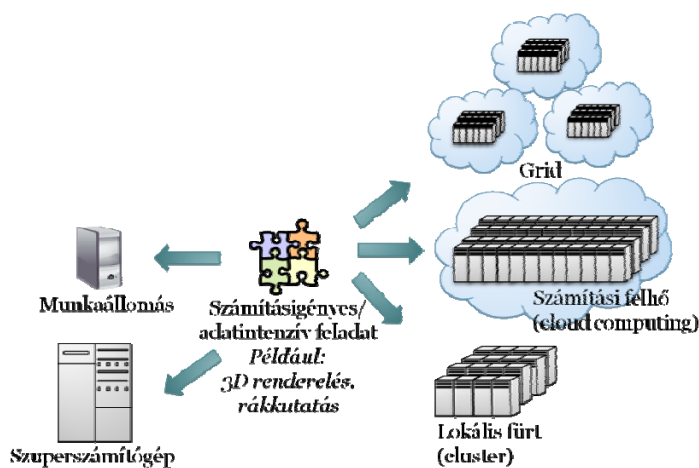


Amazon EC2 infrastruktúrán futó dinamikusan skálázható PBS klaszter

Ács Sándor, Kozlovszky Miklós, Marosi Attila Csaba, Balaton Zoltán
MTA SZTAKI

1. Bevezetés az elosztott rendszerek világába

A számolás és adatintenzív problémák kezelésére jelentős mennyiségű számítási teljesítményre és tárhelyre van szükség. A rohamos ütemben fejlődő processzorok és tárolóeszközök sem képesek hatékonyan ellátni bizonyos feladatokat. Klasszikus nagy számításigényű feladatok közé tartoznak a tudományos alkalmazások. Például atomi szintű reakciók szimulálása, szeizmológia, meteorológia vagy például biokémia. Hétköznapi felhasználást tekintve a 3D modellezést és renderelést lehet megemlíteni, mint nagy erőforrás igényű alkalmazási terület.



Ábra 1: A probléma megoldásához igénybe vehető erőforrások

A kutatók és az egyszerű felhasználók is szembesülhetnek azzal a ténnyel, hogy az általuk használt számítógép teljesítménye nem kielégítő számukra. Ebben az esetben, mint az első ábra is szemlélteti, a következő megoldások jöhetnek szóba:

1. Lokális fürt (hétköznapi nevén klaszter) használata.
2. Grid erőforrások használata.
3. Számítási felhő szolgáltatások igénybe vétele.

A különböző típusú elosztott rendszereknek megvan a maguk előnye és hátránya egyaránt. Előfordulhat, hogy a felhasználó nem tud klasztert építeni vagy bővíteni az alkalmazása futtatásához, mivel ehhez nincs meg a szükséges anyagi háttér vagy szaktudás, illetve a vásárlásra és beüzemelésre fordított idő és pénz nem lenne arányban az elérhető eredmények értékével. Egy tudományos alkalmazást akár ingyen is lehet griden futtatni, például a Magyar Nemzeti Grid bölcsőjének számító Hungrid virtuális szervezet segítségével [1]. De ehhez át kell alakítani a

futtatni kívánt alkalmazást, hogy az adott grides infrastruktúrán működhessen. A felhő infrastruktúrában futó virtuális gépek bérlése sem ad kész megoldást, hiszen az alkalmazást fel kell készíteni, hogy több virtuális számítógép erőforrásait aknázza ki, vagy a bérelt gépeken kell kialakítani klasztereket a már jól bevált technológiák segítségével. A felhőn történő klaszter kialakítása racionális megoldás, mivel az egyik legfontosabb szempont egy alkalmazás elosztott rendszerben történő futtatásakor, az hogy az eredeti kódot minél kevésbé kelljen módosítani és már rendelkezésre állnak olyan klaszterező technológiák, amelyek elvégzik a feladatok ütemezését és menedzselését (például állapotellenőrzéseket, migrációt, eredmény helyesség ellenőrzést).

A felhő és klaszter átjárhatóság témájának feldolgozása során megismert megoldások Condor[2] alapúak voltak. Az elterjedten használt és a klasztertechnológiák között fontos szerepet betöltő PBS[3] alapú klaszter felhőbe történő kiterjesztésére példát nem találtunk a kutatás kezdetén. Ezért a munka célja volt, hogy az elterjedten használt PBS számítási fürt dinamikusan és automatikusan kiterjeszhető legyen a számítási felhők között úttörőnek számító Amazon EC2 (Elastic Cloud Compute)[4] vagy annak felületével (idegen szóval interfészével) kompatibilis számítási felhőkkel. Ezáltal lokális PBS fürt hiányában is lehet felhő infrastruktúrán futó PBS klasztert használni, illetve a már rendelkezésre álló PBS fürtök kapacitása növelhető, ha az alkalmazás erőforrás szükséglete megköveteli. További előnyt jelent, hogy a már rendelkezésre álló PBS fürtre írt szoftverek változtatás nélkül futtathatóak felhő infrastruktúrán.

2. PBS klaszter

Hasonló felépítésű és feladatot ellátó számítógépek nagysebességű hálózaton keresztül összekapcsolt csoportját számítási fürtnek vagy klaszternek nevezzük. Használatuk előnye, hogy egy fürt robusztusabb (redundancia beiktatásával) és/vagy nagyobb teljesítményt nyújt, mint egyetlen számítógép. Több gépből kialakított klaszterrel hatékonyan gyorsíthatjuk a futtatott alkalmazás végrehajtását, ha az alkalmazás párhuzamosítható.

A PBS (Portable Batch System, azaz Hordozható Kötegelt Rendszer) egy számítógépes program, amely feladatok ütemezését végzi. Elsődleges célja számítási igények (például kötegelt feladatok) kiosztása az elérhető számítási erőforrásokra. Elterjedten használt UNIX/LINUX alapú fürt környezetben. A PBS számos meta-ütemező által támogatott, mint feladat ütemező mechanizmus. A PBS-t eredetileg az MRJ nevű cég kezdte fejleszteni a NASA-nak az 1990-es évek elején. Az MRJ-t felvásárolta a Veridian majd később a Veridian-t felvásárolta az Altair Engineering, amely jelenleg is forgalmazza a PBS Professional-t és az ingyenesen használható (de nem karbantartott és nem támogatott) OpenPBS-t. Léteznek további verziók is, érdemes kiemelni a TORQUE-ot, amely az OpenPBS leágazásából származik, de ellentétben azzal, aktívan fejleszti és karbantartja a Cluster Resource Inc. nevű vállalat. A munka során a TORQUE verziót használtuk, így a továbbiakban a PBS általános kifejezés alatt ez a verzió értendő.

A PBS tulajdonságai közé tartozik, hogy számos platformra elérhető, támogatja az automatikus terhelés elosztás, felhasznált be- és kimeneti fájlok előkészítését, kölcsönös függőség kezelését (végrehajtási irány, feltételek végrehajtása, szinkronizáció). Megtalálhatók benne a biztonságos elosztott rendszerekhez nélkülözhetetlen felhasználó kezelés (rendszer, csoport és felhasználó szinten). Elérhetővé teszi a párhuzamos (PVM, MPI¹) feladatok futtatását. Grafikus felhasználói felület nyújt. Továbbá széleskörű alkalmazásprogramozási felület ad,

¹ PVM (Parallel Virtual Machine), MPI (Message Passing Interface): párhuzamos futtathatóságért felelős szoftver és alkalmazásprogramozási felület

amely segítségével új parancsok és egyéni ütemezési házirendek definiálhatók, illetve a PBS beágyazhatóvá válik más alkalmazásokba.

3. Amazon EC2 számítási felhő

A számítási felhő részben már meglévő technológiák kombinációja, egy újszerű rendszer-megvalósítási megközelítéssel párosítva. Maga a felhő az Internetet szimbolizálja. Mivel az Internet egy elosztott hálózat, ahol a pontos fizikai elhelyezkedés nem fontos, ezért egy felhő jól szimbolizálja az egész rendszert. Az Internetes szolgáltatások száma az utóbbi évtizedben rohamosan nőtt, gondoljunk csak egy egyszerű email-szolgáltatásra, vagy egy kép- vagy video megosztó szolgáltatásra, (Gmail, Flickr, YouTube). A számítási felhő a szolgáltatások egyszerűbb, és hatékonyabb üzemeltetését célozza meg. Lényege, hogy a meglévő hardvereket maximálisan kihasználják, oly módon, hogy a felhasználó ne vehesse észre, ha a „háttérben” valami változás történik, tehát ne érzékeljen semmit a szolgáltatás mögött álló infrastruktúrából. Ehhez egy dinamikusan méretezhető rendszerre van szükség. A felhasználóknak az a vonzó mindebben, hogy csak annyit kell fizetni, amennyit használnak a rendszerből. Ez a szemléletmód, ami miatt egyre népszerűbb a számítási felhő típusú elosztott rendszer. Az alapján, hogy az egyes informatikai rétegek (felhasználói rendszerek és felhasználók, alkalmazások, adattárolás, felső rétegeket kiszolgáló infrastrukturális szolgáltatások, hardver) hová kerülnek fizikai értelemben, ki kezeli őket, mennyire változtathatók, többféle számítási felhőről beszélhetünk. Megkülönböztethetünk alapvetően három típust:

- infrastrukturális számítási felhő (IaaS - Infrastructure as a Service) - Virtuális gépek indíthatók az infrastruktúrán. Például az Amazon EC2 infrastruktúrája.
- platform-alapú számítási felhő (PaaS - Platform as a Service) - Egy célalkalmazás számára előre felkonfigurált virtuális gépek igénybevétele. Például összehangolt adatbázisszerverek.
- szoftver-alapú számítási felhő. (SaaS - Software as a Service) - Szoftver alapú számítási felhő igénybevételekor az alkalmazás használata közben már nem is érzékelhető, hogy elosztott rendszeren fut. Például a Google szolgáltatási (Google Docs, Youtube).

A felosztás egy másik dimenziója lehet az, hogy a felhő nyilvános vagy zárt rendszeren fut-e, így megkülönböztethetünk további három típust:

- Publikus számítási felhő. - A felhasználónak szerződést kell kötnie a szolgáltatóval és az „elfogyasztott” erőforrások alapján kell fizetnie. Tehát a felhasználó bérlő az erőforrásokat, ilyen például Amazon EC2 infrastruktúra.
- Privát számítási felhő. - Intézmények, vállalatok helyi erőforrásainak menedzselésére használható. Erre a célra fejlesztették ki az OpenNebula-t és az Eucalyptus-t.
- Hibrid számítási felhő. - A lokális privát számítási felhő mellett igénybe vehető publikus felhő infrastruktúra, például kiugró teljesítmény kezelésére.

Az Amazon EC2 (Elastic Cloud Compute) segítségével a vásárlók virtuális gépeket bérelhetnek, melyeken futtathatják alkalmazásaikat. A felhasználók létrehozhatják a saját virtuális gépeiket egy speciális formátumban (AMI - Amazon Machine Image), melyet a rendszerbe feltöltve, webes szolgáltatásokon keresztül képesek használni. A fizetés pedig az

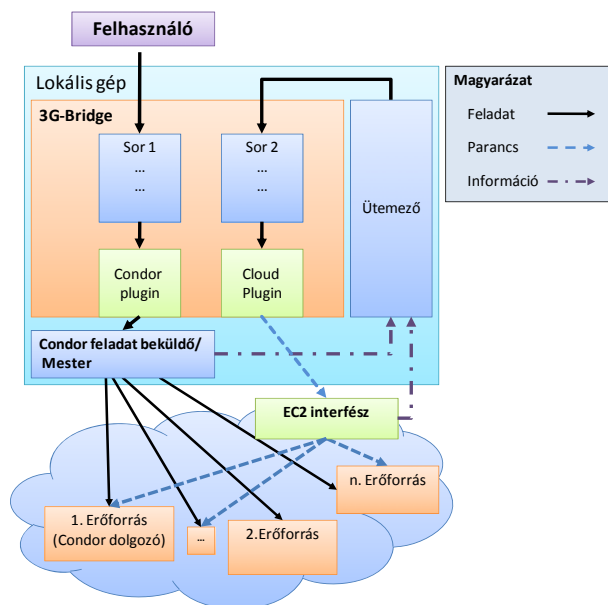
„elfogyasztott” processzoridő, tárhely és sávszélesség alapján történik. Az EC2 infrastruktúra tulajdonságai:

- Elasztikus - Könnyen és gyorsan skálázható. Egyszerűen egy weboldalon keresztül vagy parancssorból változtathatjuk a virtuális architektúra típusát, vagy a futó példányok számát.
- Teljesen vezérelhető – Futó virtuális gépek (példányok), úgy kezelhetők mintha a saját gépünkön üzemelnének, tehát teljes irányítást kapunk felettük.
- Rugalmas – Futatni kívánt alkalmazásnak megfelelő virtuális gép architektúra választható és a példányokhoz extra tárhely csatlakoztatható, a tárhely mérete szabadon változtatható.
- Megbízható - Szerződésben szerepel a 99.95% elérhetőség garantálása (, SLA biztosítva).
- Biztonságos - Biztonsági zónák definiálhatók, a példányok indulása előtt beállítható, hogy milyen portokon lehessen elérni őket.

4. Elosztott rendszerek közötti átjárhatóság

Az elosztott rendszerek együttműködéséért több nemzetközi szervezet is dolgozik. A cél, hogy a már meglévő rendszerek közötti átjárhatóság biztosított legyen. Ilyen például a Nyílt Grid Fórum (OGF - Open GRID Forum) Produkciós Grid Infrastruktúra (PGI - Production Grid Infrastructure) munkacsoport. A gridek területéről említhető az Európai Köztesréteg Kezdeményezés (EMI - European Middleware Initiative) projekt, melynek célja, hogy integrálja a három legnagyobb európai grid köztesréteget egy egységesített köztesréteg disztribúcióvá. A BOINC és XtremWeb asztali gép alapú grid rendszereket sikeresen integrálták gLite alapú szolgáltatás gridekkel, az EDGeS projekt keretein belül. [5] A kifejlesztett áthidaló technológiát 3G-Bridge-nek nevezik, amely az angol „általános grid átjáró (Generic Grid Gateway)” szavak kezdőbetűiből ered. A 3G-Bridge akadálymentesen egészíti ki a gLite alapú szolgáltatás grideket desktop gridekkel és fordítva.

A SZTAKI 3G-Bridge egy speciális esetben, az EC2-es beépülő modult és számítási felhőben induló Condor erőforrást alkalmazva a következő ábrán látható módon működik.

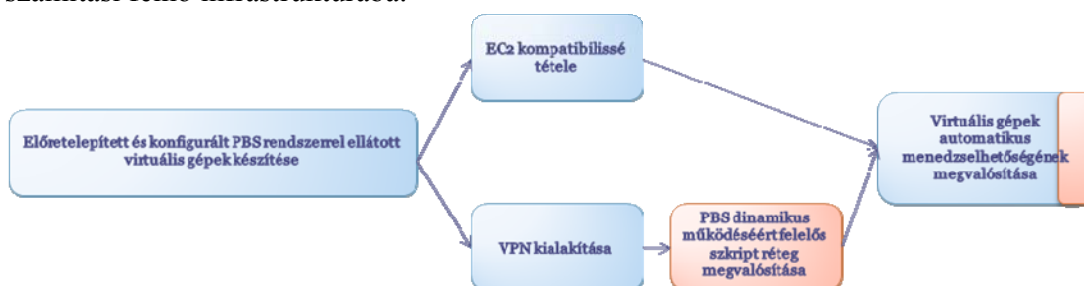


Ábra 2: 3G-Bridge és EC2 beépülő modul segítségével EC2 infrastruktúrában futó Condor erőforrások

A második ábra bal felső sarkában jelölt felhasználó feladatot küld a 3G-Bridge-be. A 3G-Bridge, a „Cloud plugin” segítségével Condor erőforrás(oka)t indít az Amazon felhőben, melyeket nyilvántart a második várakozási sorban, majd továbbküldi a feladatot a Condor számítási fűrtbe az első várakozási soron keresztül a Condor beépülő egység segítségével. A fűrt (kép alsó részében látható) erőforrásait a számítási felhő biztosítja. Az ütemező egység figyelemmel követi a beküldött feladatok és a felhőben futó erőforrások számát. Ha a számítási fűrt telítődik feladatokkal, akkor az ütemező új erőforrásokat indít a második várakozási soron keresztül. Ha a fűrtnek kihasználatlan erőforrásai vannak, akkor az ütemező a második várakozási soron keresztül erőforrást állít le[6].

5. A rendszer specifikációja

A tervezett rendszer célja, hogy a PBS számítási fűrt dinamikusan kiterjeszthető legyen Amazon EC2 számítási felhő infrastruktúrába.



3. ábra: Megoldandó részfeladatok

A következő részfeladatokat kell megoldani, melyeket a 3. ábra is bemutat:

1. virtuális gépek készítése (PBS fej és dolgozó gép), fűrt konfigurálása,

2. virtuális gépek EC2 kompatibilissé tétele,
3. a PBS dinamikusan használhatóvá alakítása, EC2 infrastruktúrán működéséhez szükséges feltételek megteremtése (kommunikációs csatornák biztosítása és dinamikus névfeloldás),
4. virtuális gépek automatikus menedzselhetőségének megvalósítása a felhőben.

A felsorolt részfeladatok közül az első és második, illetve a negyedik részfeladat jelentős részének megvalósításához már meglévő eszközök használatára és integrációjára volt szükség. A munka innovatív része a harmadik részfeladat megoldása, a PBS klaszter dinamikusan használhatóvá alakítása (piros háttérű táglalap a képen). A megoldás újszerűségét egyrészt az adja, hogy a VPN hálózat segítségével a dolgozó egységek egy adminisztrációs tartomány alá kerülnek, másrészt a PBS fej egység automatikusan csatlakoztatja az erőforrások közé a VPN hálózathoz kapcsolódó dolgozó egységeket.

A PBS dinamikus működéséért felelős szkript réteg és VPN kialakítása alapvető megoldandó probléma volt. A PBS fejnek előre tudnia kell a hozzá kapcsolódó erőforrások listáját (a dolgozógépek azonosítása miatt). Ez azt jelenti, hogy a rendszerben fix IP címeket és hoszt neveket kell használni, ez pedig alapvetően nem adott egy felhő infrastruktúrában. A problémát többféle módon kezelhetjük. Vásárolhatunk fix IP címeket a dolgozó gépeknek, de ez nem gazdaságos. Üzemeltethetünk névfeloldásért (DNS) és IP cím kiosztásért (DHCP) felelős szolgáltatásokat a fej egységen, de ettől biztonságosabb és egyszerűbb megoldás ha VPN²-t (Virtual Private Network), azaz virtuális magánhálózatot üzemeltetünk a fej és a dolgozó egységek között. A VPN hálózatoknak többféle típusa is létezik, melyek különböznek felépítésükben, működési mechanizmusaikban és az elérhető maximális biztonsági szintben. A munka során PPTP (Point-to-Point Tunneling Protocol) típusú VPN-t használtunk, az egyszerű kezelhetőség miatt.

A számítási felhőben lévő virtuális gép példányok könnyű és hatékony használhatóságának érdekében egy alrendszer alakítottunk ki, melynek feladata, hogy szükség esetén erőforrásokat indítson a felhőben, vagy ha nincs elvégzendő feladat a várakozási sorban, akkor leállítsa a felesleges erőforrásokat. Természetesen ez csak a legegyszerűbb ütemezési stratégia, de az alrendszernek lehetőséget kell nyújtania a különböző stratégiák alkalmazásának lehetőségére. Mivel a számítási felhők használatáért általában fizetni kell, külön a hálózati forgalomért és az elfogyasztott processzor teljesítményért, illetve az elvégzendő feladatok sürgőssége is eltérő lehet ezért a kívánt stratégiák implementálása túlmutat a kifizűzött célon. A munka során a SZTAKI 3G-Bridge és a hozzá tartozó EC2-es beépülő modult használtuk fel, mert könnyen adoptálható a probléma megoldására, mivel hasonló rendszerhez (, Condor számítási fürthöz) készült eredendően. Az eredeti megoldást nem lehetett módosítás nélkül használni, mivel eltérő utasításai és működési elvei vannak a Condor-nak, mint a PBS-nek.

6. Rendszermodulok

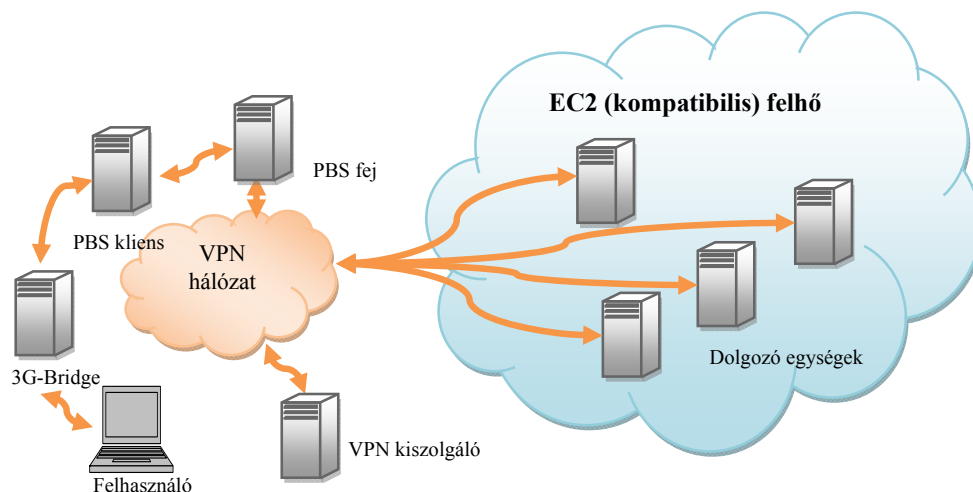
A rendszerben a következő egységeket és szolgáltatásokat kellett megvalósítani:

PBS fej egység, PBS dolgozó egység, VPN kiszolgáló, 3G-Bridge. A rendszeregységek, a funkciók definiálása után, a rendszer működésének ábráján láthatók.

² VPN (Virtual Private Network, azaz virtuális magánhálózat): olyan technológia, melynek segítségével biztonságos kommunikációt tudunk biztosítani az Interneten keresztül.

- 3G-Bridge - feladata, hogy a felhasználó igényeit (stratégiáit) figyelembe véve a feladatokat a rendszerbe juttassa, illetve menedzselje a számítási felhőben üzemelő PBS dolgozó egységeket. A feladat befejeződése után pedig az eredményt egy web szerveren tárolja.
- PBS kliens - szerepe, hogy a 3G-Bridge felől érkező lekérdezéseket, utasításokat a fej egységgel végrehajtsa, tehát egy interfész feladatot lát el.
- PBS fej egység felelőssége, hogy a beérkezett feladatokat egy várakozási sorból továbbítsa az általa összefogott dolgozó egységeknek és begyűjtse az eredményeket. (Részletek a PBS-ről szóló fejezetben olvashatók.)
- PBS dolgozó egység szerepe, hogy a beérkező feladatokat elvégezze és visszaküldje az eredményt a fej egységnek.
- VPN kiszolgáló - feladata, hogy az erőforrásokat egy biztonságos virtuális hálózattal összekötve megszüntsen a kommunikációs akadályokat. Tehát a virtuális hálózattal kialakított adminisztrációs tartományban nincs szükség tűzfalak menedzselésére mivel csak a dolgozó egységek kapcsolódhatnak rá. A fix IP címeknek köszönhetően megoldható, hogy az erőforrások címezhetőek, amely elengedhetetlen fontosságú egy PBS fürt üzemeléséhez.

7. A rendszer működésének leírása



4. ábra: Rendszer működésének ábrája

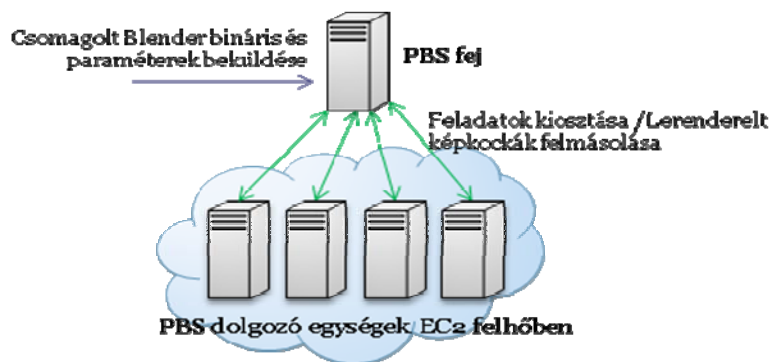
A 4. ábra mutatja, hogy a felhasználónak csak a 3G-Bridge-el kell kapcsolatban lennie. A „ws-client” nevű eszköz segítségével lehet feladatot beküldeni a rendszerbe. Az egyszerű ütemezési stratégiát feltételezve: a 3G-Bridge-en futó erőforrás menedzser a PBS kliens segítségével tájékozódik, hogy van-e szabad dolgozó egység (az ábrán kék felhőként feltüntetett EC2-es vagy azzal kompatibilis infrastruktúrában), ha van, akkor a PBS fej oda küldi a feladatot végrehajtásra. Ha nincs, akkor indít egy újabb virtuális dolgozó egységet a felhőben, ha még nem lépte át a beállított maximum egyszerre futó dolgozó egységek számát. Az erőforrás menedzser bizonyos (állítható) időközönként megvizsgálja, hogy a számítási felhőben futó dolgozó egységek között van-e olyan, ami feladat nélkül áll és letelt a paraméterben állítható készenléti idő. (Azért nem célszerű a feladat elvégzése után rögtön lekapcsolni a szabadon maradt dolgozó egységeket, mert

a számlázás minden megkezdett órára szól.) A PBS fej a VPN kiszolgáló által nyújtott virtuális magánhálózaton keresztül menedzseli a dolgozó egységeket. Tehát ezen keresztül juttatja el a feladatot és gyűjti be az eredményeket. A beérkező eredmények a 3G-Bridge által karbantartott web szerveren tárolódnak, ahonnan a felhasználó a már említett ws-client eszközzel érheti el az eredményre mutató URL-t. A felhasználó ezután akár a böngészőjével is letöltheti végeredményt a szerverről.

8. A tesztinfrastruktúra

A tesztek az Amazon Elastic Computing Cloud infrastruktúrán készültek. Mivel ez egy fizetős szolgáltatás ezért a tesztek nem szerteágazóak, az elkészített megoldás egyetlen CPU teljesítmény érzékeny alkalmazással vizsgálja és elemzi a rendszert. Ez az egyetlen alkalmazás viszont alkalmas arra, hogy bizonyítsa a tervezett és kivitelezett rendszer működőképességét. Az MTA SZTAKI LPDS laborjában, a következő hónapokban kiépítésre kerülő számítási felhő infrastruktúra hitelkártya adatok nélkül is hozzáférhető lesz, így lehetőség nyílik az elkészített rendszer széleskörű tesztelésére és továbbfejlesztésre.

A tesztek a BLENDER 3D rendering szoftver felhasználásával készültek. Mivel a 3D-s grafikának és renderelésnek komoly erőforrásigénye van ezért a fűrtben történő használat életszerű példa.



5. ábra: Blender működése PBS klaszteren

Az 5. ábra a PBS fűrtön futó Blender alkalmazás működését mutatja be. Az előre meghatározott „jelenetet” és paramétereit (bal felső sarokban látható nyíl) a fej egység küldi ki a szabad erőforrásokra. A feldolgozásának eredménye egy vagy több darab képkocka a bemeneti paramétereiktől függően, melyet a fej egységre másolnak vissza (zöld nyíl) a dolgozó gépek. Az alkalmazás nem igényel semmilyen kommunikációt futás közben, a memória használata sem haladja meg az 50MB-ot és az I/O műveletek sem jelentősek, így processzor intenzív feladatnak tekinthető.

A rendszerteszték azért készültek, hogy a rendszer működőképességét bizonyítsák és nem azt, hogy egy adott alkalmazás gyorsabban képes lefutni egy többgépes rendszeren, mint egy laptopon. A mérések három különböző erőforráson készültek az összehasonlíthatóság érdekében. Az első tesztsorozat lokális erőforráson, egy laptopon készült. A második tesztsorozat az MTA SZTAKI LPDS laborjában található lokális PBS fűrtön készült egy illetve kettő dolgozó egység bevonásával. A harmadik tesztsorozat az Amazon EC2-es infrastruktúrájában készült egy, kettő, négy illetve öt dolgozó egység bevonásával. A feladatok egyesével és tízesével hajtották végre az

említett rendszerek. A tesztkörnyezetekben található processzorok főbb paramétereit a 1. táblázat tartalmazza.

	Lokális gép (laptop)	Lokális fűt	EC2-ben futó fűt
Modell név:	Intel® U2500	Quad-Core AMD Opteron™ 2376	Intel® Xeon® E5430
Órajel:	1.2 GHz	2.66 GHz	2.3 GHz
Cache méret:	2048 KB	512 KB	6144 KB

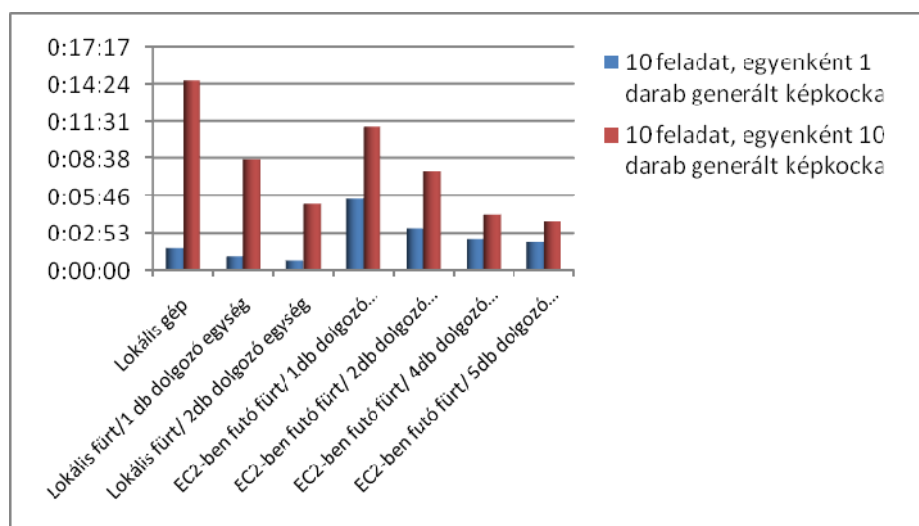
1. táblázat: A teszt erőforrásokban lévő processzorok főbb adatai

Az Amazon rendszerében különböző géptípusok igényelhetők felhasználásra. A munka során elvégzett rendszerteszt az alaptípussal készültek, melyhez egy darab logikai processzor járt.

Feladat X Kép (db)	Lokális gép	Lokális fűt		EC2-ben futó fűt			
	1 dolgozó egység	1 dolgozó egység	2 dolgozó egység	1 dolgozó egység	2 dolgozó egység	4 dolgozó egység	5 dolgozó egység
1X1	0:00:09	0:00:07	X	0:00:25	X	X	X
1X10	0:01:28	0:00:50	X	0:00:58	X	X	X
1X100	0:14:43	0:07:59	X	0:06:23	X	X	X
10X1	0:01:39	0:01:02	0:00:43	0:05:28	0:03:14	0:02:26	0:02:07
10X10	0:14:39	0:08:31	0:05:06	0:11:05	0:07:37	0:04:16	0:03:46

2. táblázat: Teszt eredmények

A 2. táblázat első oszlopában látható, hogy hány darab feladat lett beküldve a számítási fűtbe és darabonként mennyi képkockát generált le. Továbbá a felső sorban található a már bemutatott erőforrások, alattuk pedig a számításba bevont dolgozó egységek száma található. Az első három teszt sorban nincs értelme egynél több erőforrást lefoglalni feladatnak mivel egy darab dolgozó egységen fog futni mindenképpen (, ezért látható „X” a táblázatban).



6. ábra: A tesztek lefutásához szükséges idők összehasonlítása

Az 6. ábrából jól látszik, hogy a rövid ideig tartó kevés számítási kapacitást igénybevevő feladatokra nem alkalmazható hatékonyan a rendszer, ellenben az erőforrás igényes feladatoknál látványos eredményeket érhetünk el. Például 10 darab egyenként 10 képkocka generálását végző feladat közel 3,4-szeresére gyorsítható 4 darab EC2 számítási felhőben indított PBS dolgozó egység segítségével, a lokális teszt géppel szemben. Ezeknek az eredményeknek az üzenete, nem az a nyilvánvaló tény, hogy egy elosztott rendszeren hatékonyabban lehet futtatni egy erre felkészített alkalmazást, hanem az, hogy felhő infrastruktúra használatával igény szerint lehet erőforráshoz jutni anélkül, hogy lokális fürtöt kellene fenntartani, de mégis hasonló eredmények érhetők el és csak addig kell érte fizetni, amíg használatban vannak az erőforrások. Ezt a tény demonstrálja a tervezett, megvalósított és bemutatott rendszer.

9. Eredmények értékelése és a továbbfejleszthetőségi lehetőségek bemutatása

A megvalósított rendszer képes volt az Amazon EC2 infrastruktúrát kihasználni és a tesztek sikeresen lefutottak. Az eredmények ismeretében kijelenthető, hogy teljes értékű fürt infrastruktúra építhető számítási felhő infrastruktúrában. Az elkészített megoldással lehetőség nyílik számítási fürt használatára lokális erőforrások hiányában is. De használható a lokális erőforrások kibővítésére is, ha éppen megnövekedett számítási kapacitásra van szükség, így a megvalósított rendszer igénybevételevel gyorsabban dolgozható fel adott mennyiségű feladat.

A kialakított rendszer szélesebb körben való elterjedéséhez szükséges további alkalmazásokkal tesztelni, bevezetni a dolgozó egységek kommunikációjának lehetőségét az MPI típusú feladatok kezelésére, újabb ütemezési stratégiákat készíteni a 3G-Bridge-hez. Továbbá az Amazon EC2 felülettel nem kompatibilis számítási felhő rendszerekkel való integráció is megvalósításra vár.

10.Összefoglalás

A cikk bemutatja a PBS számítási fürtöt és az Amazon EC2 számítási felhőt, továbbá a bevezetett infrastruktúrák közötti „átjárhatóság” kérdésével foglalkozik. A munka bemutatja, hogyan köthető össze egy lokális PBS számítási fürt Amazon EC2 felhőben futó komponenseivel és ismerteti a megvalósított fürtből felhőbe történő átjárhatóságot megoldó rendszer elemeit. A bemutatott tesztek és futási eredmények igazolják, hogy az elkészített skálázható rendszer képes megoldást nyújtani az erőforrás hiányban szenvedő felhasználóknak.

11. Irodalomjegyzék

- [1] (2011, Márc.) Hungrid Virtuális Szervezet weboldala. [Online]. HYPERLINK
"http://grid.kfki.hu/hungrid/" <http://grid.kfki.hu/hungrid/>
- [2] D. Thain, T. Tannenbaum, and M. Livny, "Condor and the Grid," in *Grid Computing: Making The Global Infrastructure a Reality*, B. Fran, A. J. G. Hey, and G. Fox, Eds. 2003 - ISBN: 0-470-85319-0.
- [3] (2010, Nov.) Eredeti PBS weboldal. [Online]. HYPERLINK
"http://www.nas.nasa.gov/Software/PBS/" <http://www.nas.nasa.gov/Software/PBS/>
- [4] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," *IEEE Internet Computing*, vol. 13, no. 5, pp. 14-22, Sep. 2009.
- [5] E. Urbah, et al., "EDGEs: Bridging EGEE to BOINC and XtremWeb," *Journal of Grid Computing*, vol. 7, no. 3, Sep. 2009.
- [6] P. Kacsuk, A. Marosi, K. Miklós, S. Ács, and Z. Farkas, "Parameter Sweep Job Submission to Clouds," in *Grids, Clouds and Virtualization*. SpringerLink, 2011.