Přehled paralelních architektur

Přehled I.

- paralelní počítače se konstruují od poloviny šedesátých let
- vysoká cena integrovaných obvodů a tedy i řídících jednotek vedla ke stavění vektorových procesorů a procesorových polí, která obsahovala velký počet zjednodušených výpočetních jednotek
- s rozvojem VLSI obvodů klesala cena procesorů, což vedlo k rostoucí oblibě víceprocesorových systémů - 1980
- ty byly výhodnější pro zpracování kódu s podmínkami a mohlo na nich pracovat více uživatelů
- u systému se 100 a více procesory se naráží na potíže s paměťovým systémem
  - problém *cache coherence* a rovnocenné propojení procesorů s paměťovými moduly je náročné

Přehled II.

- tento problém řeší architektury s distribuovanou pamětí
- s rostoucím výkonem běžných PC a síťových komponent (Ethernet) spolu s klesajícími cenami těchto zařízení došlo v devadesátých letech k velkému příklonu k distribuovaným systémům
- po roce 2000 se ukazuje, že je obtížné nadále zvyšovat výkonu jednoprocesorových systémů zvyšováním frekvence CPU
  - ještě asi v roce 2002 IBM odhadovalo, že v roce 2010 budou mainframy používat 10 GHz procesory
- hledají se cesty, jak zvýšit výkon jednoho PC

Přehled III.

- jedním řešením jsou vícejádrové procesory = **SMP** ($\rightarrow$1980)
  - běžná CPU od Intel/AMD mají maximálně 18 jader
  - počet jader roste pomalu
- druhým řešením je GPGPU = **procesorová pole** ($\rightarrow$1970)
- trend současnosti je instalování GPU dohromady do superpočítačů
- objevuje se Intel Knight Corner

Přehled IV.

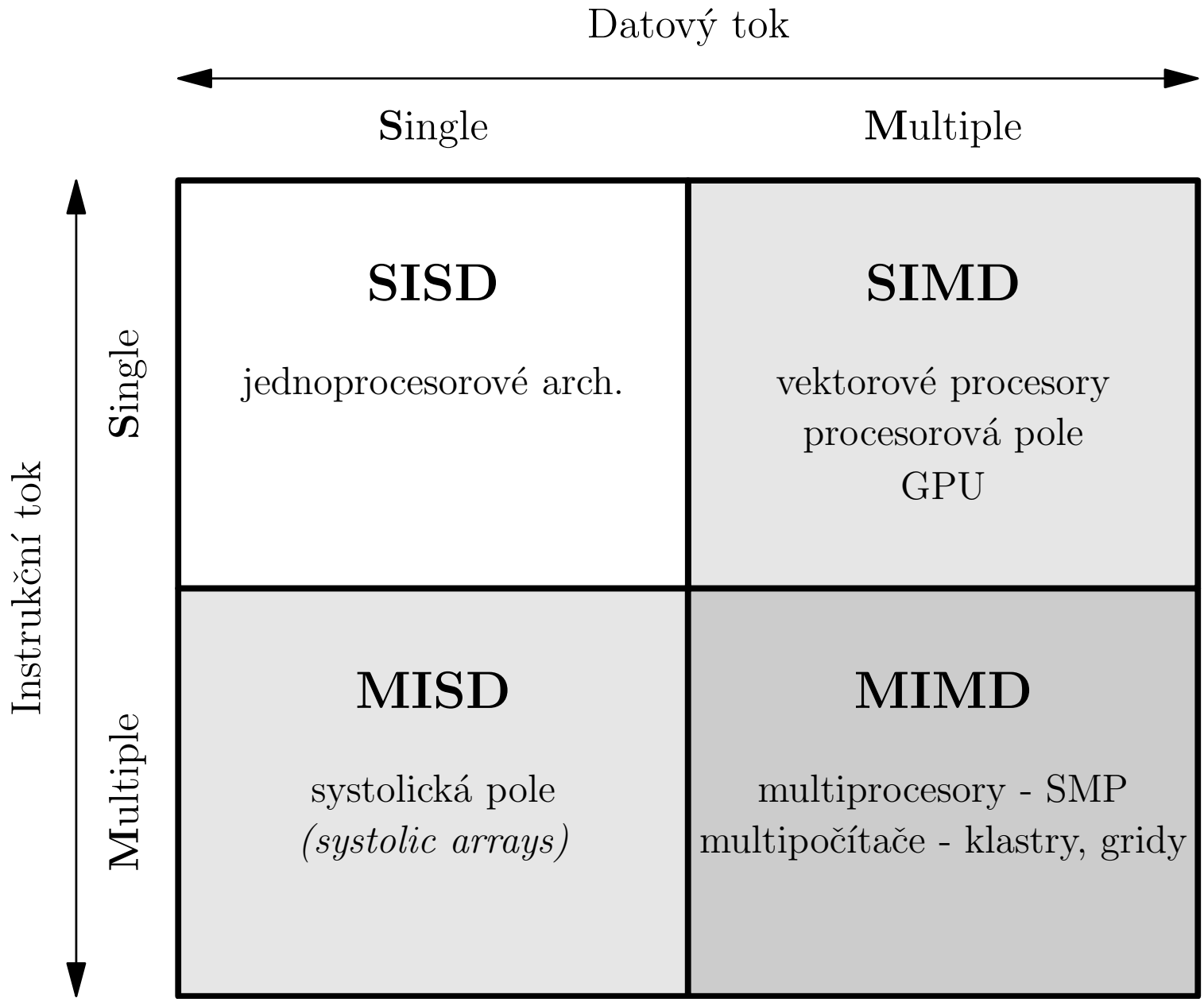
- vedle tohoto směru se rozvíjí technologie gridu
  - dnes jsou snahy využít grid i k numerickému počítání (asynchronní iterativní maticové řešiče)
- z *grid computingu* vzniká tzv. *cloud computing* (2007) apod.

Dělení paralelních architektur

Existuje mnoho způsobů, jak klasifikovat paralelní architektury. Nejznámější jsou:

- Flynnova taxonomie - 1966
  - rozděluje paralelní architektury v závislosti na toku instrukcí a dat
- rozdělení podle způsobu komunikace
  - architektury se sdílenou a distribuovanou pamětí

Flynnova taxonomie I.



Flynnova taxonomie II.

SISD architektury

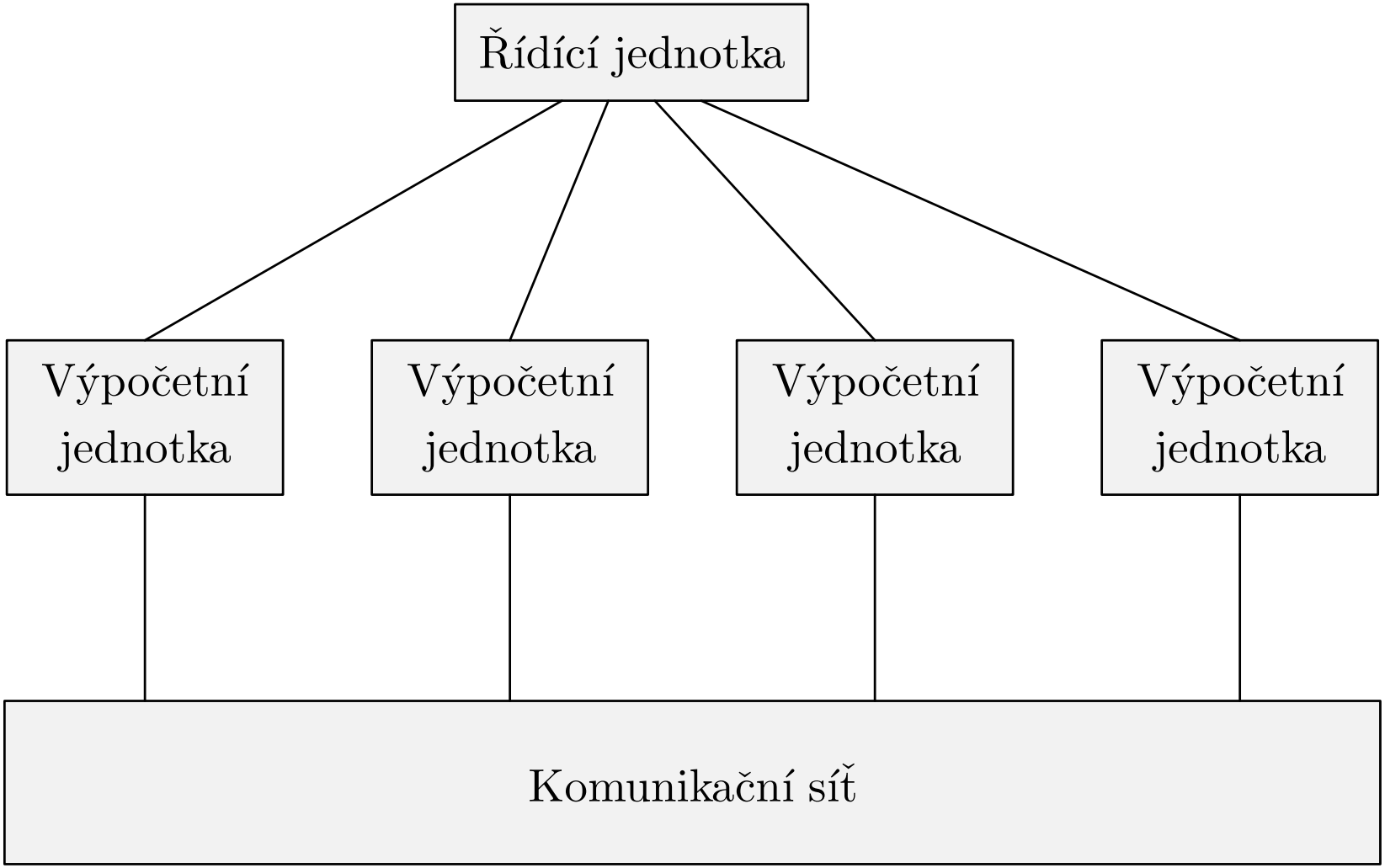
- na jednu instrukci připadá jeden jednoduchý datový typ
- jde typicky o jednoprocesorové architektury
- ty sice mohou zpracovávat více instrukcí a dat současně...
  - pipelining, superskalární zpracování, ...
- ... to je ale považováno spíše za paralelní **zpracování** sekvenčního kódu než za **provedení** paralelního kódu
- navíc - čistě sekvenční architektury dnes prakticky neexistují

Flynnova taxonomie - SIMD I.

SIMD architektury

- na jednu instrukci připadá více dat
  - např. instrukce sečíst dva vektory
- tyto architektury mají jednu řídící jednotku a několik jednotek výpočetních
- výpočetní jednotky v daný okamžik provádí stejnou instrukci, každá ale s různými daty
- jde hlavně o vektorové procesory ...
  - MMX, SSE, 3DNow! rozšíření procesorů architektury x086
  - GPU
    - moderní GPU umožňují rozdělení výpočetních jednotek do více skupin, které pak mohou zpracovávat odlišné úlohy
    - např. rozdělení na *vertex* a *pixel shadery*
- ... nebo procesorová pole

Flynnova taxonomie - SIMD II.

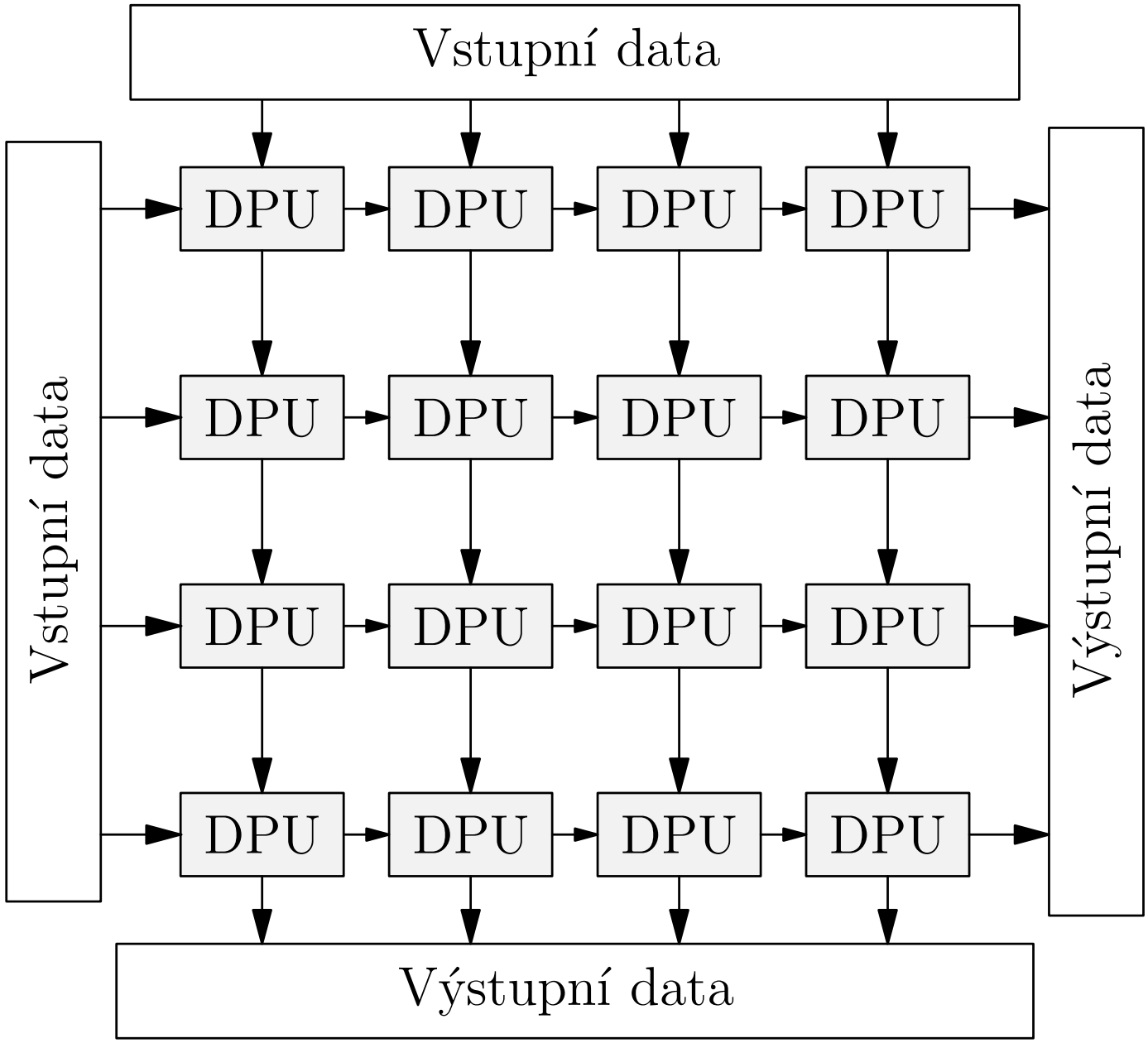


Flynnova taxonomie - MISD I.

MISD

- jedny data jsou postupně zpracovány více instrukcemi
- typickým zástupcem jsou systolická pole
  - název pochází od slova *systola* = srdeční kontrakce pumpující krev
- systolická pole jsou velmi speciální architektury
- příklady použití
  - některé třídící algoritmy
  - Hornerovo schéma pro vyčíslení polynomu
  - násobení matic - Cannonův algoritmus

Flynnova taxonomie - MISD II.

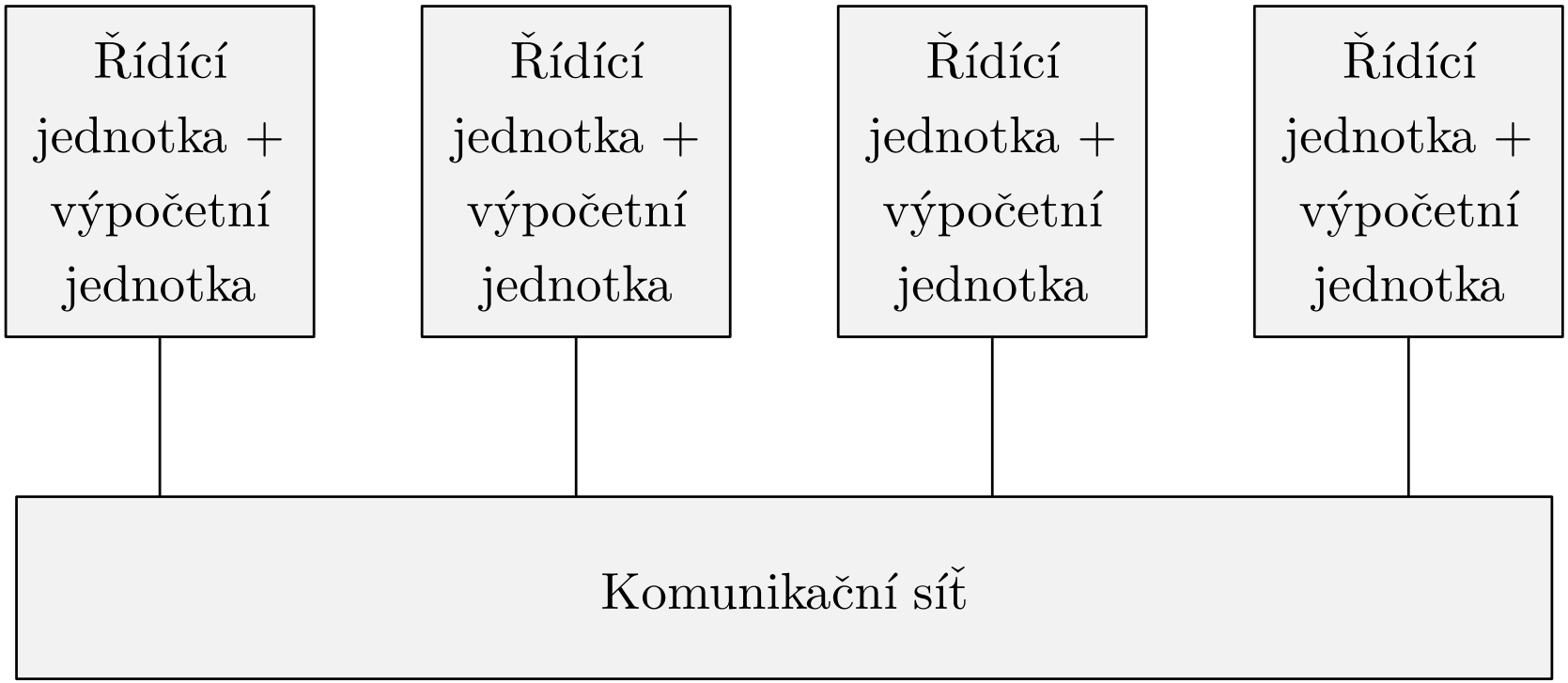


Flynnova taxonomie - MIMD I.

MIMD

- jde o systémy, které jsou schopné současně provádět různé instrukce a různými daty
- typickým příkladem jsou multiprocesory a multipočítače

Flynnova taxonomie - MIMD II.



Flynnova taxonomie - MIMD III.

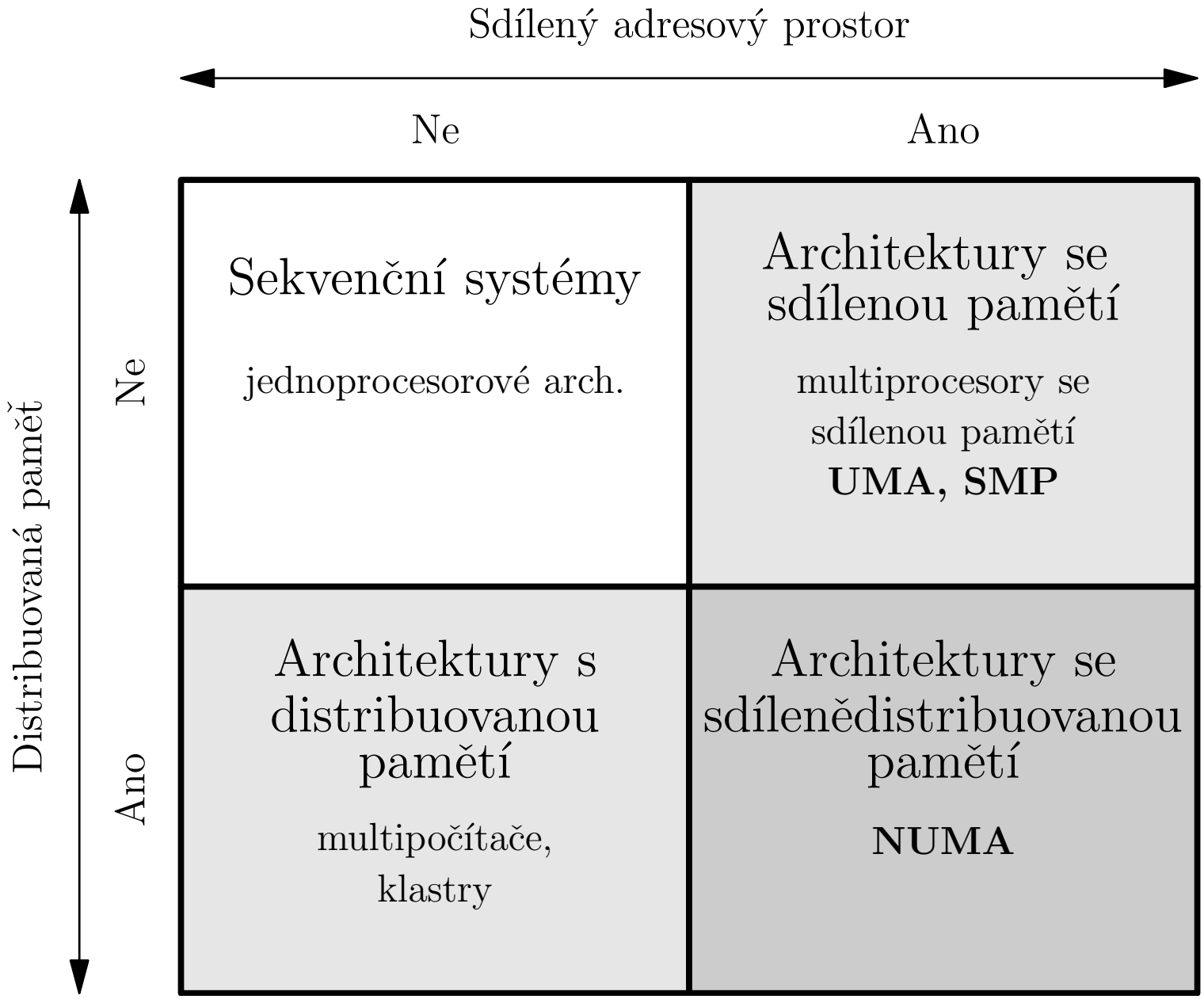
- v praxi nepíšeme zvláštní kód pro každý procesor
  - všude běží stejný program, ale podle ID procesoru se zpracovávají různé větve
- mluvíme pak spíše o **SPMD** architektuře
  - SPMD = single program multiple data
- prakticky všechny významné paralelní architektury dnes spadají do MIMD kategorie
  - to je velká nevýhoda Flynnovy taxonomie

Komunikační modely I.

Podle způsobu komunikace dělíme paralelní architektury na:

- systémy se sdíleným adresovým prostorem
- systémy s distribuovanou pamětí
- systémy se sdíleně distribuovanou

Komunikační modely II.



Architektury se sdílenou pamětí

- obsahují fyzicky sdílenou paměť, do které mají všechny procesory (výpočetní jednotky) stejně rychlý přístup
  - jde o UMA architektury
- stejná adresa na různých procesorech odkazuje na stejnou fyzickou paměťovou buňku
- sdílená paměť je tu prostředkem komunikace
- kromě sdílené paměti mohou být jednotlivé procesory vlastní lokální paměť - *cache*
  - ta bývá mnohem rychlejší ...
  - ... ale není přístupná ostatním procesorům, nejde tedy o NUMA architekturu
- typickým příkladem je SMP - symetrický multiprocesing
- sdílená paměť se stává úzkým hrdlem celého systému, proto se tyto architektury omezují na maximálně 100 procesorů
- pro vývoj se často používá standard OpenMP

Architektury s distribuovanou pamětí

- nemají společnou paměť ani virtuální adresový prostor
- komunikují spolu pomocí posílání zpráv přes komunikační síť
- to je náročnější z pohledu programátora
- odstranění společné paměti umožňuje vytvářet systémy s tisíci procesory
- pro vývoj se často používá standard MPI

Architektury se sdíleně distribuovanou pamětí

- jde o architektury s distribuovanou pamětí, které mají podporu pro sdílený virtuální adresový prostor
- jde o NUMA architektury
- podpora pro virtuální adresový prostor bývá zabudovaná již na úrovni hardware

Sdílený adresový prostor vs. posílání zpráv

- programování založené na posílání zpráv je náročnější
- posílání zpráv lze snadno a efektivně emulovat na systémech se sdílenou pamětí
  - programy napsané pomocí standardu MPI dobře běží i na SMP systémech
- neplatí to naopak