Portál pre odborné publikovanie ISSN 1338-0087

Paralelné výpočty

Hudaček Martin · Študentské práce

06.01.2010

S rozvojom informačných technológií prenikajú počítače do našich domácností v čoraz večnej miere. Dnešný svet je bez výpočtovej techniky nepredstaviteľný a je úplne normálne ak v jednej domácnosti je niekoľko počítačov, nehovoriac o školách firmách a výskumných pracoviskách. Vďaka tejto rozšírenosti už pravdepodobne každého napadlo či by sa nedalo použiť počítače, ktoré práve nič nerobia a urýchliť tak vlastný počítač.

Ušetrili by sme si takto veľa času, alebo dokonca dokázali na niekoľkých počítačoch spracovať nejakú úlohu, ktorú by náš osamotený počítač nikdy nedopočítal (napr. kvôli nedostatku pamäte). Toto je možné dosiahnuť tak že si požadovanú úlohu rozdelíme na časti a tie potom necháme spracovať na niekoľkých počítačoch. Výsledky sa odošlú na jeden lokálny počítač, ktorý ich spojí. Takéto spracovanie sa nazýva Paralelné výpočty.

S príchodom viacjadrových procesorov sa tento spôsob ešte viac rozšíril. Úlohy sa nespracúvajú na niekoľkých počítačoch, ale na niekoľkých jadrách procesora. Výhodou je že komunikácia medzi jadrami procesora je omnoho rýchlejšia ako komunikácia v sieti počítačov.

Výhody a nevýhody

Hlavnou výhodou paralelných výpočtov je to, že umožňujú využiť výkon viacerých počítačov pri riešení jedinej úlohy. V ideálnom prípade by sa úmerne so zvyšovaním počtu počítačov, zvyšovala rýchlosť výpočtu. Napríklad ak výpočet na jednom počítači trval 10 sekúnd na dvoch by trval 5 sekúnd. Lenže v realite musíme rátať aj z časom komunikácie medzi počítačmi.

- čas potrebným na zaslanie čiastkového programu, ktorý sa má spracovať
- čas na odoslanie údajov, ktoré spracúvajú čiastkové programy
- posielanie výsledkov späť na lokálny počítač

Táto komunikácia nám okrem spomalenia výpočtov aj samozrejme zaťažuje sieť. Ďalším problémom je to že medzi jednotlivými časťami úlohy sa len veľmi ťažko dajú posielať informácie. Je to preto že počítače bežia na sebe nezávisle. Ak potrebujeme nejaký medzivýsledok poslať z jednej časti do druhej tak je väčšinou najvhodnejším riešením, že si potrebný výsledok duplicitne vypočítame v oboch častiach. Na prvý pohľad sa to zdá ako veľké plytvanie, ale ak zoberieme do úvahy problémy, ktoré sú spojené z posielaním medzivýsledkov nieje to až také zlé. Takže aby sa nám výpočet

nejakého príkladu vôbec urýchlil je potrebné:

- aby bol čas čiastkového výpočtu menší, ako čas komunikácie po sieti (platí pre rozklad na dva čiastkové výpočty)
- čo najmenšia závislosť údajov v jednotlivých častiach

Príklad: Máme úlohu, pri ktorej sa spracúva veľký blok dát.

Sériovo:

spracovanie dát 10 sekúnd

Paralelne:

odoslanie dát 3 sekúnd
spracovanie dát na 1. počítači 5 sekúnd
spracovanie dát na 2. počítači 5 sekúnd
prijímanie dát 3 sekundy
celkovo 11 sekúnd

Okrem urýchlenia je výhodou paralelných výpočtov aj to že dokážeme spracovať blokom dát ktorý je väčší ako pamäť jedného počítača, kde by bolo riešenie nemožné. Ďalšou výhodou je aj dekompozícia. Ak nejaký z nekritických výpočtov zlyhá nezastaví to vykonávanie celej úlohy. Ak máme inteligentný lokálny počítač, ktorý to rozpozná jednoducho odošle čiastkovú úlohu znova.

Zrýchlenie paralelnými výpočtami

Existuje pravidlo, ktorým sa dá vypočítať možné zrýchlenie programu, ak vieme aká jeho časť sa dá rozdeliť na čiastkové úlohy. Tento vzorec však hovorí o ideálnom paralelnom výpočte na nekonečne veľa počítačoch. Čiže ide len o nejakú hranicu, ku ktorej sa môžeme len priblížiť.

Amdahlové pravidlo [1]:

$$S = \frac{1}{1-P} \tag{1}$$

P je časť programu, ktorá sa dá rozdeliť na čiastkové úlohy

Ak by celá úloha trvala 10 sekúnd a časť, ktorá by sa dala vykonávať paralelne trvá 5 sekúnd P=0,5. Podľa pravidla by bolo zrýchlenie S=2.

Viac sa môžeme dozvedieť z *Gustafsonového pravidla*, ktoré je vyhádza z Amdahlového pravidla [1]:

$$S(P) = P - \alpha(P-1) \tag{2}$$

P je počet čiastkových úloh na ktoré sme úlohu rozdelili a α je časť programu, ktorá sa dá na čiastkové úlohy rozdeliť

Príklad:

Celá úloha trvá 10 sekúnd, časť ktorá sa **nedá** vykonať paralelne trvá 5 sekúnd. Rozdelíme ju na 3 časti.

$$\alpha = 1 - \frac{5s}{10s} = 0.5$$

$$P = 3\, S(P) = P - \alpha(P-1) = 3 - 0.5(3-1) = 2$$

Úlohy, ktoré je vhodné spracovávať paralelne

Aplikácie s cyklom (cykli na sebe nezávisia):

- máme úlohu z množstvom v sérií spúšťaných cyklov – Aj keď jeden cyklus netrvá dlho, všetky spolu môžu zamestnať počítač na dlhšie ako sme ochotný čakať. V tomto prípade sa dá dosiahnuť veľké zrýchlenie pretože čas ktorý je potrebný na vykonanie množstva cyklov je omnoho väčší ako čas potrebný na odoslanie a prijatie dát.
- máme úlohu z veľmi dlho trvajúcimi cyklami – obyčajne ide v takomto prípade iba o pár desiatok cyklov avšak veľmi náročných . Opäť je tu čas vykonávania čiastkovej úlohy omnoho väčší ako čas prenosu údajov.

Aplikácie v ktorých sa dá program rozdeliť na navzájom nezávislé časti. V takomto prípade ho jednoducho rozdelíme na časti a tie potom naraz spúšťame na rôznych počítačoch.

Existujúce riešenia paralelných výpočtov

Keďže ku zrýchleniu vykonávania úloh oproti neparalelnému vykonávaniu dochádza iba pri dostatočne komplikovanom probléme, rastú oblasti aplikácie úmerne z náročnosťou úlohy. Prvý praktický príklad uvádzam z oblasti aj pre netechnickú verejnosť veľmi populárnu a to z oblasti urýchľovačov častíc. Návrh riadiaceho systému pre zaistenie presného zoskupenia lúča častíc v medzinárodnom lineárnom urýchľovači častíc. Bolo nutné dopraviť lúč častíc do interakčného bodu pričom musí byť zachovaná vysoká ostrosť lúča. Tento problém bol riešený prostredníctvom paralelných výpočtov na univerzite Queen Mary, University of London. [3]

Ďalším príkladom je projekt Ženevskej univerzity, ktorá vytvorila mnohoúčelový optimalizačný nástroj. Tento nástroj je schopný naraz spúšťať na niekoľkých počítačoch tú istú úlohu ibaže zakaždým z iného východiskového bodu. To znižuje čas výpočtu z hodín na minúty v závislosti od počtu použitých počítačov.[3]

EIM Group vyvinul kvantitatívny nástroj na spravovanie investičných fondov. Najskôr vyvinuli model na základe roku výskumu správania sa fondov. Na ňom následne spúšťali 700 sledovaných investičných fondov. Simulácia jedného fondu bola realizovaná prostredníctvom 24 000 vzoriek. Vykonávanie takejto simulácie zabralo 6 hodín. Pre urýchlenie bola simulácia rozdelená na tri 2-jadrové procesory, čo skrátilo čas na 2 hodiny. [3]

Na Massachusetts Institute of Technology bol riešený problém zlepšenia diagnostiky rakoviny identifikovaním proteínov a analýzou ich vzájomného pôsobenia. Problém sa

im podarilo vyriešiť a čas výpočtov znížili z týždňa na menej ako deň pomocou paralelných výpočtov. [3]

Inštitút Maxa Plancka vytvoril vysokokvalitný 3-D obraz proteínového komplexu. Výskumníci získavali 2-D snímky proteínu udržiavaného pri teplote tekutého dusíka. Jednotlivé proteínové komplexy boli náhodne natočené vo vnútri ľadu. Kvôli zníženiu poškodenia lúča boli použité relatívne malé dávky elektrónov. Pretože elektróny znížili kontrast a silu signálu bolo nutné odobrať milióny vzoriek na dosiahnutie požadovanej kvality. Vedci z inštitútu Maxa Plancka potrebovali vytvoriť vysoko priepustný nástroj, a procedúry schopné presne spracovať rozsiahle dáta. V sieti 32 počítačov dokázali urýchliť vypočítavanie dát 20 až 30 krát. [3]

Argonne National Laboratory (US oddelenie energetiky – DOE) – riešili problém efektívneho hodnotenia hybridných áut a áut využívajúcich palivové články. Pri vytváraní nových áut je lepšie najskôr zostrojiť počítačový imaginárny model a na ňom skúšať jeho požadované vlastnosti, ako skonštruovať drahý fyzický model auta. DOE musí často hodnotiť aj viac ako 800 automobilov pričom každé obsahuje širokú paletu technológií a rôzne meradlá výkonu ako dojazd, spotreba paliva… Rozložením práce pomocou PCT na 16 počítačov sa skrátil čas výpočtov z 2 týždňov na jeden deň. [3],[4]

Literatúra:

1. http://en.wikipedia.org/wiki/Parallel_computing
2. http://frdsa.fri.uniza.sk/~janosik/Kniha/Prudove_sprac.html
3. http://www.mathworks.ch/products/parallel-computing/userstories.html
4. http://www.transportation.anl.gov/modeling_simulation/PSAT/index.html