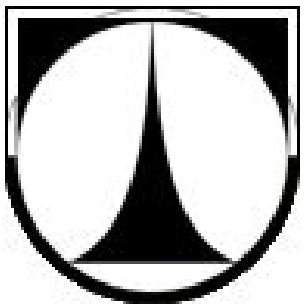
Paralelizace numerických metod

Jiří Hozman

jiri.hozman@tul.cz

Technická univerzita v Liberci
Fakulta přírodovědně-humanitní a pedagogická
Katedra matematiky a didaktiky matematiky

Přednáška k předmětu Matematika III



Osnova

1. Superpočítače
2. Paralelní výpočty
3. Úvod do paralelního programování

1 Superpočítače

2 Paralelní výpočty

3 Úvod do paralelního programování

Motivace

Moorův zákon (platí už přes 30 let)

Výkonnost procesorů se zdvojnásobuje každých 18 měsíců.

Otázka

Je výkonnost jednoprocesorových systémů dostatečná pro řešení současných i budoucích úloh?

Odpověď

Není.

Alternativně lze výkonnost systémů zvýšit (v každé vývojové etapě) nad výkonnost 1 procesoru paralelním zpracováním několika procesory současně.

Superpočítač

- vysoce výkonný počítač, resp. počítačový systém,
- relativní označení závislé na vývoji hardware

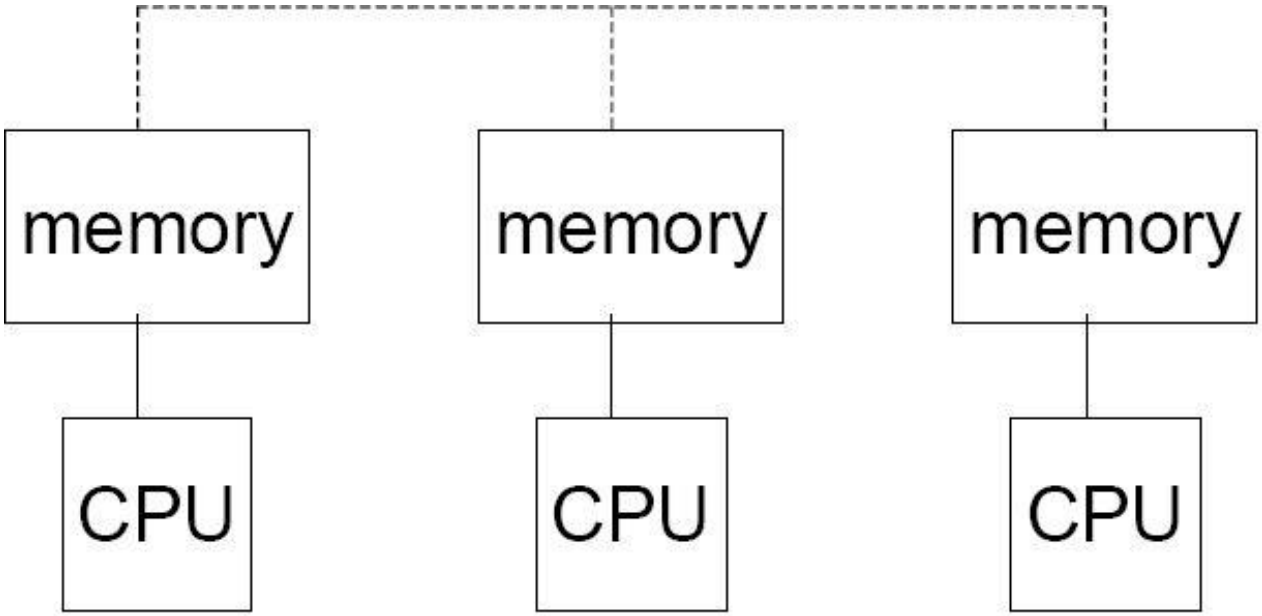
Superpočítač

Paralelní systém (superpočítač) je množina procesorů (počítačů), které vzájemně kooperují a komunikují, aby řešily "velké problémy rychle".

- systémy se sdílenou pamětí (shared memory) - komunikace pomocí synchronizace procesů
- systémy s distribuovanou pamětí (distributed memory) - komunikace mezi procesy zasílaní zpráv
- systémy s hybridním uspořádáním - dnes nejčastější

Architektura moderních superpočítačů

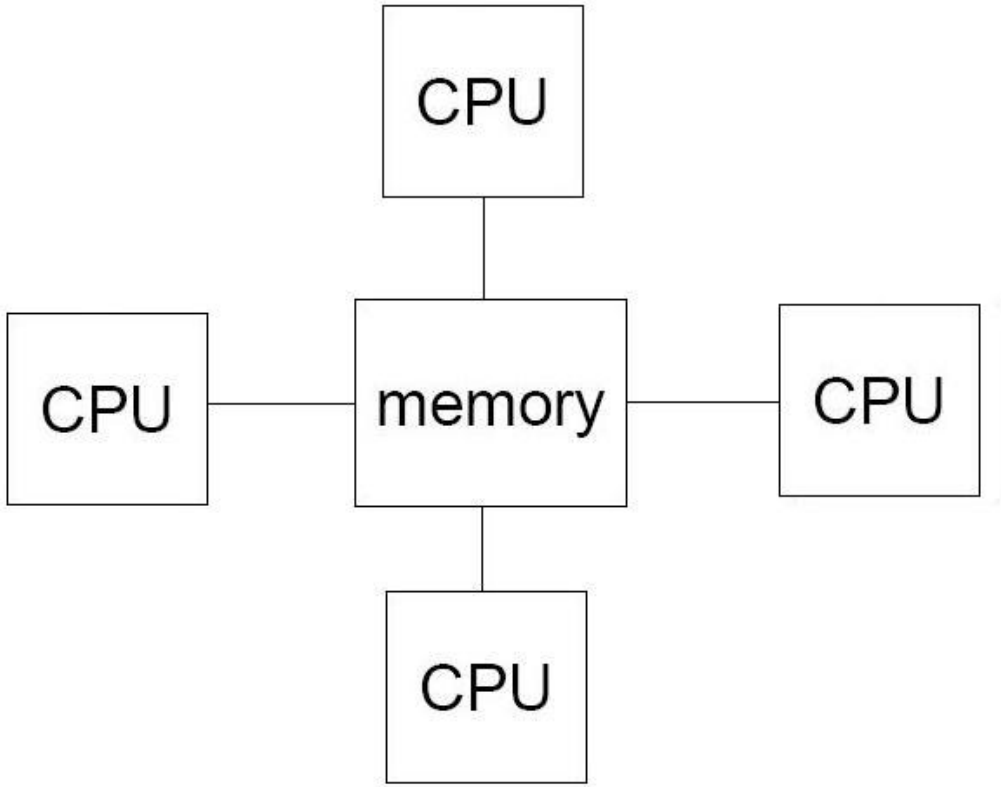
- cluster - propojení velkého množství jednoprocesorových počítačů prostřednictvím speciální vysokorychlostní počítačové sítě



- na každém procesoru běží samostatné úlohy operačního systému

Architektura moderních superpočítačů

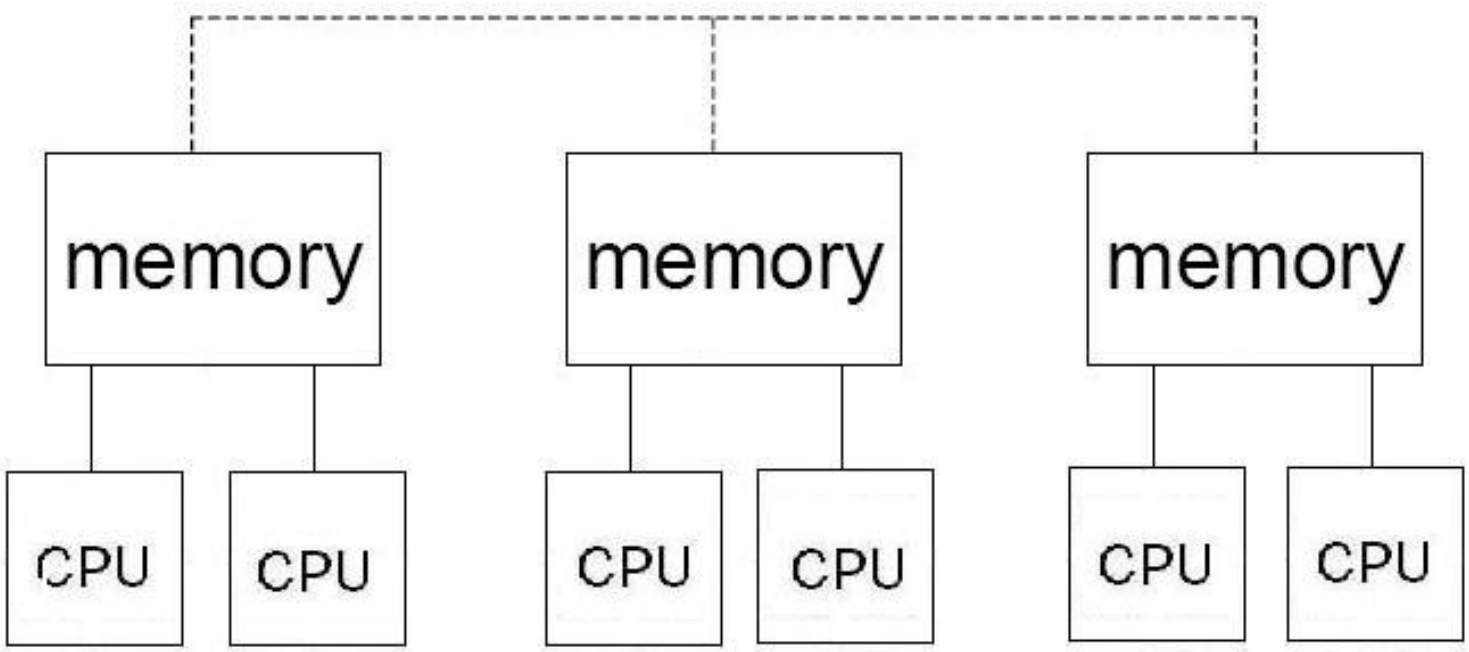
- multiprocesorový počítač - počítač, na kterém běží jeden samostatný operační systém a používá více CPU



- nevýhoda - dražší než cluster z běžných počítačů při stejné výkonnosti

Architektura moderních superpočítačů

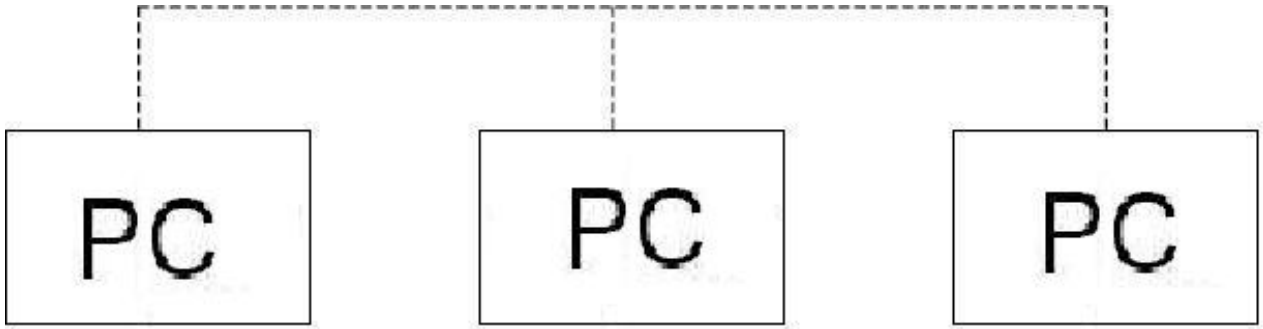
- hybridní cluster - cluster multiprocesorových sestav (dnes nejčastější uspořádání)



- např. vyhledávač Google

Architektura moderních superpočítačů

- spojení velkého množství běžných osobních počítačů (PC) pomocí internetu



- snadno paralelizovatelné výpočty nevyžadující rychlou komunikaci
- projekt SETI at home

Superpočítače ve světě

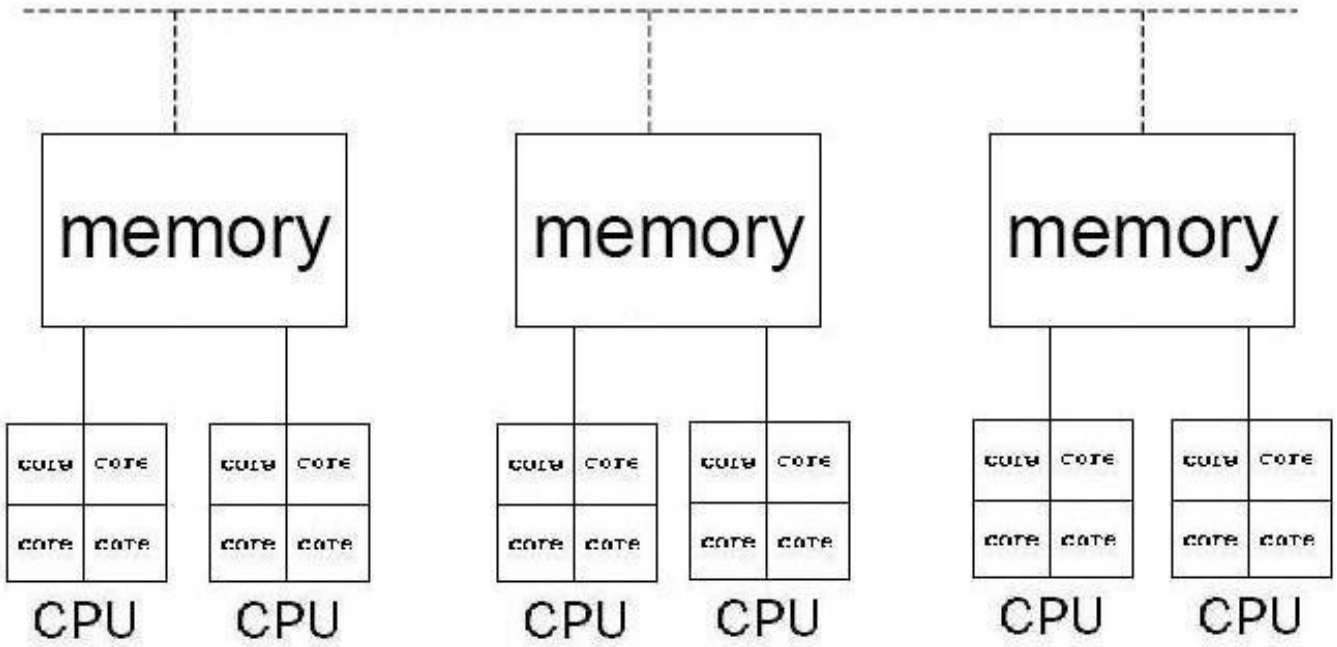
- www.top500.org - žebříček nejvýkonnějších počítačů světa

Aktuální pořadí (2010)

1. Jaguar - Cray XT5-HE Opteron Six Core 2.6 GHz,
2. Roadrunner - BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, *Los Alamos (USA)*,
3. Kraken XT5 - Cray XT5-HE Opteron Six Core 2.6 GHz, *National Institute for Computational Sciences/University of Tennessee (USA)*,
4. JUGENE - Blue Gene/P Solution, *Forschungszentrum Juelich (Německo)*,
5. Tianhe-1 - NUDT TH-1 Cluster, Xeon E5540/E5450, ATI Radeon HD 4870, *National SuperComputer Center in Tianjin (Čína)*.

Jaguar

- 1. místo,
- Oak Ridge National Laboratory, Tennessee (USA),
- model Cray (podle zakladatele Seymoura Craye),
- 37 376 procesorů typu Six-Core AMD Opteron 2.6 GHz,
- 7 832 procesorů typu Quad-Core AMD Opteron 2.6 GHz,
- 26 520 uzlů s celkově 255 584 jádry

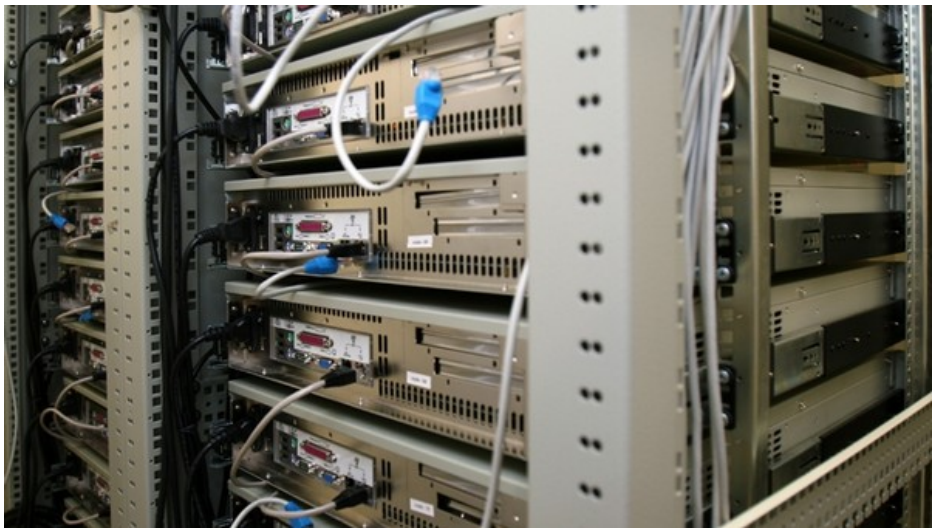


Jaguar

Superpočítače v České Republice

Amálka - nejvýkonnější paralelní systém v ČR (2010)

- Ústav fyziky atmosféry Akademie věd ČR
- 356 procesorů Intel XEON (800 jader) $\longrightarrow$ výkon $6,38$ TFlops
- náročné numerické výpočty a experimenty v rámci spolupráce s ESA a NASA



1 Superpočítače

2 Paralelní výpočty

3 Úvod do paralelního programování

Paralelní výpočty

- počítání na paralelních systémech
- využití více než 1 procesoru při výpočtech

Hlavní důvody pro paralelní počítání

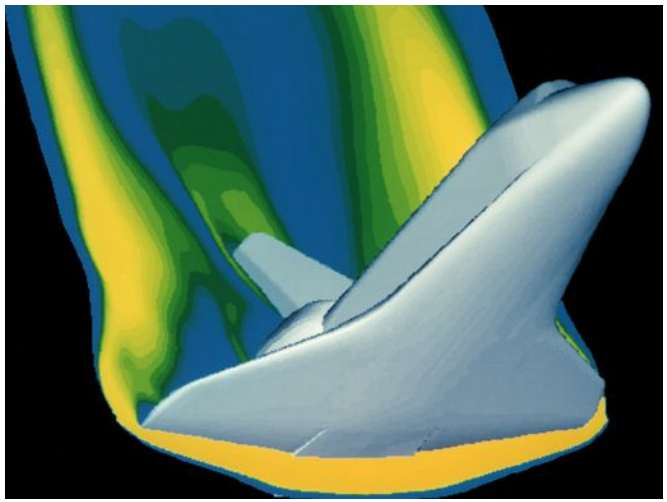
- počítačové simulace (v reálném čase)
- řešení problémů větších rozměrů (podrobněji a přesněji)
- zkrácení doby výpočtu
- velké nároky na pamět
- nemožnost zvyšovat vývoj procesorů do nekonečna
- úspornost - cena nejvýkonějších procesorů roste rychleji než jejich výkon

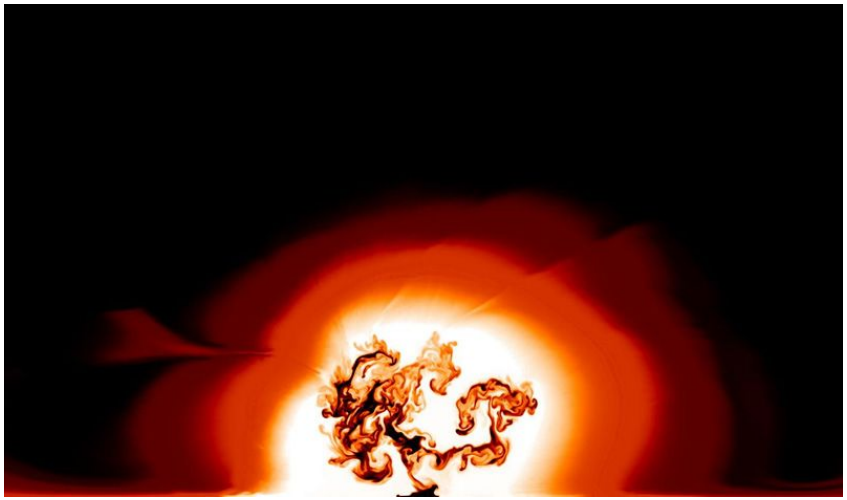
Aplikace paralelního řešení problémů

- věda - biologie, chemie, genetika, farmacie,...
- užití atomu - jaderné reaktory, plazma,...
- vesmír - planety, hvězdy, komety,...
- zdravotnictví - protézy, proudění krve,...
- průmysl, strojírenství - automobily, turbíny, kompresory,...
- počasí - meteorologie, hydrologie, vulkanologie, klimatologie,...
- vojenské aplikace, letectví, kosmonautika,...
- CAD, robotika, kartografie,...
- multimédia, počítačové vidění,...
- internetové vyhledávače, zpracování dat, databáze,...
- a další

Počítačové simulace

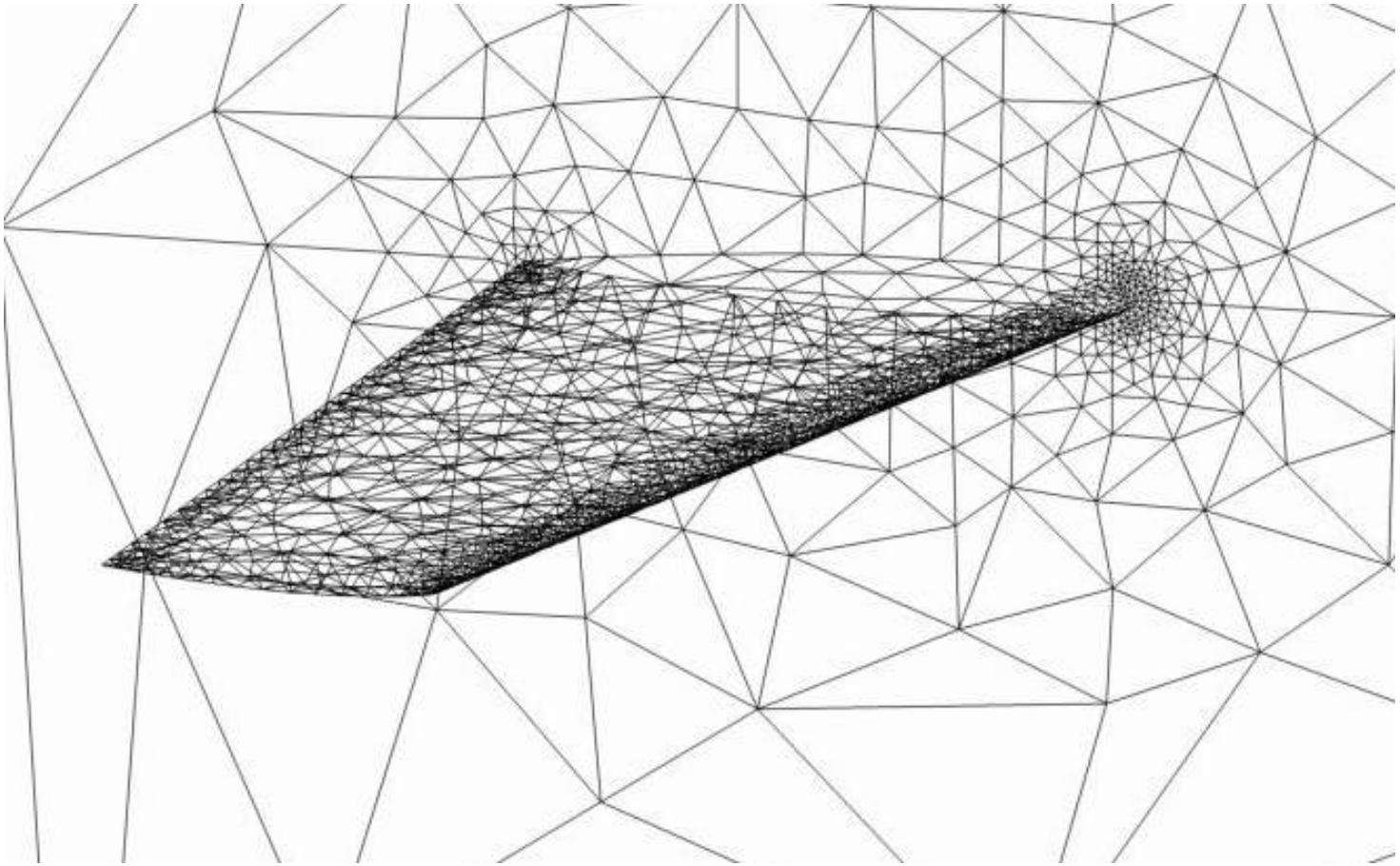
- od konce 40. let 20. století
- parciální diferenciální rovnice (a jejich systémy)
- nemožnost nalézt analytická řešení
- levnejší alternativa k náročným fyzikálním experimentům
- numerické metody
- např. obtékání raketoplánu, modelovaní jaderné reakce





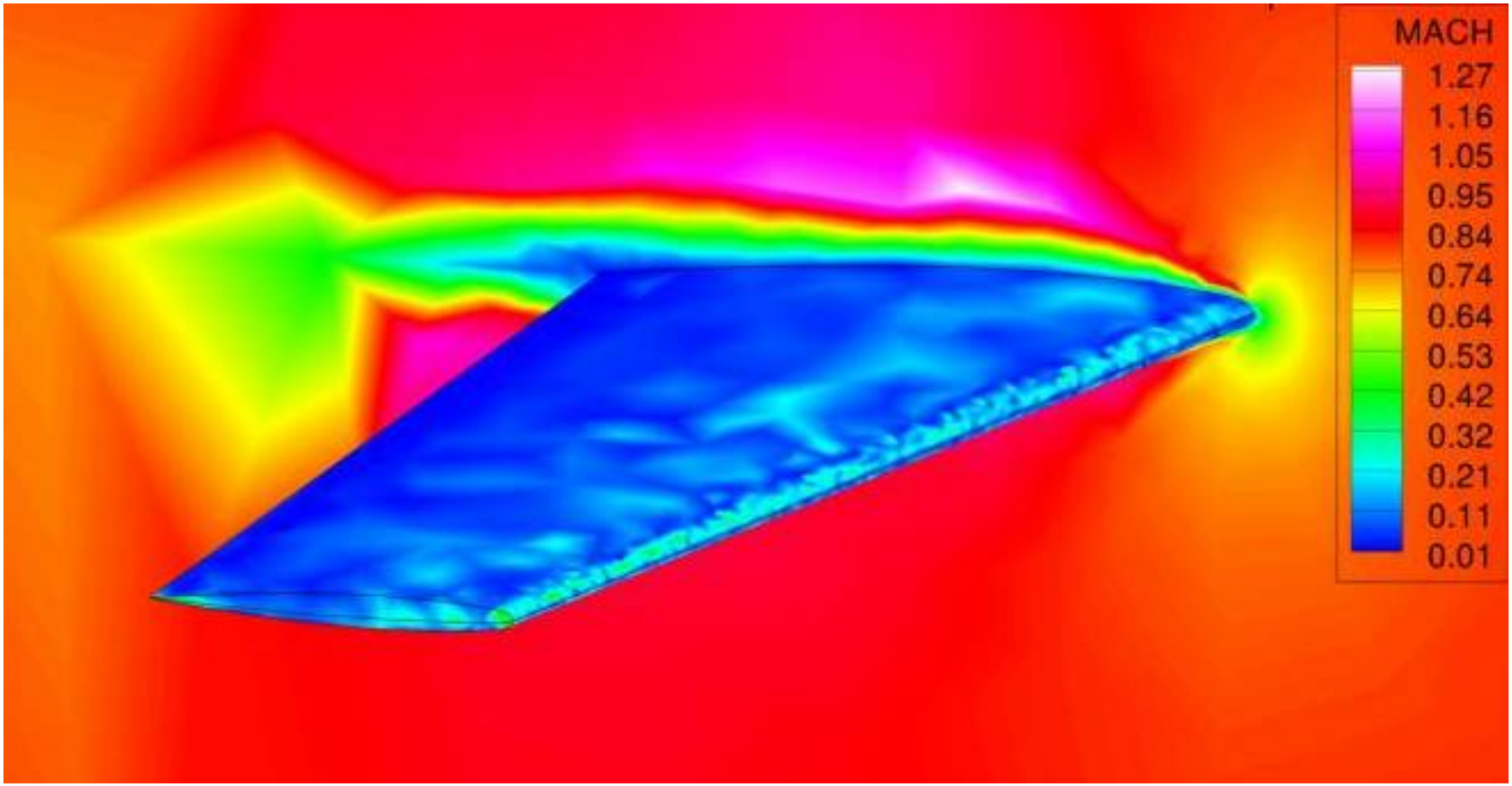
Motivace - aplikace v letectví (1)

- obtékání křídla ONERA M6 (KNM MFF UK)



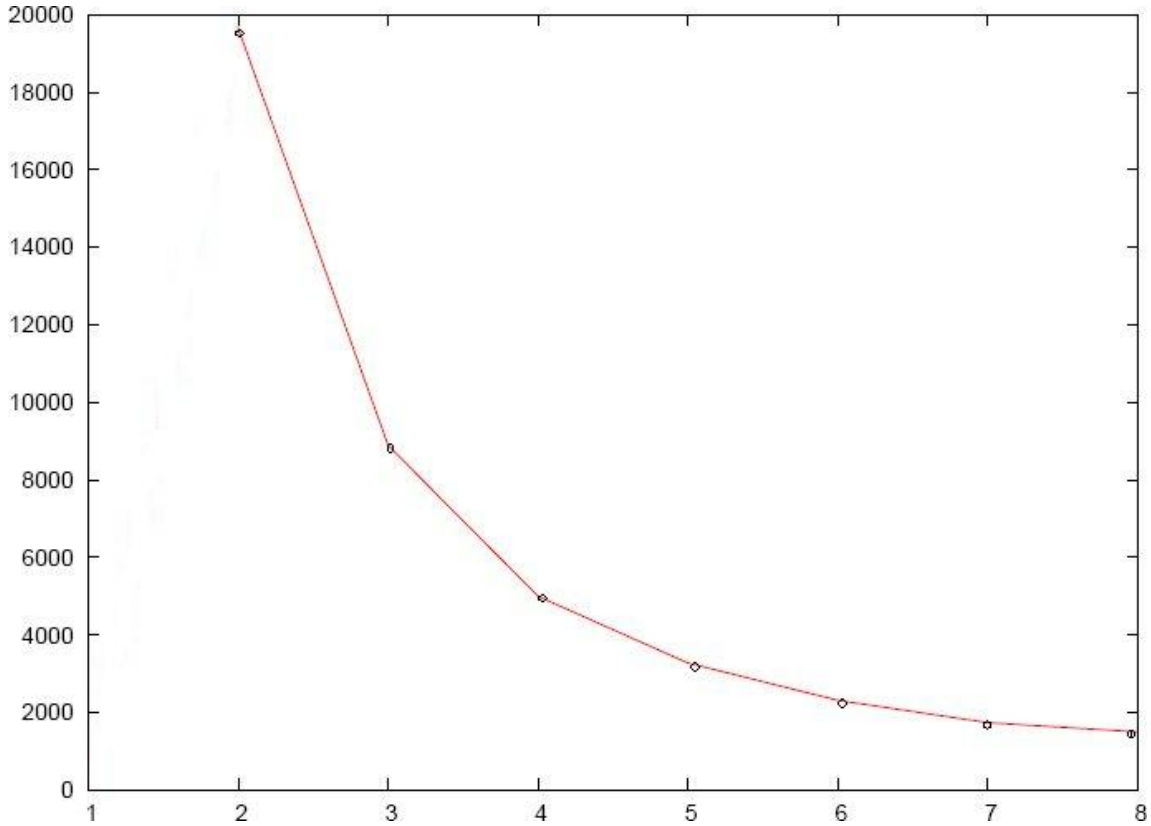
Motivace - aplikace v letectví (2)

- rozložení Machova čísla



Motivace - aplikace v letectví (3)

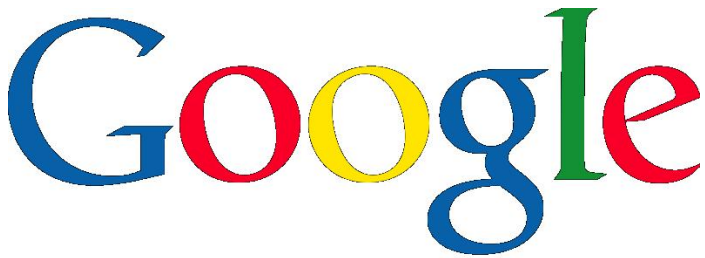
- výpočetní čas programu ADGFEM na clusteru Sněhurka



# procesor	2	3	4	5	6	7	8
čas [s]	19 645	9 292	4 999	3 751	2 813	2 007	1 501

Motivace - vyhledávač Google

- katalogizace internetových stránek
- v roce 2000 řeší $\sim$ 1000 dotazů/s $\rightarrow$ 70 mil./den
- cíl - vyhledávání < 0.5 s
- aktualizace webu každé 4 týdny - 1.3 miliardy URL (+60% anotací) $\rightarrow$ datový tok 53 Mb/s
- počet dotazů roste o 20% měsíčně
- počáteční sestava $>$ 6000 procesorů
- vždy k dispozici (+3 záložní systémy)
- numerické metody - iterační řešení stochastických matic o řádu odpovídajícímu počtu internetových stránek



1 Superpočítače

2 Paralelní výpočty

3 Úvod do paralelního programování

Modely paralelních architektur

Flynovo rozdělení podle počtu proudů instrukcí a dat:

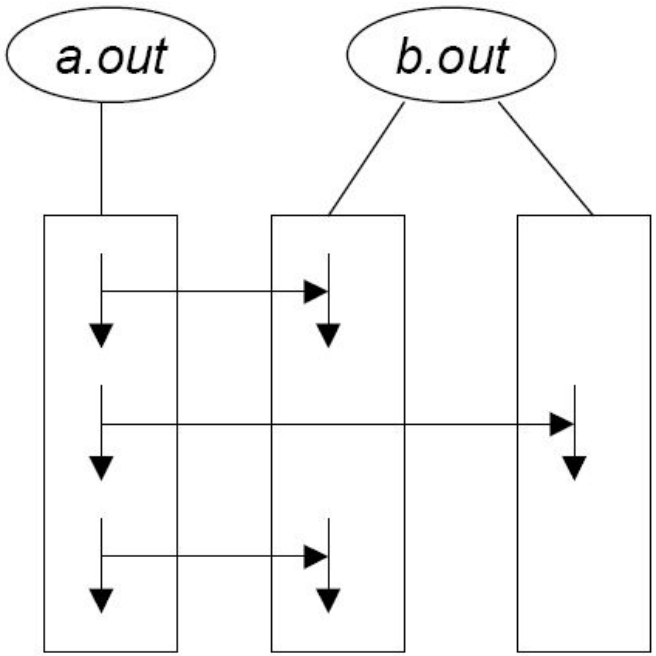
- SISD - Single Instruction, Sigle Data stream (sekvenční počítače)
- SIMD - Single Instruction, Multiple Data stream (datově paralelní systémy)
- MIMD - Multiple Instruction, Multiple Data stream (plně paralelní systémy)
  - sdílená paměť', komunikace přes sdílené proměnné
  - distribuované paměť', komunikace zasíláním zpráv
  - hybridní uspořádání - distribuovaná sdílená paměť

Druhy paralelních úloh

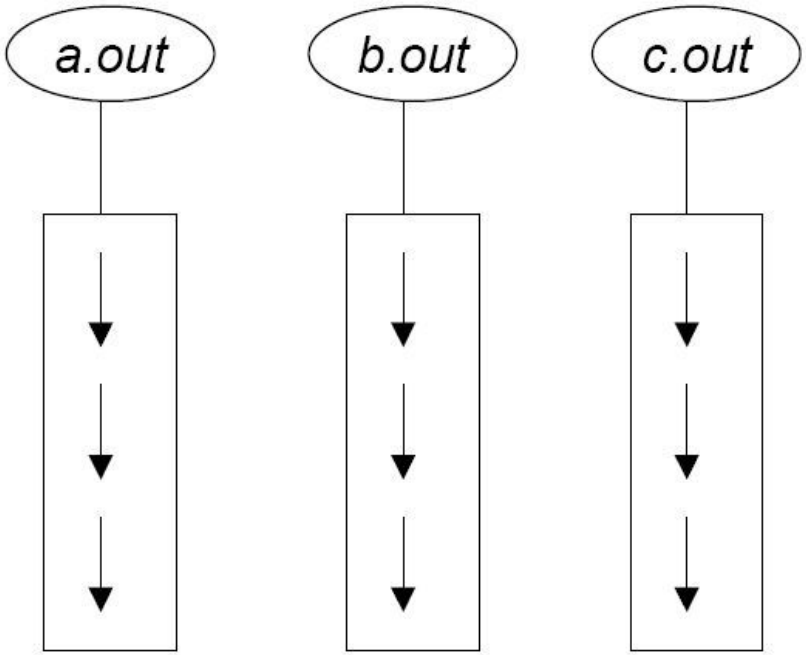
Paralelní úlohy z hlediska spolupráce během výpočtu:

- MPMD (Multiple Program, Multiple Data) úlohy

Master/Worker



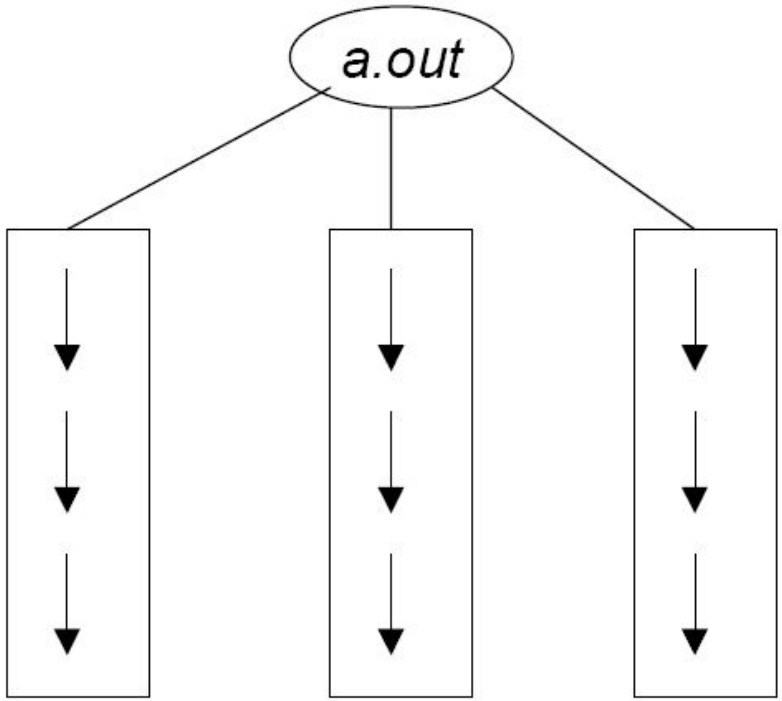
Coupled Multiple Analysis



- SPMD (Single Program, Multiple Data) úlohy

SPMD úlohy

- stejná kopie programu (single program) na všech CPU
- během výpočtu zpracovávají různá data (multiple data)
- typické pro numerické výpočty
- předmět našeho zájmu



Modely paralelního programování

- automatická paralelizace programu kompilátorem
  - nejpohodlnější pro programátora
  - nejslabší propracovanost a nízká efektivita
- datově paralelní programování - sekvenční program řídí paralelní operace
  - vyžaduje menší zainteresovanost programátora
  - např. HPF (High Performance Fortran)
- programování se sdílenými proměnnými (vlákny)
  - automatické přenosy dat, komunikace pomocí synchronizace
  - sekvenční program lze paralelizovat postupně od nejkritičtějších míst $\rightarrow$ vysoká efektivita
  - větší požadavky na programátora, např. OpenMP
- programování zasíláním zpráv (message passing)
  - komunikace pomocí explicitních příkazů (send, receive)
  - vyžaduje rozčlenění a distribuci dat a práce mezi procesory $\rightarrow$ vysoká efektivita
  - nejvyšší míra zainteresovanosti programátora, např. MPI

Problémy s vývojem paralelních systémů

- hardware - dva přístupy
  - saturace sběrnice - omezení při konstrukci multiprocesorů sdílející společnou paměť na maximální počet desítek procesorů
  - switch - rychlá komunikační síť spojující počítače do clusteru, rychlost meziprocesorové komunikace $\sim$ rychlost komunikace mezi CPU a vnitřní pamětí
- algoritmy - algoritmy pro sekvenční programy nelze vždy dobře paralelizovat $\longrightarrow$ nutnost vytvářet nové paralelní algoritmy
- software - automatické kompilátory či paralelní numerické knihovny nedosahují vždy vysoké efektivity

Výkonost paralelního zpracování

- ukazatelem doba výpočtu (computational time, c.t.) $[\mu s]$

Amdahlův zákon

$$100 - P + \frac{P}{n} \quad [\%]$$

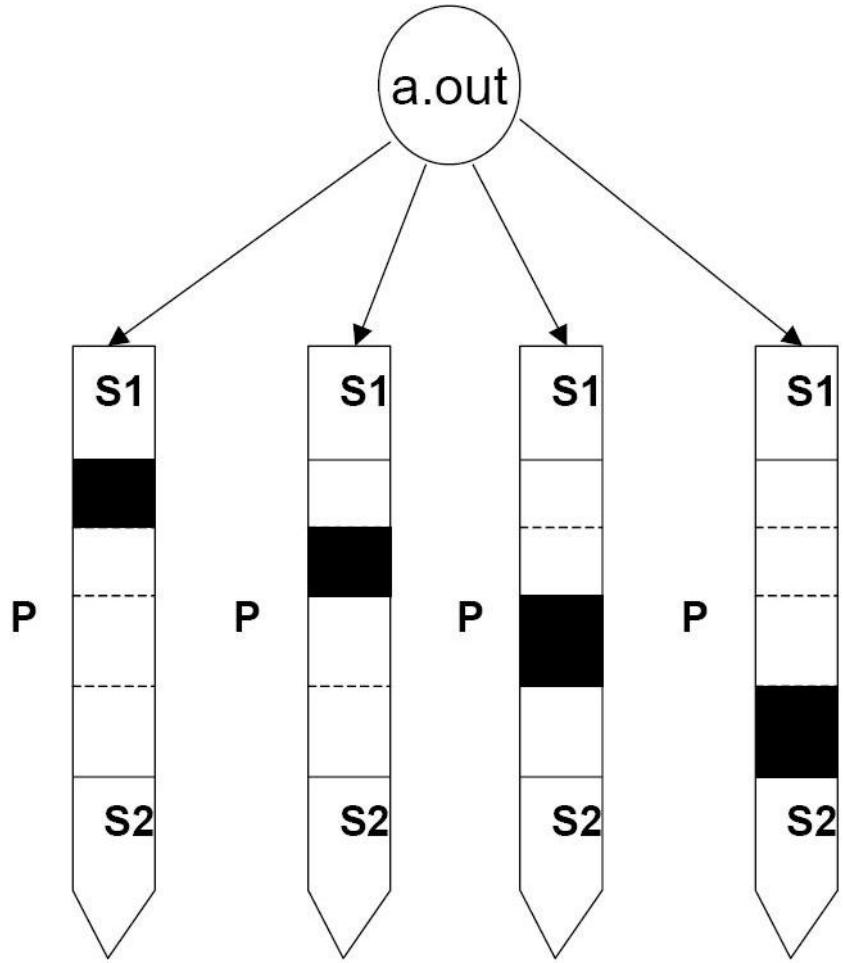
- P...část programu, kterou lze paralelizovat
- n...počet použitých CPU
- poměrné porovnání výpočtového času oproti času na jednoprocesorovému počítači
- neplatí přímá úměrnost mezi počtem CPU a zrychlením
- pro $n \longrightarrow \infty$ je zrychlení výpočtu

$$\frac{100}{100 - P}$$

Sekvenční program vs. paralelní program

Sekvenční program

Paralelní program



Motivace - doba výpočtu

- součet prvků vektoru $\vec{a} \in \boldsymbol{R}^{10^9}$: $s = \sum_{i=1}^{10^9} a_i$
- zatížení jednoho CPU

akce	input	sum	ouput
vytížení CPU [%]	6	90	4

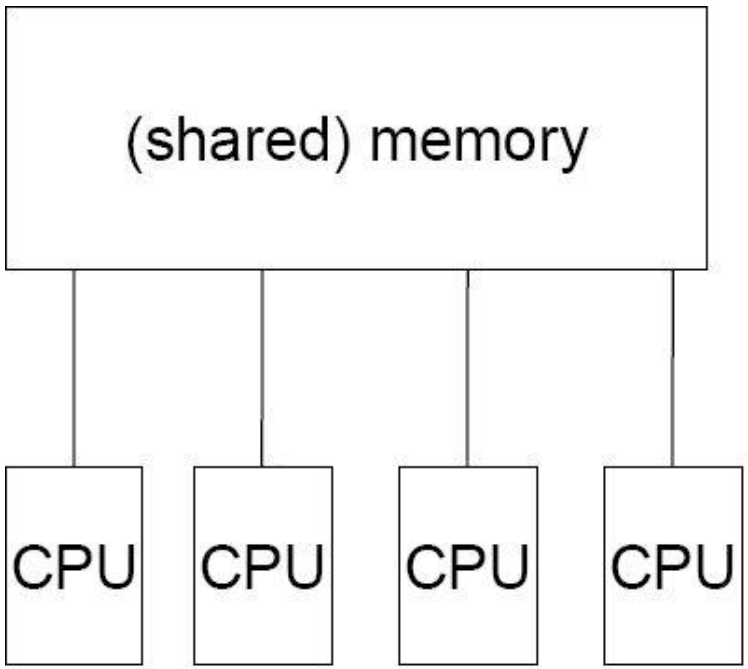
- pouze výpočet součtu lze paralelizovat $\longrightarrow P = 90\%$
- zrychlení podle počtu CPU

CPU	1	2	3	4	5	6	7	8
zrychlení [%]	100	55	40	32.5	28	25	22.9	21.3

Závislost paralelizace na architektuře systému (1)

Sdílená paměť' (shared memory)

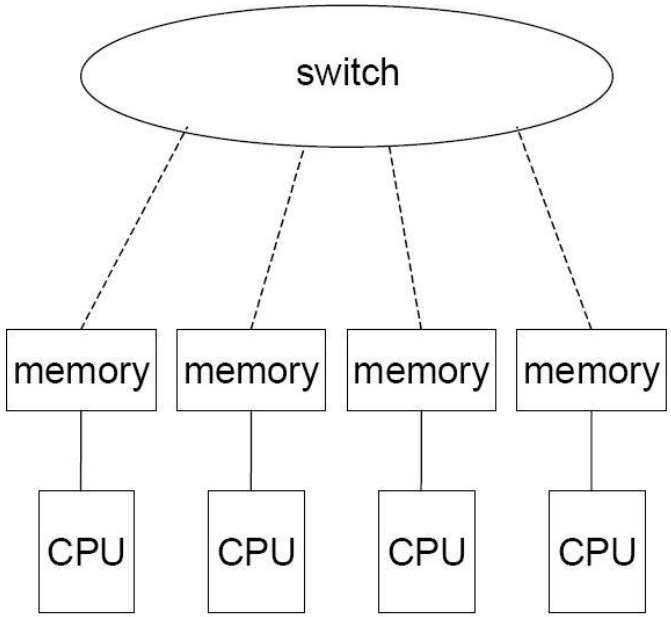
- snazší paralelizace, malé zásahy do sekvenčního programu
- sdílení a výměna informací mezi CPU přes společnou paměť'
- např. OpenMP



Závislost paralelizace na architektuře systému (2)

Distribuovaná paměť' (distributed memory)

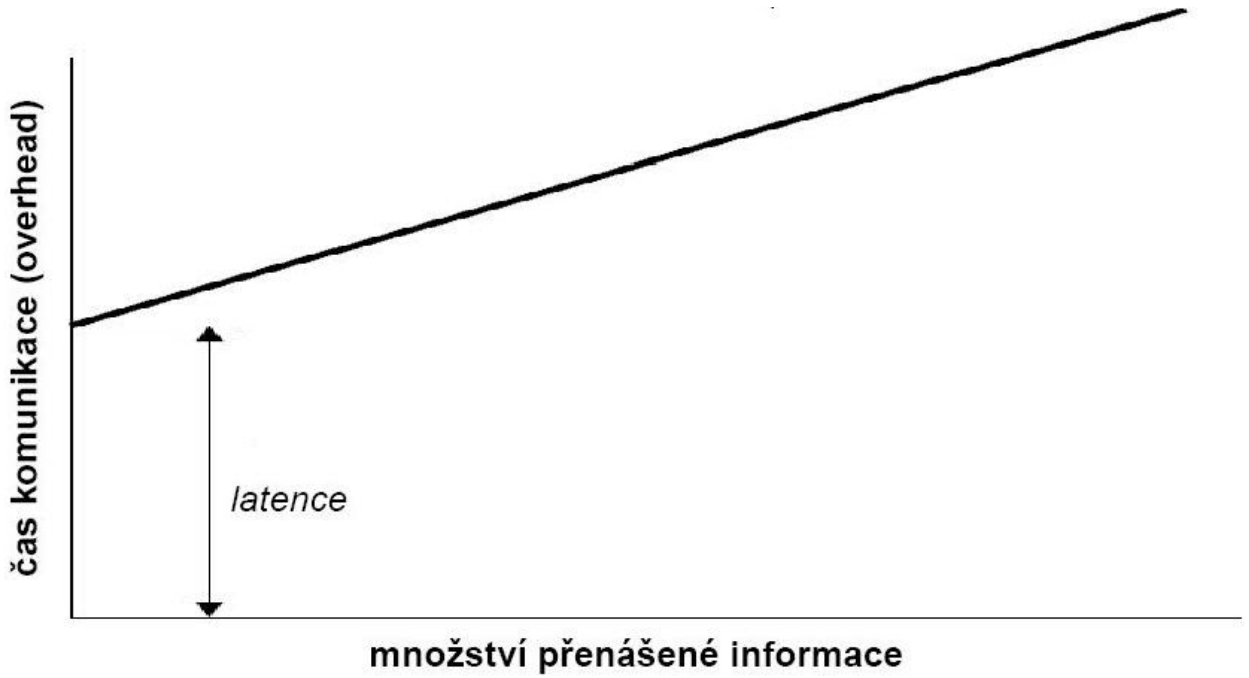
- univerzálnější, náročnější na představivost
- komunikaci a sdílení dat mezi CPU zajišťuje switch
- zajistit a pracovat s předáváním informací (message passing)
- např. MPI (Message Passing Interface)



Závislost paralelizace na architektuře systému (3)

Distribuovaná paměť' (distributed memory)

- komunikace = časová ztráta → snížení zrychlení výpočtu
- latence - časová prodleva při komunikaci mezi CPU a switch
- snaha o minimální přenos informací a komunikaci



Standard OpenMP

- aplikační programové rozhraní (API) pro Fortran nebo C/C++
- programování paralelních počítačů se sdílenou pamětí
- jednoduchost, přenositelnost, rozšiřitelnost
- seznam direktiv, knihovních funkcí a proměnných prostředí
- sekvenční jazyk s direktivami pro kompilátor $\rightarrow$ podmíněný překlad - sekvenční i paralelní zpracování
- inkrementální postup při paralelizaci programu
- vlákno OpenMP - základní stavební prvek
- začátek s jedním vláknem, které se rozvětví na větší počet vláken v paralelní sekci programu, konec opět s jedním vláknem
- komunikace mezi vlákny = zápis a čtění ve sdílené paměti $\rightarrow$ synchronizace přístupů jednotlivých vláken ke sdíleným proměnným

Standard MPI (Message Passing Interface)

- specifikace knihovních funkcí a podprogramů pro Fortran a C/C++
- paralelní programování zasíláním zpráv
- komplexnější, větší přenositelnost programů a vyšší rychlost
- první standard v komunikaci paralelních systémů
- sekvenční programovací jazyk + knihovna (externích) funkcí pro zasílání zpráv $\rightarrow$ prostředí pro paralelní počítání na clusterech
- od začátku se vyvijí paralelní program
- proces a komunikátor - základní stavební prvky
- řešení SPMD úloh
- správa a tvorba procesů - statické a dynamické
- interakce procesů - komunikace, agregace, synchronizace

Paralelní program v MPI pro Fortran

```
program par_hello_world
  use mpi
  implicit none

  integer ::  ierr, nprocs, myrank

  call MPI_INIT(ierr)

  call MPI_COMM_SIZE(MPI_COMM_WORLD, nprocs, ierr)

  call MPI_COMM_RANK(MPI_COMM_WORLD, myrank, ierr)

  print*," Hi from CPU ", myrank, " of ", nprocs

  call MPI_FINALIZE(ierr)
end program par_hello_world
```