

Condor, Globus Toolkit

Párhuzamos és Elosztott rendszerek II.

Dr. Mileff Péter

Condor Pool

- ◉ A Condor jellegzetessége az **univerzum** fogalom.
 - > a rendszerben különböző futtatási környezeteket takar: **Standard, Vanilla, PVM, MPI, Globus és Java.**
- ◉ A Standard univerzumban végrehajtott programok hibátűrők,
 - > mert a Condor rendszer állapotmentéseket (*checkpoint*) végez futás közben.
 - > Ha a program futása hardverhiba vagy a futtató gép tulajdonosának közreműködése miatt megszakad,
 - > akkor a lementett állapot segítségével a program futása egy azonos architektúrájú gépen folytatható tovább.

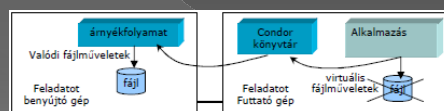
2

Standard univerzum

- ◉ A standard univerzum másik tulajdonsága:
 - > a rendszerhívásokat, a feladatot benyújtó távoli gépre viszi át (*remote system calls*)
 - > egy árnyékfolyamat, a **condor_shadow** segítségével.
- ◉ Ha a futtató gépen egy rendszerhíváshoz (például fájlművelethez) ér a program,
 - > akkor azt a távoli gépen futó **árnyékfolyamat** hajtja végre (a távoli gépen lévő fájl).
- ◉ A standard univerzum használata:
 - > a meglévő programokat újra kell fordítani, hogy képesek legyenek a fenti lehetőség kihasználására.
 - > Ilyenkor a program tárgykódjához a Condor a saját Condor könyvtárát szerkeszti hozzá.

3

Árnyékfolyamat működése



4

Standard univerzum

- ◉ A standard univerzum használata azonban megszorítással is jár:
- ◉ A legfontosabbak:
 - Nem futtathatunk több folyamatból álló programokat,
 - nem használhatunk néhány rendszerhívást (fork, exec, system)
 - és nem használhatjuk a Unix változatok biztosította folyamatok közötti kommunikációs lehetőségeket (szemaforok, unix csővezetékek).

5

Vanilla univerzum

- ◉ A Vanilla univerzum olyan programok futtatására használható, amelyeknél nincs lehetőség újrafordításra.
- ◉ Itt nincs lehetőség állapotmentésre és a rendszerműveletek átvitelére sem.
- ◉ Ez utóbbi miatt a program által használt fájloknak jelen kell lennie a futtató gépen is.
- ◉ Az elosztott fájlrendszerek használatával ez nem jelent problémát.
- ◉ A másik megoldás, hogy a feladatleíró fájlban megkérjük a Condort a fájl átvételére.

6

Condor univerzumok

- ◉ A PVM és az MPI univerzumok elosztott programok futtatására valók.
- ◉ A globus univerzum a Globus Toolkit nyújtotta lehetőségekre épít.
- ◉ A Java univerzum Java programok futtatására használható.
 - A Condor gondoskodik a Java futtatókörnyezet elindításáról és a CLASSPATH paraméter beállításáról.

7

DAGMan

- ◉ A Condor funkcionálisai közül kiemelendő a DAGMan,
 - melynek neve az irányított körmentes gráfból (Directed Acyclic Graph, DAG) származik.
 - Irányított körmentes gráffal jól leírható a feladatok egymásutánisága, függősége, azaz, hogy melyiket melyik után kell indítani.
 - A DAGMan ezt biztosítja.
 - Megadható neki, hogy mely feladatot mely másik vagy másiak lefutása után indítsa.
 - Különösen fontos együttműködő feladatok esetén, mint például a GRID computing-nál.

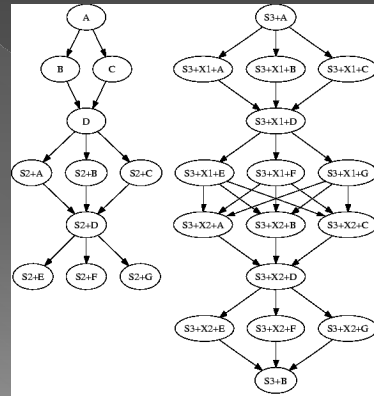
8

DAGMan

- Egy olyan adatstruktúra, amely a függőségeket ábrázolja.
- Minden feladat egy csomópont a gráfban,
- Bármely csomópontnak lehet akárhány szülő és akárhány gyermekcsomópontja, amíg a gráf **körmentes** marad.
- A DAG-ot egy .dag fájl definiálja, amely felsorolja az összes csomópontot és a függéseit.

9

• Példa DAG



10

GLOBUS TOOLKIT...

11

Bevezetés

- A Globus Toolkit egy nyílt forráskódú, nyitott architektúrájú köztesréteg szoftver számítási Grídek felépítésére.
- Argonne National Laboratory munkatársai kezdték fejleszteni,
 - > a jelenleg elérhető legújabb változat a 3.2 verziószámot viseli.
- A 3.0 verziótól az egész Globus infrastruktúra megújult.
 - > A szolgáltatások kibővített Web szolgáltatások (Web Service), úgynevezett grid szolgáltatások (Grid Services) formájában vannak jelen,
 - > amelyeket a Global Grid Forum által kiadott **OGSI** szabvány ír le.

12

Globus Toolkit

- A Globus rendszer alapjául szolgáló komponensek **három csoportba** sorolhatók:
 - > 1. az erőforrásokat kezelő komponensek,
 - > 2. az adatokat kezelő komponensek,
 - > 3. az információs szolgáltatások.

13

Globus pillérek

- Az első pillér a **grid erőforrások kezelését**, lefoglalását valósítja meg.
 - > Ehhez a pillérhez tartoznak a **GRAM** (Grid Resource Allocation Manager) és **DUROC** (Dynamically Updated Request Online Co-allocator) komponensek.
- A második pillér **az elosztott információs szolgáltatás**,
 - > amely a kliensek számára szolgáltat információt a grid elérhető erőforrásairól.
 - > Ehhez a pillérhez tartoznak az **MDS-I** (Meta-Directory Service) alkotó **GRIS** (Grid Resource Information Service) és **GLIS** (Grid Index Information Service) komponensek.

14

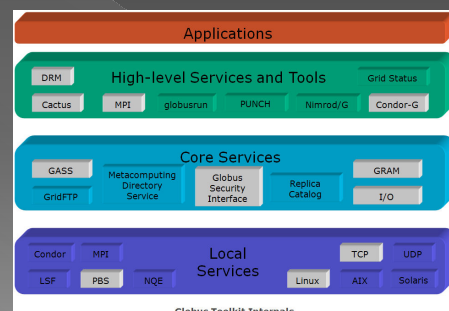
Globus pillérek

- A harmadik pillér feladata **az adatok elérése és kezelése grides környezetben**.
 - > Ide tartoznak a **GASS** (Global Access to Secondary Storage) és **GridFTP** komponensek.
- Mindhárom pillér alapjául szolgál a **GSI** (**Grid Security Infrastructure**) protokoll,
 - > amely a biztonságot hivatott biztosítani.

15

Globus Toolkit felépítése

- A Globus Toolkit-ot réteges felépítés jellemzi



16

Globus Toolkit felépítése

- ◉ A legalsó rétegekben helyezkednek az úgynevezett **helyi szolgáltatások**.
 - > Pl. TCP, MPI, stb.
 - > majd erre a rétegre épülnek a Globus magjának szolgáltatásai, azaz maga a **Globus**.
 - > Végül a harmadik, legfelső rétegben pedig magasszintű szolgáltatások és eszközök kapnak helyet.
 - > A Globus lehetővé teszi, hogy a rendszert más, már létező rendszerekkel használjuk. Ilyen például a **Condor-G**.

17

BIZTONSÁG A GLOBUS-BAN...

18

Biztonság a Globus-ban

- ◉ A számítási gridok nagy kiterjedésű, heterogén rendszerek,
 - > amelyek több adminisztrációs tartományt foglalnak magukba.
 - > A felhasználók és az erőforrások gyakran különböző adminisztrációs tartományokba esnek,
 - ezért **a biztonság kérdése alapvetően fontos**.
- ◉ A Globus Toolkit biztonsági mechanizmusát a **GSI** (Grid Security Infrastructure) protokoll adja

19

Biztonság a Globus-ban

- ◉ Létrehozásakor az elsődleges célok a következők voltak:
 - > Biztonságos kommunikáció biztosítása a számítási grid elemei között.
 - > Biztonság megvalósítása szervezeti határok mentén úgy,
 - hogy elkerülhető legyen egy központi elem bevezetése.
 - > A grid felhasználói számára **egyszeri bejelentkezés és delegáció** megvalósítása, az erőforrások használatakor.

20

Biztonság a Globus-ban

- ◉ A megvalósításához a GSI **publikus kulcsú titkosítást** alkalmaz
 - > az SSL protokoll feletti biztonságos kommunikációt és az X.509 tanúsítványokat használja fel.
- ◉ Mielőtt a felhasználó a grid bármely erőforrását használhatná, **igényelnie kell egy X.509 tanúsítványt (certificate)** a rendszertől.
- ◉ A tanúsítvány tartalmazza
 - > a felhasználó nevét,
 - > valamint a szervezetet, amelyhez tartozik.
- ◉ Az igényelt tanúsítványt ezután digitálisan alá kell írni
 - > egy arra hivatott szolgáltatással, a hitelesítési szervezettel (Certificate Authority).

21

Biztonság a Globus-ban

- ◉ Az aláírt tanúsítványok hosszú lejáratúak
 - > általában 5 évig érvényesek, lejárat után meghosszabbíthatóak.
- ◉ Az aláírt tanúsítvány meglete azonban még nem elég
 - > a grid használatához a felhasználónak be kell jelentkeznie a gridbe.
- ◉ A bejelentkezéskor egy rövid ideig – tipikusan fél napig – érvényes átmeneti tanúsítvány generálódik, amelyet **grid-proxy**-nak neveznek.
- ◉ A Globus rendszerben levő programokhoz futtatáskor hozzárendelődik ez a **grid-proxy**,
 - > s azok ennek segítségével igazolni tudják magukat az erőforrások használatakor. Ezt a folyamatot nevezik **delegációnak**.

22

GLOBUS ERŐFORRÁSKEZELÉS...

23

Globus erőforrások lefoglalása

- ◉ A Globus GRAM és DUROC komponense lehetőséget biztosít arra,
 - > hogy programokat futtassunk távoli erőforrásokon.
- ◉ Az erőforrások kezelését **két** egymásra épülő **réteg** valósítja meg:
 - > A felső réteg tartalmazza a magas szintű globális erőforrás kezelő szolgáltatásokat,
 - > az alsó a lokális erőforrásokat lefoglaló szolgáltatásokat.
- ◉ A lokális erőforráskezelő rétegben egy **GRAM Gatekeeper** nevű komponens várja az erőforrás foglalási igényeket.

24

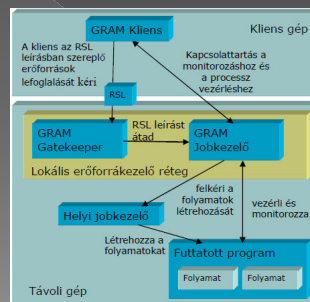
Globus erőforrások lefoglalása

- ◉ A klienseknek egy erre kifejlesztett nyelven, az **RSL** (Resource Specification Language) nyelven kell az igényeiket leírni,
 - > s ezt kell átadni a GRAM Gatekeeper szolgáltatásnak,
 - > ami továbbadja a helyi jobkezelő rendszerrel együttműködni képes **GRAM jobkezelő** komponensnek a kérést.
 - > A GRAM jobkezelő feldolgozza az RSL kérést,
 - > majd a megfelelő paraméterekkel meghívja a helyi folyamatkezelőt (pl. a Condor feladatkezelőt),
 - > ami létrehozza a folyamatokat.

25

Globus erőforrások lefoglalása

- ◉ Globus lokális erőforrás kezelő rétege



26

Globus erőforrások lefoglalása

- ◉ A kliensek tehát egy speciális, erőforrás specifikáló nyelv segítségével, az **RSL**-el tudnak erőforrásokat igényelni.
- ◉ Az **RSL** leírás egy egyszerű (*attribútum=érték*) párokból álló fájl. Legfontosabb attribútumai a következők:
 - > **executable:** a futtatandó program neve és helye, amely lehet fájlnev vagy URL
 - > **directory:** a futtatás könyvtára
 - > **arguments:** a program argumentumai
 - > **stdin:** a futtatandó program standard bemenetét adó fájl vagy URL
 - > **stdout:** a standard kimenetet az itt megadott fájlba menti le (lehet URL is)
 - > **stderr:** a standard hibakimenetet ebbe a fájlba menti le (lehet URL is)
 - > **count:** az itt megadott példányszámban indítja el a programot
 - > **jobtype:** itt a feladat típusát határozhatjuk meg

27

Globus erőforrások lefoglalása

- ◉ URL megadása esetén a Globus képes HTTP vagy FTP szerverről letölteni,
 - > illetve oda feltölteni a hivatkozott fájlt,
 - > de használhatjuk a Globus adatszolgáltató részéhez tartozó, ezen célra kifejlesztett **GASS** komponenst is.
- ◉ A kliensek képesek a GRAM jobkezelőn keresztül a létrehozott folyamatokkal kapcsolatot tartani.
 - > Pl.: lekérdezhetik a folyamatok állapotát és megszakíthatják a folyamatok végrehajtását is.

28

Globus erőforrások lefoglalása

- ◉ A legegyszerűbb GRAM kliensek a **globusrun** és a **globus-job-run** programok,
 - > amelyek a Globus Toolkit részét képezik.
- ◉ Az elsővel egy RSL leírás alapján hajthatjuk végre a feladatot,
- ◉ A másodiknál RSL leírás nélkül, parancssori paraméterek segítségével adhatjuk meg a végrehajtandó feladatot.
- ◉ Bonyolultabb GRAM kliensek írásához a **Globus programozói interfészt (API-t)** biztosít.

29

Globus erőforrások lefoglalása

- ◉ A grid rendszerek alapvető célja azonban összetett:
 - > több számítógépet használó alkalmazások futtatása.
- ◉ Ha egy kliens a számítási grid több gépén szeretné egyszerre futtatni programjait,
 - > akkor ezeket az erőforrásokat egyszerre kell lefoglalnia.
- ◉ Ennek a problémának a megoldásában nyújt segítséget a **Globus globális erőforrás-kezelő rétege**.

30

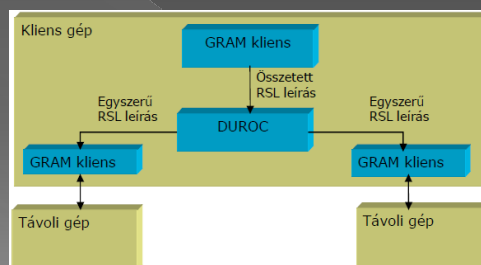
Globus erőforrások lefoglalása

- ◉ A fő szerepet itt a **Globus DUROC** komponense játssza,
 - > amely összetett RSL leírásokat képes több egyszerű RSL leírássá alakítani,
 - > majd az egyszerű leírások alapján az erőforrásokat egyszerre lefoglalni.
- ◉ Párhuzamos programok, nevezetesen MPI programok számítási gridekben történő futtatásához elengedhetetlen az erőforrások szimultán lefoglalása.

31

Globus erőforrások lefoglalása

- ◉ A Globus globális erőforrás-kezelő rétege



32

A Globus információs szolgáltatása

- A grid rendszerek használatakor a következő alapvető kérdések merülnek fel.
 - > Hogyan döntheti el az alkalmazás, hogy milyen erőforrások elérhetőek el?
 - > Hol vannak ezek az erőforrások?
 - > Mi ezeknek az erőforrásoknak az aktuális állapota?
- Ezeknek a kérdéseknek a megválaszolásához egy általános információs rendszerre van szükség,
 - > amelyet a Globus Toolkitben az **MDS (Meta-Directory Service)** szolgáltatás képvisel.

33

A Globus információs szolgáltatása

- A Globus első változataiban az MDS szerepét adminisztrációs tartományonként egy-egy központi **LDAP** szerver látta el.
- Mint minden központosított szolgáltatásnak, rossz hatása volt a rendszer teljesítményére,
 - > így szükségessé vált egy új változat tervezése.
- Az MDS második változatával egy elosztott, skálázható információs rendszert sikerült létrehozni.
 - > Az MDS-2 két szereplőből áll.

34

MDS 2 szereplője

- 1. **GRIS (Grid Resource Information Service)**,
 - > amely közvetlen az erőforrás mellett található,
 - > és magáról az erőforrásról ad információkat a klienseknek.
- 2. **GIIS (Grid Index Information Service)**,
 - > amely több erőforrásról tárol információt, temporális jelleggel (cache).

35

A Globus információs szolgáltatása

- A GRIS szolgáltatások saját információjuk terjesztéséhez:
 - > a **GRRP (Grid Resource Registration Protocol)** protokoll segítségével bejegyezhetik magukat egy vagy több GIIS szolgáltatásánál.
- Az információra szoruló kliensek a **GRIP (Grid Resource Inquiry Protocol)** protokoll segítségével feltehetik kérdéseiket a GIIS szervereknek,
 - > amelyek a tárolt információ alapján megadják a lehetséges erőforrások halmazát.

36

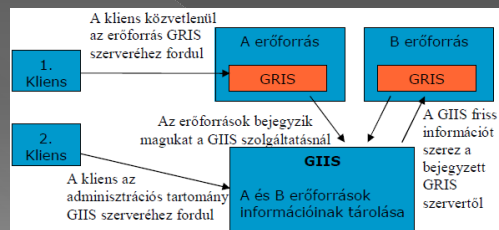
A Globus információs szolgáltatása

- Ha a kliens ismeri az erőforrás elérhetőségét,
 - akkor közvetlenül annak GRIS szolgáltatásától kérdezheti le az állapotot ugyancsak a GRIP protokoll segítségével.
- Nagyon hasonló a Webes keresőrendszerekhez:
 - ahol a beírt kifejezésre vonatkozó keresést a honlapok keresőrendszer által tárolt változatain végzik el,
 - majd a találatokat – amelyek lehetnek elavultak is – visszaadják a keresést kezdeményezőnek,
 - aki ellenőrizheti azokat.

37

A Globus információs szolgáltatása

Az MDS működése



38

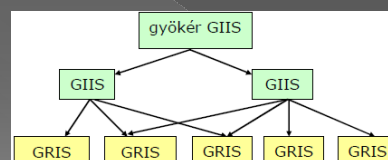
A Globus információs szolgáltatása

- A GIIS szolgáltatások **hierarchikus felépítésével** a kliensek a DNS rendszerhez hasonló módon kérdezhetik le az információkat a névszolgáltatástól.
- Ha a kliens ismeri a gyökér GIIS szerver címét, akkor az összes bejegyzett szolgáltatást megtalálhatja.

39

A Globus információs szolgáltatása

Hierarchikus erőforrás-kereső rendszer



40

