Dezvoltarea aplicațiilor pe middleware GRID

Prezentator: Tünde Bálint
Universitatea Tehnică din Cluj-Napoca

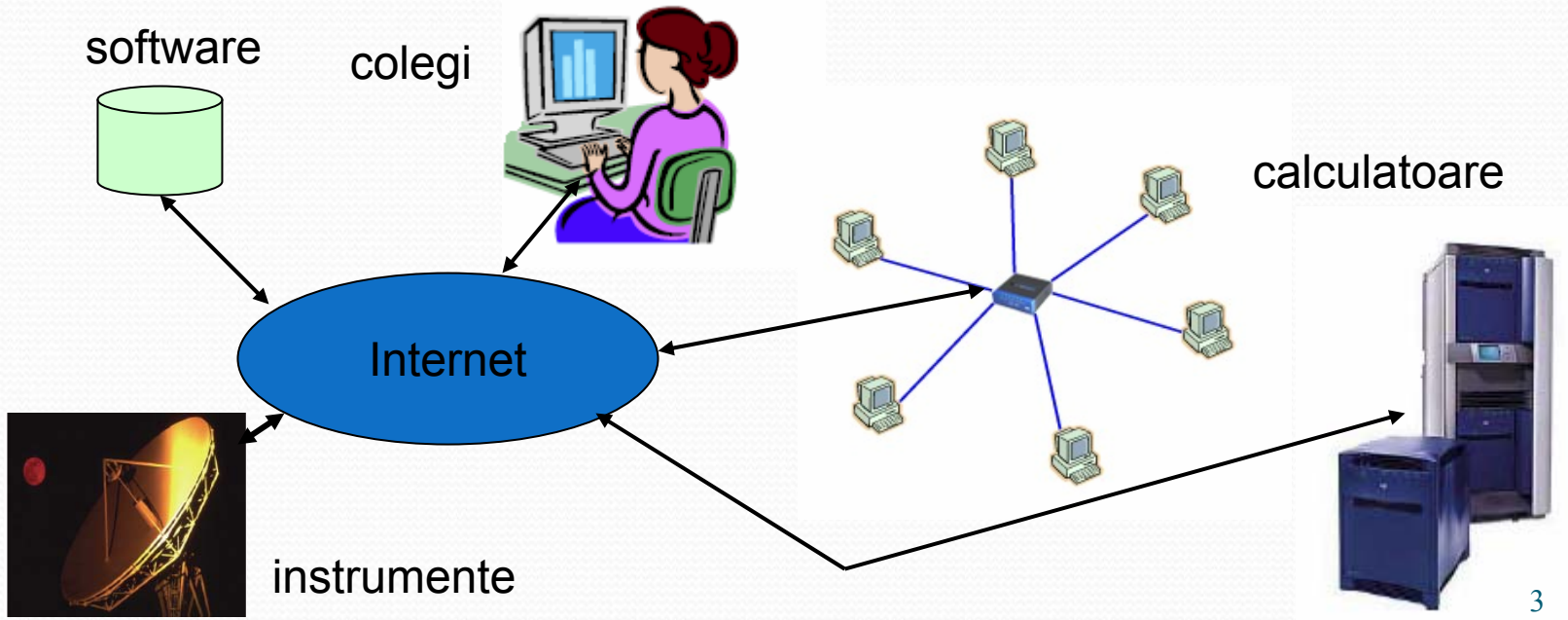
17.04.2008
Cluj-Napoca

Cuprins

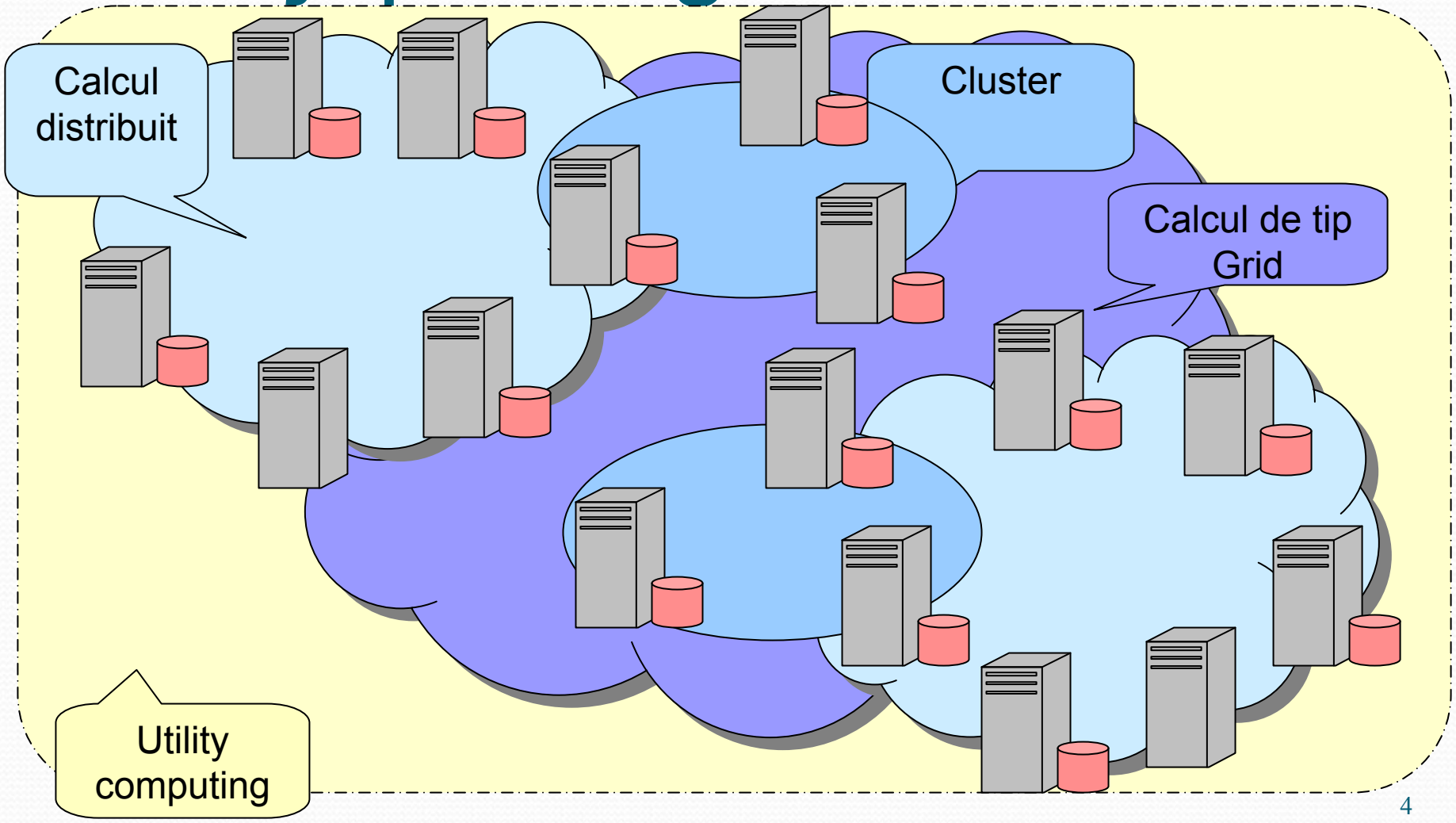
- Imagine de ansamblu : GRID
- Tipuri de aplicații
- Rularea job-urilor pe GRID
- Dezvoltarea aplicațiilor
  - Globus
  - CONDOR
  - gLite
- Manipularea datelor
  - Globus
  - gLite
  - OGSA-DAI
- Workflow-uri

Imagine de ansamblu

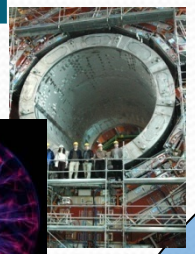
- Un sistem GRID constă din
  - Resurse eterogene distribuite, dar interconectate, nesupuse unui control centralizat
  - Software şi/sau hardware care asigură şi manipulează accesul la resursele necesare pentru a atinge un obiectiv

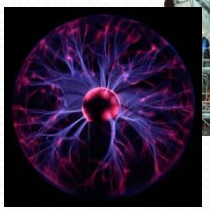


Grid şi paradigme înrudite



Aplicații





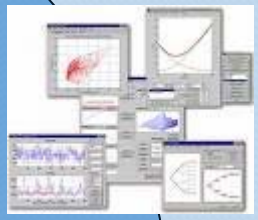
Fizica energiilor înalte

Calcul de performață înaltă

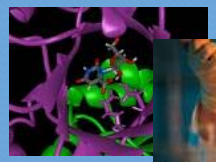
Utility computing

Design colaborativ

Modelare financiară



Automatizare a centrelor de informație

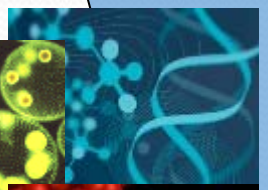


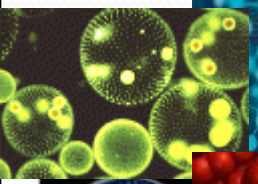
Biomedicină

E-science

Ştiințele pământului

E-business



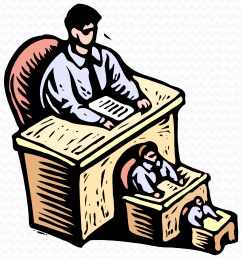


Biblioteci digitale

Distribuirea colaborativă a informațiilor

Prelucrarea limbajului natural & Data Mining

Context

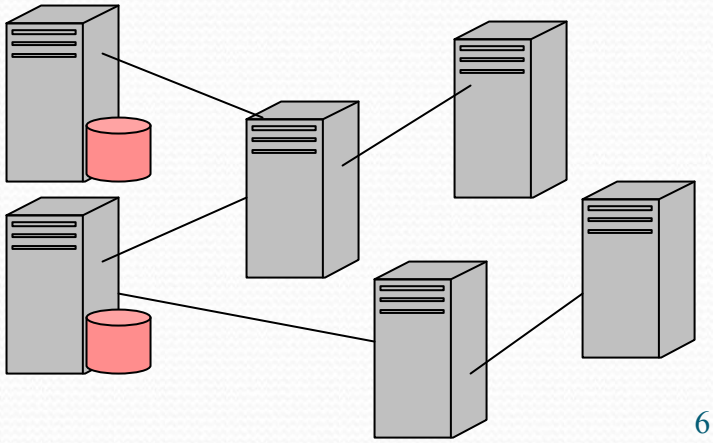


joburi /
executabile binare

Middleware

Mapare la resurse

Resurse interconectate



Paşii folosirii unui Grid

cod pentru rezolvarea problemei

compilare luând în considerare middleware-ul

lansare pe Grid (submit)

securitate

middleware

publică detalii (advertise)

transfer de date

deploy la resurse

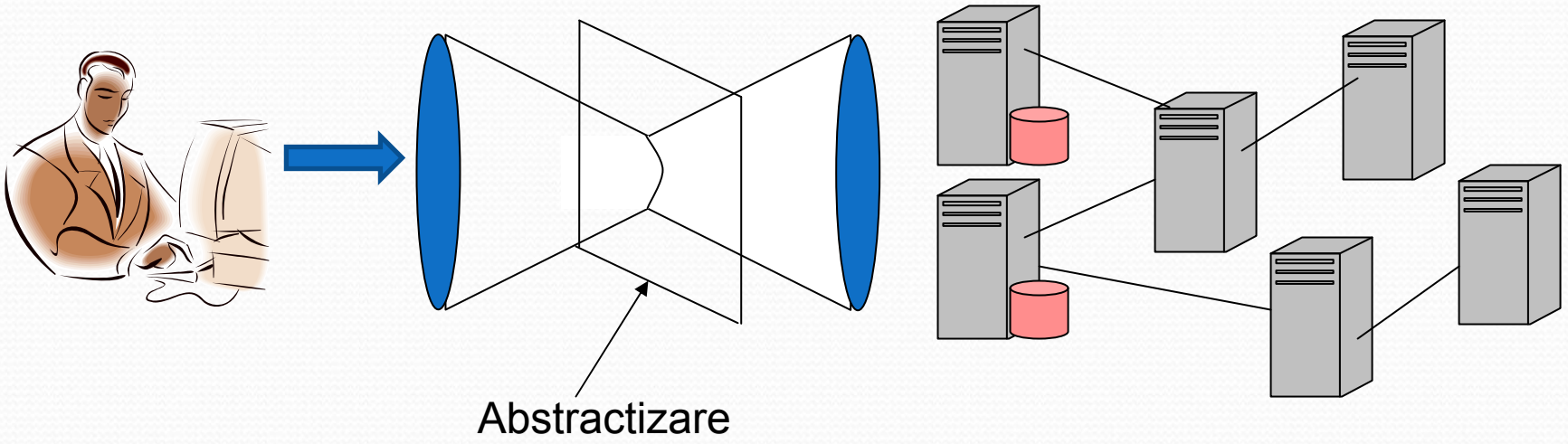
selecția resurselor

accounting

îndrumare şi vizualizare

Pentru a uşura viaţa

- Vrem să ascundem eterogenitatea Gridului



Sisteme Grid

- Globus
  - Rularea joburilor pe resurse aflate la distanță fără a avea nevoie de logarea la resursă
  - Folosit ca şi gateway la alte resurse
  - Face posibilă dezvoltarea iterativă a aplicațiilor şi instrumentelor (tools) Grid
- Condor
  - Rularea joburilor cu încărcare mare pe mai multe resurse (high throughput system)
  - Proiectat ca un middleware care fură ciclii de execuție
  - Execuția garantată a joburilor
  - Selectarea celei mai bune resurse pentru execuție
- gLite
  - Execuția joburilor pe mai multe resurse distribuite
  - Nu trebuie selectată resursa şi nu este necesară logarea pe ea

Dezvoltarea aplicațiilor pe Grid

- Specificare
- Lansare - submit
- Descoperire
- Selecție

- Securitate

Spec

- Abilita
  rulat
- Limba

Condor
Globus
gLite

```
[
Type = "job";
JobType = "normal";
VirtualOrganisation = "cms";
Executable = "test.sh";
Arguments = "1 20000 sim1";
StdInput = "file2";
StdOutput = "sim.out";
StdError = "sim.err";
OutputSandbox = {"sim.out", "sim.err"};
// disable job deep re-submission in case of failure
RetryCount = 0;
ShallowRetryCount = 5;
DataRequirements = {
[
InputData = {
"lfn:/mydata/file1", "lfn:/mydata/file2",
"guid:135b7b23-4a6a-11d7-87e7-9d101f8c8b70"
};
DataCatalogType = "RLS";
DataCatalog = "https://lcg.cern.ch/RLS";
],
...
```

Lansarea joburilor (submit)

- Mecanisme pentru a rula joburi pe Grid

Condor	- linie de comandă `% condor_submit test.submit` -servicii web Condor BirdBath -DRMAA Standard API

Globus	-linie de comandă
	`% globus-job-run <hostname> </path/executable> <arguments>`
	`% globus-job-submit <hostname> </path/executable> <arguments>`
	-servicii web
	•GT4 conţine implementări Java şi C pentru WSRF

gLite	-linie de comandă `% glite-wms-job-submit <JDL file>` -WMS client API - servicii web

Descoperire

- Procesul care descoperă resursele când acestea devin disponibile, respectiv când aceste resurse dispar
  - Trebuie să ştim ce resurse sunt disponibile pentru a face o selecție bună

Condor	Resursele trimit informațiile lor la matchmaker
Globus	Resursele trimit informațiile lor la un serviciu care poate fi interogat de scheduler
gLite	Resursele trimit informațiile lor la un serviciu de informare care poate fi interogat de WMS

Selecție

- Procesul care alege cea mai bună resursă pentru un job
  - Mecanisme care asigură că fiecare job este plasat pe resursa cea mai potrivită

Condor	Joburile sunt potrivite cu resursele Joburile vor rula când se găseşte o resursă disponibilă (idle) care satisface cerințele jobului
Globus	În majoritatea cazurilor selecția se face de către utilizator, specificând resursa
gLite	Workload Management Services sunt folosite pentru a alege cel mai bun CE la care se poate trimite jobul

Securitate

- Prevenirea folosirii inadecvate a resurselor
  - Autentificare şi Autorizare
- Trebuie dezvoltat un nivel de încredere atât pentru utilizatori cât şi pentru deținătorii de resurse
- Realizat cu X.509 şi Proxy
- gLite – organizații virtuale

Manipularea datelor

- este o problemă grea
- într-un mediu distribuit este şi mai greu:
  - Cataloage şi metadate
  - Controlul accesului
  - Consistență şi coerență
  - Revocare şi revizie (auditing)
  - Managementul replicilor
  - Eterogenitate
    - Datele sunt stocate pe diferite sisteme folosind diferite tehnologii de acces

Manipularea datelor - Globus

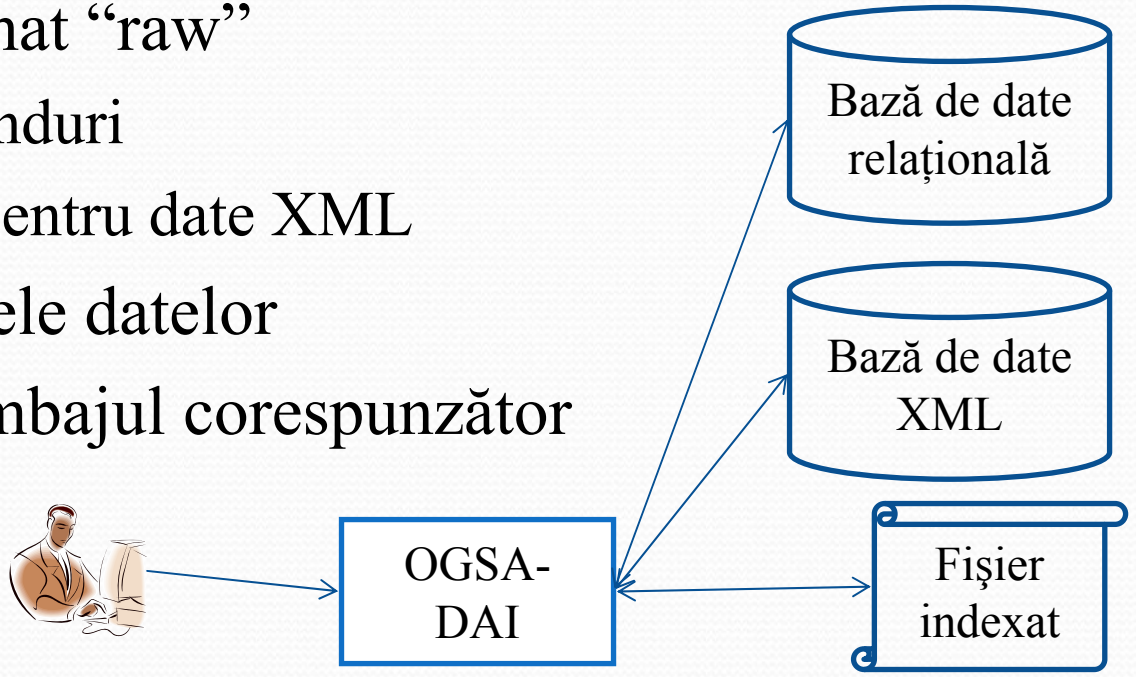
- GridFTP – protocol de transfer de date securizat, de încredere şi de performanță mare
- Serviciul Reliable File Transfer (RFT) – numărul transferurilor active, starea transferului, informații despre resursa pe care rulează serviciul
- Replica Location Service (RLS) – locația replicilor pe sisteme de stocare fizice pentru interogări

Manipularea datelor - gLite

- Protocolul Storage Resource Management (SRM)– managementul fişierelor şi a copiilor şi suport pentru rezervarea spațiului
- Serviciul Reliable File Transfer (RFT) – managementul unui număr mare de cereri GridFTP
- File Transfer Service (FTS) – manipulează cererile de transfer ale fişierelor şi suportă conceptul de “canal”

Manipularea datelor–OGSA-DAI

- Cu serviciile web OGSA-DAI putem partaja resurse de date structurate
- Se văd datele în format “raw”
  - Tabele, coloane, rânduri
  - colecții, elemente pentru date XML
- Se pot obține schemele datelor
- Se pot interoga în limbajul corespunzător
  - SQL
  - XPath



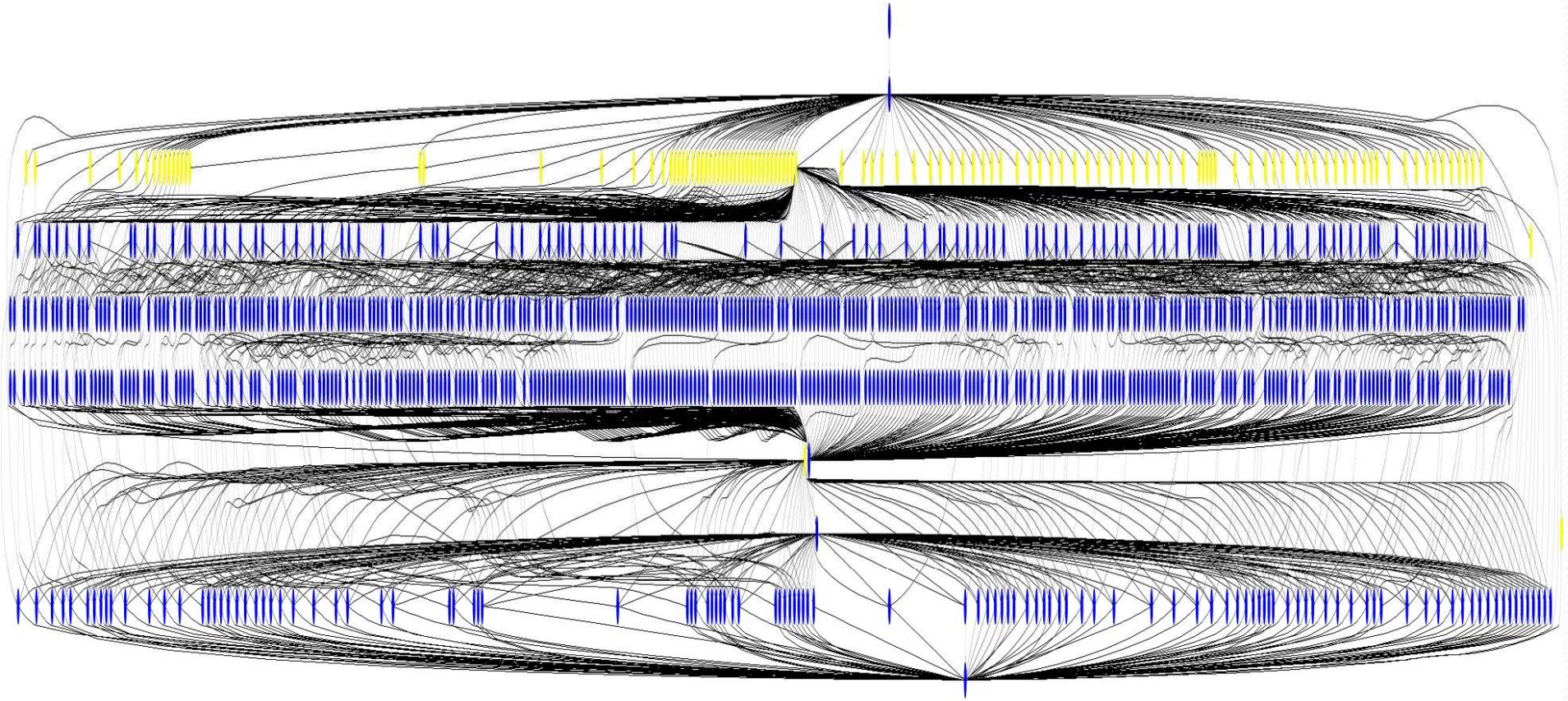
Manipularea datelor–OGSA-DAI

- Avantaje:
  - Se potriveşte cu abordarea de servicii web/grid
  - Serviciile web nu depind de limbaje de programare
- Dezavantaje:
  - Mai lent decât accesul direct datorită mesajelor SOAP
  - Introduce încă un nivel între client şi date
  - Datele nu sunt transferate în format binar

Workflow-uri

- Motivație :
  - Putem lansa joburi, să transferăm date
  - DAR aceste lucruri nu alcătuiesc o aplicație
- Ce sunt?
  - Mecanisme care leagă părțile unei aplicații într-un mod standard

Workflow cu 1200 de noduri



~1200 node workflow, 7 levels

Montage toolkit
http://montage.ipac.caltech.edu/

Mosaic of M42 created on the Teragrid using Pegasus

Reprezentare workflow

- Reprezentare vizuală - Grafuri
  - Noduri = joburi
  - Arce = dependențe
  - Dezavantaje:
    - Când avem un graf mare este greu de reprezentat şi de înțeles
    - Greu de exprimat unele caracteristici, de ex. Iterația
- Reprezentare gen limbaj de programare
  - limbaj specializat de programare - “scriptarea gridului”
  - Uşor de introdus concepte din limbaje de programare - variabile, cicluri, subrutine

Sisteme de workflowuri

- DAGMan - Directed Acyclic Graph Manager
  - un meta-scheduler pentru joburi Condor
- Pegasus - Planning for Execution in Grids
  - Maparea worflowurilor abstracte pe worflowuri concrete
  - Componente folosite:
    - Globus Monitoring and Discovery Service (MDS)
    - Globus Replica Location Service

Sisteme de workflowuri - 2

- Taverna – proiect European
  - Scop: ajutor în dezvoltarea şi execuția workflowurilor pe Grid in domeniul bioinformaticii
  - Modelele de date sunt reprezentate sau în format grafic sau într-un limbaj bazat pe XML numit Simple Conceptual Unified Flow Language (SCUFL)
- GridAnt, ICENI, GridFlow, Unicore, Triana, ...

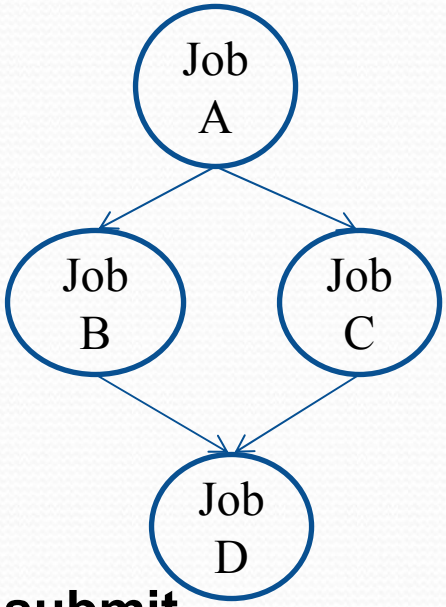
- Limbaje pentru workflowuri: BPEL, BPEL4WS, YAWL
  - Majoritatea bazate pe XML

DAGMan

- DAGMan coordonează seturi de joburi
  - Joburile sunt exprimate ca grafuri aciclice (nu există cicluri)
  - Lansează joburile cu Condor
  - Folosind loguri se pot trasa joburile
  - DAGMan se scalează bine (multe noduri)
  - Recuperare din erori
- Cu DAGMan descriem dependențele dintre joburi, Condor manipulează joburile automat

DAGMan

- Fiecare nod reprezintă un nod
- Dependențele (arcele) dau ordinea de execuție



#Fişierul de lansare (diamond.dag):

```
Job A a.submit
Job B b.submit
Job C c.submit
Job D d.submit
Parent A Child B C
Parent B C Child D
```

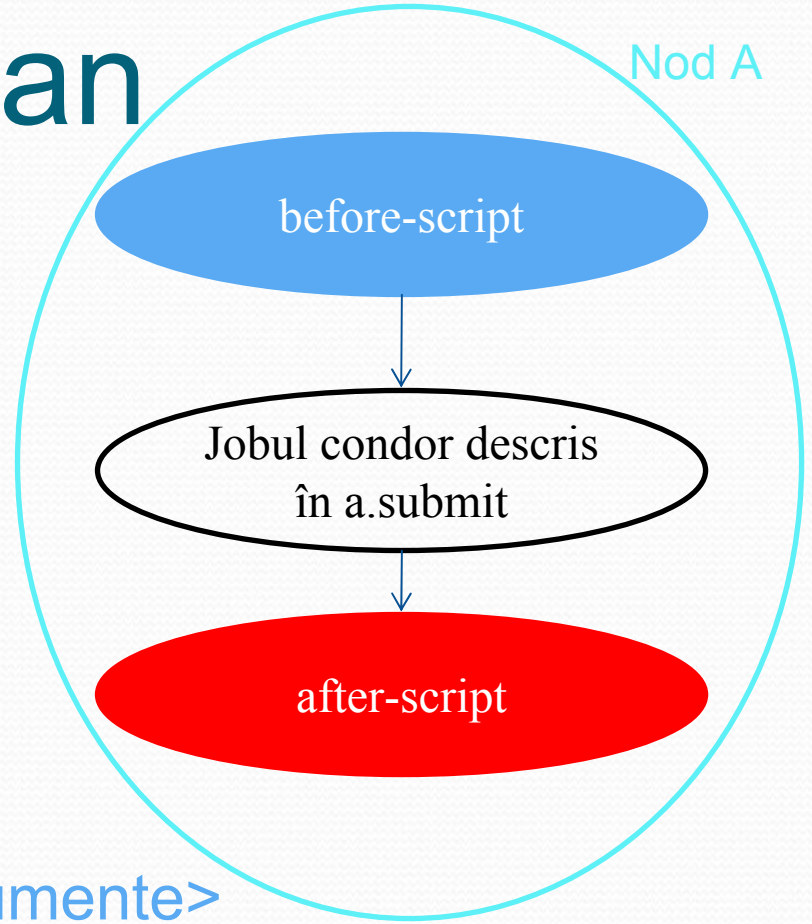
Lansare : condor_submit_dag diamond_dag

#File: b.submit

```
universe  = vanilla
executable = B
input = B.in
output  = B.out
error  = B.err
log = B.log
queue
```

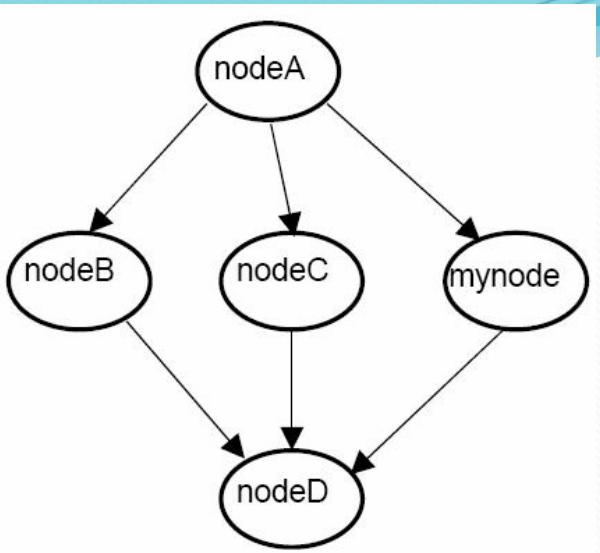
Scripturi DAGMan

- Putem avea scripturi de PRE şi/sau POST procesare rulate pe maşina de execuție



#Fişierul de lansare:
Job A a.submit
Script PRE A before-script <argumente>
Script POST A after-script <argumente>

DAGMan în gLite




```
[ Type = "dag";
  VirtualOrganisation = “gilda";
 InputSandbox = { "/tmp/foo/*.exe",
                  "/home/gliteuser/bar" };
max_nodes_running = 5;
nodes = [
    nodeA = [
            description = [
                    JobType = "Normal";
                    Executable = "a.exe";
                    InputSandbox = { "/home/data/myfile.txt"};
                            ];
             ];
    //la fel pentru nod B,C,D
dependencies = { { nodeA, nodeB }, { nodeA, nodeC }, {nodeA, mynode }, { {
    nodeB, nodeC, mynode }, nodeD } };
];
```

Workflowuri în OGSA-DAI

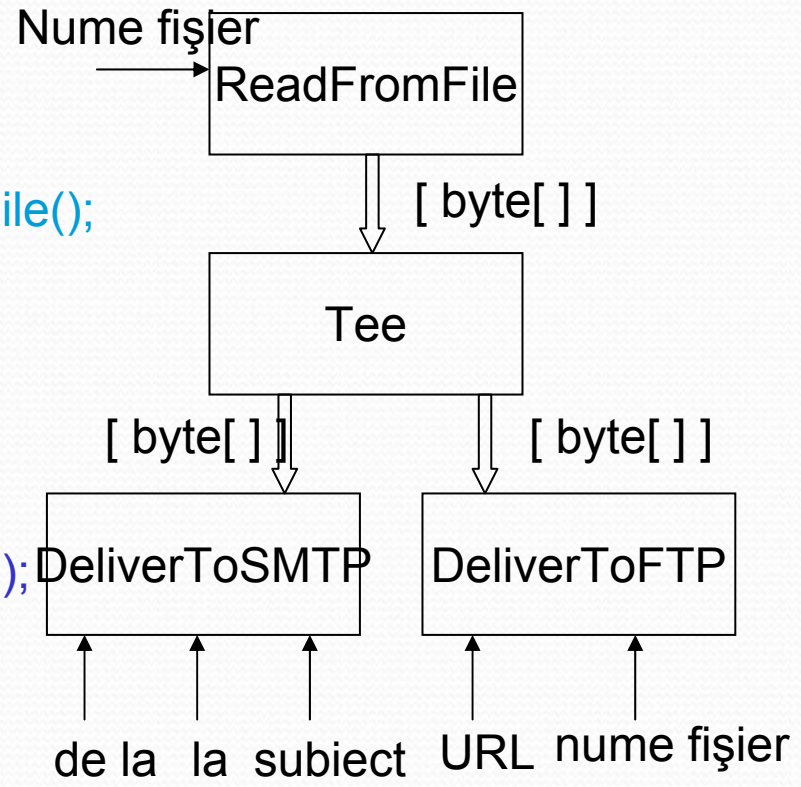
- Cererile sunt lanțuri de activități (=operații)
- Activitățile sunt instalate pe server
- Între activități avem streamuri de date
- Exemple de activități
  - SQLQuery – execută o interogare SQL pe o bază de date relațională
  - XSLTransform – execută o transformare XSL pe un document XML
  - DeliverToFTP/DeliverToSMTP – livrează date la un server FTP/SMTP
  - Tee – copiează valorile de intrare la toate valorile de ieşire
  - XQuery – executare de XQuery pe o resursă de date XMLDB

Workflowuri în OGSA-DAI - 1

```
PipelineWorkflow pipeline =
                    new PipelineWorkflow();

ReadFromFile readFromFile = new ReadFromFile();
readFromFile.setResourceID("FileResource");
readFromFile.addFile("results1.txt");
pipeline.add(readFromFile);

Tee tee = new Tee();
tee.connectInput(readFromFile.getDataOutput());
tee.setNumberOfOutputs(2);
pipeline.add(tee);
```



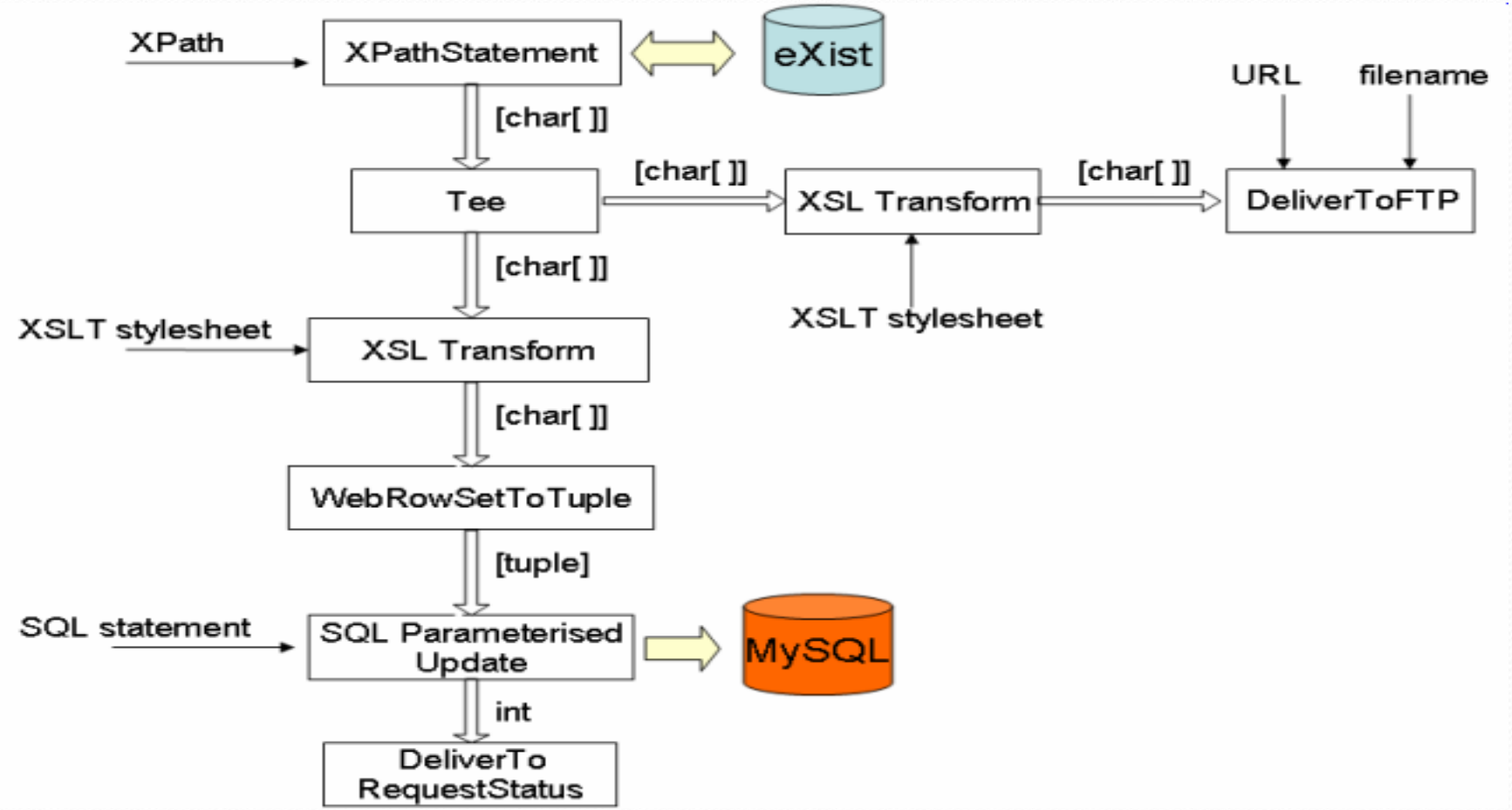
Workflowuri în OGSA-DAI - 2

```
DeliverToFTP deliverToFTP = new DeliverToFTP();
deliverToFTP.connectDataInput(tee.getOutput(0));
deliverToFTP.addFilename("/incoming/results_user00.txt");
deliverToFTP.addHost("anonymous:anonymous@tc07.nesc.ed.ac.uk");
deliverToFTP.addPassiveMode(true);
pipeline.add(deliverToFTP);

DeliverToSMTP deliverToSMTP = new DeliverToSMTP();
deliverToSMTP.connectDataInput(tee.getOutput(1));
deliverToSMTP.addFrom("youremail@example.com");
deliverToSMTP.addSubject("OGSA-DAI Test");
List to = Collections.singletonList("youremail@example.com");
deliverToSMTP.addTo(to.iterator());
pipeline.add(deliverToSMTP);

RequestResource requestResource =
                    drer.execute(pipeline, RequestExecutionType.SYNCHRONOUS);
System.out.println(requestResource.getRequestStatus());
```

Workflowuri în OGSA-DAI - 3



Rezumat

- Probleme:
  - partajarea resurselor folosind comunități virtuale dinamice
  - facilitarea colaborării prin partajarea resurselor şi a informațiilor;
  - asigurarea condițiilor de securitate, încredere, fiabilitate, contabilizare (înregistrare), manevrabilitate şi agilitate
  - calcule şi date la scară mare (extreme);
- Rularea unui job pe infrastructuri GRID
  - Specificare, lansare, descoperire, selecție
- Manipularea datelor
  - OGSA-DAI
- Workflowuri
  - DAGMan
  - OGSA-DAI

Mulțumesc pentru atenție!
Întrebări...