UNIVERSITEIT GENT

Academiejaar 2013–2014

Faculteit Ingenieurswetenschappen en Architectuur
Valentin Vaerwyckweg 1 – 9000 Gent

Automatische testbed monitoring voor toekomstig internetonderzoek

Masterproef voorgedragen tot het behalen van het diploma van
Master in de industriële wetenschappen: informatica

Andreas DE LILLE

Promotoren: *Geert VAN HOOGENBEMT*
Piet DEMEESTER

Begeleiders: *Brecht VERMEULEN*
Wim VAN DE MEERSSCHE
Thijs WALCARIUS
Wim VANDENBERGHE

Abstract

Door de huidige verschuiving naar cloudgebaseerde technologiën zal het belang van netwerkprotocollen en van de beschikbaarheid van netwerken alleen maar toenemen. Om deze verschuiving vlot te laten verlopen is er meer en meer onderzoek nodig naar netwerktechnologiën. Om onderzoekers en onderzoekscentra beter te laten samenwerken is FIRE (Future Internet Research and Experimentation) opgestart. Dit project zal trachten om de uitwisseling van ideeën tussen onderzoekers te verhogen. Daarnaast wordt, door de ontwikkeling van een gemeenschappelijke architectuur, het delen van testfaciliteiten makkelijker gemaakt. Dit wordt verwezenlijkt via een java tool, jFed (java Federation). jFed wordt gebruikt om testfaciliteiten binnen FIRE aan te sturen. Deze manier van werken zorgt ervoor dat men snel proefopstellingen kan maken op verschillende testfaciliteiten binnen FIRE. Toch zijn er enkele nadelen verbonden aan deze manier van werken. Het is voor een onderzoeker soms erg moeilijk om te bepalen of een bepaald gedrag in zijn experiment te wijten is aan eigen ontwikkelingen, of aan het falen van een testbed.

Deze masterproef verhelpt dit probleem door de invoer van een automatisch monitoringsproces. Een monitoringsservice zal instaan voor de monitoring van de testfaciliteiten. De informatie die hierbij verzameld wordt, wordt via een monitoringsAPI ter beschikking gesteld. Deze API (application programming interface) vormt een stevige basis waarop andere toepassingen kunnen gemaakt worden.

Abstract

Due to the new cloud-based technologies, network protocols and network reachability are now more important than ever. To make this change quick and clean, we need more and more network research. FIRE (Future Internet Research and Experimentation) is an European project created to improve the network and internet experimentation. FIRE is the opportunity to jointly develop potentially disruptive innovations for the future of the internet. It is about collaborative research and sharing test facilities. Researchers working for FIRE will now work closer together, sharing ideas. FIRE is also used to share test facilities from all over the world, so researchers have access to many different test facilities.

To make handling of these different testbeds easier, jFed was created. jFed is a java tool with the purpose of controlling testbeds. Unfortunately, the current situation has a major downside in that it is very hard for researchers to determine if a certain behavior is caused by the testconfiguration or by the test facility.

This thesis will try to solve the aforementioned problem by creating an automated the testbed monitoring system. A monitoringAPI will then share this information. Doing so, will provide future tools easy access to this information.

Lijst van afkortingen

AM	Aggregate Manager
API	application programming interface
FED4FIRE	Federation 4 FIRE
FIRE	Future Internet Research and Experimentation
FLS	First Level Support
GENI	Global Environment for Network Innovations
jFed	java Federation, tool voor het aansturen van de Federation architectuur
MA	Management Authority
OML	Outline Markup Language
SA	Slice Authority
SFA	slice-based federation architecture
XML	Extensible Markup Language
XML	extensible markup language

Lijst van figuren

Inhoudsopgave

Inleiding

Het gebruik van netwerken en het Internet om computers en allehande randapperatuur te verbinden zal in de toekomst alleen maar stijgen. Het is dan ook van groot belang dat onderzoek op dit gebied vlot en correct verloopt. Daarom is het aangewezen dat onderzoekers samenwerken om zo ideeën en nieuwe technologiën te delen. Daarnaast moeten er ook testfaciliteiten zijn om deze nieuwe technologiën te testen. FIRE (Future Internet Research and Experimentation) is een Europees onderzoeksproject dat zich op deze doelen richt.

Om de configuratie en werking van de verschillende testbeds gelijk te trekken, is binnen FIRE de federation architectuur ontworpen. De invoering hiervan zit in het project FED4FIRE (Federation for FIRE). De federation architectuur die in deze masterproef behandeld wordt is de SFA 2.0 (Slice-federation-architecture). SFA deelt een testbed op in meerdere niveau's, het onderste niveau zijn de componenten. Een component is een computer, router, ... van een testbed. Omdat testbeds vaak door meerdere onderzoekers tegelijk gebruikt worden, worden componenten vaak gemultiplexed. Zo krijgt elke onderzoeker bijvoorbeeld 'een stuk' van een computer, een sliver. Meerdere slivers vormen samen een slice. Een slice is dus een verzameling componenten waarop een experiment gedraaid wordt. Daarnaast kan men de componenten ook groeperen in aggregaten. Een aggregaat is een verzameling componenten die valt onder eenzelfde beheerder. Een goed voorbeeld van een aggregate is de virtual wall. De virtual wall is een testfaciliteit van iMinds, gebruikt om netwerken te simuleren. De virtual wall is een aggregate, in die zin dat alle componenten beheerd worden door iMinds.

Het beheer van al deze verschillende aggregates is geen sinecure. Om dit beheer te vereenvoudigen heeft iMinds binnen het Fed4FIRE project een monitoringsservice gemaakt. Deze service is echter snel ontwikkeld en is niet geschikt voor uitbereidingen. Deze masterproef bestaat uit 2 grote delen. Het eerste deel is een nieuwe monitoringsservice maken die de testbeds controleert. De informatie die deze service verzameld zal beschikbaar gemaakt worden door een monitoringsAPI.

FIRE werkt nog samen met een gelijkaardig project, GENI. GENI (Global Environment for Network Innovations) is een Amerikaans project met gelijkaardige doelstellingen als FIRE. Hierdoor is de samenwerking tussen beide projecten zeer hoog. Het tweede deel van de

masterproef is de integratie van monitoringsAPI in het GENI project.

Hoofdstuk 1 vertelt eerst meer over de achtergrond van het project. Hier wordt ook kort de opdracht uitgelegd.

Hoofdstuk 2 gaat dieper in op de werking van de monitoringsservice die er was. Hierbij komt ook de probleemstelling uitgebreider aan bod.

Hoofdstuk 1

Situering

In dit hoofdstuk wordt het achterliggende kader van de masterproef geschetst. Daarnaast wordt ook de opdracht uitgewerkt. De opdracht bestaat uit 2 grote delen enerzijds een monitoringsservice maken om testbeds te controleren. De informatie van deze monitoringsservice wordt via de monitoringsAPI beschikbaar gesteld aan de buitenwereld. Anderzijds wordt de monitoringsAPI geïntegreerd in een groter amerikaans framework.

1.1 Situering

Deze masterproef is een onderdeel van een groter Europees onderzoeksproject genaamd FIRE (Future Internet Research and Experimentation). FIRE is een project dat erop gericht is om onderzoek naar toekomstige internet- en netwerktechnologiën te vergemakkelijken door onderzoekscentra te laten samenwerken(FIRE, 2014). FIRE heeft twee grote doelen. Enerzijds de samenwerking tussen verschillende onderzoekscentra te verbeteren, anderzijds het delen van testfacilities makkelijker te maken.

Het eerste doel is de samenwerking tussen verschillende onderzoekscentra te verbeteren. Onderzoekers binnen eenzelfde vakgebied komen vaak gelijkaardige problemen tegen. FIRE vermijdt dat men telkens het wiel opnieuw uitvindt, door deze onderzoekers makkelijker en meer te laten samenwerken. Hierdoor worden oplossingen en ideeën meer gedeeld, zodat de ontwikkeling sneller kan verlopen.

Het tweede doel is het delen van testfaciliteiten makkelijker te maken. Door FIRE krijgt een onderzoeker van een onderzoekscentrum toegang tot testfaciliteiten van andere onderzoekscentra binnen FIRE. Testfaciliteit is een algemene term die duidt op zowel hardware als software dat gebruikt wordt om testen te verrichten. Een testbed is een concreet voorbeeld van een testfaciliteit. Een testbed kan gezien worden als een server of een verzameling servers waarop men testen kan laten lopen. Zo kan er op een testbed bijvoorbeeld een server en een

aantal cliënts gesimuleerd worden. Deze worden verbonden met een aantal tussenliggende routers. Vervolgens wordt een videostream opgestart. Op deze videostream kan men storing introduceren door pakketten te droppen. Deze storing zal ervoor zorgen dat het beeld aan de client-side hapert. Er kunnen technieken ingebouwd worden aan client-side om deze storing op te vangen. Zo kan er overgeschakeld worden naar een lagere kwaliteit indien blijkt dat de beschikbare bandbreedte onvoldoende is. Testen van degelijke technieken verloopt dan ook aan de hand van testbeds.

Het probleem dat zich hier stelt is dat elk testbed op zijn eigen manier werkt. Onderzoekers hebben nu wel toegang tot andere testbeds, maar moeten voor elk testbed eerst de nieuwe configuratie leren. Verschillende testbeds laten samenwerken is, op deze manier, geen sinecure. Om deze configuratie gelijk te maken heeft men de federation architectuur ingevoerd. Dit is onderdeel van FED4FIRE (Federation 4 FIRE) . De federation architectuur die hier gebruikt wordt is SFA 2.0. Deze architectuur heeft als doel om de configuratie en werking van alle testbeds gelijk te maken.

SFA 2.0 werkt met 3 niveau voor verantwoordelijkheid. De bovenste is de MA (Management Authority), deze is verantwoordelijk voor de stabiliteit van een heel testfaciliteit. Een SA (slice authority) is verantwoordelijk voor een of meerdere slices. De laatste is een gebruiker, bijvoorbeeld een onderzoeker die een experiment wil uitvoeren op een testbed.

Daarnaast maakt SFA ook gebruik van een specifieke naamgeving. Een component is een primaire block van de architectuur, bijvoorbeeld een computer of een router. Meerdere componenten worden vervolgens gegroepeerd in aggregaten. Alle componenten van een aggregaat vallen onder dezelfde MA (Management Authority). Elke aggregaat wordt gecontrolleerd door een AM (aggregate manager). De AM beheert de allocatie van de verschillende experimenten op de aggregaat.

De MA (Management Authority) bepaald vervolgens hoe de resources verdeeld worden. Indien een component gemultiplexed wordt spreekt van men slivers. De gebruiker heeft dan een sliver van de component tot zijn beschikking. Meerdere slivers worden gegroepeerd tot aan slice. Een experiment wordt ook uitgevoerd binnen een slice.

jFed werd door iMinds ontwikkeld(iMinds, 2014b) met als doel de SFA architectuur eenvoudig aan te sturen. iMinds is een onafhankelijk onderzoekscentrum dat opgericht werd door de Vlaamse overheid(iMinds, 2014c). iMinds is leider van het FED4FIRE project(iMinds, 2014a). Met behulp van jFed kunnen onderzoekers snel en eenvoudig netwerken simuleren en testen uitvoeren. Toch is er nog ruimte voor verbetering in jFed. Een van de voornaamste problemen is dat een onderzoeker niet weet of het testbed dat hij gebruikt betrouwbaar is. Bepalen of een vreemd gedrag in een experiment te wijten is aan eigen ontwikkelingen of aan het falen van een testbed, kan op deze manier zeer tijdrovend zijn.

Om dit probleem op te lossen heeft iMinds een monitoringssysteem uitgebouwd(Brecht Vermeulen, 2014). Dit monitoringssysteem werkt, maar is door de snelle ontwikkeling niet voorzien op uitbereidingen. Deze masterproef zal enerzijds een monitoringsservice maken die deze testbeds in de gaten houdt. Anderzijds zal deze informatie via een monitoringsAPI beschikbaar gemaakt worden voor onderzoekers. Het is de bedoeling dat deze API een stevige basis vormt waarop andere applicaties kunnen gebouwd worden. Merk op dat monitoring op 3 niveau's mogelijk is: component,slice en aggregate. De monitoringsservice die hier besproken wordt, richt zich op de bovenste laag. De monitor zal dus kijken of een testbed online is en hoeveel vrije resources er beschikbaar zijn.

FIRE reikt echter verder dan Europa alleen, zo zijn er ook overeenkomsten met onderzoeksprojecten buiten Europa. FIRE werkt samen met GENI (Global Environment for Network Innovations). Geni is een Amerikaans onderzoeksproject gericht om aggregaten te bundelen en beschikbaar te stellen aan onderzoekers(GENI, 2014a). FIRE en GENI werken ook aan de integratie van beide projecten(Piet Demeester, 2013) (geni, 2014).

GENI heeft zelf een gedistribueerde monitoringservice uitgebouwd(GENI, 2014b). Deze service maakt gebruik van datastores(GENI, 2014). Een datastore houdt de monitoring informatie van een testbed of aggregate bij. Deze informatie wordt dan opgehaald door een collector. De webservice van FED4FIRE zou ook als een datastore bekeken worden. Op deze manier kan de monitoringsAPI geïntegreerd worden in een groter monitoringsframework. Dit zal als tweede deel van de masterproef behandeld worden.

Hoofdstuk 2

Analyse en probleemstelling

Zoals besproken in het vorige hoofdstuk, zal deze masterproef een monitoringsAPI maken in opdracht van iMinds, een onderzoekscentrum. De monitoringsAPI heeft als doel de resultaten van een achterliggende monitoringsservice aan te bieden. De monitorService zal op zijn beurt alle aggregaten (testopstellingen) binnen FIRE (Future Internet Research and Experimentation) een project voor de verbetering van onderzoek naar internet en netwerken, in de gaten houden. Dit hoofdstuk zal de werking van de monitoringsservice beschrijven die er was bij de aanvang van de masterproef. Daarna zal er dieper ingegaan worden op de probleemstelling.

2.1 Werking monitoringservice bij aanvang

iMinds (onderzoekscentrum waar de masterproef uitgewerkt wordt en tevens leidinggevend in het FIRE project), heeft een monitoringsservice gemaakt die al enige tijd draait (Brecht Vermeulen, 2014). Deze monitoringsservice is ruimer dan strict testbeds monitoren, ook het monitoren van experimenten wordt door deze service afgehandeld(Brecht Vermeulen, 2014).

Deze monitoringsservice bestaat uit verschillende componenten. De eerste component is de Facility monitoring, deze monitoring wordt gebruikt bij de FLS (first level support). De first level support heeft als doel om de basis zaken te monitoring. De voornaamste test is de pingtest die kijkt of een testbed nog online is. Het testbed zelf mag deze monitoring invullen, naar gelang zijn wens. De enige vereiste hier is dat de data exporteer baar is naar een OML-stream . OML (Outline Markup Language) is een XML (Extensible Markup Language) taal die gebruikt wordt om objecten te encoderen.

De tweede component is de infrastucture monitoring. Deze component is gericht om componenten binnen een experiment. De verzamelde data bevat o.a. aantal verstuurde pakketten, aantal verloren pakketten, cpu-load, ...

Een derde component is de OML measurement library, deze bibliotheek laat het toe dat een onderzoeker zijn eigen monitoring framework gebruikt om de metingen van zijn experiment te doen.

Deze masterproef richt zich voornamelijk op de Facility monitoring. De monitoringsservice waarnaar verwezen wordt komt dus overeen met de Facility Monitoring. De andere componenten worden verder buiten beschouwing gelaten. Data van de Facility Monitoring kan gelezen worden op een website.

Deze monitoringsservice (Facility Monitoring) is opgedeeld in een aantal stukken. Het eerste stuk is de FLS-monitor (First Level Support). Deze is beschikbaar op https://flsmonitor.fed4fire.eu/,zie ook Figuur 2.1 . Deze service heeft het doel actuele informatie weer te geven over de status van het testbed.

Fed4FIRE First Level Support Monitoring

Testbed Name	Ping latency (ms)	GetVersion Status	Free Resources	Internal testbed monitoring status	Last check internal status
BonFIRE	31.17	N/A	N/A	ok	2014-03-25 11:16:09+01
Fuseco	16.83	ok	1	ok	2014-03-25 11:16:03+01
Koren	284.47	N/A	N/A	N/A	N/A
NETMODE	67.33	ok	20	ok	2014-03-25 11:13:22+01
NITOS Broker	73.03	ok	38	ok	2014-03-25 11:15:01+01
NITOS SFAWrap	31.29	ok	65	N/A	N/A
Norbit	N/A	N/A	N/A	ok	2014-03-25 11:10:29+01
Ofelia (Bristol island)	11.89	N/A	N/A	ok	2014-03-25 11:10:02+01
Ofelia (i2CAT island)	N/A	N/A	N/A	ok	2014-03-25 11:10:02+01
Planetlab Europe	32.11	ok	285	ok	2014-03-25 11:15:02+01
SmartSantander	53.82	ok	0	ok	2014-03-25 11:10:01+01
Virtual Wall	0.21	ok	23	ok	2014-03-25 11:14:39+01
w-iLab.t 2	20.33	ok	68	ok	2014-03-25 11:14:58+01

Figuur 2.1: testbed monitoring

Figuur 2.1 geeft een beeld van de monitoringssite. De laatste 2 kolommen zijn van minder belang. De eerste kolom geeft de naam van het testbed weer. Daarnaast wordt het resultaat van de laatste ping test getoond. De volgend 2 kolommen bevatten het resultaat van respectievelijk de getVersiontest en de free resources test.GetVersion geeft aan of de AM (aggregate manager) nog werkt terwijl de kolom free resources aangeeft hoeveel resources er nog beschikbaar zijn. De vorm van deze testen is relatief eenvoudig, waardoor een eenvoudige databank voldoende was om de data te bij te houden.

Het tweede deel van de monitoringsservice, nightly login testing, bevat echter complexere testen. Deze testen worden typisch 1 of 2 keer per dag uitgevoerd. Deze testen zijn echter diepgaander dan de FLS-monitor. Zo wordt er bij een logintest getest of het aanmelden op een testbed mogelijk is. Een andere test die ook uitgevoerd wordt is de stichingtest, deze zal kijken of het mogelijk is om een netwerk op te zetten tussen verschillende testbeds. Zie Figuur 2.2

jFed Automated Scenario Tests

Options: Show code tag

Category	Test Name	Last Test Start Time (CET)	Last Test Duration	Last Partial Success	Last Full Success	Time since last Failure	Last Log	History
login	Fuseco	2014-03-25 09:17:47	14 seconds	FAILURE	FAILURE		log	history
login	InstaGeni BBN	2014-03-25 09:18:01	3 minutes and 6 seconds	SUCCESS	SUCCESS	9 days and 13 hours	log	history
login	InstaGeni Clemson	2014-03-25 09:21:08	45 seconds	WARN	WARN		log	history
login	InstaGeni GATech	2014-03-25 09:21:54	2 minutes and 24 seconds	SUCCESS	SUCCESS	27 days and 14 hours	log	history
login	InstaGeni Illinois	2014-03-25 09:24:20	3 minutes and 27 seconds	SUCCESS	SUCCESS	5 months and 11 days	log	history
login	InstaGeni Kettering	2014-03-25 09:27:50	10 seconds	FAILURE	FAILURE		log	history
login	InstaGeni MAX	2014-03-25 09:28:00	2 minutes and 55 seconds	SUCCESS	SUCCESS	2 months and 9 days	log	history

Figuur 2.2: resultaten van de stiching test

Zoals zichtbaar in Figuur 2.3, is het ook mogelijk om de geschiedenis van deze testen op te vragen.

jFed Automated Scenario Tests

Options: Show code tag

Category	Test Name	Test Start Time	Test Duration	Partial Success	Full Success	Log
stitching	Stitching vwall1 - vwall2	2014-04-21 03:50:33	6 seconds	FAILURE	FAILURE	log
stitching	Stitching vwall1 - vwall2	2014-04-21 04:07:34	2 minutes and 27 seconds	FAILURE	FAILURE	log
stitching	Stitching vwall1 - vwall2	2014-04-21 04:18:35	9 minutes and 46 seconds	SUCCESS	FAILURE	log
stitching	Stitching vwall1 - vwall2	2014-04-21 10:14:35	9 minutes and 26 seconds	SUCCESS	FAILURE	log
stitching	Stitching vwall1 - vwall2	2014-04-21 20:13:10	20 minutes	FAILURE	FAILURE	log
stitching	Stitching vwall1 - vwall2	2014-04-21 20:33:29	3 minutes and 13 seconds	SUCCESS	FAILURE	log
stitching	Stitching vwall1 - vwall2	2014-04-21 21:55:49	3 minutes and 19 seconds	SUCCESS	FAILURE	log
stitching	Stitching vwall1 - vwall2	2014-04-21 23:12:05	3 minutes and 17 seconds	SUCCESS	SUCCESS	log
stitching	Stitching vwall1 - vwall2	2014-04-22 10:13:13	3 minutes and 17 seconds	SUCCESS	SUCCESS	log
stitching	Stitching vwall1 - vwall2	2014-04-22 20:22:31	3 minutes and 14 seconds	SUCCESS	SUCCESS	log
stitching	Stitching vwall1 - vwall2	2014-04-23 10:20:53	3 minutes and 16 seconds	SUCCESS	SUCCESS	log

Figuur 2.3: geschiedenis van resultaten

Echter door de komst van complexere testen volstond deze databank niet meer. Een voorbeeld van zo'n complexe test is een stichting test. Deze test zal een verbinding maken tussen verschillende aggregates. Opzetten van dergelijke verbindingen bestaat uit meerdere stappen. Aangezien elke stap kan falen, is het resultaat van deze test niet langer gelukt of mislukt, maar een verzameling van de tussenresultaten. Hiervoor moet de databank aangepast worden. De oude monitoringsservice slaat echter maar 2 tussenresultaten op. Dit wordt getoond in Figuur **??** .Hierdoor is het minder makkelijk om snel een overzicht te krijgen van de tussenresultaten. De geschiedenis van de stichingtesten wordt wel bijgehouden. Deze kan ook opgevraagd worden, zie Figuur**??** .

2.2 Probleemstelling

Het voornaamste probleem is dat de resultaten niet toegankelijk zijn. De informatie is enkel zichtbaar op de monitoringssite.

Door de complexe opbouw van de automatedtesten, is de oude databank niet meer efficiënt. Een eerste probleem is dat alle argumenten in kolommen zitten, waardoor ze vast liggen. Vermits elke automatedtest andere argumenten vereist, is het niet mogelijk om deze configuratie eenvoudig op te slaan in een databank. Op deze manier moet voor elke test een aparte databank, of minstens tabel gemaakt worden. Deze tabel is dan specifiek voorzien voor een soort testen.

Het ander probleem is het variabele aantal resultaten. Een stitching test bestaat uit een ander aantal stappen dan een login test. Al deze tussenresultaten werden samengenomen in 2 tussenresultaten. De andere resultaten zijn beschikbaar in de logfiles. Hierdoor is het zeer moeilijk om alle subresultaten weer te geven op de monitorsite, zie Figuur **??**.

De eerste taak van de masterproef is het maken van een monitoringsservice. Deze service zal instaan voor de opslag, configuratie en uitvoering van de testen. Om deze taken te vervullen zal een nieuwe databank gemaakt worden. Deze databank zal de configuratie bijhouden gecombineerd met de resultaten.Op deze manier worden de huidige propertyfiles, die de oude configuratie bevatten vervangen.

2.3 Wat kan anders

2.3.1 Samenvoegen databases

Een eerste mogelijkheid tot verbetering, is het bijhouden van de laatste resultaten. Deze resultaten zitten in een databank, die voor elk testbed een lijn bevat. Nieuwe waarde overschrijven die lijn. Geschiedenis van een test opvragen is niet mogelijk. Aanpassen van deze resultaten gebeurt door shell scripts. Deze worden periodiek uitgevoerd en lezen een configuratiefile in. Indien het testbed waarop de test uitgevoerd wordt al in de databank zit, is het id vermeld in de configuratiefile. Indien er geen id staat, zal het script zelf een nieuwe lijn aanmaken en vervolgens de id wegschrijven naar de file. Eenmaal uitgevoerd, geeft de test een resultaat terug in de vorm van een xml-file (Extensible Markup Language) die door geïnstaleerde commando's geparset wordt. Vervolgens wordt de data weggeschreven naar de databank.

Door de nieuwe resultaten toe te voegen in plaats van ze te overschrijven, is het wel mogelijk om de geschiedenis van een test op te vragen. Ook de configuratiefiles kunnen opgenomen worden in de databank. Toevoegen van een nieuwe test is dan eenmalig een entry toevoegen aan de database. Deze entry bevat dan het commando en de benodigde parameters.

Een tweede punt is de opdeling flsmonitoring en flsmonitoring-international. Deze 2 databanken zijn gelijk en kunnen gemakkelijk ondergebracht worden in een database. Dit kan door een extra punt toe te voegen dat weergeeft in welke categorie het testbed zich bevind. Een andere mogelijkheid is de internationale testbeds hardcoderen op de monitoring site.

Door de databanken scenario's en flsmonitoring in een databank onder te brengen, kunnen we het beheer vereenvoudigen. Beide gegevens zijn eenmaal resultaten van testen. Deze meer generische aanpak zal er toe leiden dat het eenvoudiger is om testen te definiëren.

2.3.2 Structuur database

De huidige opbouw van de databanken is niet flexibel genoeg. Als we de databanken zouden samenvoegen is er een flexibelere structuur nodig. Het moet mogelijk zijn dat een tests meerdere argumenten meekrijgt. Dit aantal is verschillend per type tests, werken met extra kolommen is dus niet aangeraden. Ook geven sommige testen meerdere resultaten terug. Dit aantal is ook variabel en afhankelijk van het type test dat gebruikt wordt. De structuur van de databank wordt later in deze scriptie uitgewerkt.

2.3.3 Webservice uitbouwen

Doordat de webinterfaces rechtstreeks contact maken met de databank, is er minder overhead. Als de databank echter ook vanuit jFed bereikbaar moet zijn, is het beter om een webservice tussen te voegen. Dit vermijd duplicatie van code. Deze service zal complexere zaken zoals filteren van resultaten voorzien. Daardoor kunnen zowel jFed als de webinterfaces en eventuele toekomstige toepassingen met een eenvoudige call de informatie ophalen en moet niet elke mogelijke applicatie terug zelf de filtering voorzien. Dit kan evenwel een overhead veroorzaken, die geminimaliseerd kan worden door meerdere calls te bundelen. Bijvoorbeeld door met een call resultaten voor verschillende testbeds op te vragen.

Eenmaal de webservice uitgebouwd is, kunnen de monitoringsites gemaakt worden die deze webservice contacteren om informatie op te halen. Er kan ook integratie in jFed voorzien worden. Dit kan als onderdeel van de masterproef, of als een latere uitbreiding voor jFed.

2.4 Besluit

Er zullen twee grote aanpassingen doorgevoerd worden. De eerste eerst is het samenvoegen van de databanken. Om het beheer van de data te vereenvoudigen zou 1 grote databank het werk van de 3 kleinere databanken over nemen. Die databank moet het toelaten om testen met een variabel aantal argumenten en resultaten toe te voegen. Deze databank zou de configuratie van de testen en de resultaten bijhouden.

Bibliografie

i. Brecht Vermeulen (2014). D2.4 - second federation architecture. URL http://www.fed4fire.eu/fileadmin/documents/public_deliverables/D2-4_Second_federation_architecture_Fed4FIRE_318389.pdf.

FIRE (2014). What is fire. URL http://www.ict-fire.eu/getting-started/what-is-fire.html.

GENI (2014a). About geni. URL https://www.geni.net/?page_id=2.

GENI (2014b). Geni operational monitoring project. URL http://groups.geni.net/geni/wiki/OperationalMonitoring.

geni (2014). Geni related projects. URL http://groups.geni.net/geni/wiki/RelatedProjects#GENIRelatedProjects.

GENI (2014). Overview of operational monitoring. URL http://groups.geni.net/geni/wiki/OperationalMonitoring/Overview.

iMinds (2014a). Federation for fire. URL http://www.iminds.be/nl/projecten/2014/03/07/fed4fire.

iMinds (2014b). jfed : Java based framework to support sfa testbed federation client tools. URL http://jfed.iminds.be/.

iMinds (2014c). Over iminds. URL http://www.iminds.be/nl/over-ons.

i. Piet Demeester (2013). Common roadmap of fire test facilities 4th version. URL http://www.ict-fire.eu/fileadmin/publications/deliverables/D3_7-pu-Common_roadmap_of_FIRE_test_facilities_%E2%80%93_Fourth_version_v1.0.pdf.