

# IB109 Návrh a implementace paralelních systémů

## Organizace kurzu a úvod

Jiří Barnat

# Organizace kurzu

## Místo a čas

- Středa 16:00-17:40, D3

## 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:

- Práce s daty ve sdílené paměti
- POSIX Threads
- Lock-Free programování
- OpenMP
- Generické programování a paralelismus (TBB)
- Předávání zpráv (v distribuované paměti)
- Kolektivní komunikace
- Složitostní analýza paralelních programů
- Algoritmizace paralelních aplikací

## 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.

## **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.

## **IA039 – Architektura superpočítačů a intenzivní výpočty**

- Procesory. Paralelní počítače. Překladače. MPI. PVM a koordinační jazyky. Profilování a měření výkonu.

## **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.

## Výpočetní paralelismus

- Napříč všemi úrovněmi, od implementace registru až po koexistenci rozsáhlých výpočetně distribuovaných aplikací.
- Chtěný pro zvýšení výkonu.
- Vykoupený složitostí návrhu a pořizovací cenou.

## Souběžnost v Computer Science

- Specifikace, implementace a analýza paralelních a distribuovaných systémů.
- Inherentně sekvenční algoritmy, složitostní třída  $NC$ .
- Historicky možný kandidát na problém  $P \stackrel{?}{=} NP$ .

## Výkonnost

- Umožní efektivně využít agregované výpočetní prostředky k zrychlení výpočtu.

## Proveditelnost

- Agregace výpočetní síly není volba, ale nutnost pro dokončení výpočtu.

## Bezpečnost

- Duplikace klíčových částí systému pro případ havárie, či ohrožení důvěry jedné části.

## Cena

- Oddělení nesouvisejících částí aplikace, levnější údržba.
- Agregovaná výpočetní síla je levnější.

## Abstraktní model výpočetního systému

Procesor - Datová cesta - Datové úložiště

- 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





## Pozorování

- Výrobcům procesorů se nedaří zvyšovat výkon jednoho jádra.
- Fyzikální zákony brání neustálé miniaturizaci – okolo 5nm již nelze udržet elektrony v atomu.
- Současná technologie 14-16nm (dříve 22nm, 28nm, 32nm, 45nm, 65nm, 90nm).

## Řešení

- Multi-core a many-core procesory.
- Pravděpodobný způsob zvyšování výkonu i v budoucnosti.
- Částečný odklon od jednotek monolityckých jader k desítkám menších specializovaných, či k hybridním řešením.

## Důsledek

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

## **Role paralelismu v komunikaci**

- Větší propustnost komunikačních linek s následným efektem snižování latence.
- Robustnost a spolehlivost komunikačních linek.

## **Příklady paralelismu na datové cestě**

- Šířka sběrnice 32/64/128 bitů.
- Domácí dual-band WIFI router.

## Fakta

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

## 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**

- Disková pole
- Peer-to-Peer síť
- NUMA architektury
  - Více procesorové počítače s více paměťovými moduly uspořádanými tak, že přístup jednoho procesoru do různých paměťových modulů je různě rychlý.

# Paralelní výpočty

## Multitasking na jednom výpočetním jádru

- Aplikace se v běhu na CPU jádru střídají.
- Na jednom CPU zdánlivě "běží" více aplikací.
- Jednotka plánování OS je proces.

## Multitasking na více výpočetních jádrech

- Různé aplikace přiřazeny na různá výpočetní jádra.
- Jinak standardní multitasking na každém jednom jádře.
- Jednotka plánování OS je proces.

## Multitasking a multithreading

- Každá aplikace může mít více výpočetních vláken.
- Vlákna se v běhu na jednom CPU jádru střídají.
- Vlákna jedné aplikace mohou běžet na různých jádrech.
- Jednotka plánování OS je vlákno.



## Single Instruction Single Data

- V daný okamžik je zpracovávána jedna instrukce, která pracuje nad jedním datovým proudem.
- Klasický sekvenční výpočet.

## Single Instruction Multiple Data

- Jedna tatáž instrukce je vykonávána nad více datovými proudy.
- Vektorové instrukce CPU, architektura GPU.

## Multiple Instruction Multiple Data

- Nezávislý souběh dvou a více SISD, SIMD přístupů.
- Více jádrové procesory.

## Multiple Instruction Single Data

- Prakticky se nevyskytuje.

## Distribuovaný/paralelní systém

- Specifikován po částech (procesy).
- Chování systému vzniká interakcí souběžných procesů.
- Emergentní jevy.

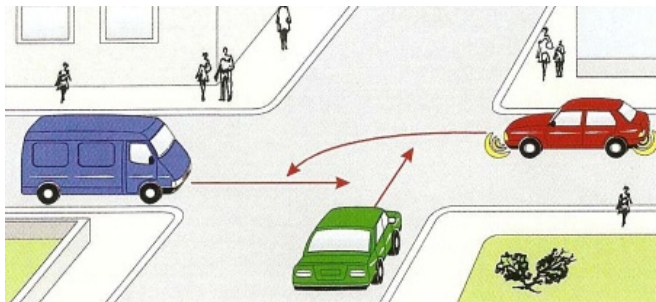
## Synchronizace

- Omezení na prokládání a souběh akcí jednotlivých procesů distribuovaného systému.

## Komunikace

- Přenos informace z jednoho procesu na druhý.

# Příklad distribuovaného systému



## Jevy

- Nekonzistentní vize konzistentního světa.
- Vzájemná interference

## Rizika

- Race-condition, nedeterministické chování.
- Uvážnutí (Deadlock, Livelock)
- Stárnutí (Starving)
- Přetečení buferů, problém producent-konzument.
- Zbytečná ztráta výkonu (aktivní čekání).

## 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?

# Vývoj paralelních systémů je náročnější

## 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 2-jádrovém stroji a na 4-jádrovém stroji běžel 2x rychleji, je preferována vyšší frekvence 2-jádrového procesoru.

# HPC

## 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))

