

Paralelno programiranje

Predavanje 1

Miloš Ivanović

Institut za matematiku i informatiku
PMF Kragujevac

školska 2013/2014. godina

O čemu će biti reči?

- 1 Motivacija
- 2 Istorija
- 3 Superkompjuteri i paralelni sistemi
- 4 Paralelno programiranje
 - Strategije paralelnog programiranja
- 5 Zaključak

O predmetu

- Nastavnik: Miloš Ivanović, asistent: Đorđe Nedić
- 46 kolokvijumi + 4 prisustvo + 50 usmeni ispit = 100 bodova
- Osnovna literatura: Michael Quinn: Parallel Programming in C with MPI and OpenMP
- Evidentan nedostatak literature na srpskom jeziku

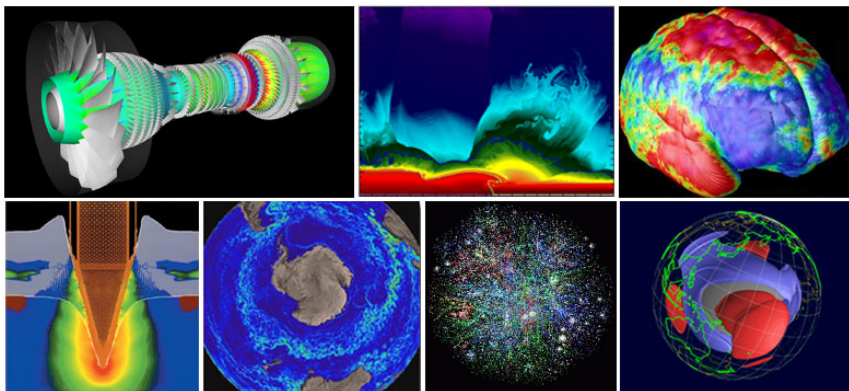
Šta je to paralelno i distribuirano računarstvo?

- Brže rešavanje problema korišćenjem većeg broja CPU-ova,
 - elementarni primer množenje matrica $C = A \times B$
- **Paralelno** = u užem smislu deljena memorija između svih CPU-ova
- **Distribuirano** = svaki procesor ima svoju lokalnu memoriju
- Uobičajeni problemi: *particionisanje, sinhronizacija, zavisnosti, balansiranje opterećenja*

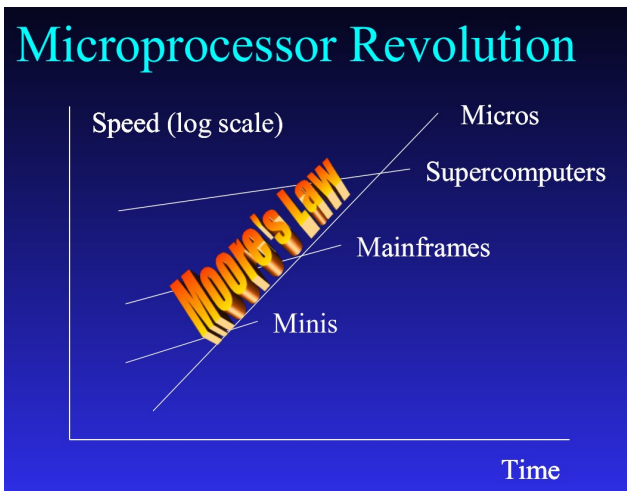
Paralelno i distribuirano računarstvo-motivacija

- Takozvani **Grand Challenge Problems**
 - Prognoza vremena i klime
 - Globalno zagrevanje
 - Dizajn materijala superprovodnici na sobnoj temperaturi; nano-uređaji, kosimčke tehnologije
 - Modeliranje organa, farmaceutska industrija, ...
- **Fizička ograničenja** u elektronici
 - Uskoro neće biti moguće dalje ubrzavanje pojedinačnih CPU-ova
 - Ograničena brzina transmisije u bakru
 - Ograničen stepen minijaturizacije - **NEMA REŠENJA!**

Kompjuterske simulacije-primena



Mikroprocesorska revolucija



Superkompjuter nekad i sad

- **Tradicionalni superkompjuter** - jedan vektorski procesor povezan na memorijsku magistralu visokih performansi
- **Današnji superkompjuter** - klaster koji se sastoji od čvorova sa po više procesora na svakom
- **VLSI** – Efekat integracije
 - 1 M tranzistora dovoljno za potpunu funkcionalnost - *Dec Alpha* (RISC, devedesete)
 - Danas se ide na više jezgara po čipu
- **Cena** – Veliki broj prosečnih CPU-ova daje bolji odnos FLOPS/\$ u poređenju sa tradicionalnim superkompjuterima

Moderni paralelni računari

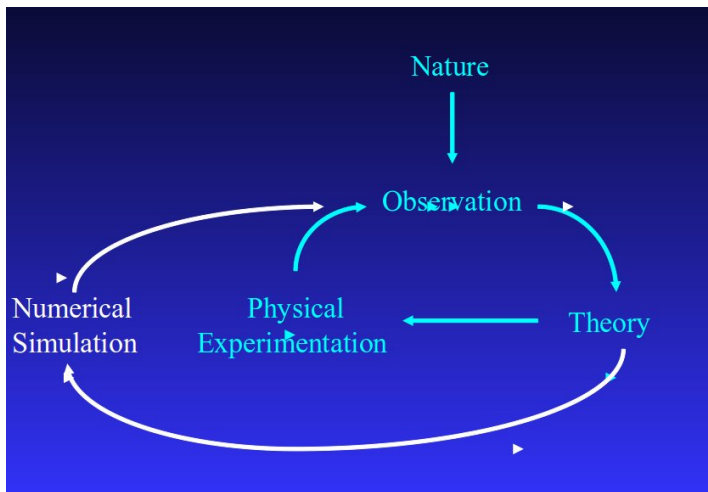
- **Caltech Cosmic Cube** (Seitz and Fox) 1981. godine, 64 Intel 8086/87 CPUs, 128kB memorije po CPU
- **Komercijalne kopije**
 - nCUBE Corporation (512 CPUs)
 - Intes Supercomputer Systems iPSC1, iPSC2, Intel Paragon (512 CPUs)
 - i mnogi drugi
- **Thinking Machines Corporation**
 - CM2 (65K 4-bit CPUs) – 12-dimensional hypercube - SIMD
 - CM5 – fat-tree interconnect - MIMD
- **Roadrunner** - Los Alamos NL 116,640 cores 12K IBM cell

Čemu služi paralelno i distribuirano računarstvo?

Opšti motivi

- Poželjno je iskoristiti kapacitet umreženih radnih stanica, *Grid i Cloud resursa*
- Rešiti računski intenzivne probleme brže
 - Nerešivi problemi postaju rešivi upotrebom paralelnog računara
 - Iskoristiti prednosti velike količine distribuirane memorije
- Rešavati veće probleme u istom vremenskom intervalu
 - Unaprediti **preciznost** odgovora
 - Redukovati **vreme projektovanja**
- Iskoristiti sveprisutne višejezgarne procesore i GPU čipove

Moderni naučni metod



Evolucija superkompjutinga

- **II svetski rat**
 - Artiljerijske tabele se računaju manuelno
 - Osnovni razlog za nastanak računara je ubrzavanje računa
 - ENIAC
- **Hladni rat**
 - Projektovanje nuklearnog naoružanja
 - Obaveštajne aktivnosti
 - Razbijanje šifri

Strateške inicijative

- Menja se nuklearna politika SAD-a
 - Moratorijum na nuklearno testiranje
 - Proizvodnja novog naoružanja je zaustavljena
- Sada su potrebne **neričke simulacije da bi se održavao postojeći arsenal naoružanja**
- U tom cilju je postavljeno pet superkompjutera, svaki vredan oko 100 miliona \$

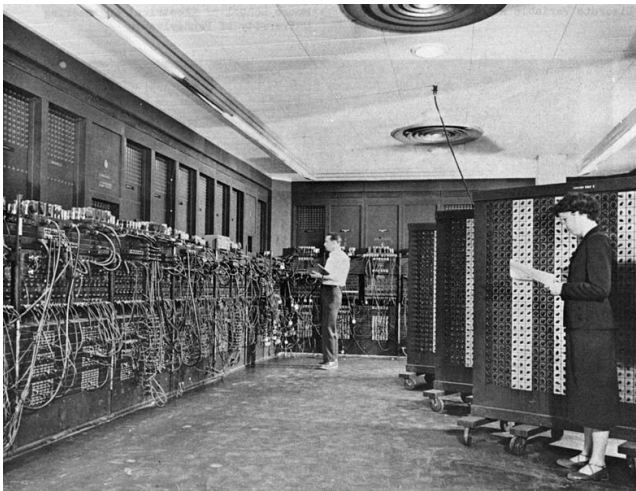
Tianhe-2 (33.862 petaFlop/s)



- Mega flops = 10^6 flops = 2^{20}
- Giga = 10^9 = billion = 2^{30}
- Tera = 10^{12} = trillion = 2^{40}
- Peta = 10^{15} = quadrillion = 2^{50}
- Exa = 10^{18} = quintillion = 2^{60}
- www.top500.org - sve o 500 najvećih superkompjuteru na svetu

ENIAC (350 op/s)

1946. godina



Superkompjuter

- Najbrži računari opšte namene
- Rešavaju ogromne probleme velikim brzinama u poređenju sa standardnim računarima
- Tipično koštaju preko 10 miliona \$
- Obično se mogu naći u državnim laboratorijama (naravno, na Zapadu)

Komercijalni superkompjuting

- Primena započeta u industrijama sa velikim kapitalom
 - Istraživanje naftnih izvora
 - Automobilska industrija
- Osali vidovi privrede prate u stopu
 - Farmaceutska industrija
 - Finansijske transakcije

60 godina ubrzavanja



Procesori su milionima puta brži

- Veće frekvencije
- Povećana konkurentnost
 - Veći broj funkcionalnih jedinica
 - Konkurentno izvršenje instrukcija
 - Spekulativno izvršavanje instrukcija
- Sistemi su sada brži trilion puta
 - Sami procesori se ubrzavaju
 - Kombinuju se hiljade procesora u superračunaru

Poređenje paralelnog sistema i tradicionalnog superkompjutera

- Mikroprocesor
 - 1% brzine superkompjutera
 - 0.1% cene superkompjutera
- Paralelni računar = 1000 mikroprocesora
 - 10 x brzina tradicionalnog superkompjutera
 - Ista cena kao superkompjuter

Zašto su superkompjuteri nedostupni?

- *Superkompjuter* $\neq \sum$ CPUs
 - Computation rate \neq throughput (#jobs/time)
 - Spora interkonekcija
 - Neadekvatni I/O
 - Posebno prilagođene komponente istovremeno povlače inerciju u prihvatanju novih čipova sa najpovoljnijim odnosom *cena/performance*
 - Fokus na high-end tržište povlači nedostatak redukcije cena
- Softver
 - Neadekvatni operativni sistemi
 - Neadekvatna programska okruženja

Neki poznati superkompjuteri

- IBM – SP1 and SP2, Blue Gene L/P
- Hewlett-Packard – Tandem
- Silicon Graphics (Cray) – Origin
- Sun Microsystems - Starfire

Komercijalni paralelni sistemi

- Visoka cena po procesoru
- Primitivna programerska okruženja
- Fokus na komercijalni domen
- Naučnici su potražili alternativu

Koncept *Beowulf* klastera

- NASA (Sterling and Becker)
- Široko dostupni procesori
- Standardna interkonekcija
- GNU/Linux OS
- MPI/PVM biblioteke
- Visoke performanse/\$ za *određene* aplikacije

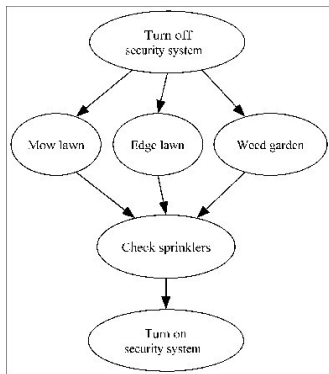


Traženje konkurentnosti

- Grafovi zavisnosti
- Paralelizam podataka
- Funkcionalni paralelizam
- Pajplajning

Graf zavisnosti

- Usmereni graf
- Čvorovi predstavljaju zadatke
- Veze predstavljaju zavisnosti



Paralelizam podataka

- Nezavisni taskovi primenjuju istu operaciju na različitim elementima skupa podataka
- U redu je da se ove operacije obave konkurentno
- Povećanje brzine: potencijalno p puta, gde je $p = \#procesora$

Primer

```
for i = 0 to 99 do  
a[i] = b[i] + c[i]  
endfor
```

Funkcionalni paralelizam

- Nezavisni taskovi primenjuju **različite funkcije** nad istim ili različitim elementima podataka
- **Brzina limitirana brojem konkurentnih podzadataka**
- U primeru prvi i drugi izraz
- Treći i četvrti izraz

Primer

$$a = 2$$

$$b = 3$$

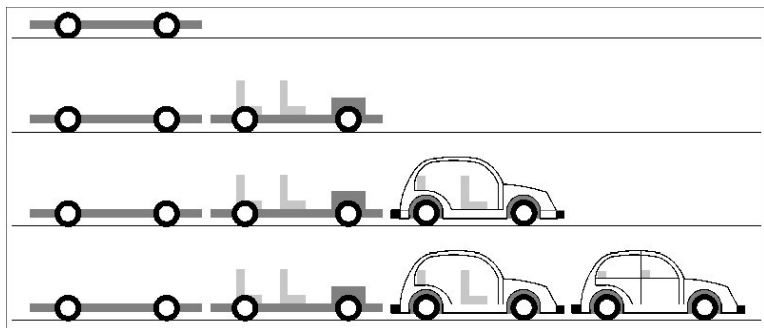
$$m = (a + b) / 2$$

$$s = (a^2 + b^2) / 2$$

$$v = s - m^2$$

Pajplajning

- Čitav proces podeliti na faze
- Proizvoditi nekoliko objekata istovremeno
- Ubrzanje: Limitirano brojem konkurentskih pod-taskova=
#broj faza pajplajna



Programiranje paralelnih računara

- **Proširenje kompajlera**: prevođenje sekvencijalnih programa u paralelne
- **Proširenje jezika**: dodavanje paralelnih operacija
- **Dodavanje paralelnog sloja** na sekvencijalni jezik
- **Definisanje potpuno novog paralelnog jezika** i kompajlerskog sistema

Strategija 1: Proširenje kompajlera

- Paralelizujući kompajler
- Detektovanje paralelizma u sekvencijalnim programima
- Produkcija paralelnog izvršnog fajla
- AKcenat na *Fortran* kompajlerima

Strategija 1: Proširenje kompajlera

Prednosti i mane

Prednosti

- Može se primeniti na milione linija već postojećeg koda
- Štedi vreme i napor
- Ne zahteva se dodatna obuka programera
- Sekvencijalno programiranje je lakše od paralelnog

Mane

- Paralelizam se može nepovratno izgubiti kada se program napiše sekvencijalno
- Ova tehnologija radi isključivo kod jednostavnih konstrukcija, kao što su petlje
- Za sada ne postoji nijedno *production-quality* rešenje ovog tipa

Strategija 2: Proširenje jezika

Dodavanje funkcija sekvencijalnom jeziku

- Kreiranje i terminiranje procesa
- Sinhronizacija procesa
- Omogućiti procesima da međusobno komuniciraju
- Primeri su mnogobrojni: *MPI, PVM, POSIX threads, OpenMP*

Strategija 2: Proširenje jezika

Prednosti i mane

Prednosti

- Najlakši, najbrži i najjeftiniji način
- Moguće je iskoristiti već postojeće kompajlere i biblioteke
- Nove paralelne biblioteke se razvijaju velikom brzinom

Mane

- Standardni kompajleri ne omogućavaju hvatanje grešaka i debugging specifičnog paralelnog koda
- Lako je pisati programe, ali ih je vrlo teško *debug*-ovati

Strategija 3: Dodavanje paralelnog programskog sloja

- **Donji sloj**
 - Kompjuciono jezgro
 - Proces manipuliše sopstvenom porcijom podataka da bi proizveo porciju rezultata (objektno)
- **Gornji sloj**
 - Kreiranje i sinhronizacija procesa
 - Particionisanje podataka među procesima
- Nekoliko istraživačkih prototipova je razvijeno – Linda, iC2Mpi platform (Prasad, et al. IPDPS-07 workshops), SyD Middleware – System on Devices (Prasad, et al., MW-04)

Strategija 4: Paralelni programski jezik

- Razviti paralelni jezik “*from scratch*”
 - Primer je jezik *Occam*
- Dodati paralelne konstrukcije u već postojeći jezik
 - *Fortran 90*, *Fortran 95*
 - *High Performance Fortran*
 - *C**

Strategija 4: Paralelni programski jezik

Prednosti i mane

- **Prednosti**

- Dozvoljava programeru da predoči paralelizam samom kompajleru
- Povećava se verovatnoća da će izvršni program dostići visoke performanse

- **Mane**

- Zahteva razvoj novih kompajlera
- Novi jezici možda neće postati standardizovani
- Otpor programera

Trenutni status

- **Pristup niskog nivoa je trenutno najpopularniji**
 - Upotreba postojećeg jezika sa *low-level* paralelnim konstrukcijama
 - MPI, PVM, pthreads bazirana konkurentnost i OpenMP su primeri
- **Prednosti pristupa niskog nivoa**
 - Efikasnost
 - Portabilnost
- **Mane**
 - Teže programiranje i *debug*

Zaključak

- Računarstvo visokih performansi (*High performance computing*)
 - US vlada
 - Industrije sa visokim kapitalom
 - Kompanije i istraživačke laboratorije
- Paralelni računari
 - Komercijalni sistemi
 - Sistemi bazirani na “standardnim” komponentama

Zaključak

Nastavak

- Snaga CPU-ova nastavlja da raste eksponencijalno
- Bliska budućnost verovatno leži u GPU i hibridnim tehnologijama (npr. *Intel Phi*)
- Okruženja za paralelno programiranje se veoma sporo menjaju
- Aktuelna su dva standarda
 - **MPI biblioteke** za procese koji ne dele memoriju
 - **threads/OpenMP** za procese koji dele memoriju