

UNIVERSITATEA BABEȘ-BOLYAI, CLUJ-NAPOCA  
FACULTATEA DE ȘTIINȚE ECONOMICE ȘI GESTIUNEA AFACERILOR

# PROTOTIP PENTRU UN SISTEM COLABORATIV ÎN AFACERI

- REZUMATUL TEZEI DE DOCTORAT -

MIRCEA MOCA

ÎNDRUMĂTOR: PROF. DR. ȘTEFAN IOAN NIȚCHI

CLUJ-NAPOCA 2010

# Cuprins

<b>Introducere</b>	<b>1</b>
<b>1 Modele Decizionale în Agregarea Resurselor</b>	<b>5</b>
1.1 Introducere . . . . .	5
1.2 Structura Sistemului și Parametrii Urmăriți . . . . .	6
1.3 Evaluarea Mecanismului de Descoperire a Resurselor . . . . .	7
1.4 Experimente și Rezultate . . . . .	8
1.5 Concluzii . . . . .	11
<b>2 Prototip Pentru MapReduce în Sisteme Desktop Grid</b>	<b>13</b>
2.1 Introducere . . . . .	13
2.2 Evaluarea Performanțelor Prototipului . . . . .	14
2.2.1 Comunicații Colective . . . . .	14
2.2.2 Evaluarea Implementării MapReduce . . . . .	14
2.2.3 Specific Desktop Grid . . . . .	15
2.3 Concluzii . . . . .	17
<b>Concluzii</b>	<b>18</b>
<b>Rezultate</b>	<b>20</b>
<b>Bibliografie</b>	<b>24</b>

# Introducere

Termenul **colaborare** provine din Latină (*collaboratus*), însemnând<sup>1</sup> a *munci împreună*. Pornind de la tipurile de entități care participă la proces, identificăm mai multe tipuri de colaborare. Astfel, avem colaborare de tip om-om, om-calculator și calculator-calculator (complet automatizată). Tanenbaum [42] definește sistemele distribuite ca o colecție de calculatoare independente care oferă utilizatorilor săi imaginea unui **sistem unic, coerent**. Pentru a obține coerența și unitatea sistemului, componentele autonome trebuie într-un fel sau altul să **colaboreze** [42]. Modul în care este realizat designul dimensiunii colaborative a sistemului distribuit depinde de arhitectură acestuia. Atât în acest rezumat cât și în teza prezentată, vom avea de a face doar cu colaborativitate de tip calculator-calculator, obiectivele noastre rămânând în zona sistemelor distribuite.

Originile Grid computing se află aproximativ la trei decenii în urmă, în anii '80. Motivația care a stat la acea vreme la baza construirii sistemelor care agregă putere de calcul, au fost aplicațiile științifice cu un necesar de putere de calcul din ce în ce mai mare. La acea vreme, puterea de calcul era deja furnizată de centrele de calcul prin intermediul supercalculatoarelor, caracterizate prin hardware omogen dar scump. Dificultățile de acces, costurile de achiziție și mentenanță a supercalculatoarelor au oferit suficientă motivație unor cercetători de a căuta o alternativă la acestea. Scopul era găsirea unei soluții prin care o mare cantitate de resurse computaționale să fie agregate mai ușor, la un preț mai accesibil. Ulterior, Foster și Kesselman [31] introduc în 1998 conceptul *Grid computing* sau pur și simplu *Gridul*, care presupunea interconectarea calculatoarelor obișnuite la scară globală într-un singur sistem pentru a oferi putere de calcul agregată. Autorii definesc clar pentru prima dată acest concept și oferă o listă de cerințe funcționale și arhitecturale ale unui sistem de tip Grid adevărat. Așadar, un Grid computațional este o infrastructură hardware și software care oferă consumatorilor un acces caracterizat de: dependabilitate, consistență, universalitate și accesibilitate la resurse computaționale [31]. Într-un articol ulterior [19], Foster et al. revin asupra definiției gridului pentru a trata laturile sociale și politice ale acestuia. Câțiva ani mai târziu, Foster publica o listă [17] cu trăsăturile esențiale ale gridului și subliniază faptul că gridul trebuie evaluat după aplicațiile suportate și **valoarea de business** pe care o oferă, nu după arhitectura sa.

Alvaro et al. definesc Organizațiile Virtuale (VO) în contextul sistemelor colaborative ca un set de utilizatori și organizații reale care oferă resurse precum putere de procesor, spațiu de stocare sau servicii online pentru utilizatorii săi pentru a realiza un **scop comun**. Un asemenea scop poate fi reprezentat de aplicații business inter-organizaționale precum lanțuri de aprovizionare bazate pe Grid [4].

Oamenii de știință au fost primii care au înțeles avantajele gridului și l-au folosit la maxim pentru a-și realiza experimentele în domenii ca fizica sau ingineria. Inițial, gridurile au fost construite la o scară redusă, agregând resurse la nivel organizațional, precum universități. Astfel, gridurile existau ca sisteme izolate. De la acea vreme, au fost depuse eforturi (financiare și de cercetare) semnificative pentru a integra diversele griduri existente într-un singur sistem global, *Gridul*.

---

<sup>1</sup>Conform definiției din Dicționarul Explicativ al Limbii Române.

Astăzi, la nivel european eforturile sunt concentrate pe interconectarea gridurilor naționale existente într-o infrastructură europeană. Acest efort este condus de organismul DEISA<sup>2</sup>, care dezvoltă o infrastructură pentru calcul distribuit la nivel de continent [24]. Mai mult, proiectul EGI-InSPIRE<sup>3</sup>, lansat la data de 1 Mai 2010, cofinanțat de către Comisia Europeană și cu o durată de patru ani, reprezintă un efort colaborativ care implică mai mult de 50 de institutii, în peste 40 de țări. Proiectul își propune realizarea unei infrastructuri funcționale de tip Grid la nivel european.

Enunțând țintele și scopul revistei FGCS<sup>4</sup>, Peter Sloot<sup>5</sup> subliniază faptul că Gridul este o structură computațională care se dezvoltă rapid și care permite componentelor din infrastructura tehnologiei informaționale, resurselor computaționale, baze de date, senzori și oameni să fie partajate într-un mod flexibil ca și **unelte colaborative**.

Stockinger<sup>6</sup> realizează o analiză [40] a tehnologiilor Grid, inventariind caracteristicile și țintele gridului în scopul de a direcționa cercetarea din domeniu spre conceptul de Grid. Deasemenea, Stockinger identifica noile oportunități ale gridului în zona de **business**. În acest context, autorul prezintă o listă cu neajunsurile pe care le au atât infrastructura software cât și cea hardware ale gridului. Astfel, infrastructura software nu este încă stabilă și nici pregătită pentru utilizarea în industrie, și, mecanisme de securitate, contabilizare și taxare mai trebuie dezvoltate pentru ca gridul să poată fi folosit cu adevărat în scop comercial [41].

Sistemele computaționale Desktop Grid au o abordare diferită în ceea ce privește infrastructura. Spre deosebire de gridurile tradiționale care se întind pe unul sau mai multe domenii administrative securizate, Desktop Gridurile integrează resurse de la calculatoarele personale conectate la Internet. Aceasta este o paradigmă diferită care permite construirea unui sistem la scară globală mult mai ușor, de exemplu proiectele BOINC<sup>7</sup>. Și totuși, calitatea resurselor agregate de aceste sisteme este mai slabă pentru că sunt de neîncredere, dinamice și eterogene. Așadar, Desktop Gridul are și el dezavantaje și conform cu Stockinger [40] acestea se adresează aplicațiilor unde datele sau rezultatele calculelor pot fi pierdute sau sunt ușor reproductibile.

Unele griduri computaționale au infrastructură de tip a peer-to-peer (P2P). Cu unele excepții, într-o arhitectură P2P, participanții au funcționalități relativ egale. Datorită schemelor permissive de comunicare în arhitecturile P2P, sistemele computaționale P2P sunt considerate ca unelte puternice pentru organizarea gridului și a calculului cooperativ [45]. Faptul că tehnologiile Grid și P2P tind spre aceeași destinație face ca eforturile de cercetare în cele două domenii să se combine natural [28]. În 2003, Foster și Iamnitchi [18] prezintă o viziune în care cele două domenii converg pentru a obține sinergia. În zilele noastre, eforturi semnificative sunt depuse în sensul convergenței celor două tehnologii sub îndrumarea ANL<sup>8</sup> și IBM<sup>9</sup>, care au propus un nou model de cercetare în griduri, numit OGSA<sup>10</sup> [20], în care conceptele Grid, P2P and Servicii Web au fost unificate pentru a partaja toate resursele ca servicii web. Un alt exemplu în care tehnologiile Grid și P2P sunt folosite împreună este modelul ierarhic prezentat în [48] pentru a construi

---

<sup>2</sup>Distributed European Infrastructure for Supercomputing Applications.

<sup>3</sup>European Grid Infrastructure; nr. contract: RI-261323.

<sup>4</sup>Future Generation Computer Systems, Revista Internațională pe Grid Computing și eScience.

<sup>5</sup>Peter Sloot este profesor în Știința Calculatoarelor la Universitatea din Amsterdam, editor șef la două reviste Elsevier Science: JoCS (The Journal of Computational Science) și FGCS (Future Generation Computing Systems).

<sup>6</sup>Heinz Stockinger este cercetător și specialist în sisteme de tip Grid la Institutul Elvețian de Bioinformatică. Acesta este doctor în Știința Calculatoarelor și Administrarea Afacerilor la Universitatea din Viena, Austria. El a lucrat mai mulți ani la CERN (The European Organization for Nuclear Research) în cadrul proiectelor europene DataGrid și EGEE (Enabling Grids for E-science), unde cercetarea sa a fost axată pe date în Grid și managementul resurselor.

<sup>7</sup>BOINC: Berkeley Open Infrastructure for Network Computing.

<sup>8</sup>Argonne National Laboratory, unul dintre cele mai mari și vechi laboratoare pentru știință, inginerie și cercetare la nivel național ale departamentului de Resurse și Energie S.U.A. [22].

<sup>9</sup>Compania IBM (International Business Machines).

<sup>10</sup>Open Grid Service Architecture.

un mediu pentru aplicații colaborative.

Aspectele menționate mai sus arată că cercetarea în sistemele computaționale Grid și P2P este deschisă și necesită soluții. Deasemenea, inovații sunt așteptate pe aceste direcții de cercetare pentru a realiza atributul de colaborativitate în aceste sisteme distribuite. În teza pe care în această lucrare o rezumăm contribuim la îmbunătățirea colaborativității în sistemele computaționale Grid și P2P folosind concepte și metode fundamentale din economie și business. Prin asemenea instrumente arătăm că putem obține o colaborare automatizată mai bună, mai robustă și de încredere în contextul sistemelor distribuite care se întind pe mai multe domenii administrative.

Acest rezumat prezintă în cele două capitole ale sale contribuții distincte pe tema sistemelor de calcul colaborativ distribuit. În capitolul 1 îmbunătățim procesul de agregare a resurselor pentru un sistem de calcul distribuit cu o arhitectură P2P folosind modele decizionale. Asemenea metode matematice joacă deja un rol important în domeniul științelor economice și ale businessului [29]. Astfel, propunem agregarea resurselor bazată pe un proces decizional sofisticat. Atributele utilizate de către modelele decizionale folosite caracterizează resursele partajate, precum: putere de procesor, viteza de transfer de date între noduri, capacitatea de transfer etc. Realizăm deasemenea o analiză comparativă a unor algoritmi din categoriile principale de modele decizionale prezentate de literatură. Un modul software cu asemenea capacități poate fi integrat în stratul middleware al unui grid pentru a îmbunătăți procesul de agregare a resurselor.

Business Intelligence reprezintă o colecție de soluții de tehnologii de: data warehousing, data mining, analytics<sup>11</sup>, raportare și vizualizare folosite pentru colectarea, integrarea, interpretarea și minarea datelor din organizații pentru a susține procesul decizional [9]. Cum marimea seturilor de date care sunt colectate de către aplicațiile de Business Intelligence este în continuă creștere [3], soluțiile tradiționale pentru procesarea datelor au devenit inaccesibile. În ultimii ani, sistemele care implementează paradigma MapReduce [10, 21] se constituie ca infrastructuri alternative avantajoase ca și cost pentru aplicațiile de Business Intelligence. Acestea oferă cantități importante de resurse agregate pentru algoritmi de minare a datelor, clasificare, analiză de text, filtrare colaborativă și altele.

MapReduce câștigă teren în permanență și din ce în ce mai mulți cercetători își manifestă interesul în dezvoltarea de sisteme care suportă aplicații de Business Intelligence. Spre exemplu, Hadoop MapReduce reprezintă un mediu software pentru scrierea de aplicații care procesează în paralel cantități mari de date, într-un timp scurt. Aceste sisteme rulează pe arhitecturi cluster [46]. X-RIME<sup>12</sup> reprezintă o bibliotecă pentru sisteme cloud pentru analiza la scară mare a rețelelor electronice de socializare, construite pe Hadoop [47]. Conform autorilor, X-RIME poate fi integrat în sisteme de Business Intelligence pentru a sprijini companiile să extragă informații prețioase la un preț convenabil din rețelele electronice de socializare.

O altă tehnologie pentru calcul distribuit este DryadLINQ [27, 49], bazată pe fluxuri de execuție care folosesc grafuri orientate aciclice. Aceasta obține însă în general performanțe mai slabe în comparație cu Hadoop [12, 13].

Investiția într-un cluster - sau *grid de companie* cum mai este numit de Stockinger, presupune noi achiziții de hardware obișnuit. Mai mult, un grid implică costuri semnificative cu energia electrică și personalul de întreținere a sistemului. În cazul companiilor în care un astfel de sistem are incertă amortizarea, investiția inițială împreună cu costurile ulterioare cu mentenanța sistemului ar fi nejustificate economic. Astfel, cel puțin temporar, luarea în calcul a resurselor oferite de un sistem Desktop Grid este o alternativă preferabilă.

În capitolul 2 introducem o infrastructură pentru aplicații de Business Intelligence. Așadar, prezentăm

<sup>11</sup> Aplicații informatice și statistice pentru rezolvarea problemelor din business și industrie [32]

<sup>12</sup> Acest proiect reprezintă efortul comun al Universității Beijing de Posta și Telecomunicații (BUPT) și al Laboratorului de cercetare IBM China, susținut de programul IBM Open Collaboration Research [25].

un prototip care implementează modelul de programare MapReduce destinat unui mediu Volunteer Computing, construit pe o infrastructură Desktop Grid, folosind BitDew<sup>13</sup> ca middleware. După cunoștințele noastre, în prezent nu mai există o abordare similară în literatură. Vom prezenta pe scurt aspectele cheie ale prototipului prezentat în teză, insistând pe testele realizate pentru a valida prototipul, arătând că acesta este rezistent la pene, gestionează replicarea datelor, efectuează concomitent fazele Map și Reduce, ascunde latența de comunicare și se bucură de un mecanism cu două niveluri de planificare a datelor.

---

<sup>13</sup>BitDew : proiect open source care are ca scop managementul de date în sisteme computaționale de tip Grid, Desktop Grid și Cloud, fondat de Gilles Fedak și Haiwu He, în cadrul INRIA Futurs [23, 15].

# Capitolul 1

## Modele Decizionale în Agregarea Resurselor

Cum sistemele orientate pe servicii se îndreaptă spre un mediu colaborativ descentralizat, agregarea resurselor devine o funcționalitate importantă, interesant de studiat. În acest capitol cercetăm eficacitatea agregării resurselor într-un sistem care are o arhitectură de tip peer-to-peer în care nodurile sunt autonome, fiecare putând oferi sau consuma servicii. În studiul realizat considerăm diverse valori pentru parametrii de interes: înzestrarea cu resurse a participanților la sistem, încărcarea sistemului cu cereri de servicii, capacitatea intrinsecă a sistemului de a descoperi resurse și gradul de satisfacție resimțit de utilizatori în urma aplicării mecanismului de agregare de resurse. Vom arăta că pentru cererile de cantități semnificative performanța sistemului nu scade pe termen lung, iar pentru cererile de cantități mici metoda de descoperire a resurselor combinată cu algoritmul de selecție a participanților la serviciu reușesc să obțină o performanță mai bună. Deasemenea, oferim rezultatele unei analize comparative a mai multor modele decizionale reprezentative pentru contextul nostru, scoțând în evidență performanțele sporite ale algoritmului sofisticat Promethee bazat pe comparații de valori ale atributelor pe perechi de alternative.

### 1.1 Introducere

Cum gridul tinde să împrumute arhitecturi distribuite de tip P2P [18], arhitecturile orientate pe servicii trebuie să se adapteze particularității aduse de tehnologiile P2P. Pentru a realiza un sistem P2P pervaziv, este necesar ca toate nodurile participante la sistem să poată juca ambele roluri: consumatori și prestatori de servicii. Un asemenea sistem ideal ar trebui să descopere și să agreghe resursele potrivite pentru a satisface cererea de resurse a unui utilizator.

În acest capitol cercetăm mai întâi<sup>1</sup> modul în care putem aplica modele decizionale pentru a agrega resurse în contextul prezentat. Prin agregarea resurselor înțelegem procesul de colectare de diverse cantități de resurse de același tip de la diverși participanți care partajează aceste resurse. Fiecare nod joacă ambele roluri, atât de consumator de servicii (resurse) cât și de ofertant și este echipat cu același model decizional) care stă la baza procesului de selecție a partenerilor de tranzacție) ca și ceilalți participanți.

În al doilea rând<sup>2</sup>, cercetăm câteva modele decizionale mai complexe. În [36] am folosit versiunile obiectivă și subiectivă ale modelului decizional Onicescu [26]. Acesta este un model neparametric, literatura [1, 16] recomandand alternative mai complexe. Așadar, pe lângă Onicescu, am considerat și Metoda Utilitatii Globale [26] din cadrul MAUT (Multi-Attribute Utility Theory) [11] și Promethee [6], acoperind întreg spectrul de tipuri de modele decizionale adecvate contextului nostru. Scopul nostru este

---

<sup>1</sup>Având în minte ideea îmbunătățirii procesului de agregare a resurselor într-un sistem P2P aplicând modele decizionale, prima noastră țintă a fost să studiem fezabilitatea acestei abordări.

<sup>2</sup>După ce am demonstrat că aplicarea modelelor decizionale în agregarea resurselor este fezabilă și utilă am continuat cercetarea [35] cu o analiză comparativă a câtorva modele decizionale din categoriile importante prezentate în literatură [1, 16].

să recomandăm un model decizional care se potrivește unui sistem cu o arhitectură P2P caracterizat de un set de parametri precum înzestrarea cu resurse a participanților și talia cererilor de resurse.

Acest capitol este structurat după cum urmează. În Secțiunea 1.2 descriem sistemul P2P și parametrii urmăriți în evaluarea eficacității procesului de agregare a resurselor. Secțiunea 1.4 prezintă rezultatele simulărilor efectuate și analiza comparativă a modelelor decizionale menționate anterior, urmând ca Secțiunea 1.5 să concluzioneze asupra discuției din acest capitol.

## 1.2 Structura Sistemului și Parametrii Urmăriți

În această secțiune prezentăm structura sistemului și mecanismul de agregare a resurselor.

Sistemul discutat este compus dintr-un set de  $N$  participanți, dispusi după o arhitectură P2P nestructurată. Fiecare nod deține o cantitate de resurse și este conectat (cunoaste) un subset de alți participanți, numiți vecini. În consecință, structura sistemului arată ca un graf conectat. Aceasta este stabilită a-priori, în sensul că ramane neschimbată pentru o rundă de experimente. Așadar, înainte de fiecare rulare, generăm structura de graf prin stabilirea pentru fiecare nod a unui subset de vecini aleși aleator din setul de participanți.

Fiecare nod  $p_i$  deține cantitatea  $q_i$  din resursa  $R$ . Resursa  $R$  (care poate fi un serviciu) este caracterizată de un set de atribute (proprietăți)  $\{is_1, \dots, is_k\}$ . Aceste atribute pot fi de exemplu prețul, calitatea etc. și pot lua valori numerice  $\{v_{i,1}, v_{i,2}, \dots, v_{i,k}\}$ , specifice nodului  $p_i$  care partajează resursa.

În experimentele noastre luăm în considerare diverse înzestrări cu resurse ale sistemului. Astfel, cantitatea de resurse poate avea diverse distribuții în sistem.

Pe această structură P2P construim funcționalitatea de agregare a resurselor utilizând două mecanisme: descoperirea resurselor și compunerea serviciului. În faza de descoperire a resurselor, procesul începe cu un nod - numit *inițiator* - care lansează o cerere de resurse pentru cantitatea  $Q_d$  din resursa  $R$  pentru o durată de timp  $T_d$ . În structura noastră presupunem existența unui mecanism de descoperire a resurselor a cărui rol este găsirea nodurilor care intenționează să partajeze resurse [44]. Mecanismul de descoperire a resurselor are o limitare intrinsecă, în sensul că poate descoperi o fracțiune  $f$  din totalul numărului de noduri.

În continuare, mecanismul de compunere a serviciului este aplicat pentru a selecta pe baza resurselor descoperite un subset de potențiali parteneri de tranzacție. Dacă inițiatorul nu reușește să agrege întreaga cantitate de resurse cerută, cererea este considerată ca fiind esuată.

Timpul  $T_d$  legat de o cerere de resurse indică durata în unități de timp pe care un inițiator ocupa resursele selectate. Când un nod  $p_i$  partajează propriile resurse pe durata de timp  $T_d$ , pe această durată nu va mai putea oferi resursele sale ca răspuns la noi cereri lansate în sistem de alți participanți. Pentru simplitate, presupunem că fiecare inițiator poate estima timpul  $T_d$  pentru o cerere de resurse, iar dacă  $T_d$  nu este suficient pentru consumarea serviciului, inițiatorul va lansa o nouă cerere de resurse pentru o nouă durată de timp.

Există diverse modele pentru mecanisme de descoperire a resurselor în arhitecturi P2P [44]. Printre acestea, considerăm mecanismul de difuzare (broadcasting) de mesaje. Cu ajutorul acestui mecanism, fiecare cerere de resurse este difuzată de inițiator în rețeaua sistemului cu o anumită valoare pentru parametrul TTL (time-to-live parameter) care indică durata de viață a mesajului. Parametrul TTL este strâns legat de gradul de conectivitate al rețelei. Acești doi parametri determină numărul de noduri vizitate de o cerere de resurse lansată de inițiator - numit și orizontul cererii.

Distingem aici și orizontul teoretic al unei cereri, care poate fi calculat pe baza mărimii și topologiei rețelei și valoarea parametrului TTL. Orizontul real este numărul total de noduri distincte care răspund unei cereri. Calculul orizontului teoretic este deseori irelevant [39] din moment ce nu se pot lua în calcul



grafurile ciclice care se formează în structura sistemului și gradele diferite de conectivitate la diverse salturi în parcurgerea grafului. Așadar, în studiul nostru folosim orizontul real, la care ne vom referi simplu prin orizont. Fiind măsurat de sistem, valoarea acestui parametru este precisă.

Pentru experimentele realizate am folosit mecanismul determinist *simple-flooding broadcasting* ca și cel prezentat în [34]. Acest mecanism este adecvat cerințelor noastre din moment ce există o relație directă între orizontul unei cereri și valoarea parametrului TTL.

O rundă de experimente conține mai multe cereri de resurse, fiecare dintre acestea fiind lansată în sistem la intervale de timp distincte. Pentru o cerere de resurse, un nod  $p_i$  este ales aleator, acesta inițiind în continuare o cerere pentru cantitatea de resurse  $Q_d$  solicitată pentru  $T_d$  unități de timp. Mecanismul de descoperire a resurselor returnează o listă cu potențiali parteneri. În continuare, inițiatorul aplică un model decizional pentru a selecta participanții de la care va obține în final resurse. După această selecție este realizată măsurarea eficacității alegerii facute, după care resursele selectate sunt blocate pentru  $T_d$  unități de timp. Acest scenariu este aplicat de mai multe ori, urmând ca în final să fie calculată eficacitatea totală și utilitatea globală.

### 1.3 Evaluarea Mecanismului de Descoperire a Resurselor

În această secțiune prezentăm criteriile de evaluare urmărite în stabilirea eficacității procesului de agregare de resurse.

Evaluarea este realizată individual, la nivel de cerere de resurse și global, la nivel de rundă de experimente.

După execuția mecanismului de agregare de resurse, inițiatorul deține o listă cu parteneri de la care va obține resurse partajate, ca rezultat al cererii introduse în sistem. Cum resursele disponibile în sistem fluctuează în permanență în ceea ce privesc nodurile care le partajează și cantitatea disponibilă, mai multe cereri lansate consecutiv în sistem generează rezultate diferite. În consecință, vom urmări utilitatea resimțită de inițiator ca urmare a cererii lansate în sistem. Rezultatul constă într-un număr de parteneri selectați  $N_p$ , prețurile individuale  $P_i$ ,  $i = \overline{1, N_p}$  cerute de parteneri în schimbul resursei partajate și cantitățile de resurse partajate  $Q_i$ . Inițiatorii sunt interesați de:

- agregarea întregii cantități de resurse  $Q_d$ ,
- minimizarea prețului total plătit,
- minimizarea riscului asociat unei tranzacții. În cazul nostru, riscul crește odată cu numărul de parteneri selectați pentru o tranzacție.

Formula 1.1 prezintă the 'utilitatea' individuală calculată pentru o cerere.

$$U_d = \frac{1}{N_p} \times \frac{1}{\sum_{i=1}^{N_p} P_i} \times \frac{1}{\sum_{i=1}^{N_p} Q_i} \quad (1.1)$$

Cu cât mai mare este utilitatea obținută în urma unei cereri, cu atât mai bine. Mai multe utilități obținute în urma tuturor cererilor pot fi agregate pentru a obține utilitatea globală (la nivel de rundă de experimente)  $U_g$  și este caracteristică unui set de valori date parametrilor cheie discutați. Utilitatea globală caracterizează conceptul de bunăstare socială, prezentat în [7].

Pe lângă utilitatea descrisă mai sus, contabilizăm și numărul de cereri care eșuează - deci nu se reușește agregarea întregii cantități de resurse solicitate. Deasemenea, urmărim costul (dat de atribute precum prețul) total la care o cerere este rezolvată. Din punctul de vedere al consumatorului de resurse obiectivele

sunt minimizarea costului și a riscului. Pentru participanții care partajează resurse însă, obiectivul este de a maximiza prețul primit în schimbul resurselor.

## 1.4 Experimente și Rezultate

În această secțiune descriem experimentele și comentăm rezultatele studiului nostru. Experimentele au fost realizate pe un simulator bazat pe mesaje [37] conceput pentru rețele P2P, a cărui versiune inițială a fost implementată în cadrul Faculté Polytechnique de Mons, Belgia<sup>3</sup>.

Folosim deci pentru experimentare o structură P2P pe care am descris-o în Secțiunea 1.2 împreună cu mecanismul de broadcasting pentru descoperirea resurselor. În continuare prezentăm setul de parametri cheie urmăriți în experimente:

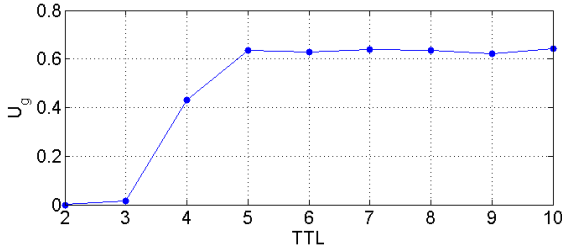
- parametrul TTL care caracterizează mecanismul de broadcasting;
- gradul de conectivitate ( $D_c$ ), reprezentând numărul de vecini al unui nod;
- orizontul unei cereri ( $H_d$ ), însemnând numărul de participanți potențiali parteneri pe care inițiatorul îi descoperă; valoarea acestui parametru ne dă și valoarea fracțiunii  $f$ , care, raportat la numărul total de noduri dă gradul de acoperire al mecanismului de descoperire de date;
- numărul de ofertanți selectați ( $N_p$  - așa cum a fost descris în secțiunea precedentă);
- cantitatea de resurse cu care un nod este dotat  $q_i$ ;
- numărul total de mesaje generate în sistem ( $N_m$ ) difuzate pentru o anumită cerere de resurse introdusă în sistem. Acesta reprezintă o măsură de cost a mecanismului de descoperire a resurselor;
- cantitatea de resurse cerută  $Q_d$ , specifică unei cereri; aceasta poate fi (i) mică, (ii) mare sau uniform distribuită între valori mici și mari; Fiecare cerere satisfăcută va conține setul de resurse blocate pentru următoarele  $T_d$  cereri lansate în sistem, unde  $T_d$  ia o valoare aleatoare între 2 și 10. Cantitatea de resurse cerută într-o cerere dă gradul de încărcare a rețelei ca în [8];
- rata eșecului ( $R_f$ ), care este numărul de cereri eșuate dintr-o rundă de experimente pentru un scenariu stabilit.

Experimentele au fost realizate pe trei tipuri de rețele cu privire la distribuția mărimii  $q_i$ . Notand  $QN$  cantitatea totală de resurse disponibile în sistem, cele trei tipuri de rețele sunt date de:

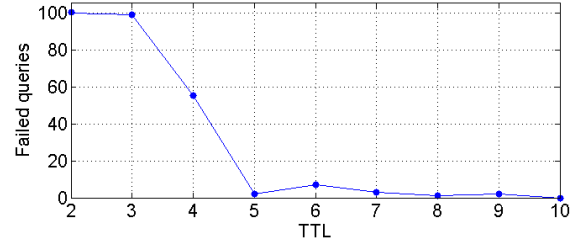
- $q_i \sim \frac{1}{QN}$  (distribuție uniformă),
- $q_i \sim Pois(1)$  (distribuție Poisson cu parametrul  $\lambda = 1$ ) - unde putine noduri dețin cantități mari de resurse și un număr mare de noduri dețin cantități mici de resurse;
- $q_i \sim Pois(4)$  (distribuție Poisson cu parametrul  $\lambda = 4$ ) - unde majoritate nodurilor dețin cantități medii de resurse  $\frac{1}{QN}$  și doar un număr mic de noduri dețin cantități mici și mari de resurse.

---

<sup>3</sup>Ii mulțumim lui Sebastien Noel de la Faculté Polytechnique de Mons, Belgia pentru amabilitatea sa de a ne permite folosirea versiunea inițială a simulatorului de rețele P2P și pentru ajutorul sau în timpul extinderii simulatorului pe care am realizat-o.



(a)



(b)

Figura 1.1: a) Total utility  $U_g$  as a function of TTL; b) The failure rate  $R_f$ .

Pentru restul prezentării din acest capitol, un scenariu (sau rundă de experimente) este dat de un set de 100 de cereri inițiate de participanți aleși aleator din sistemul format din 500 de noduri. Rezultatele prezentate în continuare (și în forma completă în teză) scalează proporțional cu numărul de participanți, păstrând aceeași topologie.

În continuare vom prezenta selectiv experimentele realizate și vom discuta asupra rezultatelor obținute.

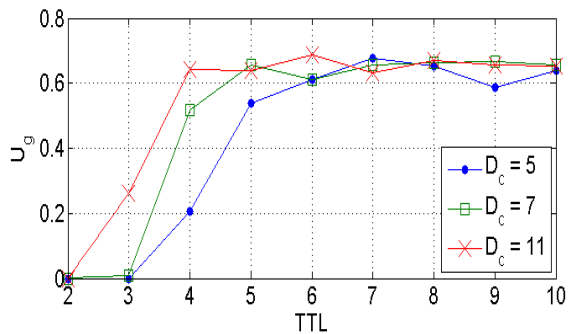
În primul rând am studiat modul în care parametrul TTL influențează utilitatea globală din moment ce valori mai mari pentru TTL duc la un orizont mai mare. Astfel, pentru valori fixe ale parametrilor  $D_c$  și  $Q_d$  rulăm scenarii pentru diverse valori ale parametrului TTL. Utilitatea globală  $U_g$  crește odată cu valoarea dată TTL, evoluție prezentată în figura 1.1a. În figura 1.1b arătăm rata de eșec  $R_f$  care descrește odată cu creșterea valorii TTL. Identificăm așadar trei etape în evoluția valorii pentru  $U_g$  ca rezultat al modificării valorii pentru TTL. În primul rând, pentru valori foarte scăzute pentru TTL,  $U_g$  este 0, din moment ce  $R_f$  este 100%. Înseamnă că orizontul este insuficient pentru ca inițiatorul să descopere cantitatea de resurse  $Q_d$ . Începând cu o anumită valoare a parametrului TTL, valoarea  $U_g$  crește după care se stabilizează,  $R_f$  descrescând.

Păstrând setările anterioare, rulăm scenarii cu diferite valori pentru  $Q_d$ , așadar lansăm în sistem cereri pentru diverse cantități de resurse. În figura 1.1a prezentăm modelul după care utilitatea globală evoluează funcție de valoarea TTL. În orice caz, observăm că pentru valori mari ale  $Q_d$ ,  $U_g$  începe să crească doar pentru valori semnificative ale TTL, deci costuri mai ridicate.

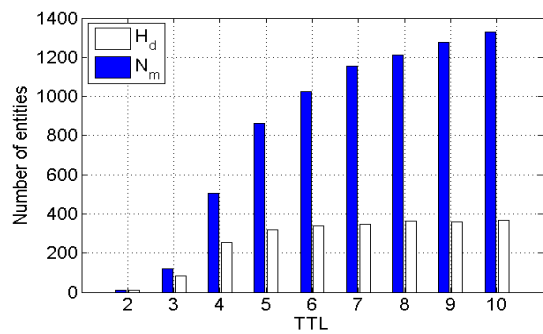
Gradul de conectivitate este un alt parametru care determină amploarea orizontului unei cereri. Astfel, pentru scenarii similare cu cele anterioare, în care considerăm un interval de valori pentru TTL și  $Q_d$  cu valoare fixă, urmărim valoarea pentru  $U_g$  pentru diverse grade de conectivitate (diverse valori pentru  $D_c$ ). Figura 1.2a prezintă curba utilității globale  $U_g$  pentru trei valori distincte ale  $D_c$ , valoarea pe care o ia  $Q_d$  rămânând fixă. Observăm că valorile mai mari ale  $D_c$  îmbunătățesc utilitatea globală obținută, în sensul că etapele de creștere și stabilizare apar mai repede pentru valori mai mici ale TTL comparativ cu scenariul anterior.

O analiză mai detaliată asupra utilității globale din perspectiva costurilor de comunicații, ne conduce la afirmația că atât parametrul  $D_c$  cât și TTL merită setați la valori mai mari însă până la un anumit nivel. Acest prag este de dat de începutul zonei de stabilizare a valorii calculate pentru  $U_g$  pentru anumite setări ale rețelei. După acest prag, valorile mai mari ale  $D_c$  și TTL aduc doar costuri suplimentare fără a îmbunătăți valoarea utilității globale. Figura 1.2b prezintă orizontul  $H_d$  care devine relativ constant începând cu o anumită valoare TTL, deși numărul de mesaje difuzate în sistem continua să crească în continuare odată cu creșterea valorii parametrului TTL. În consecință, când se realizează designul unui sistem real, valorile parametrilor  $D_c$  și TTL ar trebui ajustate astfel încât să se obțină maximizarea  $U_g$  și să se evite retransmisiile inutile de mesaje.

În teză prezentăm mai multe experimente, pentru diverse tipuri de rețele cu privire la distribuția can-



(a) Utilitatea globală.



(b) Orizontul și numărul de mesaje.

Figura 1.2: Castig și cost funcție de TTL, pentru diverse grade de conectivitate.

tității de resurse și alți parametri cheie. Deasemenea, comparăm cele două versiuni ale modelului Onicescu: versiunea subiectivă și cea obiectivă, observând că resursele agregate pot fi adaptate (conform valorilor pentru atributele care le caracterizează) specificului cererii introduse în sistem. Acest lucru este posibil datorită capacității versiunii subiective a modelului Onicescu de a lua în considerare preferințele inițiatorului cu privire la atributele resurselor.

**Analiză Comparativă Între Diverse Modele Decizionale.** În această secțiune prezentăm sumar rezultatele analizei comparative între diverse modele decizionale. Modelele alese sunt reprezentative pentru categoriile principale de modele decizionale care se aplică în contextul nostru.

În primul rând studiem rezultatele obținute (utilitatea globală) de modele în funcție de TTL (figura 1.3). Observăm că valori mai mari ale TTL duc la un orizont mai larg, și deci la o utilitate globală mai mare. Totodată, rulăm scenarii pentru fiecare model decizional, pentru valori ale  $Q_d$  uniform distribuite și pentru același interval de valori TTL ca și în experimentele prezentate anterior.

Observăm că utilitatea totală  $U_g$  crește odată cu valoarea TTL (figura 1.3). Deasemenea, observăm că diversele modele decizionale produc rezultate similare indiferent de distribuția valorilor  $q_i$ , excepție făcând modelul din categoria MAUT decision aid. Așadar, acesta obține rezultate mai bune când  $q_i$  are o distribuție  $Pois(1)$ , aproape egalând performanțele obținute cu modelul Promethee. Pentru celelalte tipuri de distribuție ale  $q_i$  Promethee, produce rezultate mai bune în ceea ce privește utilitatea globală.

În continuare, pentru aceleași setări de scenariu, vom scoate în evidență evoluția parametrilor  $H_d$  și  $N_p$  (figura 1.4). În figura 1.4 am considerat  $q_i \approx Pois(4)$ , corespunzând situației din figura 1.3c. Orizontul cererii și numărul de parteneri selectați este relativ similar și pentru celelalte modele decizionale. Aceasta înseamnă că modelul Promethee reușește să obțină valori mai bune pentru utilitatea globală nu prin minimizarea costurilor ci prin reducerea riscului asociat unei tranzacții.

Din figura 1.4, extragem o concluzie interesantă: modelul Promethee obține cea mai mare utilitate globală cu mai puțini parteneri selectați (figura 1.4b), lăsând posibilitatea formării unui orizont mai larg următoarelor cereri lansate în sistem (figura 1.4a).

În continuare studiem modul în care încărcarea rețelei influențează utilitatea globală. Astfel, pentru un interval de valori date parametrului TTL, rulăm experimente cu valori scăzute / ridicate pentru  $Q_d$ . Figura 1.5 prezintă rezultatele.

Observăm așadar că complexă pentru setările în care rețeaua are un grad mare de încărcare (figura 1.5a), tot modelul Promethee produce cele mai bune rezultate. Setările în care rețeaua are un nivel scăzut de încărcare arată cât de ineficient este un model simplu (figura 1.5b) precum Onicescu. Acest model (spre

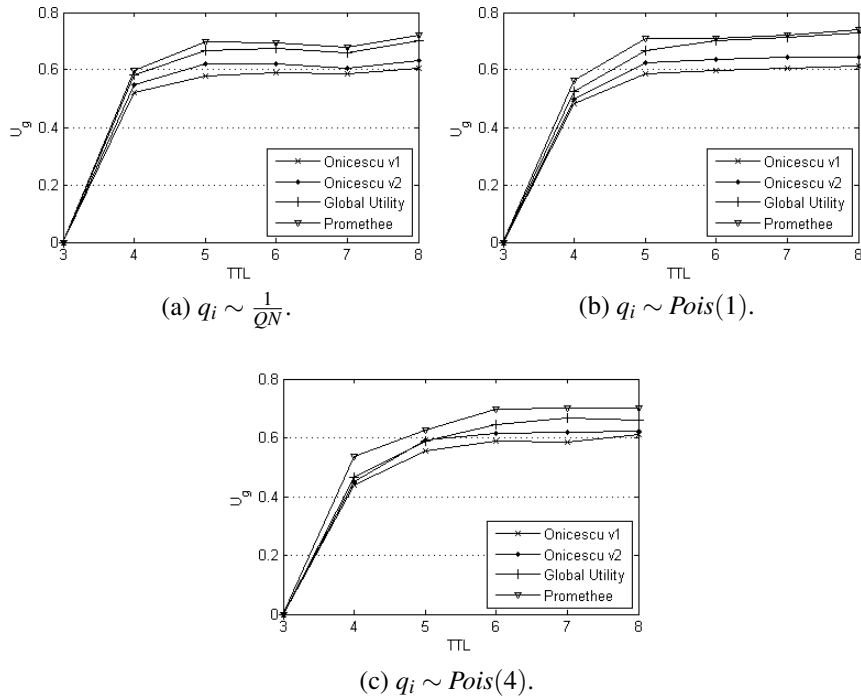


Figura 1.3: Curbele utilității totale pentru diverse distribuții ale  $q_i$ .

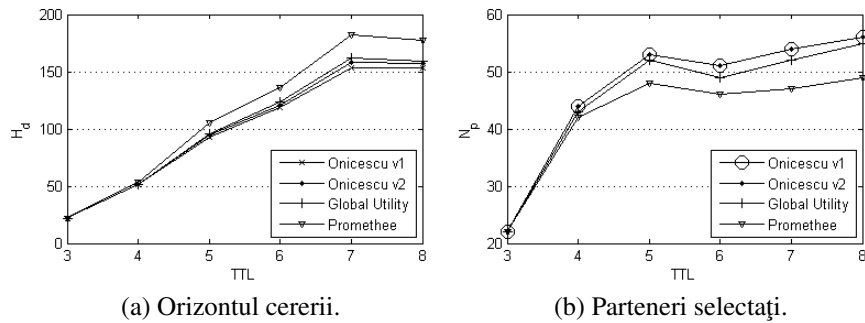


Figura 1.4: Variația orizontului și a numărului de parteneri selectați.

deosebire de Promethee) pierde din informația dată de valorile parametrilor diverselor alternative datorită ordonării valorilor atributelor realizate în fazele timpurii ale algoritmului.

## 1.5 Concluzii

În acest capitol am prezentat sumar studiul nostru cu privire la îmbunătățirea procesului de agregare a resurselor într-un sistem P2P nestructurat. Agregarea resurselor este procesul prin care un nod al rețelei colectează o anumită cantitate de resurse de la alți participanți ai sistemului.

Am observat că rețelele P2P nestructurate echipate cu un mecanism de descoperire a resurselor cu un anumit orizont sunt capabile să satisfacă (rezolve) cererile lansate în sistem de inițiatori. Prin experimentare se pot afla valorile corecte pentru setul de parametri cheie ai rețelei esențiali în construirea unui sistem real.

Mărirea gradului de conectivitate duce la creșterea puterii de descoperire a resurselor, însă trebuie ținut

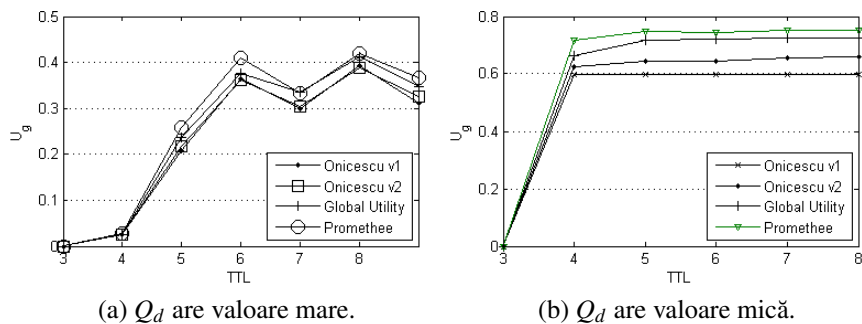


Figura 1.5: Utilitatea globală pentru încărcări mari / mici ale rețelei.

cont de nivelul optim al gradului de conectivitate pentru că altfel oricum nu se obține o utilitate globală mai mare, ci doar costuri mai mari.

Promethee este puternic subiectiv, fiecare decident putând să specifice propriile funcții de subiectivitate stabilind relații de prioritate sofisticate între criteriile de evaluare ale alternativelor. Am observat totodată că și modelul parametric din categoria MAUT produce rezultate mai bune decât modelul Onicescu pentru majoritatea scenariilor rulate, datorită funcțiilor de subiectivitate cu care modelul este dotat.

Promethee are performanțe superioare metodei MAUT datorită faptului că ordonarea alternativelor obținută în urma comparațiilor pe perechi de valori ale atributelor între alternative este superioară clasamentelor calculate la nivel global.

În cercetările viitoare suntem interesați de analiza performanțelor acestor algoritmi pe grupări sofisticate de resurse, în discuția actuală având doar un tip de resursă agregată.

Deasemenea, reputația ar juca un rol important în luarea deciziei de agregare de resurse într-un sistem P2P. Costurile reale ale unei tranzacții ar putea fi reduse de selecția bună a partenerilor, aceasta însemnând selectarea unor parteneri credibili din punct de vedere al calității serviciului.

# Capitolul 2

## Prototip Pentru MapReduce în Sisteme Desktop Grid

### 2.1 Introducere

Sistemele Desktop Grid s-au dovedit a fi de succes ca sisteme cu capacitate mare de calcul [33]. inițial ele au fost concepute pentru aplicații de tip Bag-of-Tasks, care necesitau un spațiu de stocare scăzut, lărgime de bandă modestă și fără dependențe între sarcini (sarcini de executat). O paletă largă de probleme, simulări în general, construirea indecșilor sau data mining sunt caracterizate de volume mari de date de intrare, rezultate intermediare și un grad mare de reutilizare a datelor. Aceste tipuri de probleme pot fi rezolvate cu ajutorul paradigmei MapReduce<sup>1</sup>. Modelul MapReduce este descris pe larg în teză. Așadar, existența unui mediu de rulare MapReduce pe o infrastructură Desktop Grid ar reprezenta o soluție accesibilă pentru rezolvarea problemelor din tipurile enunțate mai sus.

Data fiind cercetarea de ultimă oră în sisteme middleware pentru Desktop Grid, dezvoltarea unei soluții MapReduce pentru Desktop Griduri reprezintă o țintă care presupune provocări semnificative. În timp ce Desktop Griduri tradiționale precum BOINC [2], XtremWeb [14] sau Condor [33] au fost create pentru aplicațiile de tip Bag-of-Tasks care au un necesar redus de operații I/O, aplicațiile MapReduce procesează un volum semnificativ de date de intrare și rezultate intermediare, deci necesită o abordare diferită.

După faza de Map, modelul MapReduce presupune operații colective pe fișiere pentru a gestiona rezultatele intermediare. Asigurarea operațiilor colective în Desktop Griduri reprezintă o provocare datorită volatilității nodurilor în rețea și a fluctuațiilor mari în ceea ce privește structura și caracteristicile sistemului.

O altă provocare este dată de faptul că unele componente din Desktop Grid trebuie să fie descentralizate. De exemplu, un aspect de securitate important este verificarea rezultatelor [38], necesara pentru a asigura corectitudinea rezultatelor finale obținute de sistem. Deoarece rezultatele intermediare necesită un spațiu de stocare semnificativ, transmiterea lor pe un server ar fi prea costisitoare, așadar mecanismul de verificare a rezultatelor nu poate fi centralizat, așa cum este el actualmente implementat în Desktop Gridurile existente.

Dependențele între sarcinile Map și Reduce, împreună cu volatilitatea nodurilor și existența nodurilor leneșe pot scădea semnificativ performanțele unei execuții MapReduce în Desktop Grid. Așadar, am identificat nevoia existenței unei soluții optimizate care să răspundă cerințelor enunțate.

În acest capitol prezentăm sumar prototipul MapReduce pentru Desktop Grid pe care l-am realizat.

---

<sup>1</sup>MapReduce este un model de programare și o implementare asociată pentru procesarea și generarea seturilor mari de date [10]

Sistemul real pe care îl vom prezenta este rezultatul grupului de lucru MapReduce<sup>2</sup>, din care facem parte.

Prototipul nostru este bazat pe middlewareul BitDew [15], dezvoltate în cadrul INRIA și care este un mediu programabil pentru managementul automat și transparent de date în sisteme de tip Grid, Desktop Grid și Cloud. BitDew<sup>3</sup> este prezentat mai detaliat în teză.

În continuarea acestui capitol vom prezenta evaluarea sistemului care a asigurat validarea primei versiuni a prototipului [43], după care vom prezenta concluziile.

## 2.2 Evaluarea Performanțelor Prototipului

În această secțiune vom prezenta evaluarea performanțelor prototipului nostru de MapReduce. Pentru a măsura cu precizie performanțele obținute, am realizat experimentele într-un mediu unde rezultatele sunt reproductibile. Așadar, am folosit clusterul INRIA *Grid Explorer (GdX)* care reprezintă un fragment din infrastructura Grid5000 [5]. GdX este alcătuit din 356 eServere IBM dotate cu procesoare AMD Opteron cu viteza de 2.4Ghz, cu 2GB RAM, interconectate prin legături Gigabit ethernet. Experimentând, am realizat că majoritatea nodurilor au harddisk cu capacități disponibile între 5 și 10 GB. Pentru a emula un Desktop Grid pe GdX, am generat pene în sistem prin oprirea proceselor pe anumite noduri, iar pentru a simula eterogenitatea resurselor am lansat pe anumite noduri un număr variabil de procese concurente.

### 2.2.1 Comunicații Colective

DataCollection și DataChunk sunt două facilități noi adăgate protocolului BitDew, care sunt foarte utile implementării MapReduce, dar nu numai. În primul rând, am realizat experimente pentru a determina talia optimă a unui calup (fragment de fișier) în momentul în care se dorește transmiterea unui fișier mare. Testarea mecanismului a fost realizată prin rularea unui benchmark de tip Ping Pong folosind un fișier de 2.7GB și variind dimensiunea calupului între 5MB și 2.7GB. În consecință, numărul de calupuri a variat între 540 și 1. Pentru aceste teste am utilizat protocolul FTP. În urma testelor am observat că talia optimă a calupului pentru contextul particular BitDew este de 100MB.

### 2.2.2 Evaluarea Implementării MapReduce

În continuare prezentăm evaluarea implementării noastre MapReduce. Ca și aplicație de benchmark am folosit WordCount, care este un exemplu reprezentativ de aplicație MapReduce, asemănătoare cu cea disponibilă în distribuția Hadoop. Aplicația WordCount contabilizează numărul de apariții ale fiecărui cuvânt dintr-o colecție mare de documente (fișiere text). Pentru experimentare am folosit protocolul HTTP.

În primele experimente am urmărit evaluarea scalabilității implementării noastre pe măsură ce creștem numărul de noduri participante la sistem. Fiecare nod are de procesat (pentru o sarcină Map) un fișier de 5GB, împărțit local în 50 de calupuri. Pentru 512 noduri<sup>4</sup>, aplicația de benchmark procesează un volum de date totalizând 2.5TB și execută aproximativ 50000 de sarcini Map și Reduce. Figura 2.1 prezintă valoarea throughput corespunzătoare aplicației de benchmark WordCount exprimată în MB/s versus numărul de entități worker. Aceste rezultate arată că soluția propusă este scalabilă (scalabilitate scale-up) și ilustrează potențialul Desktop Gridului de a procesa cantități mari de date.

---

<sup>2</sup>Bing Tang (Wuhan University of Technology, China), **Mircea Moca** (Universitatea Babeș-Bolyai, Cluj-Napoca, România), Stéphane Chevalier (Ecole Normale Supérieure de Lyon, Franța), Haiwu He și Gilles Fedak (INRIA, University of Lyon, Franța).

<sup>3</sup><http://www.bitdew.net/>.

<sup>4</sup>GdX are 356 noduri dotate cu proc. double core, deci pentru a măsura performanța pe 512 noduri am lansat în execuție 2 procese worker pe 256 de noduri fizice.



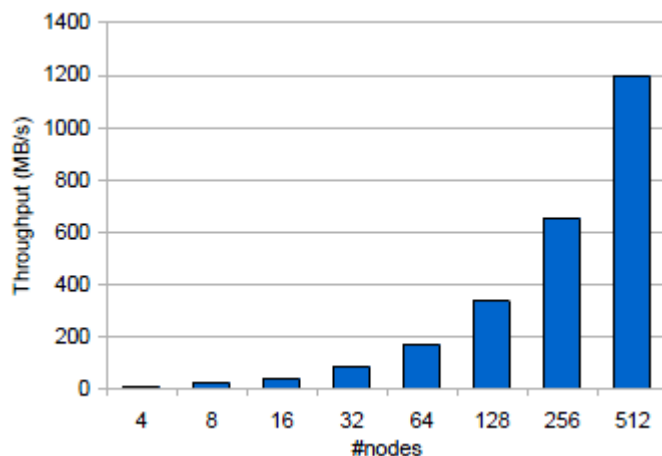


Figura 2.1: Evaluarea scalabilității pentru aplicația WordCount: axa  $y$  - throughput în MB/s și axa  $x$  - numărul de entități worker, variind între 1-512.

#Mappers	4	8	16	32	32	32	32
#Reducers	1	1	1	1	4	8	16
Map (sec.)	892	450	221	121	123	121	125
Reduce (sec.)	4.5	4.5	4.3	4.4	1	0.5	0
Timp total (sec.)	908	473	246	142	146	144	150

Table 2.1: Evoluția performanțelor funcție de numărul de mapperi și reduceri.

În următorul experiment evaluăm influența pe care o au diverse ponderi între numărul de noduri care execută sarcini Map și numărul de noduri care execută sarcini Reduce (echilibrul între mapperi și reduceri). Tabelul 2.1 prezintă timpul de execuție pentru funcția *Map*, timpul de execuție pentru funcția *Reduce* și timpul total de execuție al aplicației de benchmark, toate acestea pentru un număr de mapperi variind între 4 și 32 și un număr de reduceri variind între 1 și 16. Așa cum ne așteptam timpul de execuție atât pentru funcția Map cât și pentru funcția Reduce scade odată cu creșterea numărului de mapperi și reduceri. Diferența între timpul total de execuție și suma între timpul necesar Map și cel necesar Reduce se explică prin timpul necesar transmiterii de fișiere și ciclărilor. Deși timpul pentru Reduce pare foarte mic în comparație cu timpul pentru Map acest comportament este tipic pentru o aplicație MapReduce. Un studiu [30] pe aplicații MapReduce în domeniul științific pe clusterul Yahoo arată că mai mult de 93% dintre aplicații sunt doar Map sau majoritatea Map. Aceste observații ne-au dus la concluzia că echilibrul între mapperi și reduceri este important pentru utilizarea optimă a resurselor existente și acest raport trebuie adaptat la infrastructură disponibilă.

### 2.2.3 Specific Desktop Grid

În această secțiune, emulăm un Desktop Grid pe clusterul GdX prin crearea de scenarii în care introducem pene de sistem, noduri leneșe și conexiuni de Internet slabe.

În primul scenariu urmărim dacă sistemul nostru este rezistent la pene de sistem. În aceste scenarii considerăm că o parte din nodurile sistemului devin brusc inactive și funcționalitatea acestora este preluată de nodurile care rămân în sistem, finalizând cu succes execuția aplicației MapReduce. Pentru a demonstra această capacitate, propunem scenarii în care diferite noduri devin inactive la anumite momente de-a lungul execuției MapReduce. Astfel, prima pană ( $F_1$ ) apare la un nod în timp ce acesta descarcă un fișier

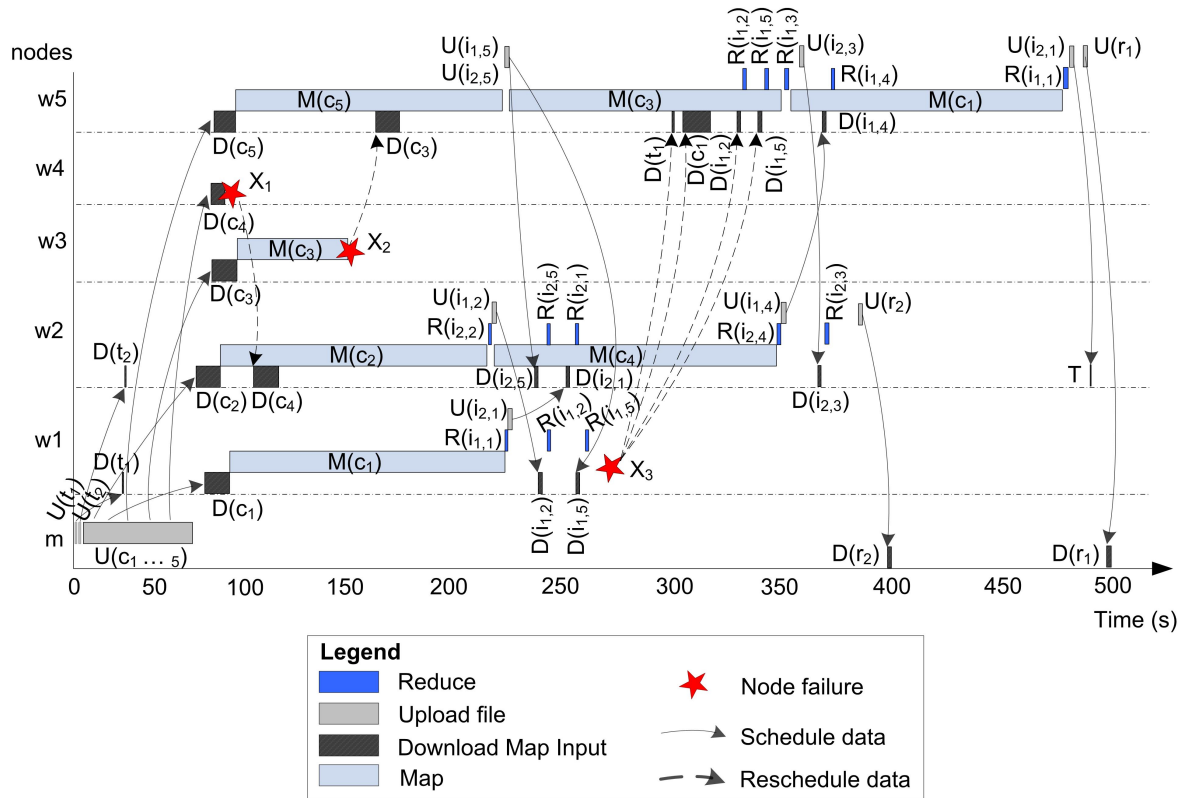


Figura 2.2: Execuție MapReduce cu scenarii de rezistență la pene de sistem.

asociat taskului Map, a două pană ( $F_2$ ) apare în timpul execuției funcției *Map*, iar a treia pană ( $F_3$ ) intervine după la un nod care a executat atât sarcini Map cât și Reduce. Scenariile au fost executate prin emularea penelor prin oprirea proceselor corespunzătoare pe anumite noduri. În figura 2.2 prezentăm printr-o diagramă Gantt o imagine fidelă a evenimentelor care au loc pe 5 dintre nodurile sistemului în timpul rulării scenariilor prezentate. Notăm cu  $w_1 - w_5$  nodurile și cu  $m$  nodul master. Execuția experimentului nostru începe cu masterul care încarcă fișierele inițiale și planificarea a două fișiere cheie (token files)  $U_{t_1}$  and  $U_{t_2}$  pentru stabilirea rolului de reducer.

Nodul  $w_1$  primește  $t_1$  iar nodul  $w_2$  primește  $t_2$ . Apoi, nodul master, încarcă și planifică fișierele de intrare pentru sarcinile *Map*, ( $U_{C_1 \dots 5}$ ). Fiecare nod descarcă un asemenea fișier eveniment notat în figura cu  $D(C_1) \dots D(C_5)$ . Nodul  $w_4$  cade ( $F_1$ ) în timp ce descarcă fișierul de intrare pentru *Map*  $D_{C_4}$ . Mecanismul de planificare cu care BitDew este dotat verifică periodic dacă participanții sunt încă prezenți în sistem, iar în cazul în care acesta detectează absența unui nod replanifică datele alocate lui altor participanți. Așadar, nodul  $w_4$  este considerat absent și datele  $C_4$  vor fi replanificate și vor fi trimise nodului  $w_2$ . Nodul  $w_3$  cade ( $F_2$ ) în timp ce execută o sarcină map  $M(C_3)$ . În consecință, calupul  $C_3$  este replanificat nodului  $w_5$ . La  $F_3$ , nodul  $w_1$  cade după ce a executat deja sarcini Map  $M(C_1)$  și mai multe sarcini Reduce:  $R_{F_{1,1}}$ ,  $R_{F_{1,2}}$  și  $R_{F_{1,5}}$ . Notația  $R_{F_{p,k}}$  se referă la sarcina Reduce care primește ca date de intrare rezultatul intermediar  $F_{i,k}$ , rezultat din procesarea calupului  $C_k$ .  $F_3$  determină sistemul de planificare al BitDew să replanifice fișierul cheie  $t_1$ , calupul  $C_1$  și rezultatele intermediare  $F_{1,2}$  and  $F_{1,5}$  nodului  $w_5$ .

După terminarea taskului Map  $M(C_1)$ , nodul  $w_5$  execută sarcina Reduce  $R(F_{1,1})$ . Pe urmă, încarcă și planifica fișierul intermediar  $F_{1,2}$  nodului  $w_2$ . Nodul  $w_2$  descarcă rezultatele intermediare  $D(F_{2,3})$ , dar fără a executa operația Reduce (simbolizată prin  $G$ ). Aceasta se datorează faptului că fiecare reducer păstrează o listă cu sarcini executate pentru un anumit input (date de intrare) și evită execuțiile redundante.

În teză mai prezentăm și alte experimente precum cel prin care urmărim impactul nodurilor leneșe

asupra timpului total de execuție. Testele realizate iau în considerare diverse ponderi de noduri în sistem care răspund lent cu rezultate. Pentru evitarea acestor efecte care există într-un sistem real, am prevăzut nodul master cu un mecanism de contabilizare a taskurilor pentru care generarea de rezultate întârzie, și ca soluție am ales replanificarea și trimiterea taskurilor în cauza altor noduri.

## 2.3 Concluzii

În acest capitol am prezentat relevanța scopului de a oferi o implementare MapReduce pentru sisteme Desktop Grid, folosind BitDew ca middleware. În această fază a cercetării am ținut validarea prototipului creat, și pentru aceasta am realizat o serie de experimente pentru a demonstra capacități precum: scalabilitatea, rezistența la pene, adaptarea sistemului la prezența nodurilor leneșe, planificarea sofisticată care să țină cont de disponibilitatea resurselor de calcul și altele.

Testele de scalabilitate au arătat că obținem o scalabilitate (în forma scale-up) liniară pe exemplul clasic de aplicație benchmark WordCount. Rularea sistemului în diverse scenarii sofisticate ne-au arătat că sistemul nostru este rezistent la pene și se adaptează la prezența nodurilor leneșe.

Am observat că, pentru ca sistemul să ruleze eficient este necesar ca datele de intrare să aibă un grad mare de partajare între membrii sistemului pentru a evita transferurile mari de date în rețea, și, mai mult, pentru ca datele să existe în locul în care ele sunt prelucrate. BitDew pune la dispoziția implementării noastre MapReduce posibilitatea partajării datelor folosind protocoale P2P.

Țintele noastre de cercetare propuse în viitorul apropiat sunt de a crea mecanisme și mai avansate de planificare a datelor, pentru a evita la maxim situațiile de replanificare de date, care în cazul MapReduce sunt costisitoare.

În teză am prezentat și un mecanism de verificare distribuită a rezultatelor, modelând rata erorii (agregată la nivel de sistem) cu care sistemul nostru, în condiții date produce rezultate eronate datorită nodurilor care sabotează execuția MapReduce.

În concluzie, am arătat că deși MapReduce este cu mult mai complexă decât aplicațiile clasice de tip Bag-of-Tasks, este posibil să construim un mediu de rulare eficient și sigur pentru a face posibilă execuția aplicațiilor care procesează un volum mare de date în sisteme Desktop Grid.

# Concluzii Finale

Am reușit în prima parte a acestui rezumat să evidențiem rolul important pe care îl are atributul de colaborativitate în sistemele de calcul distribuit folosind o arhitectură P2P nestructurată. În acest rezumat am prezentat sumar studiul nostru cu privire la îmbunătățirea procesului de agregare a resurselor într-un sistem P2P nestructurat. Agregarea resurselor este procesul prin care un nod al rețelei colectează o anumită cantitate de resurse de la alți participanți ai sistemului.

Am observat că rețelele P2P nestructurate echipate cu un mecanism de descoperire a resurselor cu un anumit orizont sunt capabile să satisfacă (rezolve) cererile lansate în sistem de inițiatori. Prin experimentare se poate afla valorile corecte pentru setul de parametri cheie ai rețelei esențiali în construirea unui sistem real.

Mărirea gradului de conectivitate duce la creșterea puterii de descoperire a resurselor, însă trebuie ținut cont de nivelul optim al gradului de conectivitate pentru că altfel oricum nu se obține o utilitate globală mai mare, ci doar costuri mai mari.

Promethee este puternic subiectiv, fiecare decident putând să specifice propriile funcții de subiectivitate stabilind relații de prioritate sofisticate între criteriile de evaluare ale alternativelor. Am observat totodată că și modelul parametric din categoria MAUT produce rezultate mai bune decât modelul Onicescu pentru majoritatea scenariilor rulate, datorită funcțiilor de subiectivitate cu care modelul este dotat.

Promethee are performanțe superioare metodei MAUT datorită faptului că ordonarea alternativelor obținută în urma comparațiilor pe perechi de valori ale atributelor între alternative este superioară clasamentelor calculate la nivel global.

În cercetările viitoare suntem interesați de analiza performanțelor acestor algoritmi pe grupări sofisticate de resurse, în discuția actuală având doar un tip de resursă agregată.

Deasemenea, reputația ar juca un rol important în luarea deciziei de agregare de resurse într-un sistem P2P. Costurile reale ale unei tranzacții ar putea fi reduse de selecția bună a partenerilor, aceasta însemnând selectarea unor parteneri credibili din punct de vedere al calității serviciului.

Deasemenea am prezentat relevanța scopului de a oferi o implementare MapReduce pentru sisteme Desktop Grid, folosind BitDew ca middleware. În această fază a cercetării am ținut validarea prototipului creat, și pentru aceasta am realizat o serie de experimente pentru a demonstra capacități precum: scalabilitatea, rezistența la pene, adaptarea sistemului la prezența nodurilor leneșe, planificarea sofisticată care să țină cont de disponibilitatea resurselor de calcul și altele.

Testele de scalabilitate au arătat că obținem o scalabilitate (în forma scale-up) liniară pe exemplul clasic de aplicație benchmark WordCount. Rularea sistemului în diverse scenarii sofisticate ne-au arătat că sistemul nostru este rezistent la pene și se adaptează la prezența nodurilor leneșe.

Am observat pentru ca sistemul să ruleze eficient este necesar ca datele de intrare să aibă un grad mare de partajare între membrii sistemului pentru a evita transferurile mari de date în rețea, și, mai mult, pentru ca datele să existe în locul în care ele sunt prelucrate. BitDew pune la dispoziția implementării noastre MapReduce posibilitatea partajării datelor folosind protocoale P2P.

Țintele noastre de cercetare propuse în viitorul apropiat sunt de a crea mecanisme și mai avansate de planificare a datelor, pentru a evita la maxim situațiile de replanificare de date, care în cazul MapReduce

sunt costisitoare.

În teză am prezentat și un mecanism de verificare distribuită a rezultatelor, modelând rata erorii (agregată la nivel de sistem) cu care sistemul nostru, în condiții date produce rezultate eronate datorită nodurilor care sabotează execuția MapReduce.

În concluzie, am arătat că deși MapReduce este cu mult mai complexă decât aplicațiile clasice de tip Bag-of-Tasks, este posibil să construim un mediu de rulare eficient și sigur pentru a face posibilă execuția aplicațiilor care procesează un volum mare de date în sisteme Desktop Grid.

# Rezultate

Rezultatele prezentate în teza rezumată în această lucrare, au fost diseminate prin intermediul următoarelor articole științifice:

## **Articole publicate în reviste de specialitate la nivel național:**

- Mircea Moca, Gheorghe Cosmin Silaghi, *A functional Sketch for Resources Management In Collaborative Systems for Business*, Analele Universității din Oradea - Științe Economice, pp 1447-1453, ISSN 1582-5450, 2008.
- Mircea Moca, Gheorghe Cosmin Silaghi, Tehnologii pentru gestiunea eficientă a resurselor de calcul în medii colaborative pentru afaceri, Studii și Cercetări Economice, Alma Mater, pp 371-379, ISBN: 978-606-504-035-9, Cluj-Napoca, 2008.

## **Articole publicate în volume ale conferințelor internaționale care au avut loc în România:**

- Mircea Moca, Gheorghe Cosmin Silaghi, *Analysis On Collaborative Aspects in P2P Architectures*, Annals of the "Tiberiu Popoviciu" Seminar, vol. 6b, International Workshop in Collaborative Systems and Information Society, pp 102-113, 2008.

## **Articole publicate în volume ale conferințelor internaționale din străinătate:**

- Bing Tang, Mircea Moca, Stéphane Chevalier, Haiwu He, Gilles Fedak, *Towards MapReduce for Desktop Grid Computing*, Fifth International Conference On P2P, Parallel, Grid, Cloud And Internet Computing, 2010, IEEE Computer Society.
- Mircea Moca, Gheorghe Cosmin Silaghi, *Decision Models for Resource Aggregation in Peer-to-Peer Architectures*, Proceedings of the CoreGrid-ERCIM-Working-Group on Grids, P2P and Services Computing held in Conjunction with EuroPar 2009, Delft, Netherlands, pp 105-117, LNCS, Springer USA, 2010, ISBN: 978-1-4419-6793-0.
- Mircea Moca, Gheorghe Cosmin Silaghi, *Resource Aggregation Effectiveness in Peer-to-Peer Architectures*, Proceedings of the 4th International Conference on Grid and Pervasive Computing, Advances In Grid And Pervasive Computing, pp 388-399, Geneva, Switzerland, 2009, LNCS, Springer-Verlag Berlin, ISSN: 0302-9743, ISBN: 978-3-642-01670-7.
- Mircea Moca, *Resource Management for a Peer-to-Peer Service Oriented Computing System*, Proceedings of the 10th European Agent Systems Summer School, pp 31-39, Lisbon, Portugal, 2008.

## **Alte articole:**

- Mircea Moca, Gheorghe Cosmin Silaghi, Gilles Fedak, *Characterizing errors in MapReduce for Desktop Grids*, INRIA Technical Report, Lyon, France, 2010.

# Bibliografie

- [1] M. Abdellaoui and D. Hey, J. *Advances in Decision Making Under Risk and Uncertainty*. Springer, 2008.
- [2] P. Anderson, D. BOINC: A System for Public-Resource Computing and Storage. In *Proceedings of the 5th IEEE/ACM International GRID Workshop*, Pittsburgh, USA, 2004.
- [3] T. Ashish, S. Joydeep Sen, J. Namit, S. Zheng, C. Prasad, A. Suresh, L. Hao, P. Wyckoff, and M. Raghotham. Hive: a warehousing solution over a map-reduce framework. *Proc. VLDB Endow.*, 2(2):1626–1629, 2009.
- [4] L Blasi, AE Arenas, B Aziz, P Mori, U Rovati, B Crispo, F Martinelli, and P Massonet. A secure environment for grid-based supply chains. 2008.
- [5] R. Bolze and all. Grid5000: A Large Scale Highly Reconfigurable Experimental Grid Testbed. *International Journal on High Peerformance Computing and Applications*, 2006.
- [6] P. Brans, J., M. Mareschal, and P. Vincke. How to select and how to rank projects: the promethee method. In *European Journal of Operational Research*, volume 2, pages 228–238, 1986.
- [7] Y. Chevaleyre, E. Dunne, P., U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, A. Rodríguez-Aguilar, J., and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.
- [8] L. Chunlin and L. Layuan. Multi economic agent interaction for optimizing the aggregate utility of grid users in computational grid. *Applied Intelligence*, 25(2):147–158, 2006.
- [9] U. Dayal, M. Castellanos, A. Simitsis, and K. Wilkinson. Data integration flows for business intelligence. In *EDBT '09: Proceedings of the 12th International Conference on Extending Database Technology*, pages 1–11, New York, NY, USA, 2009. ACM.
- [10] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI'04: Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, pages 137–149, USA, 2004. USENIX Association.
- [11] J. S. Dyer, P. C. Fishburn, R. E. Steuer, J. Wallenius, and S. Zionts. *Multiple Criteria Decision Making, Multiattribute Utility Theory: The Next Ten Years*, volume 38. INFORMS, 1992.
- [12] J. Ekanayake, S. Pallickara, and G. Fox. Mapreduce for data intensive scientific analyses. In *ESCIENCE '08: Proceedings of the 2008 Fourth IEEE International Conference on eScience*, pages 277–284, Washington, DC, USA, 2008. IEEE Computer Society.

- [13] Jaliya Ekanayake, Thilina Gunarathne, Judy Qiu, Geoffrey C. Fox, Scott Beason, Jong Youl Choi, Yang Ruan, Seung-Hee Bae, and Hui Li. Draft report: Applicability of dryadling to scientific applications. 10/16/2009 2009.
- [14] G. Fedak, C. Germain, V. Néri, and F. Cappello. XtremWeb: A Generic Global Computing Platform. In *Proceedings of 1st IEEE International Symposium on Cluster Computing and the Grid CCGRID'2001, Special Session Global Computing on Personal Devices*, pages 582–587, Brisbane, Australia, May 2001. IEEE/ACM, IEEE Press.
- [15] G. Fedak, H. He, and F. Cappello. BitDew: A Data Management and Distribution Service with Multi-Protocol and Reliable File Transfer. *Journal of Network and Computer Applications*, 32(5):961–975, September 2009.
- [16] J. Figueira, S. Greco, and M. Ehrgott. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer, 2005.
- [17] I. Foster. What is the grid? - a three point checklist. *GRIDtoday*, 1, July 2002.
- [18] I. Foster and A. Iamnitchi. On death, taxes, and the convergence of peer-to-peer and grid computing. In *Peer-to-Peer Systems II, Second Intl. Workshop, IPTPS 2003*, volume 2735 of *LNCS*, pages 118–128. Springer, 2003.
- [19] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15, 2001.
- [20] I. Foster and C. etc. Kesselman. The physiology of the grid: An open grid services architecture for distributed systems integration. 2002.
- [21] Greenplum. Greenplum mapreduce: A unified engine for rdbms and mapreduce, 2010.
- [22] <http://www.anl.gov>.
- [23] <http://www.bitdew.net/>.
- [24] <http://www.deisa.eu/>.
- [25] <http://xcrime.sourceforge.net/>.
- [26] L. Ilies, M. Mortan, D. Lungescu, I. Lazar, M. Popa, and V. Vereş. *Handbook of Management (in Romanian)*. Risoprint, 2006.
- [27] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly. Dryad: distributed data-parallel programs from sequential building blocks. In *EuroSys '07: Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, pages 59–72, New York, NY, USA, 2007. ACM.
- [28] S. Jiulong, C. Huaping, S. Guangzhong, and C. Xin. Vast: A service based resource integration system for grid society. In Jiannong Cao, Laurence Yang, Minyi Guo, and Francis Lau, editors, *Parallel and Distributed Processing and Applications*, volume 3358 of *LNCS*, pages 489–498. Springer Berlin / Heidelberg, 2005.
- [29] S. Karlin. *Mathematical methods and theory in games, programming, and economics*, volume III of *Addison-Wesley series in statistics*. Pergamon Press, London, 1959.



- [30] S. Kavulya, J. Tan, R. Gandhi, and P. Narasimhan. An analysis of traces from a production mapreduce cluster. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, Melbourne, Australia, 2010.
- [31] C. Kesselman and I. Foster. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, November 1998.
- [32] R. Kohavi, J. Rothleder, N., and E. Simoudis. Emerging trends in business analytics. *Commun. ACM*, 45:45–48, August 2002.
- [33] J. Litzkow, M., M. Livny, and W. Mutka, M. Condor - A Hunter of Idle Workstations. In *Proceedings of the 8th International Conference on Distributed Computing Systems (ICDCS)*, pages 104–111, Washington, DC, 1988. IEEE Computer Society.
- [34] I. Mkwawa and D. Kouvatsos. Broadcasting methods in mobile ad hoc networks: An overview. In *Technical Proc. of the Third Intl. Working Conf. HET-NETs 2005*, pages T9/1–14. Networks UK, 2005.
- [35] M. Moca and G. Silaghi. Decision models for resource aggregation in peer-to-peer architectures. In *Proceedings of the CoreGrid-ERCIM-Working-Group on Grids, P2P and Services Computing held in Conjunction with EuroPar 2009*, pages 105–117. SPRINGER, 233 SPRING STREET, NEW YORK, NY 10013, UNITED STATES, 2009.
- [36] M. Moca and G. Silaghi. Resource aggregation effectiveness in peer-to-peer architectures. In *Proceedings of the 4th International Conference on Grid and Pervasive Computing, ADVANCES IN GRID AND PERVASIVE COMPUTING*, pages 388–399. SPRINGER-VERLAG BERLIN, 2009.
- [37] S. Noel, P. Manneback, and C. Silaghi, G. Response deadline evaluation in point-to-point negotiation on grids. In *Grid Economics and Business Models workshop GECON 2009*, volume 5745, pages 15–27, Delft, Netherlands, 2009. Lecture Notes in Computer Science.
- [38] F. G. Sarmeta, L. Sabotage-Tolerance Mechanisms for Volunteer Computing Systems. *Future Generation Computer Systems*, 18(4):561–572, 2002.
- [39] Clip2 Distributed Search Services. *Gnutella Dynamic Query Protocol v0.4*. [http://www.stanford.edu/class/cs244b/gnutella\\_protocol.0.4.pdf](http://www.stanford.edu/class/cs244b/gnutella_protocol.0.4.pdf), 2003.
- [40] H. Stockinger. Grid computing: A critical discussion on business applicability. *J. Supercomput.*, 7(6):3–17, 2006.
- [41] H. Stockinger. Defining the grid: a snapshot on the current view. *J. Supercomput.*, 42(1):3–17, October 2007.
- [42] S. Tanenbaum, A. and V. Steen, M. *Distributed Systems Principles and Paradigms*. Pearson Education Inc., Upper Saddle River, New Jersey, USA, 2007.
- [43] B. Tang, M. Moca, S. Chevalier, H. He, and G. Fedak. Towards mapreduce for desktop grid computing. IEEE Computer Society, 2010.
- [44] P. Trunfio, C. Talia, H. Papadakis, P. Fragopoulou, M. Mordacchini, M. Pennanen, K. Popov, V. Vlassov, and S. Haridi. Peer-to-peer resource discovery in grids: Models and systems. *Future Generation Computer Systems*, 23(7):864–878, 2007.

- [45] T. Wanqing and J Weijia. End host multicast for peer-to-peer systems. In *Grid and Cooperative Computing 2004, LNCS 3251*. Springer-Verlag Berlin Heidelberg, 2004.
- [46] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media Inc., Sebastopol, CA 95472, first edition, 2009.
- [47] W. Xue, J. Shi, and B. Yang. X-rime: Cloud-based large scale social network analysis. In *Proceedings of the 2010 IEEE International Conference on Services Computing, SCC '10*, pages 506–513, Washington, DC, USA, 2010. IEEE Computer Society.
- [48] X. Yingjie, Z. Mingzhe, and Z. Nengneng. A collaborative application framework based on the p2p grid model. In *Proceedings of the 2010 Third International Conference on Knowledge Discovery and Data Mining, WKDD '10*, pages 63–66, Washington, DC, USA, 2010. IEEE Computer Society.
- [49] Y. Yuan, M. Isard, D. Fetterly, M. Budiu, U. Erlingsson, P. Kumar, and J. Currey, G. Dryadling: A system for general-purpose distributed data-parallel computing using a high-level language, 2008.