

OPEN SOURCE NÁSTROJE NA TVORBU PARALELNÝCH A DISTRIBUOVANÝCH APLIKÁCIÍ

GRONDŽÁK, Karol (SK)

Abstrakt. Riešenie mnohých komplexných úloh si vyžaduje veľké množstvo výpočtov. Pre ich urýchlenie boli navrhnuté mnohé riešenia, založené na paralelnom použití viacerých procesorov. V tomto článku popíšeme takéto architektúry a aj nástroje, ktoré je možné použiť na vytvorenie paralelnej, alebo distribuovanej aplikácie. Dôraz je kladený na nástroje, ktoré sú voľne dostupné.

1 Úvod

Vznik elektronických počítačov v päťdesiatych rokoch minulého storočia dal ľuďstvu k dispozícii nástroj na automatické a rýchle vykonávanie výpočtov. Prvé počítačové systémy boli nesmierne nákladné a využívali sa pre riešenie vedecko–technických problémov či už vo vojenskom alebo civilnom sektore.

Významným faktorom, ovplyvňujúcim výkonnosť počítača je počet inštrukcií, ktoré je jeho procesor schopný vykonať za jednotku času. Jednou z možností, ako zvýšiť počet inštrukcií vykonaných za jednotku času daného procesora je zvýšenie jeho taktovacej frekvencie. Takéto zvyšovanie je od istej úrovne technologicky veľmi náročné a existujú aj isté fyzikálne hranice, ktoré nie je možné prekročiť.

Iným spôsobom na zvýšenie výkonnosti procesora je zmena spôsobu organizácie spracovania inštrukcií (napr. pomocou pipeline a pod.). Na tomto princípe boli vybudované prvé *superpočítače* v šesťdesiatych rokoch minulého storočia [1, 2]. Ich nevýhodou bola vysoká cena.

S príchodom lacných, sériovo vyrábaných mikroprocesorov, sa ukázalo, že ich prepojením je možné zostaviť lacnejšie *superpočítače* s porovnateľným výkonom. Tak vznikli skupiny počítačov prepojených v rámci jednej organizácie (klastre), ktoré bolo možné využiť na vysokovýkonné počítanie.

S príchodom internetu vznikla obrovská sieť navzájom prepojených, geograficky oddelených, počítačov. To umožnilo rozšíriť myšlienku klastra na počítače, ktoré sú od seba geograficky vzdialené. Dobrovoľníci môžu poskytnúť nepoužívaný výpočtový výkon svojho počítača projektom, ktoré tento výkon môžu využiť na riešenie výpočtovo náročných úloh.

2 Klasifikácia paralelných architektúr

Základnou myšlienkou moderných paralelných a distribuovaných výpočtových systémov je paralelné spracovanie údajov. Existuje viacero spôsobov, ako to dosiahnuť. Jednotlivé spôsoby predstavujú rôzne paralelné a distribuované architektúry.

Jednou z hlavných charakteristík architektúry je mohutnosť toku inštrukcií. Ten môže byť buď jednoduchý, alebo násobný. Podobne aj tok spracúvaných údajov môže byť jednoduchý, alebo násobný. Kombináciou týchto dvoch charakteristík paralelných a distribuovaných systémov dostaneme štyri možné architektúry. Takúto klasifikáciu zaviedol Flynn [3].

Ak počítačový systém spracúva jeden tok inštrukcií a jeden tok údajov, označíme ho ako Single Instruction Single Data (SISD) architektúru. Táto architektúra v podstate ani nie je paralelná, vyskytuje sa len kvôli konzistencii klasifikácie. Do tejto kategórie patria všetky jednoprocessorové počítačové systémy.

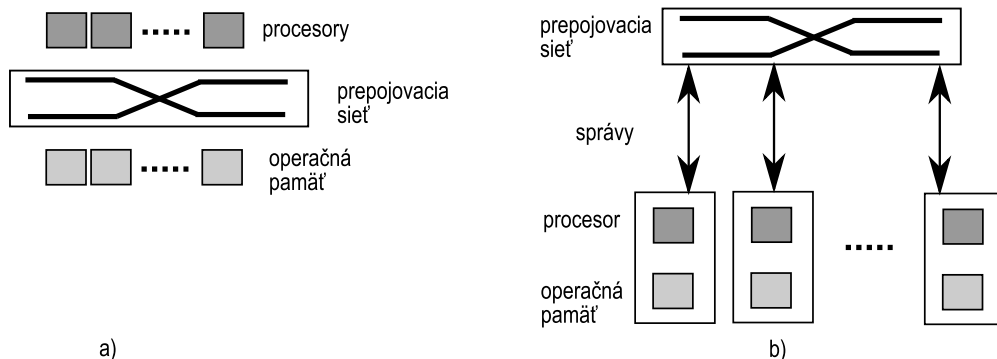
Logicky ďalšou architektúrou je Single Instruction Multiple Data architektúra. Ako už názov hovorí, ide o systémy, kedy sa v jednom čase vykonáva jedna inštrukcia, ale nad viacerými údajmi súčasne. Takýto spôsob spracovania sa označuje ako dátový paralelizmus. Je typický pre úlohy, kde sa spracúvajú veľké množstvá údajov tým istým algoritmom. Svoje uplatnenie našiel pri spracovaní obrazu a zvuku.

Moderné zobrazovacie adaptéry (GPU, graphics processing unit) a špecializované procesory na spracovanie zvuku (DSP, digital signal processor) patria do tejto kategórie. V súčasnosti výrobcovia zobrazovacích adaptérov poskytujú programové rozhranie pre sprístupnenie výpočtového výkonu GPU používateľom [4]. Do tejto kategórie možno zaradiť aj prvé superpočítače.

Kategória Multiple Instruction Single Data (MISD) nie je príliš rozšírená. Je to zrejme preto, že neexistuje relácie medzi touto architektúrou a bežne používanými programovými konštrukciami.

Poslednou architektúrou je Multiple Instruction Multiple Data (MIMD), ktorá je používaná v moderných superpočítačoch. Predstavuje skupinu procesorov, z ktorých každý vykonáva svoju činnosť na pridelených údajoch.

V mnohých prípadoch je potrebné, aby jednotlivé procesory počas výpočtu medzi sebou komunikovali. Spôsob komunikácie závisí od hardvérovej architektúry počítačového systému. V princípe najjednoduchšou je komunikácie pomocou zdieľanej pamäte. Ak procesory zdieľajú spoločnú operačnú pamäť, zapisovaním a čítaním z vopred dohodnutých adries je možné vymieňať údaje. Takéto architektúry označujeme ako systémy so zdieľanou pamäťou (obr. 1a).



Obrázok 1: Počítačový systém a) so zdieľanou pamäťou, b) s distribuovanou pamäťou

V prípade geograficky oddelených autonómnych procesorov neexistuje možnosť komunikácie pomocou zdieľanej pamäte. Každý procesor má svoju vlastnú operačnú pamäť. Na výmenu údajov musia procesory použiť komunikačnú sieť. Takéto architektúry predstavujú systémy s distribuovanou pamäťou (obr. 1b).

3 Voľne dostupné nástroje pre tvorbu paralelných a distribuovaných aplikácií

Ako bolo vyššie povedané, existujú rôzne architektúry paralelných a distribuovaných výpočtových systémov. Ich špecifiká treba brať do úvahy pri tvorbe aplikácií pre takéto systémy. Jednotlivé architektúry majú svoje špecifiká, ktoré ich predurčujú pre použitie pri riešení rôznych typov úloh.

3.1 Systémy so zdieľanou pamäťou

Ak viaceró procesorov zdieľa spoločnú operačnú pamäť, hovoríme o systéme so zdieľanou pamäťou. Typickými predstaviteľmi tejto architektúry sú viacprocesorové systémy, ktoré zdieľajú operačnú pamäť a sú riadené jedinou inštanciou operačného systému (označované ako Symmetric MultiProcessing, SMP systémy). V súčasnosti môžeme ako systémy so zdieľanou pamäťou označiť aj viacjadrové procesory.

Paralelná aplikácia môže byť realizovaná viacnásobným súčasným vykonaním toho istého programu. Komunikácia medzi procesmi je však vo väčšine operačných systémov pomerne drahá operácia (z hľadiska počtu inštrukcií potrebných na jej realizáciu). Preto moderné operačné systémy poskytujú mechanizmus vlákien, ktorý umožňuje definovať a vykonávať rôzne úlohy v rámci vykonávania jedného procesu. Podpora vlákien môže byť implementovaná na úrovni samotného operačného systému, alebo na používateľskej úrovni pomocou knižníc.

Štandard POSIX.1c, Threads extensions (IEEE Std 1003.1c-1995) definuje aplikačné programové rozhranie pre tvorbu vlákien, manipuláciu s nimi a ich synchronizáciu. Implementácie tohoto štandardu možno nájsť vo forme knižnice pthreads vo väčšine operačných systémov založených na UNIXe.

Nevýhodou knižnice pthreads je nutnosť explicitnej paralelizácie kódu. Tvorca aplikácie musí určiť časti kódu, ktoré možno vykonať paralelne a zabezpečiť paralelné vykonanie týchto častí. Pri návrhu sa tvorca môže dopustiť chýb, ktoré môžu viesť až k úplnému znefunkčneniu aplikácie (napr. nesprávnym použitím synchronizačných príkazov, a pod.). V súčasnej dobe však neexistuje plne automatický kompilátor, ktorý by dokázal bez ľudského zásahu vytvoriť paralelnú aplikáciu na základe analýzy zdrojového kódu.

Dodávatelia systémov so zdieľanou pamäťou však v snahe podporiť svoje produkty vytvorili konzorcium, ktoré špecifikovalo aplikačné programové rozhranie OpenMP [5, 6]. Toto rozhranie definuje sadu príkazov pre prekladač a knižničné funkcie, pomocou ktorých je možné vytvárať prenositeľné paralelné aplikácie. Tvorca aplikácie označí pomocou špeciálnych príkazov pre prekladač časti kódu, ktoré majú byť vykonané paralelne. Prekladač vygeneruje automaticky kód, ktorý zabezpečí paralelné vykonanie označenej časti.

Voľne dostupným prekladačom, podporujúcim štandard OpenMP je vďaka projektu GOMP [7] aj prekladač GNU GCC od verzie 4.4. Vlastné implementácie štandardu OpenMP ponúkajú vo svojich produktoch firmy IBM (XL C/C++/Fortran), Oracle, Intel, Hewlett-Packard, Microsoft (Visual Studio 2008 C++) a Cray (Cray C/C++/Fortran) [6].

3.2 Systémy s distribuovanou pamäťou

Moderné superpočítače sú tvorené množstvom procesorov, z ktorých má každý svoju lokálnu operačnú pamäť. Pre prístup do operačnej pamäte iného procesora je potrebné vytvoriť špeciálny mechanizmus. Jednou z možností je komunikácia pomocou zasielania správ. Tento mechanizmus zvolilo konzorcium dodávateľov superpočítačov, univerzít a výskumných laboratórií, ktorí vytvorili špecifikáciu Message Passing Interface (MPI). Je to špecifikácia komunikácie medzi počítačovými systémami vytvorená s ohľadom na tvorbu distribuovaných aplikácií. Umožňuje zasielanie správ medzi dvomi, resp. viacerými procesormi, pričom poskytuje aj nástroje na synchronizáciu komunikácie. Medzi najrozšírenejšie voľne dostupné implementácie patria MPICH2 a OpenMPI.

MPICH2 je projekt vyvíjaný v Argonne National Laboratory v spolupráci s mnohými univerzitami a výskumnými centrami [8]. Vďaka tejto spolupráci je možné MPICH2 prevádzkovať na bežne dostupnom hardvéri, na špecializovaných zariadeniach pre komunikáciu (InfiniBand, Myrinet, 10Gb Ethernet, a pod.), ako aj na proprietárnych počítačových systémoch (Blue Gene, Cray, a pod.). Je šírený voľne aj so zdrojovými kódmi. Podporuje platformy Linux (IA32 a x86-64), Mac OS/X (PowerPC a Intel), Solaris (32 a 64 bitov) a Windows.

Ďalším projektom, dostupným pod BSD licenciou je OpenMPI [9]. Je vyvíjaný konzorciom, ktoré zahŕňa viaceré univerzity (TU Drážďany, TU Chemnitz, University of Houston,

University of British Columbia, University of Stuttgart, University of Tennessee, ...) ako aj komerčné firmy (CISCO, IBM, a ďalšie).

Inou cestou sa vybrali tvorcovia systému BOINC (Berkeley Open Infrastructure for Network Computing) [10]. Uvedomili si, že vďaka internetu existujú stovky tisíc navzájom prepojených počítačov, ktorých procesory nie sú vždy plne využité. Vytvorili preto platformu, ktorá umožňuje riadenie poskytovania výpočtovej kapacity procesorov. BOINC predstavuje platformu, na ktorej môžu záujemcovia o masívne paralelné výpočty vytvoriť svoju aplikáciu. Ďalej je potrebné získať dobrovoľníkov, ktorí sú ochotní poskytnúť výpočtový výkon svojich procesorov na takýto projekt.

Túto infraštruktúru využívajú mnohé známe a populárne projekty, akými sú:

- **SETI@home** – projekt hľadania mimozemského inteligentného života,
- **FreeHAL** – projekt štúdia umelej inteligencie,
- **NFS@home** – projekt riešenia rozkladu veľkých prvočísel,
- **NQueens@home** – hľadanie riešenia problému N kráľovien na šachovnici, pre $N > 19$, a mnoho iných [10].

4 Záver

Jednou z ciest, ako urýchliť výpočtovo náročné úlohy je ich paralelizácia. V závislosti od charakteru riešeného problému je potrebné zvoliť vhodnú architektúru a nástroj na vytvorenie paralelnej aplikácie.

V tomto príspevku boli stručne predstavené jednotlivé paralelné a distribuované architektúry, ako i softvérové prostriedky pre tvorbu aplikácií pre tieto systémy. Dôraz bol kladený na nástroje, ktoré sú voľne dostupné.

Pre počítačové systémy so zdieľanou pamäťou existuje špecifikácia OpenMP, ktorá definuje inštrukcie pre prekladač na generovanie paralelného kódu. Príkladom voľne dostupného kompilátora kompatibilného so špecifikáciou OpenMP je projekt GNU GCC.

Pre systémy s distribuovanou pamäťou je de facto štandardom špecifikácia Open Message Interface. Aj tu existuje niekoľko implementácií s otvoreným zdrojovým kódom, napr. MPICH2 vyvíjaný v Argonne National Laboratory a OpenMPI, vyvíjaný konzorciom univerzít, výskumných ústavov a firiem.

Literatúra

- [1] WILKINSON, B – Allen, M.: *Parallel Programming*. Upper Saddle River : Pearson Prentice Hall, 2005, ISBN 0-13-140563-2

- [2] BUYIAM R. (editor): *High Performance Cluster Computing: Programming and Applications, Vol. 2*. Upper Saddle River : Prentice Hall PTR, 1999, ISBN 0-13-013785-5
- [3] FLYNN, M. J.: *Very High Speed Computing Systems*, Proc. IEEE, Vol. 12, 1966, s. 1901–1909
- [4] KIRK, D. B. – HWU W. W.: *Programming Massively Parallel Processors*. Burlington : Elsevier, 2010, ISBN 978-0-12-381472-2
- [5] QUINN, M. J.: *Parallel Programming in C with MPI and OpenMP*. Boston : Mc Graw Hill, 2003. ISBN 007-123265-6.
- [6] OpenMP [online]. Dostupné na internete: <http://openmp.org/wp/>
- [7] Projekt GOMP [online]. Dostupné na internete: <http://gcc.gnu.org/projects/gomp/>
- [8] Projekt MPICH2 [online]. Dostupné na internete: <http://www.mcs.anl.gov/research/projects/mpich2/>
- [9] Projekt OpenMPI [online]. Dostupné na internete: <http://www.open-mpi.org/>
- [10] Projekt BOINC [online]. Dostupné na internete: <http://boinc.berkeley.edu/>

Kontaktná adresa

Karol GRONDŽÁK (Ing., PhD.),

Katedra informatiky, FRI ŽU v Žiline, Univerzitná 1
010 26 Žilina, Karol.Grondzak@fri.uniza.sk

Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach

1.–4. júla 2010, Žilina, Slovensko

Organizátori: Miloš Šrámek, Spoločnosť pre otvorené informačné technológie
Tatiana Šrámková, Katedra fyziky, FEI STU Bratislava
Michal Kaukič, Aleš Kozubík, Tomáš Majer, Žilinská univerzita
Lýdia Gábrisová, Ľubica Micháľková, Žilinská univerzita
Juraj Bednár, Digmia, Slovensko
Miloslav Ofúkaný, GeoCommunity, Slovensko
Peter Mráz, Kremnica
Slavko Fedorik, SOŠ elektrotechnická, Poprad
Peter Štrba, Spojená škola/Gymnázium M. Galandu, Turčianske Teplice
Ladislav Ševčovič, FEI, Technická univerzita v Košiciach

Editori: Michal Kaukič
Miloš Šrámek
Slavko Fedorik
Ladislav Ševčovič

Recenzenti: Mgr. Juraj Bednár
Mgr. Rudolf Blaško, PhD.
RNDr. Ján Buša, CSc.
Ing. Slavko Fedorik
Ing. Karol Grondžák, PhD.
Mgr. Michal Kaukič, CSc.
Ing. Tomáš Kliment
RNDr. Aleš Kozubík, PhD.
Mgr. Juraj Michálek
doc. RNDr. Štefan Peško, CSc.
Ing. Pavel Stříž, PhD.
RNDr. Ladislav Ševčovič
Ing. Michal Žarnay, PhD.

Vydavateľ: Spoločnosť pre otvorené informačné technológie – SOIT, Bratislava

ISBN 978-80-970457-0-8

Sadzba programom pdfT_EX Ladislav Ševčovič

Copyright © 2010 autori príspevkov. Príspevky neprešli redakčnou ani jazykovou úpravou.

Ktokoľvek má dovolenie vyhotoviť alebo distribuovať doslovný opis tohoto dokumentu alebo jeho časti akýmkoľvek médiom za predpokladu, že bude zachované oznámenie o copyrighte a o tom, že distribútor príjemcovi poskytuje povolenie na ďalšie šírenie, a to v rovnakej podobe, akú má toto oznámenie.