UNIVERSITATEA BABEŞ-BOLYAI, CLUJ-NAPOCA, ROMÂNIA

Facultatea de Ştiinţe Economice şi Gestiunea Afacerilor

Departamentul de Informatică Economică

Utilizarea resurselor de calcul distribuite pentru o execuţie eficientă a aplicaţiilor de business complexe

Rezumat teză de doctorat

Coordonator ştiinţific:
Prof. univ. dr. Nicolae Tomai

Doctorand:
Gabriela Andreea Morar

Specialitatea: Cibernetică şi Statistică

2012

Cuvinte cheie: *sisteme distribuite, cloud computing, performanță, workflow, Hadoop, domeniu de negociere, negociere de resurse*

Cuprins

Cuprins

Abstract

În ultimele decenii numărul oamenilor de știință și al companiilor ce se bazează pe sisteme de calcul distribuite pentru a rămâne competitivi pe piață a crescut semnificativ. Deoarece costurile operării unor super-computere proprii este crescut, nu toate companiile sau grupurile de cercetare își permit costurile generate de achiziția infrastructurii necesare rulării propriilor aplicații sau experimente. Astefel aceștia trebuie să apeleze la resurse provenite din surse externe. La început clusterele si grid-urile au fost cele care au atras atenția mediului academic și al celui economic, însă acestea aveau anumite limitări. Noul apărut cloud computing le promite acestora toate resursele necesare, în orice moment pe baza unui model de tipul plătește doar atât cât consumi. Astfel dorindu-se accesarea acestuia ca și o simpla utilitate precum: apa, electricitate, etc. Însă, a avea infrastructura necesară la îndemână nu este un lucru suficient în cazul tuturor, deaorece unii ditre ei nu dețin mijloacele sau cunoștiințele necesare pentru a profita de avantajele noilor tehnologii.

În această lucrare ne propunem să arătăm modul în care sitemele distribuite pot fi folosite în mod eficient atât de mediul economic cât și de către cel academic cu scopul de a crește performanța propriilor aplicații și de așsi reduce costurile de execuție. În acest sens vom prezenta două paradigme ce îi pot ajuta pe utilizatorii ce nu dispun de cunoștiințe avansate de programare paralelă să își paralelizeze aplicațiile: fluxuri de lucru(workflows)și MapReduce [9].

De asemenea vom prezenta și două studii de caz constând în creare a două fluxuri de lucru bazate pe aplicații reale și testarea performanței acestora în cazul rulării lor pe resurse de tip cloud. Având în vedere

faptul că fluxurile de lucru nu reprezintă soluția ideală în cazul tuturor utilizatorilor, am realizat și un studiu privind eficiența folosirii paradigmei MapReduce. Aceasta le permite utilizatorilor să proceseze cantități mari de date în mod distribuit. În cazul de fața ne-am îndreptat atenția asupra implementării oferite de către Hadoop [15] pentru această paradigmă și am testat perfomanța acestei implementări atunci când vine vorba de rularea unor algoritmi de clusterizare pe date obținute de la Twitter. Ne-am dorit să testăm cum evoluează performanța oferită de Hadoop în cazul folosirii unor medii diferite de stocare a datelor (solid state disk sau hard disk) și prin varierea valorilor anumitor parametri ce pot fi setați de către utilizator în momentul în care își configură clusterele de tip Hadoop.

După evaluarea performanței aplicațiilor ce rulează pe resurse distribuite de tip cloud, am identificat necesitatea creării unui mecanism de selecționare a resurselor dorite din cadrul unei varietăți de resurse existente, însă fără a depăși anumite costuri. Acest lucru trebuia realizat în așa manieră încât să fie aleasă varianta ce i se potrivește cel mai bine unui utilizator. Astfel, ne-a venit idea folosirii agenților inteligenți. În acest sens am implementat un domeniu de negociere specializat, bazat pe caracteristicile resurselor cloud de tip IaaS (Infrastructura ca și un Serviciu). Am luat în considerare un scenariu bazat pe un federated cloud academic, costând în colecție de cloud-uri private aparținând unor diverse instituții academice.

În acest sens am creat un mechanism, care pe baza datelor oferite de către cloud controller-ele acestor cloud-uri private, sa genereze într-un mod semi-automat profile de negociere. Generarea semi-automată a acestor profile de negociere poate fi considerată un prim pas în intenția noastră de a ajunge la punctul în care să facem posibilă aprovisionarea la timp a aplicațiilor cu resurse de tip cloud. Integrarea acestui mecanism în cadrul unei platforme reale face parte din dezvoltările ulterioare.

Cuvinte cheie: *sisteme distribuite, cloud computing, performanță,*

workflow, Hadoop, domeniu de negociere, negociere de resurse

Documentul de față conține un scurt rezumat al tezei de doctorat având titlul: "Utilizarea resurselor de calcul distribuite pentru o execuție eficientă a aplicațiilor de business complexe". Acest rezumat este menit să ofere o viziune de ansamblu asupra principalelor realizări obținute pe perioada studiilor doctorale.

1 Motivația cercetării

Îm zilele noastre a rămâne competitive pe piață este una din principalele probleme cu care au de a face companiile. În anumite situații acest lucru poate însemna reducerea costurilor și cheltuielilor generate de procesare unor cantități mari de date.

Nu doar mediul de producție ci și oamenii de știință se confruntă cu probleme similare. Aceștia trebuie sa optimizeze timpul de execuție și costurile generate de experimentele și simulările lor, deoarece au de a face, mai mereu, cu bugete limitate și termene limită.

În decursul ultimelor decenii, progresul înregistrat de către tehnologia de calcul a contribuit semnificativ la accelerarea progresului științific prin reducerea timpului de execuție al aplicațiilor științifice și de afaceri [17]. Anumite domenii de cercetare necesită o putere de calcul ridicată pentru a rula anumite experimente și pentru a valida anumite ipoteze de cercetare. Astfel, oamenii de știință trebuie să poată refolosi anumite programe software și să poată varia cu ușurință parametrii de start sau ipotezele pe care se bazează experimentele lor.

În cazul rulării experimentelor științifice putem identifica două probleme principale ce ar putea genera un eșec:

- lipsa unor idei științifice corecte sau neidentificarea experimentelor potrivite pentru validarea acestor idei.
- incapacitatea de a implementa aceste idei științifice astfel încât să faciliteze execuția experimentelor.

Nu toți oamenii de știință dețin aptitudinile necesare implementării unor aplicații ce trebuie să reunească unelte software-uri ce aparțin unor domenii diverse: unelte de analiză a datelor, unelte specifice unui anumit domeniu, etc.

Acest lucru este și mai dificil în momentul în care trebuie să ținem cont de resurse limitate, precum timpul și costurile financiare.

În ultimii ani și-au făcut apariția câteva tehnologi menite să le dea oemenilor de știință o mână de ajutor în ceea ce privește maximizarea folosirii resurselor de calcul pe care le dețin. În primul rând ținem să menționăm paradigma fluxurilor de lucru și multitudinea de Sisteme de Management a Fluxurilor de Lucru ce au fost implementate cu scopul de a ușura maparea execuției aplicațiilor științifice complexe pe resurse distribuite. Această soluție se potrivește în mod special aplicațiilor ce necesită o putere de calcul ridicată pentru ași încheia execuția într-o limită de timp rezonabilă. În al doilea rând dezbatem tehnica MapReduce folosită pentru îmbunătățirea timpilor de execuți pentru aplicațiile ce procesează cantități mari de date. Dintre toate inmplementările MapReduce existente pe piață ne-am îndreptat atenția asupra variantei oferite de către proiectul Hadoop.

Deși sistemele ditribuite apar ca și o soluție evidentă atunci când vine vorba de imbunătățirea timpilor de execuție a aplicațiilor, acestea nu sunt ușor de gestionat și implementarea unor aplicații care să ruleze pe ele este o provocare. Complexitatea și eterogenitatea acestor sisteme poate duce la apariția unei serii de probleme. În cadrul acestor sisteme eșecuri pot apărea în orice moment, în special atunci când acestea sunt compuse dintr-un număr mare de resurse eterogene legate între ele. Câteva posibile soluții pentru a evita astfel de situații ar pute fi: replicarea datelor în cazul aplicațiilor ce procesează cantități mari de date sau retrimiterea proceselor de prelucrat (job-urilor) în cazul apariției unor eșecuri în execuția unora dintre ele.

Există trei tipuri de sisteme distribuite care au atras atenția cercetătorilor și a oamenilor din industrie: clustere, grid-uri și cloud computing.

Grid-ul este o formă de sistem de calcul distribuit care este compus din calculatoare legate între ele care apar utilizatorului final ca un supercomputer. Un grid este compus din mașini de calcul eterogene care își oferă resursele de calcul publicului larg sub forma unor organizații virtuale (VOs) [1][14].

Recent, comunitatea științifică a manifestat un interes crescut in legătură cu Arhitecturile Orinetate pe Servicii (SOA) și virtualizarea. Pe baza acestor

[1]"Organizațiile virtuale sunt definite ca un set de idivizi, instituții menite să împărtășească resurse pe baza unui set de reguli"

tehnologi a luat naștere un nou tip de sistem distribuit: cloud computing.

Cloud computing a apărut ca o alternativă la tipurile de sisteme distribuite deja existente și încercă să solușioneze problemele de cost prin folosirea uni model de taxare de tipul pay-as-you-go (plătește atâta cât consumi) și prin punerea la dispoziția utilizatorilor a unor resurse elastice ce pot fi accesate de oriunde, prin intermediul Internetului, fără a necesita rezervări anterioare. În ultima decadă, o multitudine de companii au ales să își ofere resulsele proprii, hardware sau software, folosind paradigma de tipul everything-as-a-service (totul ca și un serviciu).

Costurile generate de crearea unor supercomputere proprii sunt considerabile, iar accesarea unor infrastructuri deja existente, precum grid-urile, necesită mult timp, cunoștiințe de specilaitate, permisiunea de acces din partea deținătorilor acestor resurse, planificarea și rezervarea lor in avans, etc. Datorita flexibilității cloud computing s-a bucurat de un interes crescut din parte mediului academic și a celui de afaceri, în timp ce grid-urile sunt folosite în special în mediul academic ca și medii de mari dimensiuni pentru stocarea informațiilor sau procesarea acestora.

Numărul furnizorilor de resurse de tip cloud existenți pe piață este intr-o continuă creștere, astfel apare necesitatea folosirii unor sisteme de management automatic a resurselor de tip cloud. Scopul acestra este de a înpune regulile legate de cererea și oferta resurselor de tip cloud [33]. Livrarea serviciilor de tip cloud computing necesită ca anumiți parametri legați de caliatetea serviciilor (QoS) să fi respectați astfel încât utilizatorii să îți poate atinge obiectivele și să își execute operțiunile în cele mai bune condiții. În acest context, un sitem de management al resurselor bazat pe SLA-uri (Service Level Agreement) este necesar pentru a negocia interacțiunea dintre participanții la tranzacții pe piața resurselor de tip cloud computing.

Cercetarea de fața își propune să investigheze problema aprovisionării cu resurse de tip cloud în contextul unor medii de tip federated cloud și ținând cont de anumiți QoS. Ne propunem să cercetăm modul în care poate fi soluționată problema negocieri resurselor în cadrul unei piețe concurențiale prin folosirea agenților inteligenți.

2 Obiectivele cercetării

În cadrul acestei lucrări ne propunem să studiem modul în care resursele de calcul, oferite de către sistemele de calcul distribuite, pot să îmbunătățească performanță în execuție a diverselor tipuri de aplicații, totodată reducând și costurile. Având în minte acest țel am studiat tipurile de sisteme distribuite existente, am comparat potrivirea acestora pentru rularea aplicațiilor complexe și am estimat costurile generate de către utilizarea lor. Pornind de la acest scenariu inițial au apărut câteva întrebări:

- Care este cel mai potrivit tip de resursă de caltul distribuită pentru o rularea unei anumite aplicații?
- Care este cel mai bun mod în care un utilizator obișnuit poate profita la maxim de aceste resurse?
- Cum inflențează caracteristicile tehnice a acestor resurese (numărul de coruri, memorie, hard disk, etc.) performanța aplicațiilor rulate pe ele?
- Cum să alegi cea mai bună opțiune pentru a rula o aplicație atunci când mai mulți furnizori de resurse de calcul sunt diponibili și totodată tinând cont de limitările de buget existente?

Acestea sunt principalele întrebări la care lucraea de față încercă sa dea un răspuns. În primul rând prin investigarea caracteristicilor programelor paralele, iar apoi prin identificarea principalele metode de paralelizare a palicațiilor fără a avea însă nevoi de aptitudini specile de programare paralelă. În plus vom evalua modul în care poate fi îmbunătățită performanța execuției aplicațiilor și cum pot fi reduse costurile execuției acestora prin utilizarea sistemelor distribuite. În final vom propune un mecanism care să îi ajute pe utilizatori să obțină resursele necesare atunci când au posibilitatea de a alege între mai mulți furnizori de resurse.

3 Structura tezei

Structura tezei reiese din figura 1:

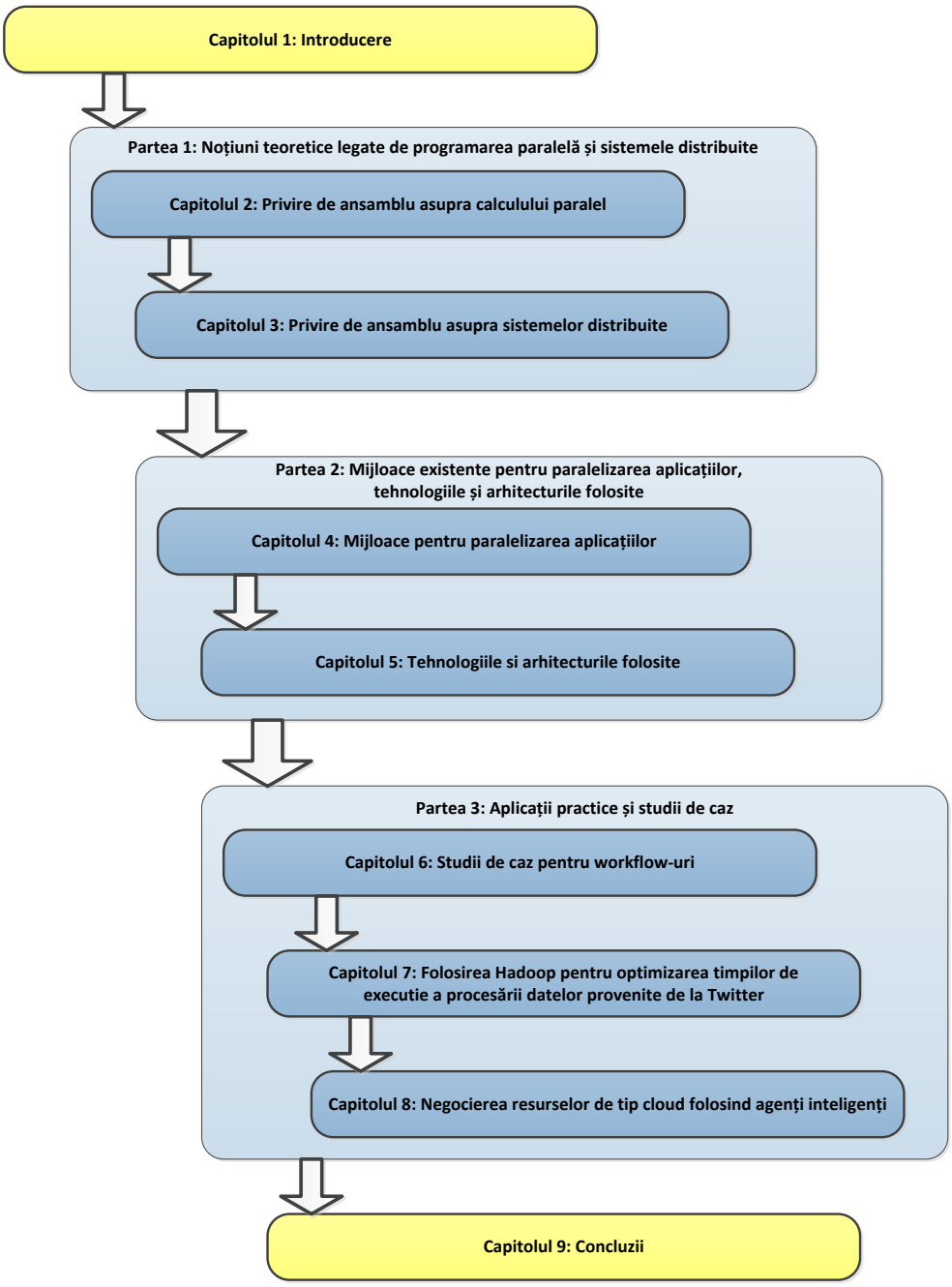

Figura 1: Structura tezei.

Partea I - Noțiuni teoretice legate de programarea paralelă și sistemele distribuite – este compusă din Capitolul 2 și Capirolul 3 și are rolul de a oferi o viziune de ansamblu asupra calculului paralel și a sistemelor distribuite, precum și de a motiva necesitatea utilizării acestora atunci când trebuie îmbunătățită perfomanța aplicațiilor.

Capitolul 2: Privire de ansamblu asupra calculului paralel – oferă detalii legate de apariția calculuilui paralel, motivează necesitatea apariția acetuia, prezintă principalel nivele de paralelism existente în cadrul unei aplicații, enumeră principalele tipuri de paralelism la nivel hardware ce au fost create be baza arhitecturii Von Neumann. Acest capitol oferă de asemenea și o perspectivă asupra mijloacelor folosite pentru evaluarea performanței paralelizării unei aplicații (speedup, eficiență, legea lui Amdahl).

În [28] sunt prezentate pricipalel motive ce au dus la apariția calculului paralel. Autorul apreciază faptul că apariția calculi parallel s-a datorat în ptincipal nevoii continue de a crește performanța aplicațiiolor. Avansarea cercetărilor în domenii precum: modelarea climaterică, descoperirea de noi medicamente, cercetări energetice, nevoia unor căutări web cît mai rapide și redarea imaginilor la o calitate cît mai înaltă a dus la necesitatea creșterii performanței aplicațiilor. Acest lucru putea fi realizat doar prin folosirea sistemelor paralele și a calculului paralel.

Creșterea performanțelor unui singur procesor a constat în creșterea numărului tranzistorilor de pe circuitele integrate. Pentru a putea obține performanțe mai ridicate în cadrul aceleași componente dimensiunea tranzistorilor a fost scăzută iar viteza lor de procesare a crescut. Însă acest lucru a dus la creșterea căldurii generate de aceste componente. Conform [16] în prima decadă a secolului 21 circuitele răcite pe bază de aer s-au apropiat simtitor de limita lor superioară în ceea ce privește capacitatea de disipare a căldurii.

Conform [4] un alt motiv ce a dus la apariția sistemelor paralele este faptul că există un nivel minim al voltajului necesar pentru a rula un microprocesor la frecvența dorită. Acest voltaj este proporțional cu frecvența. Astfel creșterea vitezei de procesare cu o anuimtă cantitate duce la creșterea energiei consumate la o valoare egală cu viteza respectivă la puterea a treia.

Software-urile și hardware-ulrile paralele au evoluat pe baza software-urilor

și hardware-urilor seriale. În 1966 Michael Flynn [13] a propus o clasificare a arhitecturilor de calcul bazată pe numărul de intrucținui și streamuri de date ce puteau fi rulate în mod concurent: SISD (O singură instrucțiune, o singură dată), SIMD (o singură instrucține, mai multe date), MISD (mai multe instrucțiuni, o sigura dată) și MIMD (mai multe instructțiuni, mai multe date).

Atunci când este necesară evaluarea performanței atinse de către o aplicație în urma paralelizării ei există câteva metode ce pot fi de mare ajutor în acest sens. În general pentru a face o evaluare obiectivă a performanței se folosec mai multe astfel de metode simultan. Câteva dintre aceste medote sunt prezentate în [28]. Atunci când dorim să parelelizăm o aplicație pentru a obține timpi de execuție mai scurți trebuie să încercăm să împarțim în mod echitabil munca între core-urile disponibile. Dacă reușim să facem acest lucru și rulăm programul nostru pe p core-uri, rulând un singur fir de execuție pe core, atunci programul nostru ar trebui să ruleze , în teorie, de p ori mai repede decât programul serial. Dacă considerăm T_{serial} ca fiind timpul necesar pentru ca aplicația să ruleze în mod serial și $T_{paralel}$ ca fiind timpul necesar unei execuții paralele, atunci putem spune că aplicația poate atinge un speedup de:

$$S = \frac{T_{serial}}{T_{parallel}}, \tag{1}$$

În anii 60, Amdahl [3] a făcut o observație privind speedup-ul maxim ce poate fi atins de către o aplicație, oservație ce este cunoscută în zilele noatre sub denumirea de legea lui Amdahl. Conform acesteia dacă un program nu poate fi paralelizat în totalitate speedup-ul maxim ce poate fi atins, indiferent de numărul de core-uri folosite, este limitat.

Putem spune despre o anumită tehnoloige că este *scalabilă* dacă poate rula cu perfomațe similare pentru diverse mărimi ale problemei. În cazul programelor paralele scalabilitatea se referă la menținerea aceluiași nivel de eficiența prin varierea numnarului de procese/fire de execuție și a dimensiunii problemei.

Acestea sunt doar câteva din mijloacele prin care poate fi evaluată performanța atinsă prin paralelizarea unei aplicații.

Capitolul 3: Privire de ansamblu asupra sistemelor distribuite– acest capitol oferă o viziune de ansamblu cu privire la diversele tipuri de sistemele

distribuite (cluster, grid, cloud), prezintă avantajele și dezavantajele acestora și insistă asupra tehnologiei cloud computing. Sunt prezentate modelel de business ale cloud computing, avantajele și dezavantajele acestuia, precum și o comparație între principalel tipuri de sisteme distribuite.

În cartea lor despre sitemele distribuite [35] Tanenbaum și Steen definesc un astfel de sistem după cum urmează:

> *Un sistem distribuit este o colecție de calculatoare independente ce-i apar utilizatorului sistemului ca fiind un singur calculator.*

Acest tip de sisteme conțin un număr variabil de calculatoare independente ce cooperează între ele, prin intermediul unei rețele de comunicații, pentru a atinge un obiectiv specific.

În continuare vom prezenta definițiile principalelor tipuri de siteme distribuite:

> *Un cluster este un tip de sistem paralel și distribuit ce consistă dintr-o colecție de calculatoate de sine stătatoate, iterconectate ce lucrează împreună ca și o singură resursă computațională integrată.* după [6] și [29]

> *Un grid este un tip de sistem paralel si distribuit ce face posibilă partajarea, selecționarea și agregarea, în mometul execuției, a unor resurse de calcul autonome distribuite din punct de vedere geografic, în funcție de disponibilitatea acestora, de capacitățile pe care le dețin, de perfomanță, de costuri și de cererile de calitate a serviciilor făcute de către utilizatori.* după [6] și [29]

Una ditre viziunile secolului 21 asupa calculuilui informatic este ca utilizatori să acceseze servicii pe Internet de oriunde pe baza unui model de taxare de tipul plătești doar atât cât consumi. În [7] cloud computing este văzut ca și cea de-a cincia utilitate (după apă, electricitate, gaz și telefonie). Acest nou tip de utiliate ar trebui să le ofere utilizatorilor un nivel de bază al serviciilor de calcul, nivel suficient pentru a răspunde nevoilor zilnice ale comunității generale. Un număr de paradigme computaționale au fost propuse în acest sens, cea mai nou fiind cloud computing. Câteva din caracteristicile de bază a cloud computing sunt:

- plătești doar atât cît consumi
- elasticitate resurselor, fatul că acestea îi apar utilizatorului ca fiind infinite
- resursele sunt abstractizate sau virtualizate
- resursele sunt disponibile la cerere pe bază de self-service

Majoritatea lucrărilor în domeniu [22], [37], [1], [25], [8] vorbesc despre principale tipuri de modele business ale cloud-ului:

- **Infrastructure-as-a-Service** – acest model pune la dispoziția utilizatoriol o infrastructură virtualizată de resurse de calcul sau de stocare a datelor.
- **Platform-as-a-Service** – sistemele de tip cloud pot furniza și platforme de tip software pe care anumite sisteme să ruleze.
- **Software-as-a-Service** – de asemenea sistemele cloud pot oferi și servicii ce sunt de interes pentru o varietate mare de utilizatori. Aceasta este o alternativă la rularea locală a aplicațiilor.

Partea II - Mijloace existente pentru paralelizarea aplicațiilor, tehnologiile și arhitecturile folosite – acestă parte este compusă din Capitolul 4 și Capitolul 5 și descrie principalel metode de paralelizare a aplicațiilor existente pe piața (fluxurile de lucru și paradigma MapReduce). De asemenea oferă o vedere de ansamblu asupra principalelor tehnologii și platforme ce au fost întebuințate în partea de aplicații practice: platforma Askalon folosită pentru dezvoltarea și rularea aplicațților de tip cloud și grid; Hadoop, o implementare în Java a paradigmei MapReduce ce le permite utiliatorilor să proceseze cantități mari de date în paralel; Eucalyptus, o platformă pentru construirea de sisteme de tip cloud ce le oferă utilizatorilor posibilitatea creării proprilor lor infrastructuri pe baza resurselor de calcul pe care le dețin deja.

Capitolul 4: Available Means for Parallelizing Applications – descrie pricipalel tehnologii ce pot fi folosite de către cercetătorii ce nu dispun de aptitudini avansate de programare paralelă pentru ași paraleliza aplicațiile cu scopul de a le reduce timpi de execuție și totodată și costurile.

Conform [19] numărul articolelor științifice legate de mutiprocesoare a crescut în mod constant din 2001 încoace. În [34] se apreciază că următoarea provocare în acest domeniu va fi concurența. În zilele noaste produsele software sunt influențate de dorința utilizatorilor de a crea sisteme din ce în ce mai ample, care să soluționeze probleme tot mai complexe și care să profite de capacitățile tot mai crescute ale resurselor de calcul și de stocare a datelor.

O altă provocare cu care actualele și viitoarele software-uri vor trebui să se confrunte este volumul tot mai crescut de date ce trebuie procesate, volum a cărui dimensiune crește mai repede decît puterea de calcul. În [18] se estimează că datele ce vor fi generate de viitoarele experimente for fi semnificativ mai mari ca și volum decît datele colectate pînă în acest moment de-a lugul istoriei umane. Procesarea acestor date va necesita folosirea unei puteri de calcul și rețele de comunicații mult mai mari decât cele disponibile până cu puțini ani în urmă.

Una dintre metodele existente pentru paralelizarea execuției aplicațiilor pe resurse de calcul distribuite sunt fluxurile de lucru. În acestă secțiune vom elabora asupa noțuni de workflowflux de lucru și vom analiza modul în care acestă paradigmă poate influemța execuția aplicașiilor științifice sau de business.

> *Un flux de lucru științific este un proces de combinare a datelor și proceselor într-un set de pași structurați și copnfigurabili ce implementează soluții computaționale semi-automate pentru o problemă științifică.* [2]

> Un Sistem de Management a Fluxurilor de Lucru este definit ca fiind: *Un sistem ce definește în mod complet, gestionează și execută fluxuri de lucru prin intermediul unor software-uri a căror ordine de execuție este determinată de o reprezentare computeriză a logicii unui workflowflux de lucru.* [20]

În anii 90, când fluxurile de lucru științifice au apărut pentru prima oară, acestea erau folosite în principal pentru soluționarea problemelor de virtualizare. Însă de atunci a existat o evoluție semnificativă în ceea ce privește acest tip de tehnologie [36]. Câteva dintre aceste evoluții sunt de interes sporit pentru Sistemele de Management a Fluxurilor de Lucru: platformele orientate pe componente, gridurile de date și cele computaționale, arhitecturile orientate pe servicii și serviciile

web, organizațiile virtuale, rețelel de tip peer-to-peer, virtualizarea și apariția cloud computing.

Astfel fluxurile de lucru științifice au evoluat pentru a putea satisface diverse nevoi ale lumii științifice ducând la schimbarea completă a metodelor științifice folosite. Cu timpul acestea au devenit o prctică uzuală având un impact semnificativ asupra studiilor științifice.

O alta tehnologie folosită pentru paralelizarea aplicațiilor MapReduce [9], care a fost implementată pentru a simplifica procesarea unor seturi de date de mari dimensiuni pe sisteme de calcul distribuite.

Această paradigma este compusă din două parți: map și reduce. Întregul set de date de intrare este divizat in mai multe bucăți ce vor fi procesate în mod individual în etapa de map. Funcția de tip map primește ca și intare o singură instanța de tip cheie/valoare. Outputul acestei etape este o colecție de date sub forma cheie/valaore ce au fost grupate după valoarea cheii, colecție ce va fi folosită ca și intare pentru etapa reduce. Pe baza perechilor de tip cheie/valoare și a unei liste de valori oferite de etapa map, etapa reduce realizează anumite calcule asupra castor perechi și returnează tot o colecție de date de tip cheie/valoare.

Capitolul 5: Tehnologiile si arhitecturile folosite – oferă o viziune de ansablu asupra platformelor ce au fost folosite pentru a testa eficiența paralelizării aplicațiilor. Este prezentată platforma Askalon ce a fost folosită pentru construirea fluxurilor de lucru din partea practică a tezei. Pentru procesarea unor cantități mari de date a fost aleasă platforma Hadoop. Principalel resurse de calcul ce au fost folosite în cadrul experimentelor rulate sunt cele de tip cloud privat și au fost create folosind platforma Eucalyptus.

Askalon [11] este o platformă bazată pe grid ce permite implementarea și rularea de aplicații de tip grid/cloud. Acestă platformă a fost extinsă pentru a putea rula aplicații și pe resurse de tip cloud. În Askalon utilizatorul compune un flux de lucru pe baza unei aplicații folosind un nivel înalt de abstractizare folosindbazat un limbaj de tip XML, (AGWL). Astfel este ascuns față de utilizator nivelul de resurse de tip grid/cloud. Implementarea fluxului de lucru de către utilizator se face, în mare parte, în mod vizual pe baza unei interfețe grafice. Repezentarea AGWL a fluxului de lucru este folosită de către serviciile de tip middleware pentru planificarea execuției aplicației pe resurse distribuite.

În 2004 Doug Cutting [27], un bine cunoscut dezvoltator de software-uri de tip open source, s-a decis să implementeze algoritmul MapReduce creat anterior de către Google. Acesta a denumit noul software Hadoop după numele unui elefant de pluș pe care îl avea fiul său. Principalul scop al acestui software era să proceseze (să copieze, să indexeze, etc.) cantități mari de date. Hadoop are două componente principale: HDFS (sitemul distribuit de fișiere al Hadoop) și o implementare a algoritmului MapReduce.

HDFS furnizează un mediu de stocare a datelor scalabil, cu toleranță ridicată la erori la un cost redus. Pe lângă stocarea fișierelor, HDFS este capabil să detecteze și să compenseze eventualele erori de hardware apărute. HDFS stochează fișierele în cadrul unei colecții de servele dintr-un cluster.

În [26] este prezentat Eucalyptus, o platforma de tip open-source pentru crearea de sisteme de tip clod pe baza infrastructurii de calcul și de stocare a datelor ce se află la îndemâna grupurilor de cercetare din domeniul academic. Eucalyptus este format din mai multe componente ce interacționează între ele prin interfețe bine definite.

Arhitectura acestui sitem este simplă, flexibilă și modulară având un design ierarhic ce reflectă resursele computaționale prezente în majoritatea instituțiilor academice. În esență, sistemul le permite utlizatorilor să pornească, să controleze, să acceseze și să termine execuția unor mașini virtuale. Toate acestea se realizează prin intermediul unor interfețe ce emulează interfețele oferite de Amazon EC2.

Fiecare componentă de bază din sistem este implementată sub forma unui serviciu web de sine stătător. Acest lucru oferă următoarele avantaje: în primul rând fiecare serviciu expune o API prin intermediul unui fisier WSDL, interfață ce este agnostică din punct de vedere al limbajului și care conține ambele operații pe care serviciul le poate executa asupra datelor de intrare/ieșire; în al doilea rînd sunt facilitate fucnționalitățiile legate de securitatea serviciilor web, trăsătură ce asigură o comunicare sigură între componente.

Partea III - Aplicații practice și studii de caz – este compusă din Capitolul 6, Capitolul 7 și Capitolul 8. Acestă parte conține aplicațiile practice relizate pentru a evalua eficiența privind timpi de execuție și costurile generate de diverse aplicații, precum și eficiența oferită de diverse tipuri de siteme distribuite

și metode de paralelizare a aplicațiilor.

Capitolul 6: Studii de caz pentru fluxuri de lucru – conține descrierea a două fluxuri de lucru diferite ce au fost implementate pe baza a două aplicații reale: fluxul de lucru eBay ce a fost implemetat pe baza unui aplicații implementate în python ce are drep scop extragerea informațiilor legate de preferințele utiliztorilor de eBay[1]; fluxul de lucru RainCloud [2] ce a fost implementat pe baza unei aplicații meteorologice menită să furnizeze previziuni legate de cantitatedea de precipitații ce ar putea cădea în regiunea Tyrol, Austria. În tip ce prima aplicație necesită multe scieri și citiri din fișiere, cea de-a doua are nevoie de resurse de calcul substanțiale. Ambele fluxuri de lucru au fost implementate folosind platforma Askalon.

Petru a putea implemeta un flux de lucru pe baza unei aplicații este nevoie ca câțiva pași să fie parcurși în avans [10]:

- identificarea secțiunilor aplicațiilei ce au cele mai puține dependețe între ele. Aceste vor deveni activități individuale în cadrul fluxului de lucru.
- stabilirea secțiunilor de cod ce ar putea fi rulate în paralel. Acestea vor face parte din secțiunea paralelă a fluxului de lucru.
- dacă unele din activitățile definite în pași anteriori are prea multe dependențe față de o altă activitate, atunci acestea ar trebui reunite în cadrul unei singure activități.
- identificarea datelor de intare/ieșire necesare fiecărei activități.
- stabilirea corelațiile existente între activitățile viitorului workflowflux de lucru.

Figura 2 reprezintă fluxul de lucru realizat pe baza aplicației de extragere a datelor de pe eBay:

[1] O parte din rezultatele legate de fuxul de lucru eBay a fost publicată în **Gabriela Andreea Morar**, Cristina Ioana Muntean, Gheorghe Cosmin Silaghi, "Implementing and Running a Workflow Application on Cloud Resources", Economy Informatics, vol. 15, no. 3/2011, pages 15-27

[2] O parte din rezultatele legate de fuxul de lucru RainCloud a fost publicată în **Gabriela Morar**, Felix Schueller, Simon Ostermann, Radu Prodan, and Georg Mayr, "Meteorological Simulations in the Cloud with the ASKALON Environment", CoreGRID/ERCIM Workshop on Grids, Clouds and P2P Computing, Rhodes Island, August, 2012, acceptată

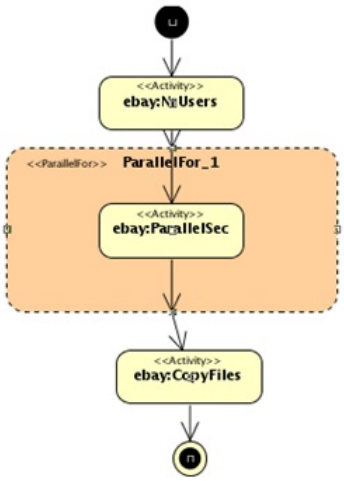

Figura 2: Fluxul de lucru eBay implementat în Askalon.

Pricipalel activități identificate în cadrul acestui workflowflux de lucru sunt: **Nrusers** activitate ce imparte fișierul de intare ce conține toți utilizatori într-un număr de fisiere egal cu numarul de core-uri folosite pentru rularea aplicatiei. Fiecare fișier va conține un numă de utilizatori egal cu numărul total de utilizatori/numarul de core-uri folosite. **ParallelSec** este secțiunea din cadrul aplicației ce poate fi executată în mod paralel. În cadrul acestei secțiuni sunt extrase datele referitoare la preferințele utilizatorilor. Ultima activitate, **CopyFiles**, are scopul de a aduna datele generate de către secțiunea paralelă a fluxului de lucru.

Am rulat experimente pentru acest workwlow pe diverse tipuri de instanțe cloud și folosind fișere de întrare cu 6000 și respectiv 12000 de utilizatori. Rezultatele arată că aplicația poate fi rulată pe resurse cloud cu un speedup de până la 40, speedup calculat pe baza timpului de execusie obținut pe baza rulării în mod serial a aplicației.

Fluxul de lucru RainCloud este baza pe o aplicație menită să simuleze nivelul precipitașiilor în regiuni montane cu ajutorul unui model meteorologic simplu, numit modelul liniar al precipitațiilor orografice (LM) [5]. Aplicațiile acestui model variază de la studii climatice la aspecte hidrologice. Figura 3 înfățisează reprezentarea grafică a wfluxului de lucru RainCloud împreună cu fișierele de intrare/ieșire necesare fiecărei activități:

Pe baza rezultatelor dorite și a scopului rulării woprkflow-ului au fost create

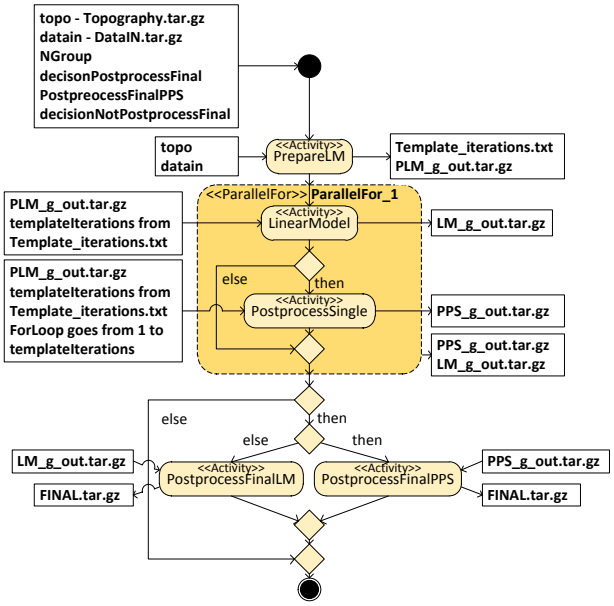

Figura 3: Reprezentarea grafică a fluxului de lucru RainCloud în Askalon.

mai multe variante de execuție ale acestuia, variante intitulate arome ale fluxului de lucru:

- *aroma ideală a fluxului de lucru* aparține ariei operaționale și folosește versiuni idealizate de topografii și date legate de condițiile atmosferice; este folosită în special pentru testarea modelului.
- *aroma semi-ideală a fluxului de lucru* aparține ariei de cercetare și folosește fie topografii, fie date atmosferice idealizate; este folosită pentru interpretarea măsurătorilor meteorologice.
- *aroma reală a fluxului de lucru* aparăține ariei de cercetare, iar atât topgrafiile cît și datele metorologice folosite au caracter real; este folosită pentru previzionarea precipitațiilor.

Am rulat experimete pentru acest flux de lucru folosind resurse provenite de la un cloud privat cât și resurse provenite de la un cloud public (Amazon EC2). Experimentele au arătat că execuția fluxului de lucru folosind date experimentale ar costa 2.72$ dacă s-ar folosi instanțe 4 `c1.xlarge` provenite de la Amazon EC2

la un preș de (0.68$/oră). Astfel o execuție zilnică timp de un an a apalicației ar avea un cost de aproximativ 992.8$.

În urma experimentelor efectuate am constatat că performațele obținute pe instante te tip cloud privat sunt similare cu cele obținute prin folosirea unui cloud public. Acest fapt ne-a permis să folosim resurse private în partea de testare și optimizare a paralelizării aplicației și doar la final să trecem pe resurse publice, astfel costurile de dezvoltare reducîndu-se semnificativ.

Fluxurile de lucru științifice sunt folosite la scară largă în momentul de față. Aceste îi ajută pe oemenii de știință, și nu numai, să profite de resursele de calcul ditribuite pe care le au la îndemână. Prin folosirea acestora timpi de execuție ai aplicațiilor sunt reduși semnificativ, iar odată cu aceștia scad și costurile.

Capitolul 7: Folosirea Hadoop pentru optimizarea timpilor de executie a procesării datelor provenite de la Twitter – acest capitol prezintă modul în care Hadoop poate fi folosit pentru a procesa mari cantități de date într-un timp mai scurt prin prizma folosirii resurselor distribuite. Pentru partea legată de procesarea datelor a fost folosită librăria Mahout [24] care pune la dispoziția utilizatorilor un număr mare de algoritmi de tip machine learning. Acestă librărie rulează pe clustere de tip Hadoop. Am rulat o serie de experimente menite să testeze influența pe care diversele tipuri de stocare a datelor folosite o au asupra performanței Hadoop. Am încercat de asemenea să variem și valorile anumitor parametri de configurare a Hadoop pentru a vedea cum aceștia influențează la rândul lor preformanța obținută. Pe baza acestor configurații am rulat algoritmul k-means[1] de clusterizare a datelor avînd diverși parametri pe date obținute de la Twitter.

Hadoop are peste 165 de parametri ce pot fi configurați de către utilizatori. Găsirea configurației optime nu este activitate trivială. Câteva sugestii legate modul în care un utilizator trebuie să își configureze clusterul Hadoop sunt prezentate în [30],[39] și [21].

Pentru a testa scenariile descrise anterior am creat două clustere Hadoop ce au aceleași caracteristici cu excepția mediului folosit pentru stocarea datelor (SSD

[1] O parte din rezultatele prezentate în acest capitol a fost publicată în Cristina Ioana Muntean, **Gabriela Andreea Morar**, and Darie Moldovan, "'Exploring the meaning behind Twitter hashtags through clustering"', în Lecture Notes in Business Information Systems, vol. 127, pag. 231 - 242. Springer-Verlag Berlin, 2012

sau hard disk). Clusterele au fost implementate folosind mașini virtuale de tip Oracle VirtualBox 4.1.18 [38]. Ambele clustere au un nod principal (master node) și trei noduri secundare (slave nodes), iar stuctura lor poate fi văzută în figura 4.

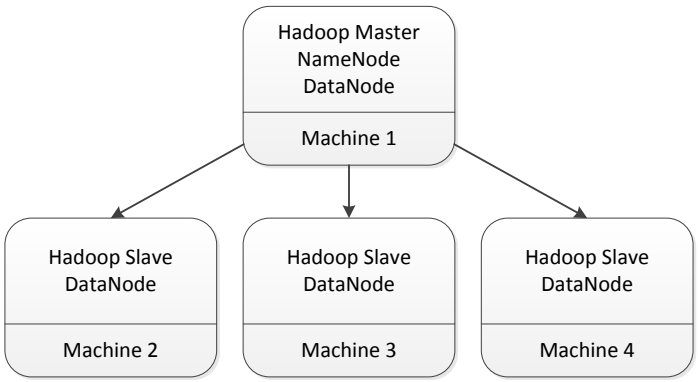

Figura 4: Arhitectura clusterului Hadoop.

După crearea clusterelor Hadoop, performața lor poate fi testată cu ajutorul unor librări de teste puse la dispoziția utiliatorilor de către Hadoop ($hadoop-*-test.jar$). Acestea îl ajută pe utilizator să estimeze dacă a efectuat o instalere reușită a Hadoop. O scurtă introducere despre folosirea acestor teste de performanță este prezenată în [39]. Este recomadată testarea performanței clusterelor cu date de intrare asemănatoare cu datele reale pe utilizatorii le vor rula ulterior.

Pentru experimentele noatre am folit seturi de date colecate cu ajutorul Twitter Streaming API timp de o săptămână. Seturile de date obținute reprezintă 10% din întreaga activitate ce are lor zilnic pe Twitter.

Algoritmul a cărui performanță a fost tetstată pentru acest set de date a fost k-means. K-means [23] este un algoritm de clusterizare a datelor în mod nesupravegheat. Pe baza unui set de date initial algoritmul împarte aceste date într-un număr de clustere stabilit anterior.

Experimentele rulate ne-au arătat ca performanța obținută de clusterele Hadoop este puternic inflențată de tipul de sistem folosit pentru stocarea datelor în cadrul cluster-ului cât și de către valorile parametrilor de configurare setați de către utilizatori.

Deoarece Hadoop a fost creat astfel încât să poată rula si pe resurse de tip cloud, putem spune că acesta reprezintă o soluție viabilă pentru urilizatorii doresc să proceseze cantități mari de date în mod ocazional și cu costuri reduse.

Capitolul 8: Negocierea resurselor de tip cloud folosind agenți inteligenți – în acest capitol este prezentat un domeniu de negociere (CloudDomain)[1] ce a fost implementat pentru a facilita negocierea resurselor de tip cloud prin intermediul folosiri agenților inteligenți. Acest domeniu a fost implementat pe baza caracteristicilor instanșelor de cloud de tip IaaS și reprezintă un prim pas în ceea ce ne dorim a fi la un momentdat negocierea resurselor de tip cloud la runtime în funcție de necesitățiile utilizatorilor și în contextul existenței mai multor furnizori de astfel de resurse. Am studiat performanța domeniului de negociere implementat prin realizarea unor negocieri între un cumpărător de resurse cloud și 10 furnizori diferiți. Agenți folositți pentru negociere au fost înzestrați cu diverse metode de învățare: învățare bayesiană și învățare Q-learning. Protocoalele de negociere folosite dunt cele prezenate în [31] și [32].

Ținem să menționăm că am implementat un sitem semi-automat de generare a profilelor de negociere. Acesta generează profilele de preferințe ale furnizorilor de resurse de tip cloud pe baza datelor obținute direct de la aceștia. Singura parte din aceste profile ce nu se realizează încă în mod automat este cea referitoare la desemnarea importanței pe care fiecare parametru negociat o are petru furnizorul respectiv. Această problemă ar putea fi soluționată prin implemetarea unui mecanism bazat pe machine learning ce ar avea drep rol genrarea acestor valori pe baza preferințelor exprimate de către acești furnizori de resurese în cadrul rundelor anterioare de negociere.

4 Contribuții

Acestă secțiune cuprinde contribuțiile ce au fost aduse în cadrul fiecărui capitol:

Capitolul 2: Privire de ansamblu asupra calculului paralel - în cadrul

[1] O parte din rezultatele prezentate în acest capitol a fost prezentată în **Gabriela Andreea Morar**, and Andreea Ilea, Alexandru Butoi, Gheorghe Cosmin Silaghi, "Agent-based Cloud Resources Negotiation", in Proceeding of 8th International Conference on Intelligent Computer Communications and Processing, pag. 297-300 30 August - 1 Septembrie, 2012 in Cluj-Napoca, Romania

acestui capitol s-a realizat un amplu studiu al literaturii de specialitate legat de calculul paralel: modul în care acesta a apărut, motivele ce au dus la apariția acestuia și sunt descrise câteva dintre principalele metode folosite în evaluarea performanței programelor paralele: speedup, eficiența, măsurarea corectă a timpilor de execuție, etc.

Capitolul 3: Privire de ansamblu asupra sistemelor distribuite - cuprinde un amplu studiu al literaturii din domeniul sistemelor distribuite. Sunt evaluate pe rând principalele tipuri de astfel de sisteme: clustere, grid-uri, cloud computing și este prezentată o comparație între acestea realizată pe baza principalelor lucrări existente în domeniu. De asemenea conține si un studiu detaliat al cloud computing: modul în care a luat naștere, modelele de business folosite de acesta, avantajele și dezavantajele lui, etc.

Capitolul 4: Mijloace existente pentru paralelizarea aplicațiilor - constă întru-n studiul al literaturii de specialitate legate de principalele metode existente pe piață ce le permit oamenilor de știință sau oemanilor de business să își paralelizeze aplicațiile pentru a obține timpi de execuție mai scurți și pentru a reduce costurile. În acest sens ne-am îndreptat atenția asupa paradigmelor fluxurilor de lucru și MapReduce. Prima dintre ele se adresează în special aplicațiilor ce necesită putere de calcul crescută, în timp ce MapReduce se adresează aplicațiilor ce sunt menite să proceseze cantități mari de date. Am studiat modul de funcționare a acestor tehnologii pentru a putea ulterior să le aplicăm pentru a reduce timpi de execuție a anumitor aplicații.

Capitolul 5: Tehnologiile si arhitecturile folosite - acest capitol descrie tehnologiile și platformele ce au fost folosite pentru a putea evalua în mod practic tehnicile de paralelizare a aplicațiilor prezentate anterior. De asemenea prezentatăm și avantajele și dezavantajele acestor platforme și evaluăm complexitatea utilizării lor.

Capitolul 6: Studii de caz pentru fluxurile de lucru - în acest capitol sunt prezentate cele două fluxuri de lucru ce au fost implementate pe baza unor aplicații reale: fluxul de lucru eBay este bazat pe un data crawler și care necesită multe citiri și scieri in/din fisiere, iar fluxul de lucru RainCloud este bazat pe o aplicație metorologică reală (folosită de institutul de avalanșe din regiune Tyrol, Austria pentru previzionare a nivelui de precipitații) ce necesită o putere de calcul

ridicată. Am implementat ambele fluxuri de lucru folosind platforma Askalon și am modificat aplicațiile astfel încât să le îmbunătățesc performanța și să le scad costurile necesare rulării. Pentru a putea implementa fluxurile de lucru a fost necesară identificarea secțiunilor din aplicații ce puteau fii parlelizate și a celor ce puteau fi rulate doar secvențial, precum și identificarea dependețele ce existau înte aceste secțiunui. Reducerea timpilor de execuție a fost de mare importanță în cazul celei de-a doua aplicații deoarece aceasta urma să fie rulată în fiecare zi la aceeași oră pentru cel puțin un an pentru a oferi previzionări legate de nivelul de precipitații dintr-o anumită regiune.

Capitolul 7: Folosirea Hadoop pentru optimizarea timpilor de execuție a procesării datelor provenite de la Twitter - acest capitol prezintă un studiu referitor la modul în care Hadoop poate fi folosit pentru rularea aplicațiilor ce trebuie să proceseze cantități însemnate de date. Este prezentat modul în care se pot crea și configura clustere pe care Hadoop să ruleze pentru a crește performanța aplicațiilor. Faptul că datele sunt procesate în paralel pe mai multe resurse de calcul duce atât la scăderea timpului de execuție cît și la scădera costurilor (Hadoop poate rula și pe resurse de tip cloud). Am realizat câteva teste de performanță folosind diverse tipuri de clustere Hadoop și având diverși parametri de configurare. Rezultatele obținute au arătat că atât tipul de mediu de stocare folosit (SSD sau hard disk) cât și modificarea diverșilor parametri de configurare pot duce ca creșterea semnificativă a performanței. Datele folosite pentru testele de performanță au fost extrase de pe Twitter.

Capitolul 8: Negocierea resurselor de tip cloud folosind agenți inteligenți - în acest capitol este prezentat un domeniu de negociere (CloudDomain) pe care l-am creat pentru a putea fi folosit de către agenți pentru a negocia resurse de tip cloud în contextul în care există mai mulți furnizori de astfel de resurse pe piață. CloudDomain a fost astfel construit încât să conțină pricipalel atribute pe care un utilizator si-ar dori să le negocieze cu un furnizor de resurse cloud de tip Iaas: număr de core-uri, memorie, numărul de instanțe disponibile dintr-un anumit tip, preț și dimensiunea hard disk-ului). Acest profil a fost impelmentat astfel încât să poată fi folosit de către platforma Genius [12]. Am creat un mecanism ce generează în mod semi-automat aceste profile pe baza datelor obținute de la cloud controller-ele cloud-urilor private. Aceste profile vor au fost folosite ca și

domenii de preferințe pentru agenți ce dispuneau de diverse tipuri de inteligență (bayesiană sau Q-learning) pentru a negocia resurse de tip cloud pe baza unui protocol de negociere de tipul unul-la-mai-mulți.

5 Stagiu de mobilitate

Pe durata studiilor doctorale am avut oportunitatea de a merge într-un stagiu de mobilitate înafara țării. Timp de opt luni am fost membru al Grupului de Sisteme Distribuite și Paralele, al Institutului de Informatică, Universitatea din Innsbruck, Austria sub coordonarea Prof. Radu Prodan. Pe durata șederii mele în Austria am colaborat cu coordonatorul de acolo la unul dintre proiectele acestuia: proiectul RainCloud. Acest proiect este o colaborare între Grupul de Sisteme Distribuite și Paralele, Institutul de Meteorologie și Geografie, al Universității din Innsbruck și serviciul Tyrolez de avalanșe("Tiroler Lawinenwarndienst" (LWD)). Scopul final al acestui proiect era acela de a-i furniza LED-ului o aplicație meteorologică care va rula în fiecare zi la aceeași oră pentru a realiza prognoze realistice în legătură cu cantitatea de precipitații ce ar putea cădea într-o anumită regiune.

Proiectul a constat în două mari parți:

- **Prima parte** – implementarea unei aplicații meteorologice care pe baza unei anumite topologii și a unor date meteorologice să genereze previziuni meteolologice. Această parte s-a aflat în grija Institutului de Meteorologie și Geografie, al Universității din Innsbruck.
- **A doua parte** – crearea unui fluxului de lucru folosind Askalon o platforma de dezvoltare a aplicațiilor te tip grid/cloud. Scopul acestuia era să optimizeze timpi de execuție ai aplicației prin paralelizarea anumitor secțiuni ale acesteia și găsirea unei modalități de rulare a aplicației în cele mi bune condiții de performanță și cu costuri minime. Acesta parte a fost responsabilitatea Grupului de Sisteme Distribuite și Paralele, al Institutului de Informatică, Universitatea din Innsbruck.

Ca și membră temporară a Grupului de Sisteme Distribuite și Paralele a trebuit sa implementez fluxul de lucru pe baza aplicației meteorologice dezvoltate de către ceilalți parteneri din cadrul proicetului. Îndatoririle mele au constat în:

- să colaborez cu persoana ce a realizat aplicația meteorologică și să identific diversele secțiuni ale aplicației ce puteu fi mapate ulterior în activități secvențiale sau paralele în cadrul fluxului de lucru.
- să identific toate dependențele existente între activitățiile găsite în pasul anterior pentru a le putea defini în cadrul fluxului de lucru.
- să colaborez cu persoana care a implementat aplicația pentru a putea face modificările necesare pentru a îmbunătății performanța aplicației.
- să implementez fluxul de lucru folosind Askalon.
- să testez performanța fluxului de lucru pe resurse de tip cloud; prima dată pe resurse din cadrul unui cloud privat pe care grupul de cercetare îl avea la dispoziție, iar apoi testarea pe un cloud public astfel încât să putem estima costurile necesare rulării zilnice a aplicației pentru un an întreg.

Drep rezultat al stagiului meu de mobilitate în Inssbruck am reuțit să public un articol pe baza fluxului de lucru implemetat cu echipa de acolo: **Gabriela Morar**, Felix Schueller, Simon Ostermann, Radu Prodan, and Georg Mayr, "Meteorological Simulations in the Cloud with the ASKALON Environment", CoreGRID/ERCIM Workshop on Grids, Clouds and P2P Computing, Rhodes Island, August, 2012, acceptat.

Stagiul de mobilitate mi-a permis sa deprind o experiență vastă în ceea ce privește sistemele distribuite, în mod special grid și cloud. De asemenea am învațat modul în care funcționează fluxurile de lucru, cum să implemetez un flux de lucru, cum să modific o aplicație astfel încât să îi cresc performanța la execuție, etc. Întreaga experiență teoretică și practică acumulată de-a lungul ecestui stagiu de mobilitate a avut un impact semnificativ asupra cercetării mele.

6 Diseminarea rezultatelor

Rezultatele obținute pe perioada studiilor mele doctorale au fost publica în cadrul unor articole prezentate la conferințte și workshopuri din tară cît și din străinătate.

- **Gabriela Andreea Morar**, Cristina Ioana Muntean, Gheorghe Cosmin Silaghi, "Implementing and Running a Workflow Application on Cloud Resources", Economy Informatics, vol. 15, no. 3/2011, pages 15-27

- Cristina Ioana Muntean, **Gabriela Andreea Morar**, and Darie Moldovan, "'Exploring the meaning behind Twitter hashtags through clustering"', în Lecture Notes in Business Information Systems, vol. 127, pag. 231 - 242. Springer-Verlag Berlin, 2012.

- Alexandru Butoi, **Gabriela Andreea Morar**, and Andreea Ilea, "Two-Phased Protocol for Providing Data Confidentiality in Cloud Storage Environments", în Lecture Notes in Business Information Systems, vol. 127, pag. 220 - 230. Springer-Verlag Berlin, 2012.

- **Gabriela Andreea Morar**, and Andreea Ilea, Alexandru Butoi, Gheorghe Cosmin Silaghi, "Agent-based Cloud Resources Negotiation", in Proceeding of 8th International Conference on Intelligent Computer Communications and Processing, pag. 297-300 30 August - 1 Septembrie, 2012 in Cluj-Napoca, Romania

- **Gabriela Morar**, Felix Schueller, Simon Ostermann, Radu Prodan, and Georg Mayr, "Meteorological Simulations in the Cloud with the ASKALON Environment", CoreGRID/ERCIM Workshop on Grids, Clouds and P2P Computing, Rhodes Island, August, 2012, acceptată

- Alexandru Butoi, **Gabriela Andreea Morar**, Andreea Ilea, "Agent-Based Framework for Implementing and Deploying of SOA", Journal of Mobile, Vol 4, No 2 (2012), Embedded and Distributed Systems (JMEDS) ISSN: 2067 - 4074, pag. 107-113

- Alexandru Butoi, Andreea Ilea and **Gabriela Andreea Morar**, "Conceptual Design for Business SOA using Object-Oriented Paradigm", în Proceedings of "The Eleventh International Conference on Informatics in Economy IE 2012", București, Romania, pag. 41-45

- **Gabriela Andreea Morar**, Cristina Ioana Muntean and Nicolae Tomai , "An Adaptive M-learning Architecture for Building and Delivering Content

based on Learning Objects", The Second Romanian Workshop on Mobile Business, Cluj-Napoca, România, publicat în Economy Informatics, vol. 10, no. 1/2010, pag. 63-73

7 Direcții ulterioare de cercetare

Cercetare realizată pe parcursul studiilor doctorale mi-a oferit un nivel ridicat de cunoștere în domeniul calculului paralel, a sitemelor distribuite (în special cloud computing)și a mijloacelor de de negociere a resurselor folosind agenți inteligenți. Experiența acumulată până în acest moment și tendințele existente în domeniul cercetării m-au ajutat să identific câteva posibile ulterioare direcții de cercetare:

- implementarea unui modul pentru platforma Askalon, modul ce va avea rolul de a negocia resurse de tip cloud în momentul execuției fluxurilor de lucru. Negocierea se va face ținând cont de cerințele QoS ale utilizatorilor precum și de estimările referitosare la necesarul de resurse al următoarei activități din fluxul de lucru. În prima fază acest mecanism de negociere ar putea fi folosit pentru negocierea resurselor provenite de la cloud-uri private ce țin de diverse instituții academice.

- implementare unui planificator (scheduler) ce va avea drep scop planificarea viitoarelor task-uri pe baza unor estimări a consumului de energie generat de către resursele disponibile. Reducerea consumului de enerie este una din principalele direcții de cercetare actuale după reducerea costurilor și optimizarea timpilor de execuție ai aplicațiilor.

- implementare unui modul pentru platforma Genius, modul ce va avea drep scop generarea automată a unor profile de negociere pe baza unor caracteristici expuse de către furnizorii de resurse de tip cloud. Aceste profile vor fi utilizate de catre niște agenți de tip broker ce vor avea rolul de a negocia resurse cu utilizatorii finali. Acești agenți vor acționa atât din partea furnizorilor de resurse cloud cât și din partea cumparatorilor acestor resurse.

- implementare unui modul ce va instala în mod automat Hadoop pe nodurile nou adăugate la un cluster și le va configura pe acestea în conformitate cu nodul principal (master node). Implementarea unui modul care să facă corespondența dintre parametri de configurare selectați de către utilizator si clusterul pe care este instalat Hadoop. Acest modul va avea rolul de a le sugera utilizatorilor configurația cea mai potrivită pentru clusterul lor. Acesta este o problema de optimizare a mai multor obiective simultan, deoarece Hadoop dispune de un număr mare de parametrii ce pot fi configurați de către utiliatori, iar alegera valorilor potrivite pentru aceștia nu este deloc intuitivă pentru noii utilizatori și nici chiar pentru cei experimentați.

Acestea sunt doar câteva din principalele direcții ulterioare de cerectare ce ar putea fi urmate be baza cercetării realizate până în acest moment. Toate se adresează unor arii de cercetare ce sunt de mare actualitate și interes pentru mediul academic.

Listă de figuri

Bibliografie

[1] Technical report. 12

[2] I. Altintas, O. Barney, Z. Cheng, T. Critchlow, B. Ludaescher, S. Parker, A. Shoshani, and M. Vouk. Accelerating the scientific exploration process with scientific workflows. In *Journal of Physics: Conference Series*, volume 46, page 468. IOP Publishing, 2006. 13

[3] Gene M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference*, AFIPS '67 (Spring), pages 483–485, New York, NY, USA, 1967. ACM. doi: 10.1145/1465482.1465560. URL `http://doi.acm.org/10.1145/1465482.1465560`. 10

[4] Nikhil Bansal. Dynamic speed scaling to manage energy and temperature. In *In IEEE Syposium on Foundations of Computer Science*, pages 520–529, 2004. 9

[5] Idar Barstad and Felix Schüller. An Extension of Smith's Linear Theory of Orographic Precipitation: Introduction of Vertical Layers. *Journal of the Atmospheric Sciences*, 68(11):2695–2709, November 2011. ISSN 0022-4928. doi: 10.1175/JAS-D-10-05016.1. URL `http://journals.ametsoc.org/doi/abs/10.1175/JAS-D-10-05016.1`. 17

[6] Rajkumar Buyya. *High Performance Cluster Computing: Architectures and Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999. ISBN 0130137847. 11

[7] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Comp. Syst.*, 25(6):599–616, 2009. 11

[8] Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski. *Cloud Computing Principles and Paradigms.* Wiley Publishing, 2011. ISBN 9780470887998. 12

[9] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008. 1, 14

[10] T. Fahringer and Askalon Team. Askalon grid environment, 2007. 16

[11] T. Fahringer, A. Jugravu, S. Pllana, R. Prodan, C. Seragiotto Jr, and H. L. Truong. Askalon: a tool set for cluster and grid computing. *Concurrency and Computation: Practice and Experience*, 17(2-4):143–169, 2005. 14

[12] Cristian Figueroa, Nicolas Figueroa, Alejandro Jofre, Akhil Sahai, Yuan Chen, and Subu Iyer. A Game Theoretic Framework for SLA Negotiation. Technical report, Enterprise Systems Storage Laboratory, HP Laboratories, 2008. http://www.hpl.hp.com/techreports/2008/HPL-2008-5.pdf, consulted on 5 August 2011. 23

[13] Michael J. Flynn and Kevin W. Rudd. Parallel architectures. *ACM Comput. Surv.*, 28(1):67–70, March 1996. ISSN 0360-0300. doi: 10.1145/234313.234345. URL `http://doi.acm.org/10.1145/234313.234345`. 10

[14] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15(3):200–222, 2001. ISSN 1094-3420. doi: http://dx.doi.org/10.1177/109434200101500302. 5

[15] Hadoop. Hadoop website. `http://hadoop.apache.org`, 2012. 2

[16] John L. Hennessy and David A. Patterson. *Computer Architecture, Fourth Edition: A Quantitative Approach.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006. ISBN 0123704901. 9

[17] A. J. G. Hey, S. Tansley, and K. M. Tolle. *The fourth paradigm: data-intensive scientific discovery*. Microsoft Research Redmond, WA, 2009. 4

[18] T. Hey and A. Trefethen. The data deluge: An e-science perspective. In *Grid computing*, pages 809–824. Wiley Online Library, 2003. 13

[19] M.D. Hill and M.R. Marty. Amdahl's law in the multicore era. *Computer*, 41(7):33–38, 2008. 13

[20] D. Hollingsworth. Workflow management coalition: The workflow reference model. Technical report, The Workflow Management Coalition, 1995. 13

[21] S.B. Joshi. Apache hadoop performance-tuning methodologies and best practices. In *Proceedings of the third joint WOSP/SIPEW international conference on Performance Engineering*, pages 241–242. ACM, 2012. 19

[22] Tobias Kurze, Markus Klems, David Bermbach, Alexander Lenk, Stefan Tai, and Marcel Kunze. Cloud Federation. In *Proceedings of the 2nd International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2011)*. IARIA, September 2011. 12

[23] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967. 20

[24] Apache Mahout. Mahout website. `http://mahout.apache.org`, 2012. 19

[25] Dan C. Marinescu. Cloud computing: Theory and practice, 2012. URL `http://www.cs.ucf.edu/~dcm/LectureNotes.pdf`. 12

[26] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The eucalyptus open-source cloud-computing system. In *Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on*, pages 124–131. IEEE, 2009. 15

[27] M. Olson. Hadoop: Scalable, flexible data storage and analysis. *IQT Quarterly*, pages 14–18, 2010. 15

[28] Peter Pacheco. *An Introduction to Parallel Programming.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2011. ISBN 9780123742605. 9, 10

[29] Gregory F. Pfister. *In search of clusters (2nd ed.).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998. ISBN 0-13-899709-8. 11

[30] B. T. Rao, N. V. Sridevi, V. K. Reddy, and L. S. S. Reddy. Performance issues of heterogeneous hadoop clusters in cloud computing. *Global Journal of Computer Science and Technology*, 11(8), 2011. 19

[31] L. Sęrban, C. Ştefanache, G. Silaghi, and C. Litan. A qualitative ascending protocol for multi-issue one-to-many negotiations. *Complex Automated Negotiations: Theories, Models, and Software Competitions*, pages 143–159, 2013. 21

[32] Liviu Dan Serban, Cristina Maria Stefanache, Gheorghe Cosmin Silaghi, and Cristian Marius Litan. A qualitative ascending protocol for multi-issue one-to-many negotiations. In *Proc. of the 2011 Workshop on Agent-based Complex Automated Negotiations*, Studies in Computational Intelligence. Springer, 2012. to appear. 21

[33] Kwang Mong Sim. Towards complex negotiation for cloud economy. In *Advances in Grid and Pervasive Computing*, volume 6104 of *LNCS*, pages 395–406. Springer, 2010. 6

[34] H. Sutter. The free lunch is over: A fundamental turn toward concurrency in software. *Dr. Dobb?s Journal*, 30(3):202–210, 2005. 13

[35] Andrew S. Tanenbaum and Maarten van Steen. *Distributed Systems: Principles and Paradigms (2nd Edition).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. ISBN 0132392275. 11

[36] I.J. Taylor. *Workflows for e-science: scientific workflows for grids.* Springer-Verlag New York Inc, 2007. 13

[37] Luis M. Vaquero, Luis Rodero-merino, Juan Caceres, and Maik Lindner. A break in the clouds: Towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, pages 50–55, 2009. URL `http://dx.doi.org/10.1145/1496091.1496100`. 12

[38] Virtualbox. Virtualbox website. `https://www.virtualbox.org/`, 2012. 20

[39] T. White. *Hadoop: The definitive guide*. Yahoo Press, 2010. 19, 20