

Paralelizarea Simularii Procesului de Magnetizare in Nanofire

Studenti

- Andrei Stefanescu
- Mihai Bucicoiu
- Vlad Ureche

Indrumatori

- Andrei Diaconu
- Emil Slusanschi
- Alexandru Herisanu

Continut

- Prezentare Problema
- Prezentare problema 1: Simulare Nanofir Singur
 - Teorie
 - Optimizari realizate
 - Rezultate
- Prezentare problema 2: Simulare Suprafata Nanofire
 - Teorie
 - Optimizari realizate
 - Rezultate
- Concluzii

Continut

- **Prezentare Problema**
- Prezentare problema 1: Simulare Nanofir Singur
 - Teorie
 - Optimizari realizate
 - Rezultate
- Prezentare problema 2: Simulare Suprafata cu Nanofire
 - Teorie
 - Optimizari realizate
 - Rezultate
- Concluzii

Nanofire si magnetism



- Fiecare nanofir se comporta ca un mic magnet
- Apar efecte legate de spinul electronului, care la dimensiunea unui nanofir devin importante
- In functie de campul exterior nanofirul poate sa-si modifice starea de magnetizare sau chiar sa-si interschimbe polii (Nord -> Sud si invers)

Aplicatii ale nanofirelor

- Electronica/calculatoare:
 - Tranzistoare foarte rapide
 - Memorii rapide (racetrack memory)
- Energie: pot converti energia cinetica (de miscare) in energie electrica, prin efect piezoelectric
- Medicina: se folosesc la implanturi, favorizeaza atasarea tesuturilor biologice la implant
- Nanoroboti

Pentru mai multe detalii

revedeti prezentarea tinuta de D-nul Andrei Diaconu

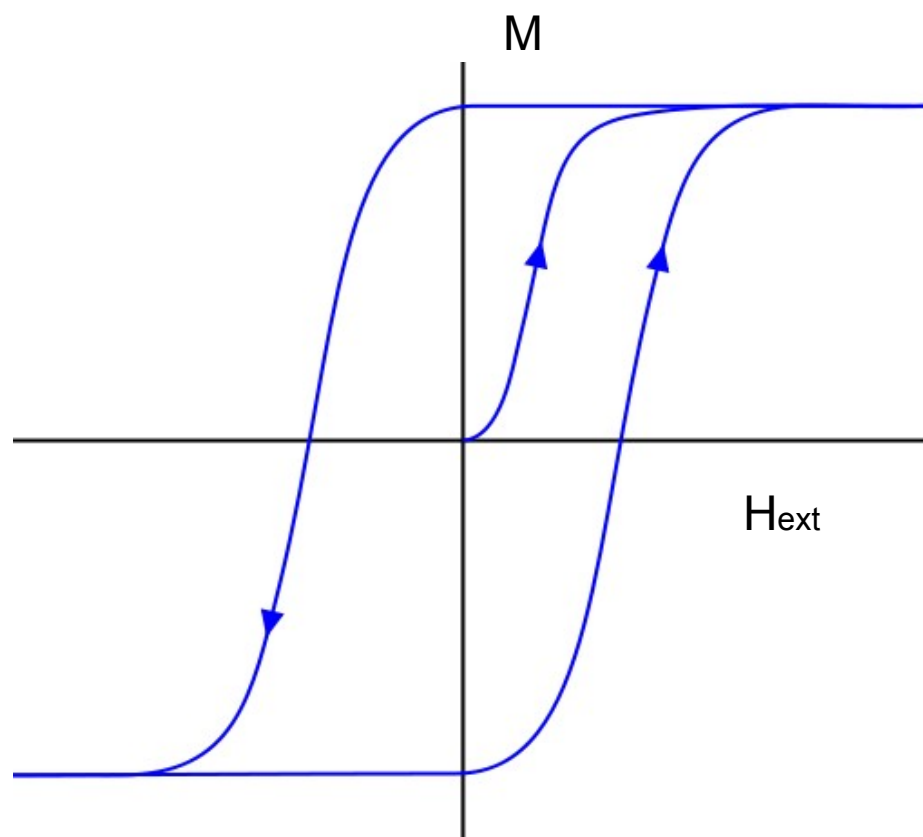
... sau ...

vizitati <http://science.howstuffworks.com/nanowire4.htm>

Continut

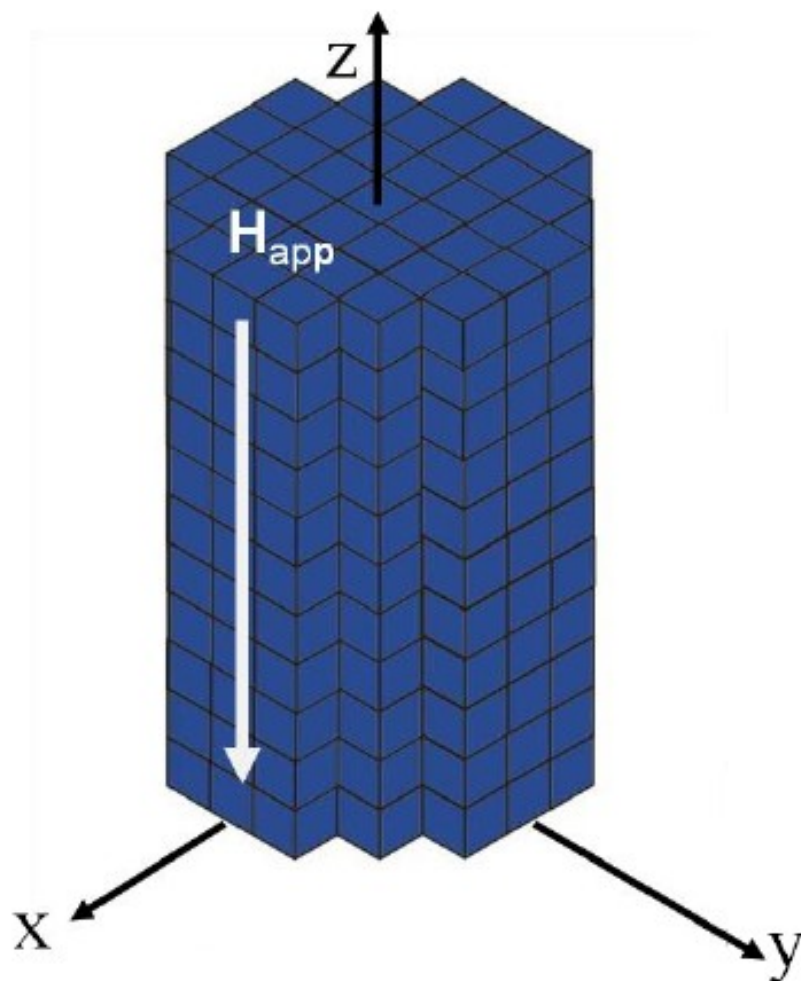
- Prezentare Problema
- Prezentare problema 1: Simulare Nanofir Singur
 - Teorie
 - Optimizari realizate
 - Rezultate
- Prezentare problema 2: Simulare Suprafata Nanofire
 - Teorie
 - Optimizari realizate
 - Rezultate
- Concluzii

Simularea unui nanofir: Teorie



- Se studiaza comportarea unui nanofir intr-un camp magnetic exterior (H_{ext})
- Comportarea nanofirului e data de marimea magnetizatie (M)
- Fenomenul care apare se numeste histerezis (nanofirul “tine minte” starea in care l-a adus campul exterior)
- Este foarte greu sa se calculeze analitic (matematic) comportarea nanofirului => se foloseste un calcul numeric

Simularea unui nanofir: Teorie 2

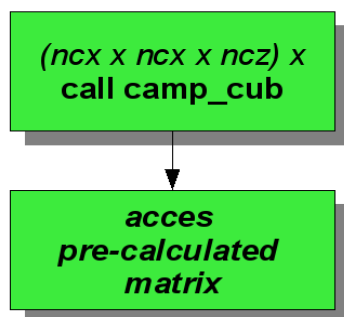
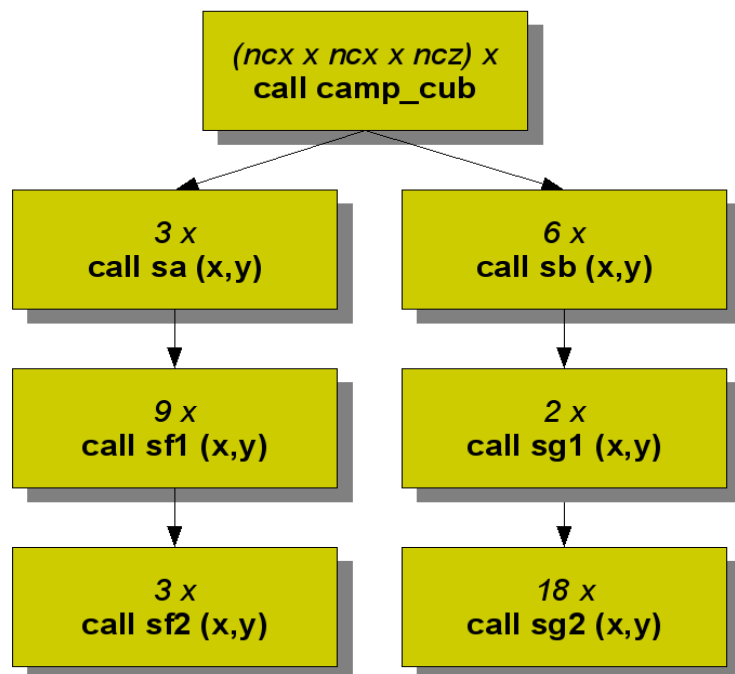


- Pentru a calcula numeric raspunsul nanofirului, acesta se imparte in bucatele mici care au magnetizare proprie
- Problema care apare e ca bucatile mici se influenteaza reciproc
- Astfel problema ajunge de dimensiune $\sim R^2 \cdot h$ unde R e raza cilindrului si h e inaltimea

Continut

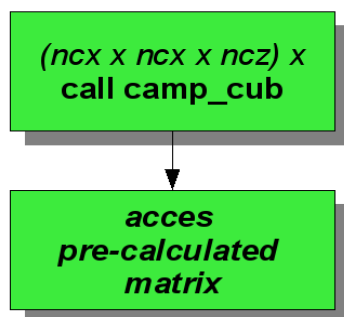
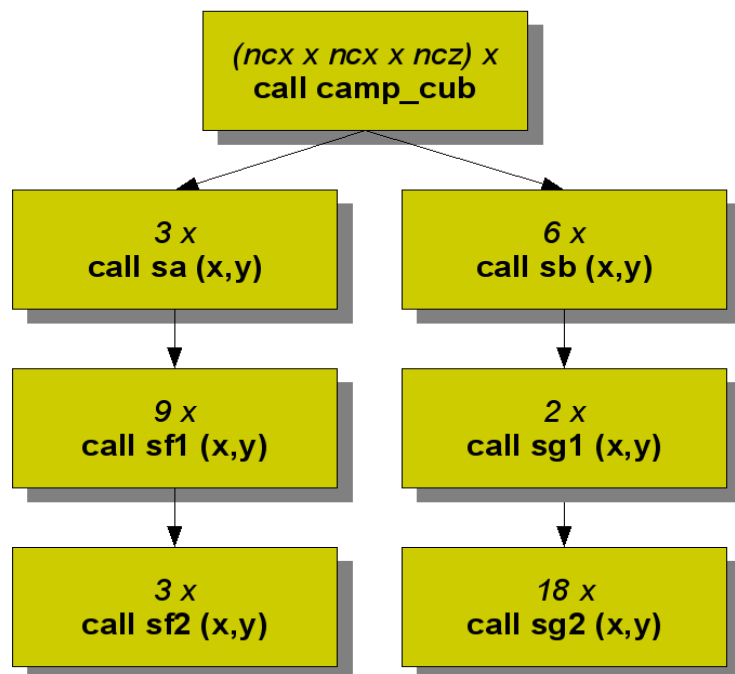
- Prezentare Problema
- **Prezentare problema 1: Simulare Nanofir Singur**
 - Teorie
 - **Optimizari realizate**
 - Rezultate
- Prezentare problema 2: Simulare Suprafata Nanofire
 - Teorie
 - Optimizari realizate
 - Rezultate
- Concluzii

Optimizare seriala - 1



- In cadrul simularii se facea un numar foarte mare de apeluri la functia camp_cub
- Din camp_cub se chemau mai multe functii care depindeau doar de pozitie (x,y)
- Am luat decizia de a calcula valorile functiilor la initializare si de a le stoca intr-o matrice
- Dar dimensiunea matricei?

Optimizare seriala - 2

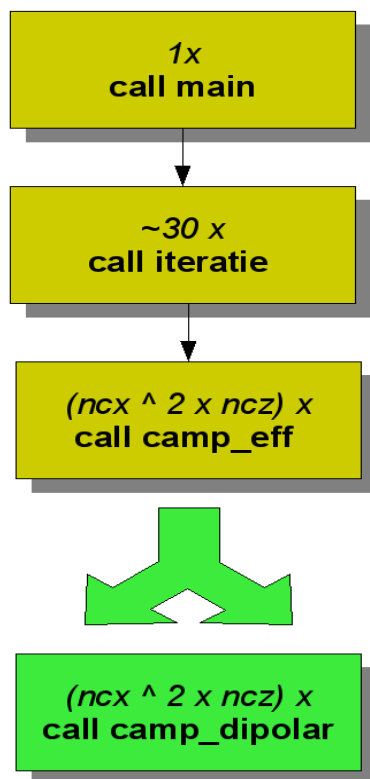


- call_camp (aa,bb,gg,x,y,z,hx,hy,hz)
 - 6 dimensiuni => matrice MARE

... insa ...

- Depinde doar de pozitia relativa => se poate face o matrice formata doar din trei dimensiuni
- (aa,bb,gg,x,y,z) => (xr,yr,zr)
- Initializarea s-a facut cu grija la modul de acces la memorie al fortran-ului => 30x speedup

Optimizare paralela



LEGEND

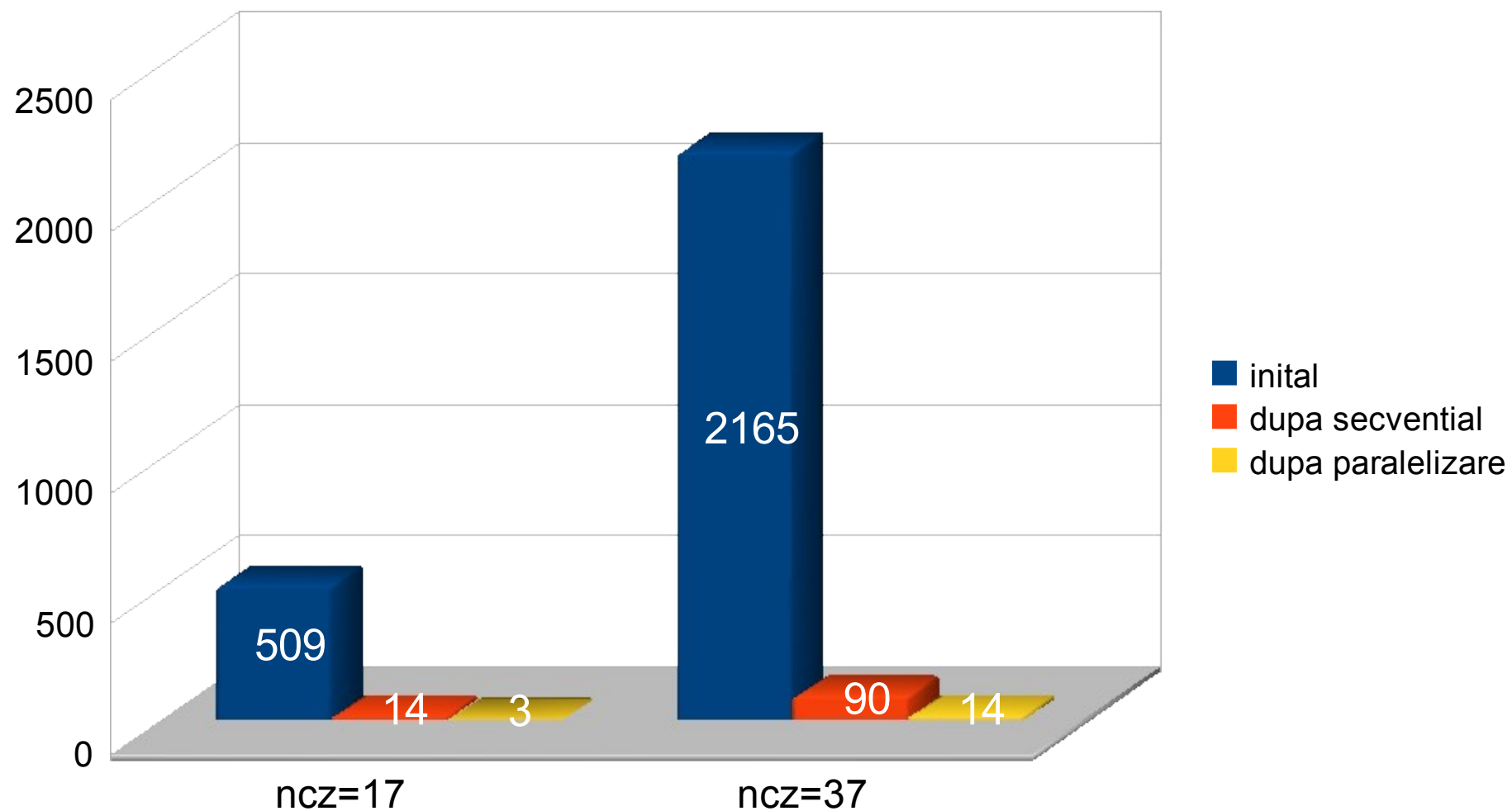
runs parallel on all machines	runs only on root machine
----------------------------------	------------------------------

- Paralelizarea a fost facuta la nivelul camp_dipolar
- Ar fi putut fi facuta la nivel de camp_eff cu un minim de afectare al rezultatelor, insa am preferat abordarea “sigura”
- Paralelizarea face numai o operatie de allreduce pentru a consuma cat mai putin timp in procesul de comunicare
- Notatiile s-au pastrat pe cat posibil pentru a nu afecta sensul fizic al relatiilor

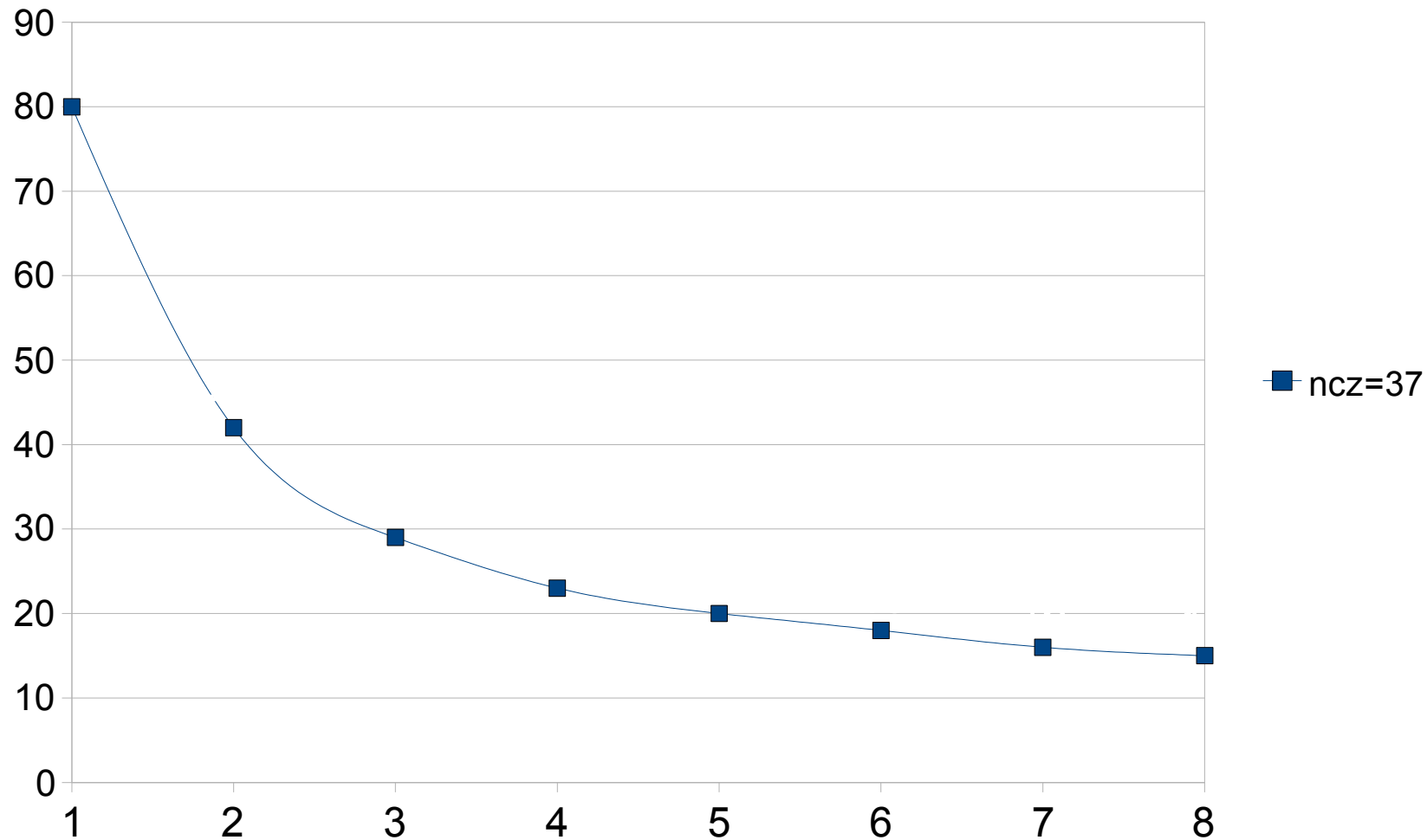
Continut

- Prezentare Problema
- Prezentare problema 1: Simulare Nanofir Singur
 - Teorie
 - Optimizari realizate
 - **Rezultate**
- Prezentare problema 2: Simulare Suprafata Nanofire
 - Teorie
 - Optimizari realizate
 - Rezultate
- Concluzii

Etapele optimizarii



Performanta: comunicare prin memorie partajata (rulare pe pe-quad)



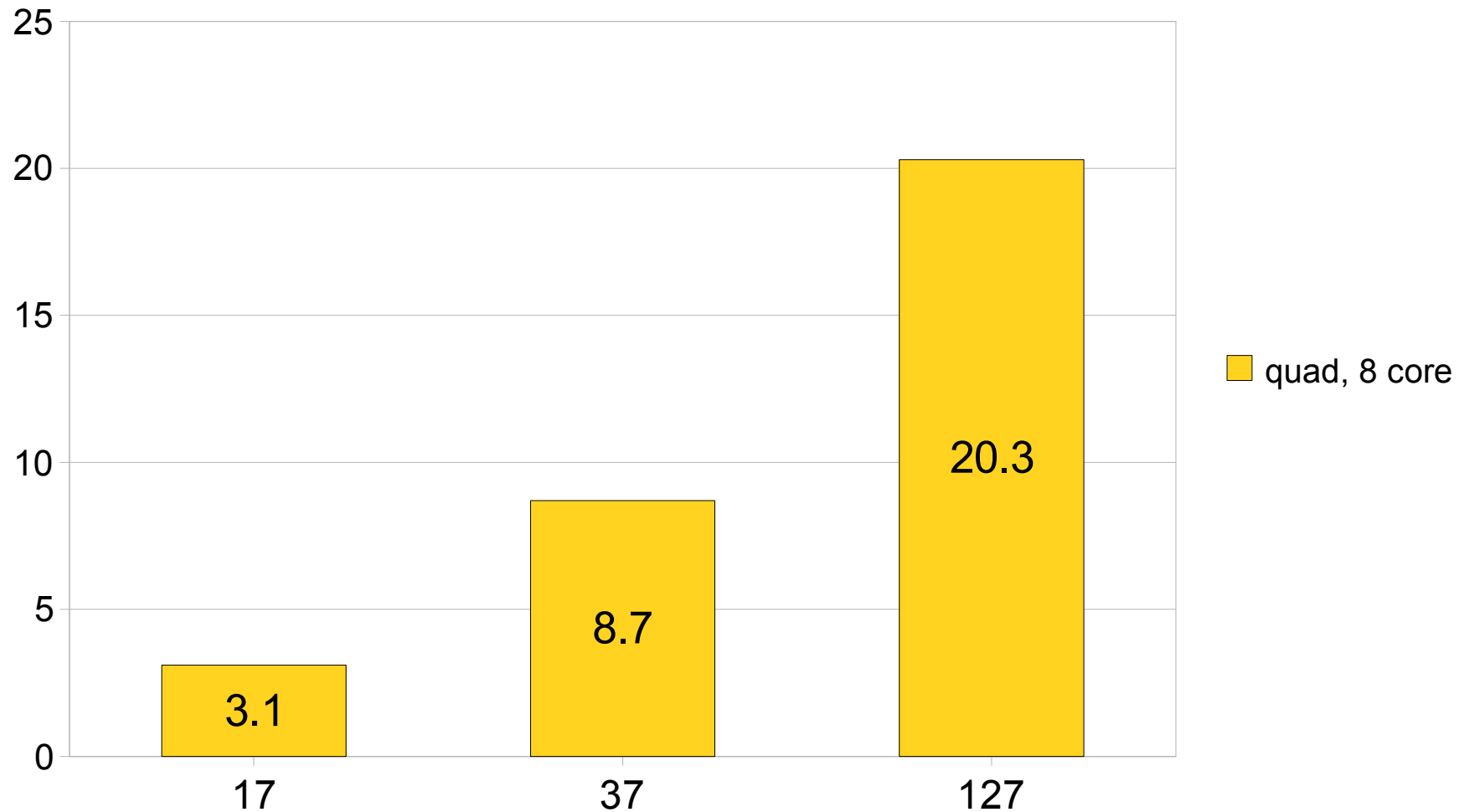
Performanta: comunicare prin mesaje (pe-dual/pe-p4)

?

ALLREDUCE

este toata comunicatia care se face intr-o iteratie

Performanta: dependenta de dimensiunea problemei



Performanta maxima atinsa

- Dimensiunea datelor: $ncz=37$
- Rulat pe fep
- programul initial \rightarrow 36 min 5sec \Rightarrow 2165 secunde
- programul optimizat, 1 proces \rightarrow 3 secunde

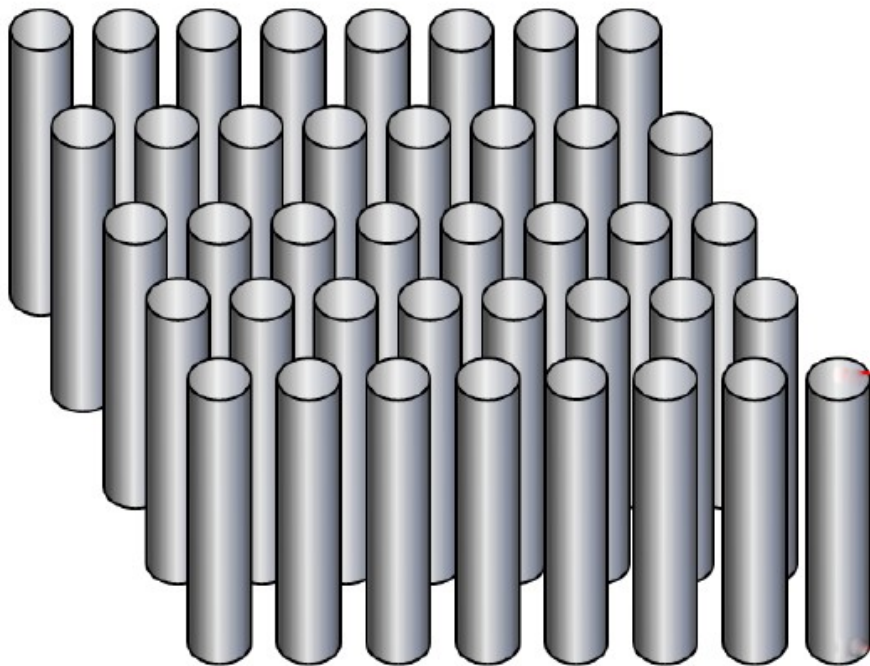
\rightarrow Speedup 721 x

Continut

- Prezentare Problema
- Prezentare problema 1: Simulare Nanofir Singur
 - Teorie
 - Optimizari realizate
 - Rezultate
- **Prezentare problema 2: Simulare Suprafata Nanofire**
 - **Teorie**
 - Optimizari realizate
 - Rezultate
- Concluzii

Simularea unei suprafete cu nanofire:

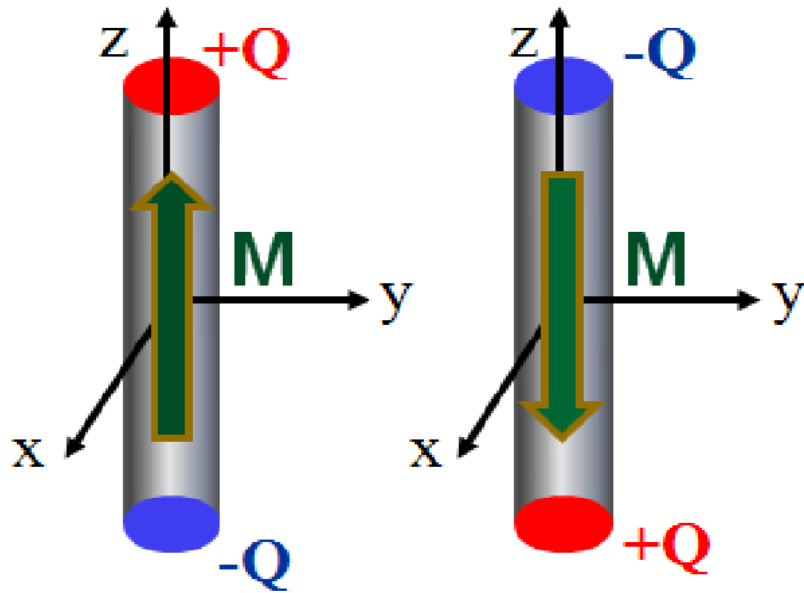
Teorie - 1



- In majoritatea cazurilor se intalnesc suprafete cu nanofire si nu nanofire singulare
- Din cauza proprietatilor de histerezis ale nanofirelor se va putea observa si un histerezis total al intregii suprafete
- Insa nanofirele interactioneaza intre ele
- Si mai exista si temperatura si camp exterior

Simularea unei suprafete cu nanofire:

Teorie - 2

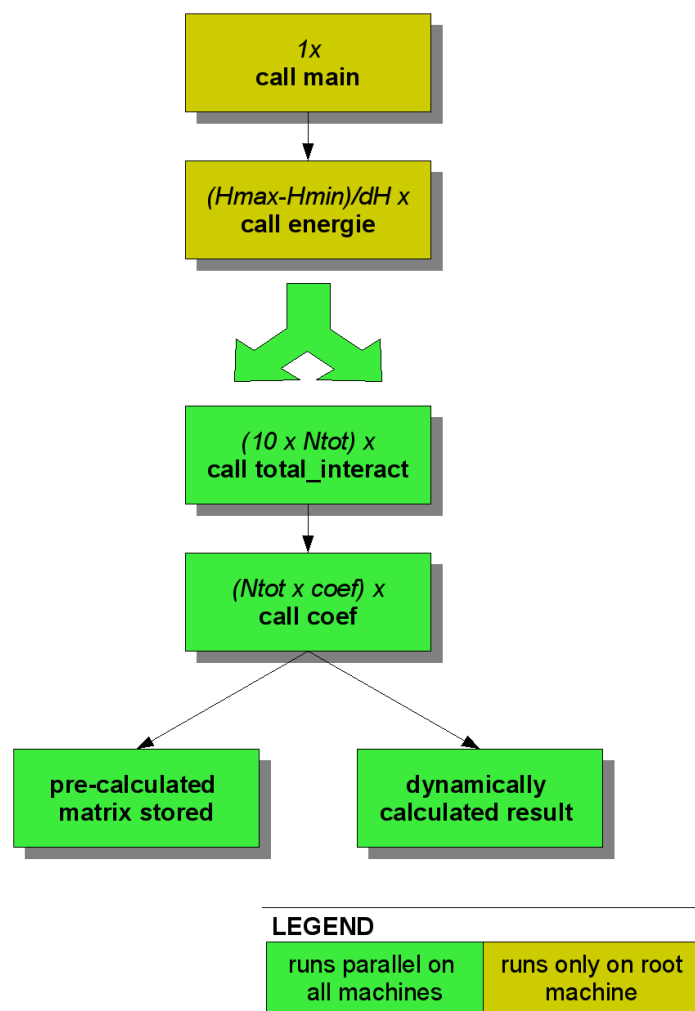


- Pentru simplificarea problemei se va considera ca magnetizarea unui nanofir poate fi doar in sus sau in jos

Continut

- Prezentare Problema
- Prezentare problema 1: Simulare Nanofir Singur
 - Teorie
 - Optimizari realizate
 - Rezultate
- **Prezentare problema 2: Simulare Suprafata Nanofire**
 - Teorie
 - **Optimizari realizate**
 - Rezultate
- Concluzii

Optimizare seriala - 1

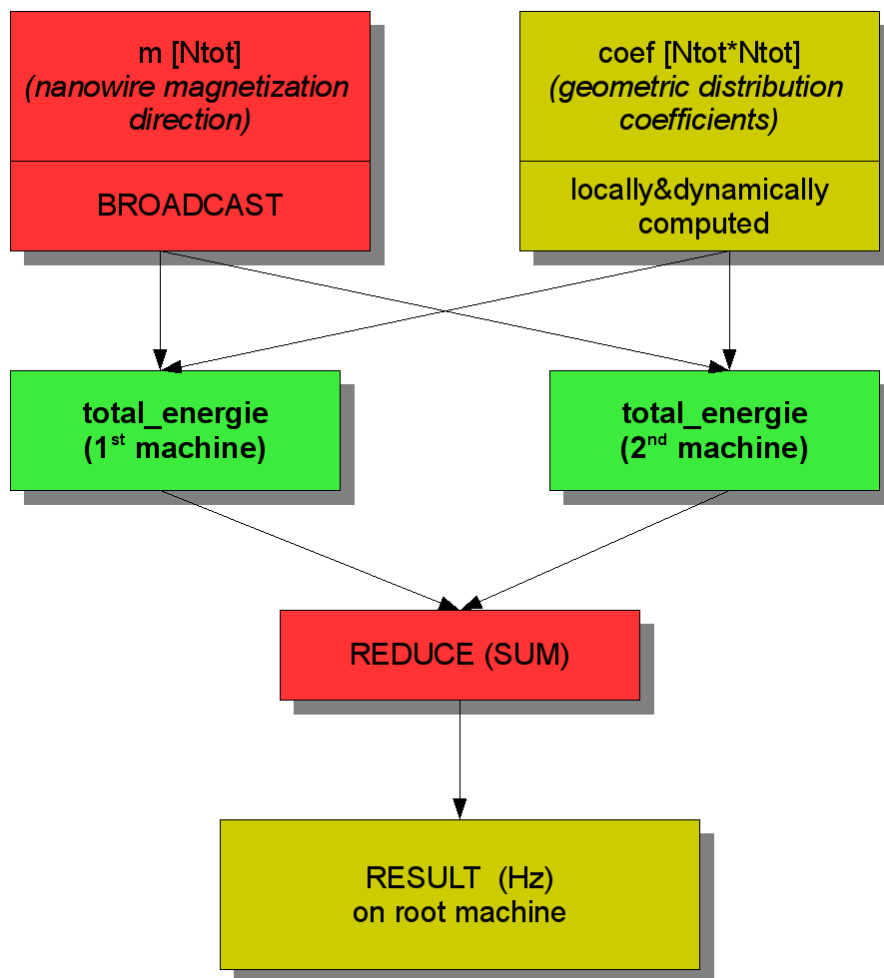


- In cadrul simulării am observat ca se facea un numar foarte mare de apelari pentru procedura **total_interact**
- Aceasta procedura realizeaza calcularea tuturor interactiilor ce au loc asupra unui nanofir pe baza unei matrice de **coeficienti geometrici**
- Coeficientii geometrici pot fi precalculati sau calculati dinamici
- Pentru precalcularea coeficientilor am optimizat modul de adresare al acestui tablou **patru-dimensional** (cache-lines)

Optimizare seriala - 2

- Calculul dinamic al coeficientilor geometrici
 - este util atunci cand tabloul patru-dimensional de coeficienti devine foarte mare ($n_i=1000 \Rightarrow 4 \text{ TB}$)
 - este mai lent pentru ca se re-calculeaza aceleasi valori
- Calculul static
 - pentru probleme de dimensiune mica
 - duce la o viteza mai mare
- Calculul static/dinamic se poate aplica la mai multe nivele
 - **coeficientii geometrici** ($\sim n^4$) se bazeaza pe **coeficientii de retea** ($\sim n^2$) \rightarrow oricare se poate calcula dinamic sau static, in functie de dimensiunea problemei

Optimizare paralela

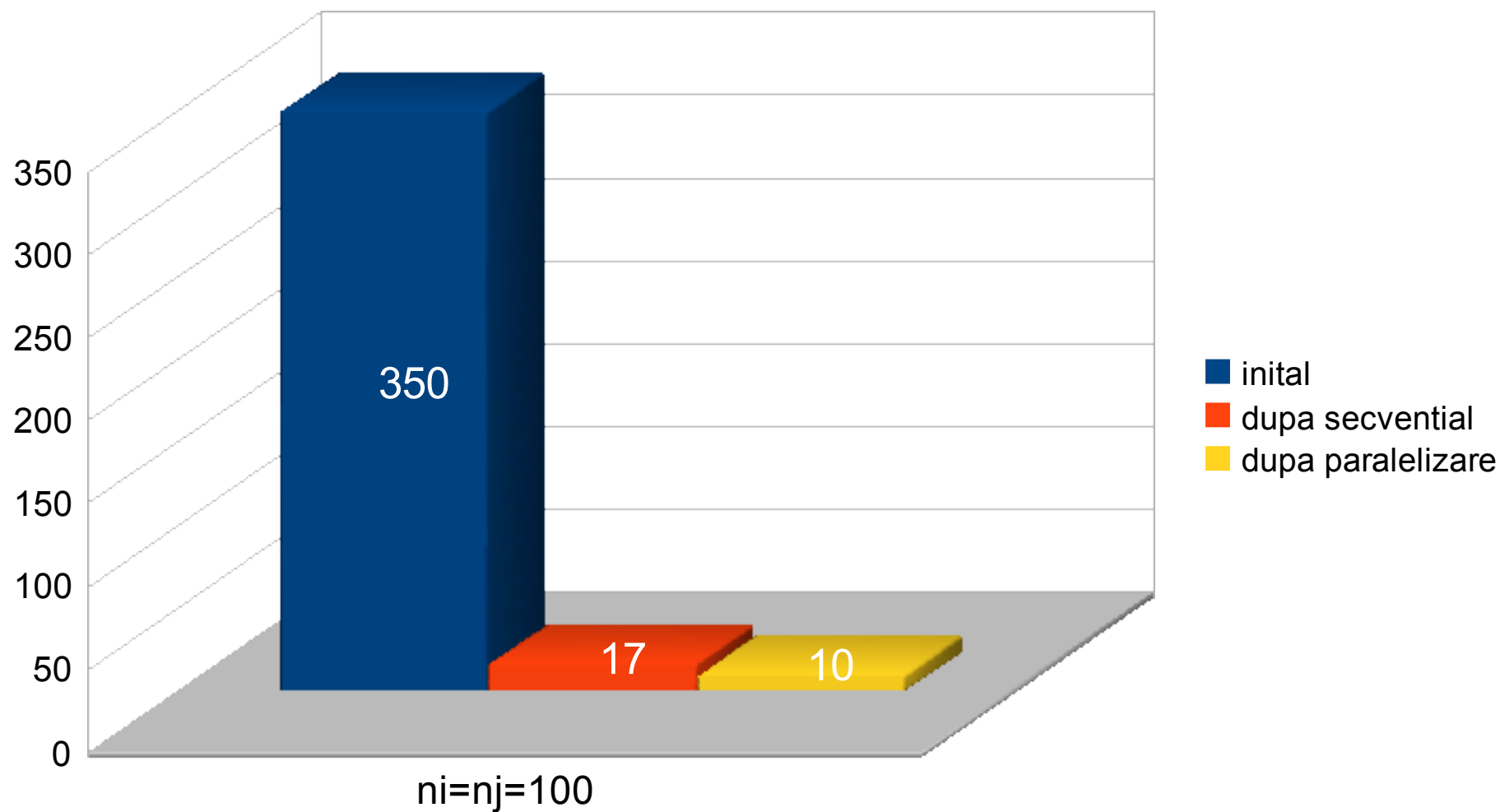


- **MPI_ALLREDUCE()** - singura comunicatie din program

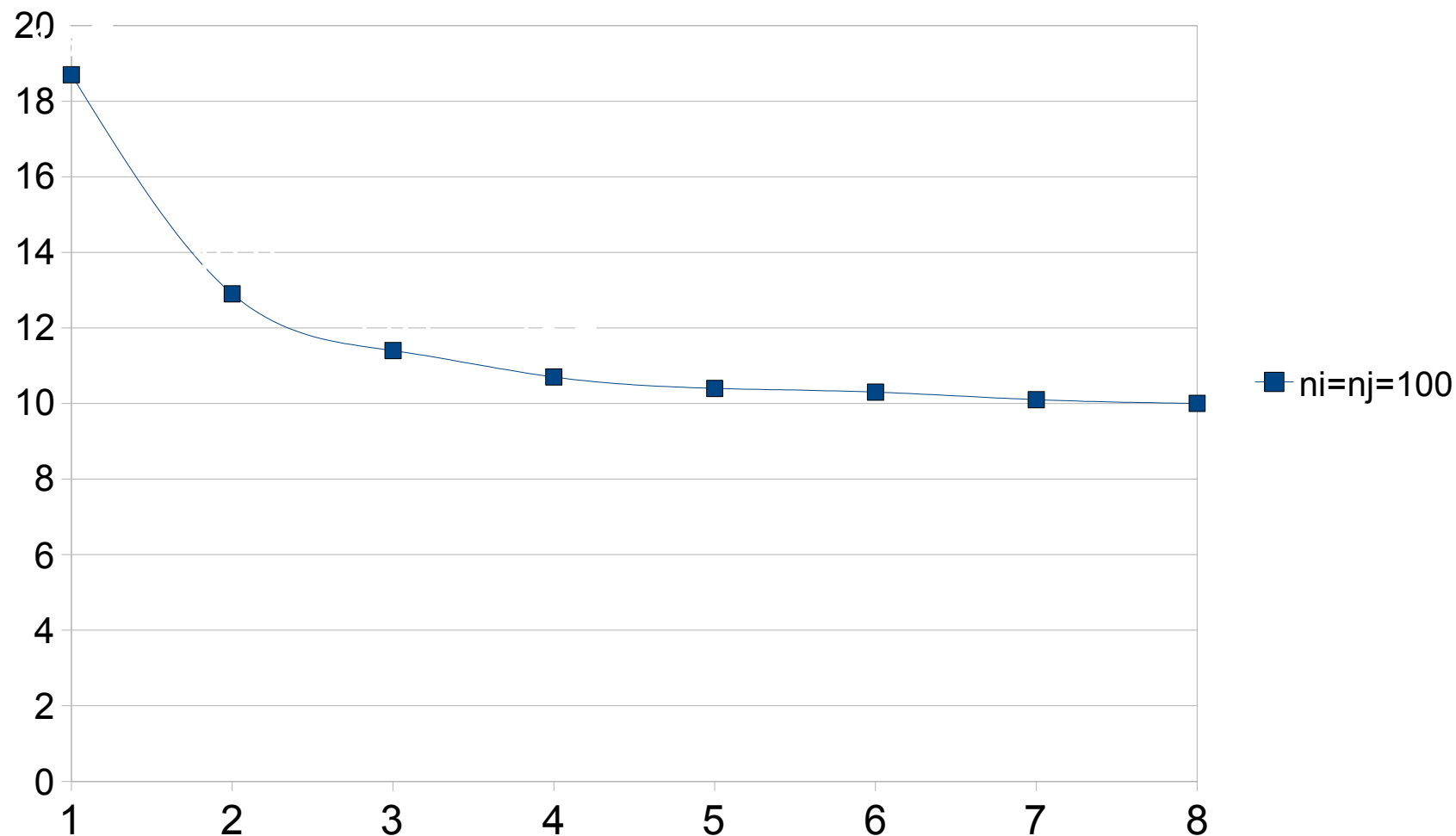
Continut

- Prezentare Problema
- Prezentare problema 1: Simulare Nanofir Singur
 - Teorie
 - Optimizari realizate
 - Rezultate
- **Prezentare problema 2: Simulare Suprafata Nanofire**
 - Teorie
 - Optimizari realizate
 - **Rezultate**
- Concluzii

Etapele optimizarii



Performanta: comunicare prin memorie partajata (rulare pe pe-quad)



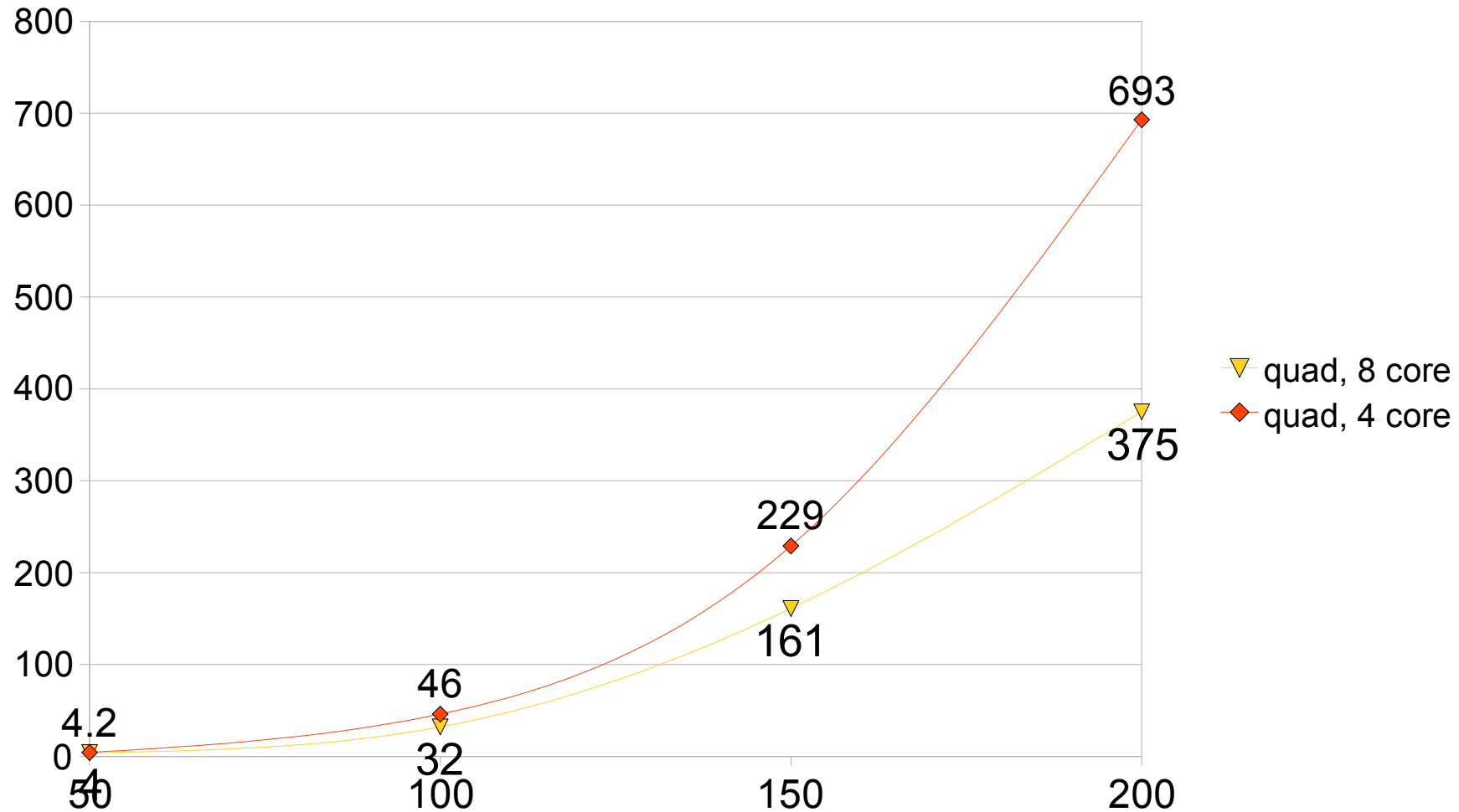
Performanta: comunicare prin mesaje (pe-dual/pe-p4)

?

ALLREDUCE

este toata comunicatia care se face intr-o iteratie

Performanta: dependenta de dimensiunea problemei



Performanta maxima atinsa

- Dimensiunea datelor: $n_i=n_j=100$ (400MB RAM)
- Rulat pe quad
- programul initial \rightarrow 350 secunde
- programul optimizat, 1 proces \rightarrow 10 secunde

\rightarrow Speedup 35 x

Continut

- Prezentare Problema
- Prezentare problema 1: Simulare Nanofir Singur
 - Teorie
 - Optimizari realizate
 - Rezultate
- Prezentare problema 2: Simulare Suprafata Nanofire
 - Teorie
 - Optimizari realizate
 - Rezultate
- **Concluzii**

Concluzii – politici de lucru - 1

- Corectitudinea rezultatelor primeaza
 - Testare continua
 - Orice optimizare care perturba rezultatele “de referinta” nu este inclusa
- Nu se lasa nimic neatins
 - Orice poate fi optimizat se optimizeaza → multe optimizari care nu au intrat in variantele finale. Exemplu: generator de numere pseudoaleatoare Mersene Twister
 - Cache lines optimization
- Respecta sensul fizic
 - Pastreaza nume si ordine de apelare pe cat posibil

Concluzii – politici de lucru - 2

- Documenteaza tot ce faci
 - Wiki, svn, mail, messenger, skype
- Sedinte de consultare pe net intre membrii echipei pentru decizii importante
- Peer code review – revizuirea codului scris/modificat de catre ceilalti
- Lucru constructiv
 - Bug-uri identificate si rezolvate (vezi wiki)
 - Indentare cod
- Scriptat tot ce se putea scripta

Concluzii – bad things

Ne pare rau ca nu am ajuns la laboratoare si
la prezentari

Andrei e plecat in America
Mihai lucreaza
Vlad face practica



**Computer Science
& Engineering
Department**

Intrebari?



Va multumim!