

# A ClusterGrid és a P-GRADE Portál összekötése

Sipos Gergely, Patvarczki József, Hermann Gabor, Drótos Dániel, Kacsuk Péter, Farkas Zoltán  
[\[sipos|patvarcz|ghermann|kacsuk|zfarkas\]@sztaki.hu](mailto:[sipos|patvarcz|ghermann|kacsuk|zfarkas]@sztaki.hu)

MTA Sztaki LPDS

## Bevezetés

A ClusterGrid[1] az elmúlt két évben sikeresen szolgálta ki a magyar kutatókat nagy számítási igényű problémáik megoldásában. A jelenlegi tapasztalatok alapján a magyar felhasználók elsősorban az ún. paraméter study alkalmazásokra használják a ClusterGrid rendszert, noha annak lehetőségei megengednék a párhuzamos és workflow jellegű alkalmazásokat is. A fejlett országok kutatói körében a párhuzamos és workflow jellegű alkalmazások sokkal népszerűbbek, mint hazánkban. Ennek egyik oka lehet, hogy a magyar alkalmazók többsége fizikus, akiknek problémája természetes módon igényli a paraméter study jellegű Grid használatot. Másik oka lehet, hogy a magyar felhasználóknak nincs olyan magas szintű Grid portál a birtokukban, melynek segítségével kényelmesen tudnának párhuzamos és workflow alkalmazásokat fejleszteni és futtatni a ClusterGrid-en.

Ugyanakkor a SZTAKI által már korábban kidolgozásra került egy olyan magas szintű programozási környezet, a P-GRADE 2] és annak portál verziója[3], ami lehetővé teszi mind PVM, mind MPI alapú párhuzamos alkalmazások gyors és hatékony fejlesztését, ill. ezek futtatását akár egyedi klaszterekben, akár Grid rendszerekben. Ezen túlmenően, mind a P-GRADE rendszer, mind portál verziója lehetővé teszi szekvenciális programokból ill. párhuzamos PVM és MPI programokból komplex workflow alkalmazások összeállítását és azok grides futtatását.

A P-GRADE portál először a GT2 alapú Gridekben volt tesztelve: magyar SuperGrid projektben, az EU GridLab testbed-ben, végül az EGEE infrastruktúra alapját képező LCG-2 alapú Gridekben, mint amilyen a SEEGRID[4] és a HunGrid[5]. A jelen cikkben arról kívánunk beszámolni, hogy a SuperGrid-ClusterGrid IKTA-6 projekt keretében elkezdjük a P-GRADE portál összekapcsolását a ClusterGriddel. A cikk első fejezete összefoglalja, hogy a P-GRADE portál és ClusterGrid együttesen milyen szolgáltatásokat kínál a felhasználóknak. A második fejezet bemutatja a P-GRADE portál belső felépítését. A harmadik fejezet taglalja azokat a problémákat, amiket a P-GRADE portál és a ClusterGrid illesztésének érdekében meg kellett oldani. A negyedik fejezet bemutatja az összekötésben alkalmazott műszaki megoldásokat.

## 1 P-GRADE portál szolgáltatások a ClusterGrid felhasználók számára

A P-GRADE Portál a felhasználóknak a következő funkciókat nyújtja:

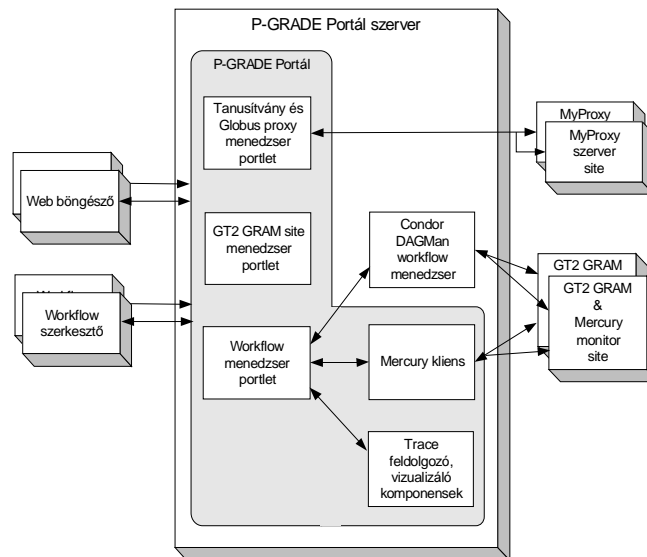
- Párhuzamos és szekvenciális job-okból összeállított workflow-k végrehajtása a ClusterGrid klaszterein.
- Mercury monitorrendszerrel felszerelt erőforrásokról a beküldött jobok processzei által „termelt” monitor-trace összegyűjtése és vizualizálása.

A portál e funkciók mellett biztosít egy kliens oldalra dinamikusan letölthető, Java Web Start alkalmazásként implementált szerkesztő környezetet is, melynek segítségével egymásra épülő jobokból álló workflow-kat lehet definiálni, illetve a portál szerverre feltölteni.

A portál tipikus felhasználási forgatókönyve a következő: a P-GRADE környezettel elkészített és lefordított jobokból a felhasználó a portálból letölthető szerkesztővel workflow-t készít, a futtatásához szükséges fájlokat a szerkesztőt használva a portál szerverre tölti, majd a workflow-t – immár a portál webes interfészét használva – Grid erőforrásokon lefuttatja. A workflow futtatását, a jobok indításának ütemezését, a monitor adatok és eredmények összegyűjtését a Portál végzi, azzal a felhasználóknak nem kell törődniük. A futás sikeres befejeződése után a workflow tulajdonosa a portál szerver gépről – ugyancsak a böngészőt használva – töltheti le számítási workflow-jának végeredményét.

## 2 A P-GRADE portál belső felépítése

A P-GRADE Portál portlet alapú technológiát használ. Ennek eredményeképpen a portál különböző funkcióit egymástól aránylag függetlenek tekinthető „portlet”-ek adják. A fontosabb portletok P-GRADE Portálban betöltött szerepét mutatja az 1. ábra:

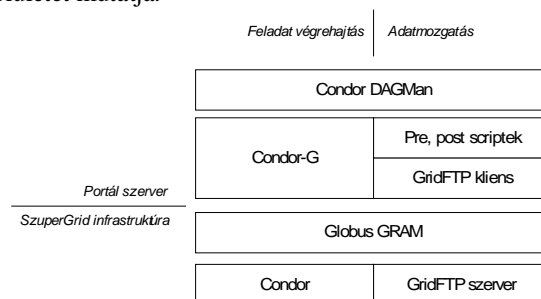


1. ábra

**A P-GRADE portált alkotó fontosabb portletok és kapcsolataik GT2 alapú Gridok esetén**

A Portál jelenleg 3 fontosabb portletet tartalmaz. Az első felelős a Grid MyProxy szervereivel való együttműködésért: a hosszú távú X.509-es tanúsítványok feltöltéséért és a rövidtávra szóló Globus proxy-k letöltéséért.

A második portlet a felhasználó által ismert GT2 GRAM erőforrások elérési címeinek beállításait teszi lehetővé. Ezen a portleten keresztül a felhasználó megadhatja, hogy mely job végrehajtó erőforrások jöhetnek egyáltalán szóba workflow-inak futása során. Itt nyilvánvalóan olyan erőforrásokat ésszerű megadni, ahol a felhasználó biztosan tudja, hogy rövid távú proxy tanúsítványa el lesz fogadva. A harmadik portlet a felhasználók által összeállított job-workflow-k lefuttatásáért és futás közbeni monitorozásáért felelős. Ennek a portletnek a back-end infrastruktúrát a portál szerver gépen installált Condor DAGMan, Mercury kliens és trace feldolgozó komponensek, míg a távoli erőforrásokon a GRAM és Mercury szerverek adják. A Condor DAGMan a workflow-t alkotó feladatok megfelelő ütemezés szerinti elindításáért, a Mercury kliens a Gridben futó joboktól érkező trace adatok összegyűjtéséért, míg a trace feldolgozó komponensek ezen adatok grafikus megjelenítéséért felelnek. A Grid szolgáltató hosztjain üzemelő GRAM és MyProxy programoknak rendre a jobok futtatása, illetve a jobok által termelt trace megfelelő helyre való továbbítása a feladatok. A 2. ábra a P-GRADE Portál és GT2 alapú SzuperGrid infrastruktúra kapcsolódási felületét mutatja:



2. ábra

**A P-GRADE Portál workflow végrehajtást megvalósító rétegei**

### 3 A ClusterGrid a P-GRADE Portál szemszögéből

A P-GRADE Portál az előzőekben ismertetett GT2 infrastruktúra tetején definiál egy olyan funkcióhalmazt, melyre a végfelhasználók által elérhető magas szintű grafikus felület ráépül. A P-GRADE Portál ClusterGridhez való illesztésének elvégzése előtt szükségszerű megnézni hogy filozófiájában és eszköztárában mennyire tér el a ClusterGrid a GT2-től. Ennek első lépéseként a fejezet áttekinti a ClusterGridet a kliens programok szemszögéből. Ennek célja megvizsgálni, hogy mely komponensek hiányoznak a P-GRADE Portálhoz való integráláshoz, és azok hogyan hozhatók létre.

#### 3.1 A ClusterGrid infrastruktúra a kliensek szemével

A ClusterGrid által a felhasználóinak (a végfelhasználók nevében fellépő kliens programoknak) nyújtott szolgáltatások a következők:

1. Brókerzés
2. Job végrehajtás
3. Job szintű adatmozgatás

Mint az a felsorolásból látható, a GT2-höz képest a ClusterGrid egy új, brókerző funkcionalitást nyújt, viszont hiányzik a job szintű monitorozáshoz szükséges funkció. A monitorozás hiánya a végfelhasználó számára nem okoz szemléletbeli különbséget, csupán megakadályozza a jobok által futás közben termelt trace adatok begyűjtését. Az új szolgáltatás, az erőforrás-réteg tetején létrehozott bróker megjelenése viszont alapjában változtatja meg az infrastruktúra használati feltételeit.

Amíg a GT2-ben az erőforrások a kliensprogramok számára (következésképpen a P-GRADE Portál számára is) egymástól független szolgáltatókként jelennek meg, addig a ClusterGridet a szemükben egyetlen központi elem, a bróker reprezentálja. Amint az az előző fejezetben már említésre került, a GT2-ben az infrastruktúrát használni kívánó kliensprogramoknak közvetlenül a jobvégrehajtásra kiszemelt erőforrásokon futó szervereket kell használniuk. (A P-GRADE Portál esetén a szerverek elérésére használt kliensek a Condor-G és a GridFTP kliens.) Mivel a GT2 infrastruktúra nem tartalmaz brókert, a kliens programoknak saját maguknak kell elintéznük az erőforrás kiválasztást, a jobjaikat alkotó fájlok megfelelő erőforrásra való másolását majd a jobok beküldését.

A ClusterGridben mindezen folyamatokat az infrastruktúra részét képező bróker program végzi. A bróker az, akinek a kliensprogramok a jobokat alkotó fájlokat a megfelelő követelmény-leíró állományok kíséretében átadják, ő az, aki a leírások alapján megtalálja a legmegfelelőbb erőforrásokat, aki átviszi a kiválasztott erőforrásokra a jobok fájljait, illetve aki a távoli erőforráson kezdeményezi a futtatást. Később a brókeren keresztül lehet a futó job állapotát lekérdezni, illetve a végrehajtó erőforrásról az eredményfájlokat „megszerezni”. A bróker tehát teljes egészében elrejtja a számítási funkciókat nyújtó erőforrásokat a kliensprogramok elől, sőt a beküldés során jelentkező feladatokat el is végzi helyettük.

Mivel a P-GRADE Portál a „ClusterGrid” kliens szerepét fogja betölteni, ezért a cikk következő része azt adja meg, hogy mely komponensek és hogyan teremthetik meg a kapcsolatot a P-GRADE Portál magas szintű, workflow-k menedzselésére használható grafikus felülete és a ClusterGrid brókere között. A bróker erőforrás-elrejtő szerepe miatt felesleges foglalkozni a bróker és a tényleges job végrehajtó funkciót nyújtó számítási erőforrások között meglévő kapcsolatokkal, protokollokkal, rétegekkel.

#### 3.2 A ClusterGrid brókerének elérése

A ClusterGrid brókere az infrastruktúra részét képező úgynevezett entry gépen futó démon. (Ha több entry gép is van, akkor mindegyiken a többitől független, teljesen egyforma démon fut.) Ez a démon a hozzá beérkező hívások hatására a megfelelő sorrendben eléri és használja a számára back-end szolgáltatóknak tekinthető számítási erőforrásokon futó szervereket, illetve az entry gép fájlrendszerét.

A bróker démon az entry gépre való belépés után az adminisztrátorok által előzetesen odatelepített kliens programokon keresztül érhető el. Ezek a kliens programok egy-egy folyamat brókernél történő kezdeményezésére használhatóak (Mint pl. job beküldés, fájlmásolás, állapot lekérdezés).

Amíg az előző fejezetben olvasható volt, hogy a GT2-ben a Grid infrastruktúra szervereinek használatához Globus tanúsítványokra van szükség (a Mercury használatához semmire) addig a

ClusterGrid esetén a hozzáférés szabályozás az entry gépre való belépés szabályozásán keresztül van megoldva. Mivel a számítási funkciókat nyújtó szerverek a brókeren keresztül érhetőek el, a bróker viszont csak az entry gépre való bejelentkezés után érhető el, ezért aki be tud jelentkezni a ClusterGrid (legalább egyik) entry gépére, az el is tudja indítani a bróker használatához szükséges klienseket. A ClusterGrid használata tulajdonképpen nem jelent mást, mint az entry gépen lévő kliensprogramok elindítását, illetve az entry gép fájlrendszerének írását, olvasását

### 3.3 A ClusterGrid brókerének használata

A bróker interfészén keresztül az alábbi funkciók érhetőek el.

1. Követelmény-leíróval ellátott job beküldése – `clgr_submit`
2. A beküldött job aktuális státuszának lekérdezése – `clgr_status`
3. Sikertelenül futott jobbal kapcsolatos hiba információk lekérdezése – `clgr_status -e`
4. Sikeresen végrehajtott job eredményfájljainak futtató erőforrásról való visszahozatala – `clgr_getout`

Az egyes sorok végén lévő parancsok az entry gépen az adott funkció igénybevétele miatt installált kliensprogram elindítására használhatóak.

#### 3.3.1 Job beküldés

A brókerrel elvégezhető job beküldési folyamat valójában több részfolyamat egy egységbe zárását jelenti. Ezek a részfolyamatok az erőforrás-keresés, az input fájlok másolása és a job beküldése. Amíg a bróker mindhárom feladatot külön-külön kell, hogy elvégezze, addig a bróker kliens programja számára azok egyetlen atomi műveletként kezelhetők.

A beküldés előtt a felhasználónak (vagy a nevében eljáró ClusterGrid kliensprogramnak) az entry gépen egy jól definiált könyvtárstruktúrába kell szerveznie a jobot alkotó fájlokat, illetve a jobhoz tartozó ClusterGrid specifikus leíró fájlt. A jobot alkotó fájlok entry gépre történő másolása SCP kliens programmal lehetséges.

#### 3.3.2 Job státuszának lekérdezése

A brókeren keresztül beküldött jobok futás közbeni állapotát lehet segítségével lekérdezni. Az eredményül kapott állapotjelző alapján a kliensprogram informálhatja a végfelhasználót, vagy önállóan a job futását befolyásoló döntéseket hozhat.

#### 3.3.3 Hibás futás okának lekérdezése

E funkció lehetőséget ad a kliensprogramnak (és rajta keresztül természetesen a végfelhasználónak), hogy egy korábban beküldött és időközben felfüggesztődött job hibás futásának okát kiderítse.

#### 3.3.4 Eredményfájlok visszahozatala

Miután egy job sikeresen lefutott a ClusterGrid brókere által kiválasztott erőforráson, e funkció lehetőséget ad a futtató erőforrásról a ClusterGrid entry gépre történő eredményfájl-áthozatalra. Mivel a ClusterGrid brókere nem követi nyomon az általa a különböző futtató erőforrásokra beküldött jobok állapotát, ezért arról sem értesül, hogy mikor melyik job futott le, melyik jobnak lehetne áthozni az eredményfájljait az entry gép fájlrendszerébe. Ennek a politikának a következménye, hogy az eredményfájlok visszamásolását a ClusterGrid kliensének kell kezdeményeznie.

Miután az eredményfájlok visszakérültek az entry gépre, ugyancsak SCP kliens program alkalmazható az állományok tetszőleges más gépre történő továbbítására.

## 4 A P-GRADE Portál ClusterGriddel együttműködni képes változata

A 2. fejezetben tárgyalásra kerültek mindazon kritériumok, melyeket a P-GRADE Portál alatt elterülő Grid infrastruktúrának teljesítenie kell. Az infrastruktúrára építve a Portál képes a végfelhasználóknak egy egyszerűen megérthető és kezelhető grafikus felületen keresztül workflow létrehozási, végrehajtási és monitorozási lehetőséget nyújtani. E fejezet ismerteti, hogy a ClusterGrid infrastruktúra mennyiben felel meg a P-GRADE Portál elvárásainak, továbbá definiálja mindazokat a fejlesztéseket melyek a komponensek integrálása miatt szükségesek.

A Portál alatt lévő Gridnek job végrehajtó, job szintű adatmozgatást megvalósító, továbbá job szintű trace összegyűjtő funkcionalitásokat kell nyújtania.

### 4.1 Workflow végrehajtás

A job végrehajtás és job szintű adatmozgatás a ClusterGridben is elérhető, viszont a funkciók igénybevételeire alkalmas ClusterGrid interfész különbözik annak GT2-es megvalósításától. Az eltérések a következők:

1. Amíg a GT2-ben a GRAM szervereken keresztül kell Condor beküldést és GridFTP hívásokat végrehajtani, addig a ClusterGridben egy brókert kell elérni a rendszer interfészül szolgáló Entry gép `clgr` hívásaival.
2. Amíg a GT2-ben a jobhoz futtató erőforrást, addig a ClusterGridben leíró adatokat kell társítani.

Mivel a job szintű funkcionalitás nem, csupán a „Grid interfésze” változik, ezért ezt az interfész változást a P-GRADE Portál job szintű feladat-végrehajtásért és adatmozgatásért felelős komponensei elrejthetik a magasabb szintű rétegek elől. Ezzel a módszerrel megtartható a jelenleg is használt workflow menedzselésért felelős logika, hiszen a workflow menedzselési szinten semmi sem változott. A megoldás tehát annyit jelent, hogy a 2. ábrán mutatott struktúrában a workflow vezérlés alatti réteget – a Condor-G-t és a GT2 specifikus pre és post scripteket le kell cserélni azok ClusterGrides megfelelőire.

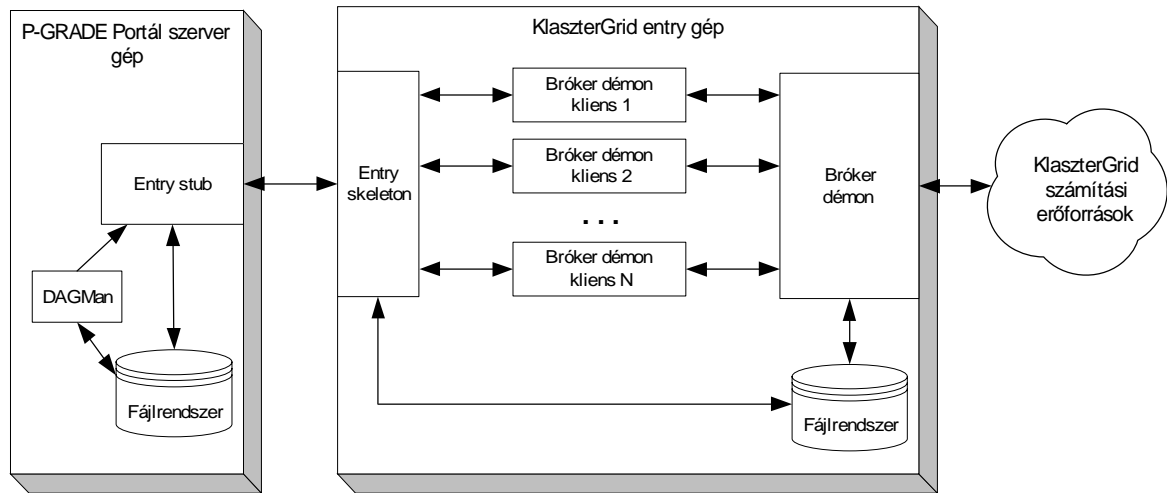
Ennek a feladatnak a konkrét implementálása jelenleg két elképzelést követ:

1. Az egyik modell esetén – kísérleti megoldásként- a P-GRADE Portál szerver a ClusterGrid UIF-jét jelentő Entry gépre kerül, így a DAGMan pre és post scriptjeiben lévő `clgr_` hívásokat a ClusterGrid közvetlenül értelmezi.
2. A későbbiek során alkalmazni kívánt általános megoldásban viszont a Portál Szerveren telepítésre kerül a ClusterGrid Entry gépet szimuláló proxy szerver (Entry stub), ami a DAGMan scriptjeinek átírása nélkül lesz képes a `clgr_` ClusterGrid vezérlő parancsok kiadására. (ld. 3. ábra)

Mindkét megoldásnak közös előnye, hogy a ClusterGrid Brókerrel való kommunikációhoz szükséges könyvtárfát, ami a job leíró, input, output és az autentikációhoz szükséges fájlokat tartalmazza, elégséges a P-GRADE Portál szerveren felépíteni.

A legfelső szinten tehát nincs változás. A DAGMan továbbra is pre scripteket, Condor jobokat és post scripteket indít el a neki átadott gráf által definiált sorrendben. A DAGMan alatti rétegekben viszont már érinti a P-GRADE Portált a ClusterGridhez való illesztés. Az adatmásolást végző oldalon a pre és post scriptek lecserélésére van szükség. Nyilvánvaló változás, hogy az új scripteknek nem GridFTP, hanem közönséges `cp` parancsokat kell használniuk a jobok input, output fájljainak másolásához. Lényeges változtatást kellett viszont alkalmazni a kommunikációs protokollok területén:

Amíg GT2 esetén a Condor G használata lehetővé tette, hogy a DAGMan értesüljön a job befejeződéséről és a post script el tudjon indulni, addig itt ezt a ClusterGrid esetén csak lekérdezéssel lehet megoldani, azaz a post script periodikusan fogja lekérdezni a `clgr_status` parancs segítségével a job állapotát.



3. ábra

### A ClusterGrid és P-GRADE Portál együttműködésében érintett rétegek

A job állapotok száma egy új feldolgozandó állapottal bővül, ami azt jelzi, hogy a ClusterGrid felfüggesztette az adott job futtatását az adott resource-on.

A feladat végrehajtásért felelős komponensek oldalán tehát az a különbség, hogy az új struktúrában a DAGMan már nem a távoli erőforrásra küldi be a workflow aktuális csomópontja által tartalmazott jobot, hanem a saját oldalán futó Condor segítségével futtatja azt. Ezt a működést nagyon egyszerű a DAGMan-ra „kényszeríteni”, csupán a „Globus” univerzum helyett más univerzumot kell az érintett job submit fájljában megadni. Mivel ezt a submit fájl a Portál generálja, a módosításnak semmi akadálya.

A ClusterGrid brókerén keresztül elérhető erőforrás-bróker funkcionalitás a Portál jelenlegi verziójában megtalálható „GT2 GRAM erőforrások beállítására” használható portletje feleslegessé válik, mivel a ClusterGrid-es portálban már nem a végfelhasználó, hanem a bróker az, aki az egyes jobokhoz végrehajtó erőforrást választ.

## 4.2 Workflow monitorozás

A ClusterGridből job szintű trace gyűjtést megvalósító funkció teljesen hiányzik. Ettől függetlenül a GT2-es Portálban workflow szintű monitorozásra használt megoldás változtatás nélkül alkalmazható lehet a ClusterGridhez illesztett Portál esetén is. Ennek oka, hogy a workflow szintű trace-t előállító hívások ugyanúgy belekerülhetnek a ClusterGrides Portál pre és post scriptjeibe ahogyan a GT2-es esetben.

A job szintű monitorozás megvalósításához mind a ClusterGriden, mind a Portálon változtatásokat kell alkalmazni:

### 4.2.1 A ClusterGridet érintő változtatások

1. Telepíteni kell a számítási szolgáltatást nyújtó erőforrásokra (clusterekre) a Mercury monitorozó rendszert, mégpedig oly módon, hogy az erőforrások vezérlő gépén - az úgynevezett n0 gépen – a Mercury Main Monitor és az adott erőforrás további gépein a Mercury Local Monitor fusson.
2. Az előző pontban említett n0 gépeket el kell látni egy olyan scripttel, ami lekérdezi az adott clusteren futó monitorozó rendszer állapotát.
3. Az előző pontban említett script hívását el kell helyezni abban a démonban, amely az n0 gépen, a ClusterGrid Bróker részeként ad rövid periódusonként (5 perc) információt az erőforrás állapotáról a ClusterGrid szerverre számára
4. A ClusterGrid szerverén az egyes erőforrások állapotát tartalmazó központi táblázatot ki kell egészíteni a Mercury állapotáról szóló információval.

5. Interfészt kell biztosítani, hogy a Portál szerver lekérdezhesse a központi táblázatban azt az információt, hogy van-e olyan élő erőforrás, ahol a Mercury fut.
6. A Brókernek a job beküldéskor figyelembe kell tudnia venni a P-GRADE Portálról az adott job monitorozására vonatkozó kiegészítő kérést: Azaz, ha van szabad monitorozható erőforrás, akkor azon próbálja elindítani a jobot, és ha nincs szabad, akkor várakoztatja, amennyiben a felhasználó monitorozást kért.
7. A ClusterGrid felhasználói felületét képező, és a P-GRADE Portál által is használt `clgr_status` parancs által szolgáltatott információ kiegészül annak nyújtásával, hogy az előzetesen beküldött job monitorozható erőforrásra (Cluster-re) vár-e, és ezért nem kerülhetett ott „Run” állapotba.

#### 4.2.2 A P-GRADE Portál szervert érintő változások

1. Át kell értelmezni a felhasználónak a monitorozásra vonatkozó kérését lehetőség szerint kielégíthető feltételes monitorozási kérésre.
2. Ha a felhasználó kérte a feltételes monitorozást, akkor a DAGman prescriptje folyamán, a 4.2.1 5. pontban leírt interfészen keresztül, a Portál szerver lekérdezi a ClusterGridtől, hogy van-e lehetőség a job monitorozására, és ennek megfelelően generálja a `clgr_submit` parancsot `MERCURY = {TRUE|FALSE}` attribútummal.
3. Amikor a DAGMan postscriptjének végrehajtása során – 4. 1-ben már említett lekérdezés révén - megtörténik a `clgr_status` vizsgálata és kiderül, hogy a monitorozandó job ütemezésre került egy adott clusteren, ez azt jelenti, hogy a job ténylegesen fenn fog akadni az adott cluster Mercury Main Monitorán. Ekkor a státusz információból a Portál számára azonosításra kerül a Bróker által futásra kiválasztott cluster, és a Portál a Brókert megkerülve most már közvetlenül végre tudja hajtani a cluster Main Monitorara való feliratkozást, aminek révén egyrészt a felhasználói job tényleges futása el tud indulni, másrészt megnyílik a monitorozó információkat a Portál szerverre eljuttató csatorna.
4. Amennyiben a 2. pontban említett monitorozó szolgáltatást nyújtó cluster nem áll rendelkezésre, akkor ennek tényét a Portál a futtatott jobhoz rendelt Standard Error fájlba való bejegyzéssel hozza a felhasználó tudomására.

#### 4.3 Hozzáférés szabályozás

Mivel a ClusterGrid és a GT2 erőforrás-hozzáférési politikája egymástól különböző, a P-GRADE Portálnak is változtatni kell a tanúsítvány alapú szolgáltatáshasználati modelljén.

A ClusterGrid esetén az elérés az entry gépre belépést biztosító felhasználói név és jelszó segítségével lehetséges. A KlszterGrides P-GRADE Portál két lehetőséget követhet:

1. Egyetlen ClusterGrides accountot használva küldi be az összes jobot.
2. Mindig az érintett Portál felhasználó ClusterGrides accountját használva küldi be a jobot.

Az első esetben a Portál teljes felelősséggel tartozik a ClusterGrid felé, ugyanis a KlszterGrid szemszögéből ekkor nem léteznek nevesíthető felhasználók, minden job egyetlen account, a P-GRADE Portál számára allokált accounttal érkezik. A P-GRADE Portál természetesen megbízható eseménynapló vezetés esetén a rá eső felelősséget továbbháríthatja a saját felhasználói felé.

A második politikát követve a P-GRADE Portál megmarad jelenlegi filozófiája mellett, vagyis hogy egy olyan grafikus felületet adjon, melyen keresztül a felhasználók a korábban szöveges felületen át használt Gridjeiket elérhetik.

A kísérleti megvalósítás során - amikor a Portál szerver az Entry gépen fut - az első megvalósítási utat választottuk. Tekintettel arra, hogy a ClusterGrid fejlesztése az utóbbi időben megengedte a tanúsítvány alapú autentikációt, a fejlesztés következő fázisában meg szeretnénk oldani a P-GRADE Portálhoz kapcsolt MyProxy szerverből kinyerhető tanúsítványok használatát.

#### 5 Következtetések és továbbfejlesztési irányok

A P-GRADE portál és ClusterGrid összkötésével nagy lépést tettünk abba az irányba, hogy a ClusterGrid felhasználók taborát jelentősen kiszélesíthessük. A P-GRADE portál segítségével a Grid technológiában és programozásban kevésbé járatos kutatók (pl. orvosbiológusok, kémikusok, stb.) is könnyen alkalmazni és kezelni tudják a ClusterGrid rendszert. Alacsonyszintű programozás helyett magas szintű workflow rendszerekben gondolkozhatnak, miközben a portál elrejtő elölük a ClusterGrid működésének belső részleteit, ahogy pl. a Windows operációs rendszer Office programcsomagja elrejtő a PC-k belső működését a végfelhasználók elől.

Ugyanakkor a P-GRADE portál lehetőséget kínál azon felhasználók számára is, akik a programvégrehajtásban maximális sebességet akarnak elérni a ClusterGrid több száz processzorának együttes működtetése révén. Számukra a portál támogatja mind a workflow szintű (klaszterek közötti), mind az egyedi job szintű (klaszteren belüli) párhuzamosság kihasználását is.

A kényelmes webes portálon keresztüli használat akár annak a lehetőségét is megteremti, hogy az ország különböző intézményei (elsősorban egyetemei) saját portált installáljanak, és azon keresztül biztosítsanak hozzáférést dolgozóiknak, hallgatóiknak, ügyfeleiknek Magyarország legnagyobb Grid infrastruktúrájához, a ClusterGridhez. A cikkben leírt portál egy általános ClusterGrid P-GRADE Portál, mely egy-egy intézmény egyedi igényei szerint bármikor és könnyen módosítható az igényeknek megfelelően.

Végezetül meg kell említeni, hogy a P-GRADE portál használatának talán legnagyobb előnye, hogy mind a két magyar Grid rendszerhez (ClusterGrid, HunGrid) hozzá lett kötve, így azon felhasználók, akik egyik Grid rendszerről a másikra akarnak áttérni, nem kell, hogy bármit is újra tanuljanak, mivel ugyanazt a felhasználói környezetet találják mind a két Grid rendszerben. Ráadásul alkalmazásukat sem kell módosítaniuk, az automatikusan hordozható lesz a két Grid között. Ezen túlmenően a P-GRADE portál most készülő multi-Grid verziója lehetővé teszi, hogy a felhasználó ne csak egy Gridet, hanem mindkét magyar Gridet egyszerre tudja elérni. Sőt miután a P-GRADE portál illetve van nemzetközi Grid rendszerekhez (SEEGRID, GridLab testbed), azok elérése is hasonlóan egyszerűen megoldható lesz.

## **KÖSZÖNETNYILVÁNÍTÁS**

A jelen munka megjelenése a következő pályázatok támogatásával készült: "IKTA6 OMFB 495/2004, OTKA T042459, IHM 40.036/2/2004

Külön szeretnénk köszönetünket kifejezni a NIIF ClusterGrid csapatának, és ezen belül munkánk létrejöttét tanácsokkal és az interfészek biztosításával lehetővé tévő Szalai Ferencnek és Stefán Péternek.

## **Referenciák**

- [1] P. Stefán: The Hungarian ClusterGrid Project, Proc. of MIPRO'2003, Opatija, 2003
- [2] KACSUK, P. – DÓZSA, G. - KOVCS, J. - LOVAS, R. - PODHORSZKI, N. - BALATON, Z. - GOMBS, G.: P-GRADE: A Grid Programming Environment.  
In: Journal of Grid Computing, Vol. 1. No. 2, pp. 171-197. 2004
- [3] NEMETH, CS. - DOZSA, G. - LOVAS, R. - KACSUK, P: The P-GRADE Grid portal. In: Computational Science and Its Applications - ICCSA 2004: International Conference, Assisi, Italy, 2004, LNCS 3044, pp. 10-19
- [4] [www.see-grid.org](http://www.see-grid.org)
- [5] Patvarczki J, Debreczeni G, Lovas R, Lagzi I, Kacsuk P, Turányi T: A HunGrid bemutatása és alkalmazása levegőszennyezés előrejelzésére. Workshop 2005, Szeged, Megjelenés alatt