

Componente interactive în cadrul unui mediu de dezvoltare a aplicațiilor Grid

Victor Băcu

Universitatea Tehnică Cluj Napoca

Str. G. Barițiu 26

victor.bacu@cs.utcluj.ro

Dorian Gorgan

Universitatea Tehnică Cluj Napoca

Str. G. Barițiu 26

dorian.gorgan@cs.utcluj.ro

REZUMAT

Gradul de utilizabilitate al aplicațiilor Grid de către specialiști din alte domenii de activitate decât cel al științei calculatoarelor este oarecum scăzut. Acest lucru se datorează inexistenței la nivelul platformelor Grid a unor componente interactive care să transparentizeze accesul la infrastructura Grid. Marea majoritate a aplicațiilor Grid se pot utiliza doar prin intermediul scripturilor din linia de comanda. Un mediu de dezvoltare a aplicațiilor Grid trebuie să ofere componente interactive pentru managementul datelor și al fluxurilor de procese pe Grid cât și componente interactive pentru dezvoltarea și includerea de noi operatori. În această lucrare se prezintă o arhitectură a unui astfel de mediu de dezvoltare punându-se accent pe tehnicile de interacțiune și pe componentele interactive.

Cuvinte cheie

Aplicații Grid, Medii de dezvoltare, Tehnici de interacțiune.

INTRODUCERE

Tehnologia Grid pune la dispoziție un mediu de execuție și o infrastructură pentru utilizarea resurselor distribuite, fie că acestea sunt reprezentate de date, de componente software sau de componente hardware. Problema care apare și care este rezolvată la nivelul de middleware Grid este legată de partajarea resurselor într-un mod flexibil, sigur și coordonat. Accesul la tehnologia Grid trebuie să fie transparent și să se realizeze printr-o serie de aplicații, middleware-uri Grid și unelte software. Unele dintre avantajele folosirii fluxurilor de procese provin din posibilitatea de a defini rapid diferite prototipuri sau a reutiliza anumite fluxuri de procese în procesul de definire și realizare al diferitelor experimente științifice.

Principalul middleware Grid, gLite [1], oferă funcționalitatea de bază utilizând interacțiuni simple și nebazate pe interfețe grafice utilizator. Din acest motiv în cadrul diferitelor proiecte se dezvoltă componente interactive de interacțiune între utilizator și un sistem Grid. Mediul Taverna [3] este o aplicație care utilizează o interfață grafică utilizator pentru a permite bioinformaticienilor crearea, adaptarea și rularea fluxurilor de procese, precum și gestionarea datelor și metadatelor. GANGA [2] este un frontend pentru definirea și managementul joburilor Grid și este implementat în Python. P-GRADE [7] este un portal Grid utilizat pentru dezvoltarea, execuția și monitorizarea fluxurilor de procese. Acest portal ascunde accesul la infrastructura

Grid prin intermediul interfeței grafice utilizator. În acest mod chiar și un nespecialist în rețele Grid poate defini și executa fluxuri de procese. În această lucrare se propune o arhitectură a unui mediu de dezvoltare al aplicațiilor Grid punându-se accent pe interacțiunea dintre utilizator și sistemul Grid.

MEDIU DE DEZVOLTARE AL APLICATIILOR GRID

Tehnicile de interacțiune dintre un utilizator și un sistem Grid nu sunt adecvate tuturor categoriilor posibile de utilizatori. Este nevoie de cunoștințe legate de sistemul de operare Linux, de folosirea unui limbaj bazat pe script-uri, etc. În plus pentru a putea folosi o astfel de rețea utilizatorul trebuie să aibă acces la o stație de tip User Interface (UI) existentă în gLite. Însă instalarea unei asemenea componente pe un calculator nu este deloc un proces foarte ușor pentru un nespecialist în domeniul calculatoarelor și mai precis în rețele Grid. Nu trebuie confundat acest tip de interfață UI cu GUI deoarece nu există o interfață grafică utilizator, modul de interacționare cu sistemul făcându-se prin intermediul unei interfețe de tip linie de comandă. Chiar și după instalarea unei asemenea componente utilizatorul se va lovi de o serie de probleme, cum ar fi generarea certificatelor proxy pentru a putea utiliza resursele Grid, definirea procesului care va fi executat într-un limbaj specific de descriere, etc. În plus marea majoritate a platformelor Grid fiind încă în faza de dezvoltare conțin și o serie de bug-uri pentru care trebuie căutată o soluție de rezolvare. Din acest motiv sistemul trebuie să ofere un punct de acces către resursele și funcționalitățile Grid în funcție de tipul utilizatorului. S-au identificat două categorii de potențiali utilizatori, fiecare dintre aceste două categorii conține specialiști în anumite domenii de activitate. Din acest motiv fiecare grup de utilizatori va avea o vedere personalizată asupra sistemului și asupra funcționalităților disponibile.

Un mediu de dezvoltare a aplicațiilor distribuite pe Grid înglobează un ansamblu de servicii care se adresează unor diferite categorii de utilizatori. Aceste servicii oferă o platformă de dezvoltare al aplicațiilor și deasemenea un mediu de management al diferitelor funcționalități oferite de către Grid. O caracteristică importantă a acestui mediu este flexibilitatea în dezvoltare și execuție a proceselor sau fluxurilor complexe pe Grid. Mediul de dezvoltare a aplicațiilor Grid se bazează pe gProcess și ESIP [4], [5]. În [6] a fost prezentat un exemplu de dezvoltare a unei aplicații pe Grid folosind platformele gProcess și ESIP.

Funcționalitatea de bază oferită de un astfel de mediu cuprinde următoarele:

- Modul de autentificare (folosind diferite metode: certificate, acces la serverul VOMS, etc.)
- Management al datelor (trebuie oferit un management atât al datelor locale cât și al celor distribuite)
- Management al fluxurilor de procese
- Acces la informațiile legate de resursele disponibile în Grid (acces la Computing Element, la Storage Element, etc.)
- Suport pentru dezvoltarea componentelor fluxurilor de procese
- Planificarea execuției fluxurilor de procese
- Optimizarea structurii fluxurilor de procese în vederea micșorării timpului de execuție
- Suport pentru mecanismul de prevenire și tratare a erorilor
- Monitorizarea execuției și realizarea de statistici
- Unelte de vizualizare a datelor de intrare, a datelor de ieșire, a statisticilor execuției, etc.

DESCRIEREA CATEGORIILOR DE POSIBILI UTILIZATORI AI MEDIULUI DE DEZVOLTARE A APLICAȚIILOR GRID

Așa cum a fost specificat anterior, un astfel de mediu se adresează diferitelor categorii de utilizatori. Pentru fiecare categorie de utilizatori interfața grafică utilizator trebuie personalizată astfel încât utilizabilitatea sistemului să fie maximă.

Specialiști în rețele Grid

Această categorie de utilizatori este axată pe dezvoltarea de componente, management al resurselor, dezvoltarea de unelte de vizualizare. Componentele accesibile pentru această categorie de utilizatori sunt:

- Managementul datelor;
- Acces la informațiile legate de resursele disponibile în Grid (acces la Computing Element, la Storage Element, etc.);
- Suport pentru dezvoltarea componentelor fluxurilor de procese;
- Monitorizarea execuției și realizarea de statistici;
- Unelte de vizualizare a datelor de intrare, a datelor de ieșire, a statisticilor execuției, etc.

Specialiști în diferite domenii de cercetare

Utilizatorii cărora le este adresată aplicația sunt specialiștii din diferite domenii de cercetare. Componentele dezvoltate vor fi axate pe acele domenii de activitate. În principiu acești utilizatori vor folosi o parte din funcționalitatea oferită de către sistem. Anumite componente, cum ar fi cele de acces la informații legate de resursele disponibile în Grid, nu vor fi accesibile în mod normal acestei categorii de utilizatori, ci doar dacă se dorește acest lucru. Componentele care sunt accesibile acestei categorii de utilizatori cuprind:

- Managementul datelor;
- Integrarea de componente pentru fluxurile de procese;
- Management al fluxurilor de procese;

- Monitorizarea execuției și realizarea de statistici;
- Unelte de vizualizare a datelor de intrare, a datelor de ieșire, a statisticilor execuției, etc.

SCENARIUL DE UTILIZARE

Principalele funcționalități ale sistemului cu care utilizatorii vor interacționa sunt acelea de management al fluxurilor de procese (atât cele abstracte cât și cele concrete sau instanțiate), integrarea diferitelor componente care alcătuiesc un flux complex de procese (respectiv resurse, operatori, subgrafuri, servicii), și de asemenea execuția și monitorizarea fluxurilor instanțiate de procese. Alte componente, cum ar fi componenta de identificare și tratare a erorilor sunt transparente pentru utilizator, în aceeași situație aflându-se și componenta de execuție a fluxurilor de procese (ciclul de execuție al proceselor componente ale unui flux complex de procese).

Integrare flux de procese abstract (PDG)

Fluxurile abstracte de procese reprezintă un șablon care va fi mai apoi instanțiat la resurse specifice de intrare, de exemplu imagini satelitare, sau date meteorologice, sau chiar tipuri primare de date de tip int, float, string. Procesul de integrare al unui flux abstract de procese constă din următoarele etape:

- Dezvoltarea fluxului abstract de procese;
- Generarea reprezentării în limbajul PDGL (Process Description Graph Language [4]);
- Validarea nodurilor;
- Validarea structurii;
- Integrarea PDG-ului în baza de date gProcess.

Instanțiere flux de procese

Fluxul instanțiat de procese reprezintă un executabil pe Grid. Dintr-un flux abstract de procese pot fi definite o serie de fluxuri instanțiate de procese. Procesul de instanțiere constă în specificarea pentru fiecare resursă de intrare a locației (locală sau distribuită) acelei date. Procesul de instanțiere și de integrare a unui flux instanțiat de procese (iPDG) constă din următoarele etape:

- Se alege din baza de date fluxul abstract de procese care va fi instanțiat;
- Pentru fiecare resursă se specifică locația (locală sau distribuită);
- Verificarea instanțierii tuturor nodurilor din fluxul de procese;
- Validarea iPDG-ului;
- Integrarea în baza de date gProcess a iPDG-ului.

Integrare operatori

Operatorii reprezintă elementele de bază, alături de resurse, pentru definirea unor fluxuri de procese complexe. Arhitectura gProcess permite integrarea de noi operatori la cei deja existenți.

Pentru integrarea unor noi operatori mecanismul este relativ simplu și constă din următoarele etape:

- Se crează o nouă clasă care extinde o clasă Operator de bază;

- Se implementează funcționalitatea dorită folosind inclusiv librării externe;
- Se testează și se validează operatorul local folosind date de test;
- Se integrează operatorul în baza de date;
- Se dezvoltă un flux abstract de procese care conține operatorul nou creat, rezultând astfel un PDG;
- Se instanțiază acest flux abstract de procese cu niște date de test, rezultând astfel un iPDG;
- Se execută iPDG-ul pe Grid pentru testare și validare.

Este importantă testarea unui nou operator atât local cât și pe Grid. În principiu dacă un operator funcționează corect local atunci și pe Grid va trebui să returneze același rezultat. Dar în cazul în care se folosesc librării adiționale de către operator există posibilitatea ca pe mașina din Grid pe care va ajunge să se execute operatorul, dintr-un anumit motiv, execuția să eșueze. Din acest motiv, înainte de face public un nou operator acesta trebuie testat atât local cât și pe Grid. În cazul majorității utilizatorilor sistemului interacțiunea dintre utilizator și sistemul Grid va fi minimă, majoritatea componentelor executându-se într-un mod semiautomat.

Interfața grafică utilizator care implementează acest scenariu de utilizare va presupune o interacțiune de tip vrăjitor (wizard) și în care utilizatorul, pas cu pas, va realiza toți pașii prezentați anterior. În funcție de categoria din care face parte utilizatorul anumite operații vor fi efectuate transparent (de exemplu primul pas care presupune anumite cunoștințe de programare) sau vor putea fi realizate explicit de către utilizator.

Integrare subgrafuri

Subgrafurile reprezintă un caz particular de fluxuri abstracte de procese. Descrierea lor în limbajul PDGL conține atribute adiționale față de descrierea unui PDG. Un subgraf va putea fi folosit după integrarea în baza de date gProcess în descrierea altor fluxuri de procese. Procesul de integrare a unui subgraf constă din următoarele etape:

- Dezvoltarea grafului ca un flux abstract de procese;
- Generarea reprezentării în limbajul PDGL (Process Description Graph Language);
- Validarea nodurilor;
- Validarea structurii;
- Integrarea subgrafului în baza de date gProcess.

Integrare resurse

Resursele reprezintă datele de intrare pentru nodurile din descrierea unui PDG sau iPDG. Fiecare resursă are atașată în baza de date un tip. Pe baza acestui tip de resurse se realizează validarea legăturii între două noduri dintr-un PDG sau iPDG. Adăugarea de noi resurse presupune definirea de către utilizator a tipului și a formatului resursei. Tipul reprezintă categoria, de exemplu imagine sau fișier de temperatură. Pentru fiecare tip de resursă pot fi disponibile mai multe formate, de exemplu pentru o imagine formatul poate fi TIFF, JPEG sau BMP.

Execuția fluxului instanțiat de procese

Așa cum a fost specificat anterior doar fluxurile instanțiate de procese pot fi executate. Etapele execuției unui iPDG sunt:

- Validare iPDG;
- Expandarea subgrafurilor;
- Generarea unei reprezentări interne a grafului de descriere;
- Identificarea nodurilor și regăsirea din baza de date a diferitelor informații necesare execuției;
- Execuția nodurilor de tip operator și de tip serviciu;
- Verificarea stării execuției până în momentul terminării execuției tuturor nodurilor din descriere iPDG;

Din punctul de vedere al utilizatorului acest proces este transparent. Interacțiunea dintre utilizator și sistem făcându-se doar la început pentru selectarea fluxului de procese care va fi executat și definirea unor parametri necesari execuției.

Monitorizarea unui flux de procese lansat spre execuție

Un utilizator are posibilitatea de a monitoriza în orice moment orice iPDG care se execută. Din punctul de vedere al utilizatorului monitorizarea presupune vizualizarea unor rapoarte, grafice, etc. Pe baza acestor informații el poate lua anumite acțiuni. Din punctul de vedere al sistemului monitorizarea presupune transmiterea unor parametri de monitorizare către componenta care gestionează informațiile de monitorizare.

MODULUL DE AUTENTIFICARE ȘI AUTORIZARE

Pentru a putea folosi o resursa Grid un utilizator trebuie să aibă dreptul de a o utiliza. Într-o rețea Grid fiecare utilizator face parte dintr-o Organizație Virtuală (VO). O astfel de Organizație Virtuală deservește o mulțime de utilizatori care partajează între ei resurse (de calcul, de stocare, date, algoritmi, etc.). Primul pas care trebuie făcut de către utilizator este autentificarea. Pe baza autentificării va putea utiliza resursele partajate împreună cu ceilalți membri ai unui VO. Securitatea la nivelul VO presupune delegarea certificatului utilizator către servicii sau procese. Delegarea permite proceselor sau serviciilor să se autentifice în contul utilizatorului. Acest lucru se realizează prin crearea unui certificat proxy. O altă caracteristică este limitarea de operații care poate fi făcută cu ajutorul lui, și deasemenea are un timp de durabilitate limitat.

În cazul dezvoltătorilor și a specialiștilor se pot genera certificate individuale care vor fi folosite pentru autentificarea și autorizarea pe Grid. Există alte situații, de exemplu în care se utilizează aplicația ca un mediu de învățare (de exemplu lecții) și în care ar fi dificil de creat și de gestionat certificatele fiecărui student care folosește aplicația. O soluție în acest caz ar fi autentificarea acestor utilizatori la nivelul aplicație (prin username și parola de exemplu) iar fiecare dintre acești posibili utilizatori să folosească pentru autentificare la nivelul Grid un certificat predefinit. Evident se poate opta pentru o limitare a

posibilităților de utilizare a rețelei Grid de către acești utilizatori la nivelul aplicației, de exemplu nu se pot lansa simultan mai mult de n procese.

Se întâlnesc următoarele cazuri:

- Certificatele utilizator se găsesc la nivelul platformei - În acest caz certificatul proxy se poate genera pe baza certificatelor utilizator, autentificarea și autorizarea utilizând acest certificat proxy; Categoriile de utilizatori care fac parte din aceasta situație: dezvoltatorii de aplicații, o parte din specialiștii care utilizează aplicațiile, testerii aplicațiilor;
- Certificatul proxy este stocat pe un server MyProxy - În acest caz utilizatorul nu are stocate certificatul la nivelul platformei, în schimb utilizatorul a delegat în prealabil certificatul proxy generat pe baza certificatului propriu pe un server MyProxy, folosind datele de acces oferite de către utilizator aplicația va putea folosi acel certificat proxy pentru autentificare și autorizare; Categoriile de utilizatori care fac parte din această situație: dezvoltatorii de aplicații, o parte din specialiștii care utilizează aplicațiile, testerii aplicațiilor;
- Utilizatorul nu are un certificat Grid propriu, în acest caz pentru autentificare și autorizare la nivelul Grid se folosesc niște certificate guest pe baza cărora se generează certificatul proxy care va fi utilizat; În cazul acestui tip de utilizator, autentificarea se va face la nivelul aplicației prin username și parola, evident acel utilizator va trebui validat în prealabil înainte de a putea folosi resursele Grid; Categoriile de utilizatori care fac parte din aceasta situație: studenți sau diferite alte categorii de utilizatori care utilizează aplicația în procesul de învățare;

MANAGEMENTUL DATELOR

Sistemele distribuite implică o gestiune diferită a datelor. Datele sunt replicate pe mai multe noduri, crescând astfel disponibilitatea datelor comparativ cu situația în care fiecare dată are o singură replică.

Sistemul de management al datelor existent în gLite constă dintr-un set de servicii și unelte. Setul de servicii

care oferă acces la fișiere și la spațiul de stocare este reprezentat de către Storage Element. În esență, Storage Element (SE) este serviciul Grid responsabil cu salvarea și regăsirea fișierelor din diferite locații de stocare. În gLite fișierele sunt identificate de către utilizator pe baza unor nume logice de fișiere (LFN). Aceste nume logice sunt gestionate de către File Catalog Interface. Numele logic al unui fișier este o mapare ierarhică a unei structuri de directoare.

Interfața pentru managementul datelor presupune încărcarea de noi date pe diferite SE-uri folosind diferite tehnici de interacțiune. Fiecare resursă nouă va fi introdusă în sistem împreună cu un set de metainformații care vor fi folosite de componenta de regăsire a datelor. De exemplu în cazul imaginilor satelitare metainformații sunt coordonatele geografice, data achiziției imaginii, satelitul care a preluat acea imagine, etc. Componenta de căutare a imaginilor va folosi o interacțiune directă de selecție a unei zone geografice de pe o hartă. În funcție de zona selectată sistemul va returna doar imaginile din acea zonă, utilizând metainformațiile atașate imaginilor.

MANAGEMENTUL FLUXURILOR DE PROCESE

Sistemele de management al fluxurilor de procese existente în platformele Grid (gLite, Globus) sunt în general limitate în descrierea unor procese complexe. Această problemă apare în principal din folosirea DAG-urilor (grafurilor directe aciclice) pentru definirea fluxurilor de procese și care nu suportă includerea de cicluri, sau instrucțiuni condiționale. Un alt dezavantaj provine din dependența dintre sistemul de management și infrastructura Grid pe care se execută.

Astfel în cazul descrierii unui flux de procese pentru o anumită infrastructura Grid, portarea pentru execuția pe o altă infrastructura Grid implică redescriserea și reimplementarea fluxului de procese. Interacțiunea dintre acestea este foarte greu de realizat sau chiar imposibil în anumite cazuri.

În gProces fluxurile de procese se descriu sub forma de grafuri. Nodurile grafurilor pot să conțină operatori, subgrafuri, servicii sau resurse. În figura 1 este

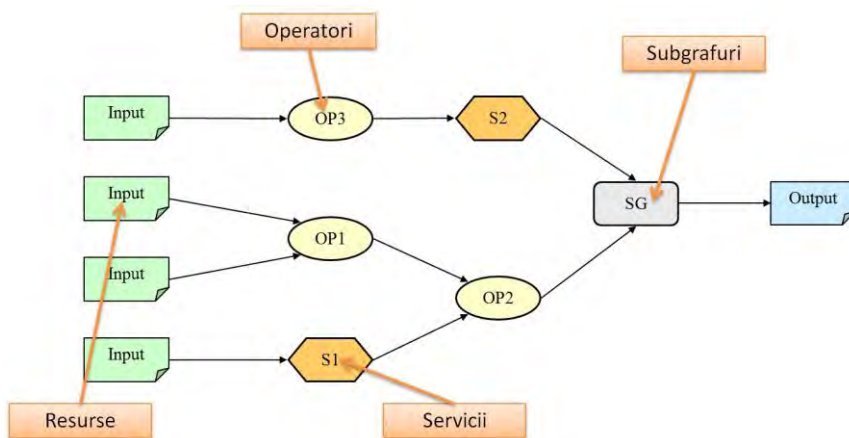


Figura 1: Exemplu de flux de procese

reprezentarea unui graf generic care utilizează diferitele tipuri de noduri definite în gProcess. Legăturile dintre noduri definesc atât fluxul datelor între diferitele noduri din graf cât și constrângerile.

Un operator poate fi privit ca și o cutie neagră cu porturi de intrare/ieșire, diferite constrângeri și proprietăți. Porturile de intrare/ieșire definesc fluxul datelor. Utilizatorul specifică fluxul de date prin conectarea porturilor de intrare și ieșire între diferiți operatori care nu trebuie neapărat să fie legați direct cu legături de control al fluxului. Informații adiționale pot fi definite cu ajutorul constrângerilor și proprietăților.

Intrările unui operator pot fi noduri de tip resurse, în acest caz datele de intrare reprezintă date de intrare primare, sau alte noduri de tip operator, în acest caz datele de intrare reprezintă datele de ieșire de la acele noduri de tip operatori.

Pentru încapsularea unui proces complex definit ca un flux de procese se folosește noțiunea de subgraf. Un flux de procese poate fi definit generic ca și subgraf care implementează o anumită funcționalitate. La definirea unui flux de procese complex anumite noduri din graf pot fi specificate să folosească un anumit subgraf, care la momentul execuției se expandează.

Fluxurile de procese care pot fi utilizate în cadrul platformei gProcess sunt de două tipuri, fluxuri abstracte și fluxuri concrete sau instanțiate. Prima categorie de fluxuri, cele abstracte, reprezintă o descriere conceptuală a unui algoritm sau al unui proces complex. Descrierea conceptuală folosește doar operatori și legături dintre aceștia. Totodată datele de intrare, resursele, sunt specificate generic, fără specificarea exactă a fișierelor de intrare. Aceste tipuri de fluxuri de procese reprezintă de fapt secvențe de procese definite generic care prin instanțiere se rafinează.

În cazul fluxurilor instanțiate de procese, nodurile sunt instanțiate la date, resurse, procese concrete. Aceste tipuri de grafuri vor putea fi ulterior executate.

ACCESUL LA SISTEMUL DE INFORMATII

În cazul utilizatorilor experimentați și al celor care execută fluxuri de procese este important accesul la diferitele informații despre diferitele resurse existente în rețeaua Grid utilizată. Din cauza faptului că resursele Grid, în special cele de calcul, sunt heterogene există posibilitatea ca nu toate nodurile de tip CE să suporte o anume prelucrare. Din acest motiv în teste se preferă execuția pe noduri pe care s-a testat deja și se cunosc performanțele sistemului în condițiile date.

Informațiile de bază care trebuie oferite utilizatorului sunt următoarele:

- Lista de WMS-uri - Fiecare VO este deservit de câteva WMS-uri, iar utilizatorul poate folosi doar aceste WMS-uri pentru rularea joburilor pe Grid. În mod curent această adresă este specificată automat de către sistem, dar există situații în care se dorește utilizarea unui anume WMS.
- Lista de CE-uri - Informațiile despre aceste noduri furnizate utilizatorului conțin date precum

numărul total de procesoare, numărul de procesoare disponibile, numărul total de joburi, numărul de joburi care sunt în execuție, numărul de joburi care așteaptă spre a fi executate, și numele nodului CE. Acest ultim parametru se utilizează în momentul în care se specifică cerințele legate de nodul CE care să deservească execuția.

- Lista de SE-uri - Deși utilizatorul lucrează cu nume logice de fișiere, deci nu ar trebui să aibă informații despre locul de stocare al fișierelor, în momentul stocării datelor inițial el trebuie să specifice un nod SE pe care îl folosește. Deasemenea este importantă componenta de management a datelor.

PLANIFICAREA EXECUȚIEI FLUXULUI DE PROCESE

Componenta de planificare al fluxurilor de procese și componenta de execuție a proceselor sunt unele dintre cele mai importante componente ale sistemului, influențând performanța (reprezentată ca și timp de execuție) a sistemului. Planificatorul, după o anumită strategie de planificare, planifică procesele care vor fi preluate și gestionate de către executor. Executorul, pentru fiecare proces, va executa o succesiune de evenimente.

Un aspect important al oricărui sistem de management și execuție a fluxurilor de procese pe Grid este acela de planificare. Având în vedere caracterul heterogen și dinamic al unei rețele Grid este importantă strategia de planificare aleasă.

Scalabilitatea, autonomia, calitatea și performanța unui sistem sunt influențate de arhitectura sistemului de planificare. Există trei categorii de arhitecturi pentru planificarea fluxurilor de procese, centralizată, ierarhică și descentralizată. Fiecare dintre aceste arhitecturi are particularitățile și avantajele ei, precum și o serie de dezavantaje. În cazul unei arhitecturi centralizate există un singur punct de decizie pentru toate procesele din cadrul unui flux de procese. Informațiile despre întregul flux de procese sunt disponibile de către acesta. Un avantaj al acestei arhitecturi vine din faptul că toate informațiile se găsesc într-un singur punct, astfel deciziile se pot lua având în vedere toate informațiile. Dezavantajul provine din faptul că acest sistem nu este scalabil cu numărul de procese și cu numărul de resurse Grid.

În cazul celorlalte două strategii, ierarhice și descentralizate, planificarea se poate realiza de mai multe planificatoare. Fiecare planificator se ocupă de un subflux de procese. Astfel poate conduce la planificare suboptimală deoarece la nivelul fiecărui planificator sunt disponibile doar o parte din informațiile complete, însă acest sistem este scalabil.

MONITORIZAREA EXECUȚIEI

Monitorizarea execuției unui flux de procese are două componente. O componentă este legată de utilizator și de posibilitatea de a-i oferi acestuia în fiecare moment o imagine de ansamblu a execuției fluxului de procese, utilizatorul putând fi astfel informat de starea sistemului. Cea de-a doua componentă este legată de generarea de

The screenshot shows a web-based monitoring interface. At the top, there are four search filters: 'Search process by name', 'Search process by words in description', 'Search process by' (with a dropdown menu showing 'status'), and 'Search process by date' (with 'From' and 'To' date pickers set to 27/03/2010). Below the filters, there is a checkbox 'Show all processes for victorbacu' which is checked. Underneath, there are radio buttons for 'Filters': 'Show only active processes', 'Show only completed processes', 'Show only cancelled processes', and 'No filters' (which is selected). The main part of the interface is a table with the following columns: 'Process name', 'Description', 'Status', 'In this status for', and 'Cancel'. The table contains six rows of data, all with 'Status' set to 'DONE'. Below this table, there is a section titled 'Current process status information' which contains another table with columns: 'Time info', 'Node name', 'Start server time', 'End server time', 'Status', and 'Options'. This table contains seven rows of data, all with 'Status' set to 'DONE'.

Figura 2: Exemplu pentru componenta de monitorizare

statistici și date relevante execuției care mai apoi vor putea fi folosite în procesul de optimizare a execuției. Din punctul de vedere al unui utilizator sistemul furnizează informații cum ar fi: timpul de start al execuției, timpul final (daca este disponibil), starea fiecărui proces, eventuale mesaje de eroare în cazul în care execuția fluxului de procese a eșuat. Aceste informații pot fi suficiente pentru a oferi o perspectivă de ansamblu a execuției fluxului de procese. Informații mai specifice, cum ar fi memoria utilizată pentru execuția unui proces, încărcarea sistemului, etc. sunt informații adiționale care pot fi deasemenea vizualizate. Vizualizarea acestor informații se poate realiza în mai multe moduri listă (figura 2) sau graf.

SUPORT PENTRU MECANISMUL DE PREVENIRE ȘI TRATARE A ERORILOR

În general eșecul execuției unui flux de procese pe Grid intervine din diferite cauze. În general cauzele sunt legate de infrastructurile Grid, heterogenitatea sistemului, resurse supraîncărcate (în general la nivelul WMS-ului), servicii sau componente inaccesibile, erori în gestiunea memoriei, și probleme care apar din cauza conexiunilor între nodurile Grid. Sistemele de management și execuție a fluxurilor de procese pe Grid trebuie să fie capabile de identificarea și rezolvarea acestor probleme, și să asigure un grad cât mai mare de succes chiar și în cazul apariției unor erori.

În general erorile apar la doua niveluri, la nivel de proces și la nivel de flux de procese. Tehnicile de eliminare a erorilor provenite la nivel de proces încearcă să reducă sau să mascheze efectul unor erori de execuție a unora dintre procesele componente ale unui flux complex de procese. La nivel de flux de procese se încearcă diferite tehnici de modificare a structurii fluxului de procese astfel încât execuția să se termine cu succes și în cazul apariției unor erori.

CONCLUZII

Un mediu de dezvoltare a aplicațiilor pe Grid presupune tehnici de interacțiune diferite în funcție de utilizatori

astfel încât să transparentizeze complexitatea unui sistem Grid.

REFERINȚE

1. gLite, <http://glite.web.cern.ch/glite/>
2. K. Harrison, W. T. L. P. Lavrijsen, C. E. Tull, P. Mato, A. Soroko, C. L. Tan, N. Brook, and R. W. L. Jones. GANGA: a user-Grid interface for Atlas and LHCb. In *Proceedings of Computing in High Energy and Nuclear Physics*, La Jolla, 2003.
3. D. Turi, P. Missier, C. Goble, D. D. Roure, and T. Oinn. Taverna Workflows: Syntax and Semantics. In *e-Science and Grid Computing*, IEEE International Conference on, pages 441-448, 2007.
4. D. Gorgan, V. Bacu, D. Rodila, and D. Petcu F. Pop. Experiments on ESIP - Environment oriented Satellite Data Processing Platform. SEE-GRID-SCI USER FORUM, 2009. Bogazici University, Istanbul, Turkey, December 9-10, ISBN: 978-975-403-510-0, pp. 157-166.
5. D. Gorgan, V. Bacu, T. Stefanut, and D. Rodila. Grid based Satellite Image Processing Platform for Earth Observation Applications Development. IDAACS'2009 - IEEE Fifth International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. 21-23 September 2009, Rende, Cosenza, Italy, 2009, pp. 247-252, ISBN: 978-1-4244-4901-9.
6. D. Gorgan, D. Mihon, V. Bacu, R. Meszaros, G. Gelybo, and T. Stefanut. Practical Considerations on the GreenView Application Development and Execution over SEE-GRID. SEE-GRID-SCI USER FORUM, 2009. Bogazici University, Istanbul, Turkey, December 9-10, ISBN: 978-975-403-510-0, pp. 167-175.
7. Kacsuk P., Sipos G., Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal, *Journal of Grid Computing*, ISSN 1570-7873, pp. 221-238