

Szuperszámítógépes teljesítmény szuperszámítógép nélkül – A BinSYS Projekt –

Kovács Attila *attila@compalg.inf.elte.hu*
Kornafeld Ádám *kadam@sztaki.hu*
Burcsi Péter *bupe@compalg.inf.elte.hu*

1. Szuperszámítógép személyi számítógépekből

Szuperszámítógép. A szó, mely mögött a mai értelemben vett számítási kapacitás felső korlátai rejtőznek, és mely sok tudományos kutatás során igényként merül fel. A csillagászati számítási kapacitásnak azonban természetesen megvan az ára. Mind előállítási, mind fenntartási költségük hatalmas méreteket ölt, aminek mellékhatásaként jelentkezik, hogy nincs belőlük túl sok. Ennek eredményeképpen a legtöbb kutatási projekt meg sem engedheti magának használatukat, vagy ha mégis, akkor azzal a képpel kell szembenéznük a kutatóknak, mint a 70-es években az első számítógépek megjelenésekor tapasztalt sorban állások.

Joggal merülhet fel a kérdés, hogy lehetséges-e a fentebb tárgyalt architektúra hátrányos tulajdonságait - ár, hozzáférhetőség - úgy mellőzni, hogy megmaradjanak annak előnyös oldalai, legfőképpen a teljesítménye. Szuperszámítógépek teljesítményét általában olyan feladatok megoldására hívják segítségül, ahol a megoldást hatalmas állapotterek bejárása jelenti, mely a jelenleg elérhető számítási teljesítménnyel évekig, de akár évtizedekig is elhúzódhat. Jellemzően egyszerű szekvenciális programok futtatásáról van szó, melyek bonyolultságukat a hatalmas bemeneti információknak köszönhetik, mely legtöbbször azonban könnyűszerrel feldarabolható diszkrét részekre. Az így nyert kis állapotterű darabkák feldolgozásával már gyengébb processzorok is emberi léptékkel mérhető időn belül végeznek, így adódik az ötlet, hogy viszonylag lassú, ugyanakkor olcsó processzorokat alkalmazva az adott algoritmust mindegyik processzoron az állapotter más-más darabkáján futtatva úgy mond párhuzamosítjuk a feldolgozást. A személyi számítógépek piacán jelenleg kapható processzorok azonban annyira lassúak, hogy azokból több milliót kell hadrendbe állítani, ahhoz, hogy felvehesék a versenyt a szuperszámítógépek világával. Könnyen látható, hogy elképzelésünk újabb akadályba ütközik, mivel személyi számítógépeket ekkora mennyiségben - gépenként maximum két darab processzorral számolva - egy helyen még elképzelni is nehéz, fantáziánkat kicsit szabadjára engedve már-már a pár évvel ezelőtt futott nagy sikerű Mátrix című mozifilm végtelen energiatermelő

mezőit juttathatja eszünkbe, nem is beszélve azok összeadó energiagigántól. Úgy tűnhet, hogy eddig körbe-körbe járkáltunk problémánk megoldásának keresgélésében, hiszen így sem árban, sem méretben nem tudunk spórolni. A probléma tehát továbbra is adott. Hogyan lehetne sok személyi számítógépet "olcsón" beszerezni és azokat egy adott probléma megoldására felhasználni. Manapság már megvalósult az a személyi számítógép néven ismertté vált paradigma, mely a 80-as években még csak álom volt. Nincs is más dolgunk, mint hogy ezeknek a személyi számítógépeknek - melyek otthonok millióiban találhatóak meg - a "felesleges" számítási teljesítményét, összeadva elérjük a kívánt célt. A szuperszámítógépet megszegyenítő számítási teljesítményt minimalizálva a szükséges költségeket. Felesleges számítási teljesítmény alatt azt értjük, hogy egy átlagos személyi számítógépet a legtöbb esetben szövegszerkesztésre és internetezésre használnak, melyek egyike sem túl processzorterhelő, így gyakorlatilag egy kis marketing tevékenységgel az otthoni felhasználó meggyőzhető, hogy számítógépe kihasználatlan teljesítményét ajánlja fel egy nemesebb cél érdekében.

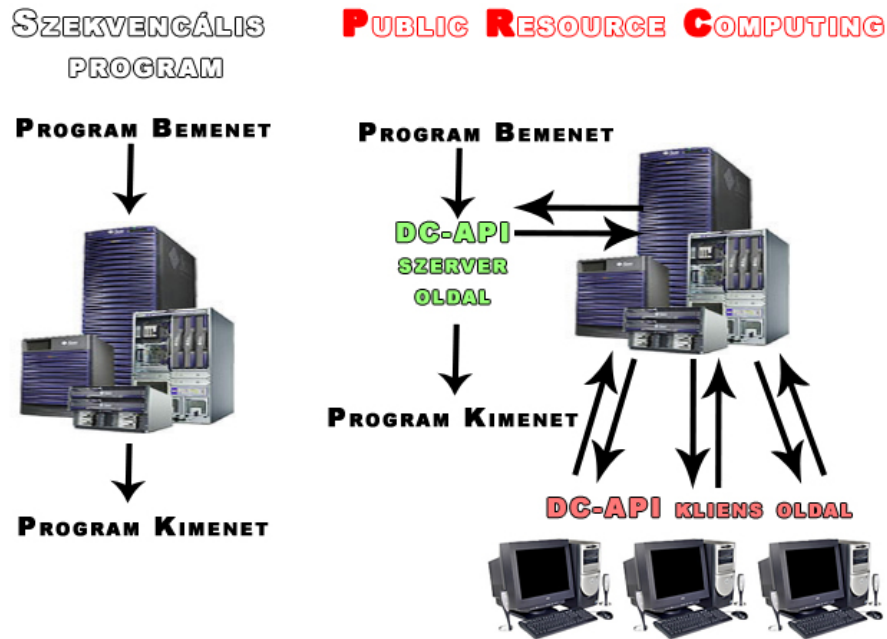
Már csak egy olyan médiumra van szükség, melyen keresztül hozzáférhetünk ezeknek az otthoni számítógépeknek a számítási kapacitásához. Itt hívjuk segítségül modern korunk legnagyobb vívmányát, az internetet. Egy elosztó szerver felállításával könnyen szétoszthatjuk a munkát személyi számítógépek között, melyek a kapott feladat eredményét visszatöltve a szerverre gyakorlatilag elértük a hatalmas feldolgozandó információ apró diszkrét részekre történő bontásával és szétosztásával a feladat egyfajta párhuzamosítását. Ezzel a gondolatmenettel jutunk el a Közösségi Erőforrás-alapú Számítási Modellhez (Public Resource Computing). A költségeket gyakorlatilag azzal minimalizáljuk, hogy a gépeket nem nekünk kell finanszírozni, hanem azok úgymond magánkézben vannak, azoknak csak a fölösleges számítási kapacitását használjuk ki. Mint az látható, egy olyan területre tévedtünk a számítástechnika világában, ahol a hangsúly a technikáról átterelődni látszik a marketingre. Itt ugyanis tényleg arról van szó, hogy magánembereket kell meggyőzni arról, hogy számítógépük úgymond szabadidejét ajánlják fel tudományos kutatásokhoz. Jól mutatja a reklám fontosságát az első, azóta széles körben elterjedt, mind a mai napig futó, PRC-modellen alapuló projekt, a SETI@Home (Search for Extraterrestrial Intelligence at home), mely egy a földönkívüli intelligencia után kutató amerikai kezdeményezés. A modell úttörőiként az amerikai University of California at Berkeley egyetem kutatói, hamar felismerték, hogy a köztudatba csak úgy lehet bevinni egy ilyen felhasználói szempontból mindenféleképpen szokatlan újdonságot, ha az kellőképpen figyelemfelkeltő hatású. És mi más keltené fel jobban az emberek figyelmét, mint a nagy dolgok. Így a SETI@Home projekt nem mást tűzött ki céljául, mint a földön kívüli intelligencia után való kutatást. A kutatáshoz szükséges információt a Cornell Egyetem által Puerto Rico-ban üzemeltetett Arecibo teleszkóp által gyűjtött napi mintegy 35 Gbyte adat szolgáltatja, melyet a projekt indulása óta csatlakozott mintegy 4 millió felhasználó személyi számítógépei dolgoznak fel. A SETI@Home gyors sikerét sok kutató megirigyelte, így hamar jelentkezett igény egy olyan általános platform kiépítésére, melynek segítségével tetszőleges kutatási témához könnyen felállítható egy

a PRC-modell alapján működő számítógép-hálózat, más néven grid. A platformot szintén a University of California at Berkeley fejlesztette ki BOINC [?], azaz Berkeley Open Infrastructure for Network Computing néven.

A platform két fő részből áll. Egy központi szerverből és a hozzá kapcsolódó kliensekből. A szerver komponensekből épül fel, ami lehetővé teszi, hogy az adott komponensek különálló hardveren futtassuk, megfelelő robusztussággal ruházva fel ezáltal a szervert. Feladata, hogy az adott projekt által szolgáltatott feldolgozandó bemenetet szétdarabolva és adatbázisba rendezve elérhetővé tegye azt a csatlakozó kliensek számára, továbbá a kliensek által feldolgozott és visszaküldött eredményeket hitelesítse és rendszerezze. A számítási kapacitásukat felajánlóknak nincs más dolguk, mint letölteni és telepíteni egy kliens programot számítógépükre, majd onnantól kezdve, hogy megadták, hogy milyen beállítások mellett engedélyezik a projekt számára a processzor kihasználását a kliens program teljes mértékben automatizáltan csatlakozva a szerverhez, letölti a projekt által feldolgozandó bemenet egy kis részletét, valamint annak feldolgozásához szükséges programot egy munkacsomag formájában. A program segítségével a kliens megkezdi a kapott bemenet feldolgozását. Amennyiben végzett a feldolgozással a kapott eredményt visszatölti a szerverre, és újabb munkát kér attól. A projekt számára ismeretlen donor számítógépére letöltött információk felvetnek bizonyos biztonsági kérdéseket. Hogyan tudjuk garantálni, hogy a donor számítógépen nem próbálják meg rosszindulatúan módosítani a feldolgozandó információt, vagy nem manipulálják-e a futtatási eredményeket? Ez ellen egyféle hatásos módszerrel védekezhetünk. Redundancia alkalmazásával. Ez azt jelenti, hogy egy adott feldolgozandó munkacsomag több, jellemzően három - de ez a szám tetszőlegesen beállítható. - donor számítógépre kerül letöltésre, és eredményük csak akkor kerül hitelesítésre, ha a különböző donorkok által szolgáltatott eredmények többsége megegyezik.

A BOINC platform megjelenése katalizátorként hatott a SETI-hez hasonlóan számításigényes kutatási projektekre, így megjelenését követően egyre másra születtek az extréménél extrémebb tudományos háttérrel rendelkező projektek, melyek kis időn belül több milliós felhasználótáborra tettek szert. A kisebb költségvetésű kutatásokat tekintve viszont mintha fordítva sült volna el a marketingfegyver. A mai napig mindössze 9 darab - a SETI@Home-ot is beleértve - nagy, a BOINC lehetőségeit kihasználó projekt fut, azt a benyomást keltve, hogy egy ilyen projekt felállítása csak akkor lehet sikeres, ha a háttérül szolgáló tudományos probléma kellően extrém ahhoz, hogy felkeltse a csatlakozók, szakszóval donorkok érdeklődését. Ezt a tévhitet megcáfolandó állította fel a Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézetének Párhuzamos és Elosztott Rendszerek laborja 2005 júliusában egy BOINC architektúráján alapuló rendszert, mely a SZTAKI Desktop Grid [?] nevet kapta. A grid felállításának célja kettős. Egyrészt a már említett tévhitet megcáfolandó olyan kis költségvetésű projektek támogatását tűztük ki célul, mellyel bemutatható, hogy egy ilyen gridhez nem szükséges dollármilliós beruházás. Másrészt szerettük volna hazánkban népszerűsíteni a BOINC rendszert, mely bár manapság a világ legtöbb részén széles körben elfogadott és támogatott, hazánkban a felajánlott számítási teljesítmény elenyésző. Itthoni népszerűsödését

elősegítendő a projekt honlapját - mely a felhasználókkal való kapcsolattartás elsődleges eszköze - elérhetővé tettük az angol mellett magyar nyelven is. Továbbá a kutatók körében való könnyebb elterjedést segítő kifejlesztettünk egy alkalmazói programozási felületet (DC-API [?]), mely megkönnyíti a BOINC rendszerre való alkalmazásfejlesztést (lásd 1. ábra). A SZTAKI Desktop Grid [?] fennállásának mintegy fél éve alatt közel háromezer donor ajánlotta fel számítógépeinek kihasználatlan kapacitását, mellyel megközelítőleg mintegy 500 GFlop/másodperces számítási teljesítményt értünk el.



[1. ábra] A PRC-modell és a hagyományos szekvenciális-modell

Az ismertetett architektúra természetesen messze nem tökéletes, mint ahogy a szuperszámítógépeknek a PRC-modellnek is vannak hátrányai, melyek közül a legkritikusabb éppen abból ered, ami a szuperszámítógépekkel szemben a legnagyobb előnyének bizonyult. Azzal ugyanis, hogy a számítási teljesítményhez szükséges hardver költségeket gyakorlatilag a csatlakozó donorok fedezik, elveszítjük a szuperszámítógépek által nyújtott a számítási teljesítményre adott minőségbiztosítást, szakmai nyelven a Quality of Service-t. Az önkéntes számítási teljesítmény felajánlásának ugyanis megvan az a nagy hátránya, hogy nem garantálható az, hogy az egyik pillanatban még effektív munkát végző donor számítógép a következő pillanatban is a projekthez lesz rendelve. Teljes mértékben a felhasználó döntésén múlik, hogy számítógépe processzora mikor és milyen körülmények között vehet részt a projektben. A donor számítógépet annak felhasználója bármikor kikapcsolva, vagy a kliens programot leállítva az

adott donor számítási kapacitására a projekt tovább nem számíthat egészen addig, amíg a kliens programot újra el nem indítják. Emiatt a donoroknak az adott projektben való részvétele egy sztochasztikus folyamattal modellezhető.

A másik felmerülő probléma a redundáns számításból ered. A kliens program ugyanis elérhető számos különböző platformra, mely platformok különbözően értelmezhetnek bizonyos dolgokat. Ezek közül talán a legfontosabb a lebegő pontos számok kezelésének gyakori eltérése. A SZTAKI Desktop Grid üzemeltetése során szembekerültünk azzal a problémával, hogy linuxon és windowson a kliensek által szolgáltatott eredménynek kerekítés miatt eltértek, annak ellenére, hogy a két operációs rendszert kiszolgáló architektúra mindkét esetben az Intel x86-os volt. További problémákat vehet fel, ha nem csak szoftverben, de hardverben is eltér a kliens program futtatási környezete. A probléma kivédéséhez az eredmények szerver oldali hitelesítésénél meg kell engedni az eredmények bizonyos hibahatáron belüli eltérését. Ennek a hibahatárnak a megállapítása azonban teljes mértékben az adott tudományos kutatás sajátossága, így előre nem definiálható.

A fentebb leírtakból látható, hogy a szuperszámítógépek egekbe szökő költségeit nem spórolhatjuk meg büntetlenül. Ismét felmerül a kérdés. Ki lehet-e kerülni a fentebb leírt problémákat? Lehet-e PRC-moddal Quality of Service szolgáltatást nyújtani? Elkerülhető-e a redundancia alkalmazása? A felmerülő kérdésekre a válasz igen, ehhez azonban szükség van a PRC-modell módosítására. A SZTAKI Desktop Grid jelenlegi formájában a hagyományos modellt követi, elsődleges célja a BOINC architektúra magyarországi elterjesztése, annak beolvasztása a hétköznapi tudományos kutatások világába. Tapasztalatunk szerint azonban a kutatók szkeptikusan állnak a BOINC architektúrához a fentebb felmerült kérdések miatt. Mindhárom felmerült kérdés megkerüléséhez a publicitás korlátozása a kulcs. Nem kell ugyanis redundanciát alkalmazni abban az esetben, ha a donorok megbízhatóak. A kutatók szemszögéből nézve továbbá Quality of Service szolgáltatás nyújtható, ha a kutató saját maga rendelkezik a donorként használt számítógépek felett. Ha minden olyan kutatólabor felállít egy saját lokális desktop gridet, melynek szüksége van nagy számítási teljesítményre, azzal a laborokban rendelkezésre álló hardvererőforrások korlátozott kapacitása miatt behatároljuk ugyan az adott kutatáshoz hozzárendelhető számítási kapacitást, ugyanakkor feleslegessé válik a redundancia alkalmazása és Quality of Service szolgáltatást is nyerünk, mivel a donor számítógépek megbízhatóak. Ezzel gyakorlatilag létrehozhatunk sok kis korlátozott számítási kapacitással rendelkező - pár száz processzorral párhuzamos kutatást végző - kutatási szívetet, úgymond Lokális Desktop Gridet, melyek gyakorlatilag mini szuperszámítógépeknek tekinthetőek, minimális beruházási költségek mellett. Kevés csatlakozó donor kihasználásához ugyanis a szerver szerepét teljes mértékben el tudja látni egy közönséges asztali számítógép. Ugyanakkor minden kutatási projekt futása során előfordul, hogy a számítási kapacitás kihasználatlanul áll, így lehetőség nyílik arra, hogy a tétlen lokális desktop gridek felajánlják kihasználatlan számítási kapacitásukat más Lokális Desktop Gridek számára. Azáltal, hogy egyik kutatási projekt számítógépei egy másik számára végeznek munkát felmerül a biztonság kérdése. Ha jobban belegondolunk azonban ebben az

esetben, a hagyományos PRC-modellel szemben, már nyomon követhető, hogy mely projektek lokális desktop gridjei vettek részt egy adott kutatásban, továbbá minden lokális desktop gridnek érdekében áll, hogy minél több donorhoz férjen hozzá, ezáltal minél több számítási kapacitásra szert téve. A Lokális Desktop Grideket megfelelő hierarchiába szervezve megoldható a számítási kapacitás biztonságos és optimális elosztása. A közeljövőben szeretnénk elérni, hogy minél több kutatási projekt éljen a szuperszámítógépes teljesítménnyel anélkül, hogy drága szuperszámítógépekre kelljen pénzt áldoznia.

2. A SZTAKI Desktop Grid BinSYS projektje [?]

Az MTA SZTAKI Desktop Gridjén 2005 közepe óta fut a BinSYS nevű projekt, melynek célja általánosított bináris számrendszerek [?] keresése. Az általánosított számrendszerek matematikai vizsgálata hosszabb ideje folyik az ELTE Informatika Karának Komputer Algebra Tanszékén. Az ELTE kutatói először saját laborjukban futtatták a probléma számítógépes vizsgálatára készített programjukat, ám szembesülve az egyre növekvő számítási igénnyel, más megoldás után néztek. A tudományos problémához grides környezetet kereső ELTE és a grides környezethez tudományos problémát kereső SZTAKI szerencsés egymásra találásából gyümölcsöző együttműködés született. A SZTAKI Desktop Gridjére került az alkalmazás, melyet a két intézmény kutatói közösen tettek alkalmassá a grides futtatásra, kihasználva a probléma párhuzamosítható szerkezetét.

Alább rövid, és komolyabb matematikai ismeretek nélkül is érthető formában vázoljuk a probléma hátterét, majd ismertetjük a kapott eredményeket – a matematikai vonatkozásokat éppen csak érintve.

2.1. Általánosított számrendszerek

A hagyományos számrendszerfogalom segítségével nemnegatív egész számokra véges felírást adhatunk a következő alakban (n egynél nagyobb egész):

$$z = \sum_j d_j n^j,$$

ahol a d_j számjegy a $0, 1, \dots, n-1$ számok közül választható. Ezzel a felírással csak nemnegatív számokat állíthatunk elő, a negatívokhoz előjelre van szükség. Ez elkerülhető, ha megengedjük, hogy n negatív egész is lehessen, pl. minden egész szám felírható véges összegként

$$z = \sum_j d_j (-2)^j$$

alakban, ahol d_j vagy 0 vagy 1. Ez az előjel nélküli változat általánosítható a racionális számoknál bővebb ún. számtestek egész számaira is. Például az $x + yi$ alakú komplex számok (x, y egész), vagyis a Gauss-egészek esetében

választhatjuk alapszámnak a $(-1 + i)$ számot, és ekkor minden Gauss-egész felírható

$$z = \sum_j d_j (-1 + i)^j$$

alakban. A legáltalánosabb változatban lineáris algebrai eszközökkel definiálható a számrendszer fogalma. Ilyenkor alapszám helyett alapmátrixról beszélünk, a számjegyek pedig vektorok. Akkor mondjuk, hogy az alapmátrix a jegyvektorokkal általánosított számrendszert alkot, ha a tér minden vektorára egyértelműen létezik a fentiekhez hasonló alakú felírás. Ha a továbbiakban csak dimenziót mondunk, mindig a jegyek dimenzióját értjük rajta. Egy számrendszert jellemez tehát a dimenziója, és a jegyek száma. Ha két jegy van, azt mondjuk, hogy a számrendszer bináris.

A kutatás célja általánosított számrendszerek keresése. Mivel ezekből végtelen sok van, teljes keresést nem lehet megvalósítani. A korábbi tapasztalatok és futási eredmények alapján az összes legfeljebb 11 dimenziós bináris számrendszer megkeresése reális célnak tűnt. A 2005-ös év végére már eredmények is születtek.

2.2. A BinSYS projekt eddigi eredményei

A számrendszerek keresése két lépcsőben történik: először egy hatalmas állapottér bejárásával megkeressük adott dimenzióban az összes ún. expanzív polinomot, majd a kapott néhány száz polinomból kikeressük azokat, melyekből számrendszer alkotható. A második lépés kevésbé számításigényes, így azt nem a Desktop Griden hajtottuk végre. A Desktop Griden futó program kimenete tehát egy százas nagyságrendű lista, mely polinomokat tartalmaz. A futtatás eredményeképpen 11 és annál kisebb dimenziókban rendelkezésünkre áll ez a teljes lista.

A 11 dimenziós esetben a program összesen 550 polinomot adott meg kimenetként. A program megírásánál a sebesség optimalizálása érdekében kellett engednünk, hogy a kimenetében olyan számsorozatot is megadhasson, amely nem expanzív polinomnak felel meg. Esetünkben a kimenet ellenőrzése után 338 polinom bizonyult valóban expanzívnek. (A többi polinom egy alacsonyabb fokú expanzív polinom és körosztási polinomok szorzata. Ezek is érdekes esetek, ezért nem került bele a kódba még egy ellenőrző rutin, mely kiszűri az ilyen polinomokat.)

A 338 polinom vizsgálatánál várakozásainknak megfelelően kevés, mindössze 11 darab olyan polinomot találtunk, melyek számrendszert alkotnak. Az alábbi táblázatban az alacsonyabb dimenziós expanzív polinomok és számrendszerek számát is felsoroljuk.

Dimenzió	2	3	4	5	6	7	8	9	10	11
Expanzív	5	7	29	29	105	95	309	192	619	338
Számrendszer	4	4	12	7	25	12	20	12	40	11

A kapott eredmények matematikai kiértékelésével és gyakorlati alkalmazásainak vizsgálatával jelenleg is foglalkozunk. Addig is a SZTAKI Desktop Griden a BinSYS projekt módosított változata fut, illetve készül a GenSYS projekt, mely a nem bináris általánosított számrendszereket vizsgálja. Reményeink szerint azonban ez csak egy lesz a számos újonnan csatlakozó tudományos projekt közül.

Hivatkozások

- [1] David P. Anderson: BOINC: A System for Public-Resource Computing and Storage *5th IEEE/ACM International Workshop on Grid Computing, November 8, 2004, Pittsburgh, USA.*
- [2] <http://szdg.lpds.sztaki.hu/szdg>
- [3] Podhorszki Norbert, Vida Gábor: Alkalmazói programozási felület SETI-jellegű elosztott programokhoz és végrehajtó rendszer a BOINC infrastruktúrára *Networkshop 2005, Szeged. NIIF, 2005.*
- [4] <http://www.lpds.sztaki.hu/desktopgrid>
- [5] <http://compalg.inf.elte.hu/projects/binsys/>
- [6] Kovács Attila: Generalized binary number systems. *Annales Univ. Sci. Budapest, Sect. Comp. 20, 2001, 195-206.*