

Uitgebreid voorstel Masterproef Informatica

Titel van het project : Geavanceerde gebruikersinterface voor toekomstig internet onderzoek en automatische monitoring.

Datum indienen : 17/12/2013

Naam student : Andreas De Lille

Interne promotor: Geert Van hoogenbemt

In samenwerking met : iMinds

Algemene informatie voor extern bedrijf:

Naam van het bedrijf : Iminds

Adres: Gaston Crommenlaan 8, 9050 Gent

- Is dit de 1^e masterproef in het bedrijf in samenwerking met onze opleiding? Nee
- Is er in het bedrijf inhoudelijke en technische begeleiding mogelijk ? Ja
- Kan de student in het tweede semester (februari-mei) 3 dagen per week in het bedrijf/onderzoekscentrum aanwezig zijn om te werken aan de masterproef? Ja

Begeleiding :

Externe promotoren – andere begeleiders :

Piet Demeester, Wim Vandenberghe, Brecht Vermeulen, Wim Van de Meerssche

Bespreking door de werkgroep (niet invullen bij indienen van een voorstel)

Beslissing:
goedgekeurd - herwerken tegen ... / ...
Minimale uitbreidingen:
Opmerkingen:
Advies van collega's:

Bestaande situatie

iMinds is een onafhankelijk onderzoekscentrum dat opgericht werd door de Vlaamse overheid. Het is voornamelijk bezig met onderzoek omtrent ICT-innovatie. Bij dit onderzoek wordt veelvuldig gebruik gemaakt van testbeds. Een testbed is een verzameling van nodes waarmee netwerkopstellingen kunnen gesimuleerd worden.

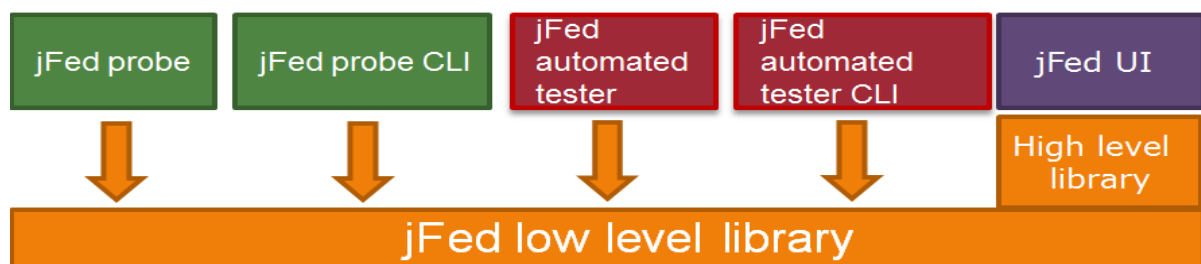
Een goed voorbeeld van een testbed is de Virtual Wall. De Virtual Wall is ontworpen voor het experimenteren met nieuwe netwerkoplossingen voor de volgende generatie van het Internet. De kracht van deze infrastructuur is dat ze de mogelijkheid biedt om op een eenvoudige wijze specifieke netwerkopstellingen op te zetten volgens de noden van het onderzoek.

Een voorbeeld waarvoor dit testbed wordt gebruikt is dat men een server en een aantal cliënts definieert. 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 cliënt-side hapert. Er kunnen technieken ingebouwd worden aan cliënt-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.

Onderzoekers bij iMinds hebben niet alleen toegang tot hun Virtual Wall, maar door samenwerkingsakkoorden met diverse onderzoekscentra hebben ze ook wereldwijd toegang tot andere en soms erg verschillende testbeds. Deze zijn elk apart ontwikkeld. Hierdoor maakt elk testbed gebruik van een eigen interface. Deze interface wordt aangeboden door middel van een federation API. Een federation API is een interface die alle mogelijk functionaliteiten van een specifiek testbed aanbiedt. Hij staat in voor de communicatie tussen een programma en het testbed. Onderzoekers die met meerdere testbeds werken, moeten dan ook de werking van elk testbed apart bestuderen alvorens ermee aan de slag te kunnen. Ook rechtstreekse verbindingen leggen tussen verschillende testbeds wordt hierdoor moeilijker.

Daarom werd in het Europese onderzoeksproject Fed4FIRE beslist om een aantal federation API's vast te leggen die gelijk moeten zijn op elk testbed. Dit maakt het mogelijk om tools te ontwikkelen die bruikbaar zijn op alle testbeds verenigd in dit project. Hiervoor werd de tool jFed ontwikkeld door iMinds¹. jFed biedt een wrapper aan rond de specifieke federation API's die in dat project worden toegepast. Daarenboven voorziet jFed in verschillende applicaties om enerzijds te kunnen testen of testbeds deze API's correct ondersteunen, en anderzijds om als onderzoeker op een eenvoudige wijze een experiment te kunnen opzetten.

De kernmodule waar alle andere modules gebruik van maken is de "jFed library". De voornaamste functie van deze library is om de communicatie met de testbeds te vereenvoudigen. De hiervoor bruikbare testbeds implementeren enkele gestandaardiseerde API's, die de benodigde functionaliteiten aanbieden. De jFed library implementeert de client kant van deze API's, en voorziet Java interfaces voor de API-calls. De gebruiker zal hierdoor niet geconfronteerd worden met deze details. Zo zal de library o.a. de achterliggend SSL-connectie en de bijhorende certificaten en private keys beheren. Verder zullen ook de nodige omzettingen gebeuren. Daarnaast bevat deze library vele hulpklassen en methoden om de API's aan te spreken.



¹ <http://jfed.iminds.be>

Een goed voorbeeld van een API die door jFed wordt geïmplementeerd is de Aggregate Manager API, versie 2 en 3. Een voorbeeld van een call, is de "CreateSliver" call in de AMv2 API. De AMv2 API specificeert dat alle calls gedaan worden over een SSL verbinding met client-side certificate. Verder specificeert de API de exacte argumenten die vereist zijn en het formaat van het resultaat. De jFed library biedt een Java methode "createSliver" in de klasse "AggregateManager2" aan. De argumenten worden opgegeven in een Java formaat. Zo zal bijvoorbeeld een datum-argument als een Java Date-object doorgegeven worden. De jFed library zal de datum naar het RFC3339 formaat omzetten voor het naar de server verstuurd wordt. Ook het resultaat van de call wordt verwerkt door de jFed library terug omgezet.

Steunend op deze library zijn enkele tools zoals jFed probe en jFed compliance tester gebouwd. Deze hebben als doel het valideren of testbeds de federation API's correct ondersteunen. Tenslotte is er de jFed GUI tool. Deze tool maakt het mogelijk om als eindgebruiker een experiment op een intuïtieve manier op te zetten. De jFed GUI heeft momenteel echter een relatief complexe interface die gebruiksvriendelijker gemaakt kan worden.

Probleemstelling

Er is echter nog ruimte voor verbetering in jFed. Zo worden er op geregelde tijdstippen verschillende testen automatisch uitgevoerd op testbeds. Deze testen geven o.a. weer of testbeds online zijn en of het aanmelden via een SSH-verbinding gelukt is. Deze resultaten worden momenteel nog redelijk ad hoc bijgehouden. Men maakt geen gebruik van een databank, maar van bash scripts om de resultaten te archiveren. Resultaten op deze manier bijhouden, is uiteraard inefficiënt. Bovendien staat deze informatie momenteel enkel op een enkele specifieke website vermeld.

Een onderzoeker wordt dus niet via zijn primaire gebruikersinterface op de hoogte gebracht van eventuele storingen met de testbeds opgenomen in zijn experiment. Dit maakt het voor onderzoekers soms moeilijk om te identificeren of een bepaald gedrag in hun experiment te wijten is aan de eigen ontwikkelingen, of aan de testbeds zelf. Hierdoor kan veel tijd onnodig verloren gaan tijdens de uitvoering en de analyse van het experiment. Een betere oplossing zou zijn dat bij het selecteren van een testbed de betrouwbaarheid weergegeven wordt. Zo kan een betrouwbaar testbed gekozen worden, om dergelijke situaties zoveel mogelijk te vermijden.

Daarnaast zou het ook zeer handig zijn als de jFed GUI niet alleen de functionaliteit zou bieden om een experimentele opstelling te gaan ontwerpen en uitrollen, maar ook in real time zou weergeven of alle onderliggende componenten wel 'up and running' zijn. Deze masterproef zal zich concentreren op het integreren van zulke monitoring informatie in de jFed UI voor onderzoekers, zowel op het niveau van de testbeds zelf, als op het niveau van de specifieke resources eigen aan het experiment.

Wat in deze masterproef uitgevoerd zal worden

Deze masterproef zal in de context van het Europese onderzoeksproject Fed4FIRE de tool jFed verder uitwerken. Het voornaamste doel is dat de onderzoeker via zijn primaire gebruikersinterface op de hoogte gebracht wordt van eventuele storingen. Deze storingen kan men opdelen in 2 categorieën, enerzijds op het niveau van testbeds en anderzijds op het niveau van componenten eigen aan een experiment.

In een eerste deel zal de monitoring op niveau van testbeds geïntegreerd worden in de gebruikersinterface. Om de huidige ad-hoc oplossing efficiënt te maken, moet een nieuwe backend geschreven worden. Deze zal zich bezighouden met het aggregeren van de monitoringinformatie. De toegang tot deze backend zal voorzien worden via een API. Hiervoor moet naast het opstellen van een backend en een bijhorende API ook de gebruikersinterface van jFed aangepast worden. Deze is geschreven in JavaFX. Aangezien dit onderwerp niet behandeld werd tijdens in de opleiding, moet er tijd voorzien worden om JavaFX te leren.

Deze ontwikkeling kan verder uitgebouwd worden tot het niveau van de verschillende componenten in een testbed. Zo is het niet alleen belangrijk om de betrouwbaarheid van een testbed te bekijken, maar ook om na te gaan of afzonderlijke componenten die deel uit maken van een test 'up and running' zijn. Een real-time weergave van deze resultaten zou weergegeven kunnen worden in de jFed GUI voor onderzoekers. Het tweede deel van de masterproef zal zich hierop toespitsen. Merk op dat beide delen in principe gelijktijdig ontwikkeld kunnen worden.

Technologieën die aan bod komen

De eerste technologie die aan bod komt is het uitbouwen van een database om de debug/monitoring-informatie op te slaan. Dit kan gebeuren met MySQL. Vervolgens wordt toegang tot die database voorzien met een API. Deze API zal vervolgens verbinding maken met de server. Deze verbinding kan bv. voorzien worden via een Java-rest service. De technologie die hiervoor gebruikt zal worden, moet nog gekozen worden. Dit zal vooraf gegaan worden door een vergelijkende studie.

Om de monitoring informatie te integreren in de jFed GUI voor onderzoekers zal JavaFX gebruikt worden. JavaFX is de opvolger van Java Swing. JavaFX laat onder andere toe om een grafische gebruikersinterface op te bouwen met behulp van een XML file.

Mogelijke uitbreidingen en opties :

Een mogelijke uitbreiding is de backend voor OMF schrijven. De OMF-backend is een module van de jFed-tool dit nog niet geïmplementeerd is. OMF staat voor Orbit Management Framework en wordt gebruikt om tests te definiëren en uit te voeren op de nodes. Ook ophalen van testresultaten vormt een onderdeel van OMF.