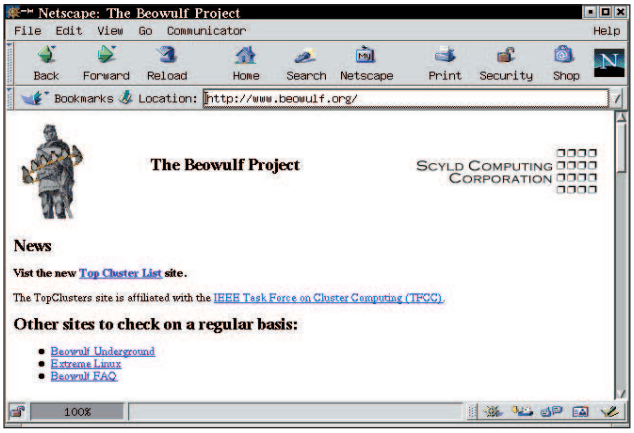
A Beowulf-tudatállapot

Mindenki találkozik OSCAR-ral, Scyld pedig rendszergazda lesz, és csodálatos géptelepeket épít.

Szeretnél linuxos géptelepet építeni? Mostanában sokan így vannak ezzel. Nem mindenki érti azonban ugyanazt a linuxos géptelep fogalmán. Ha megkérdeznénk, sokan azt válaszolnák, hogy a linuxos géptelep egyenlő a magas rendelkezésre állóságot felmutató, hibatűrő, redundáns és terheléselosztást megvalósító rendszerekkel, amelyeket az e-kereskedelmi webhelyeknél vagy az alkalmazáskiszolgálókban használnak. Mások a linuxos géptelepen a párhuzamos számításokat végző hatalmas teljesítményű Beowulf-telepeket értik. Mindannyiuknak igaza van. Ha úgy gondolod, hogy ez zavaros, próbálj csak egy kicsit elgondolkodni a bioinformatika fogalmán. Pillanatnyilag a linuxos géptelepek és a bioinformatika a leggyorsabban fejlődő technikák közé tartoznak, nem csoda, hogy fogalmaik ilyen bizonytalanok. Tisztában vagyok ezzel, hiszen hivatásszerűen foglalkozom bioinformatikával és linuxos géptelepekkel, bár e két téma kalandnak sem utolsó. Ennyit rólam, most lássuk a Beowulfot! A Beowulf ötlete ***Donald Becker*** és ***Thomes Sterling*** agyából pattant ki 1994-ben, amikor külsős munkatársként NASA-nál dolgoztak. Az ötlet hamar a nyílt forráson alapuló elosztott számítások szabványává nőtte ki magát. A recept mindössze annyi, hogy vegyünk néhány mindennapi számítógépet, kössük össze őket, telepítsünk rájuk nyílt forrású programokat, és kész is a virtuális szuperszámítógép. „Mihez kezdhetnék egy saját szuperszámítógéppel?” – kérdezheted joggal. Gyakorlatilag bármit: például MP3-zenéket gyárthatsz, vagy megfejtheted az emberi DNS-t. Ez a két alkalmazási terület kissé eltérően veszi hasznát a több processzornak. Ha a Beowulfon MP3-at gyártasz, az MP3-gyártást lényegében az egész telepen szétosztod, például minden processzorra jut egy MP3. Így nagyszámú soros feladatot párhuzamosítasz oly módon, hogy egyszerre indítod el őket, mindegyiket külön processzoron. Minden egyes MP3-feladat független a többitől, futása közben nem kell adatot cserélnie a többi MP3-folyamattal. Elvileg ezer (azonos hosszúságú) MP3 legyártása ezer (azonos sebességű) processzoron annyi ideig tart, mint amennyi ideig egyetlen MP3 elkészítése egy processzoron. Ez történne egy „tökéletes világban”. Valójában nem tapasztalnánk soros sebességnövekedést, ha ezer MP3-feladatot indítanánk egyszerre, mert ezer processzor esetén jelentős hálózati késleltetéssel és sávszélesség-korlátokkal kellene szembenéznünk.

A legtöbb Linuxon futtatható program kis erőfeszítéssel futtatható a Beowulfon a fenti MP3-as példához hasonló módon, így számítási teljesítménye nagymértékben növelhető. A legtöbb kísérleti adathalmaz viszont – mint például az időjárás-előrejelzési adatok vagy a DNS-szakaszok – nem bonthatók fel az MP3-gyártás módján, mert az adathalmaz egyik részén végzett számítások eredménye befolyásolja az adathalmaz más részein végzendő számításokat. Az ilyen esetben végzendő számítás ahhoz hasonló, mint amikor egy óriási MP3-fájlt kell készíteni több processzor igénybevételével. Könnyű belátni, hogy ha egyetlen MP3 elkészítésének feladatát több processzorra bízzuk, a processzoroknak a munka egyeztetéséhez beszélgetniük kell egymással. Az ilyen programok erre a célra üzenetküldő programkönyvtárakat használnak, így a program által elvégzett számításról a többi processzoron futó programrész is értesül.

Két elterjedt párhuzamos programozási modell létezik: az egyik az üzenetküldő felület (MPI) programkönyvtárat használja, a másik a PVM, a párhuzamos virtuális gép. Az elvek egyszerűek, de a gyakorlatban nehéz őket hatékonyan megvalósítani. A párhuzamos programozás összetett feladat. A nagyteljesítményű számítások önmagukban is épp elég bonyolultak és kiábrándítóak, hát még, ha kódunkat drága, szeszélyes és egyedi nagyszámítógépeken kell futtatnunk! Régen a nagyteljesítményű számítások végrehajtása csak nehezen kezelhető,



szabadalmakkal védett gépeken volt elképzelhető, de ezek a megoldások szerencsére lassanként kihalnak. Egyre többen használnak a sarki számítógépboltban is beszerezhető alkatrészekből összeszerelt gépeket Linuxszal, melyekből Beowulf-géptelepet építenek. A Beowulf ár-teljesítmény aránya verhetetlen, ráadásul a nyílt és szabványos felületen a programozás szórakozásnak sem utolsó.

Mindez jól hangzik, azonban a kezdeti időkben a Beowulf – akárcsak a többi nagyteljesítményű számítási módszer – minden volt, csak nem egyszerű, leginkább a boszorkánysághoz volt hasonlatos. A Beowulf-telep létrehozása külön programok, programkönyvtárak és segédeszközök letöltését és telepítését igényelte minden egyes Linux munkaállomásra, amelyek általában vegyes hálózatra csatlakoztak. Minden Beowulf egyedi volt, akár a gépek összekötését, akár a rajtuk futó programokat nézzük, és állandó változás jellemezte őket. A géptelep kezelése és fenntartása a helyi gépek és programok mély ismeretét igényelte. Sok gond akadt akkoriban, de akárcsak a többi nyílt forrású sikertörténetnél, a közösség megtalálta a megoldást. 1994 óta a közösség és a vállalati partnerek jóvoltából a Beowulf sokat fejlődött, és megjelent a Beowulf-programterjesztések második nemzedéke. Igen, terjesztésekről van szó, amelyek CD-n jelennek meg. Nincs szükség többé az Internet

legtávolabbi zugaiból összeszedegetni a Beowulf telepítéséhez szükséges eszközöket, programokat és meghajtókat. Túl jól hangzik ahhoz, hogy igaz legyen? Olvass tovább, mert a különböző Beowulf-terjesztéseket, beszerzési helyüket és a telepítésüket is be fogom mutatni.

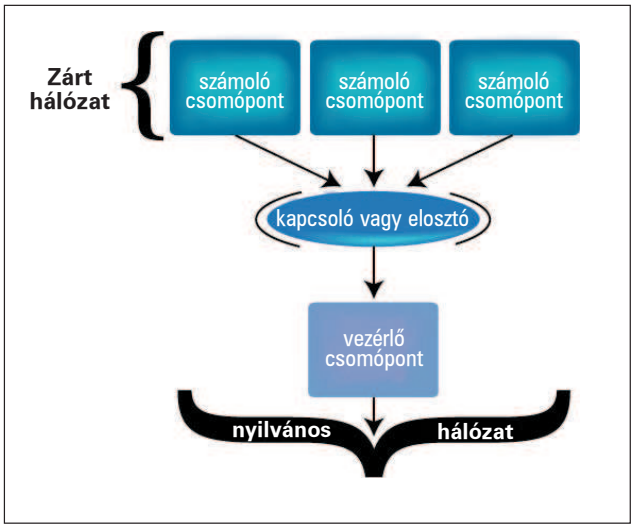
A vas szintjén minden Beowulf rendelkezik néhány olyan közös tulajdonsággal, ami megkülönbözteti néhány munkaállomás általános hálózatba kapcsolásától. A megszokott hálózattal szemben nem minden Beowulf-csomópont egyenlő, „osztálytársadalomba" rendeződnek. Minden Beowulfban egy vezérlő csomópont, és sok számoló csomópont helyezkedik el. A vezérlő csomópont a Beowulf vezérlőközpontja, ez futtatja azokat a démonokat, amelyek a számítógépek párbeszédéhez szükségesek. A vezérlő csomópont az a hely, ahonnan a rendszer agya (ez te vagy) programjaik telepítése és beállítása által kormányozza a számoló csomópontokat. A fájlrendszerek itt csatolhatók, a feladatok végrehajtása megfigyelhető, az erőforrások itt oszthatók el, és szintén itt teremthető kapcsolat a külvilággal.

A számoló csomópontok a vezérlő csomópont számítási parancsait hajtják végre, és létezésükről tájékoztatják a vezérlőt. A számoló csomópont lehet nagyon buta (kevés kódot használó), de viszonylag okos is (teljes Linux-telepítés futhat rajta). Azonban még a teljes Linux-telepítést hordozó számoló csomópontokból is hiányzik néhány olyan tulajdonság, amelyet a vezérlő csomópont nyújt, így biztosítva uralmát. Például az NFS-en keresztül a csomópontokra befűzött fájlrendszerek (mint például a felhasználók saját könyvtárai) a vezérlő csomóponton helyezkednek el. Mivel minden itt tárgyalandó Beowulf-terjesztés alapértelmezetten ezt a megközelítést alkalmazza, az egyszerűség kedvéért a cikkben felépített géptelep is ezt a megközelítést fogja utánozni. Ne feledjük azonban, hogy a valóságban bizonyos I/O-szolgáltatások az egész telepre kiterjedhetnek, a fájlok írása és olvasása több helyen is történhet, ha az adatfolyam kiegyenlítése ezt kívánja. Kezdetnek viszont talán az a legegyszerűbb, ha a vezérlő csomópontot bízzuk meg a géptelep szolgáltatásainak futtatásával és a számoló csomópontoknak szükséges adatok tárolásával.

A Beowulfban a vezérlő és a számoló csomópontok hálózatba vannak kötve. Ez a hálózat zárt, az általános hálózati forgalomtól el van különítve. A hálózati eszközök választása többnyire a felhasználó anyagi lehetőségeitől függ. A 10 Mb/s sebességű ethernettől a nagysebességű (nagyobb, mint 1 Gb/s) különleges eszközökig (pl. a Myrinet az USA-ban) széles a választék. A legolcsóbb hálózat ethernetkártyák, jelelosztók és Cat5 UTP-kábelek segítségével alakítható ki. Hacsak nem akarod a géptelep összes felhasználóját a vezérlő csomópontra telepíteni, nem árt, ha a Beowulf-vezérlőközpont egy második hálózati kártyán keresztül a külső hálózatra is csatlakozik. Ebben az elrendezésen a vezérlő csomópont a zárt Beowulf-hálózat és a munkahelyi nyilvános hálózat között az átjáró szerepét is játssza. A felhasználók a nyilvános hálózaton keresztül a vezérlő csomópontra távolról be tudnak jelentkezni, és a második hálózati kártyán át elérhetik a géptelep erőforrásait, azonban nem tudnak átlépni a vezérlőn és közvetlenül csatlakozni a számoló csomópontokhoz.

A Beowulfot a fent vázolt módon különálló számítási egységként érdemes kezelni a szervezeten belül. Ennek a megközelítésnek számos előnye van a teljesítmény, a kezelhetőség és a biztonság szempontjából. Nem csoda, hogy a jelenlegi Beowulf program ezt az elrendezést támogatja. A Beowulf fizikai kialakítása *ábránkon* figyelhető meg.

Lényegében két nézet létezik a Beowulf operációs rendszerét illetően. Hadd hangsúlyozzam, hogy mindkettő jó, bár eltérő. A két megoldás a géptelep célját illetően különböző igényeknek felel meg. Többek között a következő szempontok határozzák meg a géptelepet működtető program beállításait, valamint a hozzáférés szabályozását és ellenőrzését: a géptelep feladata, a felhasználók fajtája és száma és a futtatandó alkalmazások.



Egy jellemző Beowulf-rendszer fizikai kialakítása

Ha ezeket az elején figyelmen kívül hagyod, később bajba kerülhetsz, ezért először a két géptelep tervezésének filozófiáját tárgyaljuk.

Az eredeti Beowulfban minden csomópont teljes értékű Linuxot futtatott, és a felhasználónak az alkalmazás futtatásához minden csomóponton azonosítóval kellett rendelkeznie. Ez az elrendezés rengeteg pluszköltséget okozott minden olyan csomópontban, ahol az alkalmazásnak futnia kellett, ráadásul a helytelenül viselkedő folyamatok kezelése is nehézkes volt. Azóta ez a gépteleptípus valamelyest fejlődött. A DHCP, a Red Hat Kickstartja, az SSH, az MPI, a HTTP és a MySQL kétségkívül leegyszerűsítették a géptelep telepítésének és kezelésének feladatait, de a vezérlő csomópontra bejelentkezve a felhasználók továbbra is elérhetik a számoló csomópontokat, és a csomópontok önálló életet élhetnek.

A számoló csomópontok elérhetősége és vezérelhetősége kívánatos lehet egy adott géptelep és adott felhasználói kör esetében, ezért ez lényeges felügyeleti döntés. Két terjesztés használja ezt a modellt: az NPACI által készített Rocks és az Open Cluster Group által készített OSCAR.

A másik géptelepelméletet a Beowulf alkotója dolgozta ki. Ez a megközelítés az vezérlő-számoló viszonyt a méhkaptár mintájára valósítja meg. A vezérlő csomópont a méhkirálynő, aki teljes kromoszómakészlettel rendelkezik, tud magáról gondoskodni, és vezérli a kaptárt. A számoló csomópontok csak a kromoszómakészlet felével rendelkeznek, ők a herék, és annyi eszük sincs, mint egy marék molylepkének. A számoló csomópontokra nem lehet távolról belépni, azok egyszerűen csak az uralkodó parancsait hajtják végre. Bár a rovarpárhuzam nagyon jól hangzik, ezt az elrendezést egyszerűen SSI-nek nevezik, és az ezt megvalósító terjesztés neve Scyld. Az SSI a másik véglet a számoló csomópontok elméletében, de egyértelműen megvannak az előnyei.

Melyik modell felel meg céljaidnak? Nehéz kérdés, és végül oda lyukadunk ki, hogy attól függ, milyen jogokat szeretnél

adni a felhasználóknak a teljes rendszeren. Csak akkor hozhatsz körültekintő döntést, ha előbb eljátszol egy kicsit a különféle alapértelmezett beállításokkal.
Kezdjük az ismerkedést a különféle géptelepkezelő programokkal, a kollégáim által készített terjesztés, az NPACI Rocks telepítésével. A San Diego Supercomputer Centerben dolgozó kis munkacsoport rendkívül megbízható és könnyen használható terjesztést készített. A géptelep kiépítéséhez mindössze ábránkhoz hasonló x86-os gépek (IA32 vagy IA64) hálózata, internetkapcsolat, CD-író, CD és hajlékonylemez szükséges.
Először is látogass el a Rocks webhelyére, a ➲ http://rocks.npaci.edu címre. Rövid általános bevezetőt olvashatsz a géptelepek építéséről és a Rocks projekt sajátos módszereiről. Egy dolog azonnal világossá válik a webhely tanulmányozása során, mégpedig hogy a Rocks-terjesztés készítőinek egy cél lebegett a szemük előtt: a Beowulf géptelepeket legyen egyszerű telepíteni és kezelni! A Rocks csoport e cél elérésének érdekében

1. a terjesztést a Red Hat Linux alapján készítette el,
2. a Red Hat Linuxhoz hozzáadott olyan minden nyílt forrású programot, ami a Beowulf használatához szükséges,
3. minden géptelepkezelő programot rpm-csomagokba csomagolt,
4. az vezérlő és a számoló csomópontok telepítésének gépesítésére a Red Hat Kickstart programot használta,
5. a vezérlő csomóponton létrehozott egy MySQL-adatbázist, amely a géptelep adatait rendezi,
6. hozzáadott egy-két programot, amely az egészet egybefogja.

Csupa nagyszerű ötlet. A Rocks-terjesztés a következő telepkezelő programokat tartalmazza: PBS, Maui, SSH, MPICH, PVM, tanúsítványhitelesítő program, a Myricom általános üzenetküldő rendszere a Myrinet kártyákhoz. Amennyiben még ez sem lenne elegendő, a fejlesztők a továbbfejlesztés lehetőségét is nyitva hagyták: saját Beowulf-változatodat testreszabhatod és programjaid hozzáadásával felépítheted. Ugye, fantasztikus?
A Rocks webhelyén a bal oldali ***Getting Started*** hivatkozás elvezet egy olyan leíráshoz, amely lépésről lépésre elmagyarázza a Rocks-géptelep telepítését. A ***Step 0*** röviden leírja a géptelep fizikai kialakításának tudnivalóit. Az építőelemek másmilyenek is lehetnek, de az összeállításnak az ábránkon látható felépítésre kell emlékeztetnie.
A ***Step 1*** leírja, hogyan töltsd le a rendszerindításra képes lemezlenyomatot (ISO) a Rocks FTP-kiszolgálójáról, és miként írd fel CD-re. Az NPACI Rocks jelenleg Red Hat Linux 7.1-sen alapul, azaz két telepítő CD-je van, azonban csak az első szükséges a Rocks-géptelep telepítéséhez.
A ***Step 2*** leírja a ***kickstart*** beállítófájl elkészítését és a vezérlőcsomópont (a Rocks szóhasználatában front end) telepítését. Szerencsére a Rocks csoport webhelyén egy CGI-űrlap található, ennek kitöltésével a fájl könnyen elkészíthető. Kattints a ***Build a configuration file*** hivatkozásra, és az űrlap bekéri a ***kickstart*** fájl elkészítéséhez szükséges adatokat. Ez a fájl fogja beállítani a front end külső és belső csatolófelületeit, valamint a géptelep számára nyújtott nyilvántartó, időzítő és névkiszolgáló szolgáltatásokat. Számos tanács és alapérték segít az űrlap helyes kitöltésében.
Az űrlap kitöltése után kattints a ***Submit*** gombra. Ezután a böngésző felajánlja a fájl mentését. A fájlt ***ks.cfg*** néven mentsd, és egy DOS formátumú hajlékonylemezre másold át. Minden együtt van a Beowulf-géptelep telepítéséhez – a Rocks CD-k és a rendszeredre jellemző kickstart lemez. Kezdődhet a telepítés! A leendő front end csomópontot kapcsold be, győződj meg arról, hogy a CD-ről tud-e rendszert indítani, tedd be az első CD-t, és indítsd újra a gépet. A `boot:` parancssorba írd be a `frontend` szót, a lemezt helyezd be, és végig kísérd figyelemmel, ahogy a rendszer települ. A telepítés után a gép kiadja a CD-t, és újraindul. Az újraindulás előtt vedd ki a CD-t és a lemezt, nehogy megint elkezdődjön a telepítés. Rendszergazdaként jelentkezz be, ekkor a gép meg fog kérni az SSH-kulcsok elkészítésére. Miután a kulcsok elkészültek, futtasd az `insert-ether` parancsot. Ez a program megkönnyíti a Beowulf telepítését és kezelését, mert a számoló csomópontok adatait, amelyeket a DHCP-kérésekből nyer ki, a Rocks MySQL adatbázisába helyezi. A program menüjéből válaszd a `Compute` menüparancsot, helyezd be az első CD-t az első számoló csomópont CD-meghajtójába, kapcsold be a gépet, és várd meg, amíg a rendszer települ. A telepítés végén a gép kiadja a CD-t. Tedd át a CD-t egy másik számoló csomópontba, és kapcsold be. A fenti műveletet az összes számoló csomóponton ismételd meg.
Ennyi az egész. Az biztos, hogy sok minden egyéb zajlik a háttérben, de a Rocks készítői ezeket a folyamatokat szándékosan elrejtették előled. Ha a érdekelnek részletek, a folyamatosan bővülő leírás a webhelyen fellelhető.
Gondolom, már alig várod, hogy Beowulfodon alkalmazásokat futtass. A webhelyen a ***Step 4*** pontban rövid ismertető olvasható az MPI-t, illetve a PBS-t használó alkalmazások futtatásáról. Még egy tanács: az MPI-feladatokat az `mpi-launch` parancs a Myrineten keresztül, míg az `mpi-run` parancs az etherneten keresztül indítja el.
A cikk itt véget ér, azonban nemsokára újra jelentkezem egy másik könnyen telepíthető Beowulf-terjesztés ismertetésével, azután egy harmadikkal és még eggyel. Azt követően számítórácsba kötjük őket. Végül jöhet a világuralom.

Linux Journal május, 97. szám

Glen Otero
Ph.D. fokozatot szerzett immunológiából és mikrobiológiából, továbbá a Linux Prophet nevű tanácsadó céget vezeti a kaliforniai San Diegóban.

Kapcsolódó címek

Beowulf.org ➲ http://www.beowulf.org
Kickstart ➲ http://www.redhat.com/docs/manuals/linux/RHL-7.2-Manual/custom-guide/ch-kickstart2.html
Linux Clustering Information Center ➲ http://www.lcic.org
Maui ➲ http://www.supercluster.org/maui
MPI ➲ http://www-unix.mcs.anl.gov/mpi/index.html
Myrinet ➲ http://www.myri.com
OSCAR ➲ http://oscar.sourceforge.net
PBS ➲ http://www.openpbs.org
PVM ➲ http://www.csm.ornl.gov/pvm
Red Hat Linux ➲ http://www.redhat.com
Rocks ➲ http://rocks.npaci.edu
RPM ➲ http://www.rpm.org
Scyld ➲ http://www.scyld.com