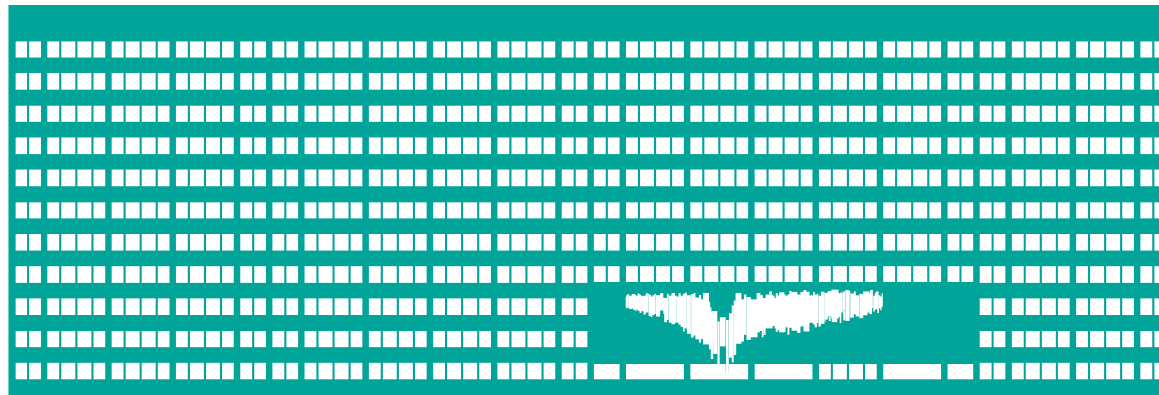


VŠB TECHNICKÁ  
UNIVERZITA  
OSTRAVA

VSB TECHNICAL  
UNIVERSITY  
OF OSTRAVA



[www.vsb.cz](http://www.vsb.cz)

# APPS

## Architektury počítačů a paralelních systémů / Architecture of Computers and Parallel Systems

### Parallel systems

Ing. Petr Olivka, Ph.D.  
Department of Computer Science,  
FEECS, VSB-TUO  
[petr.olivka@vsb.cz](mailto:petr.olivka@vsb.cz)  
<http://poli.cs.vsb.cz>

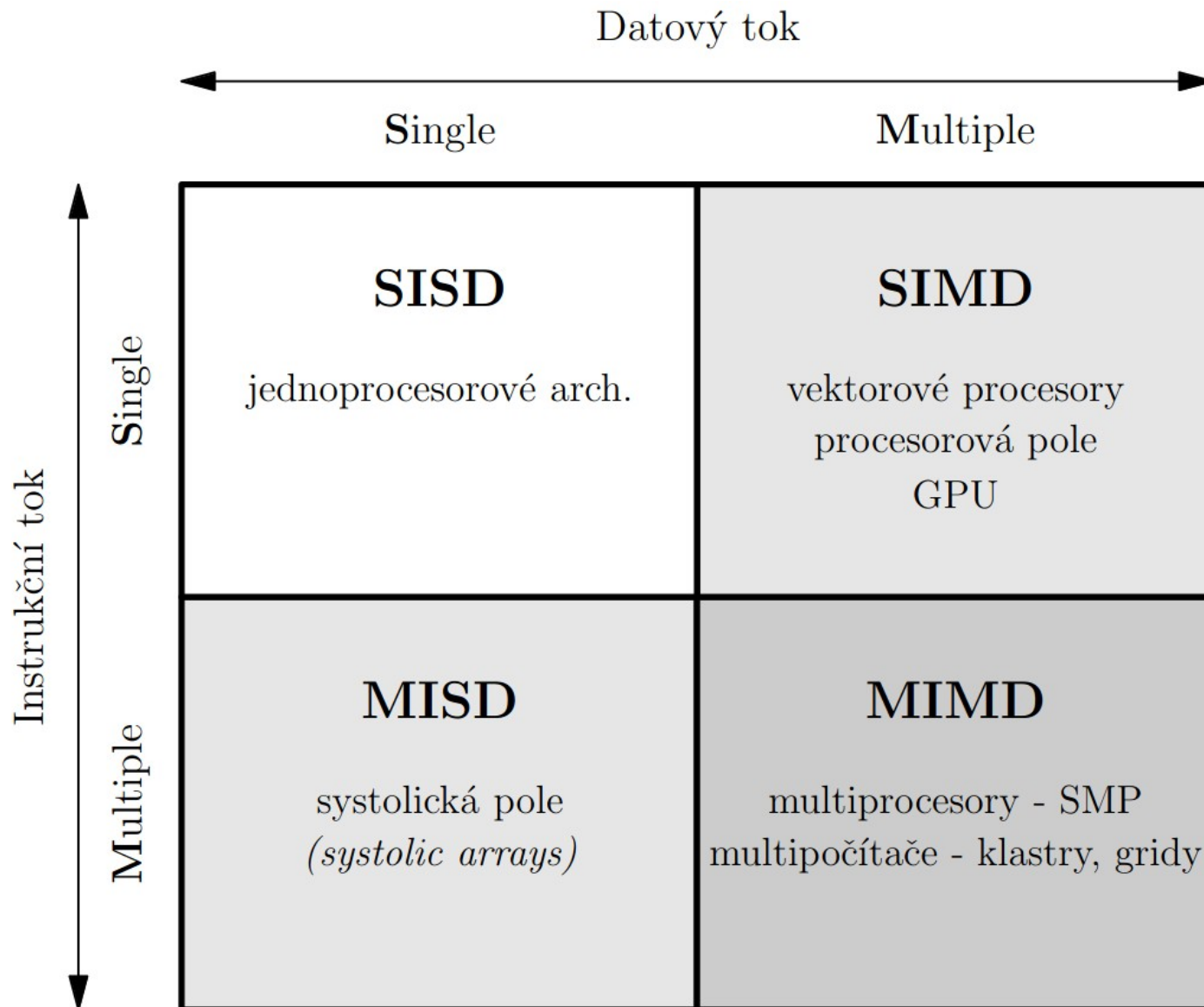
# Parallel systems

Aktuální problém je vývoj programů, které jsou schopné využívat více procesorů a jejich jader paralelně. Profesor David Patterson z katedry počítačů na univerzitě v Kalifornii publikoval v roce 2010 článek "The Trouble with Multicore". Důležitý je v tomto článku následující odstavec:

*“One of the biggest factors, though, is the degree of motivation. In the past, programmers could just wait for transistors to get smaller and faster, allowing microprocessors to become more powerful. So programs would run faster without any new programming effort, which was a big disincentive to anyone tempted to pioneer ways to write parallel code. The La-Z-Boy era of program performance is now officially over, so programmers who care about performance must get up off their recliners and start making their programs parallel.”*

Pokud tedy chceme, aby naše programy pracovaly rychleji, není jiná cesta, než psát programy jako paralelní.

# Flynnova taxonomie



# Flynnova taxonomie

## SISD architektury

- na jednu instrukci připadá jeden jednoduchý datový typ,
- jde typicky o jednoprocessorové 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,

# Flynnova taxonomie

## SIMD architektury

- na jednu instrukci připadá více dat,
- např. instrukce sečíst dva vektory,
- tyto architektury mají jednu řídicí 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 x86,
  - 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

## MISD architektury

- 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.

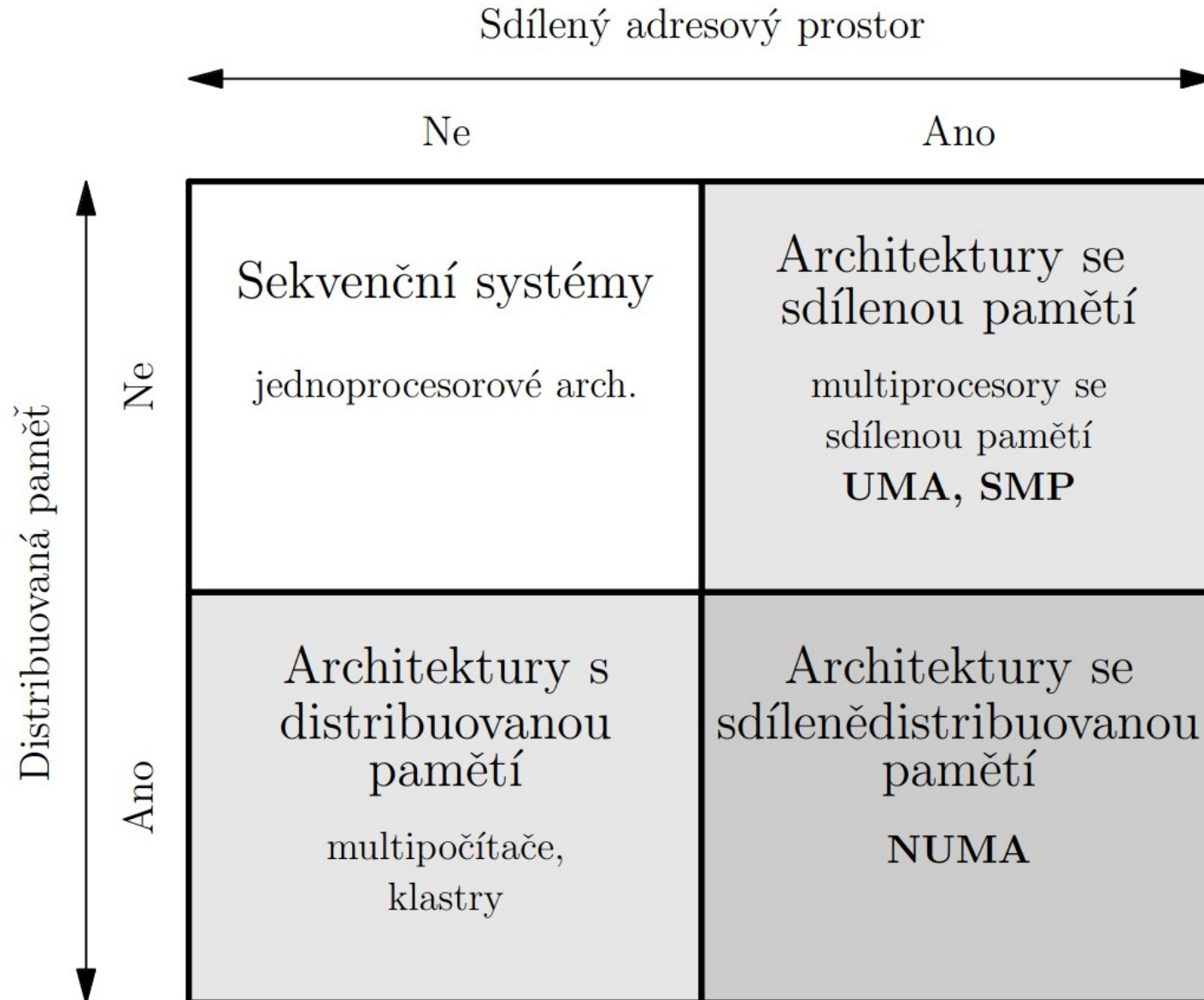
# Flynnova taxonomie

## MIMD architektury

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



# Komunikační modely

## Architektury se sdílenou pamětí

- obsahují fyzicky sdílenou paměť, do které mají všechny procesory (výpočetní jednotky) stejně rychlý přístup (výpočetní jednotka, výpočetní uzel, PE, processing element)
  - 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 mít 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 multiprocessing,
- sdílená paměť se stává úzkým hrdlem celého systému, proto se tyto architektury omezují na maximálně 100 procesorů,
- pro vývoj se často používá standard OpenMP.

# Komunikační modely

## 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.

# Komunikační modely

## 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 zabudována již na úrovni hardware

# Komunikační modely

## Sdílený adresový prostor vs. posílání zpráv

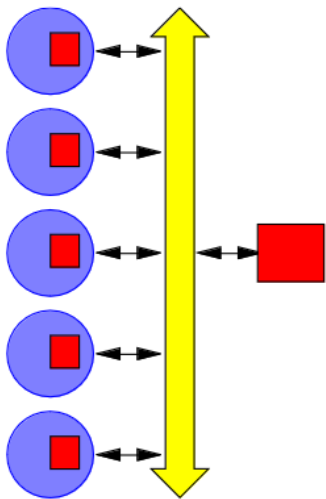
- programování založené na posílání zpráv je náročnější,
- posílání zpráv lze snadno a efektivně emulovat na systémech se sdílenou pamětí
  - programy napsané pomocí standardu MPI dobře běží i na SMP systémech,
- neplatí to naopak.

# SMP – symetrické multiprocesory

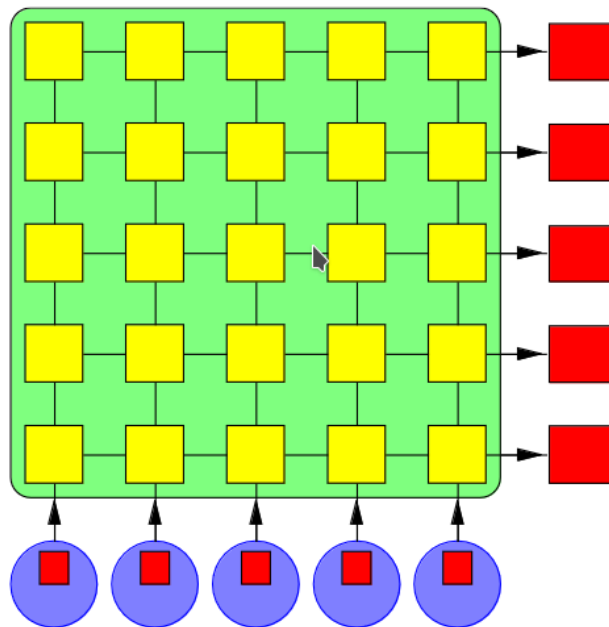
- PE = výkonné procesory s velkými skrytými paměťmi,
- Centrální sdílená paměť (1 či více bank),
- Nutnost synchronizace přístupu k sdíleným datům.
- Škálovatelnost limitovaná propustností paměťového rozhraní:

# SMP – symetrické multiprocesory

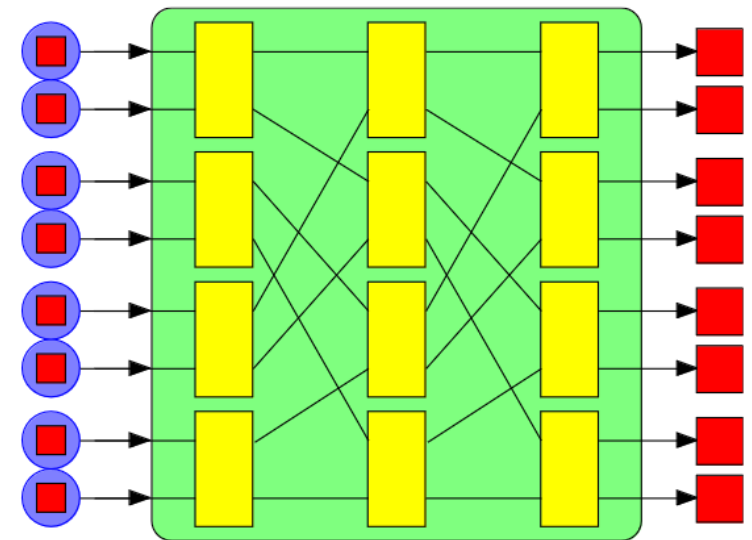
- (a) centrální sběrnice: kolem desítky procesorů,
- (b) křížový přepínač (crossbar): několik desítek procesorů (vysoká cena),
- (c) nepřímá vícestupňová síť: levnější ale chudší náhražka kř. přepínače.



(a)



(b)



(c)

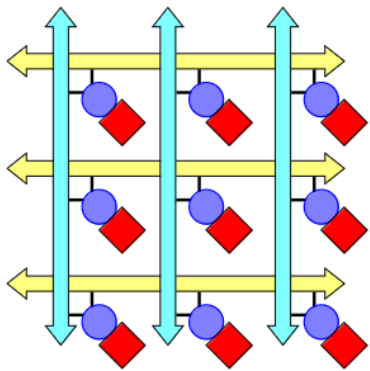
# DMP – multiprocessory s distribuovanou pamětí

- PE = výkonné samostatné počítače s lokálními (soukromými) paměťmi.  
mohou to být taktéž SMP!!!!!!,
- Všechny procesory mohou současně přistupovat do svých lokálních pamětí.
- V/V jsou distribuované a škálovatelné podobně jako paměť.
- Škálovatelnost je mnohem vyšší, neboť je limitovaná celkovou propustností propojovací sítě,
- ta je závislá na rychlosti kanálů a vlastnostech topologie (stupeň souvislosti, bisekční šířka, atd.):

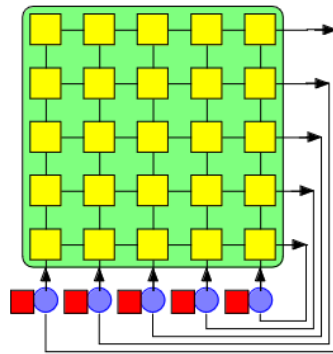


# DMP – multiprocesory s distribuovanou pamětí

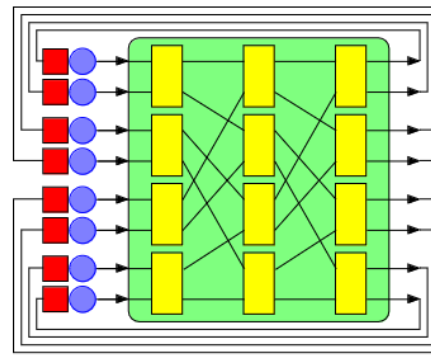
- (a) 2-D mřížka sběrnic.
- (b) křížový přepínač
- (c) nepřímá vícestupňová síť
- (d) přímá síť



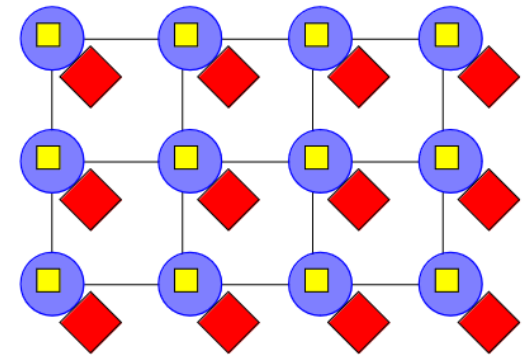
(a)



(b)



(c)



(d)

# Amdahlův zákon – růst rychlosti výpočtu

Problém růstu výkonnosti systému jako celku

- ztráty spojené s komunikací,
- nedokonalé vytížení procesorů,
- neznalost vhodných algoritmů,
- zrychlení (lineární (ideálně), superlineární(pouze v některých případech), zpomalení – často)

# Amdahlův zákon – růst rychlosti výpočtu

Čistě paralelních úloh velmi málo, většinou kombinováno se sériovým zpracováním.

- $S_z$  součinitel zrychlení (poměr doby, která by byla potřebná, kdyby celý výpočet probíhal na 1 procesoru k době potřebné při částečné paralelizaci),
- $f_s$  - podíl sériové délky,
- $f_p$  - podíl paralelní délky ( $f_s + f_p = 1$ ),
- $t$  - celková doba sériového výpočtu
- při použití  $N$  procesorů klesne délka paralelního výpočtu  $f_p$  na  $t \cdot \frac{f_p}{N}$

$$S_z = \frac{t}{t \cdot f_s + \left(t \cdot \frac{f_p}{N}\right)} = \frac{1}{(1 - f_p) + \frac{f_p}{N}}$$

