

Amdahlův a Gustafsonův zákon, Karp-Flattova metrika.

Tuesday, June 4, 2013 11:58 AM

Hodně dobrý materiál přímo od intelu, který popisuje nejen Amdahlův a Gustafsonův zákon, ale i jejich důsledky a důvody, proč dnes už Amdahl nestačí:

http://software.intel.com/sites/default/files/m/7/e/2/3/0/1638-Gillespie-0053-AAD_Gustafson-Amdahl_v1_2_rh.final.pdf

Amdahlův zákon

$$E(p) = G + \frac{H}{p}$$
$$S(p) = \frac{(G+H)}{G + \frac{H}{p}}$$
$$S(p) \leq \frac{1}{f + \frac{1-f}{p}} \leq \frac{1}{f} \quad (f = \text{sekvenčně prováděná část kódu})$$

Equation 1. A general representation of Amdahl's Law

$$\text{Speedup } (N) = \frac{1}{S + \frac{1-S}{N}} - O_N$$

Where:
 N = Number of processor cores
 S = Serial percentage of workload
(expressed as a decimal in the range 0-1)
 O_N = Parallelization overhead for N threads

Předpokládá, že **velikost problému** zůstane při paralelizaci **stejná**

- Předpokládá tedy problém o konstantní velikosti
- Tzn. že s rostoucím počtem CPU se nemění rozsah úlohy
- Nevyužívá tak plně výpočetní sílu, která je k dispozici při navýšení počtu procesorů

Nelze dosáhnout perfektního urychlení, protože vždy bude nějaká část výpočtu provedena sekvenčně. G , H – čas strávený sekvenčně prováděným výpočtem. G je čas strávený vykonáváním neparalelizovatelného kódu. H je čas strávený výpočtem paralelizovatelného kódu, ale sekvenčně. Pro velké p platí $S(p) = 1 + H/G$. To by znamenalo, že perfektní urychlení je možné, pokud se lze vyhnout kódu, který nelze paralelizovat. Ale čím větší počet procesorů, tím větší bude režie jejich komunikace, takže to přece jen nepůjde (režie OS je další faktor).

Nebere v potaz load balancing, režii komunikace

- Pro problém pevné velikosti s narůstajícím počtem CPU narůstá režie, což od jistého počtu CPU vede k pomalejšímu běhu než při menším počtu CPU

Anomální urychlení.

Distribuce rozsáhlých dat u distribuované aplikace může omezit nutnost stránkovat RAM. S dostatečně rychlými komunikačními kanály pak dojde k rychlejšímu vykonávání programového kódu, protože odpadá čekání na zpomalující IO operace provádějící stránkování, včetně obsluhy příslušných přerušování. Např. paralelizované vyhledávací algoritmy mohou mít větší než lineární urychlení – lze rychleji upřesnit výběrová kritéria.

Superlineární urychlení.

Je možné, aby S vyšlo lépe než perfektní urychlení. Nejedná se o chybu ve výpočtu, zejména, pokud porovnáváme proti nejlepšímu sekvenčnímu algoritmu. Vliv na to má cache procesoru, především případ, kdy nastane mnohem častější cache hit než je obvyklé. Výpočet je pak prováděn mnohem rychleji, než když se programový kód dostává k procesoru z pomalejší paměti. Záleží na konkrétním programovém kódu, zda se ho bude dostatečné množství nalézat právě v lokální cache jednotlivých

procesorů. Analogicky to platí i pro datovou cache.

Gustavsonův zákon

Říká, že výpočty nad velkou množinou dat mohou být efektivně paralelizovány.

Změna oproti Amdahlovi - místo konstantního rozsahu **se uvažuje konstantní doba běhu**.

- Pokud jde něco paralelizovat, vyřeší se za stejnou dobu větší problémy ("vypočte se toho víc")

Pokud bude sériová část zanedbatelná a budeme mít k dispozici spoustu procesorů s distribuovanou pamětí, Amdahl nám nepomůže. $A(n)$ = čas výpočtu sériově, $b(n)$ = čas výpočtu paralelně na p procesorech.

$$a(n) + b(n) = 1 \text{ výpočet na } p \text{ CPU}$$

$$a(n) + p * b(n) = 1 \text{ počet na jednom CPU}$$

$$S = \frac{a(n)+p*b(n)}{a(n)+b(n)}$$

Equation 3. A computational representation of Gustafson's Trend

$$\text{Speedup } (N) = \frac{S + N (1 - S)}{S + (1 - S)} - O_N$$

Where:

N = Number of processor cores

S = Serial percentage of workload

(expressed as a decimal in the range 0-1)

O_N = Parallelization overhead for N threads

Karp-Flattova metrika

$$e = \frac{\frac{1}{y} - \frac{1}{p}}{1 - \frac{1}{p}} =$$

určuje podíl, kolik kódu se provedlo sériově - y (má to být $psí$) určuje urychlení na p procesorech.

- Např. z tabulky s počtem využitých procesorů (první řádka) a dosaženého urychlení (druhá řádka) lze vypočítat pro jednotlivé počty CPU sériově prováděný podíl - e
 - Pokud e zůstává stejné s přibývajícím počtem CPU - hlavním důvodem je omezená možnost paralelismu (a urychlení $psí$ narůstá čím dál méně)
 - Pokud e s přibývajícím počtem CPU roste, důvodem k slabému urychlení je režie paralelismu (buď se týká nastartování procesů, komunikace, synchronizace nebo omezení architektury)
 - Pokud e postupně klesá = ideál, čím dál větší část výpočtu se provádí paralelně