

---

# I. Introdurre

---

---

# Continut

- Calculatoare paralele
  - De ce calcul paralel
  - Exemple de aplicatii
  - Istorie scuita
  - A porta sau a nu porta
  - Performanta sistemelor paralele:
-

---

# Calculatoare paralele

---

# Calculator paralel

- Un *calculator paralel* este o colectie de elemente de procesare care coopereaza si comunica pentru a rezolva rapid probleme de dimensiuni mari

=>Intrebari:

1. Cat de mare este colectia?
2. Cat de puternice sunt elementele de procesare individuale
3. Poate fi numarul elementelor crescut intr-o maniera directa?
4. Cum coopereaza si cum comunica?
5. Cum sunt transmise datele inter procesoare, ce interconectare este disponibila, ce operatii sunt disponibile pentru actiune pe diferite procesoare?
6. Care sunt primitivele (abstracte) oferite de hardware si software pentru programator?
7. Cum sunt acestea translatate in performanta?

# Comentarii generale

- Masinile paralele ocupa un spatiu de design bogat si divers: de la mic la mare
- Domenii ale procesarii paralele
  - Se refera la metodele arhitecturale si algoritmice pentru cresterea performantei sau altor attribute (ex. incredere) ale calculatoarelor prin forme variate de concurenta
- Schimbari istorice:
  - Acum 30 de ani: calculul paralel a pornit din laboratoarele de institutelor de cercetare si firme care se ocupau de tehnologii la frontiera
    - Mijloc 90': rare si pentru rezolvarea unor probelem critice
  - Astazi majoritatea calculatoarelor personale au incorporate unitati multiple de procesare:
    - Se prevede proliferarea in curand a sistemelor cu 64 de coreuri
    - Calculatoarelor paralele devin omniprezente.

---

# Clarificarea termenilor

- Procesarea paralela, *in sens literal* al termenului, este utilizata virtual in fiecare calculator modern:
    - Suprapunerea I/O cu calculele este o forma de procesare paralela,
    - Suprapunerea intre pregatirea instructiunii si executia intr-un procesor pipeline.
    - Utilizarea unor unitati functionale multiple (ex., ALU separate pentru intregi si virgula mobila sau doi multiplicatori in virgula mobila intr-un ALU)
    - multitasking (care permite suprapunerea intre calcule si incarcarea memoriei necesara la o eroare de pagina).
    - Microprogramare orizontala, si incarnarea sa de nivel inalt in calculatoarele cu very-long-instruction-word (VLIW)
  - In acest curs, termenul de *procesare paralela* este utilizat in urmatorul sens:
    - A dispune de procesoare multiple (uzual identice) pentru calcule si nu pentru I/O sau alte activitati cu perifericele.
-

---

Ce este calculul paralel?

---

---

# Efort mare

Dezvoltarea de software pentru sisteme paralele este in mod traditional consumatoare de timp si effort, datorita:

- ❑ Complexitatii inerente si coordonarea sarcinilor concurente,
  - ❑ Lipsa de algoritmi portabili, medii standardizate si unelte de dezvoltare software.
-



# Motivatii pentru procesarea paralela

- 1. Viteza mare, sau rezolvarea rapida a problemelor.**
  - Importanta cand aplicatiile au termene “hard” sau “soft”.
  - De exemplu, necesitatea de a oferi o previziune a vremii pe urmatoarele 24 de ore in catea ore sau avertizarea in timp la producerea de tornade.
- 2. Putere de calcul mare, sau rezolvarea problemelor de dimensiuni mari.**
  - Efectuarea de simulari pentru perioade lungi de timp (ex., 5-zile, in loc de 24-de ore pentru previzunea vremii).
- 3. Patrundere mai mare, sau rezolvarea de mai multe instante ale acelorasi probleme.**
  - Important cand sunt necesare efectuarea de numeroase sarcini similare.
  - De exemplu, bancile si linii aeriene, printre altele, utilizeaza sisteme de procesare de tranzactii care trateaza un volum mare de date.

---

# Scopul ultim: accelerarea

- Factorul de **accelerare** a calculului comparativ cu un sistem uni-procesor.
    - Eficienta ideala in sistemele paralele este aceea care atinge un factor de accelerare  $p$  utilizand  $p$  procesoare.
    - Desi in majoritatea cazurilor aceasta valoare nu poate fi atinsa, o anumita accelerare este in general posibila.
    - Castigul in accelerare depinde de arhitectura utilizata pentru sistem si de algoritmul care se ruleaza pe aceasta.
  - Pentru o sarcina care este imposibil sa fie efectuata pe un singur procesor datorita timpul excesiv de rulare, factorul de accelerare a calculului poate fi considerat mai mare decat  $p$  sau chiar infinit.
    - Analog unei echipe de muncitori la o masina care o rezolva in cateva minute, pe cand unul nu se poate descurca singur,
    - Referit ca ***sinergie paralela***.
-

---

# Mai rapid = mai bun?

- Avantaj de piata
  - Salveaza vietii
  - A face posibil imposibilul
    - Modelarea mediului
    - Explorarea spatiului
    - Cercetare biologica
-

# Legea lui Moore

- Creșterea performanței/vitezei microprocesorului cu un factor de 2 la fiecare 18 luni (sau aprox. 60% pe an) este cunoscută ca **legea lui Moore**.
- Această creștere este rezultatul combinării a doi factori:
  1. Creșterea în complexitate (legată atât de o densitate mai mare a dispozitivului dar și o dimensiune mai mare) a cipurilor VLSI
  2. Introducerea, și îmbunătățirea, caracteristicilor arhitecturale precum
    - Memorii cache pe cip, buffere mari pentru instrucțiuni, mai multe instrucțiuni lansate per ciclu, multithreading, pipeline-uri mai mari, execuția de instrucțiuni în avans

---

# Legea lui Moore: bazata pe obs. empirice

- 1965: rationamentul se baza pe o relatie empirica log-liniara intre complexitatea dispozitivului si timp, observata folosind trei date.
  - Relatia empirica a rezistat de-a lungul anilor atat pentru procesoare cat si pentru DRAMuri.
  - Legea a fost extrapolata pentru a afirma ca, cantitatea de putere de calcul la un cost dat se dubleaza la aproximativ fiecare 18 luni.
-

---

# Limitările legii lui Moore

- Limitele fizice sunt impuse de viteza de proagare a semnalelor pe un cablu:
    - Acest argument este referit adesea ca **argumentul vitezei luminii (sau limita)**
  - In contextul motivarii calculului paralel, acesta este considerat **argumentul puterii de calcul**
-

# Argumentul vitezei luminii

- Viteza luminii este de aprox. 30 cm/ns.
- Semnalele calatoresc pe un cablu la o fractiune a vitezei luminii.
- Daca diametrul unui cip este de 3 cm, sa zicem, orice calcul care implica transmitere de semnale de la un cap al cipului la altul nu pot fi executate mai rapid decat de  $10^{10}$  ori pe secunda.
- Reducand distantele cu un factor de 10 sau 100 se va ajunge la o crestere a limitei de mai sus cu acest factor; tot nu putem merge mai departe decat  $10^{12}$  calcule pe secunda.
- Pentru a lega limita de mai sus cu rata de executie a instructiunilor (MIPS sau FLOPS), trebuie sa estimam distanta pe care semnalele *trebuie* sa o parcurga in cadrul unui ciclu de instructiune.
- Astfel nu suntem departe de limitele impuse de viteza semnalului de propagare si alte limite fizice.

---

# Consecinta

- Argumentul vitezei luminii sugereaza ca odata ce limitele de mai sus au fosta tinse, **singura cale** pentru a imbunatati performanta este utilizarea unor procesoare multiple.
  - Acelasi argument poate fi invocat pentru a concluda ca orice sistem paralel va fi **limitat de viteza cu care procesoarele diferite pot comunica** intre ele.
    - Totusi, deoarece comunicatiile nu sunt la fel de frecvente ca si calculele, aceasta limita este mai putin serioasa.
    - Pentru numeroase aplicatii, un numar mare de pasi de calcul pot fi efectuati intre doi pasi consecutivi de comunicare, astfel amortizand surplusul datorat comunicarii.
-



---

# Argumentul vitezei memoriei/diskului

- Viteza de calcul este determinata nu numai de viteza procesorului, ci si de abilitatea sistemului de memorii de a-i furniza date.
  - Desi frecventa ceasului pentru procesoare a crescut cu aprox. 40% per an in ultima decada, timpul de acces la DRAM s-a imbunatatit cu o rata de aproape 10% per an in acest interval.
  - Cuplata cu cresterea numarului de instructiuni executate per ciclu de ceas, aceasta diferenta intre viteza procesorului si cea a memoriei este o gatuire insemnata a performantei.
-

# Argumentul vitezei memoriei/diskului

- Performanta sistemului memoriilor este determinata de fractia de cereri asupra memoriei care poate fi satisfacuta de cache.
- Platformele paralele in mod tipic au o performanta buna a memoriei deoarece ofera
  - (i) Cache agregat mai mare si
  - (ii) Latime de banda agregata mai mare la sistemul de memorii(amandoua de obicei liniare cu nr. de procesoare).
- Inima algoritmilor paraleli o constituie asigurarea localitatii datelor care sunt referite

# Argumentul comunicarii datelor

- Pe masura ce infrastructura rețelor evoluează, se conturează treptat viziunea utilizării Internetului ca un mediu de calcul paralel/distribuit vast .
- În numeroase aplicații există constrângeri asupra localității catelor și/sau resurselor în Internet.
  - Un exemplu de asemenea aplicație este explorarea datelor într-un set vast de date comerciale distribuit într-o rețea cu lățime de bandă redusă.
  - În asemenea aplicații, chiar dacă puterea de calcul este disponibilă pentru a realiza taskul solicitat fără a utiliza calculul paralel, este nefezabil ca datele să fie colectate la o locație centrală.
  - În asemenea cazuri, motivarea paralelismului vine nu numai din necesitatea de resurse de calcul dar și din infeasibilitatea unei abordări (alternative) centralizate.

---

# Exemple de aplicatii

---

---

# Exista o necesitate de a rezolva probleme de dimensiuni mari

- Predictia evolutiei mediului terestru
  - Testarea armelor nucleare
  - Chimie cunatica
  - Biologie computationala
  - Explorarea datelor pentru multimi de date foarte mari
  - Astronomie si cosmologie
  - Criptografie
  - Algoritmi aproximativi pentru probleme *NP*-complete etc
-

---

# Animatie pe calculator

- Randarea (Rendering) este pasul in care informatia din fisierele animatiei, precum iluminarea, texture, umbrirea sunt aplicate la modele 3D pentru a genera imagini 2D care constituie cadre ale filmului.
  - Calculul paralel este esential pentru a genera numarul necesar de cadre (24 fps) pentru un film de lungime medie.
  - 1995 - Pixar: Toy Story, primul film complet generat pe calculator, a fost procesat utilizand o "ferma de randare" constituita din 100 de masini cu procesor dual.
  - 1999 – Pixar: Toy Story 2, un sistem cu 1400 de procesoare a fost folosit pentru procesare – reflectat in detaliile imbunatatite in ceea ce priveste texturile, imbracamintea, efectele atmosferice.
  - 2001: Monsters, Inc. a utilizat un sistem cu 250 servere fiecare cu 14 procesoare – total 3500 procesare.
-

---

# Științele biologice

- Ultimele realizări se bazează pe disponibilitatea informațiilor legate de secvențele DNA a unor organisme diverse, incluzând și specia umană.
  - Celera Corp.: algoritm paralel pentru genomica umană
    - Ideea este aceea de a împărți genomul în segmente mici, determinarea experimentală a secvențelor DNA a segmentelor, și apoi utilizarea calculatorului pentru a construi întreaga secvență din segmente prin găsirea zonelor de suprapunere.
    - Facilitățile de calcul utilizate de Celera pentru secvențierea genomului uman au inclus 150 servere cu 4 procesoare și unul cu 16 procesoare și 64GB de memorie.
    - Calculul a implicat 500 milioane de trilioane de comparații bază cu bază.
-

---

# Astrofizica

- Explorarea evolutiei galaxiilor,
  - procese termo-nucleare,
  - analiza seturilor de date foarte mari de la telescoape.
    - Seturile de date Sky Survey (precum Sloan Digital Sky Surveys) reprezinta unele dintre cele mai mari seturi de date stiintifice.
    - Analiza efectiva a acestor seturi de date necesita o putere de calcul extrem de mare
-



# Simulari numerice – un exemplu

- Pentru a descoperi cum transporta caldura oceanele din sud catre Polul Sud, se dezvoltă următorul model:
  - Oceanul este divizat în 4096 regiuni E–V, 1024 regiuni N–S, și 12 nivele în adâncime (celule 50 M 3D).
  - O singură iteratie a modelului simulează circulația oceanului pentru 10 minute și implică 30B operații în virgulă mobilă.
  - Pentru a realiza simularea pentru 1 an, sunt necesare aproximativ 50 000 iteratii.
  - Simularea pentru 6 ani implică  $10^{16}$  operații în virgulă mobilă.

---

# Inginerie si proiectare

- Proiectoarea aripilor de avion: optimizare lift-drag, stabilitate,
  - Motoare cu combustie interna: optimizarea distributiei incarcarii, ardere,
  - Circuite de viteza mare: studiul intarzierilor si efectelor inductive,
  - Structuri: optimizarea integritatii structurale, parametrii de proiectare, cost,
  - Proiectarea de sisteme microelectro-mecanice si nanoelectro-mecanice
-

# Aplicatii comerciale

- Servere cost-efective capabile sa ofere performante in scalabilitate
  - Platformele paralele de la multiprocesoare la clustere Linux sunt utilizate frecvent ca servere Web si pentru baze de date.
  - Casele mari de brokeraj pe Wall Street trateaza sute de mii de sesiuni simultane de utilizatori si milioane de ordine.
  - Desi nu este vizibil, unele dintre cele mai mari super-calculatoare sunt gazduite de Wall Street.
  - Platforme precum IBMs SP si Sun Ultra HPC servesc aceste site-uri critice pentru afaceri.
- Disponibilitatea datelor de tranzactii pe scara mare a condus la un interes considerabil pentru exploatarea datelor si analiza pentru optimizarea afacerilor si decizii de piata:
  - Distributia geografica a acestor date necesita utilizarea unor algoritmi paraleli pentru asemenea probleme precum explorarea regulilor de asociere, grupari, clasificari si analiza seriilor in timp.

---

# Aplicatii in informatica

- In cazul detectiei intruziunii in retea, datele sunt colectate la siteuri distribuite si trebuie analizate rapid pentru semnalarea intruziunii.
    - Imposibilitatea de colectare a acestor date la o locatie centrala pentru analiza implica necesitatea utilizarea calculului paralel sau distribuit.
  - In domeniul criptografiei, unele dintre cele mai spectaculoase aplicatii ale calculului paralel se focuseaza pe factorizarea unor intregi extrem de mari.
  - Un automobil modern consta in zeci de procesoare care comunica pentru a efectua sarcini complexe pentru optimizarea performantei.
-

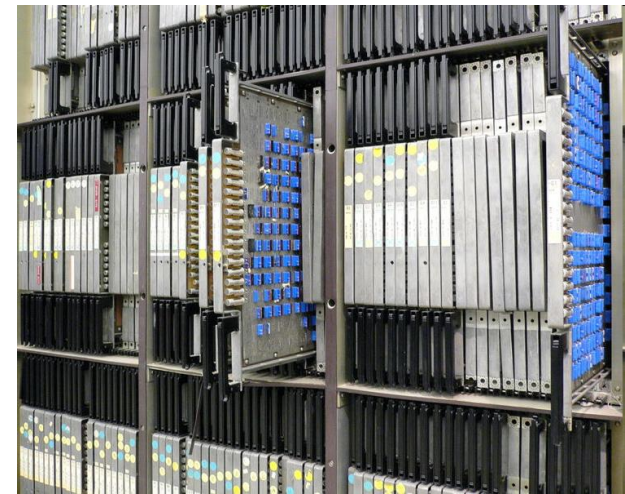
---

# Istorie scurta

---

# Sus si jos | Inceputurile

- Istoria procesarii paralele are suisurile si coborasurile sale (formare de companii si falimente) care apar cu un ciclu de 20 de ani.
- Interesul pentru calculul paralel a pornit in anii '60:
  - ILLIAC IV,
    - Construit la Universitatea din Illinois
    - Ulterior construit si operat de Burroughs Corporation,
    - Primul calculator paralel de dimensiuni considerabile;
    - Arhitectura de interconectare tip grila 2D
    - O unitate de control comuna care a fost dezvoltata pe baza teoriilor aparute la sfarsitul anilor '50.
    - A fost proiectat pana la 256 procesoare (4 cadre cu 64 de procesoare fiecare).
    - A fost construit numai un cadru de 64-procesoare
    - A demonstrat clar fezabilitatea calculatoarelor paralele
    - A relevat de asemenea dificultatile in utilizarea acestora.



# Interesul comercial

- resuscitat in anii '80:
    - Datorat interesului unitatilor federale pentru aparare din SUA
    - Companii numeroase au fost formate pentru a dezvolta sisteme paralele.
  - Trei factori au condus la o alta recesiune:
    1. Fondurile guvernamentale din SUA si alet tari s-au oprit ca urmare a terminarii razboiului rece.
    2. Utilizatorii comerciali si alte industrii bazate de exploatarea intensiva a datelor fie au fost saturate fie dezamagite de dificultatile aplicatiilor.
    3. Microprocesoarele s-au dezvoltat foarte rapid in termeni de rata performanta/cost a.i. masinile paralele au ramas in urma in ceea ce priveste efectivitatea in cost.
- => Numeroase companii nou formate au dat faliment sau si-au schimbat activitatea catre dezvoltarea de aplicatii distribuite pentru clustere de staii de lucru.

# Puncte turnante

- Sfarsitul anilor '60: profit considerabil al investitorilor, interes in arhitecturi, tehnologii si utilizarea sistemelor.
- A doua jumatate a anilor '70: introducerea sistemelor de calcul vectoriale au marcat inceputul super-calculului modern
- Prima jumatate a anilor '80: integrarea sistemelor vectoriale in mediile de calcul conventionale
- Sfarsitul anilor '80: “atacul criminalului micro” => utilizarea microprocesoarelor “off-the-shelf” in loc de procesoare speciale pentru sisteme masiv paralele
- Incepurile anilor '90: o noua generatie de sisteme masiv paralele (MPP) a aparut pe piata, reclamand egalitatea sau depasirea performantelor multi-procesoarelor vectoriale.
- Iunie 1993: listaTop500 a inceput sa ofere o baza pentru statistici legate de calculatoarele super-performante
- 1994: SGI, Digital & Sun au inceput vanzarea modelelor multiprocesoarelor simetrice (SMP) in familia statiilor de lucru pentru clienti industriali (ex. IBM SP2)



---

# Suntem intr-o perioada de urcus!

- Condusa de revolutia Internet-ului si ofertele asociate de oferire a informatiei, este prezenta o a 3a resurectie a calculatoarelor paralele.
  - Masini cu super-performante, centralizate, satisfac necesitatile de procesare si acces pentru anumiti furnizori.
  - Calculul paralel este mai mult decat o strategie pentru a atinge performanta— este o viziune incitanta despre cum pot fi scalate calculele de la un singur procesor la o putere de calcul virtual nelimitata.
-

---

A porta sau a nu porta

---

---

# Portarea unui cod spre o arhitecturi paralele

- E mai mult decat a pune un cod existent pe o masina noua.
    - Masinile paralele sunt fundamental diferite de predecesorii lor vectoriali,
    - Portarea este o oportunitate pentru
      1. Reformularea codului de baza si a structurilor de date
      2. Regandirea reprezentarii de baza a proceselor si dinamicii implicate.
-

# Dificultatea sarcinii programarii paralele

- Exprimarea unui program paralel explicit este dificila
  - Dezvoltatorul trebuie sa specifice
    - Calcului si cum acesta este partitionat intre procesoare,
    - Sincronizarea si mutarea datelor necesara pentru corectitudinea rezultatelor si pentru a atinge a performanta inalta.
- Natura sistemelor de calcul de inalta performanta se schimba rapid =>
  - Necesitatea de a explima programele intr-o maniera independemta de masina,
    - Cu alte cuvinte, programele paralele trebuie portare intre diferite arhitecturi.
      - Acest lucru este dificil deoarece pretul portabilitatii este adesea scaderea performantei

---

# Complexitatea problemelor de rezolvat

- Un factor de complicare pentru calculul paralel
  - Aceasta complexitate necesita:
    - Cunostinte deosebite din partea dezvoltatorilor de aplicatii
    - Flexibilitate deosebita in aplicatiile dezvoltate
  - Adesea aceasta inseamna ca progarmele paralele vor fi dezvoltate utilizand paradigme multiple de programare si adesea limbaje multiple.
  - Interoperabilitatea este astfel luata in considerare in limbajele de dezvoltare pentru o componenta particulara a aplicatiilor.
-

# Portabilitatea este delizorie

- La inceputurile erei calculului paralel, fiecare furnizor de sisteme paralele a construit un API diferit
  - Acest lucru a facut extrem de dificila dezvoltarea de aplicatii paralele (lucrul repetat pentru fiecare arhitectura noua).
- Standardul Message Passing Interface (MPI) a fost bine venit
- Portabilitatea nu este legata numai de implementarea unei interfete standard: majoritatea utilizatorilor sunt interesati in *performanta portabila*:
  - Abilitatea de a atinge o fractie ridicata a performantei pe fiecare masina din aceeași imagine a programului.
  - Implementarile interfetelor standard nu sunt aceleasi pe fiecare platforma, iar portabilitatea, chiar si pentru programe scres in MPI, nu este atinsa in mod automat.
  - Implementatorul trebuie sa petreaca un timp semnificativ pentru a adapta aplicatia la fiecare platforma noua.

---

# Algoritmii nu sunt intotdeauna portabili

- Un algoritm nu lucreaza bine pe fiecare arhitectura.
  - Diferentele pot aparea datorita
    - Numarului si granularitatii procesoarelor
    - Conectivitatea si latimea de banda
    - Performantele ierarhiei de memorii pentru fiecare procesor individual.
  - Librariile portabile ce contin algoritmi trebuie sa fie parametrizate pentru a permite selectia algorimilor pe baza arhitecturii in care rutinele individuale sunt rulate
-

---

# Paralelismul nu este totul

- Problema principala pe masinile scalabile, alta decat paralelismul, este mutarea datelor
  - A doua problema: latimea de banda pentru memoria principala a multi-procesoarelor cu memorie partajata
  - Algoritmii si software-ul trebuie sa trateze problemele ierarhiei de memorii care sunt fundamentale pentru programarea paralela.
-



---

# Acceptarea de catre comunitate

- Excelenta tehnica, singura, nu poate garanta ca noua abordare software va avea succes.
  - Pentru a atinge o utilizare pe scara larga, trebuie sa existe o siguranta ca sistemul software va supravietui testului timpului.
  - Standardele joaca un rol important, dar nu pot garanta succesul.
  - Un contraxemplu este HPF (High Performance Fortran).
    - HPF n-a atins nivelul de acceptare pe care l-a atins MPIul deoarece compilatoarele comerciale nu s-au maturizat in timp pentru a castiga confidenta din partea comunitatii.
    - OpenMP a avut succes deoarece a tintit piata, pe cand HPF numai performanta in cadrul unuor grupuri restranse de cercetare.
-

---

# Performanta sistemelor paralele

---

---

# Metriци de performanta pentru sistemele paralele

1. Performanta bruta de baza
  2. Eficienta masinii
  3. Modelul Hockney-Jesshope pentru procesoarele vectoriale
  4. Top 500
-

# Performanta bruta de baza - Flops

- Viteza de procesare a calculatoarelor implicate in calcule stiintifice este exprimata adesea in termeni de numar de **operatii in virgula mobila efectuate pe secunda**,
  - Masura pentru a descrie puterea de calcul a celor mai mari super-calculatoare din lume
- Pentru un timp lung, masura de baza a fost Mflops exprimata ca:  $r = N / t$  Mflops unde  $N$  reprezinta numarul de operatii in virgula mobila executate in  $t$  secunde.
  - Cand  $N$  operatii in virgula mobila sunt executate cu o viteza medie de  $r$  Mflops, timpul de executie a unui algoritm dat poate fi exprimat prin:  $t = N/r$ .
- Masura Mflop a fost depasita de masurile de ordin mai mare:
  - Gflops (gigaflops),
  - Tflops (teraflops),
  - Pflops (petaflops) =  $10^{15}$  operatii in virgula mobila per secunda.
- Operatiile in virgula mobila pot fi utilizate pentru a caracteriza un algoritm ce se executa pe o masina independenta cu caracteristici particulare de hardware, pe care algoritmii sunt executati, precum si pentru a descrie hardware-ul insusi.

# Perf. bruta de baza – perf. de varf (maxima)

- Numerosi furnizori de calculatoare paralele anunta performanta maxima, de varf, a masinilor lor
  - Aceasta este viteza maxima cu care orice algoritm poate fi potential executat pe hardware-ul lor.
  - In practica computationala (in afara unor cazuri simple, precum multiplicarea matricelor), aceasta performanta nu este atinsa.
  - In acelasi timp, totusi, indica ce performanta poate fi potential asteptata pentru o masina data.
-

# Perf. bruta de baza – benchmark

- Exista mai multe **programe de benchmark** utilizate pe scara industrială
  - Precum Whetstone, ScaLAPACK si LINPACK.
  - Utilizate intensiv in evaluarea avansata a sistemelor de calcul.
- Benchmarkuri specifice au fost dezvoltate pentru evaluarea calculatoarelor cu memorie partajata, distribuita sa hibrida, care variaza intre:
  - Cicluri paralele simple pentru a masura abilitatea de paralelizare a compilatoarelor
  - Benchmarkul PERFECT care consta intr-un set de programe ce provin din aplicatii reale pentru dinamica fluidelor,
  - Genesis care se refera la FFT, ecuatii cu derivate partiale, dinamica moleculara si algebra liniara.
  - Populare sunt benchmarkurile dezvoltate de NASA:
    - NAS parallel benchmark: contine o suite de 8 programe cu 3 versiuni pentru implementari secventiale, dependente de masina, si bazate pe MPI
    - <http://www.nas.nasa.gov/Software/NPB>.

# Eficiența mașinii – performanța teoretică

- Masuratorile descrise anterior oferă o măsură absolută a timpului de calcul utilizat de program.
  - Este interesantă cunoașterea a cât de eficiente sunt aceste calcule!
- **Eficiența** este raportul între performanța curentă și cea teoretică a sistemului.
- **Performanța teoretică** a unui calculator superscalar computer de exemplu este calculată astfel:  $R_{peak} = n_{cores} \cdot n_{FPU} \cdot f$ , unde
  - $n_{cores}$  este numărul de core-uri ale calculatorului,
  - $n_{FPU}$  este numărul de unități în virgulă mobilă per core
  - $f$  este frecvența ceasului.
  - Ex: Pentium D 830 cu 2 core-uri, 2 FPU-uri per core, frecvență 3 GHz,  
 $R_{peak} = 2 \cdot 2 \cdot 3 \cdot 10^9 \text{ FLOPS} = 12 \text{ GFLOPS}$

# Eficiența mașinii – performanța practică

- Pentru operații matrice x matrice, ( $m \times k$  vs.  $k \times n$  matrice dense) numărul de operații în virgula mobilă este aprox.  $n_{flop} = 2mnk$ ,
  - Performanța poate fi calculată prin  $R = n_{flop} / \text{Timp\_ceas}$ ,
  - Eficiența poate fi calculată prin raportul  $R/R_{peak}$ .
- Pentru CPU-uri Intel, Intel oferă o unealtă de măsurare numită VTUNE,
  - Fără plată pentru scopuri necomerciale, pentru Linux
  - Oferă funcționalitatea unui profiler,
  - Poate de asemenea măsura numărul de operații în întregi și virgula mobilă în cadrul unui program
- Numeroase supercalculatoare sunt furnizate cu contoare integrate pentru măsurarea performanței.
  - Acestea fac posibilă măsurarea performanței.



# Modelul Hockney-Jesshope pentru procesoare vectoriale

- Model actual?
  - Aplicabilitatea a fost reinviata odata cu dezvoltarea recenta a Earth Simulator, construite de corporatia NEC din procesoare vectoriale.
  - Majoritatea procesoarelor moderne sunt constituite din mia multe linii de procesare (pipeline)
    - Performantele unei asemenea linii pot fi conceptualizate in termenii modelului de performanta vectorial
- Performanta  $r_N$  unui ciclu de procesare vectoriala de lungime  $N$  poate fi exprimat in termenii a doi parametrii:
$$r_N = r_\infty / (n_{1/2}/N + 1) \text{ Mflops.}$$
  - $r_\infty$  - reprezinta performanta in Mflops pentru fiecare ciclu lung
  - $n_{1/2}$  lungimea ciclului pentru care este atinsa performanta de  $r_\infty/2$ .

# Modelul Hockney-Jesshope pentru procesare vectoriala

## ■ Exemplu: actualizarea unui vector

- Fie operatia  $x \leftarrow x + \alpha y$  (operatia AXPY)
- Poate fi explicata printr-un ciclu de lungime  $N$  in care fiecare pas consta in operatii in virgula mobila.
- Timpul de executie:

$$T_{\text{AXPY}}(N) = 2N / 10^6, r_N = 2 \times 10^{-6} (n_{1/2} + N) / r_{\infty} \text{ secunde.}$$

## ■ Acest model poate fi aplicat nu numai pentru a prezice timpul de executie a programelor vectorizate dar si pentru a dezvolta algoritmi vectoriali optimali.

- Numerosi algoritmi (ex. *divide-and-conquer*) consta din cicluri de lungimi diferite care sunt legate inter ele si care pot fi tratate ca si parametrii ai programului vectorizat.

# Top 500

- Comentariu asupra performanței calculatoarelor paralele:
    - Nu există o măsură universală,
    - Utilizarea unui singur număr pentru a caracteriza performanța, precum ar fi performanța maximă citată de furnizori, este adesea neadecvată.
    - Performanța este de obicei evaluată în termeni de rezultate ale benchmark-urilor care constă în mai mulți algoritmi și aplicații a.i. pot fi măsurate aspecte diferite ale sistemelor.
      - Totuși dependent de calitatea software-ului mai mult decât caracteristicile hardware.
    - Controversa legată de metodele de evaluare a performanței sunt recunoscute de comunitate și sunt astfel câteva încercări recente de a asigura criterii obiective ale performanței pentru calculatoare paralele.
  - O bază bună pentru evaluarea performanței este lista Top500:
    - Performanța măsurată utilizând benchmarkul LINPACK
      - Măsură este utilizată pentru a clasifica calculatoarele
      - code that solves a sys.of linear eqs, using the best software for each platform.
-

---

# Top 500

- 2004:
    - Primul: *Earth Simulator din Yokohama* care are 5120 SX-6 procesoare vectoriale de la NEC Corporation.
    - Intrările din varf se bazează pe un număr mare de procesoare comune:
      - 65536 procesoare IBM PowerPC 440 la Livermore National Laboratory;
      - 40960 procesoare IBM PowerPC la IBM Research Laboratory in Yorktown Heights;
      - 10160 procesoare Intel Itanium II conectate printr-o rețea Infiniband Network și construit de Silicon Graphics, Inc. la NASA Ames Research Centre.
  - 2010:
    - Supercalculatoarele cu PETAflop/s –  $10^{15}$  operații pe secundă sunt disponibile
-

# 11/2008

## Cray XT5



## Blue Gene



## Cluster



## SGI Altix ICE



Rank	Site	System	Cores	R <sub>max</sub>	R <sub>peak</sub>
1	DOE/NNSA/LANL United States	BladeCenter QS22/LS21 Cluster, PowerXCell8i 3.2 Ghz / Opteron DC 1.8 Ghz, Voltaire Infiniband IBM	129600	1105	1456.7
2	Oak Ridge National Laboratory United States	Cray XT5 QC 2.3 Ghz Cray Inc.	150152	1059	1381.4
3	NASA/Ames Research Center/NAS United States	SGI Altix ICE 8200EX, Xeon QC 3.0/2.66 Ghz SGI	51200	487.01	608.83
4	DOE/NNSA/LLNL United States	eServer Blue Gene Solution IBM	212992	478.2	596.38
5	Argonne National Laboratory United States	Blue Gene/P Solution IBM	163840	450.3	557.06
6	Texas Advanced Computing Center/Univ. of Texas United States	SunBlade x6420, Opteron QC 2.3 Ghz, Infiniband Sun Microsystems	62976	433.2	579.38
7	NERSC/LBNL United States	Cray XT4 QuadCore 2.3 Ghz Cray Inc.	38642	266.3	355.51
8	Oak Ridge National Laboratory United States	Cray XT4 QuadCore 2.1 Ghz Cray Inc.	30976	205	260.2
9	NNSA/Sandia National Laboratories United States	Sandia/ Cray Red Storm, XT3/4, 2.4/2.2 Ghz dual/quad core Cray Inc.	38208	204.2	284
10	Shanghai Supercomputer Center China	Dawning 5000A, QC Opteron 1.9 Ghz, Infiniband, Windows HPC 2008 Dawning	30720	180.6	233.47
11	Forschungszentrum Juelich (FZJ) Germany	Blue Gene/P Solution IBM	65536	180	222.82
12	New Mexico Computing Applications Center (NMCAC) United States	SGI Altix ICE 8200, Xeon quad core 3.0 Ghz SGI	14336	133.2	172.03
13	Computational Research Laboratories, TATA SONS India	Cluster Platform 3000 BL460c, Xeon 53xx 3Ghz, Infiniband Hewlett-Packard	14384	132.8	172.61
14	Grand Equipement National de Calcul Intensif - Centre Informatique National de l'Enseignement Supérieur (GENCI-CINES) France	SGI Altix ICE 8200EX, Xeon quad core 3.0 Ghz SGI	12288	128.4	146.74
15	National Institute for Computational Sciences/University of Tennessee United States	Cray XT4 QuadCore 2.3 Ghz Cray Inc.	17956	125.13	165.2