeti practic despre ce e vorba...