

## Current Developments in Grid Computing

Asist. Alexandru Dan CĂPRIȚĂ  
Catedra de Contabilitate și Informatică Economică,  
Facultatea de Științe Economice, Galați

*This paper presents the recent developments in the emerging grid computing paradigm. A background to grid technologies that try to provide dependable, consistent, pervasive and inexpensive on demand access to computer resources is introduced. Also, research issues and challenges of grids systems are presented. Finally, distributed systems structures for online transactions based on economic models are proposed.*

**Keywords:** distributed systems, grid technologies, economy grid.

### Introducere

Termenul de „grid” s-a conturat la mijlocul anilor 1990 pentru a se referi la o infrastructură computațională distribuită capabilă să pună la dispoziție resursele în funcție de necesitățile specifice fiecărui client, așa cum poate fi utilizată rețeaua (eng. grid) de electricitate. Tehnologia grid poate spori în mare măsură productivitatea și eficiența organizațiilor (virtuale) care trebuie să facă față provocărilor, prin optimizarea proceselor și resurselor, partajarea acestora în rețea și colaborare. Soluțiile grid vin în întâmpinarea organizațiilor ce doresc dezvoltarea unei arhitecturi orientate pe servicii care să facilitează managementul și integrarea proceselor afacerii. Pe piață există un număr mare de ofertanți

de astfel de soluții, dintre aceștia putând fi amintiți: Oracle, IBM, DataSynapse, GreenTeaSoft, Sun, Hewlett-Packard, United Devices, etc.

O definiție pentru grid ar putea fi cea dată de Wolfgang Gentzsch în „The Grid, The Time Machine of the 21st Century”: **grid-ul** „reprezintă o infrastructură software/hardware/instrumente care furnizează acces nelimitat, dependent, consistent și omniprezent la resurse de tip computațional și date”. De fapt, grid-ul poate fi văzut ca cea mai recentă și completă evoluție a procesării distribuite, în tabelul de mai jos fiind prezentată o comparație ([3],[5]) cu alte tehnologii binecunoscute.

**Tabelul 1.** Comparație alte tehnologiei Grid cu alte tehnologii distribuite

Tehnologie	Similarități	Diferențe
<b>Web</b>	Un grid ascunde complexitatea sa, interconectează resurse diferite și oferă o percepție unificată asupra mediului;	Un grid permite mașinilor să lucreze împreună și să colaboreze într-o formă care presupune mai mult decât o simplă comunicarea;
<b>Peer-to-peer</b>	Ambele pot partaja fișiere și pot comunica direct, uneori prin intermediul unui broker central;	Grid-urile pot utiliza relații „mai mulți la mulți” pentru a partaja o mare varietate de resurse, nu numai fișiere;
<b>Cluster-e</b>	Cluster-ele și gridurile grupează resursele pentru a rezolva o problemă și pot avea un management unitar al instrumentelor pentru toate componentele;	Un cluster este alcătuit din calculatoare de același tip aflate într-o locație centrală și necesitând o imagine unică a sistemului (SSI). Spre deosebire de cluster unde accentul cade pe performanța procesării paralele, într-un grid acesta revine partajării resurselor. SSI nu mai este o cerință, calculatoarele din grid fiind eterogene și distribuite geografic;
<b>Virtualizare</b>	Grid-ul permite componentelor să fie percepute ca un fond comun de resurse. Activitatea poate fi desfășurată într-un mod asemănător pe oricare dintre ele;	Spre deosebire de virtualizare care se referă la un singur sistem, un grid permite virtualizarea resurselor eterogene de pretutindeni, chiar globale, pentru a forma un fond comun de servicii IT;

Grid-urile de pe piață sunt, de fapt, combinații ale **tipurilor principale de grid**, și anume:

- *grid-uri computaționale* (eng. computational grids) - orientate pe furnizarea accesului la un fond comun de putere de calcul. Grid-urile computaționale exploatează capacitatea de procesare agregată a calculatoarelor eterogene, incluzând în general cluster-e de calculatoare dedicate, supercalculatoare;

- *grid-uri de date* (eng. data grids) - se concentrează pe management, distribuție și replicarea eficientă a unor mari cantități de date, asigurând accesul la date pentru organizațiile din grid. Utilizatorii nu sunt interesați de locația în care sunt stocate datele, ci au nevoie de disponibilitatea datelor pentru a-și putea îndeplini sarcinile de lucru;

- *grid-uri colaborative* (eng. scavenging grids) - se folosesc în sistemele cu un număr mare de desktop-uri, acestea fiind baleiate (eng. scavenging) pentru a determina ciclul CPU neutilizați precum și alte resurse. Proprietarii resurselor au controlul atunci când resursele lor sunt disponibile pentru a participa la grid (e.g. Condor, Entropia).

Ținând cont de aplicațiile ce vor rula în grid, trebuie ales tipul potrivit de pentru organizație.

Cu toate că există numeroase modele de servicii grid, sunt implementate cu succes în cadrul sistemelor grid **arhitecturi de sistem** de tip *protocoale stratificate* (e.g. Globus) și de tip *mașină virtuală* (e.g. Legion). În cadrul arhitecturii de tip protocoale stratificate, structurile sistemului sunt succesiv construite pe funcționalitatea nivelurilor inferioare. Stratificarea sistemului facilitează independența resurselor și aplicațiilor din grid, componentele eterogene putând interacționa utilizând interfețele definite de protocol. În mod similar, arhitecturile de tip mașină virtuală abstractizează eterogenitatea resurselor din grid. Totuși, abstractizarea arhitecturii mașinii virtuale este realizată la un nivel superior, componentele și resursele grid-ului fiind integrate într-o singură entitate de calcul ce poate fi adresată global. Prin comparație, sistemele stratificate abstractizează domeniul de

aplicație și resursele ce stau la baza grid-ului pentru a furniza o interfață comună independentă de diferitele componente hardware și software conținute în grid.

În general, arhitecturile grid de tip protocoale stratificate sau mașină virtuală suportă aceleași funcționalități de bază, ca alocarea resurselor, managementul proceselor, servicii de informare, securitate, comunicare, managementul datelor, replicare, toleranța la defecte, precum și parametri legați de calitatea serviciilor. Aceste mecanisme nu sunt specifice sistemelor grid, ci ele au fost adaptate pornind de la sistemele paralele și distribuite astfel încât să răspundă cerințelor sistemelor grid.

Există și încercări de standardizare, proiectul celor de la Global Grid Forum urmărind să organizeze protocoalele sub Open Grid Services Architecture (OGSA), arhitectură de sistem grid bazată pe conceptele și tehnologiile serviciilor Web.

### Probleme și provocări ale grid-urilor

Procesarea grid are drept scop să cupleze resursele distribuite geografic și să ofere servicii într-o manieră transparentă, puternică și ieftină indiferent de locația fizică a resurselor sau a punctelor de acces. Pentru a realiza aceasta, trebuie rezolvate provocările în ceea ce privește dezvoltarea de instrumente și middleware-uri capabile să integreze cu ușurință dispozitive de calcul, cluster-e, depozite de date și rețele ([3]). Deși standarde ca OGSA încercă să definească un număr de servicii principale ale grid-ului, există încă numeroase probleme care trebuie soluționate înainte ca aplicațiile grid să devină la fel de pătrunzătoare ca aplicațiile Web și Internetul. Aceste provocări includ probleme legate de ingineria software, planificarea și echilibrarea încărcării, procesarea autonomă, replicarea, etc. În continuare se vor prezenta doar două dintre aceste probleme și posibilele modalități de rezolvare a acestora.

### Planificarea și echilibrarea încărcării

La fel ca și în cazul altor sisteme distribuite, alocarea cererilor, planificarea lucrărilor și

echilibrarea încărcării reprezintă o problemă importantă care are un efect direct asupra ieșirilor sistemului și timpului total de execuție al lucrărilor. Sistemele bazate pe planificare, în general, analizează sau prezic starea unui sistem pentru a determina cel mai bun nod căruia să-i aloce lucrarea. În prezent, sistemele grid utilizează modele predictive care alocă lucrările sistemelor neutilizate pe baza unei aproximări a stării sistemului. Aceste modele sunt mai eficiente deoarece nu trebuie determinată starea globală a sistemului.

Există un trend de a utiliza în sistemele grid a modelelor economice sau de piață. Aceste modele partiționează resursele grid-ului într-o economie computațională (eng. computational economy) în care resursele sunt schimbate ca și cum ar fi monedă fizică. Utilizatorii pot specifica bugetul și termenul limită pentru lucrări, permițând astfel planificatorului să aloce resursele pentru a îndeplini cerințele legate de timp și costuri (e.g. Nimrod-G, Grid Application Development Software, Condor-G).

Alte modele alocă resursele pe baza performanței precise, alocând lucrările mașinilor în funcție de cât de repede se așteaptă să se termine o lucrare pe un anumit sistem (e.g. AppLeS, Matchmaker). Din păcate, aceste modele bazate pe performanță sunt adesea lacome, neluând în considerare alte lucrări nou intrate în sistem, ceea ce poate conduce la performanțe slabe.

Multe sisteme bazate pe planificarea lucrărilor încearcă să reechilibreze resursele sistemului în mod dinamic și să migreze lucrările de la sistemele ocupate la cele neutilizate.

### **Probleme legate de ingineria software**

Dezvoltarea și analiza aplicațiilor grid ridică, ca și alte sisteme distribuite, numeroase probleme. Programatorii sunt expuși în mod direct complexității sistemelor grid și sunt confrunțați cu sisteme care necesită coordonarea simultană a mai multor componente independente ale sistemului. Până când bibliotecii la nivel de aplicație nu vor fi larg răspândite, dezvoltarea aplicațiilor grid va fi îngreunată. În prezent, sunt dezvoltate câteva interfețe de programare a aplicațiilor care să simplifice

accesul la resursele grid-ului. Aceste sisteme integrează diferite servicii componente ale grid-ului într-o singură interfață coerentă (e.g. MPICH-G2, GraDS, Cactus).

În sistemele distribuite, ca grid-urile computaționale sau de date, aplicațiile pot utiliza mii de sisteme eterogene, făcând operațiile de monitorizare și depanare foarte complicate. Aceasta se datorează unor probleme majore asociate cu sistemele paralele și distribuite:

- dimensiunea mare a sistemului și eterogenitatea sa;
- numărul mare de programe care se execută în paralel;
- latență crescută a rețelelor de mare întindere;
- dificultatea obținerii unei stări globale a grid-ului;
- imposibilitatea recreării exacte a secvenței de execuție datorită variațiilor de încărcare a sistemului.

Evaluarea performanțelor este o altă problemă de importanță critică în dezvoltarea sistemelor distribuite. Performanța sistemului este o măsură a eficienței cu care diferite componente ale grid-ului își completează sarcinile în diferite condiții. În prezent, este aproape imposibil să se creeze teste controlabile și cu rezultate ce pot fi reproduse, datorită condițiilor în schimbare ca latența rețelei, blocajele, încărcarea sistemului și deferitele defecte ale sistemului. De aceea, rezultatele sistemului grid pot furniza doar o aproximare brută a eficienței unui algoritm particular. Există câteva medii de simulare (e.g. GridSim și Bricks) care utilizează simularea cu evenimente discrete pentru a modela interacțiunea dintre componentele grid-ului și pentru a înregistra performanțele ([4]). Aceste sisteme furnizează un mediu controlabil și reproductibil pentru a genera rezultatele testelor.

### **Grid-uri bazate pe modele economice**

Managementul și planificarea resurselor într-un grid reprezintă o inițiativă complexă deoarece resursele sunt distribuite, eterogene ca natură, deținute de diverși indivizi sau organizații, cu propriile politici, au diferite modele de acces și cost, precum și variații în ceea

ce privește încărcarea și disponibilitatea resurselor. Aceasta conduce la un număr de probleme legate de autonomie, eterogenitatea interacțiunilor, extensibilitatea politicilor, alocarea resurselor, broker-ul de resurse, controlul online, scalabilitate și probleme economice.

Un număr de sisteme grid (ca Globus și Legion) rezolvă mare parte dintre aceste probleme cu excepția economiei computaționale. Buyya argumentează că economia computațională este necesară pentru a crea grid-uri scalabile reale, deoarece aceasta furnizează un mecanism pentru a regla cererea și oferta de resurse din grid. Astfel, proprietarii de resurse sunt cointeresați să facă parte din grid iar consumatorii de resurse sunt interesați să

optimizeze utilizarea resurselor și să realizeze un echilibru între timp și costurile de acces. În lucrarea [2], se propune un cadru de lucru al economiei computaționale construit deasupra middleware-ului sistemelor grid existente care oferă o infrastructură pentru managementul și schimbul resurselor în mediile grid.

Un grid orientat pe modele economice include o gamă întreagă de tehnologii software, de la sisteme de operare locale la broker-ii de resurse globale și aplicații special proiectate pentru a exploata capacitățile grid-ului. Interacțiunea dintre aceste componente trebuie să fie sigură și să se adapteze la resursele a căror stare se schimbă.

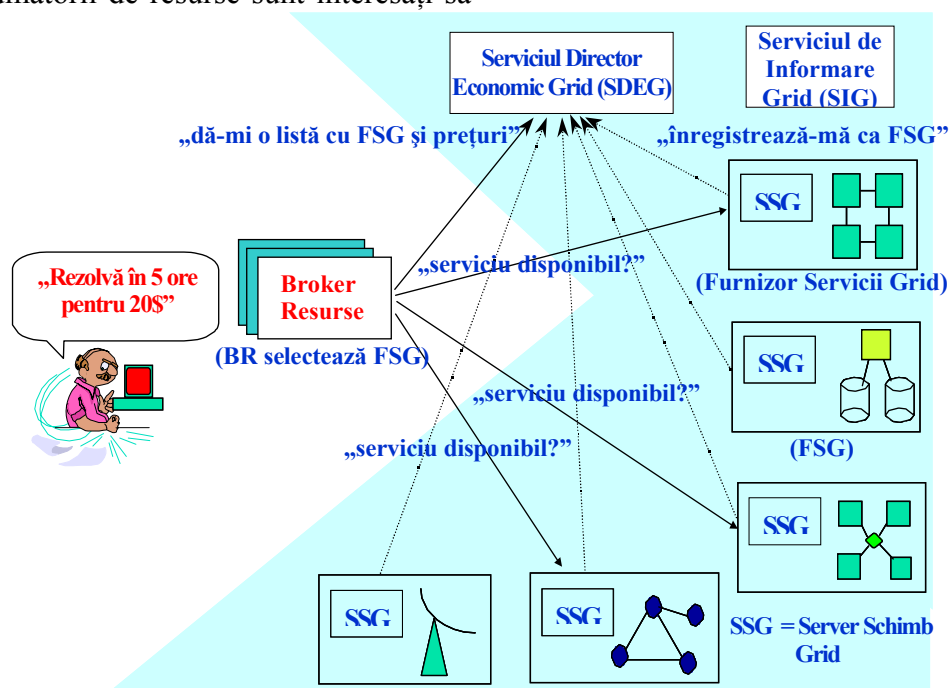


Fig.1. Interacțiunile dintr-un grid bazat pe modele economice

După cum reiese și din figura 1, utilizatorii grid-ului interacționează cu *broker-ul de resurse grid* (eng. Grid Resource Broker) care ascunde complexitatea managementului și planificării resurselor. Broker-ul descoperă resursele utilizând *serviciul de informare grid* (eng. Grid Information Service - GIS), negociază cu resursele grid sau cu agenții lor pentru costurile serviciilor, efectuează selecția resurselor, mapează și planifică task-urile cu resursele, împarte în faze aplicația și datele pentru procesarea lor distribuită iar în final, adună rezultatele și le transmite utilizato-

rilor. De asemenea, GRB este responsabil cu monitorizarea progresului execuției aplicației precum și cu managementul și adaptarea la schimbările din grid (e.g. defecte ale resurselor).

### Concluzii

Impulsionată atât de dezvoltarea rețelelor de mare viteză cât și de creșterea puterii de calcul a microprocesoarelor, procesarea grid are un impact remarcabil nu numai la nivel academic, ci din ce în ce mai mult și la nivelul întreprinderilor din toate domeniile de activi-

tate. Cu tot avansul tehnologic însă, mai puțin de 5% din puterea de procesare a serverelor Windows și calculatoarelor desktop, respectiv 15-20% pentru server-ele UNIX [1] este utilizată. Întreprinderile realizează că procesarea grid oferă rate ale profitului extraordinare, cu o perioadă de înapoiere mică și cu un cost total scăzut față de alte soluții tehnologice.

În cazul structurii de sistem distribuit cu aplicații în tranzații on-line, ideea este de a identifica o bază de modele economice (al căror succes a fost probat în practică, reprezentând o extindere naturală a sistemelor care implică scopuri, resurse și acțiuni, cu mecanisme de piață) care să sprijine managementul și schimbul resurselor într-un grid destinat tranzacțiilor on-line. Trebuie realizată o arhitectură scalabilă bazată pe serviciile oferite de sistemele grid existente (dezvoltate folosind unelte ca Globus Toolkit) care să furnizeze mecanisme pentru rezolvarea cerințelor utilizatorilor. Când răspunsul este generat este la fel de important ca și ce răspuns este produs, sistemele de timp real interacționând cu lumea reală într-un mod care implică timpul. Buyya propune diferite modele economice, variind de la cele ale pieței de materii prime până la cele bazate pe licitații, pentru a putea astfel stabili strategia de prețuri. Mode-

larea sistemului va utiliza rețele Petri, acestea permițând atât o analiză calitativă (cu scopul de a demonstra corectitudinea modelului conceptual aferent sistemului modelat) cât și o analiză cantitativă (permite evaluarea eficienței sistemului modelat).

### Bibliografie

- [1] A. Abbas - *Grid Computing: A Practical Guide to Technology and Applications*, Editura Charles River Media, 2005, ISBN 1-58450-276-2
- [2] R. Buyya, D. Abramson, J. Giddy - *A case for economy grid architecture for service oriented grid computing*, IEEE, Parallel and Distributed Processing Symposium., Proceedings 15th International, pg. 776 - 790
- [3] W. Jia, W. Zhou - *Distributed Network Systems. From Concepts to Implementations*, Editura Springer, 2005, ISBN 0-387-23839-5, pg. 435-470
- [4] <http://www.buyya.com/ecogrid/>, 11.2005
- [5] [http://www-1.ibm.com/grid/about\\_grid/ibm\\_grid/get\\_star\\_ted.html](http://www-1.ibm.com/grid/about_grid/ibm_grid/get_star_ted.html)  
<http://www-128.ibm.com/developerworks/grid/library/overview/>, 03.2006
- [6] [http://en.wikipedia.org/wiki/Grid\\_computing](http://en.wikipedia.org/wiki/Grid_computing), 02.2006