# Klastrové a Gridové Počítanie

Viera Šipková
Ústav informatiky SAV
Bratislava, Dúbravska cesta 9

# Obsah

- **Základné princípy klastrového a gridového počítania**
  - paralelizmus
  - programovacie modely

# Paralelizmus – hardvér

- **Moderné architektúry počítačov**
  - multi-jadrové procesory
    - multi-/hyper-threading, vector processing, superscalar, **...**
  - floating-point akcelerátory
  - grafické procesory (GPU)
  - komplexné hierarchie pamätí
  - HPC klastre, gridy, superpočítače

# Paralelizmus – softvér

- **Architektúra aplikácie**
  - vhodný programovací model schopný zamestnať čo najviac úrovní hardvérového paralelizmu
- **Paralelizmus**
  - lokálne optimalizácie na úrovni CPU: jazyky a kompilátory
  - globálne optimalizácie sa nevykonávajú automaticky
    - dekompozícia programu a dát, manažment rôznych typov pamätí, výber algoritmov, ...
  - nástroje a vývojové prostredia: mali by poskytovať možnosť automatického návrhu paralelnej aplikácie na rôznych úrovniach abstrakcie, uplatnením rôznych stratégií na distribúciu kódu a údajov a použitím paralelných objektov rôznej granularity

# Paralelné programovanie

- **Prechod od sekvenčného programovacieho modelu k paralelnému** ⇒ prehodnotenie predstavy toku procesu
  - detekovanie činností (funkcií, komponentov), ktoré môžu byť vykonávané paralelne
  - zostavenie aplikácie ako štruktúrovanej množiny úloh spolu s definovaním závislostí medzi jednotlivými úlohami
  - v rámci úlohy
    - rozdelenie dát
    - vytvorenie súbežných vlákien/procesov a definovanie koordinácie medzi nimi a ich operáciami
    - využitie paralelných algoritmov a knižníc

# Dekompozícia aplikácie

- **Stratégie dekompozície**
  - dekompozícia podľa funkcií
    - vlákna vykonávajú rozdielne činnosti
  - dekompozícia podľa dát
    - vlákna vykonávajú tú istú činnosť nad rôznymi dátami
  - dekompozícia podľa toku dát
    - výstup jedného vlákna slúži ako vstup pre iné vlákno
  - voľba stratégie závisí od vlastností konkrétnej aplikácie
    - granularita, dominantnosť (výpočty, práca s údajmi, kolaboratívne činnosti)
    - každá vyžaduje manažment simultánnych procesov a ich interakcií (komunikácie, synchronizáciu, vyváženie vyťaženia, škálovateľnosť)

# Granularita

- **Hrubo-zrnné paralelné výpočty**
  High throughput computing
  - pri vykonávaní úlohy každá z jej pod-úloh je relatívne nezávislá na výsledku iných pod-úloh (tj. oneskorenie pri obdržaní výsledku z jedného procesora nemá významný vplyv na činnosť iných procesorov)
    - voľne viazané siete výpočtových prostriedkov (napr. grid)
- **Jemno-zrnné paralelné výpočty**
  High performance computing
  - pri vykonávaní úlohy každá z jej pod-úloh je silne závislá na výsledku iných pod-úloh
    - klastre, superpočítače, tesne viazané klastre s veľkým počtom procesorov a rýchlou komunikačnou sieťou

# Architektúry počítačov

- **Základný klasifikačný model** (M.J. Flynn, 1966)
  - SISD (Single Instruction, Single Data)
  - SIMD (Single Instruction, Multiple Data)
  - MISD (Multiple Instruction, Single Data)
  - MIMD (Multiple Instruction, Multiple Data)

- **MIMD architektúra**
  - SPMD (Single Program, Multiple Data) (F. Darema, 1984)
    - najrozšírenejší model paralelného programovania
  - MPMD (Multiple Program, Multiple Data)
    - master-worker, workflow, DAG
  - podľa vlastností pamäte (pre SPMD, MPMD)
    - systémy s distribuovanou pamäťou, zdieľanou pamäťou, distribuovanou a zdieľanou pamäťou

# Klastrové počítanie

- **Klaster s viac-jadrovými CPUs** – počítačový systém s distribuovanou a zdieľanou pamäťou
  - množina nezávislých viac-jadrových procesorov (uzlov), navzájom spojených prostredníctvom prepojovacej siete, ktorá umožňuje medzi-uzlovú komunikáciu
  - každý uzol má svoju vlastnú pamäť (Distributed Memory - DM)
  - všetky jadrá v rámci uzla zdieľajú jednu spoločnú pamäť (Shared Memory - SM)
- **Klastrové počítanie** ⇒ hybridný programovací model
  - uplatnenie technológií pre DM a technológií pre SM (prípadne aj GPU technológií)

# DM Systém

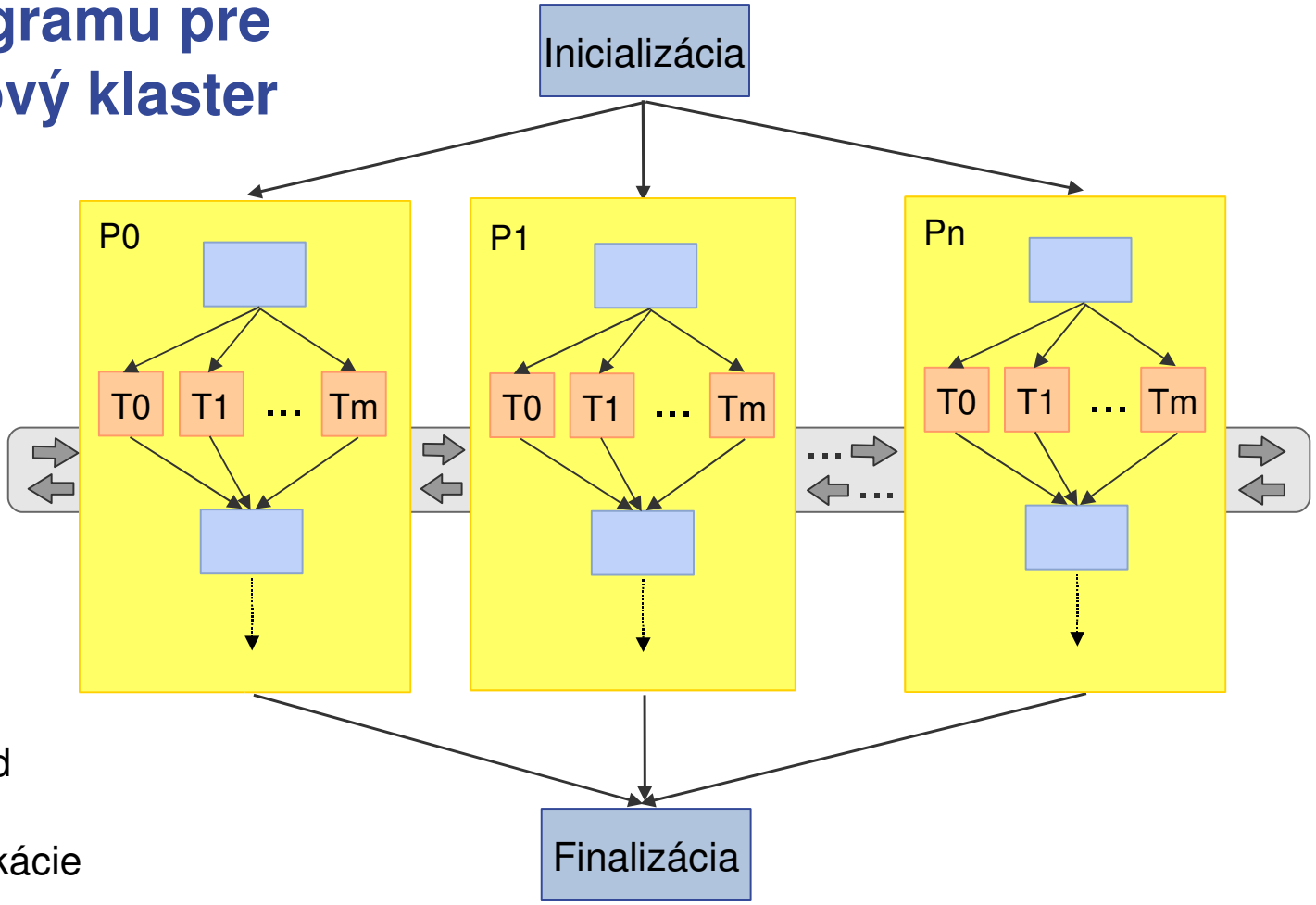
- **Systém s distribuovanou pamäťou**
  - každý procesor má svoj vlastný adresný priestor: výpočtové úlohy môžu operovať iba nad svojimi lokálnymi dátami
    - vzdialené dáta je nutné prenášať prostredníctvom komunikácií s inými procesormi cez prepojovaciu sieť (“message passing”)
    - kľúčovou otázkou je: ako distribuovať dáta aby sa predišlo častým komunikáciám medzi procesormi
  - MPI (Message Passing Interface) – špecifikácie MPI-1 a MPI-2 sa stali štandardom pre implementáciu komunikácií a kolektívnych operácií nad procesormi
    - existuje veľa implementácií (pre Fortran a C) pre rôzne platformy
    - väčšina implementácií alokuje pri inicializácii programu fixný počet MPI procesov (štandard: jeden proces na CPU/jadro)
    - point-to-point komunikácie, kolektívne operácie, “komunikátor”

# SM Systém

- **Systém so zdieľanou pamäťou**
  - viac-jadrový procesor umožňuje zdieľanie jedného spoločného adresného priestoru pre všetky jadrá
    - nie je potrebné explicitne špecifikovať dátové komunikácie
    - rôzne jadrá sa môžu navzájom rušiť pri zápise na rovnaké miesto pamäti – je nutné použiť synchronizačné mechanismy (semafory, bariéry)
    - pochopenie konzistencie a manažment “lokálnosti” dát je zložitejší problém
  - OpenMP (Open Multiprocessing) – dominantný programovací model na implementáciu “multi-threading” konceptu pre SM
    - API pre paralelné programovanie v jazykoch C/C++ a Fortran
    - explicitný “Fork-Join” model pre vykonávanie programu
    - hlavné komponenty: environment variables, compiler directives, library routines

# Klastrové počítanie

- **Model programu pre multi-jadrový klaster**




Pi – MPI proces
Tj – OpenMP thread
→ tok riadenia
⇨ dátové komunikácie

# Gridové počítanie

- **Grid** – distribuovaný paralelný systém
  - umožňuje zdieľanie, výber a zoskupenie geograficky distribuovaných autonómnych prostriedkov (výpočtových, úložných, softvérových, a iných) dynamickým spôsobom, v závislosti od ich dostupnosti, vybavenia, výkonnosti, ceny a používateľských požiadaviek na kvalitu služieb
  - súčasné výpočtové prostriedky: výkonné počítačové klastre

- **Gridové počítanie** ⇒ hybridný programovací model
  - uplatnenie technológií pre klastrové počítanie a technológií webových a gridových služieb

# Linky

- **MPI** http://www.mcs.anl.gov/research/projects/mpi
- **OpenMP** http://openmp.org/wp
- **PBS** http://www.pbsworks.com
- **gLite** http://glite.web.cern.ch/glite
- **EGEE** http://www.eu-egee.org
- **EGI** http://www.egi.eu
- **SlovakGrid** http://www.slovakgrid.sk

# Ďakujem za pozornosť !