



# 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)



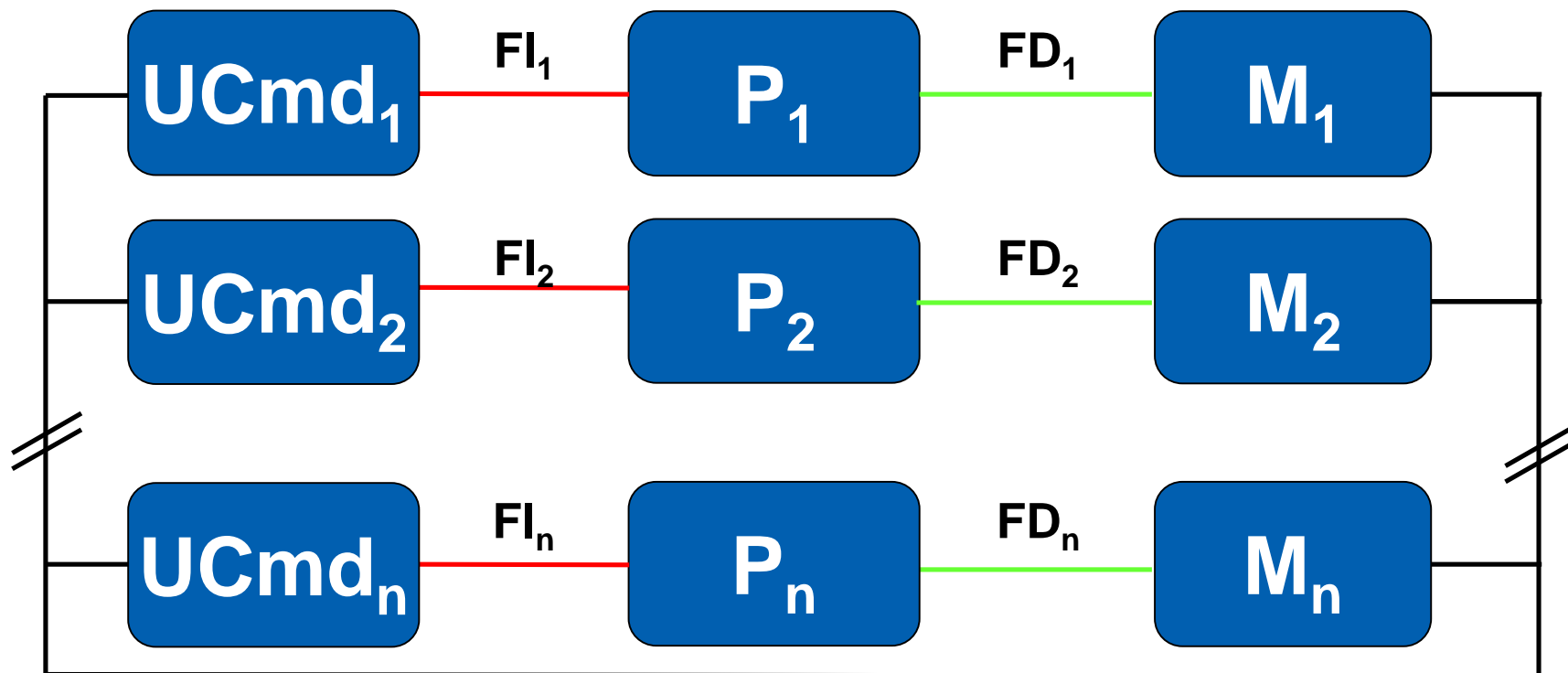
- Arhitecturi MIMD – Caracteristici
- Sisteme multiprocesor/multicalculator
- Sisteme cu memorie partajata:
  - UMA
  - NUMA/CC-NUMA
  - COMA
- Sisteme cu memorie distribuita
- Sisteme strans/slab cuplate



# MIMD

3

- MIMD pot comunica: (P-P sau P-M)
- Toate procesoarele participa la acelasi program
- Mult mai des folosite ca SIMD      **MIMD = n FI & n FD**





# Arhitecturi MIMD – Caracteristici

4

- Structura & arhitectura procesoarelor e eterogena
- Procesoarele pot functiona autonom ca un sistem universal:
  - Pot prelua task-urile altor procesoare din sistem
  - Astfel se asigura degradarea performantei sistemului
- Organizarea memoriei
  - Fiecare procesor trebuie sa aiba acces la toata memoria sistemului
  - Capacitatea de adresare a fiecarui procesor e mai mica ca dimensiunea memoriei → mecanisme speciale
  - Asigura protectia memoriei pentru asigurarea integritatii datelor din sistem



# Arhitecturi MIMD – Caracteristici

5

- Modularitatea presupune:
  - Expandabilitate – numărul de procesoare poate fi modificat fără a fi necesară reconfigurarea sistemului
  - Reconfigurabilitate
    - Structurală: procesul de refacere a stării sistemului la apariția unor defecte → se realizează prin distribuirea structurală a funcțiilor; permite preluarea sarcinilor unui proc de către alte procesoare
    - Arhitecturală: schimbarea caracteristicilor de bază
      - Dimensiunea registrilor de memorie
      - Lungimea cuvântului
      - Structura UAL



# Arhitecturi MIMD – Probleme

6

- Decompozitia algoritmilor
  - Manual – la conceptia algoritmilor
  - Automat – prin limbaje de programare concurenta
- Planificarea pentru executie a task-urilor si maparea pe structura de procesoare utilizata
- Sincronizarea proceselor/task-urilor astfel incat sa nu se ajunga la blocarea sistemului
- Siguranta in functionare e asigurata de
  - Disponibilitate: capacitatea sistemului de a raspunde cererilor utilizatorului
  - Integritate: capacitatea sistemului de a mentine consistenta starii in conditiile aparitiilor unor defecte



- Arhitecturi MIMD – Caracteristici
- **Sisteme multiprocesor/multicalculator**
- Sisteme cu memorie partajata:
  - UMA
  - NUMA/CC-NUMA
  - COMA
- Sisteme cu memorie distribuita
- Sisteme strans/slab cuplate



# Clasificarea Structurilor MIMD

8

- Din punct de vedere arhitectural, structurile MIMD se impart in doua mari categorii:
  1. Sisteme multiprocesor:
    - sistem de calcul unitar unde o colectie de procesoare autonome functioneaza concurent si coopereaza la realizarea unei functii logice si de calcul de sub controlul unui “executiv” unic
  2. Sisteme multicalculator:
    - mai multe calculatoare autonome, interconectate, ce colaboreaza la realizarea unor task-uri din cadrul unei lucrari complexe
- Din punct de vedere al accesului la memorie, structurile MIMD sunt:
  1. Sisteme cu memorie partajata (Shared Memory)
  2. Sisteme cu memorie distribuita (Distributed Memory)





# Sisteme Multiprocesor

9

- Au un singur spatiu de adresare (comunicare prin shared memory)
- Se impart in sisteme
  1. Cu memorie distribuita (scalabilitate):
    - Gestiune dinamica a accesului la memoria distribuita
    - Gestiune statica a accesului la memoria distribuita
  2. Cu memorie centrala (nu exista scalabilitate):
    - Comunicarea se face prin retele de comutatie (e.g. crossbar switch)
    - Conexiuni de tip magistrala (bus) multiplu



# Sisteme Multicalculator

10

- Au spatii multiple de adresare (comunicare prin transfer de mesaje)
- Se impart in sisteme

## 1. Multicalculator distribuite:

- Mesh – plasa de interconectare P-M
- Conexiuni Butterfly (Perfect Shuffle) – CM5
- Hypercube – la sistemele NCUBE
- Retele LAN rapide
  - Myrinet, ATM, GigaEthernet, MPLS ...

## 2. Multicalculator centralizate



- Arhitecturi MIMD – Caracteristici
- Sisteme multiprocesor/multicalculator
- Sisteme cu memorie partajata:
  - UMA
  - NUMA/CC-NUMA
  - COMA
- Sisteme cu memorie distribuita
- Sisteme strans/slab cuplate



# Sisteme cu Memorie Partajata

12

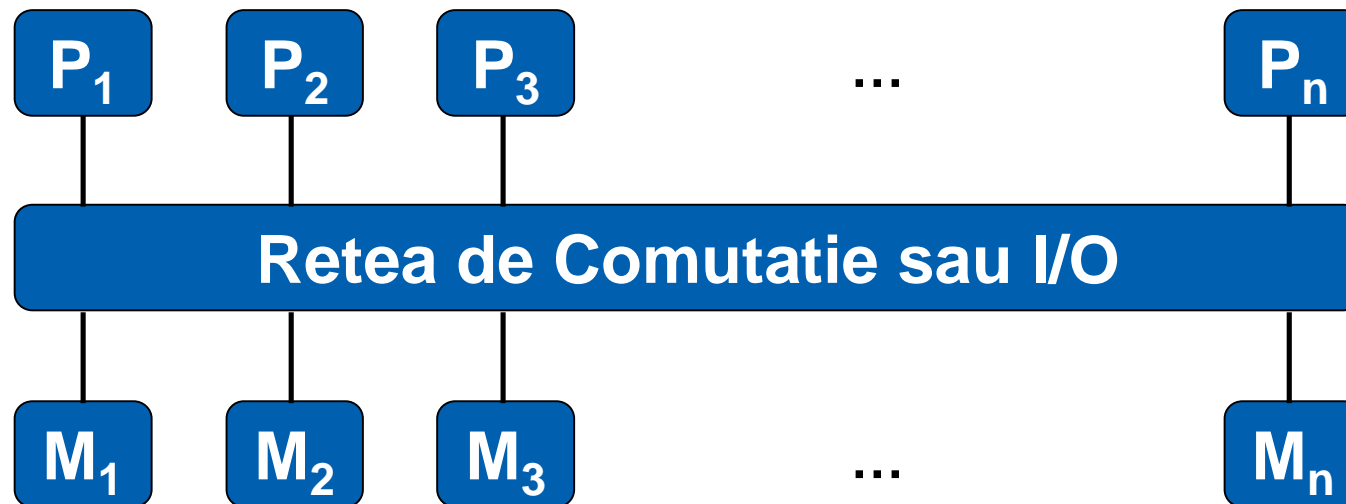
- Sistemele cu memorie partajata (Shared Memory) pot fi implementate in una din urmatoarele configuratii:
  1. UMA = Uniform Memory Access
  2. NUMA = Non-Uniform Memory Access
    - CC-NUMA
  3. COMA = Cache-Only Memory Architecture



# 1. Structura UMA

13

- UMA = Uniform Memory Access
- Accesul este unic: orice procesor se poate conecta la un modul de memorie in mod uniform
- Reteaua de comunicatie (I/O) poate fi de orice tip

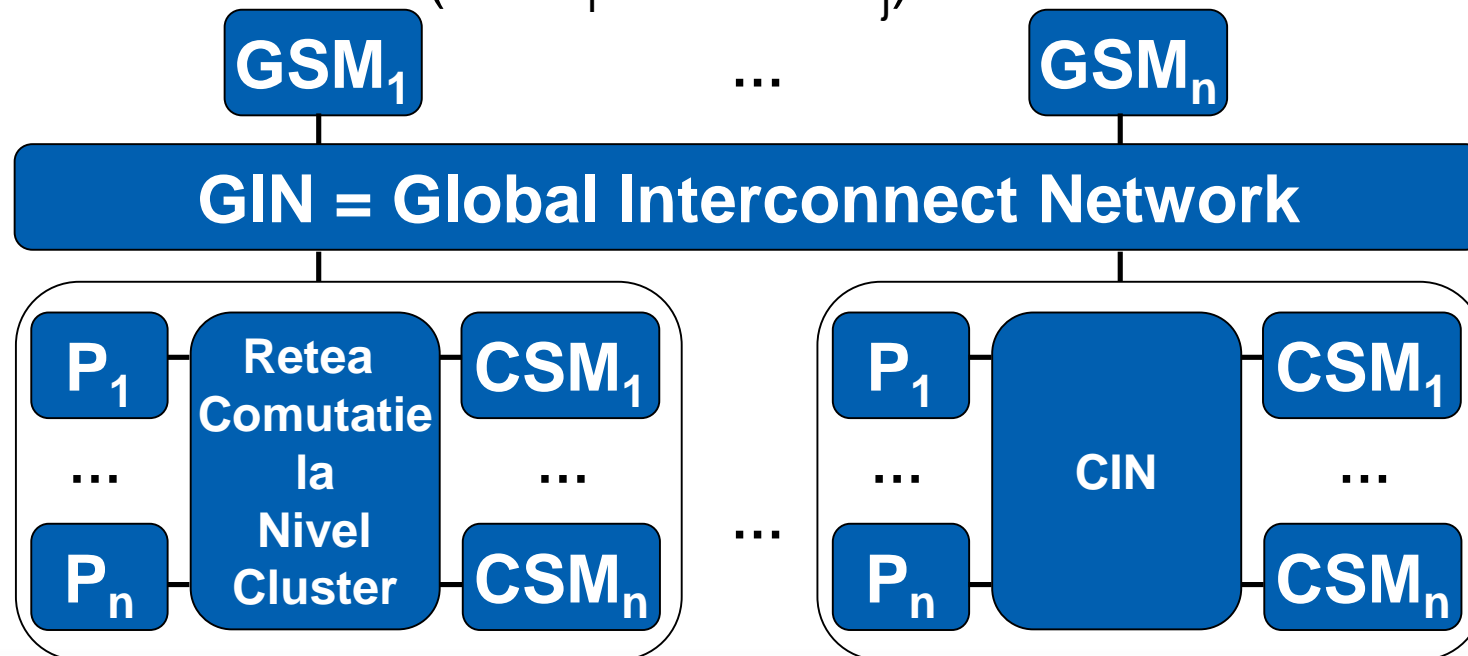




## 2. Structura NUMA

14

- CSM = Cluster Shared Memory
- CIN = Cluster Interconnect Network
- GSM = Global Shared Memory
- GIN = retea de comutare globala la care se conecteaza clustere formate din procesoare
- Accesul la memorii depinde de modulul unde trebuie efectuat accesul ( $CSM_i$  sau  $GSM_i$ )





## 2. Structura CC-NUMA

15

- CC-NUMA = Cache Coherent NUMA
  - Coerenta accesului la memorie e mai bine stabilita ca la NUMA
  - Ex: Silicon Graphics (Origin 2000):
    - sunt 2 procesoare pe nod
    - accesul e divizat la componentele nodului printr-un Bus intern
    - un Hub de conectare asigura accesul partajat la Cache-ul nodului → asigurand coerenta cache-ului
  - Ex: Sequent NUMA-Q = implementat pe structuri de noduri interconectate cu o structura rapida inelara (Quad)
    - Standardul IEEE – SCI = Scalable Coherent Interface – structura de comunicare (1GB/s)
  - Ex: Stanford DASH (Directory Architecture for SHared memory) – 4 MIPS R3000 Procs

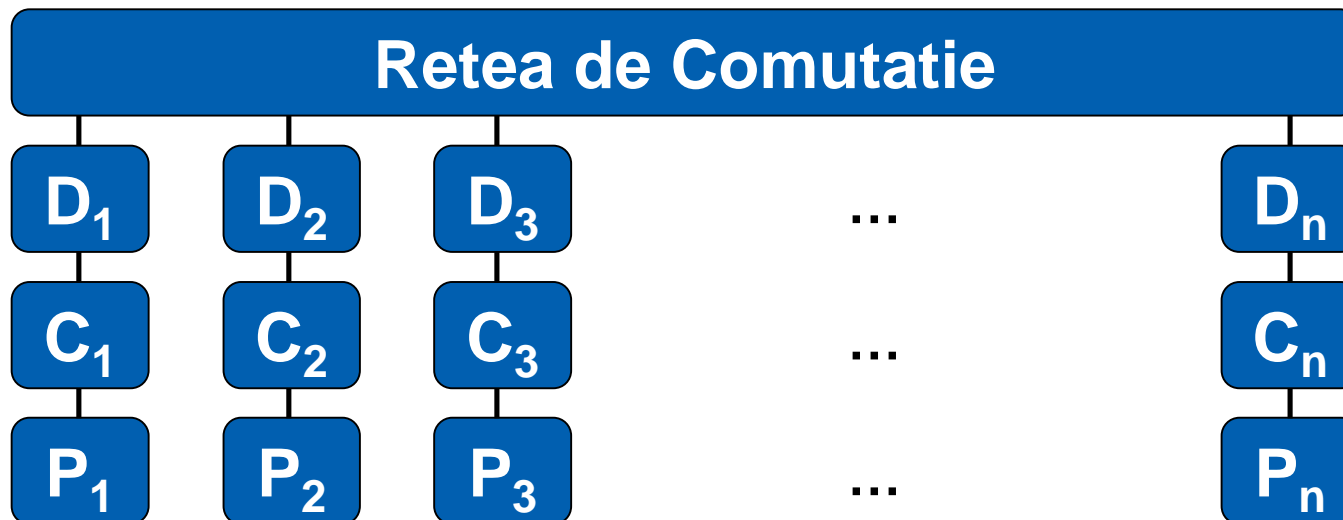




### 3. Structura COMA

16

- Procesoarele nu au acces direct
  - la propria zona de date
  - la cea a celorlalte procesoare
- Procesoarele au acces direct doar la Cache! → Cache Only
- $D_i$  = Memoria de Date
- $C_i$  = Memoria Cache
- $P_i$  = Procesoare







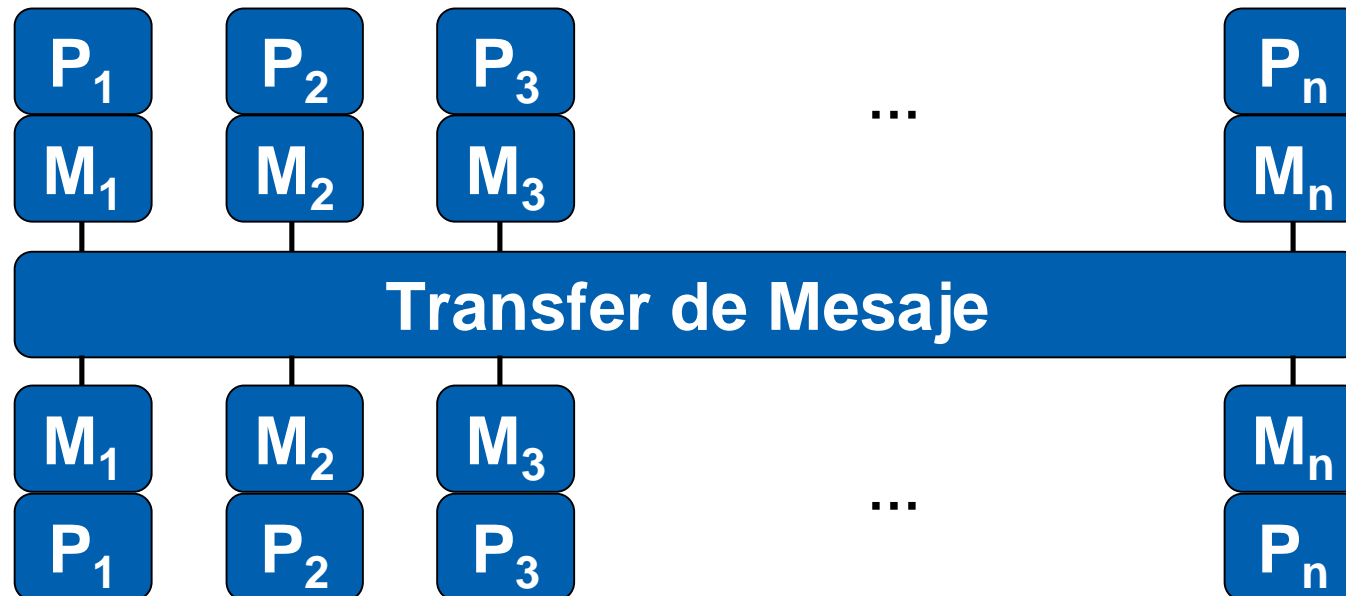
- Arhitecturi MIMD – Caracteristici
- Sisteme multiprocesor/multicalculator
- Sisteme cu memorie partajata:
  - UMA
  - NUMA/CC-NUMA
  - COMA
- Sisteme cu memorie distribuita
- Sisteme strans/slab cuplate



# Sisteme cu Memorie Distribuita

18

- Accesul se face prin transfer de mesaje
- Sistemele au la baza mai multe noduri legate printr-o retea de comutatie de mesaje (nu neaparat fizica)
- Conexiunea este statica, punct la punct intre noduri
- Memoria locala este adresabila direct doar de catre procesorul local





- SUN Enterprise 10000 - Starfire
  - Reteaua de comutatie este de tip Gigaplane XB (102.4GB/s; p2p: 1.6GB/s)
  - Schema de comutare de pachete este cu cai separate de adrese si date (Crossbar)
  - Adresele sunt distribuite printr-un router ce are functia de broadcast
  - Acest fapt permite
    - Partitionarea dinamica a setului de procesoare si crearea unor partitii
    - Aceste partitii sunt complet izolate intre ele si pot fi utilizate ca atare pentru sarcini distincte
  - Starfire suporta paralelism de tip shared-memory in Fortran & C/C++ (OpenMP)





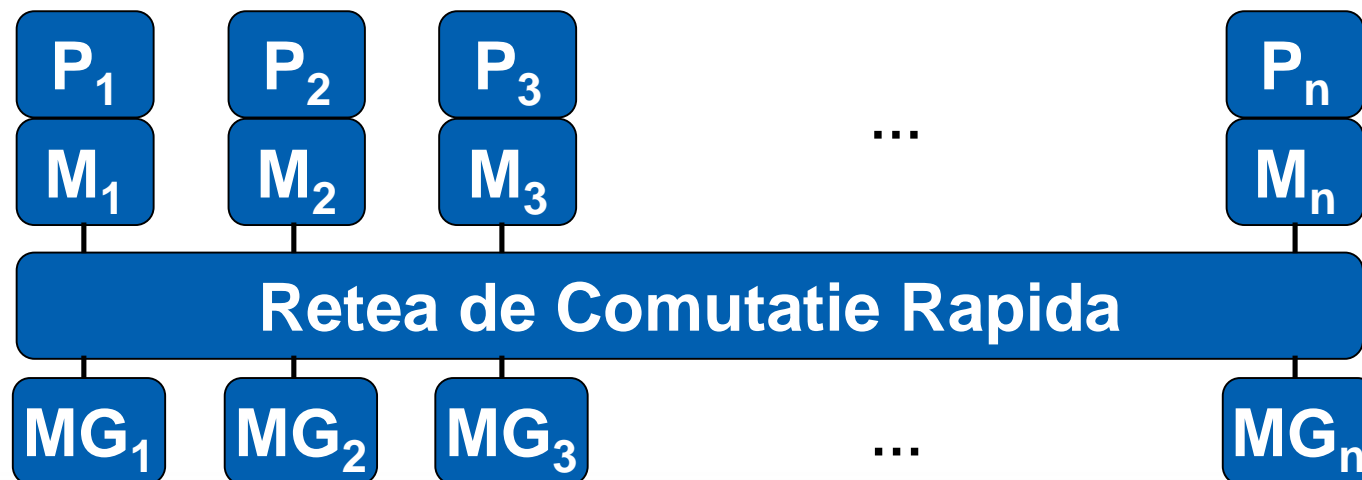
- Arhitecturi MIMD – Caracteristici
- Sisteme multiprocesor/multicalculator
- Sisteme cu memorie partajata:
  - UMA
  - NUMA/CC-NUMA
  - COMA
- Sisteme cu memorie distribuita
- Sisteme strans/slab cuplate



# Sisteme Strans (Puternic) Cuplate

21

- Dispun de procesoare  $P_i$  cu memorie locala asociata  $M_i$  de tip multiport
- O retea de comutatie rapida RC (dar fixa) ce permite accesul uniform la memoria globala partajata  $MG_i$
- Comunicarea si sincronizarea se face prin  $MG_i$  cu o rata de transfer similara cu a memoriei locale
- Asigura o scalabilitate redusa datorita faptului ca RC este fixa si de aceea au o structura dedicata
- Performantele sistemului sunt date de modul de acces la memoria partajata





# Sisteme Slab Cuplate

22

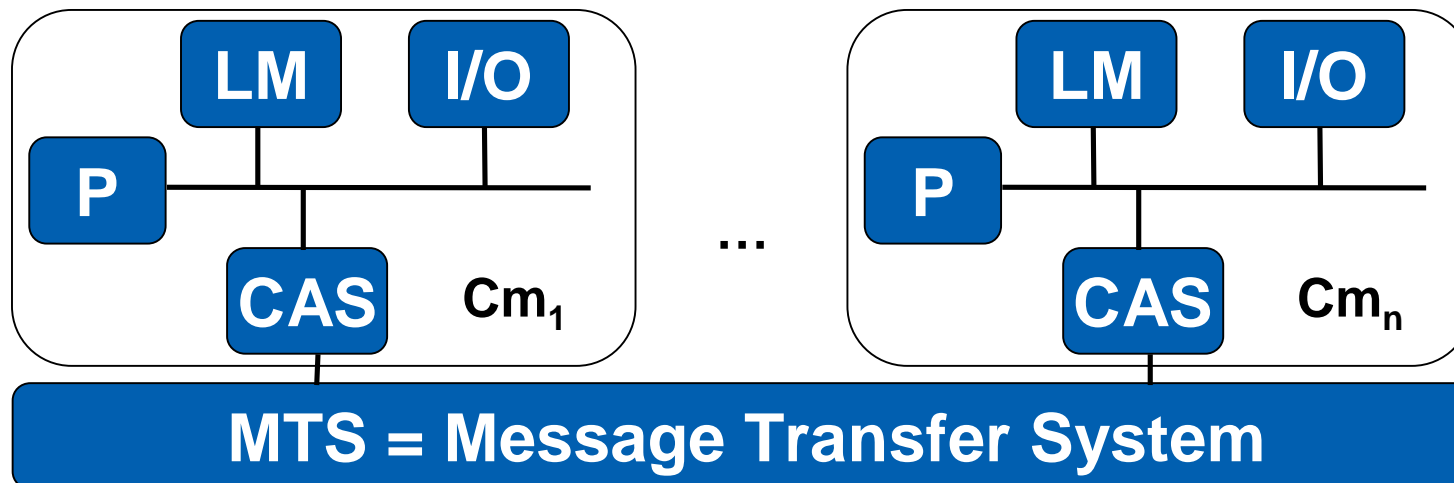
- De tip cluster → calculatoarele au resurse locale necesare functionarii independente
- Procesele se executa pe procesoare care dispun de toate resursele pentru implementarea task-ului respectiv
- Comunicarea intre procesoare se face prin
  - Transfer de mesaje
  - Comutare de circuite
- Sistemele slab cuplate (distribuite) sunt eficiente atunci cand interactiunea intre procesoare este minima (granularitatea problemei este mare)



# Sisteme Slab Cuplate

23

- Resurse locale ale  $CM_i$  = Computer Module:
  - LM = local memory (acces din interior/exterior)
  - I/O = magistrala de intrare/iesire (acces din interior/exterior)
  - MTS = Message Transfer System – poate fi implementata cu comutare de circuite sau transfer de mesaje
  - MTS dicteaza performantele prin lungimea mesajelor & viteza de lucru
  - CAS = Collision Avoidance System – legatura si accesul la MTS se face prin excludere mutuala:
    - un singur transfer odata daca MTS este magistrala
    - mai multe transferuri simultane daca MTS este de tip retea





## Exemplu – Cm<sup>\*</sup>

24

- Structura Cm<sup>\*</sup> a fost dezvoltata arhitectural si implementata la Carnegie Mellon in anii '70
- Cm<sup>\*</sup> e proiectat cu intentia de a obtine in acelasi timp:
  - Performante bune
  - Fiabilitate sporita
- In 1975 structura e compusa din 10 procesoare
- Ajunge in 1977 la 50 de procesoare
- Cm<sup>\*</sup> beneficiaza de doua sisteme de operare distincte



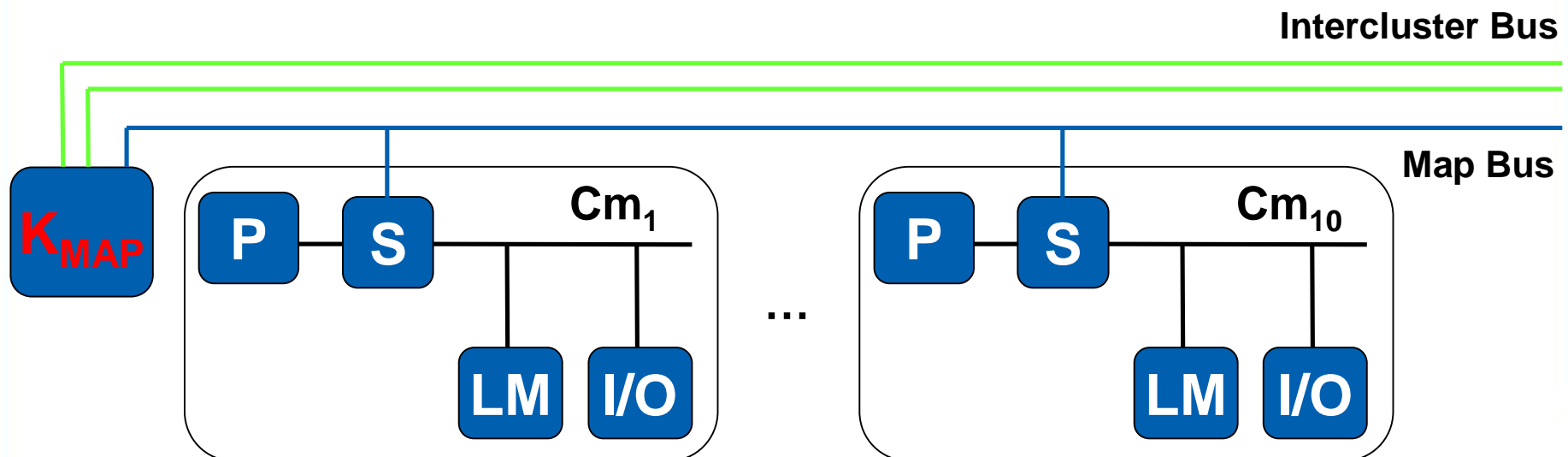




# Arhitectura Cm\*

25

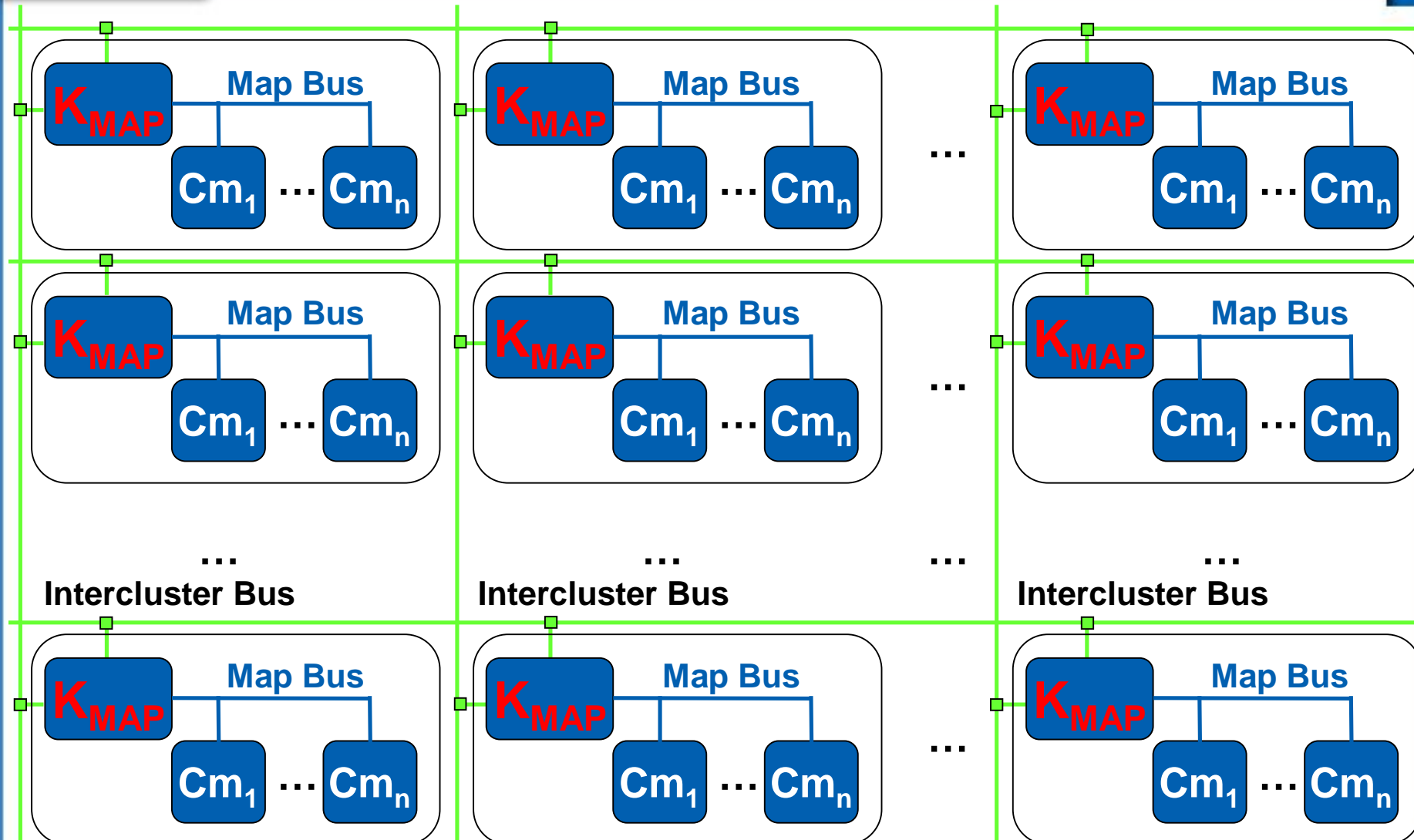
- Structura Cm\* conecteaza mai multe module individuale → structura multipla de calculatoare == cluster
- $K_{MAP}$  = gestioneaza magistrala Map Bus si cererile pentru Cm-uri
- Un cluster e format din mai multe Cm-uri,  $K_{MAP}$  & Map Bus
- Cm-urile sunt fiabile → asigura semiparalelism





# Conectare Matriceala de Clustere

26





# Caracteristicile Cluster-ului

27

- Un Cluster faciliteaza cooperarea intre resursele sistemului cu un overhead minim
- Sursa de bottleneck intr-un cluster este Map Bus  $\leftrightarrow$  posibila doar o singura tranzactie la un moment dat  $\rightarrow$  cea mai avantajoasa este comutarea de circuite
- Orice referire la memorie este tratata de  $K_{MAP} \rightarrow$  rutare
- Clusterelor comunica intre ele prin  $K_{MAP}$  conectate la Intercluster Bus
- Functii  $K_{MAP}$ :
  - Maparea adreselor
  - Gestionarea circuitelor
  - Transmiterea mesajelor pe Ethernet; TokenBus; TokenRing etc.
  - Sincronizarea



# What Next?

28

- Q & A?
- Next time:
  - Elementele Structurii Cm
  - Accesul la resursele locale – Switch-ul S
  - Structura si functiile  $K_{MAP}$ 
    - Comunicatia  $K_{BUS} - P_{MAP}$
    - Comunicatia Link –  $P_{MAP}$
    - Comunicatia  $K_{BUS} - \text{Map Bus}$
  - Arhitectura procesorului de mapare  $P_{MAP}$
  - Comunicarea in cadrul unui cluster
  - Comunicarea intercluster