

FAULT INJECTION FOR WEB SERVICES

Martin Žouželka

Master Degree Programme (2), FIT BUT

E-mail: xzouze00@stud.fit.vutbr.cz

Supervised by: Marek Rychlý

E-mail: rychly@fit.vutbr.cz

Abstract: Web services are now considered as a technology with great potential for the future because of its multiplatform properties and growing popularity. Nevertheless, the assurance of reliability and robustness of these systems is becoming more and more complicated matter. For this reason, the new testing methods, such as fault injection and specialized tools for automated validation of web services appear. This paper discusses the design of a SWIFI tool, whose main function will be to monitor and test the most common types of web services according to setup criteria.

Keywords: web services, fault injection, SWIFI tools

1 ÚVOD

S příchodem internetu a World Wide Web technologií se lidem otevřelo mnoho možností jak získávat, zveřejňovat a zpracovávat informace téměř odkudkoliv na světě. Zmíněné technologie se postupem času stávaly čím dál složitějšími, přičemž byl kladen stále větší důraz na jejich spolehlivost a interoperabilitu. Vzhledem k těmto rostoucím požadavkům bylo nutné vyvíjet techniky a standardy, které budou zaručovat jejich kvalitu a bezpečnost.

Do nejdynamičtěji rostoucí skupiny webových technologií lze zařadit webové služby, které dnes tvoří důležité součásti informačních systémů, jenž společně dokáží nezávisle komunikovat, plnit stanovené business cíle a vytvářet tak hmotný či nehmotný zisk.

Vzhledem k provázanosti jednotlivých systémů využívajících výhod webových služeb je kladen nemalý důraz na jejich testování a bezporuchovost. Technika zvaná jako softwarová injekce poruch přináší řadu ověřovacích možností a stavů systému, kterých při běžném testování lze jen těžko dosáhnout, což, jak se již v minulosti několikrát ukázalo, může být pro správnou funkčnost systému kritickým aspektem [3].

2 INJEKCE PORUCH V PROSTŘEDÍ WEBOVÝCH SLUŽEB

Metody pro vkládání poruch jakožto součást rozšířeného testování softwarových aplikací jsou stále součástí výzkumu, během kterého vznikají nástroje pro automatickou injekci chyb a analýzu zpětné vazby ze strany testované služby. Zmíněné nástroje jsou obecně označovány zkratkou SWIFI (*Software Implemented Fault Injection*). SWIFI nástroje umožňují zaměřit testování na specifickou část systému bez toho, aniž by byly poruchy vloženy do zbytku systému. Lze je rozdělit do dvou kategorií podle techniky vkládání chyb:

- **Vkládání poruch v době kompilace** je technikou, kde zdrojový kód programu je pozměněn za účelem vložení chyby. Hlavní nevýhodou této techniky je potřeba zdrojového kódu k testování.
- **Vkládání poruch za běhu** je odlišnou technikou, která používá softwarové spouštěče pro injekci poruch přímo do běžícího programu. Tyto spouštěče mohou být reprezentovány časem či

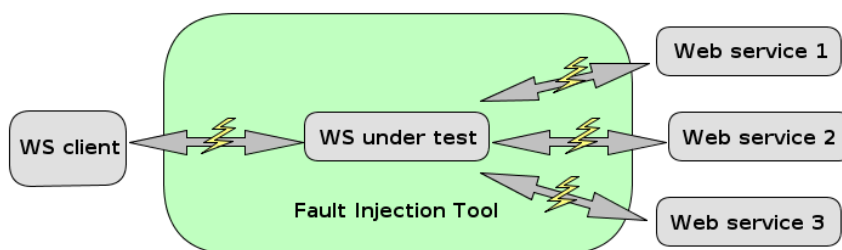
jinou událostí, která má za následek změnu stavu testované komponenty [2].

3 NÁVRH SWIFI NÁSTROJE

Projekt byl během návrhu pojmenován zkratkou FIWS (*Fault Injector for Web Services*), který si klade za cíl vytvoření nástroje s grafickým uživatelským rozhraním pro monitorování komunikace a možností aplikace uživatelských testů spojených s nejpoužívanějšími typy webových služeb (XML-RPC, SOAP, REST). Návrh byl zčásti inspirován vědeckým projektem WSInject, jehož dokumentaci lze nalézt zde [1].

3.1 SCHÉMA KOMUNIKACE

Vzhledem k výše zmíněným technikám pro injekci poruch lze navrhovanou aplikaci zařadit mezi SWIFI nástroje pro vkládání poruch za běhu, kdy během komunikace mezi službami dochází k narušování obsahu zpráv a tímto způsobem vytváření cílených závad, jejichž následky jsou monitorovány.



Obrázek 1: Schéma komunikace z pohledu testované webové služby

Na obrázku 1 je znázorněno schéma zasílání zpráv mezi testovanou webovou službou a jejím okolím. Veškerá komunikace bude probíhat přes proxy server, na kterém bude operovat program FIWS monitorující příchozí a odchozí zprávy a v závislosti na nastavených pravidlech bude vkládat různé typy poruch na různá místa v XML dokumentu. Je třeba nakonfigurovat testovanou webovou službu tak, aby její komunikační rozhraní odpovídalo proxy IP adrese a portu nastavenému v FIWS.

3.2 MODEL PODMÍNEK A PORUCH

Nyní je třeba určit, za jakých podmínek a jaké typy chyb během testování vkládat. Následuje neúplný seznam typů podmínek a poruch, které bývají nejčastěji používány během testování webových služeb.

- **Podmínky:**

- **Podmínka vázaná na obsah zprávy** - je splněna, pokud daná zpráva obsahuje zadaný řetězec.
- **Podmínka vázaná na URI zprávy** - je splněna, pokud daná zpráva obsahuje v URI zadaný řetězec.
- **Podmínka vázaná na cíl zprávy** - je splněna, pokud je zpráva směřována zadanému cíli.

- **Poruchy na úrovni rozhraní:**

- **Nahrazení řetězce** - nahradí všechny výskyty jednoho řetězce za jiný řetězec. Pracuje čistě na úrovni řetězců bez ohledu na syntaxi XML.
- **Nahrazení řetězce pomocí XPath** - má stejnou funkci jako předchozí typ poruchy, avšak s tím rozdílem, že jsou nahrazeny všechny řetězce nacházející se na místě určeném pomocí XPath. Poruchu lze použít k cílené změně daného elementu nebo jeho atributu.
- **Znásobení výskytu řetězce pomocí XPath** - tento typ poruchy pracuje s celým podstro- mem XML dokumentu určeným pomocí XPath, který je následně na daném místě n-krát znásoben.
- **Nahrazení parametru pomocí XPath WSDL** - nahradí ve zprávě parametr operace za- daným řetězcem. Parametr je určen pomocí XPath cesty v odpovídajícím WSDL popisu služby.
- **Odstranění obsahu zprávy** - odstraní veškerý obsah zprávy (doručení HTTP zprávy bez obsahu).

- **Poruchy na úrovni komunikace:**

- **Zpoždění zprávy** - zpozdí doručení zprávy o zadaný počet milisekund.
- **Předčasné uzavření spojení** - uzavře spojení mezi webovou službou a proxy serverem.

Vytváření testů pro webové služby bude probíhat tím způsobem, že uživatel v příslušné nabídce zadá jméno testu a typ webové služby, pro kterou bude test určen. Následně je určena množina testova- cích pravidel, přičemž každé pravidlo je tvořeno množinou podmínek určujících, zda se celá operace vložení chyby vůbec provede, a množinou vkládaných poruch.

Doménový model aplikace byl navržen s ohledem na předpoklad, že typy podmínek a poruch se budou v průběhu životního cyklu měnit, což do budoucna zajišťuje jednoduchou rozšiřitelnost FIWS.

4 ZÁVĚR

Tento dokument se pokusil čtenáři přiblížit problematiku spojenou s testováním bezporuchovosti a robustnosti webových služeb pomocí metodiky injekce poruch. Z návrhu programu bylo představeno komunikační schéma proxy serveru, na kterém FIWS poběží, a model podmínek a poruch, který de- finuje možnosti tvorby uživatelských testů. Výsledkem celého projektu bude aplikace s uživatelským rozhraním, která dokáže pomocí předvolených nastavení monitorovat a narušovat komunikaci mezi testovanou webovou službou a jejím okolím.

Tato práce byla podpořena výzkumným záměrem č. MSM 0021630528 a projektem Centra excelence IT4Innovations (CZ.1.05/1.1.00/02.0070).

REFERENCE

- [1] Valenti, A. W.; Maja, W. Y.; Martins, E.; aj.: WSInject: A Fault Injection Tool for Web Services: Technical Report 1.0[online].
<http://www.ic.unicamp.br/~reltech/2010/10-22.pdf>, 2010-07 [cit. 2012-01-04].
- [2] Looker, N.; Munro, M.; Xu, J.: Testing Web Services [online].
<http://www.comp.leeds.ac.uk/edemand/publications/looker04c.pdf>, [cit. 2012-01-05].
- [3] Voas, J.: Software Fault Injection: Growing 'Safer' Systems [online].
<http://www.cigital.com/papers/download/snowmass97.pdf>, [cit. 2012-01-05]