

# IB109 Návrh a implementace paralelních systémů

## Organizace kurzu a úvod

Jiří Barnat

# Organizace kurzu

## Místo a čas

- úterý 14:00-15:40, B410

## Ukončení předmětu

- Závěrečný písemný test na odpřednášený obsah
- Možno získat několik bodů za nepovinné domácí úlohy
- Požadavky na úspěšné ukončení předmětu

Z: bodové hodnocení testu nad 50%

ZK: bodové hodnocení testu nad  
50%(E), 60%(D), 70%(C), 80%(B), 90%(A)

## **Cílem předmětu je seznámit studenty s**

- Problematikou programování paralelních aplikací,
- Programátorskými prostředky pro vývoj paralelních aplikací,
- Možnostmi studia tématu na FI.

## **Úspěšný absolvent kurzu**

- Umí identifikovat paralelně proveditelné úlohy.
- Má základní přehled o problémech souvisejících s paralelizací.
- Nebojí se implementovat vlastní vícevláknové nebo jinak paralelní aplikace či systémy.
- Má představu o tom, co se děje v zákulisí použitých knihoven pro podporu programování paralelních aplikací.
- Umí tyto knihovny správně použít.

## Osnova:

- Úvod a motivace.
- Paralelní algoritmy v prostředí se sdílenou pamětí. OpenMP, Intel TBB, POSIX Threads.
- Základní metody v návrhu paralelních algoritmů.
- Paralelní algoritmy v prostředí s distribuovanou pamětí. Message Passing Interface (MPI).
- Výkonnostní analýza paralelních algoritmů.
- Příklady paralelních řešení různých problémů.

## IB109

- Úvodní kurz určený pro bakalářské studium.
- Povinný v rámci oboru *Paralelní a distribuované systémy*

## Předpoklady

- Základní znalosti o fungování výpočetních prostředků a operačních systémů.
- Základní zkušenost s imperativním programováním sekvenčních algoritmů.

## Předměty

- IA039 – Architektura superpočítačů a intenzivní výpočty
  - Procesory. Paralelní počítače. Překladače. MPI. PVM a koordinační jazyky. Profilování a měření výkonu.
- PV192 – Paralelní algoritmy
  - Paralelní zpracování, Klasifikace paralelních systémů, Úrovně paralelismu, Paralelní počítače, Systémy s distribuovanou pamětí, MPI
- PV197 – GPU Programming
  - Paralelní výpočty na grafických kartách s technologií CUDA.
- PA150 – Principy operačních systémů
  - Vlákna, procesy, monitory, semaforey, synchronizace.
  - Hierarchie pamětí.

## Předměty

- IV100 – Paralelní a distribuované výpočty
  - Distribuované systémy a algoritmy. Komunikační protokoly. Směrovací algoritmy a tabulky. Distribuované algoritmy pro detekci ukončení, volbu vůdce, vzájemné vyloučení, hledání nejkratší cesty. Byzantská shoda.
- IV010 – Komunikace a paralelismus
  - Teoretický model paralelních procesů a komunikace. CCS. Synchronizace, vnitřní akce. Ekvivalence systémů pomocí slabé/silné bisimulace a relace kongruence.
- IV113 – Úvod do validace a verifikace
  - Model checking, verifikace paralelních systémů.

## Laboratoře

- PV177 Laboratoř pokročilých síťových technologií (vývojová)
- IV074 Laboratoř paralelních a distribuovaných systémů (výzkumná)
- ?????? Laboratoř architektury a konstrukce číslicových počítačů

## Projekty

- IV112 – Projekt z programování paralelních aplikací
- PV197 – GPU Programming

## Knihy

- Maurice Herlihy, Nir Shavit: *The Art of Multiprocessor Programming*
- A. Grama, A. Gupta, G. Karypis, V. Kumar: *Introduction to Parallel Computing*
- I. Foster: *Designing and Building Parallel Programs*
- W. Group, E. Lusk, A. Skjellum: *Using MPI*
- ...

## E-zdroje:

- <http://www.wikipedia.org>
- Kurzy a jejich studijní materiály na různých univerzitách
  - <http://www.cs.arizona.edu/people/greg/mpdbook>  
(Foundations of Multithreaded, Parallel, and Distributed Programming)
  - <http://renoir.csc.ncsu.edu/CSC495A>
- [http://www.hlrs.de/organization/par/par\\_prog\\_ws](http://www.hlrs.de/organization/par/par_prog_ws)  
Parallel Programming Workshop (MPI, OpenMP)
- Domovské stránky projektů MPI, TBB, OpenMP, ...
- Online tutoriály
- ...

# Motivace

## Souběžnost

- Existence dvou a více procesů (v obecném smyslu slova) v jeden časový okamžik.
- Všudypřítomný a nevyhnutelný jev.

## Prakticky je nevyhnutelný i v Computer Science

- Fyzikální důvody.
- Vyšší výkon (např. GPU)

## Souběžnost je teoreticky zajímavá

- Historicky možný kandidát na problém  $P=NP$ .
- Složitostní třída NC.
- Inherentně sekvenční problémy.

## **Proč by se o paralelismus měl zajímat řadový programátor?**

- Zvýšení výkonu aplikace na paralelních platformách.
- Nutnost agregace výpočetní síly pro dokončení výpočtu.
- Oddělení nesouvisejících částí aplikace.

## Abstraktní model výpočetního systému

Procesor - Datová cesta - Paměť

- Všechny části systému mohou být úzkým místem vůči výkonnosti aplikace jako celku.
- Paralelismus je přirozený způsob překonání úzkých míst.

## Procesory

- Neustálá potřeba zvyšovat výkon.
- Výkon procesoru spojován s Moorovým zákonem.

## Moorův zákon

- Gordon Moore, spoluzakladatel Intelu
- Počet tranzistorů v procesoru se zdvojnásobí přibližně každých 18 měsíců.

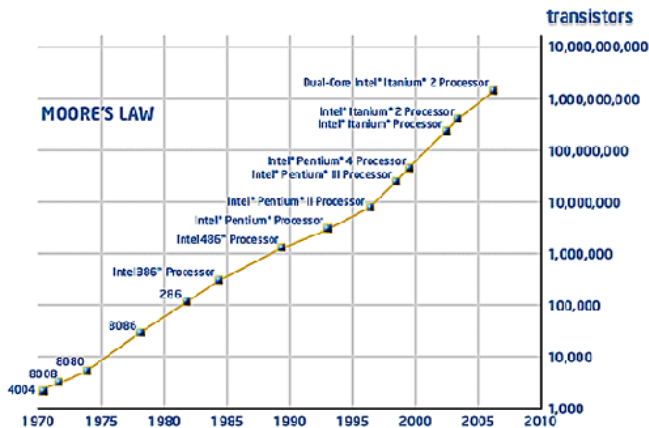
## Metody zvyšování výkonu procesorů

- Zvyšování frekvence vnitřních hodin.
- Multiplicita, Paralelismus

Procesor	Rok uvedení	Počet tranzistorů
4004	1971	2 250
8008	1972	2 500
8080	1974	5 000
8086	1978	29 000
286	1982	120 000
Intel386™	1985	275 000
Intel486™	1989	1 180 000
Intel® Pentium®	1993	3 100 000
Intel® Pentium® II	1997	7 500 000
Intel® Pentium® III	1999	24 000 000
Intel® Pentium® 4	2000	42 000 000
Intel® Itanium®	2002	220 000 000
Intel® Itanium® 2	2003	410 000 000

*Zdroj: Intel*

# Moore's Law



## Pozorování

- Výrobcům procesorů se nedaří zvyšovat výkon jednoho jádra.
- Fyzikální zákony brání neustálé miniaturizaci.
  - V současnosti 28nm technologie (předchozí technologie 32nm, 45nm, 65nm, 90nm).
  - Fyzikální limit je 5nm (nelze udržet elektrony v atomu).

## Řešení

- Vyrábí se vícejaderné procesory.
- Pravděpodobný způsob zvyšování výkonu i v budoucnosti.
- Částečný odklon od jednotek monolityckých jader k deítkám menších specializovaných.

## Související problém

- Sekvenční algoritmy nemohou nadále těžit z rostoucího výkonu procesorů.
- **Paralelizace software je nevyhnutelný směr vývoje.**

## Role paralelismu v komunikaci

- Větší propustnost komunikačních linek s následným efektem snižování latence.
- Alternativní/záložní komunikační linky.

## Příklady

- Urychlovače produkují data rychleji než je možné je doručit k výpočetním prostředkům a zpracovat je, dokonce rychleji, než je možné je sekvenčně uložit.
- P2P sítě řeší problém asymetrie down/up rychlostí a zátěže exponovaných uzlů.

## Fakta

- Výkon procesorů převyšuje výkon ostatních komponent.
- Cesta **procesor – paměť – disk** je zdlouhavá.
- Doba nutná pro získání jednotky informace z paměti roste se vzdáleností od procesoru.

## Víceúrovňové uložení informací

- Registry procesoru
- L1/L2/L3 cache
- Operační paměť
- Cache I/O zařízení
- Magnetické/optické mechaniky

## Cache paměť obecně:

- Kopie části dat v rychleji dostupném místě.
- Může a nemusí být kontrolovatelná uživatelem nebo OS.

## Příklady Cache s různou možností kontroly

- L1/L2 cache v rámci CPU nekontrolovatelná programátorem
- I/O efficient algoritmy
  - Obcházejí virtualizaci paměti kontrolovanou OS, a místo toho realizují vlastní způsob použití operační paměti jako cache pro data na disku.

## Multiplicita paměťových modulů

- Větší množství uložitelných/zapamatovatelných informací.
- Větší množství linek do paměti (větší propustnost).
- Větší režie na udržení konzistence.

## Příklady

- Násobné použití pevných disků
  - RAID 1, RAID 5 disková pole
  - SSD disky pro operační systém a swapovací prostor
- NUMA architektury
  - Pozorování: Čím větší blok paměti má být uniformě přístupován, tím pomalejší bude tento přístup.
  - Více procesorové počítače s více paměťovými moduly uspořádanými tak, že přístup jednoho procesoru do různých paměťových modulů je různě rychlý.

## Důvody:

- Nutnost specifikace souběžných úkolů a jejich koordinace.
- Paralelní algoritmy.
- Nedostačující vývojová prostředí.
- Nedeterminismus při simulaci paralelních aplikací.
- Absence reálného modelu paralelního počítače.
- Rychlý vývoj a zastarávání použitých technologií.
- Výkon aplikace náchylný na změny v konfiguraci systémů.
- ...

## Příklad

- V minulých letech byl doporučován pro hry 2-jádrový procesor, proč ne 4-jádrový, když byl zcela určitě výkonnější?

## Důvody:

- Nutnost specifikace souběžných úkolů a jejich koordinace.
- Paralelní algoritmy.
- Nedostačující vývojová prostředí.
- Nedeterminismus při simulaci paralelních aplikací.
- Absence reálného modelu paralelního počítače.
- Rychlý vývoj a zastarávání použitých technologií.
- Výkon aplikace náchylný na změny v konfiguraci systémů.
- ...

## Příklad

- V minulých letech byl doporučován pro hry 2-jádrový procesor, proč ne 4-jádrový, když byl zcela určitě výkonnější?
- Je obtížné napsat herní engine, který by fungoval dobře na 1-jádrovém stroji a na 4-jádrovém stroji běžel 4x rychleji.

## **HPC (High Performance Computing)**

- Oblast Computer Science
- Výpočty na vysoce paralelních platformách

## **Nejvýkonější počítač světa [Jaro 2010]**

- Roadrunner
- Procesory: PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz
- Počet jader: 129 600
- Rok výroby: 2008
- Výrobce: IBM
- GFLOPS: 1 105 000 (ve špičce 1 456 704)
  
- Více viz [www.top500.org](http://www.top500.org).

# Nejrychlejší počítač světa dle ([www.top500.org](http://www.top500.org))

