5. Introducere în arhitecturi paralele

- Tipuri și nivele de paralelism
- Clasificarea arhitecturilor paralele
- Arhitecturi vectoriale
- Arhitecturi SIMD
- Arhitecturi sistolice
- Arhitecturi MIMD
- Arhitecturi specifice unor domenii
- Arhitecturi cu fire de execuție multiple

Arhitecturi MIMD

- Arhitecturi MIMD
  - Prezentare generală
  - Arhitecturi cu memorie partajată
  - Arhitecturi cu transmitere de mesaje
  - Alte arhitecturi MIMD
  - Performanța arhitecturilor MIMD

Prezentare generală (1)

- Arhitecturile MIMD sunt mai generale comparativ cu celelalte arhitecturi paralele → se pot utiliza pentru o clasă largă de aplicații
- Fiecare EP poate executa o instrucțiune diferită în orice moment
  - Fiecare EP are propria UAL și unitate de comandă → procesor independent
  - Elementele de procesare sunt interconectate → transferuri de date, sincronizare

Prezentare generală (2)

- Programarea arhitecturilor MIMD este mai dificilă
  - Algoritmul de procesare trebuie să prezinte un grad ridicat de paralelism
- Performanța teoretică maximă a unui sistem cu *n* procesoare nu poate fi atinsă
  - Comunicarea între procesoare
  - Sincronizarea activității procesoarelor
  - Planificarea alocării taskurilor la procesoare

Prezentare generală (3)

- Dificultăți de proiectare a arhitecturilor MIMD
  - Alocarea procesoarelor la procese: în mod dinamic
  - Sincronizarea procesoarelor: prevenirea modificării simultane a datelor
  - Proiectarea rețelelor de interconectare: între procesoare și memorie; între procesoare
  - Partiționarea: identificarea paralelismului în algoritmii de prelucrare

Prezentare generală (4)

- Avantajele arhitecturilor MIMD
  - Performanța ridicată: în mod ideal, toate cele *n* procesoare execută în paralel taskuri ale unei prelucrări
  - Toleranța la defecte: la defectarea unui procesor sau bloc de memorie, taskurile pot fi realocate altor resurse
  - Este posibilă reconfigurarea dinamică a resurselor

Prezentare generală (5)

- Într-o arhitectură MIMD, fiecare procesor funcționează independent
- Rezultatele taskurilor unui procesor pot fi necesare unui alt procesor
  - Memorarea lor într-o memorie partajată
  - Transmiterea directă a rezultatelor
- Tipuri principale de organizare
  - Cu memorie partajată
  - Cu transmitere de mesaje

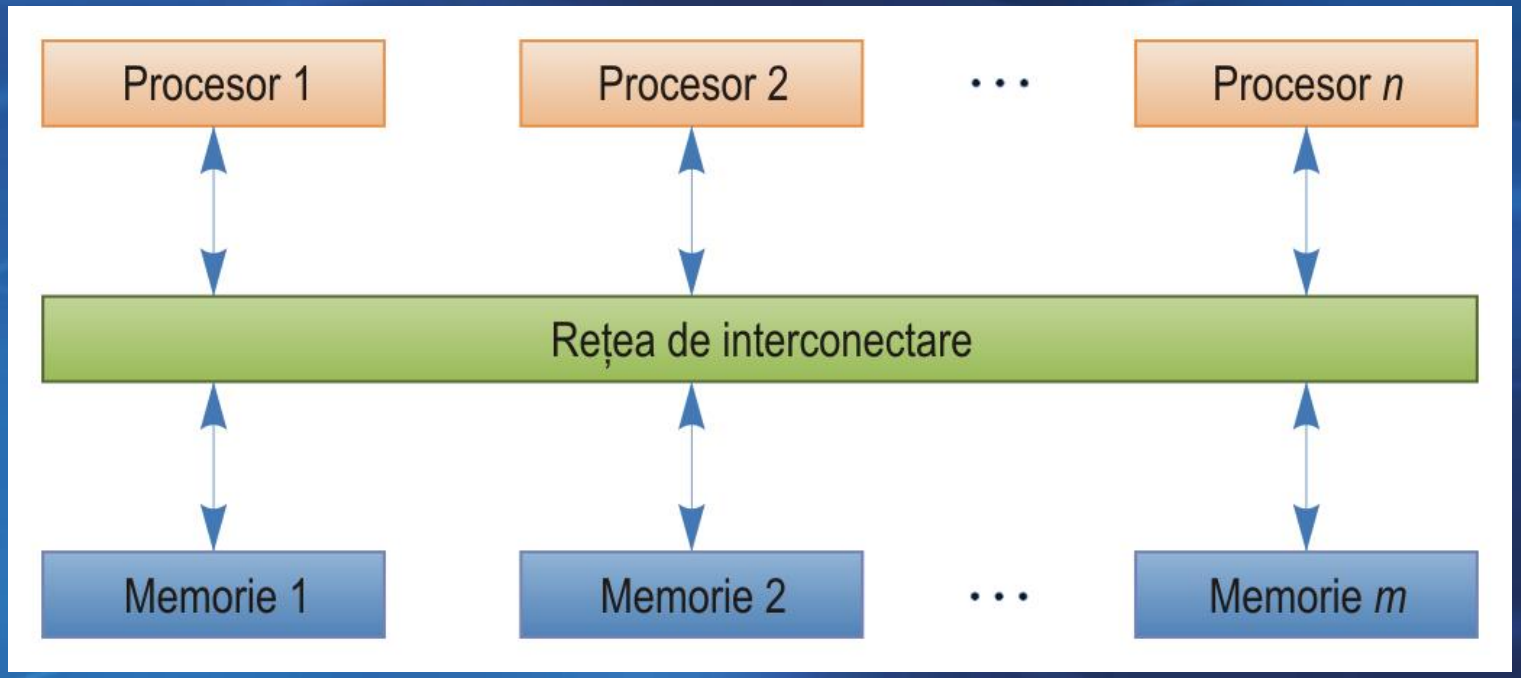
Arhitecturi MIMD

- Arhitecturi MIMD
  - Prezentare generală
  - Arhitecturi cu memorie partajată
  - Arhitecturi cu transmitere de mesaje
  - Alte arhitecturi MIMD
  - Performanța arhitecturilor MIMD

Arhitecturi cu memorie partajată (1)

- Se mai numesc multiprocesoare
- Procesoare interconectate cu module de memorie
  - Oricare procesor poate accesa oricare modul de memorie → se simplifică programarea
  - Transferul datelor între procesoare prin RI este rapid → arhitectură cu legătură strânsă
  - Timpul de acces la memorie este același → arhitectură cu memorie uniformă (UMA – *Uniform Memory Architecture*)

Arhitecturi cu memorie partajată (2)



Arhitecturi cu memorie partajată (3)

- Procesoare care nu sunt conectate la același modul de memorie: este necesară o secvență de transferuri
- Procesoare neomogene: sunt necesare transformări ale datelor
- Pot apare conflicte la accesul memoriei
  - Memorii cu unități multiple
  - Intercalarea adreselor de memorie
  - Memorii cu porturi multiple

Arhitecturi cu memorie partajată (4)

- Avantaje:
  - Tehnicile de programare utilizate pentru uni-procesoare pot fi adaptate în mod simplu pentru multiprocesoare
  - Nu este necesară mutarea fizică a datelor atunci când procesele comunică între ele → comunicația între procese este eficientă
- Dezavantaje:
  - Accesul la datele partajate necesită construcții de sincronizare → semafoare

Arhitecturi cu memorie partajată (5)

- Scalabilitatea redusă din cauza conflictelor la accesul memoriei; probabilitatea conflictelor crește cu creșterea numărului de procesoare
- Soluții pentru îmbunătățirea scalabilității
  - RI de viteză ridicată între EP și memorii
  - Utilizarea unor memorii *cache* locale → apare problema coerenței memoriilor *cache*
  - Implementarea memoriei partajate ca o colecție de memorii locale → arhitectură cu memorie partajată virtuală

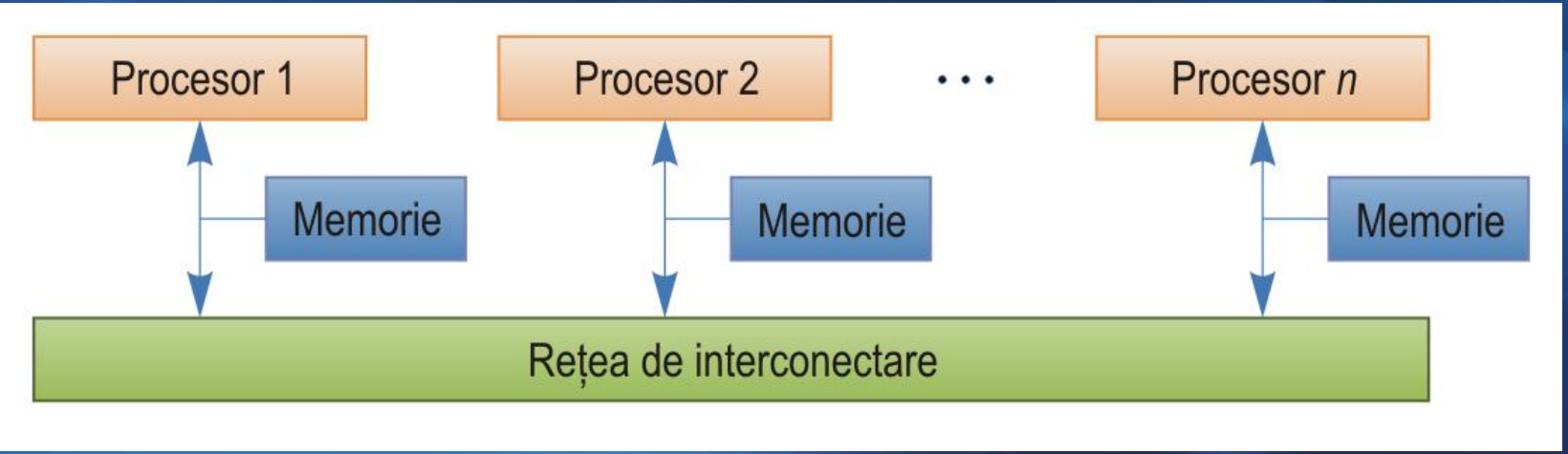
Arhitecturi MIMD

- Arhitecturi MIMD
  - Prezentare generală
  - Arhitecturi cu memorie partajată
  - Arhitecturi cu transmitere de mesaje
  - Alte arhitecturi MIMD
  - Performanța arhitecturilor MIMD

Arhitecturi cu transmitere de mesaje (1)

- Se mai numesc multicalculatoare
- Fiecare procesor are un bloc de memorie locală
  - Arhitectură cu legătură slabă sau cu memorie distribuită
  - Timpul de acces la memorie este diferit → arhitectură cu memorie neuniformă (NUMA – *Nonuniform Memory Architecture*)
  - Niciun procesor nu poate accesa direct blocul de memorie al altui procesor

Arhitecturi cu transmitere de mesaje (2)



Arhitecturi cu transmitere de mesaje (3)

- Interacțiunea între diferite procesoare se realizează prin transmiterea unor mesaje între procesoare
- Transferul datelor între două procesoare
  - Procesorul solicitant transmite un mesaj procesorului în memoria căruia se află datele
  - Procesorul solicitat citește datele din memoria locală și le transmite prin RI
  - RI rutează datele către procesorul solicitant

Arhitecturi cu transmitere de mesaje (4)

- Avantaje:
  - Problema conflictului la memorie este mai puțin severă → scalabilitate
  - Nu sunt necesare tehnici de sincronizare complexe
- Dezavantaje:
  - Timpul de așteptare pentru datele solicitate poate fi semnificativ → procesorul solicitant este inactiv

Arhitecturi cu transmitere de mesaje (5)

- Comunicația și sincronizarea bazată pe transmitere de mesaje poate conduce la blocaje
- Transmiterea de mesaje necesită copierea unor structuri de date între procese → poate reduce semnificativ performanțele
- Pentru performanțe ridicate, este necesară încărcarea echilibrată a procesoarelor → partiționarea codului și a datelor între procesoare

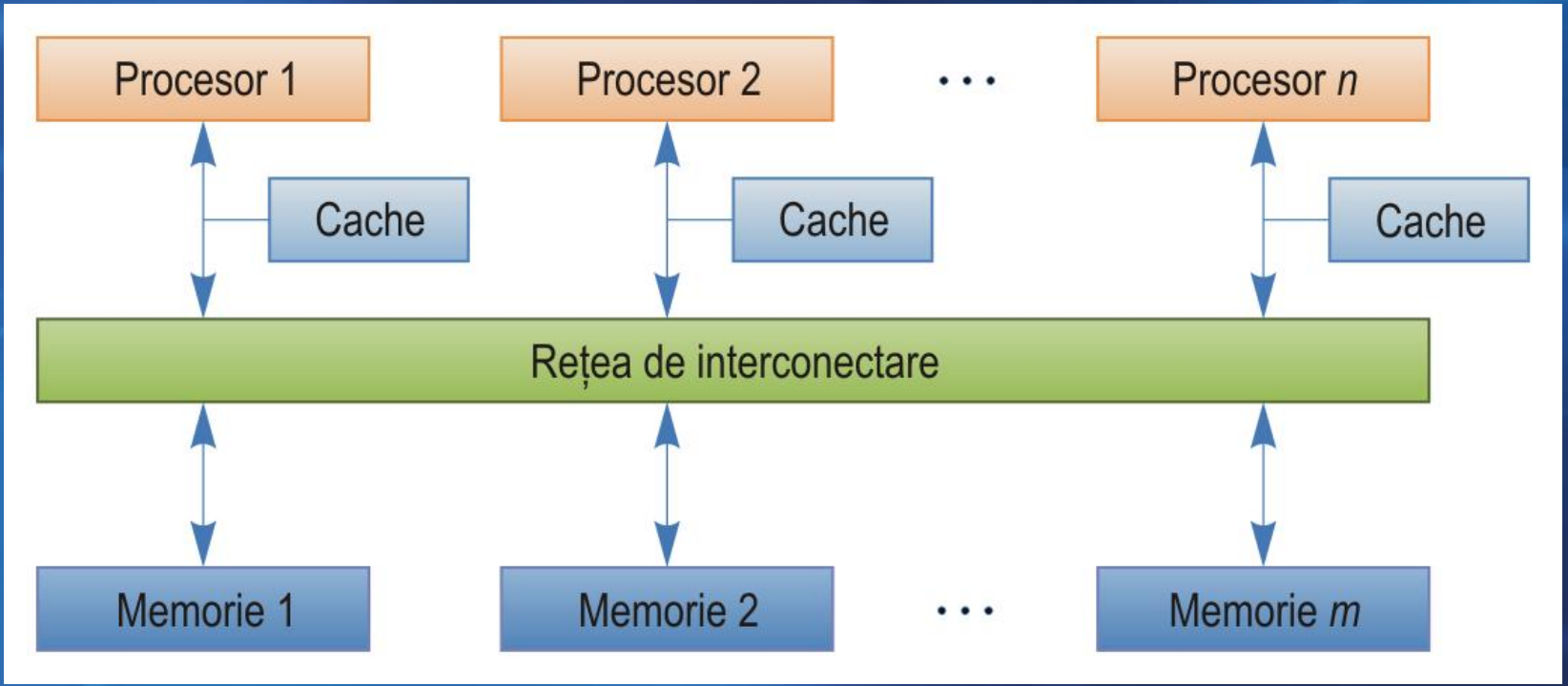
Arhitecturi MIMD

- Arhitecturi MIMD
  - Prezentare generală
  - Arhitecturi cu memorie partajată
  - Arhitecturi cu transmitere de mesaje
  - Alte arhitecturi MIMD
  - Performanța arhitecturilor MIMD

Alte arhitecturi MIMD (1)

- Multiprocesoarele și multicalculatoarele reprezintă două extreme
  - Sistemele MIMD pot utiliza o combinație a celor două arhitecturi
- Arhitectură care utilizează numai memorii *cache* (COMA – *Cache Only Memory Architecture*)
  - Calculatoarele DDM (*Data Diffusion Machine*)
- Arhitectură cu memorie distribuită și memorii *cache* →

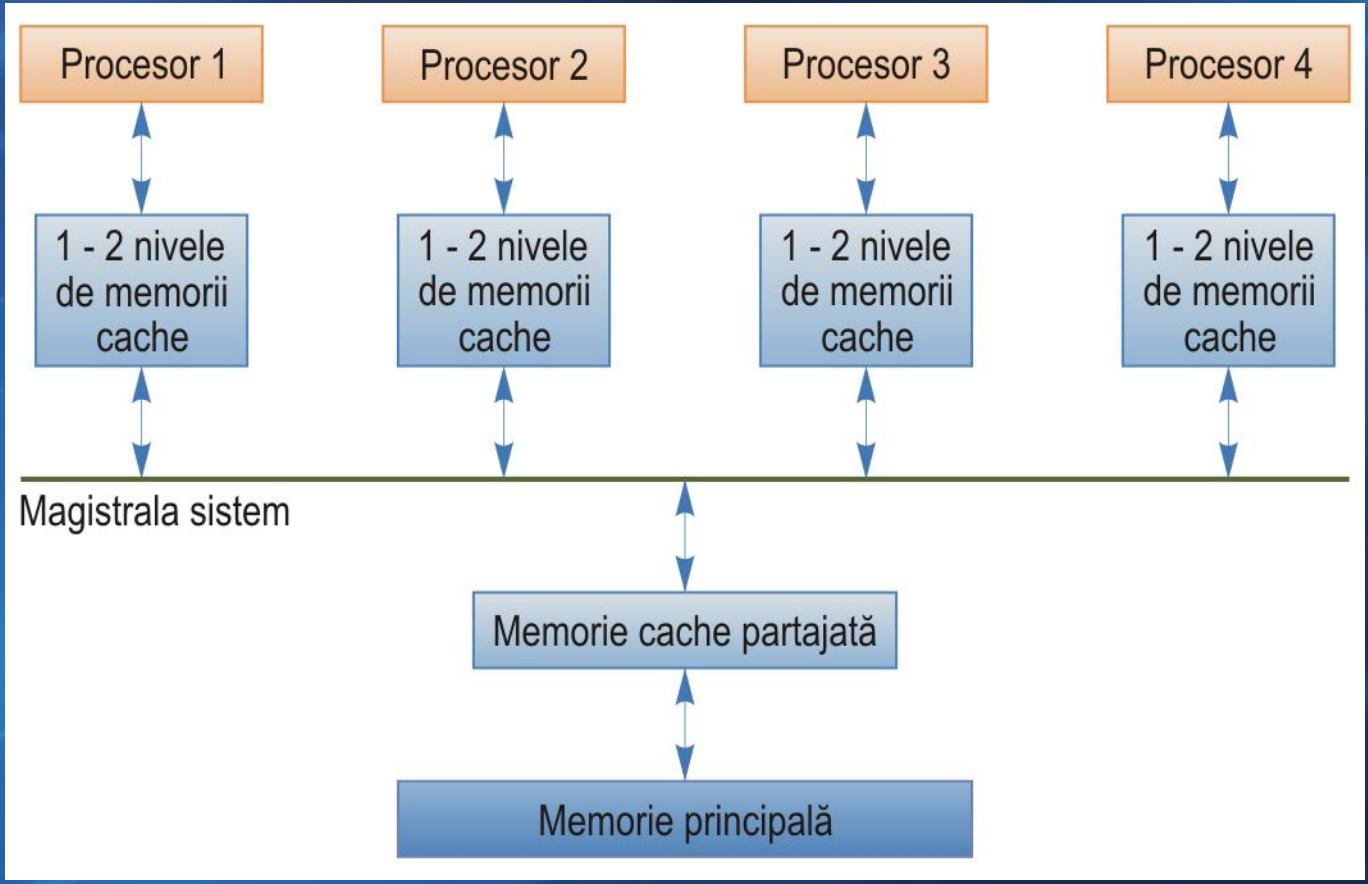
Alte arhitecturi MIMD (2)



Alte arhitecturi MIMD (3)

- Multiprocesor format dintr-un singur circuit cu nuclee multiple de procesare
  - Conține 1-2 nivele de memorii *cache* private pentru fiecare nucleu de procesare
  - Nucleele de procesare partajează un nivel de memorie *cache*
  - Memoria principală este partajată
  - Rețeaua de interconectare este reprezentată de magistrala sistem

Alte arhitecturi MIMD (4)



Alte arhitecturi MIMD (5)

- Spațiul de adrese poate fi privat sau partajat
- Arhitecturi cu memorie locală și spațiu de adrese partajat
  - Arhitecturi NUMA
  - Avantaje: scalabilitate, programare simplă
  - Exemplu: J-Machine (MIT)
- Configurația unui sistem MIMD depinde de caracteristicile aplicațiilor

Arhitecturi MIMD

- Arhitecturi MIMD
  - Prezentare generală
  - Arhitecturi cu memorie partajată
  - Arhitecturi cu transmitere de mesaje
  - Alte arhitecturi MIMD
  - Performanța arhitecturilor MIMD

Performanța arhitecturilor MIMD (1)

- Performanța este afectată de doi factori:
  - Gradul de paralelism disponibil în aplicațiile rulate
  - Comunicația necesară accesului la datele aflate la distanță
- Gradul de paralelism disponibil
  - O parte din codul aplicațiilor se va executa în mod secvențial
  - Creșterea vitezei la execuția în mod paralel se poate exprima utilizând legea lui Amdahl

Performanța arhitecturilor MIMD (2)

- $\Delta v_{tot} = \dfrac{1}{1 - F_{paralel} + \dfrac{F_{paralel}}{\Delta v_{paralel}}}$
  - $F_{paralel}$: Fracțiunea timpului de execuție în care se utilizează modul paralel
  - $\Delta v_{paralel}$: Creșterea vitezei dacă s-ar utiliza numai modul paralel
- Aplicațiile pot utiliza un număr variabil de procesoare la rularea în mod paralel
- Se poate utiliza o formă extinsă a legii lui Amdahl pentru aplicațiile care pot utiliza un număr diferit de procesoare

Performanța arhitecturilor MIMD (3)

- Exemplul 5.1
  - Considerăm o aplicație care rulează pe un multiprocesor cu 100 procesoare
  - Aplicația poate utiliza 1, 50 sau 100 de procesoare
  - Presupunem că 95% din timp se pot utiliza 100 procesoare
  - Ce procent din timp trebuie să se utilizeze 50 de procesoare pentru a se obține o creștere totală a vitezei de 80?

Performanța arhitecturilor MIMD (4)

- Forma extinsă a legii lui Amdahl:

$$\Delta v_{tot} = \frac{1}{1 - F_{100} - F_{50} + \frac{F_{100}}{\Delta v_{100}} + \frac{F_{50}}{\Delta v_{50}}}$$

$$\Delta v_{tot} = 80$$

$$F_{100} = 0{,}95;\ \Delta v_{100} = 100;\ \Delta v_{50} = 50$$

$$80 = \frac{1}{1 - 0{,}95 - F_{50} + \frac{0{,}95}{100} + \frac{F_{50}}{50}}$$

$$80 = \frac{1}{0{,}05 - F_{50} + 0{,}0095 + \frac{F_{50}}{50}}$$

$$80 = \frac{50}{50 \times 0{,}0595 - 50 \times F_{50} + F_{50}}$$

Performanța arhitecturilor MIMD (5)

$$80 \times 2{,}975 - 80 \times 49 \times F_{50} = 50$$

$$238 - 50 = 3920 \times F_{50}$$

$$F_{50} = \frac{188}{3920} \cong 0{,}048\ (4{,}8\%)$$

- Comunicația pentru accesul datelor aflate la distanță
  - La multiprocesoare cu nuclee multiple:
    - Transferul datelor între nuclee din același circuit poate necesita ~50 cicluri de ceas
    - Transferul datelor între nuclee din circuite diferite poate necesita 100 .. 300 cicluri de ceas

Performanța arhitecturilor MIMD (6)

- Exemplul 5.2
  - Considerăm o aplicație care rulează pe un multiprocesor
  - O referință la memoria partajată necesită un timp de $t_A = 100\ ns$
  - Presupunem că pentru toate referințele la memoria locală se obțin succese
  - Valoarea CPI dacă pentru toate accesele la memoria *cache* se obțin succese: $CPI_{ideal} = 1$
  - Frecvența semnalului de ceas: 3,33 GHz

Performanța arhitecturilor MIMD (7)

- Cu cât este mai rapid multiprocesorul dacă nu este necesar accesul la memoria partajată față de cazul în care 0,2% din instrucțiuni necesită accesul la memoria partajată?

$$f = 3{,}33\ GHz = 3{,}33 \times 10^9\ Hz$$

$$t_C = \frac{1}{f} = \frac{1}{3{,}33 \times 10^9} \cong 0{,}3 \times 10^{-9}\ s = 0{,}3\ ns$$

$$CPI_{real} = CPI_{ideal} + 0{,}002 \times \frac{t_A}{t_C} = 1 + 0{,}002 \times \frac{100}{0{,}3}$$

$$= 1 + 0{,}002 \times 333{,}33 = 1 + 0{,}67 = 1{,}67$$

$$\frac{CPI_{real}}{CPI_{ideal}} = 1{,}67$$

5. Introducere în arhitecturi paralele

- Tipuri și nivele de paralelism
- Clasificarea arhitecturilor paralele
- Arhitecturi vectoriale
- Arhitecturi SIMD
- Arhitecturi sistolice
- Arhitecturi MIMD
- Arhitecturi specifice unor domenii
- Arhitecturi cu fire de execuție multiple

Arhitecturi specifice unor domenii

- Arhitecturi specifice unor domenii
  - Prezentare generală
  - Microsoft Catapult
  - Google Tensor Processing Unit

Prezentare generală (1)

- Îmbunătățirea semnificativă a arhitecturilor generale este dificilă
  - Consumul de energie este limitat
  - Trebuie să se mărească numărul operațiilor aritmetice executate de fiecare instrucțiune
- Este necesară modificarea drastică a arhitecturilor generale
  - Arhitecturi specifice: execută o gamă redusă de operații, dar cu eficiență foarte ridicată
  - Extind arhitecturile generale cu funcții specifice

Prezentare generală (2)

- Unele inovații arhitecturale nu sunt eficiente pentru toate domeniile
  - Exemplu: memoriile *cache* nu sunt eficiente pentru aplicații la care datele au un grad redus de reutilizare
  - Resursele necesare pentru aceste inovații se pot reutiliza într-un mod mai eficient
- Principii de proiectare a arhitecturilor specifice unor domenii
  - Cresc eficiența utilizării spațiului
  - Simplifică arhitectura și reduc costurile

Prezentare generală (3)

- 1. Utilizarea unor memorii dedicate
  - Memorii de lucru adaptate funcțiilor specifice
  - Minimizează distanța de la care trebuie transferate datele
  - Se pot utiliza în locul memoriilor *cache*
- 2. Reutilizarea resurselor eliberate prin eliminarea unor optimizări arhitecturale
  - Mai multe unități aritmetice
  - Memorii de dimensiuni mai mari
- 3. Utilizarea celui mai simplu tip de paralelism
  - Exemplu: paralelismul de tip SIMD este mai simplu de utilizat decât paralelismul de tip MIMD

Prezentare generală (4)

- 4. Reducerea dimensiunii datelor și utilizarea unor tipuri mai simple de date
  - Se îmbunătățește utilizarea memoriei integrate în circuitul procesorului
  - Va fi disponibil mai mult spațiu pentru unitățile aritmetice
- 5. Utilizarea unui limbaj sau platformă de programare specifică domeniului
  - Exemple: OpenCL (*Open Computing Language*) pentru sisteme eterogene (UCP, GPU, DSP, FPGA); TensorFlow pentru rețele neuronale artificiale

Arhitecturi specifice unor domenii

- Arhitecturi specifice unor domenii
  - Prezentare generală
  - Microsoft Catapult
  - Google Tensor Processing Unit

Microsoft Catapult (1)

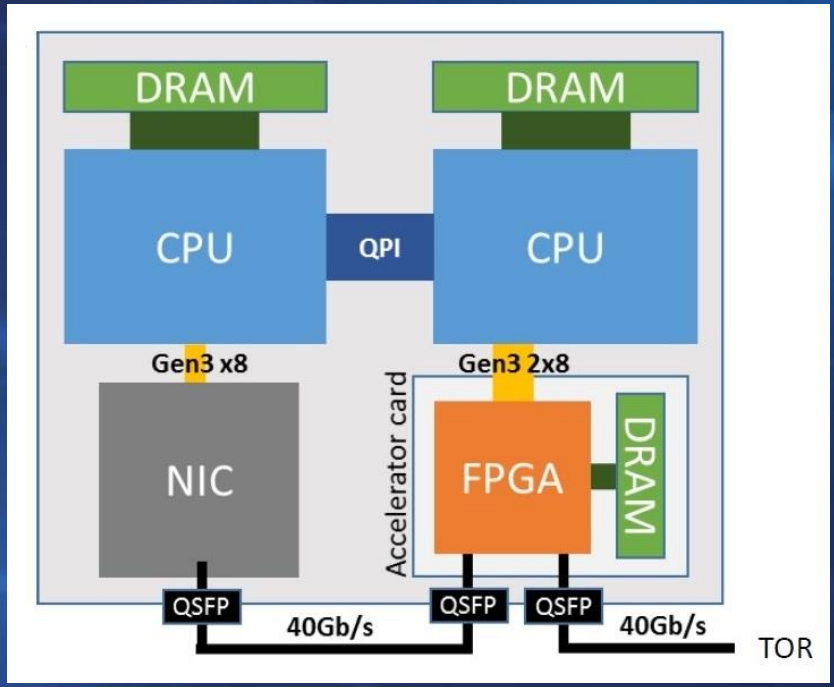
- Accelerator configurabil pentru serverele din centrele de date Microsoft
  - Placă de extensie PCI Express
  - Conține un circuit FPGA
- Catapult v1
  - Primul prototip dezvoltat în 2011
  - Circuit FPGA Altera Stratix V
  - Memorie DDR3-1600 (8 GB), *flash* (32 MB)
  - 48 de circuite FPGA conectate direct printr-o rețea secundară

Microsoft Catapult (2)

- Proiect pilot pentru 1.632 servere (2012)
- S-a utilizat pentru accelerarea funcției de ordonare a motorului de căutare Bing
- Catapult v2
  - Dezvoltat începând cu anul 2014 pentru a elimina limitările primei versiuni
  - Circuitele FPGA sunt amplasate între adaptorul de rețea și comutatoarele de rețea
    - Întregul trafic al rețelei trece prin circuitele FPGA
    - Protocol de comunicație LTL (*Lightweight Transport Layer*), cu întârziere foarte redusă

Microsoft Catapult (3)

- Toate circuitele FPGA pot comunica între ele în 20 μs (la 250.000 servere)
- Există și două conexiuni PCI Express cu procesoarele serverului
- Procesoare: Intel Xeon
- Conexiune QPI (*QuickPath Interconnect*)
- Interfețe de rețea cu conectori QSFP+ (*Quad Small Form-factor Pluggable*)



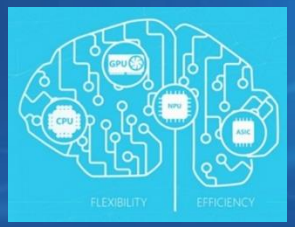
Microsoft Catapult (4)

- Plăcile Catapult v2 se pot utiliza ca acceleratoare locale, de rețea sau globale
- Acceleratoare locale
  - Utilizate pentru motorul de căutare Bing, rețele neuronale artificiale etc.
- Acceleratoare de rețea
  - Utilizate pentru servicii de rețea: inspecția pachetelor, detecția intruziunilor, criptare
- Acceleratoare globale
  - Acceleratoarele neutilizate de serverele gazdă pot fi puse la dispoziție pentru alte aplicații

Microsoft Catapult (5)

- Exemple de utilizare pentru Catapult v2
  - Instalate în serverele Bing și Azure (din 2015)
  - Proiectul *Azure Accelerated Networking*
  - Proiectul *Brainwave* pentru accelerarea aplicațiilor de inteligență artificială
  - Pun la dispoziție micro-servicii hardware
    - Se alocă un număr de circuite FPGA pentru accelerarea diferitelor aplicații
  - Utilizate de grupurile *Microsoft Research*, *Azure Machine Learning*, *Azure Storage* etc.

Microsoft Catapult (6)



- Proiectul *Brainwave*
  - Destinat accelerării etapei de inferență (predicție) pentru rețelele neuronale profunde (DNN – *Deep Neural Network*)
  - Utilizează acceleratoarele Catapult v2
    - Circuite FPGA Intel Stratix 10, Intel Arria 10
  - Folosește micro-serviciile hardware pentru paralelizarea modelelor DNN antrenate
  - Arhitectura permite răspunsuri în timp real
    - Căutare inteligentă, interacțiune cu utilizatorul, traducere automată, recunoașterea vorbirii

Microsoft Catapult (7)

- Utilitarele *Brainwave*
  - Convertesc un model DNN într-un graf
  - Noduri: operații tensoriale, muchii: fluxul de date
  - Pot diviza graful DNN în mai multe sub-grafuri pe baza unor constrângeri
- Arhitectura permite accelerarea modelelor care necesită utilizarea intensivă a memoriei
  - Ponderile necesare modelului se păstrează în memoria circuitului FPGA → rate de ordinul TB/s
  - Dacă memoria circuitului FPGA nu este suficientă, utilizatorul poate aloca alte circuite FPGA

Microsoft Catapult (8)

- Unitatea de procesare neuronală
  - NPU (*Neural Processing Unit*)
  - Procesor sintetizat prin resursele circuitului FPGA
  - Model de programare secvențial
  - Set de instrucțiuni CISC extensibil
  - Arhitectură vectorială “mega-SIMD”: o singură instrucțiune poate produce > 1 milion de operații
  - Conține un procesor de control și o unitate de execuție neuronală NFU (*Neural Functional Unit*)
  - NFU conține o unitate matriceală-vectorială (MVU) și unități multifuncționale (MFU)

Microsoft Catapult (9)

- Unitatea MVU (*Matrix-Vector Unit*) conține unități MAC (*Multiply and Accumulate*) organizate în unități paralele de calcul al produselor scalare
  - Până la 80.000 unități MAC (Intel Stratix 10 280)
  - Matricea de activare: intrările stratului de neuroni (linii)
  - Matricea ponderilor: ponderile neuronilor (coloane)
  - Ponderile sunt păstrate în memoria circuitului FPGA (~30 MB la circuitul Stratix 10 280; 35 TB/s la 600 MHz)
- Unitățile MFU (*Multifunctional Unit*) execută operații de înmulțire și adunare cu vectori intermediari și implementează funcțiile de activare

Microsoft Catapult (10)

- Se utilizează tipuri de date în VM ne-standard, cu precizie redusă
  - ms-fp8: 8 biți, cu 2 biți pentru mantisă
  - ms-fp9: 9 biți, cu 3 biți pentru mantisă
  - Permit o gamă extinsă și creșterea performanțelor față de formatele întregi (până la 7,8 ori față de 16 biți)
- Performanța
  - Unitatea NPU: 39,5 TFLOPS (Intel Stratix 10 280 la 300 MHz) (performanța la vârf: 48 TFLOPS)
  - Timpul de răspuns la căutarea inteligentă Bing
    - Modelul DNN Turing Prototype 1 (TP1): 0,85 ms (9 ms dacă se utilizează doar UCP cu un model redus)
    - Modelul DNN DeepScan: 5 ms (15 ms doar cu UCP)

Arhitecturi specifice unor domenii

- Arhitecturi specifice unor domenii
  - Prezentare generală
  - Microsoft Catapult
  - Google Tensor Processing Unit

Google Tensor Processing Unit (1)

- TPU (*Tensor Processing Unit*)
- Circuit integrat specific aplicației destinat accelerării rețelelor neuronale DNN
- Două variante de circuite TPU
  - Cloud TPU: instalate în centrele de date Google (din 2015)
  - Edge TPU: utilizate ca acceleratoare în diferite tipuri de calculatoare și dispozitive (din 2018)
- Programarea: prin platforma TensorFlow (Cloud TPU) sau TensorFlow Lite (Edge TPU)

Google Tensor Processing Unit (2)

- Un circuit Cloud TPU are rol de coprocesor pentru procesorul serverului
  - Primește instrucțiunile CISC de la procesorul serverului prin magistrala PCI Express
  - Placă de extensie PCI Express cu 4 circuite
- Componente:
  - Unitate scalară, unitate vectorială
  - Memorie HBM (2 x 8 GB sau 2 x 16 GB)
  - Unitate de înmulțire matriceală MXU (*Matrix Multiply Unit*)

Google Tensor Processing Unit (3)

- Unitatea de înmulțire matriceală MXU
  - Matrice bidimensională de unități MAC
    - 128 x 128 unități MAC
    - Poate executa 16.384 operații de înmulțire și adunare în fiecare ciclu de ceas
    - Performanța: 22,5 TFLOPS
    - Date în VM în formatul bfloat16
      - Semn, 8 biți pentru exponent, 7 biți pentru mantisă
  - Arhitectură sistolică
    - După încărcarea ponderilor și a datelor de intrare, nu mai sunt necesare accese la memorie

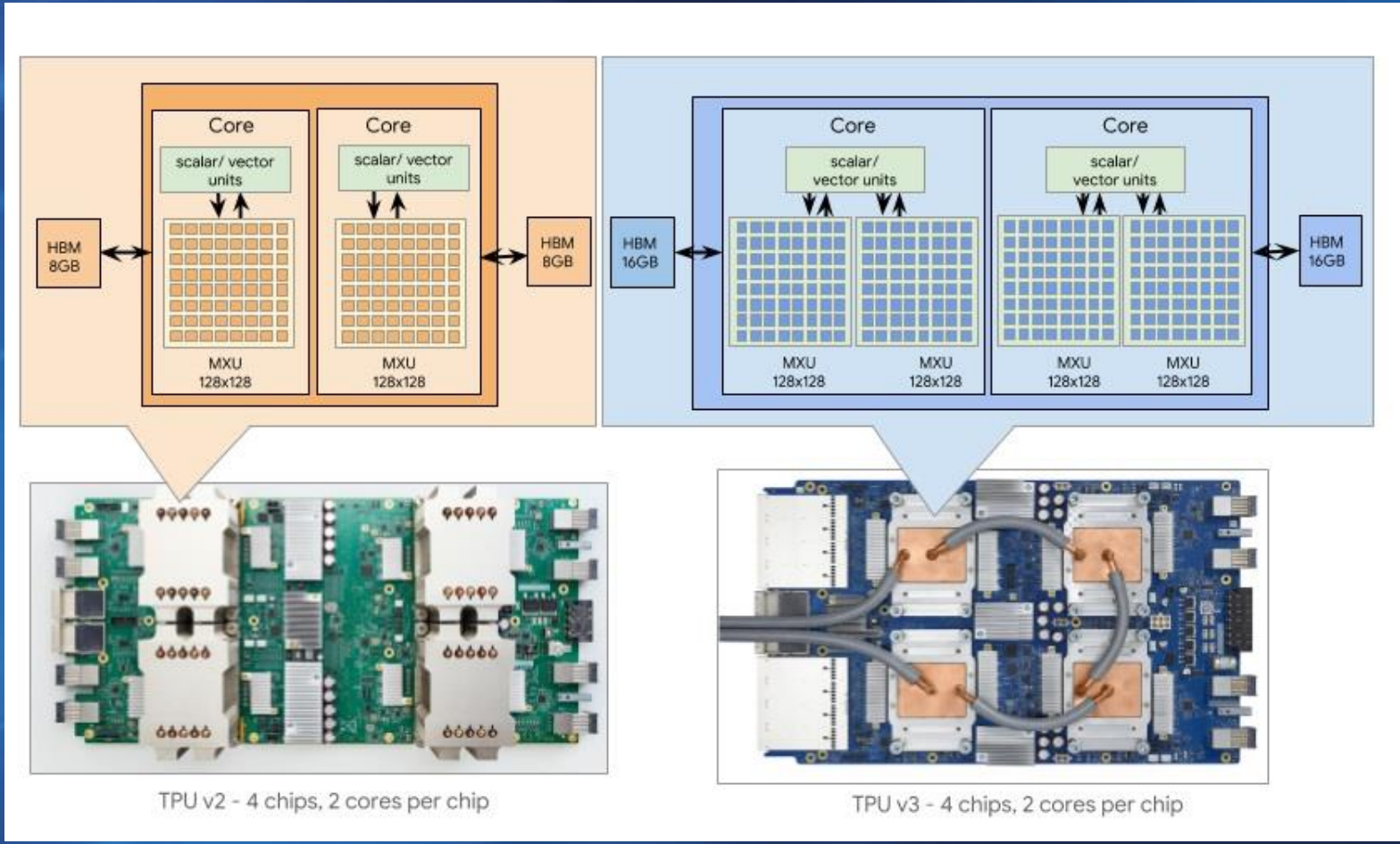
Google Tensor Processing Unit (4)

- Versiuni ale circuitelor Cloud TPU
  - Versiunea inițială (2015)
    - Unitate MXU cu 256 x 256 unități MAC
    - Date întregi de 8 biți
    - Memorie DDR3-2133, 8 GB (34 GB/s)
  - Cloud TPU v2 (2017)
    - Două nuclee TPU (45 TFLOPS)
    - Câte o unitate MXU pentru fiecare nucleu
    - Memorie HBM, 8 GB pe nucleu (~600 GB/s)
    - 4 circuite TPU pe o placă de extensie PCI Express

Google Tensor Processing Unit (5)

- Cloud TPU v3 (2018)
  - Răcire cu apă
  - Două nuclee TPU
  - Câte două unități MXU pentru fiecare nucleu
  - Performanța circuitului TPU: 90 TFLOPS
  - Memorie HBM, 16 GB pe nucleu
- Cloud TPU v4 (2020)
  - Performanța circuitului TPU: dublată
  - Performanța unui set de 64 circuite: de 2,7 ori mai mare față un set echivalent Cloud TPU v3 pentru setul de evaluare MLPerf v0.7

Google Tensor Processing Unit (6)



Google Tensor Processing Unit (7)

- Interconectarea plăcilor Cloud TPU
  - Rețea toroidală de viteză ridicată
  - Grup de plăci Cloud TPU și servere (“*Pod*”)
  - Cloud TPU v2
    - Maxim 64 plăci Cloud TPU (512 nuclee TPU, 4 TB) și 64 plăci de servere (128 procesoare)
    - Performanță de 11,5 PFLOPS
  - Cloud TPU v3
    - Maxim 256 plăci Cloud TPU (2048 nuclee TPU, 32 TB) și 128 plăci de servere (256 procesoare)
    - Performanță de ~92 PFLOPS

Google Tensor Processing Unit (8)

© Google LLC

Google Tensor Processing Unit (9)

- Utilizarea acceleratoarelor Cloud TPU
  - Motorul de căutare, algoritmul RankBrain
  - Google Assistant
  - Google Translate
  - Google Photos
  - Google Street View – procesarea textelor
  - Programul AlphaZero – crearea programelor pentru jocurile de șah, Shogi și Go
  - Competiția AlphaGo vs. Lee Sedol (2016)

Google Tensor Processing Unit (10)

- Edge TPU
  - Module destinate accelerării etapei de inferență a modelelor de inteligență artificială
  - Modul accelerator Coral
    - Performanța: 4 TOPS (8 biți)
    - Consum: 2 W
    - Dimensiuni: 15 x 10 x 1,5 mm
    - Frecvențe de funcționare: 125 MHz; 250 MHz; 500 MHz
    - Exemplu de model: MobileNetV2 (clasificarea obiectelor din imagini)

Google Tensor Processing Unit (11)

- Accelerator conectat prin interfața USB
  - Conține un modul Coral
  - Interfață USB 3.0
  - Conector USB Type-C
  - Dimensiuni: 65 x 30 x 8 mm
  - Sisteme de operare: Linux, macOS, Windows

- Accelerator dual M.2
  - Două module Coral (8 TOPS)
  - Două interfețe PCI Express Gen2
  - Dimensiuni: 30 x 22 x 2,8 mm

Rezumat (1)

- Arhitecturile MIMD sunt mai generale față de alte arhitecturi paralele, dar programarea lor este mai dificilă
- Categorii principale de arhitecturi MIMD: cu memorie partajată și cu transmitere de mesaje
- Arhitecturile MIMD cu memorie partajată se numesc și multiprocesoare sau arhitecturi cu memorie uniformă
  - Programarea este simplă, dar scalabilitatea este redusă din cauza conflictelor la accesul memoriei

Rezumat (2)

- Arhitecturile MIMD cu transmitere de mesaje se numesc și multicalculatoare sau arhitecturi cu memorie distribuită
  - Scalabilitatea este îmbunătățită
- Arhitecturile specifice unor domenii au eficiență ridicată pentru unele tipuri de operații
  - Arhitectura Microsoft Catapult utilizează circuite FPGA pentru accelerarea diferitelor aplicații
  - Circuitele Google TPU sunt specializate pentru accelerarea aplicațiilor care utilizează rețele neuronale DNN

Noțiuni, cunoștințe (1)

- Prezentare generală a arhitecturilor MIMD
- Prezentare generală, avantaje și dezavantaje ale arhitecturilor cu memorie partajată
- Soluții pentru îmbunătățirea scalabilității arhitecturilor cu memorie partajată
- Prezentare generală, avantaje și dezavantaje ale arhitecturilor cu transmitere de mesaje
- Arhitecturi cu memorie distribuită și memorii *cache*
- Multiprocesoare formate dintr-un singur circuit cu nuclee multiple de procesare

Noțiuni, cunoștințe (2)

- Evaluarea performanței arhitecturilor MIMD
- Principii de proiectare pentru arhitecturile specifice unor domenii
- Arhitectura acceleratoarelor Microsoft Catapult v2
- Unitatea de procesare neuronală utilizată de proiectul Microsoft Brainwave
- Unitatea de înmulțire matriceală a circuitelor Google TPU
- Versiunile v2 și v3 ale circuitelor Google TPU
- Interconectarea plăcilor Google Cloud TPU