UNIVERSITATEA POLITEHNICA DIN BUCUREŞTI

Facultatea de Automatică şi Calculatoare

Computer Science & Engineering Department

TEZĂ DE DOCTORAT

REZUMAT

Servicii distribuite

Alocarea dinamică a resurselor de reţea pentru transferuri de date de mare viteză folosind servicii distribuite

***

Distributed Services

Dynamic network resources allocation for high performance transfers using distributed services

Autor
Ing. Ramiro Voicu

Conducător ştiinţific
Prof. Dr. Ing. Nicolae Ţăpuş

- 2013 -

Cuprins

1. Introducere

De la dispozitive electronice precum telefoane mobile și tablete, devenite deja un truism al zilelor noastre, până la calculatoare și super-calculatoare, sateliți și avioane, toate dispun într-o formă sau alta de o unitate de procesare și o memorie atașată acestei unități. Din momentul în care aceste entități incep să transmită și să coreleze informații folosind orice tip de infrastructură de rețea, ele pot fi considerate parte integrantă a unui *Sistem Distribuit.*

Principalul scop al unui Sistem Distribuit este partajarea resurselor cum ar fi: putere de calcul, spațiu de stocare a datelor, capacitate de rețea, aplicații sau documente. Una din problemele actuale ale marilor comunități științifice, ca de exemplu Astronomie și Astrofizică, Fizica Energiilor Înalte (High Energy Physics – HEP), Bioinformatică, etc., o reprezintă cantitatea mare de date produse și care necesită să fie transferate între diferite locații pentru a fi stocate.

Principalul obiectiv al acestei teze este adresarea problemelor actuale ale aplicațiilor data-intensive (“Big Data”). Pornind de la cerințele și provocările impuse de această problemă, argumentăm că acest subiect poate fi soluționat dintr-o perspectivă unitară, ce implică trei mari aspecte: aplicații de transfer a datelor, infrastructurile de monitorizare si control și resursele fizice implicate: rețeaua si stocarea. Lucrarea de față prezintă principiile care au stat la baza modelului arhitectural ales, detaliile de implementare ale unei platforme care adreseaza în mod coerent și omogen problema aplicațiilor data-intensive.

Cercetarea prezentată în această lucrare este bazată pe rezulatul unei colaborări de succes între Universitatea Politehnica din București, California Institute of Technology (CALTECH), în Statele Unite ale Americii, și Centrul European pentru Cercetări Nucleare (CERN), in Elveția.

Rezultatele prezentate în cadrul acestei teze au fost foarte apreciate în cadrul comunității științifice, autorul fiind parte din grupurile de cercetare care au obținut **două premii pentru inovație – CENIC în 2006 și 2008**. De asemenea a fost parte a grupului ce a câștigat **“Bandwidth Challenge award in 2009”**. Autorul este membru al Monitoring Committee and Advanced Technologies within the ICFA's Standing Committee on Inter-regional Connectivity (SCIC)[1] și al grupului de lucru “HEPiX IPv6 Working Group[2]”.

1.1. Provocări actuale în domeniul aplicațiilor data-intensive

Începând cu mijlocul anilor 1990 rețele de calculatoare au cunoscut un ritm susținut de dezvoltare, datorită progreselor și a extinderii rețelelor optice de mare viteză. Aceasta a dus la o creștere semnificativă a lărgimii de bandă, fapt ce a constituit catalizatorul aplicațiilor data-intensive (“Big Data”), cum ar fi aplicațiile multimedia, partajarea de fișiere de tip torrent, etc. După cum se poate observa în Figura 1 cantitatea totală de date stocată la sfârșitul anului 2011 la CERN in Cern Advanced STORage manager (CASTOR) [1] depășește pragul de 60 Petabytes. Este

[1] International Committee for Future Accelerators (ICFA); Standing Committee on Inter-Regional Connectivity (SCIC) http://icfa-scic.web.cern.ch/ICFA-SCIC/index.html [Online] [Accesat Ian2012]

[2] The HEPiX IPv6 Working Group: https://w3.hepix.org/ipv6-bis/doku.php?id=ipv6:introduction [Online] [Accesat Ian2012]

necesar ca toate aceste date să poată fi migrate eficient în diferite locații geografice în jurul lumii. Un aspect important al acestei probleme, pe lângă cel legat de sistemul de stocare a datelor, este reprezentat de infrastructura de rețea capabilă să răspundă cerințelor unor transferuri de date de foarte mari dimensiuni. La începutul anilor 2000 apare o nouă paradigmă care urmărește să adreseze această problemă: Grid-urile de Date (***Data Grid***) [2]. Pe lângă serviciile de bază, ce se referă în principal la modul de acces al sistemelor de stocare, paradigma asumă un set de bază de servicii descrise mai jos:

- Rezervarea resurselor și **mecanisme de co-alocare pentru resursele de stocare și pentru cele de rețea** pentru a suporta o performanță garantată și predictibilă a transferurilor de date
- **Măsurători de performanță** și tehnici de estimare a resurselor implicate în operațiile pe grid-urile de date (sisteme de stocare, rețele, calculatoare)
- **Servicii de instrumentare** care permit monitorizarea unitară a transferurilor

Chiar și în cazul mediului actual în care dispunem de rețele de mare viteză, transferurile mari de date (zeci de Petabytes) reprezintă o provocare. Pe lângă rețele de mare viteză și sisteme de stocare performante un aspect important îl reprezintă sistemele și aplicațiile de transfer. Este necesar ca serviciile de transfer să poată augmenta capabilitățile noilor tehnologii de rețea, cum ar fi circuite de rețea cu garantarea lărgimii de bandă. Chiar și în cazul în care aceste capabilități nu sunt prezente, aplicațiile de transfer ar trebui să poată adresa aceste deficiențe.

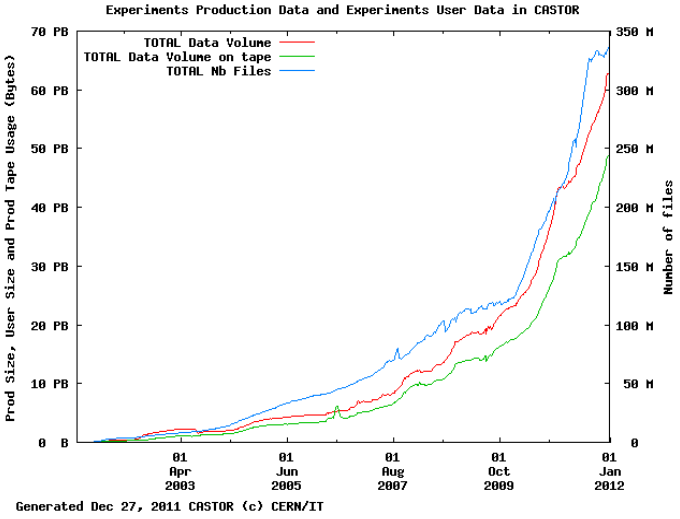


Figura 1 Cantitatea de date stocate la CERN – Decembrie 2011[3]

1.2. Obiectivele tezei

În cadrul acestei teze sunt studiate și adresate aspecte importante din domeniul transferurilor de date de mare viteză. În contextul provocărilor prezentate în subcapitolul anterior 1.1 identificăm **trei obiective majore**:

- O infrastructură de monitorizare și control capabilă să ofere informații despre performanța sistemului într-un mod unitar. Este necesar ca platforma să poată monitoriza toate componentele implicate într-un transfer (sisteme de operare,

[3] Sursă: Statistici Castor, Departamentul CERN IT, Decembrie 2011

aplicații, dispozitive de rețea, sisteme de stocare, etc) și să ofere în același timp mijloacele necesare controlului acestora
- O aplicație de transfer a datelor capabilă să ajusteze în mod dinamic lărgimea de bandă, cu posibilitatea de control de către servicii externe
- Un sistem eficient pentru alocarea resurselor de rețea la Nivelul 1 ISO/OSI capabil să reruteze traficul în caz de probleme

Întreaga lucrare analizează provocările și prezintă soluții dintr-o perspectivă unitară și omogenă.

1.3. Structura tezei

Această lucrare este structurată în opt capitole, după cum urmează:

Capitolul 2 face o trecere în revistă a **sistemelor de alocare a resurselor de rețea, platformelor de monitorizare** și a **aplicațiilor de transfer de date** folosite în ziua de azi. Concluziile de la sfârșitul capitolului pun în evidență calitățile și lacunele tehnologiilor actuale, argumentând necesitatea unei abordări unitare a problemei, în care deciziile sunt luate pe baza monitorizării tuturor componentelor implicate: sisteme de operare, aplicații de transfer și rețele.

În Capitolul 3 se prezintă arhitecura de comunicație, monitorizare și control folosite pentru implementarea sistemului de alocare a căilor optice. Este descrisă arhitectura sistemului subliniindu-se contribuția autorului la proiectul MonALISA.

Capitolul 4 prezintă o nouă aplicație pentru transferul de date proiectată și dezvoltată de autorul acestei lucrări: Fast Data Transfer (FDT). Aplicația expune o interfață utilizator intuitivă și ușor de folosit, furnizând în același timp o perfomanță deosebită prin utilizarea mai multor canale de comunicație TCP în paralel.

Capitolul 5 prezintă un model inovator pentru alocarea de căi în rețele optice. Pronind de la o reprezentare formală a modelului, se **demonstrează** faptul că **într-un multigraf bazat pe comutări optice toate căile posibile sunt disjuncte**. Bazat pe acest rezultat se propune un algoritm de calcul al căii optice. O optimizare importantă a fost adusă folosirea unei **tranzacții distribuite** și a unei **strategii de retragere și reîncercare** (**back-off**) în cazul unui conflict de resurse. În continuarea capitolului sunt prezentate arhitectura și detaliile implementării unui **serviciu distribuit pentru alocarea dinamică de căi optice**. Arhitectura **serviciului distribuit** este o abordare inovatoare în cadrul serviciilor de alocare a resurselor de rețea prin **lipsa unei entități centrale** care să orchestreze toate cererile și prin **paralelizarea alocării resurselor de rețea**.

În Capitolul 6 sunt prezentate rezultatele experimentale și realizări importante bazate pe tehnologiile dezvoltate în timpul acestei cercetări. În prima parte este prezentată **arhitectura distribuită de monitorizare** a rețelei **USLHCNet,** bazată pe platforma **MonALISA**, accentul fiind pus pe acuratețea datelor de monitorizare. Se prezintă apoi rezultate remarcabile ale aplicației **FDT** în cadrul conferințelor de **SuperComputing**. Capitoulul se încheie cu un rezultat experimental care îmbină toate aspectele prezentate în cadrul tezei.

Capitolul 7 prezintă concluziile și contribuțiile originale ale acestei teze, precum și posibile cercetări ulterioare.

2. Concepte și tehnologii în sistemele distribuite pentru mediile data-intensive

Pornind de la cele **trei obiective majore** prezentate anterior, în acest capitol facem o trecere în revistă a sistemelor actuale de alocare a resurselor de rețea, a platformelor de monitorizare distribuite și a aplicațiilor de transferuri de date. În finalul capitolului sunt prezentate concluziile argumentând necesitatea unei platforme integrate: monitorizare, sistem de alocare și instrumente de transfer.

2.1. Concepte fundamentale ale sistemelor distribuite

Orice sistem distribuit poate fi descris ca un sistem "*în care componentele comunică și își coordonează acțiunile pe baza mesajelor schimbate prin intermediul rețelei*" [3] fiind "*perceput de utilizatori ca un sistem coerent*" [4].

Domeniul *sistemelor distribuite* a apărut la sfârșitul anilor 1970 datorită răspândirii pe scară largă a rețelelor de calculatoare. Indiferent de scop și mărime, orice sistem distribuit are la bază câteva concepte simple și în același timp esențiale. De la Internet, unul din cele mai mari sisteme distribuite actuale, și până la sistemele de tip Grid [5] și Cloud [6] [7], aceste aspecte reprezintă fundația pe care sunt bazate.

Eterogenitatea este caracteristica incontestabilă a oricărui sistem distribuit. Orice astfel de sistem reprezintă un mediu *eterogen*, reunind o gamă largă de platforme de calcul, sisteme de operare și rețele de comunicație.

Interoperabilitatea unui sistem distribuit este descrisă prin publicarea serviciilor oferite utilizatorilor și aplicațiilor externe prin intermediul interfețelor de acces, folosind protocoale deschise cum ar fi Interface Description Language (IDL), Web Service Description Language (WSDL), REST/JSON, XDR, ProtoBuf.

Transparența este abilitatea de a ascunde detaliile interne utilizatorilor sistemului. ANSA [8] propune opt forme de transparență: *Transparența accesului* permite accesul la resurse fără a face cunoscut modul în care datele sunt accesate. *Transparența locației* permite accesul fără a divulga locația resurselor. *Transparența migrației* ascunde deplasarea resurselor. *Transparența concurenței* ascunde utilizatorului partajarea resurselor, permițând mai multor precese să ruleze în paralel. *Transparența replicării* permite folosirea mai multor instanțe replicate. *Transparența la defecte* ascunde defectele și revenirea din acestea. *Transparența performanței* permite reconfigurarea resurselor cu scopul de a crește performanța sistemului. *Transparența scalării* permite cresterea capacității fără schimbări structurale

Concurența este o trăsătură moștenită din paradigma programării paralele. Presupune execuția în paralel a mai multor cereri și în același timp sincronizarea accesului la resursele partajate.

Scalabilitatea reprezintă abilitatea sistemului de a acomoda o creștere însemnată în numărul de cereri utlizator fără o scădere semnificativă a performanței.

Securitatea poate fi considerată una din cele mai dificile aspecte din punct de vedere al arhitecturii și implementării ce includ o multitudine de ramificații cum ar fi cartele criptografice (cryptographic smart-cards), parafocuri (firewall), sisteme de detecție a intruziunilor (IDS), atacuri (distribuite) de tip refuz-servicu (distributed denial of service). De exemplu, doar studiile în domeniul criptografiei [9], depășesc scopul acestei teze.

Toleranța la defecte este proprietatea sistemului de tratare a erorilor parțiale fără scăderea semnificativă a performanței. Există două posibile abordări: *mascarea* și *tolerarea* defectelor. *Redundanța* și *replicarea*[4] sunt tehnicile uzuale folosite pentru *mascarea erorilor*. Câteva bine cunoscute exemple sunt: baze de date replicate, Ethernet bonding, volume RAID de discuri [10].

Disponibilitatea este probabilitatea unui sistem să funționeze corect la un anumit moment de timp. ***Robustețea*** unui sistem reprezintă *disponibilitatea* acestuia într-un interval de timp, fiind exprimată ca *integrala disponibilității* pe acel interval.

$$R_{t_0 \to t_1}(S) = \frac{\int_{t_0}^{t_1} u(A(t_i))\, dt}{t_1 - t_0}$$

$$u\big(A(t_i)\big) = \begin{cases} 1, & pentru\ A(t_i) = 1 \\ 0, & pentru\ A(t_i) \neq 1 \end{cases}, A(t_i) \text{ este disponibilitatea în } t_i$$

Ecuația 1 *Robustețea* ca integrală a disponibilității pentru un interval de timp $t_0 - t_1$

Toate aspectele prezentate în această lucrare, de la analiza critică a tehnologiilor actuale până la arhitecturile și implementările propuse, au la bază aceste aspecte fundamentale ale sistemelor distribuite.

2.2. Sisteme de alocare a resurselor de rețea

Principalele rețele academice actuale USLHCNet, ESnet [11], GÉANT [12], Internet2 [13], NLR [14], SURFnet [15] sunt rețele hibride. Ele au posibilitatea de a oferi pe lângă servicii de Nivel3 (IP rutat) și circuite dinamice de rețea, unde bine cunoscuta paradigmă "cel mai bun efort"(best-effort) poate evolua spre o garantare a calității serviciului oferit în ceea ce privește fluxurile de date.

Rețelele naționale și internaționale de mare viteză sunt esențiale pentru aplicațiile cu foarte multe date, incluzând aici și pe cele din domeniul fizicii energiilor înalte (HEP). La baza modelului de distribuție a datelor produse acceleratorul de particule de la CERN se află "*LHC Optical Private Network*" (LHCOPN) [16] prezentată in Figura 2. Aceasta este o rețea optică privată care face legătura între CERN "Tier0" și celelalte 11 mari centre naționale "Tier1". Arhitectura rețelei este conformă modelului dezvoltat în proiectul MONARC [17].

O mare parte a aspectelor prezentate în cadrul acestei cercetări au fost testate practic pe infrastructura de rețea USLHCNet [18], care asigură partea transatlantică de conectivitate din rețeaua LHCOPN. Infrastructura de rețea conectează facilitățile Tier1 de calcul de la Fermilab - Fermi National Accelerator Laboratory (FNAL) [19] și Brookhaven National Laboratory (BNL) [20] cu Tier0 la CERN. Rețeaua este compusă din legături de 10 Gbps interconectând CERN în Geneva, MANLAN în New York, STARLIGHT în Chicago, și Netherlight în Amsterdam. Coloana rețelei include șase[5] circuite transatlantice OC-192 distribuite pe cinci cabluri optice diferite. Totodată mai include și două segmente terestre OC-192 în Europa și unul în America,

[4] Există o mică diferență între cele două. *Redundanța* nu implică neapărat replicarea totală (ex. volumele RAID de discuri [10]), dar cele două paradigme se regăsesc în multe situații împreună având același scop: *mascarea defectelor*

[5] În Noiembrie 2011

interconectând cele patru puncte de prezență (PoP) (Geneva, New York, Chicago and Amsterdam) într-o configurație de tip mesh parțial, prezentată în Figura 3.

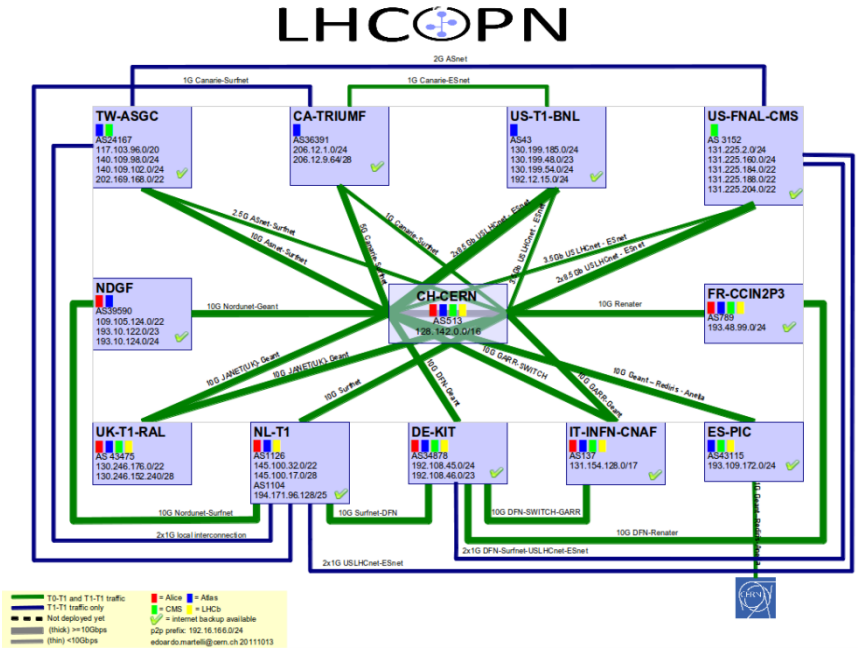


Figura 2 Rețeaua LHCOPN: LHC Optical Private Network - October 2011[6]

Nucleul rețelei USLHCNet este bazat pe comutatoarele de rețea multi-serviciu Ciena [21] CoreDirector [22], asigurând servicii de tip circuit("circuit-oriented"). Prin folosirea acestei arhitecturi, conexiunile în cadrul rețelei sunt compuse din circuite virtuale(*virtual circuits (VCs)*) cu o capacitate de bandă dinamică, ajustabilă folosind protocolul VCAT/LCAS. *VCAT – Virtual Concatenation* [23] [24] este o tehnologie care crează canale pentru transportul datelor peste SDH. *LCAS – Link Capacity Adjustment Scheme* [25] combinat cu *VCAT* asigură capacitate dinamică de transport "la cerere" ("on-the-fly").

OpenDRAC [26] este un sistem cu sursă deschisă (open-source) fiind o continuare a proiectului DRAC, dezvoltat inițial de Nortel și SURFnet [15]. Sistemul oferă servicii hibride, fiind capabil să aloce resurse la Nivelul 1 și respectiv 2 din stiva ISO/OSI. Redundanța sistemului de control este implementată folosind două servicii replicate și un mecanism verificare tip "*keep-alive*" pentru alegerea sistemului master.

On-demand Secure Circuits and Advance Reservation System (OSCARS) [27] este un proiect cu sursă deschisă (open-source) care urmărește furnizarea de circuite virtuale de rețea cu garantarea lărgimii de bandă. Sistemul este dezvoltat și folosit în producție de către Energy Sciences Network (ESnet) [11]. Administrarea și operarea circuitelor virtuale este realizată la Nivelul 3 de rețea. Circuitele sunt realizate folosind protocoalele Multi-Protocol Label Switching (MPLS) și Resource Reservation Protocol (RSVP) sau Label Switched Paths (LSP's).

Dynamic Resource Allocation in GMPLS Optical Networks (DRAGON) [28] este rezultatul unei cercetări ce include următorii parteneri: Universitatea Maryland (UMD), Mid-Atlantic Crossroads (MAX), Universitatea California de Sud, Information Sciences Institute East (USC/ISI) și Universitatea George Mason (GMU).

[6] Sursa: Edoardo Martelli, Departamentul CERN IT-CS

Scopul proiectului este dezvoltarea de tehnologii care să permită alocarea dinamică a circuitelor de rețea folosind protocolul GMPLS [29].

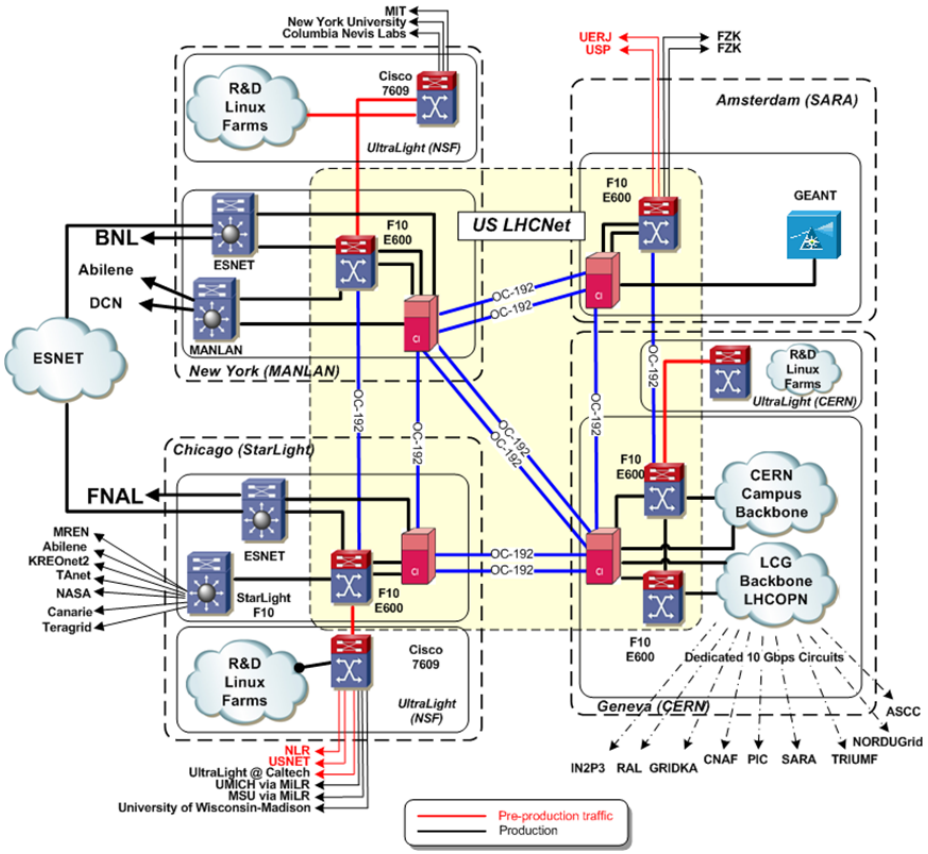


Figura 3 Diagrama USLHCNet în Noiembrie 2011

2.3. Monitorizarea sistemelor distribuite de mari dimensiuni

Monitorizarea reprezintă un aspect foarte important în orice sistem distribuit, nu numai prin faptul că asigură suportul necesar operării eficiente a sistemului, ci și prin faptul că poate semnala și chiar fixa anumite condiții de eroare. În cadrul acestui subcapitol vom analiza principalele soluții de monitorizare existente la ora actuală din perspectiva Concepte fundamentale ale sistemelor distribuite prezentate anterior.

Grid Monitoring Architecture (GMA) [30] este o specificație generică propusă de către Open Grid Forum[7] (OGF). GMA are la bază trei tipuri de componente: *Serviciul Director*, care are rolul descoperirii și publicării informațiilor, *Producătorul* care face disponibile datele de monitorizare și *Consumatorul* care primește datele. Arhitectura definește trei tipuri de interacții pentru transferul datelor între producători și consumatori: "***publicare/subscripție***", "***întrebare/răspuns***" și ***notificări***.

Monitoring and Discovery System (MDS) [31], versiunea 4, este un sistem de servicii de informare (monitorizare) parte integrantă din Globus Toolkit [32] și furnizează informații despre resursele disponibile în Grid și statutul acestora. MDS4 dispune de două servicii de nivel înalt: un *serviciu de index*, care colectează și publică informații de la sursele de informare, și un *serviciu declanșator*, care execută acțiuni în cazul în care sunt îndeplinite anumite condiții.

[7] Cunoscut în trecut sub denumirea de Global Grid Forum (GGF)

Relational Grid Monitoring Architecture (R-GMA) [33] este o implementare a specificației Grid Monitoring Architecture (GMA). R-GMA aduce în plus un limbaj standard de interogare (un subset din SQL) modelului GMA, astfel încât **consumatorii trimit cereri SQL și primesc răspuns tupluri (linii din baza de date)** publicate de către producători.

Gridscape II [34] este un portal web cu scopul de a ușura administrarea și prezentarea datelor de monitorizare.

GridICE [35] este un sistem de monitorizare puternic axat pe arhitecturile actuale de tip Grid. Este structurat în cinci nivele, începând cu producătorii datelor de monitorizare și terminând cu ultimul nivel al consumatorilor acestor date. Nivelele intermediare sunt responsabile pentru agregarea datelor, notificări, ultimul nivel fiind o interfață utilizator (GUI) pe web.

Ganglia [36] este un sistem de monitorizare scalabil și distribuit pentru sisteme de calcul tip cluster și tip Grid. Este bazat pe o arhitectură ierarhică destinată federațiilor de tip cluster și folosește un protocol de tip multipunct (multicast) pentru distribuția datelor local și un protocol punct-la-punct pentru conexiunea între nodurile de tip master între grupuri de calculatoare.

Nagios [37] este o platformă de monitorizare ușor extensibilă pentru sisteme și rețele de calculatoare. Pentru colectarea informațiilor de monitorizare se bazează pe un sistem extern de programe externe, denumite plugin-uri. Acestea pot fi atât aplicații compilate cât și scripturi shell.

2.4. Tehnologii pentru transferul datelor în medii data-intensive

Serviciile pentru transferul datelor sunt responsabile pentru administrarea transferurilor de date între facilități de stocare distribuite în întreaga lume.

GridFTP [38] este la ora actuală standardul *de facto* pentru transferurile de date în mediile de tip Grid. Aplicația este bazată pe protocolul FTP și folosește un canal separat de control pe lângă cele de date. Printre extensiile care merită puse în evidență enumerăm: *controlul transferului de date de către o terță parte, autentificarea, integritatea și confidențialitatea datelor, transferuri paralele, transferuri parțiale de fișiere, negocierea automată a ferestrei/memoriilor tampon pentru TCP, suport pentru transfer rezilient la erori prin rezumarea transferurilor eșuate*.

File Transfer Service (FTS) [39] reprezintă nivelul inferior al serviciilor de date din arhitectura gLite [40]. O instanță FTS utilizează un set configurabil de canale. Un canal reprezintă o abstractizare a unei legături de rețea (posibil dedicată) pentru transferul de fișiere între două centre de calcul. Există două asfel de canale: *Canale de producție*, de obicei fiind legături dedicate Tier0-Tier1, Tier1-Tier1 or Tier1-Tier2 și *Canale non-producție*, care sunt rețele normale partajate cu alte aplicații.

2.5. Concluzii

Acest capitol analizează elementele de bază necesare pentru implementarea platformelor de transferuri de date de mare viteză. Am început prin trecerea în revistă a **sistemelor de alocare a resurselor de rețea** ce reprezintă cel mai de jos nivel dintre cele trei obiective majore identificate în primul capitol. Remarcăm cu ușurință ca niciunul dintre sisteme **nu dispune de un sistem activ de autocontrol**, fie el

imbricat sau bazat pe un sistem de monitorizare extern. Toate sistemele de alocare folosesc o **abordare serială**, pas-cu-pas, pentru alocarea resurselor.

Din studiul **sistemelor actuale de monitorizare** in sisteme distribuite se poate identifica ușor model comun întâlnit în aproape toate sistemele studiate și anume faptul că efortul este in special concentrat pe **reprezentarea grafică** a datelor de monitorizare, cu mici excepții unde se pot defini și alarme. În modelele unde datele de monitorizare sunt expuse într-un mod mai generic identificăm probleme ce pot apărea prin inflexibilitatea adusă de o **schemă fixă** de reprezentare a datelor.

GridFTP a fost primul nostru candidat pentru a fi folosit ca și aplicație de transfer în cadrul acestei teze. Unul din principalele dezavantaje l-a constituit procedura greoaie de instalare și lunga listă de dependențe de biblioteci externe. Trecând peste această deficiență identificăm o altă problemă și anume **lipsa unei monitorizări active** și **imposibilitatea de ajustare dinamică a ratei de transfer** la nivel aplicație după ce transferul a început.

Pentru a obține performanțe sporite în cazul transferurilor de date de mare viteză susținem ideea unei abordări unitare în care aplicațiile de transfer și sistemele de alocare a resurselor de rețea folosesc în mod activ sisteme de monitorizare distribuite ca mecanism de autocontrol.

3. Contribuții la proiectarea și implementarea sistemului distribuit de monitorizare și control MonALISA

În cadrul acestui capitol prezentăm detaliile de proiectare și de implementare a platformei de monitorizare și control MonALISA: Monitoring Agents using a Large Integrated Service Architecture (*Agenți de Monitorizare bazați pe o Arhitectură de Servicii Distribuite*), subliniind contribuțiile originale ale autorului la acest proiect. Întreaga arhitectură, de la alegerea limbajului de programare și până la modelul de date și interfețele de acces, au la bază Concepte fundamentale ale sistemelor distribuite prezentate la începutul acestei teze.

3.1. Arhitectura sistemului

Platforma MonALISA [41] a fost proiectată și este concepută ca un ansamblu de subsisteme autodescriptive ce pot colecta, corela și publica orice tip de informație. Aceste sisteme sunt implementate ca agenți care se înregistrează și se descoperă în mod dinamic ca și servicii distribuite. Platforma oferă acestor servicii mecanismele de comunicare prin intermediul cărora agenții pot coopera și colabora în mod eficient la un nivel global, ceea ce face posibilă operarea într-o manieră robustă a unor sisteme complexe și larg răspândite. Arhitectura platformei este prezentată în Figura 4.

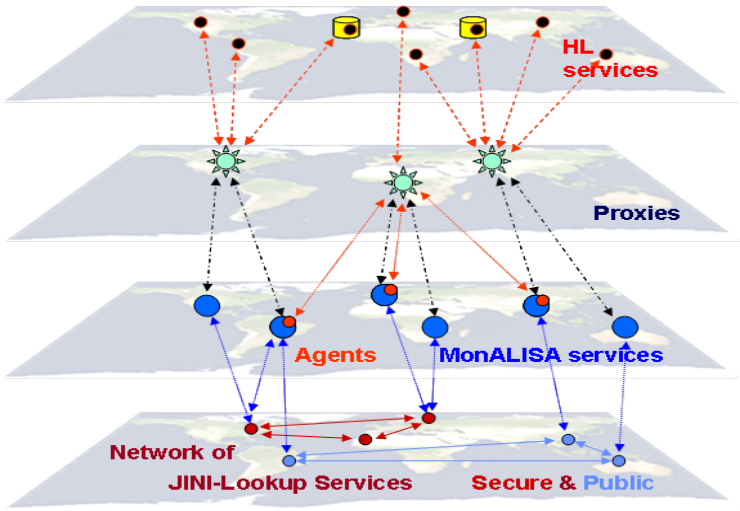


Figura 4: Arhitectura platformei MonALISA

Primul nivel este consituit de un ansamblu de *servicii de căutare* (*Lookup Discovery Services (LUS)*), cunoscute și sub numele de directoare de servicii, ce asigură înregistrarea și descoperirea serviciilor și a agenților în mod dinamic. Cel de-al doilea nivel este reprezentat de serviciile de monitorizare MonALISA care găzduiesc agenții și reprezintă sursa de informații de monitorizare. Al treilea nivel este reprezentat de serviciile Proxy care asigură o comunicație robustă între agenți asigurând în același timp acesul clienților și al serviciilor de nivel înalt la date printr-o multiplexare inteligentă a informației de monitorizare. Aceste servicii pot fi utilizate ca un nivel intermediar de securitate prin restricționarea accesului la date. Ultimul nivel este reprezentat de consumatorii datelor: servicii de nivel înalt și clienți ce pot colecta și corela informațiile din mai multe surse de la nivelul secund. Trebuie menționat faptul că serviciile MonALISA pot fi în același timp și consumatori ai datelor provenite de la alte servicii de monitorizare aflate pe același nivel.

Fiecare serviciu MonALISA se înregistrează într-un set de servicii de căutare, sau directoare de servicii (Lookup Services (LUSs)) ca membru într-unul sau mai multe grupuri publicând un set de atribute descriptive. Conceptul este similar cu cel al sistemului de nume în Internet (Domain Name System (DNS)), însă la nivel aplicație. Fiecare înregistrare are asociată o cuantă de valabilitate în timp, serviciul fiind obligat să o reînnoiască înainte ca aceasta să expire (lease)[8]. Serviciile de căutare sunt la ora actuală instalate în locații geografice diferite, rulând cel puțin unul în Europa, la CERN și cel puțin încă unul în America, la Caltech, Toleranța la defecte fiind în acest caz realizată prin replicarea serviciilor.

3.2. Serviciul MonALISA și colectarea informației de monitorizare

Serviciul MonALISA, aflat la nivelul secund în Figura 4, are la bază un ansamblu de subsisteme ce folosesc fire de execuție (multi-threaded), separate prin cozi de procesare. Aceste subsisteme operează în mod autonom fără a afecta performanța celorlalte. Detaliile și provocările implementării sunt prezentate în subcapitolul 3.5 (Figura 9). Dintre cele mai importante funcții enumerăm:

[8] Mecanismul similar și uneori folosit și sub denumirea de “keep-alive” sau “heartbeat”

- *Monitorizarea* unui număr mare de entități (calculatore, aplicații, servicii, dispozitive de rețea, etc.)
- *Filtrarea și agregarea* datelor de monitorizare, cu posibilitatea producerii de serii derivate prin reprocesarea și corelarea celor existente
- *Stocarea* datelor pentru perioade configurabile de timp atât în memorie cât și în baze de date locale serviciului
- *Servicii web* pentru accesul direct și Interoperabilitate la datele de monitorizare
- *Alerte, alarme și acțiuni locale* bazate pe datele de monitorizare
- *Control extern* folosind module speciale cu acces securizat ce permit acțiuni complexe care nu pot fi luate pe baza datelor locale. Aceste acțiuni sunt rezultatul analizei datelor de către servicii și clienți externi

Sistemul poate monitoriza și colecta date de la o serie de resurse precum clustere de calculatoare, aplicații, legături de rețea, comutatoare și rutere, servicii externe, camere video, etc, sau din interfațarea cu alte aplicații de monitorizare, cum ar fi de exemplu Ganglia. Un Modul de Monitorizare reprezintă unitatea de bază pentru colectarea datelor. Acesta poate fi încărcat dinamic într-un mod securizat, din biblioteci locale sau de pe rețea, și poate executa o procedură, rula un script sau poate folosi protocoale externe, ca de exemplu SNMP sau TL1, pentru a colecta un set de parametrii ce reprezintă datele de monitorizare.

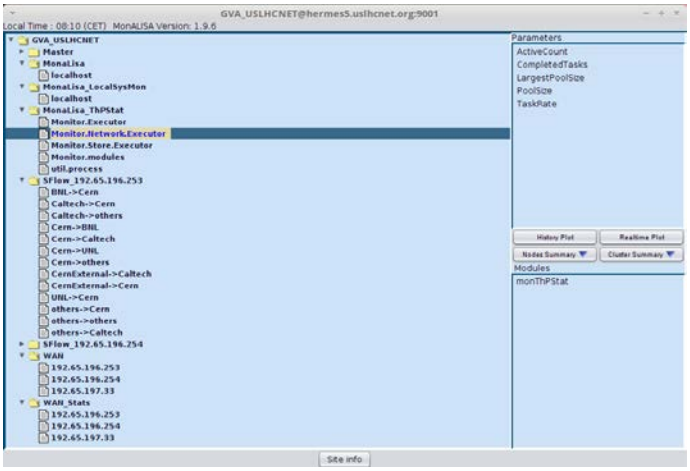


Figura 5 Ierarhia de bază a datelor de monitorizare

O **caracteristică importantă** a platformei o constituie faptul că organizarea datelor are la bază o **schemă flexibilă** nefiind impuse restricții asupra tipului sau modului de adresare a acestora. **Orice tip de date** pot fi monitorizate, publicate și diseminate în cadrul sistemului. Consumatorul trebuie însă să le poată decoda, însă nu și subsistemele intermediare din cadrul platformei. Clasa de bază este de tipul `Result`(Rezultat), care poate avea trei nivele de posibile chei de indexare (Figura 5). Pentru exemplul prezentat în Figura 5 ierarhia de bază a unui `Result` constă în:

- *Numele serviciului*; `GVA_USLHCNET` în exemplu
- *Numele clusterului*; de sus în jos: `Master, MonaLisa, Monalisa_LocalSysMon, MonaLisa_ThPStat, SFlow_192.65.196.253, etc`
- *Numele nodului*; `localhost, Monitor.Network.Executor, BNL->Cern, etc`
- *Numele parametrului*; `ActiveCount, CompletedTasks, LargestPoolSize, etc`

`Rezultatul` mai conține timpul când a fost produs, în milisecunde de la *Unix epoch* – 1 Ianuarie 1970, și un șir de valori de monitorizare, cele mai comune fiind `double` or `String`. Structura este suficient de generică pentru a putea reprezenta orice tuplu de forma `<cheie,valuare>`. "Cheia" care este formată de tuplul `<Serviciu, Cluster, Nod, Parametru>` poate fi ușor extinsă folosind marcatori pentru oricare din componentele tuplului ce formează "Cheia".

Ambele comutatoare optice, Glimmerglass [42] și Calient [43], folosite în timpul acestei cercetări pentru sistemul de alocare a resurselor de rețea descris în Capitolul 5, furnizează doar protocolul Transaction Language 1 (TL1) [44] pentru monitorizare și control. Același protocol este folosit și de dispozitivele de rețea Ciena CoreDirector ce constiuie nucleul rețelei USLHCNet.

Pentru comutatoarele de rețea am dezvoltat două module de monitorizare: unul pentru monitorizarea interconectărilor în interiorul comutatorului optic și unul pentru monitorizarea puterii optice pe porturi. O interfață generică de control a fost de asemenea dezvoltată, ce furnizează o interfață aplicație (API) simplă pentru interconectare sau deconectarea a două porturi.

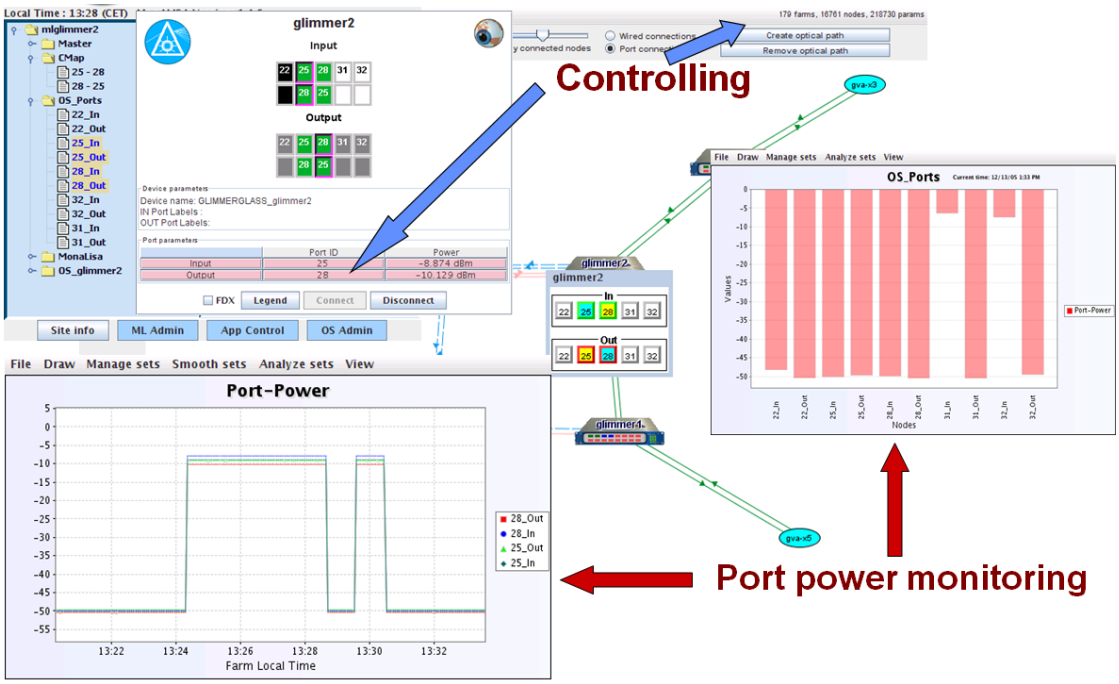


Figura 6 Interfața MonALISA pentru monitorizarea și controlul comutatoarelor optice

În Figura 6 sunt prezentate rezultatele ambelor capabilități: monitorizare și control folosind interfața grafică din clientul MonALISA. Bazat pe certificate X.509 importate în agentul optic ce controlează comutatorul (Optical Switch Agent (OSA)), un utilizator poate controla de la distanță comutatoarele optice direct din clientul grafic. Panoul din Figura 6 prezintă topologia rețelei în **timp real** inclusiv căile optice și parametrii de monitorizare precum puterea optică și interconectările. Această informație de monitorizare este propagată aproape în **timp real** către agenții optici OSA găzduiți de către serviciile MonALISA. Pentru o alocare mai rapidă a resursele sunt folosite două conexiuni TL1 independente, pentru monitorizare și pentru control.

Un aspect important în studiul si cuantificarea problemei transferurilor de date de înaltă performanță îl constituie asigurarea unei monitorizări extensive a sistemelor

implicate în transferurile de date. Un efort important a fost depus pentru dezvoltarea modulelor de monitorizare care să furnizeze informații despre sistem cum ar fi utilizarea procesorului, memoriei, rețelei și a discurilor, incluzând aici tranzacții, viteza de scriere și procentajul de ocuparea a benzii de intrare/ieșire a discului.

ApMon (de la **Ap**plication **Mon**itoring – Monitorizarea Aplicațiilor), este o bibliotecă eficientă de monitorizare a aplicațiilor oferind acestora posibilitatea publicării datelor de monitorizare într-un mod simplu si eficient. ApMon folosește ca protocol de transport datagrame UDP encodând informația transmisă într-un **format binar** (**XDR**), foarte eficient și în același timp **independent de platformă**.

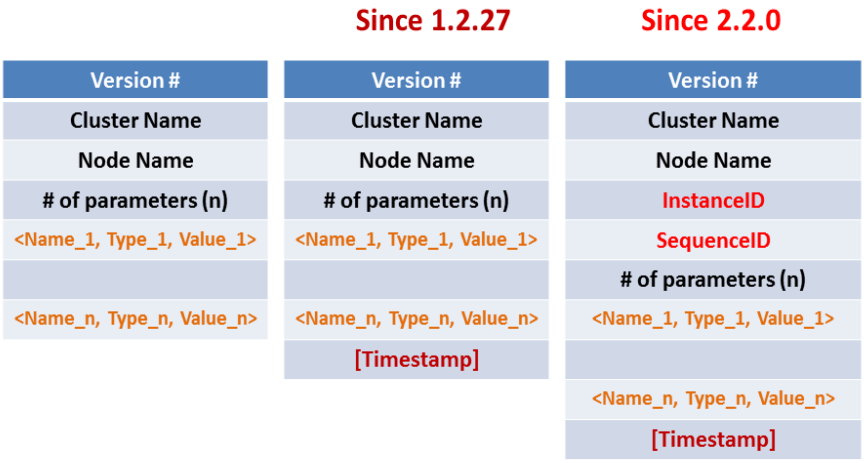


Figura 7 Evoluția protocolului ApMon

Prima versiune a protocolului a fost concepută de către autorul acestei teze, implementând în același timp și prima variantă de interfață aplicație (API) pentru aplicațiile Java și modulul de monitorizare MonALISA care decodează informația primită de la clienți. În cadrul datagramei UDP primul bloc este reprezentat de un antet ce conține numărul de versiune, limbajul folosit de client, urmat de numele clusterului și al nodului din ierarhia de bază a unui rezultat prezentat above, urmat de numărul total de parametrii. Ultima parte a pachetului este formată din tupluri `<parameter, type, value>`. Valorile pot fi numere sau șiruri de caractere.

Decizia de a adăuga un număr de versiune s-a dovedit a fi o strategie foarte utilă deoarece protocolul a putut evolua menținând compatibilitatea cu versiuni mai vechi. După cum se poate observa în Figura 7 timpul (timestamp) a fost adăugat în versiunea 1.2.27, în timp ce mecanismul de urmărire a pierderilor bazat pe un identificator de instanță a fost adăugat în versiunea 2.2.0.

În cadrul infrastucturii de monitorizare din experimentul Alice, bazat pe platforma MonALISA, agenții responsabili de execuția programelor în Grid, sunt instrumentați cu ApMon trimițând informații de monitorizare despre evoluția acestora cum ar fi utilizarea procesorului și a memoriei dar și informații despre sistemul unde aceste programe rulează.Toate aceste informații sunt trimise către un serviciu MonALISA instalat în fiecare din facilitățile de calcul ce fac parte din Grid-ul Alice.

În experimentul CMS este folosită o strategie diferită, centralizată (**care nu este agreată de autorul acestei teze**). Datele de la toate programele ce rulează în Grid-ul CMS sunt trimise folosind ApMon în câteva servicii centrale la CERN. După

cum se poate observa în Figura 8 serviciile MonALISA pot acomoda foarte ușor rate de colectare ce pot atinge 15 KHz cu pierderi neglijabile. Pierderile se datorează protocolului UDP.

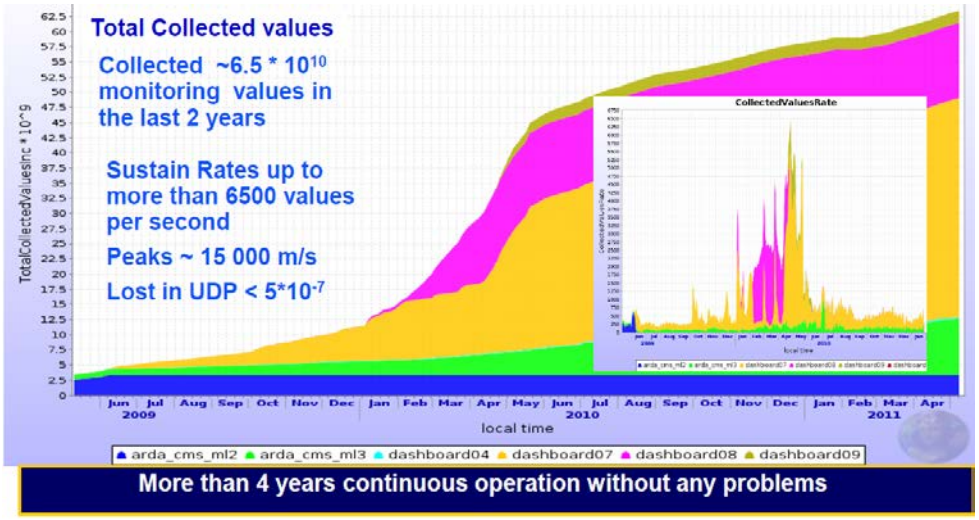


Figura 8 Performanța modului de colectare ApMon în CMS în ultimii doi ani (2009 – 2011)[9]

Sistemul de stocare este capabil să stocheze, să extragă și să proceseze datele de monitorizare în timp real. Platforma dispune de diferite implementări pentru stocarea datelor în funcție de scopul problemei: fișiere text sau baze de date SQL. Datele pot fi scrise direct, sau pot fi mediate înainte de a fi scrise cu păstrarea valorilor maximale. Baza de date preconfigurată cu serviciul de monitorizare este PostgreSQL. Sistemul de stocare este același pentru serviciul MonALISA și clienții tip repository, care arhivează datele pe termen lung. Pe lângă stocarea persistentă sistemul folosește o memorie tampon, care optimizează acesul la ultimele date folosite. Dimensiunea memoriei tampon se ajustează dinamic în funcție de memoria disponibilă în mașina virtuală Java.

3.3. Sistemul de agenți distribuiți și infrastructura de securitate

Serviciul MonALISA prezentat anterior în subcapitolul 3.2 folosește serviciile de descoperire (LUS) pentru a putea descoperi serviciile de Proxy, ce reprezintă cel de-al treilea nivel în Figura 4. Pe baza parametrilor obținuți din LUS serviciul MonALISA se conectează cu serviciul Proxy prin intermediul unei conexiuni TCP[10]. Clienții folosesc același mecanism pentru a se conecta cu serviciile de proxy. Aceste servicii de proxy sunt folosite ca să multiplexeze datele de monitorizare, în cazul în care mai mulți clienți sunt interesați de același set de date, și pentru comunicația între agenți. La momentul actual sunt active cinci astfel de servicii de proxy în locații total diferite: trei instanțe rulează la CERN, două în centrul de calcul și o a treia într-o altă clădire (Camera de Calcul Alice), și alte două în America, la Caltech. Dorim să subliniem încă o dată atenția deosebită acordată Toleranța la defecte prin replicarea serviciilor critice.

---

[9] Sursa: Iosif Legrand (CALTECH): CMS Monitoring Workshop, May 2011: [Online], Disponibil: https://indico.cern.ch/getFile.py/access?contribId=18&sessionId=1&resId=0&materialId=slides&confId=137822 [Accesat Noiembrie 2011]

[10] Mai multe detalii despre cum am ajuns să folosim conexiuni TCP sunt prezentate în subcapitolul 3.5

În cadrul platformei sunt folosiți o serie de agenți distribuiți capabili să colaboreze foarte rapid și să optimizeze la nivel global operarea și administrarea serviciilor distribuite de mari dimensiuni. Acești agenți comunică prin intermediul serviciului de proxy fiind găzduiți ca **entități autonome în interiorul serviciului MonALISA**. Schimbul de mesaje între agenți se realizează peste aceeași conexiune TCP stabilită între serviciul de monitorizare și proxy, ambele folosind cozi de priorități în funcție de urgența mesajelor. Sunt suportate atât mesaje directe (unicast), multipunct (multicast) cât și mesaje către toți agenții aceluiași grup (broadcast). Pentru o comunicare rapidă și eficientă este de dorit ca mesajele să rămână scurte.

Infrastructura de securitate se bazează pe protocoale standard având drept punct de plecare autentificarea pe bază de certificate X.509, acestea fiind cel mai des folosite în ziua de azi în sistemele Grid și nu numai. Pentru confidențialitatea și criptarea informațiilor senzitive este folosit protocolul Secure Sockets Layer (SSL). În cadrul platformei sunt folosite doar protocoale standardizate: TLS, PKI și GSS-API. Toate entitățile din cadrul platformei pot stabili canale virtuale securizate peste conexiunea standard TCP cu serviciul de proxy, platforma fiind capabilă să ascundă detaliile implementării acestora de către utilizatorii API-ului. Serviciul de proxy acționează doar ca un multiplexor de mesaje neputând decripta datele transmise pe canalele virtuale. Verificarea acestora se face doar de către cei doi parteneri care comunică peste canalul virtual securizat. Este nevoie ca în prealabil cei doi parteneri să fie în posesia cheilor publice ale partenerilor de comunicație, sau să folosească PKI cu o autoritate de certificare (CA), pentru a putea stabili canalele virtuale securizate. Oricare din clienți, agenți, servicii de monitorizare, chiar și module de monitorizare, pot folosi aceste canale securizate de transmisie.

3.4. Clienți și servicii de optimizare

În cadrul platformei există trei tipuri de clienți consumatori ai datelor de monitorizare, care au acces la informațiile produse de toate serviciile MonALISA. *Clienții grafici* oferă utilizatorului accesul interactiv furnizând o serie de posibilități de reprezentare a datelor în mod grafic. Pe lângă tipurile standard de reprezentare istorică sau în timp real platforma oferă posibilitatea reprezentărilor specializate, ca de exemplu cea de monitorizare a topologiilor de rețea. *Cleinții de tip Repository* permit stocarea datelor pe termen lung într-o bază de date pentru un set mai restrâns de parametrii ce pot fi preconfigurați. *Clienții de bază* reprezintă nucleul comun al tuturor consumatorilor de date din cadrul platformei, incluzând aici și cele două tipuri de clienți prezentate anterior. Aceștia furnizează funcțiile comune de acces la date în cadrul platformei, reprezentând fundația pe care se pot construi servicii de nivel înalt pentru optimizări și decizii la nivel global. Interacția între serviciile MonALISA și clienți este mediată de serviciile Proxy. Pe lângă posibilitatea de subscriere la informații de monitorizare, bazat pe restricțiile impuse în platforma de securitate, clienții pot configura și controla de la distanță parametrii serviciilor sau ai altor aplicații externe prin intermediul modulelor specializate.

3.5. Provocările implementării

Prezentăm în continuare provocările și soluțiile propuse în timpul dezvoltării platformei de monitorizare și control MonALISA. Autorul acestei lucrări și-a adus

contribuții majore pe plan teoretic dar și în procesul de implementare practică a arhitecturii de bază a sistemului, fiind implicat în toate aspectele legate de partea de comunicație între subsisteme, adresând diferite categorii de probleme specifice sistemelor distribuite cum ar fi Eterogenitatea, Concurența, Scalabilitatea și Robustețea generală a platformei.

Una dintre provocările dificile apărute în timpul implementării a fost legată de partea de comunicație pe rețea, datorită mediului Eterogenitatea compus din diferite sisteme de operare[11], diferite politici de securitate, caracteristic oricărui sistem distribuit. Prezentăm în continuare un sumar concis al celor mai des întâlnite ***premise false*** ale comunicației în sisteme distribuite, cunoscute ca "*Cele Opt Erori ale Calcului Distribuit*" "*The Eight Fallacies of Distributed Computing*" [45]:

1. Rețeaua este robustă.
2. Latența este zero.
3. Lățimea de bandă este infinită.
4. Rețeaua este sigură.
5. Topologia nu se schimbă.
6. Există un singur administrator.
7. Costul de transport este zero.
8. Rețeaua este omogenă.

Inițial comunicația între serviciul MonALISA și clienți a fost bazată pe implementarea Java a paradigmei de apel a procedurilor la distanță, cunoscută sub denumirea **Remote Method Invocation (RMI)**. Ascunzând detaliile de comunicație, aceast mecanism a constituit o soluție elegantă și, în acelși timp, extrem de potrivită pentru a adăugarea de funcționalități noi într-un ritm rapid. În momentul în care baza de utilizatori a crescut, serviciul de monitorizare începând să fie instalat în medii din ce în ce mai diferite, cu mecanisme și politci de securitate variate, am început să întâlnim tot mai des probleme cu protocolul de comunicație ales. Din multitudinea de probleme, cele mai dificile au fost legate de accesul peste rețea când conexiunea are pierderi nesemnalizate de stiva TCP ("silent drop") conexiunea rămânând blocată perioade lungi de timp. Din cauza unui control restrâns asupra tratării erorilor în cazul apelurilor la distanță, **RMI-ul a trebuit înlocuit cu o conexiune TCP**, pe bază de schimb de mesaje, folosind un mecanism de autocontrol pentru detecția erorilor, inclusiv pauzele ce indică pierderi nesemnalizate de protocolul TCP. Acest mecanism a fost imbricat chiar la nivelul conexiunii. Pentru păstrarea compatibilității ambele implementări au mers în paralel până în momentul în care toți consumatorii au folosit doar implementarea nouă. Noul mecanism este folosit acum între toate subsistemele platformei, asigurând o eficiență sporită, dar mai ales o strategie de detecție rapidă a erorilor nesemnalizate în nivelele inferioare ale sistemului de operare.

Printre contribuțiile practice importante aduse de autorul acestei teze în timpul dezvoltării proiectului evidențiem în continuare pe cele legate de Concurența, Scalabilitatea și Robustețea a sistemului, acestea numărându-se printre Concepte fundamentale ale sistemelor distribuite. Una din provocările majore ce influențează puternic timpul de răspuns a fost legată de **operațiile de intrare/ieșire (I/O**

[11] Deși majoritatea au fost și sunt sisteme Linux, am întâlnit probleme în nuclee (kernel) și distribuții diferite legate atât de implementări ale stivei TCP, cât și a sistemelor de fișiere

operations). Atât discul cât și rețeaua pot introduce întârzieri ce pot varia între câteva sutimi de secundă și până la nivelul zecilor sau sutelor de minute. Acestea sunt influențate nu numai de viteza dispozitivelor de stocare, sau cea a rețelei, dar mai ales de faptul că orice **operație blocantă** de **I/E (I/O)** suspendă firul de execuție sau procesul până când datele implicate în operație pot fi scrise sau citite, indiferent că este vorba de disc sau de rețea. Un alt aspect foarte important care dorim să-l subliniem este acela că **operațiile de I/O pot eșua**. Varianta optimistă este aceea că ele se întorc cu eroare, însă este de asemenea posibil ca acestea să rămână **blocate** vreme îndelungată. O schiță a componentelor și a fluxului de date în cadrul serviciului de monitorizare este prezentată în Figura 9.

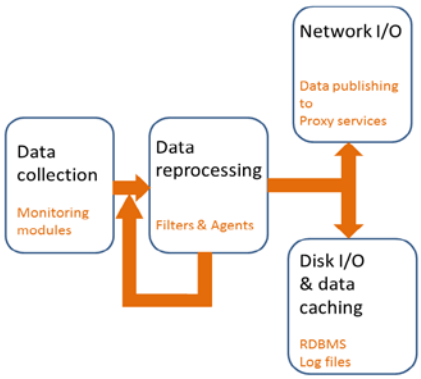


Figura 9 Fluxul datelor în serviciul MonALISA

Datele de moniorizare sunt produse în componenta "*Colectarea Datelor*" ("*Data collection*") de către modulele de monitorizare și transmise apoi în componenta "*Reprocesarea Datelor*" ("*Data reprocessing"*) unde sunt prelucrate în timp real de către filtre și agenți, rezultatele putând fi reinserate în fluxul de date. Atât în cazul producerii cât și în cel al reprocesării sunt folosite fire de execuție independente. În final datele sunt împinse în paralel în cele două mari componente de Intrare/Ieșire (I/O): stocare și rețea. Și aceastea folosesc tot fire independente de execuție. Toate subsistemele din cadrul platformei, sunt decuplate prin cozi de schimb (*rendezvous points*), folosind fire de execuție independente pentru a nu influența execuția și performanța celorlalte subsisteme. Pe baza Concurența impuse în toate subsistemele serviciului de monitorizare, și a platformei în general, s-a obținut îmbunătățirea timpului de răspuns al aplicației și, în același timp, a fost crescută Robustețea și Scalabilitatea platformei atît la nivel local cât și la nivel global.

Unul dintre cele mai importante calități ale platformei este posibiltatea de actulizare automată a serviciului de monitorizare, singura componentă cu răspândire largă neaflată sub administrarea noastră directă. Implementarea inițială a avut la baza mecanismul deja existent în Java, și anume Java Web Start, dar bazat pe o bibliotecă externă (NetX), folosită la ora actuală în distribuția OpenJDK. Mecanismul de WebStart distribuit cu platforma Java nu poate fi folosit, deoarece este adresat mai mult clienților grafici, în timp ce serviciile de monitorizare rulează pe platforme fără posibilitatea afișării grafice. Una din problemele întâlnite de-a lungul timpului a fost coruperea bibliotecilor sistemului de monitorizare mai ales pe sistemele de fișiere distribuite, gen NFS, AFS, deși am întâlnit cazuri și pe sisteme de fișiere locale. Datorită faptului că verificarea arhivelor JAR se face doar în momentul execuției, a

apărut necesitatea unui nou mecanism de actualizare. Aceasta folosește sume de control criptografice (MD5, SHA1, SHA512, etc.) pentru verificarea bibliotecilor imediat după ce au fost descărcate de pe Web și un mecanism tranzacțional către sistemul de fisiere (stocare) înainte de înlocuirea efectivă a fișierelor platformei. Pentru Robustețea mecanismului de actualizare sunt folosite două servere de web în locații geografice diferite, unul la CERN, în Europa și unul la Caltech, în America. Încă o dată dorim să subliniem folosirea replicării pentru o mai bună Toleranța la defecte a întregului sistem.

3.6. Sumar

Sistemul MonALISA pune la dispoziția utilizatorilor o platformă matură și în același timp ușor extensibilă pentru monitorizarea și controlul distribuit al sistemelor de mari dimensiuni. Pornind de la arhitectura de bază întemeiată Concepte fundamentale ale sistemelor distribuite și terminând cu detaliile de implementare a tuturor subsistemelor, platforma urmărește și adresează aspecte importante ale sistemelor distribuite. Întregul sistem pune la dispoziție resursele necesare adresării problemei aplicațiilor data-intensive:

- O monitorizare completă a sistemelor implicate: calculatoare (procesor, stocare, rețea), aplicații și dispozitive de rețea
- O infrastructură matură și robustă pentru dezvoltarea de agenți cu posibilitatea ca aceştia să colaboreze și să proceseze datele de monitorizare din sistem

Datorită calităților enumerate mai sus putem concluziona că platforma dispune de mijloacele necesare implementării serviciilor distribuite pentru alocarea dinamică a resurselor de rețea pentru modelul propus în capitolul 5.

4. Proiectarea și implementarea unei soluții performante pentru transferat date: Fast Data Transfer (FDT)

Unul din elementele cheie din timpul acestei cercetări l-a constituit o aplicație performantă pentru transferul de date, aspect evidențiat ca unul din cele trei obiective majore pentru adresarea problemei aplicațiilor data-intensive. Pe baza concluziilor prezentate în subcapitolul 2.5, prezentăm în acest capitol un efort susținut pentru dezvoltarea unei noi aplicații de transfer, cu posibilitatea **ajustării dinamice ratei de transfer**.

4.1. Obiective principale

Poiectul Fast Data Transfer (FDT) [46] este o aplicație relativ recentă, dezvoltată de autor în decursul acestei teze. Principalele deziderate care au stat la baza proiectului au încercat surmontarea lacunelor din aplicațiile actuale, după propriile încercări de a folosi aplicațiile existente:

- Performanță cel puțin comparativă cu cea a actualelor aplicații similare
- Procedură de instalare cât mai simplă, pe cât posibil fără dependințe externe
- Interfață de linie de comandă intuitivă și usor de folosit
- Disponibilitatea pe toate platformele majore de calcul
- Securitate compatibilă cu cea din Grid și suport inclus pentru protocolul SSH
- Posibilitatea controlului dinamic al ratei de transfer de către servicii externe
- Posibilitatea monitorizării active

- Posibilitatea paralelizării accesului la sistemul de stocare acolo unde este posibil; de exemplu în sistemele de fișiere precum Lustre și HDFS(Hadoop)

Cel mai important obiectiv, și anume performanța, a fost atins prin folosirea mai multor canale TCP pentru transportul datelor, augmentând capabilitățile de transfer DMA din librăriile NIO (New I/O) din Java.

4.2. Arhitectura și detaliile implementării

Aplicația este structurată în trei mari nivele prezentate în Figura 10. În partea superioară se află Controlorul de Sesiuni (**Session Manager**), responsabil cu inițializarea canalului de control, a sesiunilor de transfer citire/scriere (Reader/Writer Session) și asocierea de identificatori unici acestora. Componenta următoare, cea de securitate și al permisiunilor de acces a fost conceput ca un sistem de "plug-in"-uri. La ora actuală filtrarea pe bază de IP și protocolul SSH sunt distribuite o dată cu aplicația, în timp ce autentificările de Grid (Globus-GSI și GSI-SSH) necesită biblioteci externe. Nivelul de control permite interfațarea cu aplicații externe, MonALISA and LISA [47] dispunând de module speciale pentru controlul FDT.

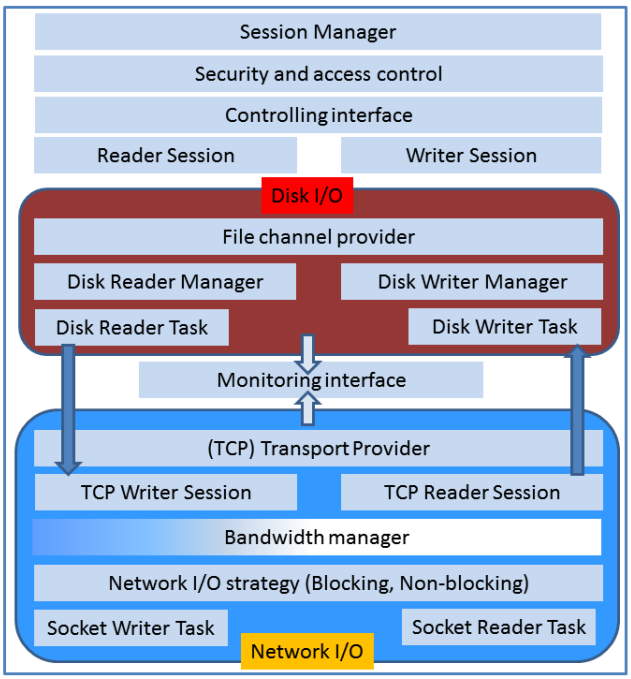


Figura 10 Arhitectura FDT

Ultimele două nivele tratează aspectele de **I/O** (**I/E Intrare/Ieșire**). Bazat pe experiența implementării mecanismelor de comunicație în cadrul platformei MonALISA, autorul a adoptat aceleași strategii, prezentate în cadrul subcapitolului 3.5 Provocările implementării.

În cadrul subsistemului de acces la disc (**Disk I/O**) a fost implementă o interfață generică (`FileChannelProvider`) pentru accesul la fișiere. Majoritatea sistemelor de fișiere asigură o interfață standard (POSIX), dar există și cazuri în care este necesară folosirea unor apeluri non-standard, de exemplu HDFS. Un alt exemplu ar fi folosirea de pipe UNIX pentru a scrie date, în loc de fișiere. Cele două componente care gerează accesul la discuri (**Disk Reader/Writer Managers)** sunt responsabili pentru citirea și scrierea eficientă a datelor, folosind optimizări cum ar fi

recunoașterea partițiilor fizice și folosirea lor în paralel. Pentru sisteme de fișiere precum HDFS s-au obținut performanțe mai bune [48] prin folosirea mai multor operații I/O, citiri și scrieri, în paralel (**Disk Reader/Writer Task**) pentru o aceeași partiție fizică.

Subsistemul de rețea (**Network I/O**) supervizează toate operațiile de I/O la nivelul rețea. Fișierele sunt împărțite în blocuri (`FileBlock`) de către subsistemul de acces la disc (`DiskReaderTask`) care sunt apoi "consumate" de către subsitemul rețea în partea diametral opusă, rolurile sunt inversate rețeaua fiind cea care citește blocurile și subsistemul de acces la disc cel care le consumă. Partea de rețea suportă două strategii pentru transferul blocurilor: **blocant** și **non-blocant**. Deși cel din urmă prezintă avantaje din punct de vedere al scalabilității, există chiar și la ora la care scriem această teză probleme cu sistemele de operare Linux[12] care au dus la folosirea implicită a primei strategii, cea de-a doua putând fi activată prin intermediul unui parametru. În cazul primei strategii pentru fiecare canal de date este folosit un fir de execuție ceea ce îl face să nu scaleze la un număr prea mare de clienți. Pe baza testelor, cea de-a doua strategie scalează până la zeci de mii de canale TCP.

Interfața de monitorizare folosește protocolul ApMon, descris anterior, pentru a trimite date de monitorizare precum rata de transfer, ocuparea internă a cozilor de transfer, dar și o monitorizarea completă a sistemului de calcul folosit, reușindu-se astfel o mai bună înțelegere a perfomanțelor aplicației.

Pentru controlul ratei de transfer (Bandwidth Manager) s-a folosit strategia "token bucket" (găleată cu jetoane) [49] [50] prin auto-limitarea transmițătorului. Pentru fiecare sesiune FDT se folosește un `SpeedLimiterTask` ce se execută periodic la un interval fix de timp. Acesta notifică subsistemul de rețea (`TCPWriterSession`) apariția unui nou "jeton", ca număr de octeți ce pot fi transferați în următoarea cuantă de timp: `ratăTransfer*(timpCurent-timpExecuțieAnterioară)`. Acuratețea depinde foarte mult de rata cu care se execută `SpeedLimiterTask`, și rata de transfer dorită.

4.3. Teste inițiale de perfomanță

Din multitudinea de teste realizate pentru înțelegerea și adresarea problemelor de performanță prezentăm în continuare, din motive istorice și oarecum nostalgice, primul test realizat peste o rețea WAN. Testul a fost realizat peste rețeaua USLHCNet folosind segmentul dintre Geneva și New York (RTT 93ms). Sistemele de calcul folosite în cadrul acestui test au avut următoarele specificații: 2 Procesoare Dual Core Intel Xenon @ 3.00 GHz, 4 GB RAM, 4 discuri SATA de 320 GB SATA fiind conectate cu plăci de rețea de 10Gbps Myricom direct în ruterele de la CERN și respectiv MANLAN. Topologia rețelei este prezentată în Figura 11. Comutatorul Optic Glimmerglass a fost folosit în al doilea test, de data aceasta disc la disc, pentru a comuta optic un al doilea sistem de stocare cu o performanță de disc mai bună.

[12] Implementarea stivei de TCP pe sisteme RedHat, CentOS, Scientific Linux, versiunile 5.x, datorate în principal notificărilor selective din TCP (SACK).

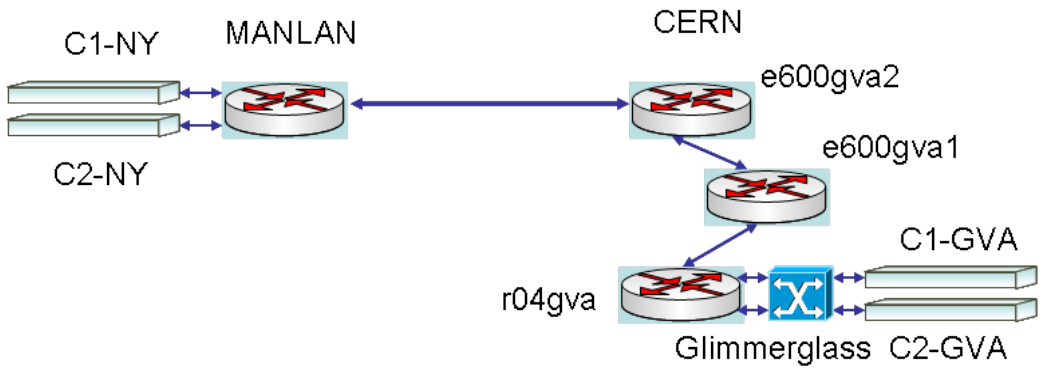


Figura 11 Topologia rețelei pentru testele Disc la Disc peste o rețea WAN

Rezultatele primului test sunt prezentate în partea stângă din Figura 12. Aplicația a fost testată citind și scriind patru discuri fizice în paralel. Rata de transfer scade în timp datorită faptului că rata de scriere pe discuri normale scade o dată cu gradul de umplere. Traficul mediu s-a situat în jurul valorii de 210 Mbytes/s.

Un al doilea test a fost efectuat pe un segment de rețea mai lung (RTT 170ms), de data aceasta între CERN, Geneva, și Caltech, Pasadena. Topologia rețelei pe segmentul Geneva – New York a fost aceeași ca și în testul anterior. Între New York și Pasadena, traficul a fost rutat peste rețeaua Internet2 și apoi rețeaua academică din California, CENIC. Cele două sisteme de discuri au avut următoarea configurație: 2 procesoare Intel Dual Core Woodcrest (3.00GHz), 6 GB RAM, 2 controloare RAID ARECA cu 24 de discuri SATA. Rezultatele obținute (partea dreaptă din Figura 12) arată faptul că aplicația FDT rulează practic la capacitatea maximă (~550MB/s) de scris/citit pe disc a celor două sisteme folosite. Vom prezenta mai multe rezultate experimentale ce vor proba performanța de excepție a aplicației în subcapitolul 0.

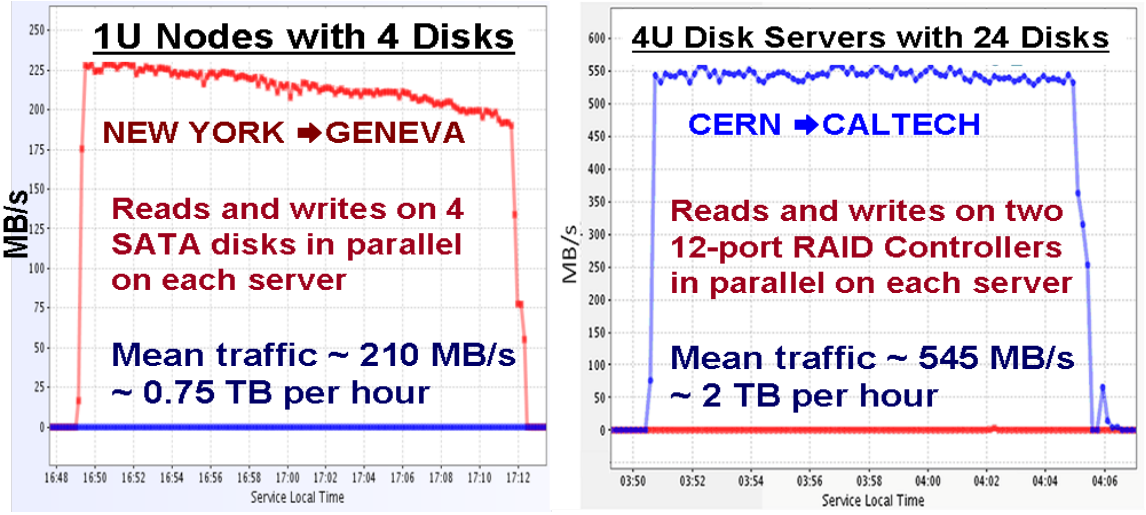


Figura 12 Performanța aplicației FDT în teste peste rețele WAN

4.4. Sumar

Pornind de la cele trei obiective majore subliniate la începutul lucrării și datorită faptului că niciuna din aplicațiile actuale nu suporta capabilități de ajustare dinamică a ratei de transfer, am dezvoltat o aplicație nouă, ce dovește o performanță excelentă, fiind capabilă să transfere datele la viteza maximă a rețelei sau a mediului de stocare.

Efortul prezentat în acest capitol surmontează lacunele aplicațiilor existente, având o procedură de instalare simplă, dispunând de interfețe de control extern al ratei de transfer precum și posibilitatea unei monitorizării active atât a transferurilor cât și a sistemelor de calcul pe care rulează. Deși a pornit ca un proiect de cercetare, baza de utilizatori ai aplicației FDT depășind comunitatea HEP (High Energy Physics) și mediul de cercetare, fiind ultilizată cu succes chiar în mediul comercial.

5. Arhitectura și implementarea sistemului de servicii distribuite pentru alocarea dinamică a cailor optice

Ultimul din cele **trei obiective majore** identificate în primul capitol al tezei este reprezentat de un sistem eficient de alocare a resurselor de rețea. Vom prezenta în continuare un model formal pentru un astfel de sistem pentru alocarea dinamică de căi optice la Nivelul1 de rețea. Un important aspect arhitectural este impus de faptul că nu există nici o modalitate de semnalizare "în-bandă" (in-band), deci orice tip de comunicație de control trebuie să folosească semnale "în-afara-benzii" (out-of-band). Dispozitivele fizice de rețea ce trebuiesc controlate sunt doar **Comutatoare** pur **Optice** la Nivelul 1 ISO/OSI pentru Planul de Date (*Data Plane*). Un model simplificat de Comutator Optic este prezentat în Figura 13. Din convenienţă porturile și fibrele sunt etichetate **IN**put și **OUT**put. Operația de bază a unui Comutator Optic FXC este interconectarea unui port de intrare cu unul de ieşire.

Proprietatea FXC: La orice moment de timp **un port** al unui Comutator Optic FXC poate fi **liber** sau implicat într-o interconectare (**FXC**) cu **un singur alt port** pereche.

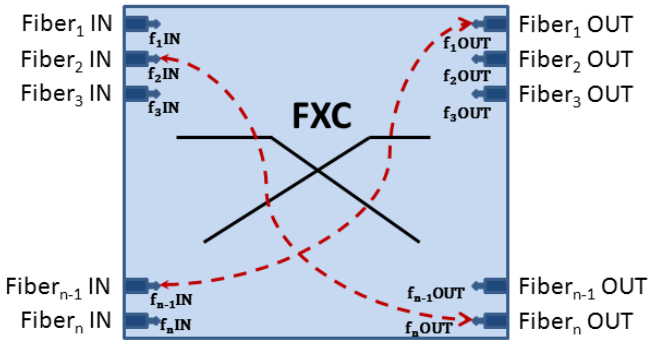


Figura 13 Schiță a unui Comutator Optic FXC

cu două interconectări (FXC) active

5.1. Abstractizarea unei topologii de rețea pur optică

În Figura 14 este prezentat un model generic de rețea optică bazată pe Comutatoare Optice. Liniile albastre reprezintă conexiunile la Nivel1. Modelul teoretic care descrie o astfel de topologie cu mai multe conexiuni posibile între două noduri este cunoscut sub denumirea de *multigraf* [51][52].

Definiţa 1: Un *multigraph* $M = (V, E)$ este o pereche de mulţimi disjuncte de *noduri* şi *muchii* şi o funcţie $E \longrightarrow V \cup [V]^2$ care asociază fiecărei muchii unul sau două noduri (capetele muchiei).

De aici înainte modelul studiază nucleul topologiei (muchiile și dispozitivele albastre din Figura 14), deoarece extremele pot fi atașate unei căi optice printr-o simplă interconectare FXC.

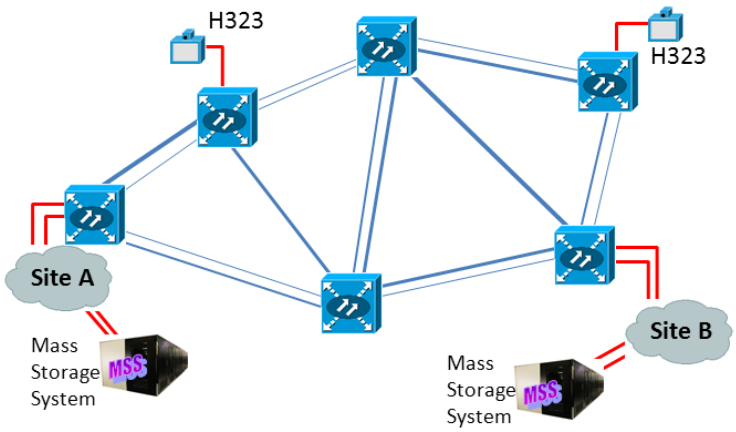


Figura 14 Reprezentarea unei topologii de rețea compusă din Comutatoare Optice FXC

În cazul modelului nostru formal fiecare Comutator Optic FXC devine un nod în graf și fiecare fibră optică ce interconectează comutatoarele devine o muchie.

Definiția 2: Un comutator optic $os \in O^S$ este o entitate de rețea capabilă să interconecteze semnalele optice dintr-un circuit în alt circuit.
Definiția 3: Un port optic $f_k \in F^O$ este o entitate a unui comutator optic în care se conectează o fibră optică.

O subcategorie a acestor dispozitive o reprezintă Comutatoarele Optice FXC:

Definiția 4: Un Comutator Optic FXC $os^f \in O^F$, cu $O^F \subset O^S$, este un comutator optic capabil să comute toate semnalele optice (lungimile de undă) dintr-un port de intrare într-un port de ieșire.
Definiția 5: Un port de intrare $f_k^{IN} \in F^{IN}$ este un port optic pentru care direcția semnalului optic este dinspre exterior spre interiorul comutatorului optic. Un port de ieșire $f_k^{OUT} \in F^{OUT}$ este un port optic pentru care direcția semnalului optic este dinspre interior spre exterior.

Cele două mulțimi sunt disjuncte $\mathbf{F^{IN} \cap F^{OUT} = \emptyset}$. Funcționalitatea unui comutator optic de a interconecta o fibra de intrare cu una de ieșire poate fi exprimată în mod formal după cum urmează:

Definiția 6: O interconectare optică este o funcție:

$$fxc: \mathbf{F^{IN} x F^{OUT}} \longrightarrow \mathbb{Z}_2, unde\ \mathbb{Z}_2 = \{0,1\}$$

$$fxc(f_i^{IN}, f_j^{OUT}) = \begin{cases} 1, & f_i^{IN}\ este\ conectat\ cu\ f_j^{OUT} \\ 0, & f_i^{IN}\ nu\ este\ conectat\ cu\ f_j^{OUT} \end{cases}, \text{ unde } \begin{matrix} f_i^{IN} \in \mathbf{F^{IN}} \\ f_j^{OUT} \in \mathbf{F^{OUT}} \end{matrix}$$

Pe baza acestei definiții și a Proprietatea FXC putem deduce următorul corolar, care descrie în mod formal proprietățile unei interconectări:

Corolarul 1a: Fie $\langle f_i^{IN}, f_j^{OUT} \rangle \in F^{IN} x F^{OUT}$ un tuplu ce reprezintă o interconectare $fxc(f_i^{IN}, f_j^{OUT}) = 1$, atunci:

(i) $fxc(f_i^{IN}, f_k^{OUT}) = 0$ pentru $\forall k \neq j, f_k^{OUT} \in F^{OUT}$, și
(ii) $fxc(f_k^{IN}, f_j^{OUT}) = 0$ pentru $\forall k \neq i, f_k^{IN} \in F^{IN}$.

Datorită faptului că fiecare fibră între oricare două cumutatoare optice este conectată în două porturi se poate usor folosi o funcție de etichetare cu ajutorul căreia să se poată identifica unic o muchie în multigraf. Orice muchie între două noduri, $u, v \in O^F$ este identificată în mod unic prin tuplul: $\langle u, f_u^{OUT}, v, f_v^{IN} \rangle$, unde $f_u^{OUT} \in F_u^{OUT}$ este portul de ieșire din nodul sursă, iar $f_v^{IN} \in F_v^{IN}$ este portul de intrare în nodul destinație.

Definiţia 7: O topologie FXC este un multigraf etichetat:

$$M^F = (O^F, E, l)$$

unde O^F este mulţimea nodurilor, F^{IN}, F^{OUT} sunt mulţimile porturilor de intrare, respectiv ieşire, iar E este mulţimea muchiilor şi fie l o funcţie de etichetare a muchiilor după cum urmează:

$$l: E \longrightarrow O^F x F^{OUT} x O^F x F^{IN}$$

$l(e_{ij(uv)}) = <u, f_{iu}^{OUT}, v, f_{jv}^{IN}>$, unde

$u, v \in O^F$, nodurile sursă respectiv destinaţie a muchiei

f_{iu}^{OUT} portul sursă pentru nodul u şi

$f_{jv}^{IN} \in F_v^{IN}$ portul destinaţie pentru nodul v

În cazul conectivităţii de tip *simplex* (fluxuri multimedia) topologia este reprezentată printr-un ***multigraf orientat***. Cazul *duplex* devine un ***multigraf neorientat***. În cazul general din Figura 14 topologia devine un ***multigraf mixt***. Datorită codomeniului discret de valori, funcţia fxc din **Definiţia 6** poate fi modelată şi ca o *funcţie caracteristică* a următoarei *relaţii*:

Definiţia 8: ***R-FXC*** este o *relaţie diadică* (*binară)* între mulţimea porturilor de intrare F^{IN} şi mulţimea porturilor de ieşire F^{OUT}. Definim că două porturi $f_i^{IN} \in F^{IN}$ şi $f_j^{OUT} \in F^{OUT}$ sunt într-o relaţie ***R-FXC-asociate***, $(f_i^{IN}, f_j^{OUT}) \in$ ***R-FXC***, dacă există o interconectare între ele. Fie χ notaţia pentru relaţia binară *R-FXC*. Următoarele notaţii sunt echivalente: $(f_i^{IN}, f_j^{OUT}) \in R\text{-}FXC \equiv f_i^{IN} \chi f_j^{OUT}$

Corolarul 1b: Relaţia ***R-FXC***, χ, este ***injectivă*** şi ***funcţională***.

Demonstraţie:

1) ***Injectivitatea*** :

$\forall f_i^{IN}, f_k^{IN} \in F^{IN}$ şi $f_j^{OUT} \in F^{OUT}$ dacă $f_i^{IN} \chi f_j^{OUT}$ şi $f_k^{IN} \chi f_j^{OUT}$ atunci $f_i^{IN} \equiv f_k^{IN}$

Prin *"Reducere la absurd"*: $\exists f_i^{IN}, f_k^{IN} \in F^{IN}$ ($i \neq k$) şi $f_j^{OUT} \in F^{OUT}$, pentru care $f_i^{IN} \chi f_j^{OUT}$ şi $f_k^{IN} \chi f_j^{OUT}$. Din Definiţia 6 a funcţiei fxc aceasta presupunere devine $fxc(f_i^{IN}, f_j^{OUT}) = 1$ şi $fxc(f_k^{IN}, f_j^{OUT}) = 1$

Din **Corolarul 1a** (ii): $fxc(f_k^{IN}, f_j^{OUT}) = 0$ pentru $\forall k \neq i$, $f_k^{IN} \in F^{IN}$ dacă $fxc(f_i^{IN}, f_j^{OUT}) = 1$, ceea ce contrazice ipoteza iniţială

2) ***Funcţionalitatea:***

$\forall f_i^{IN} \in F^{IN}$ şi $f_k^{OUT}, f_j^{OUT} \in F^{OUT}$ dacă $f_i^{IN} \chi f_j^{OUT}$ şi $f_i^{IN} \chi f_k^{OUT}$ atunci $f_j^{OUT} \equiv f_k^{OUT}$

Demonstraţia este banală folosind aceeaşi metodă ca şi în cazul injectivităţii combinat cu prima concluzie din **Corolarul 1a** (i).

O relaţie binară injectivă şi funcţională este cunoscută şi sub denumirea de relaţie ***unu-la-unu***. Inversa relaţiei, χ^{-1} posedă aceleaşi proprietăţi.

Corolarul 1c: Inversa relaţiei ***R-FXC***, χ^{-1}, este ***injectivă*** şi ***functională***.

Relaţia *R-FXC* transpune funţionalitatea de bază a unui comutator optic din funcţia de interconectare într-o relaţie între porturi, lucru ce ajută la o exprimare mai elegantă a căilor optice în cadrul topologiei modelate prin intermediul multigrafului. Pornind de la definiţia unui drum, cale, în graf [52] avem:

Definiţia 9: Un *drum* (sau *cale*) este un graf $\mathcal{P} = (V, E)$, unde

$$V = \{x_0, x_1, \ldots, x_n\}\ E = \{x_0x_1, x_1x_2, \ldots, x_{n-1}x_n\},$$

pentru care toate elementele x_i sunt distincte. Nodurile x_0 şi x_n sunt numite *capete*, x_0 fiind *nodul sursă* şi x_n *nodul destinaţie*, iar $x_1, \ldots, x_{n-1}$ nodurile intermediare.

În cazul mai generic al unui multigraf (**Definiţia 7**) orice drum în cadrul acestuia este descris formal după cum urmează:

Definiţia 10: Un *drum* (sau *cale*) în multigraful M^F este un multigraf:

$$\mathcal{P}^M = (O_P^F, E_P, l), unde\ O_P^F \subseteq O^F, E_P \subseteq E$$

$O_P^F = \{u_0, u_1, \dots, u_m\}, u_0$, *nodul sursă* iar u_m *nodul destinaţie*

$$E_P = \{e_0, e_1, \dots, e_{m-1}\}$$

$l: E_P \longrightarrow O_P^F x F_P^{OUT} x O_P^F x F_P^{IN}, F_P^{OUT} \subseteq F^{OUT}, F_P^{IN} \subseteq F^{IN}$ *funcţia de etichetare a muchiilor drumului*

$l(e_k) =< u_{k-1}, f_{o_{u_{k-1}}}^{OUT}, u_k, f_{i_{u_k}}^{IN} >$, pentru care

porturile de intrare şi ieşire ale tuturor nodurilor e_k *sunt* ***R-FXC-asociate***

Un drum în *multigraful* ce modelează topologia noastră de reţea are drept noduri comutatoarele, cu specificitatea că muchiile adiacente sunt interconectate, sau, formal: porturile muchiilor adiacente sunt în relaţia ***R-FXC-asociate***.

Lemă: Fie $\mathbb{P} = \cup \mathcal{P}_i^M$ mulţimea tuturor căilor în multigraful M^F, m fiind cardinalul acestei mulţimi, şi fie $E_{\mathcal{P}_i}$ mulţimea muchiilor pentru o cale $\mathcal{P}_i^M$, atunci:

$$\bigcap_{i=1}^{m} E_{\mathcal{P}_i} = \emptyset, pentru\ oricare\ m \geq 2, unde\ m = |\mathbb{P}|$$

Demonstraţie: Prin "*Reducere la absurd*" şi combinând tehnica demonstraţiei prin construcţie cu ipoteză iniţială vom demonstra că oricare două căi ce au o muchie comună sunt identice:

$\exists\ \mathcal{P}_x^M$ şi $\mathcal{P}_y^M$ unde $\mathcal{P}_x^M \cap \mathcal{P}_y^M \neq \emptyset, x \neq y$ drumuri îm $M^F \Leftrightarrow$

$\exists\ e_k \in E_{\mathcal{P}_x}$ şi $e_k \in E_{\mathcal{P}_y}$ o muchie comună în ambele căi $\mathcal{P}_x^M$ şi $\mathcal{P}_y^M$

Fie e_{k+1}^x următoarea muchie în calea $\mathcal{P}_x^M$ şi fie e_{k+1}^x şi fie e_{k+1}^y următoarea muchie în calea $\mathcal{P}_y^M$. Din **Definiţia 10** fie $l(e_{k+1}^x)$ funcţia de etichetare pentru $e_{k+1}^x \Leftrightarrow f_{i_{u_k}}^{IN} \chi f_{o_{u_k}}^{OUT}$, unde $f_{o_{u_k}}^{OUT}$ în tuplul $l(e_{k+1}^x)$.

Din **Corolarul 1b** relaţia χ este ***unu-la-unu*** $\Leftrightarrow f_{o_{u_k}}^{OUT}$ este şi în tuplul $l(e_{k+1}^y) \Leftrightarrow e_{k+1}^x \equiv e_{k+1}^y$

Folosind construcţia deducem: $e_q^x \equiv e_q^y, \forall\ q \geq k$, deci muchiile până la destinaţie sunt comune

Folosind **Corolarul 1c** pentru relaţia inversă χ^{-1} şi aceeaşi tehnică demonstraţia este trivială şi pentru muchiile înapoi spre sursă $e_q^x \equiv e_q^y, \forall\ q \leq k$

Din cele două rezultate $e_q^x \equiv e_q^y, \forall\ q \Leftrightarrow \mathcal{P}_x^M \equiv \mathcal{P}_y^M$, ceea ce contrazice ipoteza iniţială.

În mod informal acest rezultat este echivalent cu faptul că absolut **toate căile** în multigraful din **Definiţia 7** **nu au muchii comune**. Din demonstraţia **Lemă** avem:

Corolarul 2: Fie $\mathbb{P} = \cup \mathcal{P}_i^M$ mulţimea tuturor căilor în multigraf, unde $m = |\mathbb{P}|$ este cardinalul acestei mulţimi; fie $F_{\mathcal{P}_i}^{OUT} \subseteq F^{OUT}, F_{\mathcal{P}_i}^{IN} \subseteq F^{IN}$, mulţimea porturilor de intrare respectiv ieşire din cale $\mathcal{P}_i^M$ şi fie $E_{\mathcal{P}_i}$ mulţimea muchiilor din calea $\mathcal{P}_i^M$, atunci:

(ii) $\bigcap_{i=0}^{m} F_{\mathcal{P}_i}^{OUT} = \emptyset$ and

(iii) $\bigcap_{i=0}^{m} F_{\mathcal{P}_i}^{IN} = \emptyset$

Este important de notat faptul că orice cale optică în cadrul oricărei topologii de reţea formate din comutatoare optice poate fi considerată ca fiind exclusivă din punct de vedere al resurselor implicate. Toate porturile şi fibrele optice dintre comutatoare pot fi parte dintr-o singură cale optică. Acest rezultat este esenţial în cadrul algoritmului de calcul al drumului optim (cel mai scurt) în multigraf.

5.2. Determinarea drumului optim în reţele pur optice

Pornind de la premisa că fiecare legătură optică poate avea diferite proprietăţi specifice furnizorul circuitului de reţea cum ar fi calea fizică a fibrei optice, precum şi parametrii specifici cum ar fi lărgimea de bandă şi timpul "dus-întors" (Round Trip

Time (RTT)), vom introduce o funcție de cost asociată fiecărei muchii. În acest fel determinarea drumului optim devine o problemă de calcul a drumului cel mai scurt, foarte asemănătoare cu aceeași problemă din teoria grafurilor [53]:

Definiția 11: Funcția $f: E^P \longrightarrow \mathbb{R}^+$ definită pe mulțimea muchiilor multigrafului cu valori în mulțimea numerelor reale, este o funcție de cost ce asociază un număr real pozitiv pentru fiecare muchie a multigrafului, unde $\mathbb{P} = \bigcup \mathcal{P}_i^M$ este mulțimea tuturor căilor posibile în multigraful M^F.

Costul total al unui drum $\mathcal{P}^M$ (**Definiția 10**) în multigraful M^F (**Definiția 7**) este:

$$C(\mathcal{P}^M) = \sum_{k=0}^{m-1} f(e_k), unde\ e_k \in E_P$$

Drumul de cost minim de la un nod sursă u la un nod destinație v este:

$$\delta(u,v) = \begin{cases} \min\left(C\left(\mathcal{P}_{u\to v}^M\right)\right) & \mathcal{P}_{u\to v}^M \in \mathbb{P}_{u\to v} \\ \infty & \text{dacă nu există nici un drum posibil} \end{cases}$$

Cel mai scurt drum în multigraf de la nodul sursă u către nodul destinație v este orice cale $\mathcal{P}_s^M$ care satisface următoarea relație: $C(\mathcal{P}_s^M) = \delta(u,v)$

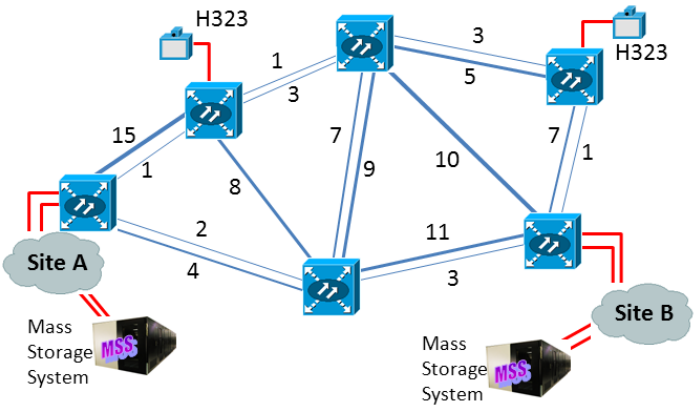


Figura 15 Schița pentru *multigraful mixt* suprapus peste costul asociat muchiilor

Problema celui mai scurt drum pornind de la o sursă este una bine cunoscută, cu aplicativitate directă în protocoalele de rutare. În cazul în care costurile asociate muchiilor pot avea valori negative se poate aplica algoritmul Bellman-Ford [54] pentru calculul drumului. Pentru cazul în care costurile asociate au doar valori pozitive algoritmul Dijkstra's [55] este mai rapid, fiind folosit în protocoale de rutare precum OSPF [56] and IS-IS [57]. În urma rezultatelor prezentate în cadrul **Lemă** și al **Corolarul 2** devine posibilă adaptarea algoritmului **Dijkstra** pentru ***calculul celui mai scurt drum*** în multigraful M^F prezentat în Figura 15.

În cazul nostru toate muchiile implicate într-o cale optică vor fi marcate ca nedisponibile și nu vor mai fi luate în calcul de către algoritm. Din motivul că sunt posibile mai multe muchii între două noduri ale multigrafului, vor fi folosite liste de adiacență în cadrul algoritmului. Diferența constă în faptul că fiecare element al listelor va fi implementat ca o coadă de priorități în loc de o referință către nodul vecin. Algoritmul este foarte asemănător cu cel al lui Dijkstra's, folosind două cozi

sortate: QE pentru lista de adiacență, și QM pentru a menține referințele către nodurile vecine.

```
SPMF (M^F, s)
        //Inițializare; toate distanțele către toate nodurile sunt infinite
        for each v in M^F do
                dist[v] = ∞;
                prev_edge[v] = NULL;
        end for
        //Inițializare cozii de priorități de la sursă
        dist[s] = 0;
        for each QE in AdjQ[s] do
                ENQ(QM, DEQ(QE));
        end for

        while QM not Ø do
                e = DEQ(QM);
                RELAX(src(e), dst(e), e);
                for each QE in AdjQ[d(e)] – {src(e)} do
                        ENQ(QM, DEQ(QE));
                end for
        end while
end SPMF
RELAX (u, v, e)
        //un nou posibil cost
        newDist = dist[u] + w(e);
        if dist[v] > newDist then
                dist[v] = newDist;
                prev_edge[v] = e;
        end if
end RELAX
```

Algoritmul 1: ***Cel mai scurt drum optic în multigraf*** **(SPMF** ***Single-source shortest-path)***

În momentul în care este determinat un drum, acesta va avea asociat un identificator unic, folosit în cadrul funcției de etichetare din Definiția 7 pentru a marca muchiile implicate în drum, și care nu vor mai fi luate în calcul la o iterație viitoare.

5.3. Considerații arhitecturale pentru alocarea distribuită a drumurilor optice

Din punct de vedere al implementării practice a Algoritmul 1 există două posibile alternative: fie o abordare centralizată în care există o entitate sau un serviciu satisface toate cererile controlând toate comutatoarele din rețea, sau o abordare total distribuită în care fiecare comutator optic are asociată o entitate, serviciu sau agent. În ambele cazuri toată topologia rețelei este cunoscută de către entitatea de control. Având în vedere aspectele evidențiate la începutul acestei teze (*2.1* Concepte fundamentale ale sistemelor), mai ales pe cele referitoare la Toleranța la defecte, cea mai naturală abordare este cea total distribuită, dar va necesita mai multe mesaje pentru propagarea infomațiilor despre schimbările din topologie. Acest fapt necesită o platformă inteligentă de comunicație între agenți [58]. Vom adresa acest aspect în următorul capitol al tezei.

Datorită alocări exclusive a resurselor în cadrul unei căi optice și a faptului că o interconectare in cadrul comutatorului optic poate dura de la cateva zeci de milisecunde la câteva minute se impune un nivel de pre-alocare a resurselor la nivelul

entităților implicate în calea optică. Soluția este o abordare tranzacțională distribuită, implementată la nivelul agenților care controlează comutatoarele. Soluția propusă este folosirea unei tranzacții distribuite în doi pași (two-phase commit (*2PC*)). Entitatea care primește cererea devine automat coordonator pentru acea cale optică, indiferent dacă ajunge sau nu să fie și el implicat în aceasta. Coordonatorul este desemnat ca fiind acel serviciu care primește cererea pentru calea optică.

```
MASTER_COMPUTE_AND_ALLOCATE_PATH(M^F, s, d)
        Path = SPMF (M^F, s)
        if !PREALLOC_LOCALLY(Ports[Path[s])) then
                ABORT();
                return FAIL;
        end if
        for all nodes n in Path
                SEND_PREALLOC(n, ports);
        end for
        Responses[] = WAIT_RESPONSE_FROM_SLAVES(timeout);
        if timeout OR then
                SEND_ABORT_AND_ABORT_LOCALLY();
                return FAIL;
        end if
        if Responses[] are OK then
                SEND_COMMIT_AND_COMMIT_LOCALLY();
                return SUCCESS;
        else
                SEND_ABORT_AND_ABORT_LOCALLY();
                return FAIL;
        end if
end COMPUTE_AND_ALLOCATE_PATH
```

Algoritmul 2 Protocolul tranzacției în doi pași (two-pahse commit *2PC*) pentru coordonator

Algoritmul 2 prezintă implementarea tranzacției în doi pași (*2PC*) la nivelul coordonatorului. Pentru ceilalți agenți subordonați coordonatorului implementarea este similară și este prezentată în cadrul Algoritmul 3.

```
SLAVE_ALLOCATE_PATH(Ports)
        if !PREALOC_LOCALLY(Ports) then
                SEND_PREALLOC_NOT_OK();
                return FAIL;
        end if
        SEND_PREALLOC_OK();
        canCommit = WAIT_RESPONSE_FROM_MASTER(timeout);
        if canCommit then
                COMMIT_LOCALLY();
                return SUCCESS;
        else
                ABORT();
                return FAIL;
        end if
end SLAVE_ALLOCATE_PATH
```

Algoritmul 3 Protocolul tranzacției în doi pași slave (two-pahse commit *2PC*) pentru subordonați

Ambele implemetări tratează cazul în care rețeaua de control devine indisponibilă prin intermediul unei limitări în timp a tranzacției (timeout). Fiecare tranzacție primește un identificator unic și o cuantă de valabilitate în timp. În cazul expirării tranzacția este abandonată, resursele fiind redate înapoi în sistem. Datorită naturii total distribuite a modului de tratare a cererilor este posibil să apară dispute ale

resurselor. O posibilă soluție este adăugarea unui câmp adițional în răspunsul de abandonare al tranzacției prin care coordonatorul poate încerca o strategie de revenire și reîncercare (back-off), similar cu abordarea din protocolul Ethernet [59] în cazul coliziunilor. Din concluziile studiilor [60] [61] care investighează stabilitatea tranzacțiilor distribuite și performanța algoritmilor de revenire și reîncercare (back-off) și din faptul că în cazul nostru rata de sosire a cererilor este cu ordine de mărime mai mică decât durata de viață a căilor optice, argumentăm că un algoritm de tipul binary exponential back-off [62] corespunde situației noastre.

Arhitectura reprezintă o **abordare inovatoare**, neîntâlnită în celelalte sisteme de alocare a resurselor de rețea, cererile fiind deservite într-o **manieră distribuită** prin folosirea unui mecanism de **tranzacții distribuite**. În același timp sistemul folosește un mecanism de **autocontrol** prin augmentarea unei infrastructuri mature de comunicație, care furnizează în același timp și capabilități robuste de **monitorizare și control**.

5.4. Detaliile implementării

Soluția de implementare [63] aleasă a fost făcută pornind de la faptul că platforma de monitorizare și control Contribuții la proiectarea și implementarea sistemului distribuit de monitorizare și control MonALISA, prezentată în capitolul 3, dispune de capabilitățile necesare implementării de servicii distribuite bazate pe agenți. Bazat pe această platformă am implementat sistemul de control pentru alocarea căilor optice: Optical Control Plane System (OCPS), care alocă, monitorizează și controlează căile optice, folosind comutatoare optice.

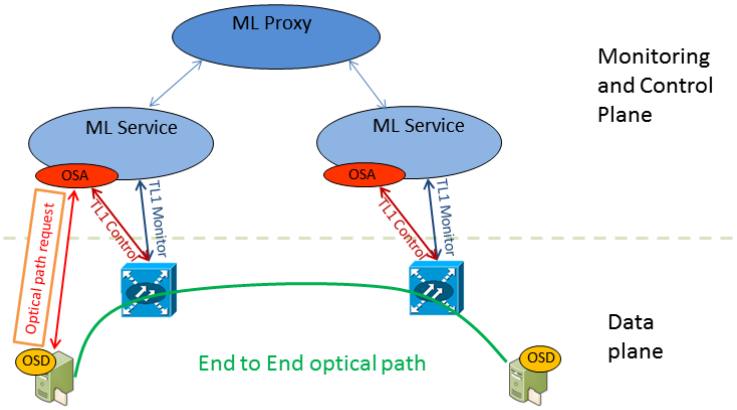


Figura 16 Schița sistemului de alocare a căilor optice

O prezentare de ansamblu a sistemului de alocare este prezentată în Figura 16. După cum am subliniat în deschiderea acestui capitol nu este posibilă nici un fel de comunicație "în-bandă"("in-band") între comutatoare optice. Toată partea de comunicație, monitorizare și control va folosi ceea ce se numește un *Plan de control*, prezentat în partea superioară din Figura 16, în timp ce datele efective vor folosi căile optice alocate în planul inferior, ce conține doar comutatoarele optice și sistemele implicate în transferul datelor, cunoscut în literatura de specialitate sub denumirea de *Plan de Date* ("Data Plane"). Din punct de vedere al implementării, partea de control augmentează capabilitățile Sistemul de agenți distribuiți din platforma MonALISA.

Arhitectura este slab cuplată, fiecare comutator optic fiind monitorizat de către o Serviciul MonALISA și colectarea informației de monitorizare, care găzduiește în același timp și un agent ce controlează comutatorul optic, **Optical Switch Agents (OSA)** (Agentul pentru Controlul Comutatoarelor Optice).

Ansamblul de agenți OSA colaborează folosind Sistemul de agenți distribuiți din cadrul platformei MonALISA pentru descoperirea topologiei rețelei și alocarea căilor optice, având în acelați timp acces la informația de monitorizare ce descrie starea comutatoarelor optice din întreaga rețea.

5.5. Agentul pentru Controlul Comutatoarelor Optice

Agentul care controlează comutatorul optic, se înregistrează în fluxul de date de monitorizare prezentat în Figura 9, având astfel acces la informațiile despre interconectările optice în interiorul comutatorului și a puterii optice[13] pentru porturile implicate într-o cale optică. În Figura 17 sunt prezentate principalele subsisteme ale agentului.

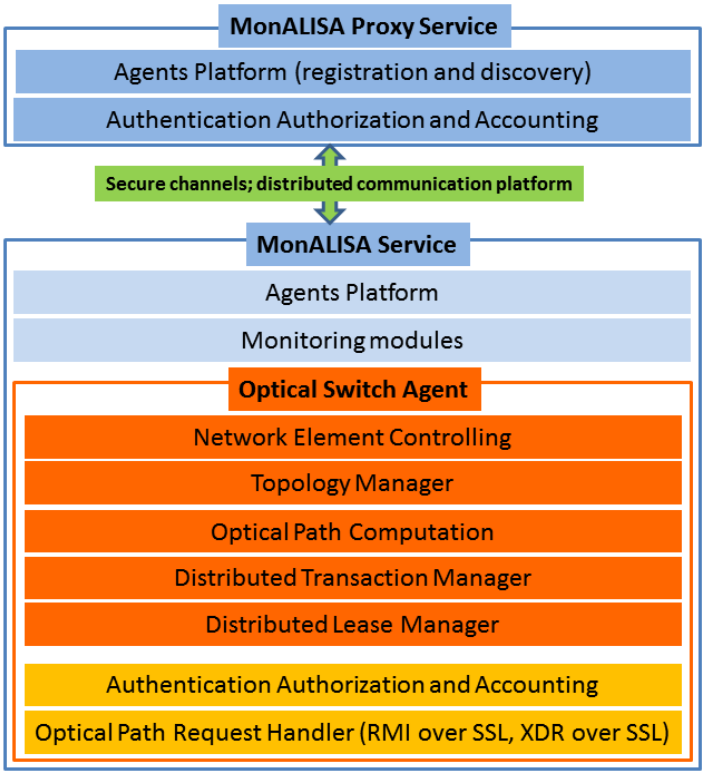


Figura 17 Arhitectura Agentului Optic de Alocare

Pe baza informațiilor schimbate între agenți de către Controlorul Topologiei (**Topology Manager**), aceștia sunt capabili să își formeze **propria perspectivă a întregii topologiei de rețea**. Agentul expune două interfețe de interacție cu programele, prin intermediul cărora se pot cere căi optice. Componenta care gerează cererile este reprezentată **Optical Path Request Handler**, cele două interfețe fiind:

- Apelul la distanță peste SSL (RMI over SSL) a fost integrat pentru simplicitatea și eleganța apelării direct din clientul grafic MonALISA

[13] Puterea optică poate fi monitorizată doar pentru anumite comutatoare optice; de obicei cele bazate pe tehonologii MEMS expun această informație.

- Pentru aplicații scrise în alte limbaje, este folosit un Interoperabilitate XDR peste SSL (XDR over SSL)

Ambele interfețe folosesc SSL și certificate X.509, accesul fiind permis de către componenta de Autentificare, Autorizare și Audit (**Authentication Authorization and Accounting**) din Figura 17, folosind infrastructura de securitate din MonALISA.

Determinarea drumului optim în rețele pur optice este implementat în componenta de calcul (**Optical Path Computation**) pe baza topologiei oferite de către Controlorul Topologiei. Fiecărui drum optic i se asociază un identificator unic (UUID - Unique Identifier). Agentul optic care primește cererea devine **Agentul coordonator** al căi optice, fiind responsabil pentru calculul drumului optim și coordonarea agenților implicați în calea optică, însă nu este necesar ca acesta să facă parte din acest drum. Imediat ce Algoritmul 1 determină o cale optică în cadrul topologiei, procedura de alocare începe prin operația de pre-alocare a resurselor implicate în calea optică. De această fază tranzacțională este responsabil Controlorul de Tranzacții Distribuite (**Distributed Transaction Manager - DTM**):

1. **DTM** pre-alocă porturile optice locale[14] și pornește o tranzacție prin trimiterea în paralel a unui mesaj **PRE-ALOCARE** către toți agenții implicați
2. Bazat pe configurația din acel moment agenții pre-alocă resursele implicate și apoi trimit agentului coordonator mesajul **COMITE**. Dacă nu este posibilă prealocare este trimis înapoi mesajul **ANULEAZĂ**.
3. În baza răspunsurilor primite coordonatorul trimite **COMITE** sau **ANULEAZĂ** către agenții implicați
4. Toate **DTM**-urile implicate folosesc apoi elementul de control al comutatorului optic (**Network Element Controlling**) pentru alocarea fizică a interconectărilor.

Având provocările prezentate în subcapitolul 3.5 și arhitectura total distribuită a agenților absolut **toate mesajele și tranzacțiile au o durată de viață** (lease) asociată[15]. Acesta reprezintă **unicul mod de implementare robustă a consistenței la nivel distribuit**. O îmbunătățire adusă algoritmului a fost posibilitatea unei abordări optimiste (ca și parametru) prin care interconectările sunt alocate imediat ce răspunsul COMITE este trimis către coordonator. Această optimizare are sens doar în cazul în care interconectarea porturilor se realizează în sute de milisecunde, interval de timp echivalent RTT-ului. În cazul în care această interconectare se realizează în interval de câteva zeci de secunde, se așteaptă prealocarea tuturor resurselor.

Din testele efectuate peste rețeaua USLHCNet timpul de satisfacere a unei cereri, incluzând toate etapele intermediare: de la primirea acesteia de către agent, calculul și alocarea căii, este de obicei sub o secundă, independent de numărul de comutatoare optice, motivul principal fiind că toată comunicația de control este realizată în paralel. Această abordare asigură un timp de răspuns suficient de mic pentru rerutarea căi optice, în cazul în care sunt detectate erori. Din acest motiv protocoalele de nivel înalt

[14] Este posibil să existe situații în care comutatorul optic controlat de Agentul Coordonator să **nu** fie implicat în calea optică pe care acesta o coordonează

[15] Strategia se regăsește sub câteva denumiri diferite: **lease, keep-alive sau hearbeat**, scopul rămânând același: detectarea autonomă a erorilor la nivel distribuit.

cum ar fi TCP, vor putea să continue neîntrerupt transferul de date cu o mică degradare a performanței în timpul rerutării.

5.6. Programul rezident al sistemului de transfer

Unul din obiectivele sistemului de alocare a căilor optice(OCPS) l-a reprezentat posibilitatea interfațării eficiente cu aplicațiile de transfer. Pentru aceasta am dezvoltat un program rezident(**Optical Switch Daemon - OSD**) care are scopul de a facilita interacția rapidă cu sistemul de agenți. OSD rulează ca un demon Java pe sistemul de pe care se pot porni transferurile de date și folosește pipe Unix cu nume pentru interfațarea cu aplicațiile locale. Pentru a reduce timpul de autentificare și autorizare acest program rezident menține o conexiune activă cu unul sau doi agenți. A doua conexiune este menținută din motive de redundanță.

5.7. Sumar

Modelul formal prezentat în acest capitol propune o abordare nouă în alocarea resurselor de rețea în cadrul rețelelor pur optice. Strategia propusă este una cu totul nouă comparativ cu sistemele de alocare prezentate în subcapitolul 2.2, eliminând entitatea centrală responsabilă de alocarea căilor optice. Modelul propus urmează îndeaproape Concepte fundamentale ale sistemelor distribuite prezentate în prima parte a acestei lucrări. Este exclus în totalitate punctul singular de eșec (single point of failure – SPOF), sistemul putând deservi utilizatorii chiar și în cazul disponibilității parțiale a resurselor de rețea.

Datorită faptului că durata de transmisie a unui mesaj de control este de ordinul sutelor de milisecunde, care este comparabilă cu cea mai optimistă durată de realizare a unei interconectări[16], și din faptul că durata de viață a unui drum optic este cel puțin de ordinul minutelor argumentăm că penalizarea în performanță adusă de tranzacție este insignifiantă.

Două aspecte importante sunt necasare pentru implementare modelului propus:

- O infrastructură de comunicație eficientă pentru o coordonare distribuită a serviciilor care controlează comutatoarele optice, pentru a putea notifica schimbările topologice și pentru alocarea căi optice

O platformă de monitorizare capabilă să asigure suficiente informații pentru a detecta eventualele probleme ce necesită o rerutare a căilor optice

Sistemul prezentat reprezintă o abordare inovatoare, prin folosirea câtorva tehnici care nu se regăsesc în sistemele actuale de alocare a resurselor de rețea. Prin utilizarea capabilităților de comunicație, monitorizare și control din platforma MonALISA, a fost posibilă implementarea unei infrastructuri performante pentru alocare a căilor optice, prin **alocarea resurselor de rețea în paralel**. Datorită faptului că fiecare agent poate satisface cereri în paralel am adresat atât aspectele legate de Scalabilitatea cât și pe cele legate de Toleranța la defecte la un nivel distribuit. Prin mecanismul de pre-alocare a resurselor prin folosirea de **tranzacții distribuite**, au fost adresate într-un mod elegant problemele de consistență ce pot apărea la nivelul topologiei globale. Pe baza capabilităților de monitorizare a platformei MonALISA a fost posibilă implementarea unui mecanism de autocontrol capabil să detecteze și să acționeze rapid în cazul în care apar erori de-a lungul oricărui segment al drumului

[16] Tehnologia folosită pentru realizarea interconectărilor este factorul determinant. În cazul sistemelor MEMS poate dura câteva sute de milisecunde, iar în cazul mecanic până la secunde sau chiar minute.

optic. Odată detectate aceste erori, sistemul este capabil să reruteze suficient de rapid calea optică, astfel încât transferurile de date care folosesc protocolul TCP să poată continua neîntrerupte.

6. Rezultate experimentale

Având în vedere trei obiective majore prezentăm în acest capitol rezultatele experimentale și realizările importante obținute pe baza conceptelor și a tehnologiilor dezvoltate în decursul acestei teze.

6.1. Monitorizarea distribuită a rețelei USLHCNet folosind capabilitățile sistemului MonALISA

MonALISA este platforma aleasă pentru a asigura o monitorizare completă a infrastructurii USLHCNet. *Provocarea majoră* a fost aceea de a asigura o disponibilitate sporită a platformei și a datelor de monitorizare, având în vedere natura distribuită și răspândirea geografică a rețelei.

Toată istoria de monitorizare este prezentată și menținută pe termen lung în cadrul Repository-ului MonALISA [64]. Sistemul monitorizează un ansablu variat de parametrii: statutul și traficul interfețelor de rețea și a circuitelor virtuale, măsurători de accesibilitate (ping), oferind în același timp mecanisme de notificare gen email sau SMS în cazuri de eroare. Din punct de vedere al monitorizării provocarea a fost legată de Eterogenitatea rețelei având în componența sa dispozitive de rețea variate provenind de la diferiți producători: comutatoare optice (Nivel 1 ISO/OSI), switch-uri (Nivel 2), rutere (Nivel 3) și dispozitive hibride cum ar fi Ciena CoreDirector, ce asigură servicii de circuite virtuale. Protocolul SNMP este folosit pentru interogarea dispozitivelor de rețea gen rutere și switch-uri (Foundry, Force10, Cisco), în timp ce TL1 este folosit pentru dispozitivele de rețea Ciena CD/CI și comutatoarele optice Glimmerglass.

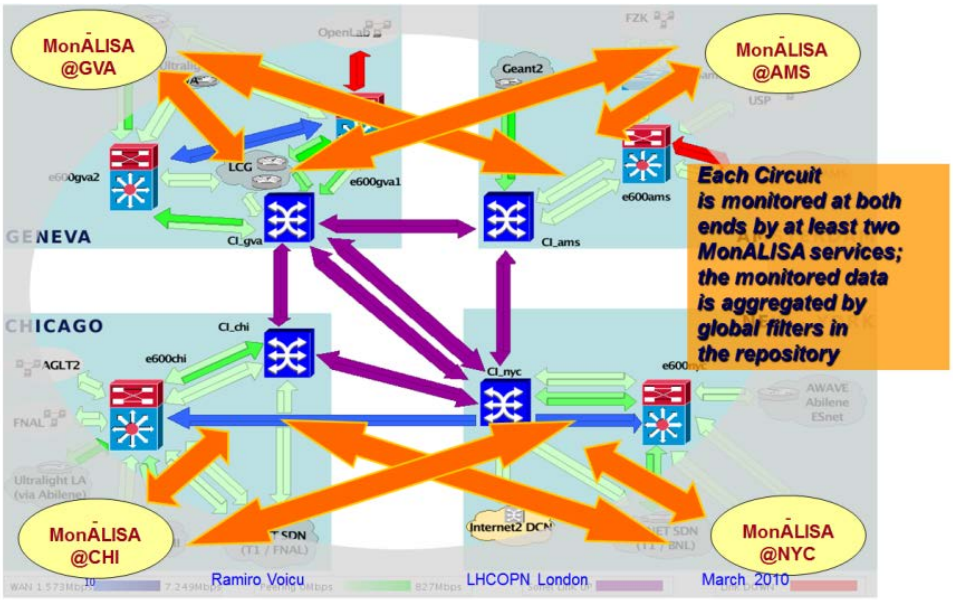
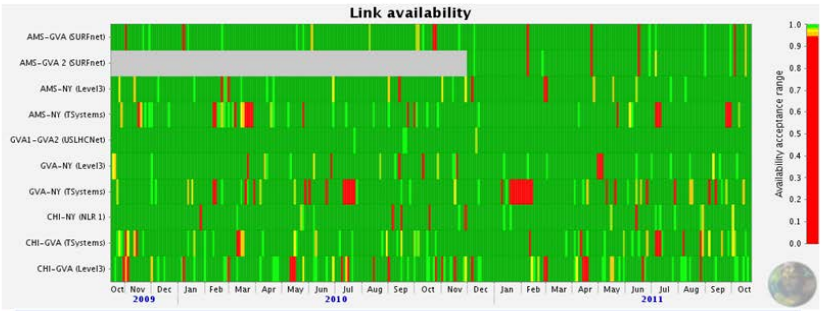


Figura 18 Monitorizarea distribuită a infrastructurii USLHCNet

Costul unui circuit de trans-atlantic diferă foarte mult în funcție de nivelul de disponibilitate agreat cu furnizorii circuitelor de rețea, cunoscut sub denumirea de "Service Level Agreement" (SLA), care cuantifică Robustetea circuitului într-un

interval de timp agreat (de obicei un an). Calculul robusteții circuitului, exprimat formal în Ecuația 1, se face pe baza stării operaționale[17] a circuitului. Valorile tipice vehiculate in cadrul SLA-urilor sunt de peste 95%. Din acest motiv a fost necesar ca și disponibilitatea datelor de monitorizare să fie cel puțin la același nivel cu cel din SLA. Având în vedere răspândirea geografică a punctelor de prezență (PoP) și inevitabilitatea (vezi premise false prezentate în 3.5) instabilităților de rețea, soluția propusă a fost folosirea a cel puțin unui serviciu MonALISA în fiecare PoP, care să monitorizeze resursele locale acelui PoP și, în același timp, și pe cele ale locației cea mai apropiată geografic. Această monitorizare "în cruce" este schițată în Figura 18. Prin această schemă de monitorizare fiecare capăt al unui circuit este monitorizat din două locații diferite, ceea ce duce ca ambele capete ale unui circuit trans-atlantic să fie monitorizate din patru locații diferite.

Repository-ul de MonALISA analizează informația de monitorizare din toate cele patru puncte fiind astfel capabil să determine în mod robust starea operațională a circuitului. Măsurătorile se realizează la fiecare 30 de secunde pentru intreaga arhivă fiind tinută in baza de date a repository-ului. Toate serviciile de monitorizare păstrează datele de monitorizare și local, însă nu mai mult de o lună de zile, repository-ul fiind capabil să ceară datele din urmă în cazul pierderii conectivității cu una din locații.



Statistics

Link name	Data		Monitoring		Link
	Starts	Ends	Availability(%)	Gaps	Availability(%)
AMS-GVA (SURFnet)	15 Oct 2009 14:12	24 Oct 2011 14:11	100%	-	99.38%
AMS-GVA 2 (SURFnet)	02 Dec 2010 17:01	24 Oct 2011 14:11	100%	-	99.75%
AMS-NY (Level3)	15 Oct 2009 14:11	24 Oct 2011 14:11	100%	-	99.64%
AMS-NY (TSystems)	15 Oct 2009 14:11	24 Oct 2011 14:11	100%	-	97.78%
GVA1-GVA2 (USLHCNet)	15 Oct 2009 14:12	24 Oct 2011 14:10	100%	-	99.99%
GVA-NY (Level3)	15 Oct 2009 14:11	24 Oct 2011 14:11	100%	-	98.66%
GVA-NY (TSystems)	15 Oct 2009 14:11	24 Oct 2011 14:11	100%	-	94.12%
CHI-NY (NLR 1)	15 Oct 2009 14:11	24 Oct 2011 14:10	100%	-	99.82%
CHI-GVA (TSystems)	15 Oct 2009 14:11	24 Oct 2011 14:11	100%	-	98.67%
CHI-GVA (Level3)	15 Oct 2009 14:11	24 Oct 2011 14:11	100%		97.60%

No monitoring gaps for 2 years (~30 seconds polling interval)

Figura 19 Robustețea monitorizării distribuite a legăturilor trans-atlantice în USLHCNet[18]

În Figura 19 este prezentat un panel specializat din repository-ul de MonALISA pe baza căruia se poate analiza disponibilitatea circuitelor de rețea, în același timp cu cea a propriei monitorizări. În cazut monitorizării sunt prezentate și intervalele de timp în care lipsesc datele de monitorizare, circuitul de rețea fiind

17 Starea operațională reprezintă de fapt Disponibilitatea circuitului la un moment de timp

18 Cel de-a doua legătură, AMS-GVA 2 (SURFnet) a fost dată în folosință în luna Decembrie 2010 și de aceea nu are istorie până în acel moment.

considerat disponibil pe acel interval. Din acest motiv a fost foarte important ca monitorizarea să fie cât mai robustă cu cât mai puține date de monitorizare lipsă. Se poate observa (penultimele două coloane din Figura 19) că disponibilitatea monitorizării a fost de 100% pe ultimii doi ani, deși rețeaua pe care a fost făcută monitorizarea, a avut multe instabilități (ultima coloană din aceeași figură).

6.2. Rezultate importante ale soluției de transfer rapid de date: Fast Data Transfer (FDT)

Performanța aplicației de transfer de date FDT a fost demostrată în câteva runde ale competiției "Bandwidth Challenge" în timpul conferințelor de SuperComputing. Prezentăm în continuare, în ordine cronologică, cele mai importante rezultate obținute în timpul desfășurării acestei cercetări.

Primul rezultat major a fost obținut în timpul conferinței de SuperComputing din 2006 în timpul competiției Bandwidth Challenge (BWC). Aplicația FDT a fost folosită ca și instrument de transfer, fiind controlată de un agent extern, LISA, pentru orchestrarea transferurilor. Rata maxima de transfer a fost de 17.77 Gbps și a fost obținută între un cluster de calcul din cadrul conferinței și un altul aflat la Caltech. Provocarea din timpul competiției a fost folosirea unei singure linii de 10Gbps în ambele direcții. FDT a asigurat o rată stabilă de transfer folosind 10 perechi de servere fiecare dispunând de 4 discuri SATA configurate ca RAID0 software și o placă de 1Gbps. Rezultatul oficial este prezentat în partea stângă a Figura 20.

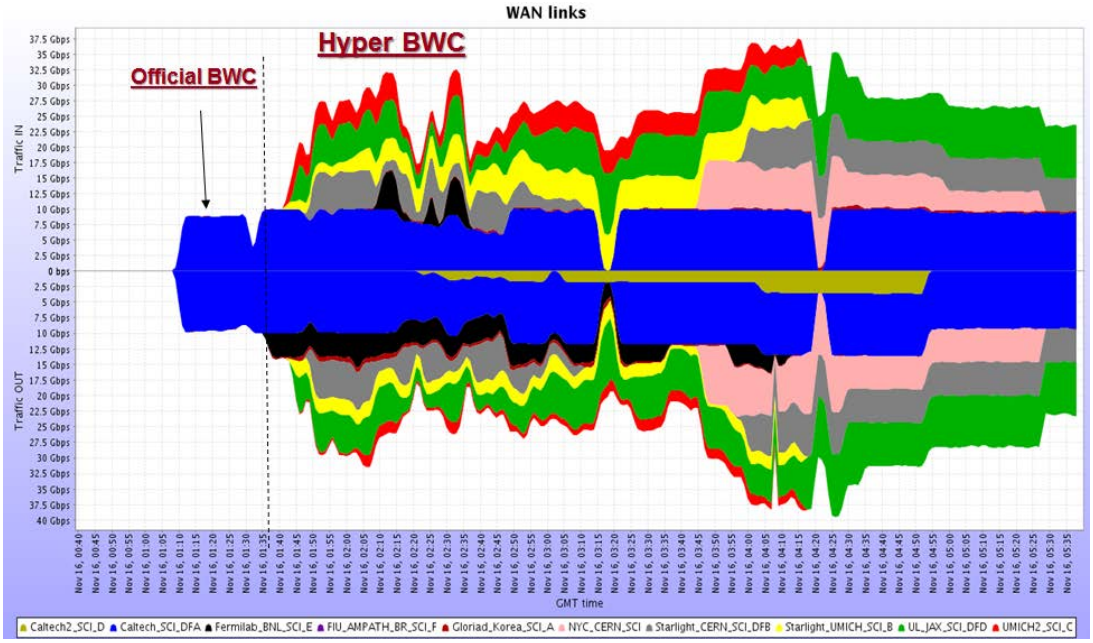


Figura 20 Un singur flux FDT peste WAN în timpul competiției Bandwidth Challenge (stânga); Fluxuri multiple FDT folosind destinații multiple (dreapta) coordonate de agentul LISA

În cadrul conferinței SuperComputing 2008 (SC08), care a avut loc în Austin, Texas, au fost demonstrate performanțele deosebite ale aplicației FDT, inclusiv capabilitățile de control de către servicii externe precum MonALISA și LISA[65].

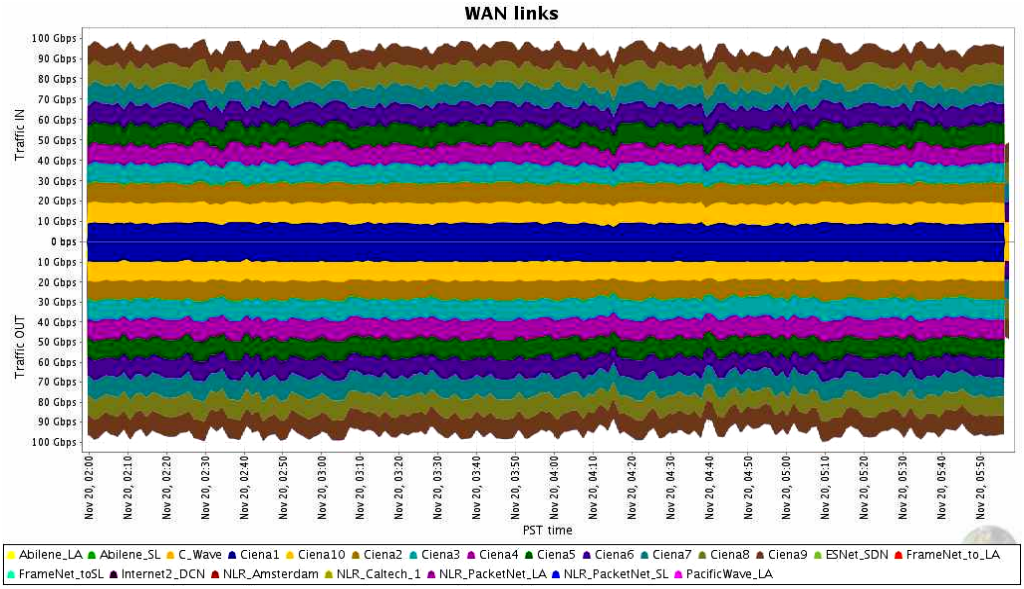


Figura 21: Un eșantion al performanței fluxurilor de date obținute între standurile Caltech și Ciena folosind zece legături de 10 Gbps multiplexate pe o interfață OTU-4 pe o distanță de 80Km de fibră optică. Viteza medie 191 Gbps cu maxime ce ating 199.9 Gbps.

O altă realizare majoră obținută în timpul conferinței SC08 (Figura 21) a fost obținută în colaborare cu Ciena, care tocmai terminase prima implementare a standardului OTU-4 (112 Gbps) cu o capacitate utilă de 100 Gbps (sau 200 Gbps bidirecțional). Pentru aceasta intre standurile Caltech și Ciena a fost folosit un singur cablu optic având 10 perechi de fibre, dispozitivul de rețea Ciena fiind folosit pentru multiplexarea și demultiplexarea celor zece linii de 10 Gbps într-o singură lungime de undă OTU-4 peste o buclă de 80 de km lungime. Datorită performanțelor deosebite ale aplicației FDT și, în același timp, a legăturilor de rețea fără erori, s-a putut atinge viteza maximă posibilă de 199.9 Gbps bidirecțional în cateva minute de la începerea testului, viteza medie de transmisie în timpul celor 12 ore de test fiind de 191 Gbps.

Performanța deosebită a aplicației FDT a fost din nou demonstrată și în timpul conferinței SuperComputing 2011 (SC11) ce a avut loc în Seattle. De această dată transferurile disc-la-disc au depășit 60 Gbps între standul Caltech din cadrul conferinței și Universitatea Victoria, Canada. Utilizând două plăci de rețea de 40 GE (Gigabit Ethernet) PCI Genarația3 și 2 plăci 40GE PCI Generația2 a fost posibil să se atingă viteza de 98Gbps pe o legătură de rețea de 100Gbps. În același timp s-au obținut 88Gbps în direcția opusă, atingând astfel o rată sustinută (Figura 22) de 186 Gbps între cele două locații pe o singură lungime de undă de 100 Gbps (200 Gbps bidirecțional), doborând recordul anterior de 119Gbps obținut în 2009, în timpul aceleași conferințe de SuperComputing. Cantitatea totală de date transferate în timpul celor câteva zile din perioada conferiței a depășit 4 PetaBytes.

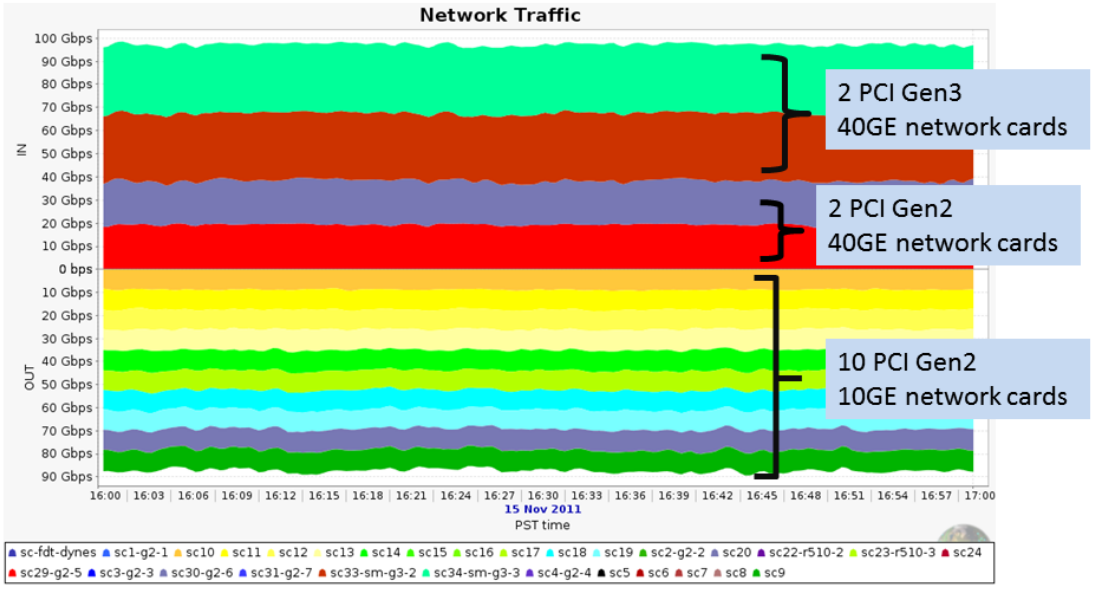


Figura 22 Transferuri paralele memorie-la-memorie cu 4 calculatoare ce primesc date la o viteză agregată de peste 98 Gbps. Fluxul total bidirecțional este 186 Gb/s

Cele două culori din partea superioară a Figura 22 reprezintă traficul primit de cele două calculatoare echipate cu plăci de rețea **PCI Gen3** de 40GE, fiecare din ele reușind să transporte aproximativ ~30Gbps. Următoarele două culori reprezintă traficul primit de celelalte două plăci de rețea **PCI Gen2** 40 GE. Se poate observa limitarea hardware a benzii **PCI**, aplicația FDT fiind capabilă să utilizeze întreaga lărgime de bandă.

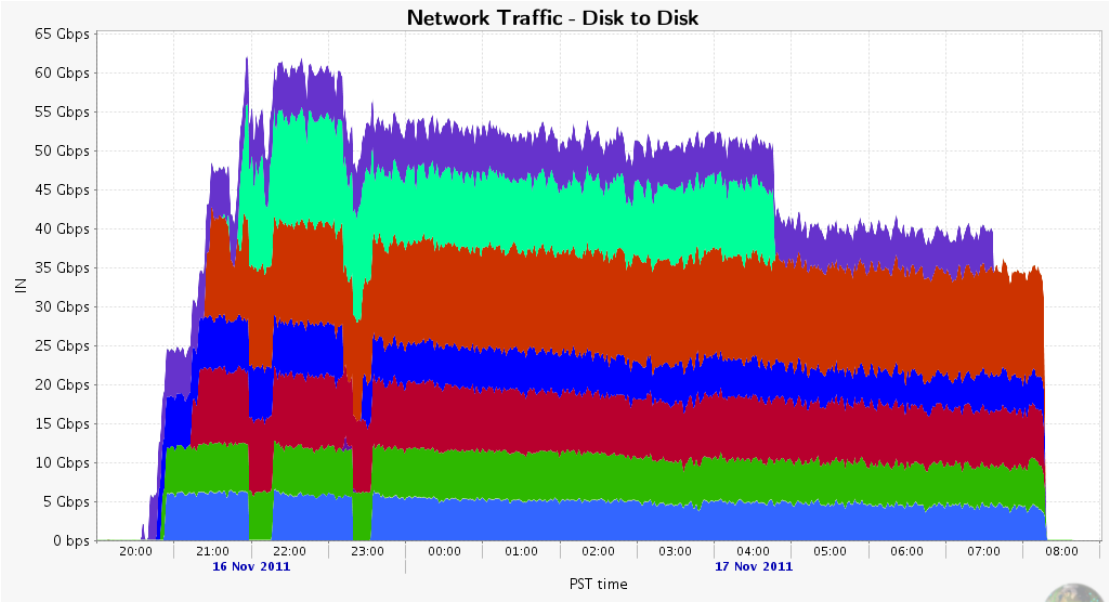


Figura 23 Transferuri disc-la-disc în timpul conferinței SC11

În Figura 23 este prezentată performanța aplicației FDT în transferurile disc-la-disc. Fiecare culoare reprezintă fluxul recepționat de către un singur sistem de calcul. Fluxul a fost menținut stabil pe o perioadă de 11 ore între 16 și 17 Noiembrie 2011 cu mici întreruperi datorită problemelor hardware ale sistemelor de calcul.

6.3. Transferuri de date de mare viteză folosind FDT și sistemul de alocare a căilor optice rețele hibride

Acest subcapitol prezintă rezultate experimentale realizate peste infrastructura de rețea dintre CERN, Geneva și Caltech, Pasadena. Figura 24 prezintă o imagine a clientului grafic MonALISA folosit pentru prezentarea informațiilor de monitorizare și a căilor optice peste topologia de rețea.

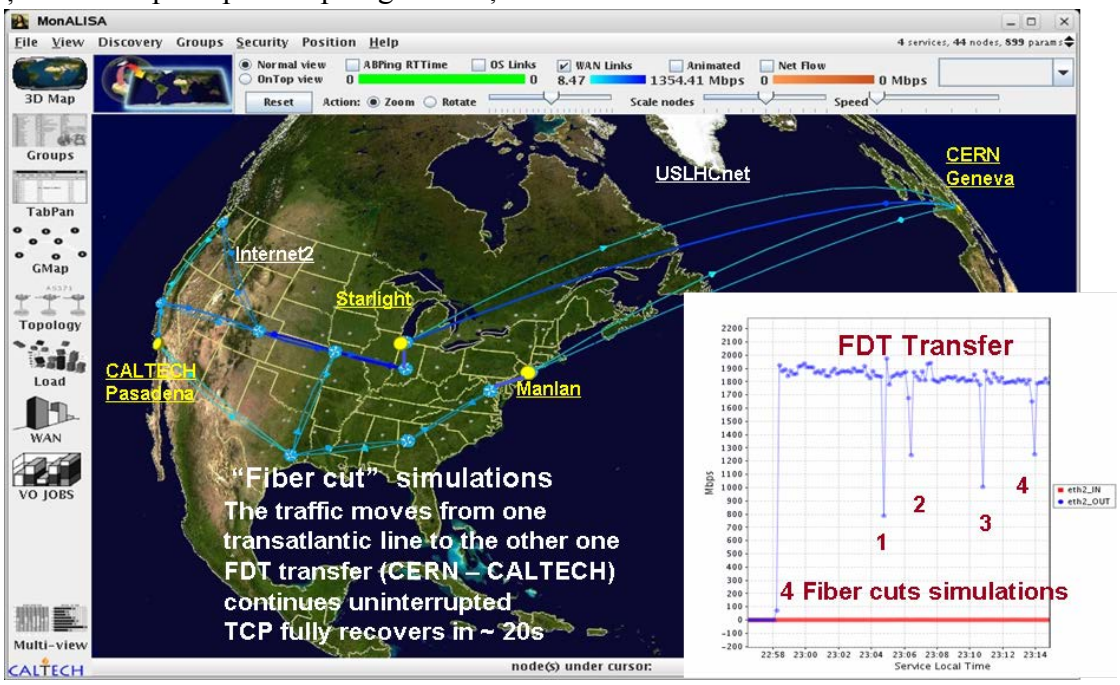


Figura 24: Căi optice folosind agenți OSA în cadrul platformei MonALISA; FDT aplicația pentru transferul datelor peste căile optice

În cadrul experimentului au fost înglobate, într-un mod unitar, toate cele trei obiective majore indentificate în deschiderea acestei lucrări. Prototipul expune și demonstrează practic posibilitatea integrării aplicațiilor sau a sistemelor de transfer a datelor într-un mod eficient cu sistemele de alocarea și planificare a resurselor de rețea împreună cu o soluție de monitorizare și control distribuit. Aplicația de transfer folosită a fost FDT. Sistemul distribuit de agenți pentru controlul comutatoarelor optice OSA a fost folosit pentru alocarea resurselor de rețea. Toți parametrii implicați în acest test precum topologia rețelei, interconectările optice, viteza de transfer, puterea optică pe porturile comutatoarelor, etc au fost monitorizate de către sistemul MonALISA și prezentate în timp real în clientul grafic. Pentru legăturile optice între CERN și Caltech au fost folosite VLAN-uri ce au traversat cele două rețele majore USLHCNet, peste Atlantic, și Internet2 în America. Orice cale optică a fost realizată în aproximativ 500ms. Această durată este explicată de cele două mesaje schimbate în timpul tranzacției distribuite: primul fiind mesajul de pre-alocare, urmat de cel de COMMIT. Timpul de RTT între CERN și Caltech este aproximativ 200ms seconds. Este trivial de observat avantajul abordării paralele și faptul că orice abordare serială ar fi dus la timp de alocare peste secundă la nivelul rețelelor WAN. Imediat ce este stabilită calea optică este pornit transferul de date disc-la-disc folosind aplicația FDT. În timpul experimentului au fost simulate 4 întreruperi consecutive ale fibrei optice (“fiber cuts”) peste Atlantic. Agenții OSA au fost cabili să detecteze și să reruteze

traficul în aproximativ o secundă. Calea optică alternativă a fost realizată suficient de rapid pentru evitarea întreruperii conexiunii TCP, astfel încât transferul datelor a putut continua neîntrerupt (Figura 24, dreapta).

7. Concluzii

Această lucrare prezintă arhitectura precum și detaliile de proiectare și implementare ale unui sistem integrat care încearcă să adreseze problema aplicațiilor data intensive. Cercetarea de față propune o abordare originală a problemei dintr-o perspectivă unitară pornind de la un sistem integrat pentru alocarea resurselor de rețea ce augmentează capabilitățile unei întregi infrastructuri de comunicație, monitorizare și control, integrându-se în același timp într-un mod eficient cu o aplicație performantă pentru transferul datelor.

Un aspect important care merită mentionat este reprezentat de platforma distribuită de monitorizare și control care este capabilă să ofere informații despre toate componentele sistemului: dispozitive de rețea, aplicații și intreaga infrastructură de stocare (calculatoare, sisteme de operare). O platformă de comunicație eficientă a constituit unul din obiectivele modelului formal dezvoltat în capitolul 5. Această platformă a avut la bază aceeași infrastructură de monitorizare și control.

Principalul obiectiv al acestei cercetări a fost un sistem performant pentru transferul datelor care să folosească în mod eficient resursele de rețea. Datorită naturii distribuite a problemei cercetarea de față adresează alte două aspecte pe lângă cel al alocării resurselor de rețea: primul fiind reprezentat de o platformă integrată de monitorizare și control cu capabilități de a rula algoritmi distribuiți, și cel de-al doilea de o aplicație eficientă pentru transferul datelor cu posibilitatea de control dinamic al vitezei de transfer.

Arhitectura distribuită a platformei MonALISA a fost adaptată pentru a putea acomoda implementarea modelui formal de alocarea a resurselor de rețea. Proiectul MonALISA este rezultatul unei colaborări de succes între Universitatea Politehnica din București, California Institute of Technology (CALTECH) și Centrul European pentru Cercetări Nucleare (CERN). În timp sistemul și-a demonstrat robustețea fiind folosit în producție la ora actuală în peste 350 de locații în diferite colțuri ale lumii.

De-a lungul cercetării am notat lipsa unei aplicații de transfer cu capabilități de modificare dinamică a vitezei de transfer. În cadrul acestei teze propunem o nouă aplicație de transfer: Fast Data Transfer – FDT pentru a surmonta această carență. Din rezultatele experimentale prezentate am demonstrat performanțele notabile ale aplicației, folosind mai multe canale de transfer în paralel. În același timp, aplicația asigură capabilități de integrare cu servicii de nivel înalt, de exemplu MonALISA și LISA.

Majoritatea rezultatelor prezentate în această lucrare sunt folosite cu succes în medii aflate în producție la ora actuală. Eficiența modelelor prezentate și implementate de-a lungul acestei cercetări a fost de asemenea demonstrată în medii de cercetare și pre-producție.

7.1. Contribuții majore ale acestei tezei

Principalele contribuții ale acestei teze de doctorat sunt:

- Realizarea unui studiu critic al tehnologiilor și sistemelor actuale de alocare a resurselor de rețea, precum și a platformelor de monitorizare și a aplicațiilor pentru transferul datelor
- Contribuții majore la arhitectura și implementarea platformei MonALISA
- Adresarea unor aspecte importante legate de Concurența, Scalabilitatea și Robustețea întregii platforme MonALISA: îmbunătățirea timpul de răspuns și eficiența comunicării folosind și reutilizând fire de execuție în paralel, decuplând componentele majore prin cozi cu diferite strategii (blocant sau non-blocant "drop") pentru componentelor majore.
- Proiectarea și implementarea conexiuneii de bază TCP folosită ca mecanism de comunicație între toate componentele platformei. Pentru a furniza o transmisie rapidă și în "timp-real" (la nivel de sistem distribuit) a mesajelor, a fost implementată o strategie de "eșuare-rapidă" ("fail-fast") prin intermediul unui mecanism de "*keep-alive*" la nivel de aplicație. În același timp a fost implementat și un mecanism de priorități pentru mesaje care s-a dovedit foarte folositor în mod special în cadrul platformei de agenți
- Tot în cadrul proiectului MonALISA autorul a proiectat un protocol de publicare a datelor de monitorizare din diferite limbaje de programare: ApMon. Am implementat prima variantă a modulului pentru recepționarea mesajelor ApMon și a primului prototip de bibliotecă pentru clienți Java. Protocolul este folosit la ora actuală în multe aplicații aflate în producție în facilități distribuite în toată lumea.
- Proiectarea și implementarea unui mecanism generic pentru actualizarea automată a serviciului de monitorizare MonALISA. Pentru eficientizare este utilizat un sistem cu descărcarea în paralel a resurselor folosind mai multe canale de comunicație (multi-stream) de rețea. Pentru adresarea problemelor sistemelor de fișiere distribuite sunt folosite sume de control criptografice și un mecanism tranzacțional de scriere a acestora pe mediul de stocare.
- Implementarea modulelor de monitoriazare și control pentru două tipuri de comutatoare optice de tip MEMS: Glimmerglass [42] și Calient [43], implementând în același timp un suport generic pentru interogarea dispozitivelor de rețea ce folosesc protocolul TL1
- Bazat pe platforma MonALISA autorul a proiectat și implementat arthitectura distribuită de monitorizare a rețelei trans-atlantice USLHCNet, demonstrând eficiența și robustețea monitorizării: disponibilitate maxima (100%) a datelor.
- Dezvoltarea de module pentru monitorizarea dispozitivelor de rețea Ciena CoreDirector [22] precum și a mecanismelor de alarmă și notificare
- Proiectarea și implementarea unei noi aplicații pentru transferul eficient al datelor peste rețea: Fast Data Transfer – FDT, care asigură o performanță deosebită prin folosirea mai multor canale de date în paralel. Au fost implementate diferite strategii de I/E (I/O) atât blocant (un fir de execuție per canal) cât și non-blocant (canalele sunt deservite de un ansamblu (pool) de fire de execuție). Aplicația suportă în același timp și ajustarea dinamică a vitezei de transfer.

- Propunerea unui model innovator de alocare a resurselor de rețea la Nivelul1 combinând un sistem de tranzacții distribuit și o strategie de pre-alocare a resurselor pentru o eficiență sporită.
- Bazat pe modelul formal propus s-a realizat implementarea un serviciu distribuit bazat pe agenți pentru alocarea căilor optice. Pe baza monitorizării puterii optice pe porturi sistemul este capabil să reruteze calea optică suficient de rapid pentru ca transferurile de date să continue neîntrerupte. Pentru asigurarea consistenței a fost implementat un mecanism de "*lease*" distribuit.
- Integrarea sistemului de alocare cu aplicația de transfer FDT și demonstrarea eficienței în cadrul rețelei trans-atlantice USLHCNet

7.2. Dezvoltări ulterioare

Deși majoritatea dezvoltărilor prezentate în cadrul acestei teze sunt deja folosite în producție, rămân câteva subiecte interesante pentru cercetări viitoare:

- Adaptarea serviciului distribuit de alocare a resurselor de rețea pentru dispozitive de rețea folosind protocolul OpenFlow.
- Investigarea noilor capabilități apărute în platforma Java7 și integrarea lor în cadrul aplicației FDT: strategii de I/O asincron și unificarea abstractizării sistemului de fișiere din FDT cu interfața din Java7.
- O altă direcție insteresantă o constituie integrarea protocoalelor de tip RDMA (ex. iWARP, RoCE) ca mod de transport în FDT
- Adăugarea unui algoritm de rutare a mesajelor între serviciile de Proxy cu scopul de a optimiza comunicația în cazul în care sistemul de monitorizare va trebui să scaleze la zeci de mii de servicii de monitorizare

7.3. Publicații

1. **MonALISA: A Distributed Monitoring Service Architecture**, *H.B. Newman, I.C. Legrand, P.Galvez,* ***R. Voicu****, C. Cirstoiu,* CHEP 2003, La Jola, California, March 2003
2. **Using a Mobile Agent Architecture to Monitor, Control and Optimize the Operation of Distributed Systems**, *I.C.Legrand, H.B. Newman, P. Galvez, C. Cirstoiu,* ***R. Voicu***, ACAT03, Tsukuba, Japan, December 1-5, 2003
3. **MonALISA: An Agent based, Dynamic Service System to Monitor, Control and Optimize Grid based Applications**, *I.C.Legrand, H.B.Newman,* ***R.Voicu****, C.Cirstoiu, C.Grigoras, M.Toarta, C. Dobre,* CHEP 2004, Interlaken, Switzerland, September 2004
4. **MonALISA: A Distributed Monitoring Service Architecture**, *I.C.Legrand, H.B.Newman,* ***R.Voicu****, C.Cirstoiu, C.Grigoras, M.Toarta, C. Dobre, M.Thomas* - Grid 2004 - 5th IEEE/ACM International Workshop on Grid Computing, Pittsburgh, USA, November 2004
5. **A Simulation Study For T0/T1 Data Replication And Production Activities**, *I.C.Legrand, C.M.Dobre,* ***R.Voicu****, C.Stratan, C.Cirstoiu, L.Musat,* 15th International Conference on Control Systems and Computer Science, 25-27 May, 2005
6. **LISA (Local Host Information Service Agent),** *I.C. Legrand, C. Dobre,* ***R. Voicu****, C. Cirstoiu,* Proc. of the 15th International Conference on Control

Systems and Computer Science (CSCS-15), Bucharest, Romania, 2005, pp. 127-130, ISBN: 973-8449-89-8

7. **VINCI: Virtual Intelligent Networks for Computing Infrastructures**, *Legrand, I. C., C. Cirstoiu, S. McKee, H. Newman,* ***R. Voicu****, C. M. Dobre* – Proceedings of CHEP06 – Mumbai, India, 2006
8. **A Distributed Agent Based System to Control and Coordinate Large Scale Data Transfers**, *Ciprian Dobre,* ***Ramiro Voicu****, Adrian Muraru, Iosif C. Legrand,* Proc. of the 16th International Conference on Control Systems and Computer Science (CSCS-16), Bucharest, Romania, May 2007, ISBN: 978-973-718-741-3
9. **A distributed agent system for dynamic optical path provisioning,** ***Voicu, R****., Legrand, I., Newman, H., Dobre, C., Tapus, N.*, IADIS Multi Conference on Computer Science and Information Systems, Lisbon, Portugal, July 2007, ISBN: 978-972-8924-39-3
10. **A Monitoring Architecture for High-Speed Networks in Large Scale Distributed Collaborations**, *A. Costan, C. Dobre, V. Cristea,* ***R. Voicu***, in Proc. of 7th International Symposium on Parallel and Distributed Computing (ISPDC'08), Krakow, Polonia, July 2008, pp. 409 – 416, ISBN: 978-0-7695-3472-5
11. **MonALISA: A Distributed Service System for Monitoring, Control and Global Optimization**, *I. C. Legrand,* ***R. Voicu****, C. Grigoras, C. Cirstoiu, C. Dobre*, ACAT 2008
12. **Framework for high-performance data transfers optimization in large distributed systems**, *C. Cirstoiu,* ***R. Voicu****, N. Tapus* – ISPDC2008 – Krakow, Poland, 1-5 July, 2008
13. **A Distributed Service for on Demand End to End Optical Circuits,** ***R. Voicu,*** *I. Legrand, H. Newman, N. Tapus, C. Dobre,* in Proc. of the 17th International Conference on Control Systems and Computer Science (CSCS-17), CoRR abs/0910.0708, Bucharest, Romania, 2009
14. **Monalisa: An agent based, dynamic service system to monitor, control and optimize distributed systems**, *Legrand, I., Newman, H.,* ***Voicu, R.****, Cirstoiu, C., Grigoras, C., Dobre, C., Muraru, A., Costan, A., Dediu, M., and Stratan, C,* Computer Physics Communications 180, 12 (2009), 2472 – 2498
15. **Monalisa: Monitoring and control of large scale distributed systems**, *Legrand, I.,* ***Voicu, R.****, Cirstoiu, C., Grigoras, C., Betev, L., and Costan, A,* Communications of the ACM 52, 9 (2009), 49–55
16. **Monitoring and control of large systems with MonALISA**, Legrand, I., **Voicu, R**., Cirstoiu, C., Grigoras, C., Betev, L., and Costan, A, ACM Queue 7, 6 (2009), 40–49
17. **Automated agents for management and control of the ALICE Computing Grid**, *C.Grigoras, L. Betev, F. Carminati, I.Legrand and* ***R. Voicu***, 2010 J. Phys.: Conf. Ser. 219 062050
18. **MonALISA-based Grid monitoring and control,** *C. Grigoras,* ***R. Voicu****, N. Tapus, I. Legrand, L. Betev,* European Physical Journal, vol 126, no 1, Jan 2011, DOI 10.1140/epjp/i2011-11009-9

19. **A Monitoring Framework for Large Scale Networks, *R.Voicu,*** *I. Legrand, C. Dobre, IEEE ICCP 2011*

20. **Replication mechanisms for a distributed time series storage and retrieval service**, M. I. Andreica, I. C. Legrand, and R. Voicu, in *Proceedings of the 8th ACM international conference on Autonomic computing*, New York, NY, USA, 2011, pp. 161–162.

21. **Powering physics data transfers with FDT,** *Z. Maxa, B. Ahmed, D. Kcira, I. Legrand, A. Mughal, M. Thomas, and* ***R. Voicu****, Journal of Physics: Conference Series, vol. 331, no. 5, p. 052014, Dec. 2011.*

22. **Workflow management in large distributed systems,** *I. Legrand, H. Newman,* ***R. Voicu****, C. Dobre, and C. Grigoras, Journal of Physics: Conference Series, vol. 331, no. 7, p. 072022, Dec. 2011.*

23. **The Dynamics of Network Topology, *R. Voicu,*** *I. Legrand, H. Newman, A. Barczyk, C. Grigoras, and C. Dobre, Journal of Physics: Conference Series, vol. 331, no. 5, p. 052033, Dec. 2011.*

24. **The DYNES Instrument: A Description and Overview**,J. Zurawski, R. Ball, A. Barczyk, M. Binkley, J. Boote, E. Boyd, A. Brown, R. Brown, T. Lehman, S. McKee, B. Meekhof, A. Mughal, H. Newman, S. Rozsa, P. Sheldon, A. Tackett, R. Voicu, S. Wolff, and X. Yang, *J. Phys.: Conf. Ser.*, vol. 396, no. 4, p. 042065, Dec. 2012.

25. **Disk-to-Disk network transfers at 100 Gb/s**, *A. Barczyk, I. Gable, M. Hay, C. Leavett-Brown, I. Legrand, K. Lewall, S. McKee, D. McWilliam, A. Mughal, H. Newman, S. Rozsa, Y. Savard, R. J. Sobie, T. Tam, and R. Voicu*," *J. Phys.: Conf. Ser.*, vol. 396, no. 4, p. 042006, Dec. 2012.

26. **Efficient LHC Data Distribution across 100Gbps Networks**, *H. Newman, A. Barczyk, A. Mughal, S. Rozsa, R. Voicu, I. Legrand, S. Lo, D. Kcira, R. Sobie, I. Gable, C. Leavett-Brown, Y. Savard, T. Tam, M. Hay, S. Mckee, R. Hocket, B. Meekhof, and S. Timoteo*, in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*, 2012, pp. 1594–1599.

8. Bibliografie

[1] *CERN Advanced STORage manager (CASTOR)*, [Online]. Available: http://castor.web.cern.ch/, [Accessed: Nov -2011].

[2] Ann Chervenak, Ian Foster, Carl Kesselman, Charles Salisbury, Steven Tuecke, *The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets*

[3] George Coulouris, Jean Dollimore, Tim Kindberg, *Distributed Systems: Concepts and design (fourth edition)*, Addison-Wesley, 2005

[4] A. S. Tanenbaum and M. V. Steen, *Distributed Systems: Principles and Paradigms*, 2nd ed. Prentice Hall, 2006.

[5] I. Foster, C. Kesselman, and S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, International Journal of High Performance Computing Applications, vol. 15, no. 3, pp. 200 -222, Fall 2001.

[6] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, *Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility* Future Generation Computer Systems, vol. 25, no. 6, pp. 599-616, Jun. 2009.

[7] I. Foster, Yong Zhao, I. Raicu, and S. Lu, *Cloud Computing and Grid Computing 360-Degree Compared* in Grid Computing Environments Workshop, 2008. GCE '08, 2008, pp. 1-10.

[8] The Advanced Network Systems Architecture (ANSA), *Reference Manual*, Castle Hill, Cambridge, UK, 1989

[9] B. Schneier, *Applied Cryptography: Protocols, Algorithms, 2nd Edition*, 1995.

[10] D. A. Patterson, G. Gibson, and R. H. Katz, *A case for redundant arrays of inexpensive disks (RAID)* in Proceedings of the 1988 ACM SIGMOD international conference on Management of data, New York, NY, USA, 1988, pp. 109–116.

[11] *ESnet: Energy Sciences Network*, [Online]. Available: http://es.net/ , [Accessed: Nov-2011].

[12] *GÉANT pan-European research network*, [Online]. Available: http://www.geant.net, [Accessed: Nov-2011].

[13] *Internet2 networking consortium*, [Online]. Available: http://www.internet2.edu, [Accessed: Nov-2011].

[14] *NLR: National LambdaRail*, [Online]. Available: http://nlr.net, [Accessed: Nov-2011].

[15] *SURFnet – The dutch academic network*, [Online]. Available: http://www.surfnet.nl, [Accessed: Nov-2011].

[16] *LHCOPN - LHC Optical Private Network*, [Online]. Available: http://www.cern.ch/lhcopn, [Accessed: Nov-2011].

[17] *MONARC project*, [Online]. Available: http://cern.ch/monarc, [Accessed: Nov-2011].

[18] *USLHCNet – High speed TransAtlantic network for the LHC Community*, [Online]. Available: http://uslhcnet.org/, [Accessed: Nov-2011].

[19] *FNAL: Fermilab - Fermi National Accelerator Laboratory*, [Online]. Available: http://www.fnal.gov/, [Accessed: Nov-2011].

[20] *BNL: Brookhaven National Laboratory*, [Online]. Available: http://www.bnl.gov/, [Accessed: Nov-2011].

[21] *Ciena Corporation*, [Online]. Available: http://www.ciena.com, [Accessed: Nov-2011].

[22] *Ciena CoreDirector*, [Online]. Available: http://www.ciena.com/products/core-director/, [Accessed: Nov-2011].

[23] ITU-T, *G.707/Y.1322 , Network node interface for the synchronous digital hierarchy (SDH)*, 2007

[24] ITU-T, *G.783, Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks*, 2006

[25] ITU-T, *G.7042/Y.1305 - Link capacity adjustment scheme (LCAS) for virtual concatenated signals*, 2006

[26] *OpenDRAC: Open Dynamic Resource Allocation Controller*, [Online]. Available: https://www.opendrac.org/, [Accessed: Nov-2011].

[27] Chin Guok, D. Robertson, M. Thompson, J. Lee, B. Tierney, and W. Johnston, "Intra and Interdomain Circuit Provisioning Using the OSCARS Reservation System," in 3rd International Conference on Broadband Communications, Networks and Systems, BROADNETS 2006, pp. 1-8.

[28] T. Lehman, J. Sobieski, and B. Jabbari, *DRAGON: a framework for service provisioning in heterogeneous grid networks*, IEEE Communications Magazine, vol. 44, no. 3, pp. 84- 90, Mar. 2006.

[29] X. Yang, T. Lehman, C. Tracy, J. Sobieski, S. Gong, P. Torab, and B. Jabbari, *Policy-based resource management and service provisioning in GMPLS networks*, in INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, 2006, pp. 1–12.

[30] B. Tierney, R. Aydt, D. Gunter, W. Smith, M. Swany, V. Taylor, and R. Wolski, *A Grid Monitoring Architecture*, 2002.

[31] J. M. Schopf, L. Pearlman, N. Miller, C. Kesselman, I. Foster, M. D'Arcy, and A. Chervenak, *Monitoring the grid with the Globus Toolkit MDS4*, Journal of Physics: Conference Series, vol. 46, pp. 521-525, Sep. 2006.

[32] I. Foster, *A Globus Toolkit Primer*, [Online]. Available: http://www.globus.org/primer, [Accessed: Nov-2011].

[33] A. Cooke, A. Gray, L. Ma, W. Nutt, J. Magowan, M. Oevers, P. Taylor, R. Byrom, L. Field, S. Hicks, and others, *R-GMA: An information integration system for grid monitoring*, On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, pp. 462–481, 2003.

[34] H. Gibbins and R. Buyya, *Gridscape II: A customisable and pluggable grid monitoring portal and its integration with Google Maps*, in Grid and Cooperative Computing, 2006. GCC 2006. Fifth International Conference, 2006, pp. 257–265.

[35] S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G. L. Rubini, G. Tortone, and M. C. Vistoli, *GridICE: a monitoring service for Grid systems*, Future Generation Computer Systems, vol. 21, no. 4, pp. 559-571, Apr. 2005.

[36] M. L. Massie, B. N. Chun, and D. E. Culler, *The ganglia distributed monitoring system: design, implementation, and experience*, Parallel Computing, vol. 30, no. 7, pp. 817-840, Jul. 2004.

[37] Nagios monitoring system, [Online]. Available: http://www.nagios.org/ , [Accessed: Nov-2011].

[38] I. Mandrichenko, W. Allcock, and T. Perelmutov, *GridFTP v2 protocol description*, GGF Document Series GFD, vol. 47, 2005.

[39] *File Transfer Service (FTS)*, [Online]. Available: http://cern.ch/egee-jra1-dm/FTS/default.htm , [Accessed: Nov-2011].

[40] *gLite - Lightweight Middleware for Grid Computing*, [Online]. Available: http://glite.cern.ch/ , [Accessed: Nov-2011].

[41] H. B. Newman, I. C. Legrand, P. Galvez, **R. Voicu**, and C. Cirstoiu, *MonALISA : A Distributed Monitoring Service Architecture*, arXiv:cs/0306096, Jun. 2003.

[42] *Glimmerglass Optical Cyber Solutions*, [Online]. Available: http://www.glimmerglass.com, [Accessed: Nov-2011].

[43] *Calient Technologies*, [Online]. Available: http://www.calient.net, [Accessed: Nov-2011].

[44] Telcordia Technologies, *Operations Application Messages - Language For Operations Application Messages*, GR-831 Generic Requirements document, Transaction Language 1 (TL1), 1996

[45] Peter Deutsch, James Gosling, *The Eight Fallacies of Distributed Computing*, [Online]. Available: http://blogs.oracle.com/jag/resource/Fallacies.html [Accessed: Nov-2011].

[46] FDT: Fast Data Transfer project, [Online]. Available: http://fdt.cern.ch, Caltech [Accessed: Nov-2011].

[47] I.C. Legrand, C. Dobre, **R. Voicu**, C. Cirstoiu , LISA (Local Host Information Service Agent), Proc. of the 15th International Conference on Control Systems and Computer Science (CSCS-15), Bucharest, Romania, 2005, pp. 127-130, ISBN: 973-8449-89-8

[48] Z. Maxa, B. Ahmed, D. Kcira, I. Legrand, A. Mughal, M. Thomas, and **R. Voicu**, *Powering physics data transfers with FDT*, Journal of Physics: Conference Series, vol. 331, no. 5, p. 052014, Dec. 2011.

[49] H. Ohnishi, T. Okada, and K. Noguchi, *Flow control schemes and delay/loss tradeoff in ATM networks*, IEEE Journal on Selected Areas in Communications, vol. 6, no. 9, pp. 1609-1616, Dec. 1988.

[50] S. Shenker and J. Wroclawski, *General characterization parameters for integrated service network elements*, RFC 2215, September, 1997.

[51] B. Bollobás, *Modern graph theory*, Springer, 1998.

[52] R. Diestel, *Graph Theory*, Fourth edition, Springer, 2010 (2005, 2000, 1997)

[53] T. H. Cormen, *Introduction to algorithms*, MIT Press, 2001.

[54] R. Bellman. *On a routing problem*, Quarterly of Applied Mathematics, 16(1):87–90, 1958.

[55] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische Mathematik, vol. 1, pp. 269-271, Dec. 1959.

[56] J. Moy, *OSPF Version 2*, [Online]. Available: http://tools.ietf.org/html/rfc2328 [Accessed: Nov-2011].

[57] H. Gredler and W. Goralski, *The complete IS-IS routing protocol*, Springer, 2005.

[58] **R. Voicu**, I. Legrand, H. Newman, C. Dobre, and N. Tapus, "A distributed agent system for dynamic optical path provisioning," Proceedings of Intelligent Systems and Agents (ISA), 2007.

[59] *IEEE Standard 802.3-2008*, IEEE, [Online]. Available: http://standards.ieee.org/getieee802/download/802.3-2008_section1.pdf, [Accessed: Nov-2011].

[60] B. Hajek and T. van Loon, *Decentralized dynamic control of a multiaccess broadcast channel*, IEEE Transactions on Automatic Control, vol. 27, no. 3, pp. 559- 569, Jun. 1982.

[61] P. R. Srikanta Kumar and L. Merakos, *Distributed control of broadcast channels with acknowledgement feedback: Stability and performance*, in The 23rd IEEE Conference on Decision and Control, 1984, 1984, vol. 23, pp. 1143-1148.

[62] J. Goodman, A. G. Greenberg, N. Madras, and P. March, *Stability of binary exponential backoff*, J. ACM, vol. 35, no. 3, pp. 579–602, Jun. 1988.

[63] **R. Voicu**, I. Legrand, H. Newman, N. Tapus, and C. Dobre, "A distributed service for on demand end to end optical circuits," arXiv:1106.5570, Jun. 2011.

[64] MonALISA USLHCNet Repository, [Online]. Available: http://repository.uslhcnet.org, [Accessed: Nov-2011].

[65] CERN Courier, High-energy physics team sets data-transfer world records, [Online]. Available: http://cerncourier.com/cws/article/cern/37317 [Accessed: Nov-2011].