

ÚVOD DO GRIDOVÉHO POČÍTANIA

Pojem Grid sa prvýkrát objavil v septembri roku 1997 kedy ho Ian Foster a Carl Kesselman použili na označenie *novej infraštruktúry pre vedu 21. storočia*. Hneď v nasledujúcom roku vyšla ich publikácia: *The Grid: Blueprint for a New Computing Infrastructure*, často nazývaná aj ako *Biblia gridového počítania*. Grid je považovaný za ďalšiu evolučnú fázu vo vývoji distribuovaného počítania. Zatiaľ čo Internetové a Webové technológie umožňujú prístup k distribuovaným informáciám, Gridové technológie umožňujú prístup k výpočtovým a pamäťovým prostriedkom priestorovo distribuovaným po celom svete. Na Grid možno pozerat' ako na nejakú globálnu sieť počítačov, ktoré reprezentujú jedno obrovské výpočtové prostredie. Tým, že poskytuje bezpečné, vysoko-výkonné metódy na prístup k vzdialeným prostriedkom, umožňuje rôznym skupinám, inštitúciám a jednotlivcom, tzv. *Virtuálnym organizáciám*, tieto prostriedky spoločne používať. Pritom, pod pojmom *prostriedok* (resource) sa myslia nielen výpočtové prostriedky, ale všeobecne nejaké entity, (aj napr. znalosti, vedecké expertízy, medzinárodné archívy), ktoré sú potrebné na vykonanie operácie na danom systéme.

1. Architektúra Gridu

Veľa úsilia sa v súčasnosti venuje vytvoreniu otvorených gridových štandardov, ktoré by umožňovali, aby aplikácia vytvorená pre jeden Grid, mohla byť spustená aj na inom Grid. Špecifikácie otvorených štandardov pre Grid vyvíja komunita *Global Grid Forum* (GGF) sú definované v dokumente *Open Grid Services Architecture* (OGSA). Tento dokument špecifikuje základné komponenty Gridu, opisuje jeho cieľ a funkciu a naznačuje ako by mali jednotlivé komponenty vzájomne spolupracovať. Definuje štruktúru distribuovaného systému pre gridové aplikácie - určitú predstavu architektúry na vyššej úrovni. Jej cieľom je štandardizovať všetky služby, ktoré by sa prakticky mohli vyskytnúť v gridových aplikáciách, tým, že definuje ich rozhranie a vlastnosti. Gridovú službu (Grid Service) definuje ako štandardnú Webovú službu (Web Service) s určitými rozšírenými vlastnosťami, ktoré poskytujú bezpečnosť, riadenie životnosti, vnútorného stavu, a iné funkcie vyžadované v gridovom prostredí.

Architektúra Gridu sa často popisuje pomocou modelu presýpacích hodín, ktorý pozostáva zo série vrstiev rôznej hrúbky a šírky.

Najnižšiu úroveň tvorí fyzická infraštruktúra, ktorá zahŕňa rôzne technické zariadenia: vysoko-rýchlostné komunikačné siete, počítačové a pamäťové systémy, elektronické dátové katalógy, a rôzne typy senzorov a prístrojov, ktoré je možné pripojiť na sieť.

Nad ňou sa nachádza niekoľko softwarových vrstiev, tzv. *Middleware* úroveň, ktorá umožňuje vykonávanie vlastného gridového počítania. Middleware poskytuje metódy na overovanie identity používateľa aj zdrojov, umožňuje bezpečné vykonávanie, monitorovanie a riadenie operácií na zdieľaných prostriedkoch, a zahrňuje taktiež *kolektívne služby* ako sú sprostredkovateľské, monitorovacie, diagnostické, strategické a iné. Všeobecne povedané, úlohou middlewaru je spojiť rôznorodé výpočtové prostriedky Gridu do jedného koherentného celku. Vývojom middlewaru pre Grid sa zaoberá veľa výskumných projektov na celom svete. Prakticky skoro všetky gridové projekty sú postavené na protokoloch a službách, ktoré poskytuje *Globus Toolkit* vyvíjaný komunitou *Globus Alliance* (GA).

Najvrchnejšiu úroveň gridovej štruktúry predstavujú *používateľské aplikácie, nástroje, portály a vývojové prostredia*. Patria sem aj tzv. *Servisné služby*, ktoré poskytujú určité všeobecné manažérske funkcie.

Vývoj middlewaru a tiež aplikácií sa testuje na nejakom *testovacom Grid* (testbed), ktorého použitie je vyhradené pre určitý obmedzený okruh vedcov a vývojárov na určitú limitovanú dobu.

2. Distribuovanie komplexných výpočtových problémov

Predpokladajme, že máme nejaký výpočtový problém, ktorý je príliš zložitý na to aby sa dal zrealizovať na jednom procesore. Aké sú možnosti jeho riešenia?

Ak výpočtový problém je možné rozdeliť na viac menších pod-problémov, ktoré sa dajú vyriešiť paralelne, potom distribuovaním pod-úloh na rôzne počítačové systémy sa výpočet dá naozaj urýchliť. Takéto riešenie sa nazýva *distribuované počítanie*, pričom tento pojem sa vzťahuje na ľubovoľný systém, v ktorom nejaká spojená množina počítačov je použitá na riešenie jedného problému. V tomto zmysle gridové počítanie sa môže považovať za jeden z typov distribuovaného počítania. Existujú aj mnohé iné typy, napr. klastrové počítanie, peer-to-peer počítanie, internetové počítanie, metapočítanie.

Všeobecne, komplexné výpočtové problémy môžu byť klasifikované rôznym spôsobom. Z hľadiska informatického, ich môžeme rozdeliť buď podľa ich všeobecnej výpočtovej charakteristiky - podľa stupňa paralelizmu a granularity, alebo ich môžeme kategorizovať podľa toho čo prevláda, či matematické výpočty, manipulácie s údajmi, alebo nejaké kolaboratívne činnosti.

Vysoko-výkonné počítanie

Jedna z kategórií je granularita výpočtového problému. Ak pri riešení úlohy každý z pod-problémov je silne závislý na výsledku ostatných pod-problémov, hovoríme o tzv. *jemnozrnitých* (fine-grained) paralelných výpočtoch. Takéto počítanie sa nazýva *vysoko výkonné* (High-performance computing / Supercomputing), a je pre neho výhodné použiť veľké superpočítače, počítačový klastre alebo veľmi tesne zviazané počítačové klastre, ktoré obsahujú veľký počet identických procesorov s extrémne rýchlou, spoľahlivou komunikačnou sieťou. Pre zaujímavosť, japonský NEC *Earth Simulator* superpočítač, považovaný v civilnom sektore za jeden z najvýkonnejších počítačov na svete, dosahuje 40 teraflops/s, obsahuje 640 8-procesorových uzlov a ponúka 10 terabytov pamäti a 700 terabytov diskového priestoru. Vyjadrenie paralelizmu v jemnozrnitých výpočtových úlohách si vyžaduje pomerne sofistikované programovacie metódy, ktoré musia zabezpečiť, aby správna informácia bola pre procesory k dispozícii v správnom čase. Na vývoj aplikácií sa najčastejšie používajú dva paralelné programovacie modely - *Message Passing Interface* (MPI) a *Application Program Interface* (OpenMP). Štandardná špecifikácia MPI definuje knižnicu podprogramov, ktoré obsahujú komunikačné funkcie na prenos údajov medzi procesormi, funkcie vykonávajúce kolektívne operácie nad nejakou množinou procesorov, a mnohé iné funkcie, ktoré sa zaoberajú prenosom správ a dynamickým vytváraním nových procesov. Existuje vysoko portabilná implementácia MPI štandardu - MPICH, pre rôzne paralelné a distribuované počítačové prostredia. OpenMP model je vhodný pre multiprocessorové systémy so zdieľanou pamäťou. Pomocou špeciálnych príkazov umožňuje označiť niektoré oblasti kódu, ktoré je možné vykonávať paralelne. Oba modely podporujú paralelné programovanie v jazykoch Fortran a C/C++.

Vysoko-priepustné počítanie

Na opačnom konci granularity sa nachádzajú tzv. *hrubo-zrnité* (coarse-grained) paralelné výpočty, v ktorých každý pod-problém úlohy je relatívne nezávislý od výsledkov ostatných pod-problémov, tj. oneskorenie pri obdržaní výsledku z jedného procesora nemá vplyv na činnosť iných procesorov. Takéto typy výpočtov sa nazývajú *vysoko priepustné* (High-throughput computing) a sú ideálne pre voľne viazané počítačové siete. Na vývoj hrubo-zrnitých aplikácií existuje veľa vysoko distribuovaných programovacích modelov zahrňujúcich rôzne formy objektovo-orientovaných systémov, '@home' technológie, Webové technológie, workflow systémy, kompilátory, a iné. Existuje tiež kompletná implementácia MPI štandardu pre heterogénne gridové prostredie, MPICH-G2 .

Webové počítanie

Webové služby (Web Services) je termín, ktorý je všeobecne používaný pre označovanie úplne nového typu aplikácií. Reprezentuje platformu, na ktorej je možné budovať distribuované klient-server aplikácie kde najvyššou prioritou je schopnosť ich vzájomnej spolupráce a komunikácie. Platforma Webových služieb predstavuje evolúciu nedávnych distribuovaných technológií postavených na komponentoch, akými bolo vzdialené volanie procedúr RPC a ORPC (DCOM, Corba, Java RMI), zasielanie správ (MSMQ, MQSeries), či dokonca moderné webové aplikácie ako je Google.com. Počas minulých rokov vývojári museli pri budovaní distribuovaných aplikácií zápasiť so vzájomnou komunikáciou medzi rôznymi ORPC systémami a mechanizmami výmeny správ. Nakoniec však uvedené platformy zlyhali, pretože všetky boli príliš komplikované. Taktiež na dosiahnutie transparentnosti vyžadovali aby bol na všetkých počítačoch kde boli takéto aplikácie prevádzkované veľmi zložitá operačná rutina. Preto sa stal Web potenciálnym riešením pre tieto výzvy. Veľmi významným faktorom bolo tiež zabezpečenie vyššej miery interoperability, ktorá zaznamenala v poslednom desaťročí veľký záujem organizácií, a to hlavne v dvoch oblastiach: Enterprise Application Integration (EAI - konsolidácia podnikových aplikácií) a Business-to-Business Integration (B2Bi - integrácia medzi partnermi v e-businesse). Jednou zo základných charakteristík, ktorá odlišuje platformu Webových služieb od predchádzajúcich ORPC systémov je, že komunikácia na nižšej úrovni je primárnym prvkom aplikácií, ktorý je definovaný v rámci rôznych XML technológií. XML 1.0, ako formát na prezentáciu dát, získalo v roku 1998 odporúčanie od W3C, kedy boli XML parsery dostupné prakticky pre všetky komerčné platformy a programovacie jazyky. V súčasnosti existuje niekoľko XML technológií, ktoré umožňujú reprezentovať, popisovať a objavovať komunikačnú úroveň pre Webové služby. Z tohto pohľadu je Webová služba definovaná ako nejaký aplikačný komponent, ktorý:

- komunikuje prostredníctvom otvorených protokolov HTTP, SMTP, atď.,
- spracováva XML správy rámcované pomocou SOAP (Simple Object Access Protocol)
- popisuje svoje správy s použitím XML Schema (definuje typ a štruktúru XML dokumentov),
- ponúka rozhranie - popis koncových bodov pomocou WSDL (Web Services Description Language),
- môže byť sprostredkovaný prostredníctvom UDDI (Universal Description, Discovery, and Integration).

Dominantnosť : výpočty / údaje / spoločensvo

Aplikácie, v ktorých prevládajú matematické výpočty (computation-intensive) sú doménou pre vysoko výkonné počítanie a taktiež pre gridové počítanie. Do tejto triedy patria rôzne simulácie vedecko-technických problémov v oblasti automobilového a astronautického priemyslu, fyziky, biochemických procesov, modelovanie predpovedí počasia a rôznych prírodných katastrof, ekonomických problémov a ešte mnoho iných.

Aplikácie, v ktorých prevládajú manipulácie s údajmi (data-intensive) sú v súčasnosti hlavnou hnacou silou pre Gridy, keďže odšadiaľ prichádza obrovské množstvo vedeckých údajov. Gridy sú a v najbližšej budúcnosti budú použité na zbieranie, spracovávanie a uchovávanie dát na geograficky distribuovaných pamäťových médiách, v digitálnych knižniciach a databázach.

Kolaboratívne aplikácie (community-centric) sú zamerané hlavne na sprostredkovanie rôznych druhov spolupráce medzi jednotlivcami a skupinami ľudí. Často sú štruktúrované ako virtuálne organizácie alebo laboratória, ktoré umožňujú zdieľať výpočtové a pamäťové prostriedky, softwarové balíky, dátové archívy, a iné.

Gridové počítanie

Na prvý pohľad sa možno zdá, že Gridy sú vhodné iba pre hrubo-zrnné paralelné výpočty, ale v skutočnosti všetky náročne vedecko-technické problémy predstavujú kombináciu jemno- aj hrubo-zrnného prístupu. V takomto prípade viac nezávislých úloh, ktoré samotné sú jemno-zrnnéj povahy, môže byť distribuovaných na rôzne počítačové klastre v Gride, čo predstavuje hrubo-zrnný paralelizmus. Toto je situácia kedy sa stáva Grid enormne výkonný.

Všobecne, vykonanie nejakej aplikácie na Gride bude pozostávať z nasledujúcich krokov:

- 1) gridifikovanie – adaptovanie aplikácie na príslušné gridové prostredie,
- 2) získanie autentifikačných certifikátov,
- 3) získanie informácií o dostupných gridových prostriedkoch,
- 4) spustenie aplikácie - zaslanie požiadavky výpočtovému prostriedku, alebo plánovaču (resource broker), na vykonanie úlohy resp. prenosu údajov,
- 5) monitorovanie priebehu výpočtu resp. prenosu údajov.

Hoci Grid závisí na podriadenom hardwari, jeho jadro tvorí middleware – software, ktorý sprostredkuje a manažuje jeho skutočné použitie. Middleware musí zariadiť bezpečný prístup k zdieľaným prostriedkom pomocou autentifikačných a autorizačných mechanizmov, prideliť prostriedky optimálnym spôsobom, a zabezpečiť prenos údajov, vykonanie úlohy a monitorovanie.

Referencie a Linky

- (1) Ian Foster, Carl Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco, Calif. (1999). *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufman (Elsevier), 2003.
- (2) I. Foster, *The Grid: A New Infrastructure for 21st Century Science*. Physics Today, 55(2):42-47, 2002. <http://www.aip.org/pt/vol-55/iss-2/p42.html> .
- (3) I. Foster, C. Kesselman, S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*. *Int. J. Supercomputer Applications*, 15(3), 2001. <http://www.globus.org/research/papers/anatomy.pdf> .
- (4) I. Foster, C. Kesselman, J. Nick, S. Tuecke, *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002. <http://www.globus.org/alliance/publications/papers/ogsa.pdf> .
- (5) Message Passing Interface Forum. <http://www.mpi-forum.org/> *MPI: Message Passing Interface standard*. *Int. Journal of Supercomputing Applications*, 8(3/4), pp. 165-414, 1994. *MPI2: Message Passing Interface standard*. *Int. Journal of High Performance Computing Applications*, 12(1-2), pp. 1-299, 1998. <http://www-unix.mcs.anl.gov/mpi/> .
- (6) *The OpenMP Application program Interface*. <http://www.openmp.org/> .
- (7) N. Karonis, B. Toonen, and I. Foster: *MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface*. *Journal of Parallel and Distributed Computing (JPDC)*, Vol. 63, No. 5, pp. 551-563, May 2003. <http://www3.niu.edu/mpi/> .
- (8) GridCafé, <http://gridcafe.web.cern.ch/gridcafe/> .
- (9) Grid Computing Information Centre. <http://www.gridcomputing.com> .
- (10) Distribuovaný Computing a XML. INFOWARE 5.4.2005. http://itnews.sk/buxus_dev/generate_page.php?page_id=2645 .
- (11) World Wide Web Consortium. <http://www.w3.org> .

- (12) Web Services Interoperability Organization. <http://www.ws-i.org/> .
- (13) OASIS Consortium. <http://www.oasis-open.org/> .
- (14) Global Grid Forum. <http://www.ggf.org> .
- (15) Globus Alliance. <http://www.globus.org/> .
- (16) Globus Toolkit. <http://www.globus.org/toolkit> .
- (17) Enabling Grids for E-science. <http://www.eu-egee.org> .
- (18) Lightweight Middleware for Grid Computing (gLite). <http://glite.web.cern.ch/glite/> .