Měření výkonnosti počítačů

INP 2025

FIT VUT v Brně

Amdahlův zákon o urychlení výpočtu

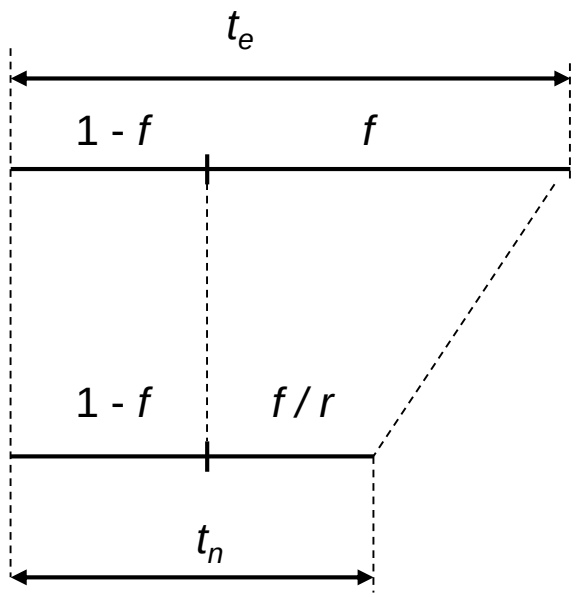
- Amdahlův zákon udává urychlení výpočtu zavedením rychlejšího zpracování jisté části úlohy. Předpokládáme, že zbylou část úlohy nelze urychlit.
- Část úlohy, kterou lze zrychlit *r*-krát, označíme *f*, *f* < 1. Zbylou část označíme 1- *f*.
- Doby provedení výpočtu před a po zavedení urychlení si označíme jako t_e a t_n.

$$t_n = t_e((1-f)+\frac{f}{r})$$

Celkové urychlení v_r je rovno

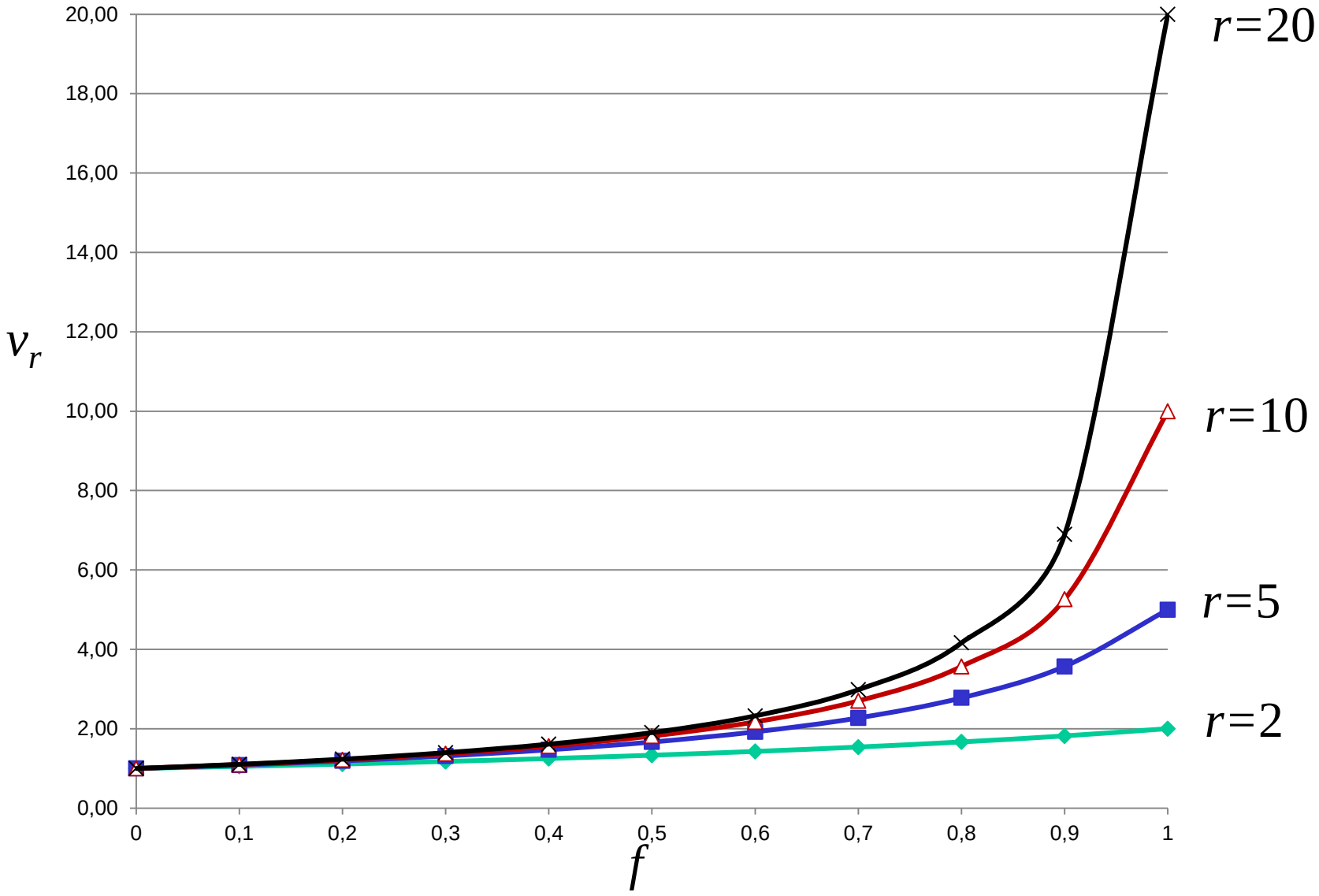
$$v_r = \frac{t_e}{t_n}$$

$$v_r = \frac{1}{(1-f)+\frac{f}{r}}$$



Zrychlení jako funkce *f* , r = konst.

Př. Jakého urychlení dosáhneme, pokud vložíme do počítače RVP, která je r-krát rychlejší než hlavní paměť a je využita v f% případů?



Definice výkonnosti počítače

- Výkonnost počítače se posuzuje několika způsoby, nejčastěji jako:
  - doba (provedení) výpočtu t_e (execution time).
  - propustnost systému P_e - počet transakcí, které je systém schopen uskutečnit za časovou jednotku.
- Absolutní měření výkonnosti je velmi problematické.
- Schůdnější je hodnotit výkonnost relativně, tj. vzhledem k referenčnímu počítači, jako např. PC XT 4,7 MHz, nebo VAX-11/780.
- Je-li výkonnější počítač označen *X* a méně výkonný *Y*, lze pro poměr jejich výkonností (propustností) P_e a dob výpočtu t_e, které byly získány pro nějaký program H, psát

$$\frac{P_{eX}}{P_{eY}} = \frac{t_{eY}}{t_{eX}} = n$$

a říkáme "počítač *X* je *n* krát výkonnější než počítač *Y*".

Počítač *X* je o *k* % výkonnější než *Y*, pokud platí:

$$(\frac{t_{eY}}{t_{eX}} \text{-} 1).100 = k$$

Měření výkonnosti

- Uvažme počítač zkonstruovaný s využitím konstantního hodinového kmitočtu f_C (doba cyklu $T_C = 1/f_C$).
- Doba výpočtu:

$$t_e = CPP \,.\, T_c$$

CPP je počet hodinových cyklů během programu

- Můžeme spočítat počet provedených instrukcí programu (I) a určit důležitý parametr procesoru - počet cyklů na provedení jedné instrukce (CPI - Cycles Per Instruction):

$$CPI = \frac{CPP}{I} \Rightarrow CPP = CPI.I$$

- Po dosazení

$$t_e = CPI.I \,.T_C$$

- Počet instrukcí v programu je dán řešenou úlohou a je ovlivněn architekturou instrukčního souboru (Instruction Set Architecture - ISA) a technologií kompilátoru. Parametr CPI je dán architekturou ISA, a T_C je dána použitou obvodovou technologií a organizací obvodů počítače.

Měření výkonnosti v jednotkách MIPS

- Alternativou k době provedení výpočtu je uvedení výkonnosti jako počet milionů instrukcí provedených za sekundu (MIPS).

$$P_{MIPS} = \frac{I}{10^6 . te} = \frac{f_c}{10^6 . CPI}$$

Pozn.: Byl použit vztah $t_e = \frac{I . CPI}{f_c}$

- Poznámky:
  - Výhodou této definice je její jednoduchost, ale její použití pro srovnávání výkonnosti počítačů nese některé problémy.
  - Např. P_{MIPS} je závislý na instrukčním souboru, takže je problematické srovnávat počítače s různými instrukčními soubory.
    - Př. Na počítači RISC i CISC trvá výpočet stejného problému 1 ms. CISC potřebuje 20 instrukcí, RISC potřebuje 60 instrukcí. Je opravdu RISC 3krát výkonnější??? Ne!
  - MIPS se hodí např. pro srovnání různých generací procesorů jednoho výrobce.

Relativní MIPS

- Řešením by mohlo být definovat výkonnost relativně vzhledem k referenčnímu počítači a pracovat s relativním R_P_{MIPS}

$$R\_P_{MIPS} = \frac{t_e^{ref} . P_{MIPS}^{ref}}{t_e}$$

- P_{MIPS}^{ref} je dohodnuté číslo. Např. počítače VAX - 11/780 bývá označen za "1 MIPS" stroj. Výkonnost v MIPS dalších procesorů je např. zde: http://en.wikipedia.org/wiki/Instructions_per_second
- Dále je vhodné definovat sadu úloh pro výpočet – např. Dhrystone.

Měření v jednotkách MFLOPS

- Pro počítače s jednotkou pro práci s čísly v pohyblivé řádové čárce (FP) reflektuje výkonnost P_{MFLOPS} počet provedených operací s pohyblivou řádovou čárkou (I_{FP}).

$$P_{MFLOPS} = \frac{I_{FP}}{10^6 . te}$$

- Problém je, že instrukční soubory FP nejsou u všech počítačů stejné.
- Složitost FP operací je různá. Operace sčítání je značně rychlejší než operace dělení. Pro program se 100% operací sčítání pak bude zjištěna vyšší výkonnost než pro program se 100% operací dělení.
- Aby parametr P_{MFLOPS} tyto rozdíly správně zachytil, byly definovány kanonické počty FP operací ve zkušebním programu. Každá operace FP se pak ohodnotí vahou, která reprezentuje její složitost, a dostaneme parametr normalizovaný P_{MFLOPS}.
- Váhy FP operací podle Livermore Loops
  - ADD, SUB, CMP, MPY 1
  - DIV, SQRT 4
  - EXP, SIN 8

Programy pro hodnocení výkonnosti

- Pro měření se používají
  - reálné programy,
  - jádra (kernels) - nejvýznamnější části skutečných programů,
  - demonstrační zkušební úlohy (např. Quicksort), a
  - syntetické zkušební úlohy - mají obdobnou filosofii jako jádra, snaží se vystihnout průměrné frekvence operací a dat na rozsáhlých množinách programů.
- V minulosti bylo nejrozšířenější použití benchmarků jako Whetstone benchmark s převahou operací v pohyblivé řádové čárce a Dhrystone benchmark s převahou operací v pevné řádové čárce.
- Dnes se často používají
  - SPEC – System Performance Evaluation Cooperative
  - TPC – Transaction Processing Performance Council

SPEC (od 1988)

- Obsahuje sadu reálných úloh SPEC (Standard Performance Evaluation Corporation): SPEC89, SPEC92, SPEC95, SPEC2000 … SPEC2017.
- Skupiny měřicích programů se neustále mění tak, aby se objektivizovalo měření a pokrývaly se moderní aplikace, např.
  - SPECint (CINT2006) pro operace s pevnou řádovou čárkou (12 programů).
  - SPECfp (CFP2006) pro operace s pohyblivou řádovou čárkou.
- Pro každou úlohu se zjistí relativní doba výpočtu (vzhledem k referenčnímu počítači). Ve skupinách INT a FP se pak odděleně vypočte tzv. geometrický průměr, což je *n*-tá odmocnina ze součinu *n* relativních dob provedení. Výsledkem jsou dva výkonové parametry SPECint a SPECfp.

SPECint 2006

Benchmark	Language	Category
400.perlbench	C	Programming Language
401.bzip2	C	Compression
403.gcc	C	C Compiler
429.mcf	C	Combinatorial Optimization
445.gobmk	C	Artificial Intelligence
456.hmmer	C	Search Gene Sequence
458.sjeng	C	Artificial Intelligence
462.libquantum	C	Physics / Quantum Computing
464.h264ref	C	Video Compression
471.omnetpp	C++	Discrete Event Simulation
473.astar	C++	Path-finding Algorithms
483.xalancbmk	C++	XML Processing

$$SPEC\,\text{int}\,2006 = \sqrt[12]{\prod_{i=1}^{12} tr_i} \qquad tr_i = \frac{t_{ref_i}}{t_{hod_i}}$$

t_{ref} – doba výpočtu referenčního počítače

t_{hod} – doba výpočtu hodnoceného počítače

Měření výkonnosti se provádí odděleně pro programy s operacemi FX a FP.

Sada testovacích úloh SPEC CPU2006

http://www.spec.org/cpu2006/results/

SPECINT 2006 (12 benchmarků)

Benchmark	Language	Application Area
400.perlbench	C	Programming Language
401.bzip2	C	Compression
403.gcc	C	C Compiler
429.mcf	C	Combinatorial Optimization
445.gobmk	C	Artificial Intelligence: Go
456.hmmer	C	Search Gene Sequence
458.sjeng	C	Artificial Intelligence: chess
462.libquantum	C	Physics / Quantum Computing
464.h264ref	C	Video Compression
471.omnetpp	C++	Discrete Event Simulation
473.astar	C++	Path-finding Algorithms
483.xalancbmk	C++	XML Processing

SPECFP 2006 (17 benchmarků)

Benchmark	Language	Application Area
410.bwaves	Fortran	Fluid Dynamics
416.gamess	Fortran	Quantum Chemistry.
433.milc	C	Physics / Quantum Chromodynamics
434.zeusmp	Fortran	Physics / CFD
435.gromacs	C, Fortran	Biochemistry / Molecular Dynamics
436.cactusADM	C, Fortran	Physics / General Relativity
437.leslie3d	Fortran	Fluid Dynamics
444.namd	C++	Biology / Molecular Dynamics
447.dealII	C++	Finite Element Analysis
450.soplex	C++	Linear Programming, Optimization
453.povray	C++	Image Ray-tracing
454.calculix	C, Fortran	Structural Mechanics
459.GemsFDTD	Fortran	Computational Electromagnetics
465.Tonto	Fortran	Quantum Chemistry
470.lbm	C	Fluid Dynamics
481.wrf	C, Fortran	Weather
482.sphinx3	C	Speech recognition

$$\mathrm{SPEC}_{\mathrm{int}2006} = \sqrt[12]{\prod_{i=1}^{12} t_{rel\,i}} \qquad t_{rel_i} = \frac{t_{ref_i}}{t_{act_i}}$$

$$\mathrm{SPEC}_{\mathrm{fp}2006} = \sqrt[17]{\prod_{i=1}^{17} t_{rel\,i}} \qquad t_{rel_i} = \frac{t_{ref_i}}{t_{act_i}}$$

Proč se používá geometrický průměr?

Výsledek testování nezávisí na volbě referenčního počítače

Mean on Normalized Numbers

Benchmark	Processor		
	R	M	Z
E	417 (1.00)	244 (0.59)	134 (0.32)
F	83 (1.00)	70 (0.84)	70 (0.85)
H	66 (1.00)	153 (2.32)	135 (2.05)
I	39,449 (1.00)	33,527 (0.85)	66,000 (1.67)
K	772 (1.00)	368 (0.48)	369 (0.45)
Arithmetic mean	(1.00)	(1.01)	(1.07)

The numbers in parentheses are normalized to Machine R.

Výsledek testu (pořadí): 1. 2. 3.

Benchmark	Processor		
	R	M	Z
E	417 (1.00)	244 (0.59)	134 (0.32)
F	83 (1.00)	70 (0.84)	70 (0.85)
H	66 (1.00)	153 (2.32)	135 (2.05)
I	39,449 (1.00)	33,527 (0.85)	66,000 (1.67)
K	772 (1.00)	368 (0.48)	369 (0.45)
Geometric mean	(1.00)	(0.86)	(0.84)

The numbers in parentheses are normalized to Machine R.

Výsledek testu (pořadí): 3. 2. 1.

Benchmark	Processor		
	R	M	Z
E	417 (1.71)	244 (1.00)	134 (0.55)
F	83 (1.19)	70 (1.00)	70 (1.00)
H	66 (0.43)	153 (1.00)	135 (0.88)
I	39,449 (1.18)	33,527 (1.00)	66,000 (1.97)
K	772 (2.10)	368 (1.00)	369 (1.00)
Arithmetic mean	(1.32)	(1.00)	(1.08)

The numbers in parentheses are normalized to Machine M.

Výsledek testu (pořadí): 3. 1. 2.

Benchmark	Processor		
	R	M	Z
E	417 (1.71)	244 (1.00)	134 (0.55)
F	83 (1.19)	70 (1.00)	70 (1.00)
H	66 (0.43)	153 (1.00)	135 (0.88)
I	39,449 (1.18)	33,527 (1.00)	66,000 (1.97)
K	772 (2.10)	368 (1.00)	369 (1.00)
Geometric mean	(1.17)	(1.00)	(0.99)

The numbers in parentheses are normalized to Machine M.

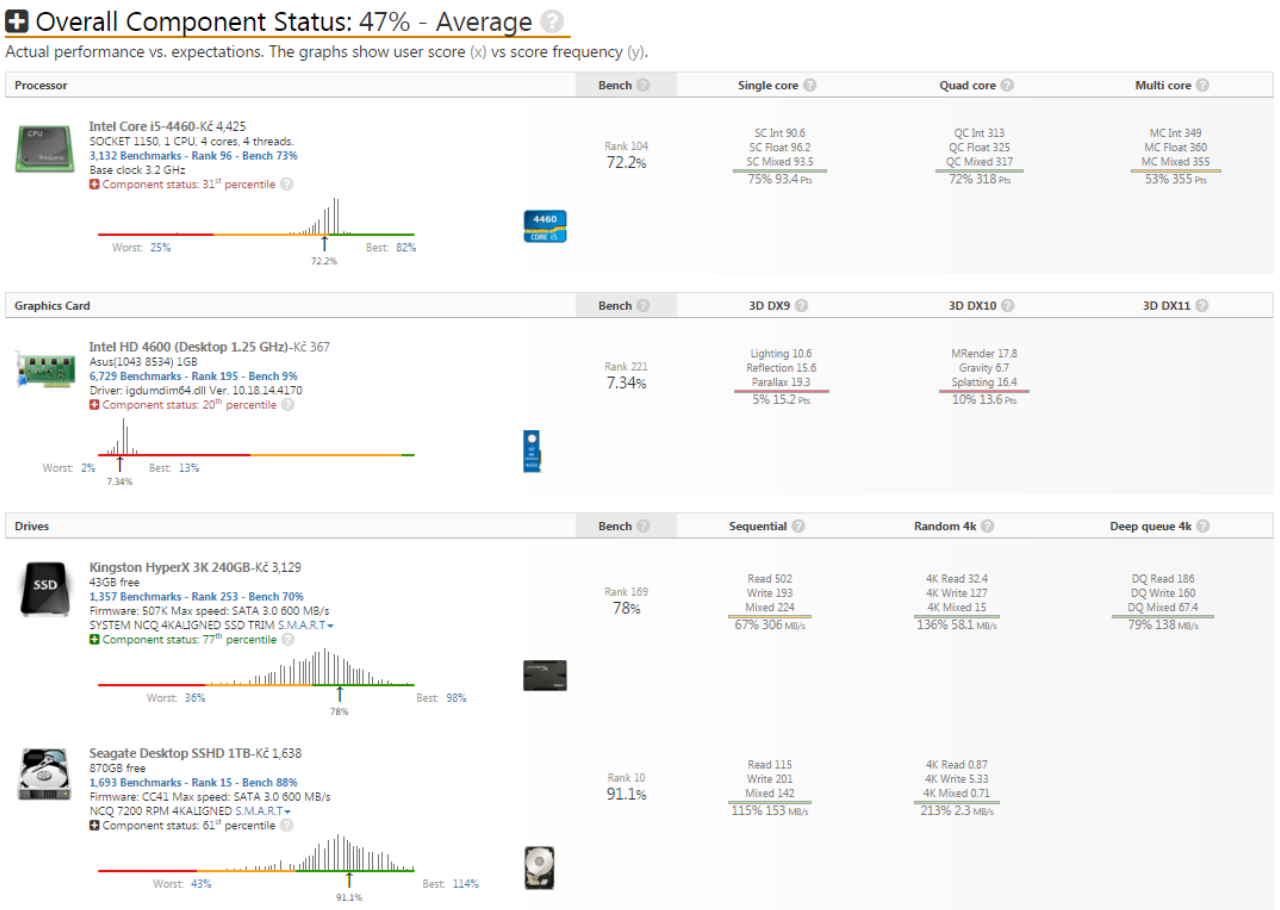
Výsledek testu (pořadí): 3. 2. 1.

Fleming, P., Wallace, J. How not to lie with statistics: the correct way to summarize benchmark results. Comm. of the ACM, 29(3),1986

Detailní výsledky testů CPU, komponent atd. jsou na webu, např.

https://www.cpubenchmark.net/

http://www.userbenchmark.com



Shrnutí

- Výkonnost závisí na optimalizaci CPU (program, kompilátor, ISA a HW), paměťového subsystému a I/O subsystému.
- Principy zrychlování popisuje Amdahlův zákon (další přístupy diskutovány v pokročilejších kurzech).
- Neexistuje jediná univerzální metrika pro hodnocení výkonnosti.
- Nejobjektivnější je použít sadu benchmarků
  - Problém aritmetický vs. geometrický průměr při sumarizaci výsledků
    - aritmetický průměr – výsledek závisí na volbě referenčního počítače; zkresluje, pokud jsou naměřená data velmi variabilní.
    - geometrický průměr - není lineární (např. 2x vyšší geometrický průměr neznamená 2x vyšší kvalitu).