

ARP – architektura počítačů

Jakub Nečásek (jakub.necasek@tul.cz)

podmínky pro zápočet:

- vypracování samostatné úlohy
- aktivní účast na cvičeních – 2 povolené absence

literatura – viz sylabus

dotazy...?

**Výkonnost počítačů,
Amdahlův zákon,
výkonnostní rovnice CPU**

Výkonnost počítačů

Základní požadavek kladený na počítač je schopnost provádět zpracování informací. Tuto schopnost označujeme jako **výkonnost počítače**.

Výkonnost je obtížné hodnotit jediným číslem – objektivnější je použít tzv. **vektor výkonnosti**, jehož struktura se vyvíjí.

Základem bývá **počet operací (příp. instrukcí) za sekundu**, buď v pevné nebo pohyblivé řádové čárce.

Dalšími složkami mohou být **propustnost systému**, doba odezvy, stupeň využití, aj.

Hodnocení výkonnosti by mělo být podkladem pro optimalizaci

Výkonnost a propustnost systémů

Výkonnost $P_T(T)$ – inverzní hodnota doby T provedení jednoho úkonu (programu)

$$P_T(T) = \frac{1}{T}$$

Propustnost $P_R(n, T)$ – počet n úkonů (úloh) za čas T

$$P_R(n, T) = \frac{n}{T}$$

„úkonem“ často bývá instrukce, ale problém je v různých instrukčních souborech

Metriky výkonnosti

MIPS (Million Instructions Per Second)

MOPS (Million Operations Per Second)

MFLOPS (Million FLoating point Operations Per Second)

$$P_{MIPS} = \frac{IC}{T_{CPU}} \times 10^{-6} \quad IC \dots \text{počet instrukcí}$$

$$P_{MOPS} = \frac{OC}{T_{CPU}} \times 10^{-6} \quad T_{CPU} \dots \text{doba výpočtu}$$

$$P_{MFLOPS} = \frac{OC_{FP}}{T_{CPU}} \times 10^{-6} \quad OC \dots \text{počet operací}$$

$OC_{FP} \dots \text{počet operací}$
v pohyblivé řádové čárce

užívají se také GIPS (BIPS), GFLOPS, TFLOPS, PFLOPS ...

Doba jedné instrukce

Výkonnost: $P = 1/T_p$ [*MIPS*, μs]

T_p ... čas potřebný na provedení jedné průměrné strojové instrukce

Pro zjištění T_p je třeba sestavit tabulku četnosti výskytu jednotlivých instrukcí při „běžném provozu“ počítače. Každá instrukce má přiřazenu svoji váhu a_i , která vyjadřuje pravděpodobnost výskytu instrukce v programu.

$$T_p = \frac{\sum_{i=1}^n a_i t_i}{\sum_{i=1}^n a_i} \quad [s]$$

t_i ... doba provádění i -té instrukce

a_i ... váha i -té instrukce

n ... počet instrukcí zařazených do mixu

Dhrystone, Whetstone, DMIPS

Všechny instrukce netrvají stejně dlouho => lépe určit trvání dané úlohy, kolik udělá CPU práce.

Dhrystone – test používající jen celočíselné operace;

Whetstone – test používající operace v pohyblivé řád. čárce;

DMIPS – udává, kolik umí CPU spočítat Dhrystone za sekundu (udává se v miliónech) – DMIPS nelze jednoznačně přepočítat na MIPS (závisí na tom, kolik instrukcí potřebuje daný CPU na výpočet Dhrystone (DMIPS nezávisí na architektuře – oproti MIPS));

DMIPS/MHz – udává jaký výkon má CPU při 1 MHz.

Výkonnost DSP procesorů

Signálové procesory jsou orientované na optimální provádění součtu součinů (číslicová filtrace, FFT, ...)

Výkon se často uvádí v počtu provedených akumulovaných součinů: $A = A + (B \times C)$, v jednotkách **MMAC/s** (Million Multiply Accumulates per Second) - milion násobení a mezisoučtů za sekundu.

Někdy se redukuje na **max. teoretický výkon**
= počet násobiček \times max. hod. frekvence násobičky

Další používanou jednotkou výkonnosti je **MSPS** (Million Samples Per Second)

$$\text{MSPS} = \frac{\text{max. hod. frekvence}}{\text{pocet hod. taktu na vzorek}}$$

Amdahlův zákon

Významný zákon informatiky

Popisuje výpočet výkonového zisku (zrychlení S) dosaženého vylepšením nějaké části počítače.

Zrychlení S je číslo, které udává kolikrát je rychlejší běh úlohy na počítači s vylepšením oproti běhu stejné úlohy na původním počítači.

$$S = \frac{\text{výkonnost při využití vylepšení}}{\text{výkonnost bez využití vylepšení}} = \frac{P_{NEW}}{P_{OLD}}$$

$$S = \frac{\text{doba výpočtu bez využití vylepšení}}{\text{doba výpočtu při využití vylepšení}} = \frac{T_{OLD}}{T_{NEW}}$$

Poměry F_E a S_E

Definujme si poměry:

F_E ... udává, jakou část výpočtu lze vylepšit

$$F_E = \frac{\text{původní doba výpočtu zlepšené části úlohy}}{\text{původní celková doba výpočtu}} \leq 1$$

S_E ... udává, kolikrát se zrychlil výpočet zlepšené části úlohy

$$S_E = \frac{\text{původní doba výpočtu zlepšené části úlohy}}{\text{doba výpočtu zlepšené části úlohy}} > 1$$

Doba výpočtu

Doba výpočtu na vylepšeném počítači se bude skládat z:

$(1 - F_E) T_{OLD}$ = doba výpočtu té části úlohy, kterou nelze vylepšit

$\frac{F_E}{S_E} T_{OLD}$ = doba výpočtu vylepšené části úlohy

Tedy **doba výpočtu** T_{NEW} na vylepšeném počítači je:

$$T_{NEW} = T_{OLD} \left((1 - F_E) + \frac{F_E}{S_E} \right)$$

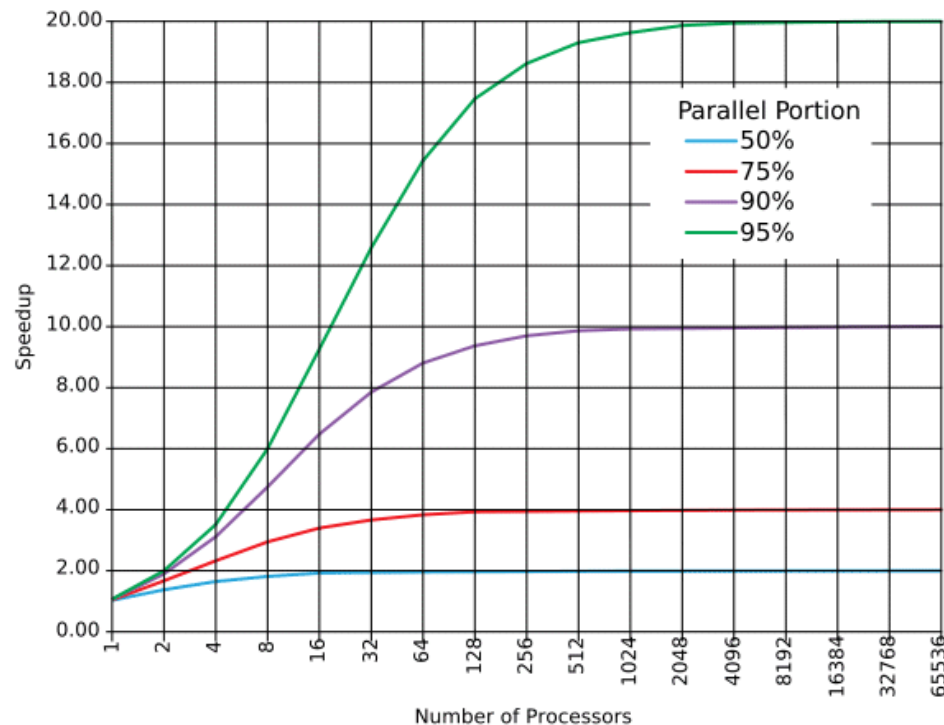
Celkové zrychlení (Amdahlův z.)

Celkové zrychlení $S_{OVERALL}$ odpovídající danému vylepšení:

$$S_{OVERALL} = \frac{T_{OLD}}{T_{NEW}} = \frac{1}{(1 - F_E) + \frac{F_E}{S_E}}$$

Zrychlení víceprocesor. systémů

Vždy je určitá část výpočtu provedena sekvenčně (pokud nelze úloha výrazně paralelizovat, víceprocesorové systémy nepřinesou zrychlení)



Amdahlův z. pro víceproc. systémy

Celkové zrychlení výpočtu multiprocessorového systému:

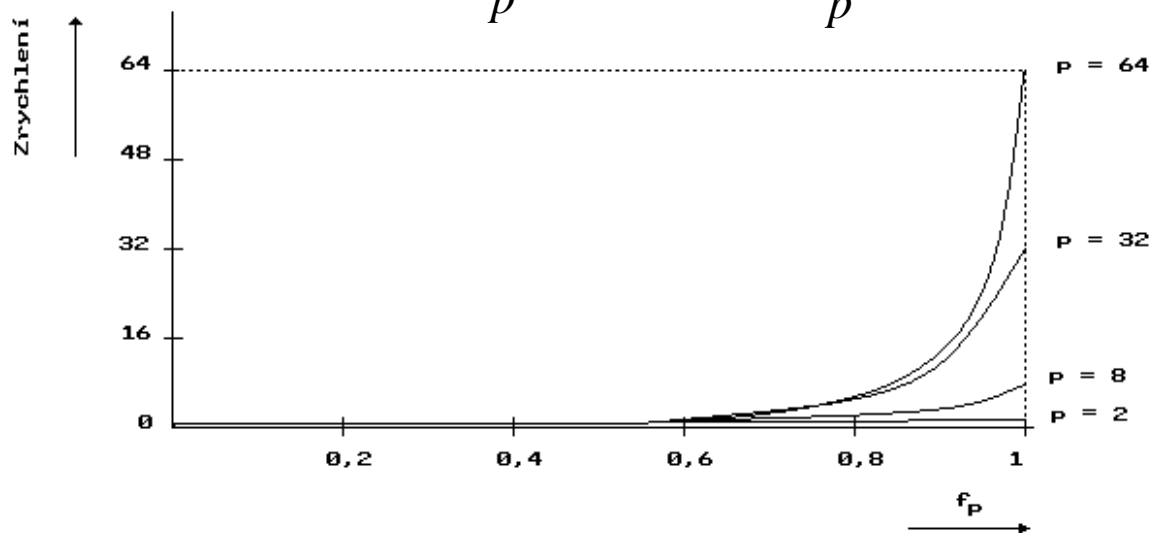
p počet procesorů

f_s část programu proveditelná jen jediným procesorem

f_p paralelizovatelná část výpočtu

$$f_s + f_p = 1$$

$$S = \frac{1}{f_s + \frac{(1-f_s)}{p}} = \frac{1}{(1-f_p) + \frac{f_p}{p}}$$



Instrukční paralelismus - zrychlení

U superskalárních architektur lze využít Amdahlův zákon pro víceprocesorové systémy, který zjednodušeně předpokládá části algoritmu, které lze/nelze paralelizovat. Zavedením stupně p_x paralelizace jednotlivých částí (zde dvou) algoritmu můžeme zobecnit:

$$S_p = \frac{1}{\frac{1-f}{p_1} + \frac{f}{p_2}}$$

Paralelismus algoritmu

Zrychlení (speedup) $S(n,p)$

poměr doby výpočtu nejlepšího známého sekvenčního algoritmu a doby výpočtu paralelního algoritmu na témže (paralelním) počítači, využíváme-li p procesorů

$$S = \frac{T_{serial}}{T_{parallel}}$$

Paralelní účinnost (efficiency) $E(n,p)$

jedná se o zrychlení dělené počtem použitých procesorů

$$E = \frac{S}{p}$$

Výkonnostní rovnice procesoru

Výkonnost CPU závisí na:

- počtu instrukcí (IC – Instruction Count)
- (průměrném) počtu taktů na instrukci (CPI – Cycles Per Instruction)
- periodě hodinového signálu (T_{clk}) – doba cyklu (taktu)

Doba provádění programu T_{CPU} je dána počtem hod. cyklů během programu násobená dobou cyklu T_{clk} (je-li konst.)

$$T_{CPU} = IC \times CPI \times T_{clk} \quad (\text{platí pro systémy bez cache})$$

$$P_{MIPS} = \frac{IC}{T_{CPU}} \times 10^{-6} = \frac{10^{-6}}{CPI \times T_{clk}} = \frac{f_{clk}}{CPI} \times 10^{-6}$$

Přesnější výkonnostní rovnice

$$T_{CPU}(prg) = \sum(IC_i \times CPI_i) \times T_{clk}$$

Součet se provádí přes všechny instrukce z architektury
instrukčního souboru

$T_{CPU}(prg)$... doba provádění programu prg procesorem

IC_i ... počet provedení instrukcí i programu prg

CPI_i ... (průměrný) počet hodinových cyklů instrukce i

Vlivy na výkonnostní rovnici

	IC	CPI	T_{clk}
Program	x		
Překladač	x	x	
ISA	x	x	x
Architektura počítače		x	x
Technologie výroby			x

ISA (Instruction Set Architecture) – architektura souboru instrukcí

Příklad 1

Počítač zpracovává program, který má 5 milionů 1-CPI (jednotaktových instrukcí), 1 milion 2-CPI a 1 milion 3-CPI. Kmitočet hodinových taktů je 100 MHz. Jaká je jeho výkonnost v MIPS?

Příklad 1

Počítač zpracovává program, který má 5 milionů 1-CPI (jednotaktových instrukcí), 1 milion 2-CPI a 1 milion 3-CPI. Kmitočet hodinových taktů je 100 MHz. Jaká je jeho výkonnost v MIPS?

$$IC = 5 \times 10^6 + 1 \times 10^6 + 1 \times 10^6 = 7 \times 10^6$$

$$N_{clk} = 5 \times 10^6 + 2 \times 10^6 + 3 \times 10^6 = 10 \times 10^6$$

$$\text{Potřebný čas: } T_{CPU} = \frac{N_{clk}}{f_{clk}} = \frac{10 \times 10^6}{100 \times 10^6} = 0.1 \text{ s}$$

$$P_{MIPS} = \frac{IC}{T_{CPU}} \times 10^{-6} = \frac{7 \times 10^6}{0.1} \times 10^{-6} = 70 \text{ MIPS}$$

Příklad 2

Výpočetní úloha je rozdělena na 4 části, z nichž každá trvá daný čas ($P1 = 11\%$, $P2 = 18\%$, $P3 = 23\%$ a $P4 = 48\%$). Jaké je celkové zrychlení, jestliže se nám část P1 nepodaří zrychlit, část P2 zrychlíme $5\times$, část P3 zrychlíme $20\times$ a část P4 zrychlíme $1,6\times$?

Příklad 2

Výpočetní úloha je rozdělena na 4 části, z nichž každá trvá daný čas ($P1 = 11\%$, $P2 = 18\%$, $P3 = 23\%$ a $P4 = 48\%$). Jaké je celkové zrychlení, jestliže se nám část P1 nepodaří zrychlit, část P2 zrychlíme $5\times$, část P3 zrychlíme $20\times$ a část P4 zrychlíme $1,6\times$?

$$T_{OLD} = 1$$

$$T_{NEW} = \frac{0,11}{1} + \frac{0,18}{5} + \frac{0,23}{20} + \frac{0,48}{1,6} = 0,4575$$

$$S_{OVERALL} = \frac{T_{OLD}}{T_{NEW}} = \frac{1}{0,11 + \frac{0,18}{5} + \frac{0,23}{20} + \frac{0,48}{1,6}} \cong 2,19$$

(v podstatě zobecněný Amdhalův zákon)

Příklad 3

Jak se zrychlí výpočet dvouprocesorového systému, jestliže 80% výpočetního algoritmu lze paralelizovat?

Příklad 3

Jak se zrychlí výpočet dvouprocesorového systému, jestliže 80% výpočetního algoritmu lze paralelizovat?

$$S = \frac{1}{(1-f_p) + \frac{f_p}{p}} \quad \text{pro } p = 2 \text{ a } f_p = 0,8$$

$$S = \frac{1}{(1 - 0,8) + \frac{0,8}{2}} = 1,67$$

Příklad 4

Předpokládejme vylepšení procesoru pro web. Nový CPU je 10× rychlejší pro webové aplikace než nynější. Dále víme, že nyní je CPU zaměstnán ze 40% výpočty a 60% času čeká na I/O operace. Jaké bude celkové zrychlení po plánovaném vylepšení?

Příklad 4

Předpokládejme vylepšení procesoru pro web. Nový CPU je 10× rychlejší pro webové aplikace než nynější. Dále víme, že nyní je CPU zaměstnán ze 40% výpočty a 60% času čeká na I/O operace. Jaké bude celkové zrychlení po plánovaném vylepšení?

$$F_E = 0,4 \quad \text{část, kterou lze zlepšit } (\leq 1)$$

$$S_E = \frac{1}{0,1} = 10 \quad \text{kolikrát se zrychlil výpočet zlepšené části}$$

$$S_{OVERALL} = \frac{1}{(1 - F_E) + \frac{F_E}{S_E}} = \frac{1}{(1 - 0,4) + \frac{0,4}{10}} = 1,56$$

Příklad 5

Předpokládejme, že při FP výpočtech v programu, operace odmocniny FPSQRT odpovídá 20% a všechny FP instrukce odpovídají 50% doby výpočtu úlohy. Úkolem je rozhodnout, zda je výhodnější 10× zrychlit provádění operace FPSQRT nebo 1,6× zrychlit provádění všech FP instrukcí.

Příklad 5

Předpokládejme, že při FP výpočtech v programu, operace odmocniny FPSQRT odpovídá 20% a všechny FP instrukce odpovídají 50% doby výpočtu úlohy. Úkolem je rozhodnout, zda je výhodnější 10× zrychlit provádění operace FPSQRT nebo 1,6× zrychlit provádění všech FP instrukcí.

$$S_{FPSQRT} = \frac{1}{(1-0,2)+\frac{0,2}{10}} = 1,22 \qquad F_E = 0,2 \qquad S_E = 10$$

$$S_{FP} = \frac{1}{(1-0,5)+\frac{0,5}{1,6}} = 1,23 \qquad F_E = 0,5 \qquad S_E = 1,6$$

⇒ výhodnější je tedy zrychlit vše i když jen „nepatrně“

Příklad 6

DSP pracuje s frekvencí 80 MHz a k provedení jedné instrukce vyžaduje 4 hodinové takty. Během jedné instrukce udělá 2 FP aritmetické operace (součet a součin), 3 přístupy do paměti (přečte 2 operandy a uloží výsledek), obnoví jeden FP registr a inkrementuje 3 adresové ukazatele (čítače). Jaký je výkon procesoru v MIPS a MOPS?

Příklad 6

DSP pracuje s frekvencí 80 MHz a k provedení jedné instrukce vyžaduje 4 hodinové takty. Během jedné instrukce udělá 2 FP aritmetické operace (součet a součin), 3 přístupy do paměti (přečte 2 operandy a uloží výsledek), obnoví jeden FP registr a inkrementuje 3 adresové ukazatele (čítače). Jaký je výkon procesoru v MIPS a MOPS?

$$P_{MIPS} = \frac{f_{clk}}{CPI} \times 10^{-6} = \frac{80}{4} = 20 \text{ MIPS}$$

2 FP operace = 40 MOPS

3 přístupy do paměti = 60 MOPS

obnova FP registru = 20 MOPS

3 inkrementace = 60 MOPS

CELKEM tedy 180 MOPS

Příklad 7

Přepokládejme, že 32% algoritmu lze paralelizovat stupněm 2, 45% algoritmu lze paralelizovat stupněm 5 a zbývající část algoritmu stupněm 6. Jaké je celkové zrychlení?

Příklad 7

Přepokládejme, že 32% algoritmu lze paralelizovat stupněm 2, 45% algoritmu lze paralelizovat stupněm 5 a zbývající část algoritmu stupněm 6. Jaké je celkové zrychlení?

$$S = \frac{1}{\frac{0,32}{2} + \frac{0,45}{5} + \frac{0,23}{6}} \cong 3,47$$

Příklad 8

Nejlepší sekvenční algoritmus se počítá na uvažovaném počítači 10 s, výpočet hodnoceného paralelního algoritmu pro 4 procesory trvá 5 s. Jaké je zrychlení a jaká je paralelní účinnost?

Příklad 8

Nejlepší sekvenční algoritmus se počítá na uvažovaném počítači 10 s, výpočet hodnoceného paralelního algoritmu pro 4 procesory trvá 5 s. Jaké je zrychlení a jaká je paralelní účinnost?

$$S = \frac{T_{OLD}}{T_{NEW}} = \frac{10}{5} = 2$$

$$E = \frac{S}{p} = \frac{2}{4} = 0,5 \Rightarrow 50\%$$