

A TOOL FOR ROBUSTNESS TESTING OF WEB-SERVICES

Tomáš Zelinka

Master Degree Programme (3), FIT BUT

E-mail: xzelin15@stud.fit.vutbr.cz

Supervised by: Marek Rychlý

E-mail: rychly@fit.vutbr.cz

Abstract: This project deals with testing of web services. The result of this work will be a tool for load testing of web services using fault injection in their communication. The first part of the project discusses the basic aspects of testing web services. The second part of the work is more focused on testing high loads in combination with fault injection. The tool will allow automated run of the tests. The distributed model of the tool was designed to simulate real loads. In the end, the connection of distributed testing and fault injection is discussed.

Keywords: web services, distributed load testing, fault injection

1 ÚVOD

Webové služby nabízejí nové možnosti vývoje a použití softwaru. Jejich vývoj a používání přinesly mnoho standardů a způsobů jejich implementace. Již od počátku se webové služby potýkaly se zásadním problémem - důvěrou v jejich spolehlivost. Pro dokázání spolehlivosti webových služeb je potřeba testování. Je nutné dokázat, že implementovaná služba odpovídá specifikaci, neobsahuje chyby, nebo vymežit míru spolehlivosti. Pro různé účely existují různé techniky testování. Tato práce je zaměřena na zátěžové testování ve spojení s injekcí chyb [1]. Práce navazuje na diplomovou práci o injekci poruch pro webové služby [4].

2 TESTOVÁNÍ WEBOVÝCH SLUŽEB

Na testování webových služeb se můžeme dívat jako na testování jakéhokoliv jiného software, musíme však vzít v úvahu několik zásadních skutečností. Webové služby, ať už s použitím SOAP protokolu nebo REST principů, jsou vzdálené aplikace. Mohou sestávat z dalších služeb, které se mohou skládat z různých vrstev a mít odlišné doby odezvy. Při vytváření testů pro webové služby je v první řadě třeba zjistit, zda-li implementace odpovídá specifikaci. Poté je možné zaměřit se při testování na chování služby v zátěži, nebo na její reakce na neočekávané vstupy.

2.1 DISTRIBUOVANÉ TESTOVÁNÍ WEBOVÝCH SLUŽEB

Distribuované testování webových služeb je možno použít pro zvýšení počtu odeslaných testovacích požadavků, které jsou cíleny na danou službu. Lze jej také využít ke sběru dat z různých testovacích prostředí.

3 EXISTUJÍCÍ NÁSTROJE PRO TESTOVÁNÍ WEBOVÝCH SLUŽEB

Pro testování webových služeb existuje řada komerčních i volně dostupných nástrojů. Tyto nástroje obsahují velmi komplexní možnosti použití. Diplomová práce je inspirována některými funkcemi, které je možné nalézt ve volně dostupných nástrojích. Ty ale nenabízejí možnost kombinace zátěžového testování ve spojení s injekcí poruch.

3.1 SOAPUI

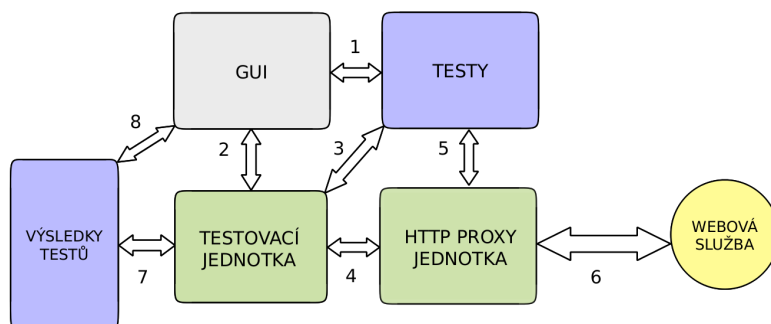
SoapUI [2] je nástroj s otevřeným kódem zaměřený na testování. Disponuje velkým množstvím funkcí včetně možnosti testování webových služeb. Umožňuje testovat SOAP i REST webové služby funkcionálními testy. Pokud je doinstalováno rozšíření loadUI, lze provádět i zátěžové testy. Rozšíření LoadUI disponuje možností distribuovaného testování.

3.2 THE GRINDER

Tento nástroj je velmi podobný předchozímu. Jeho hlavní odlišnost spočívá v ovládní. The Grinder [3] je vytvořen v jazyce Java. Je uzpůsoben k ovládní pomocí skriptovacích jazyků Jython a Closure, jež jsou používány pro ovládní The Grinder.

4 KONCEPT NÁSTROJE

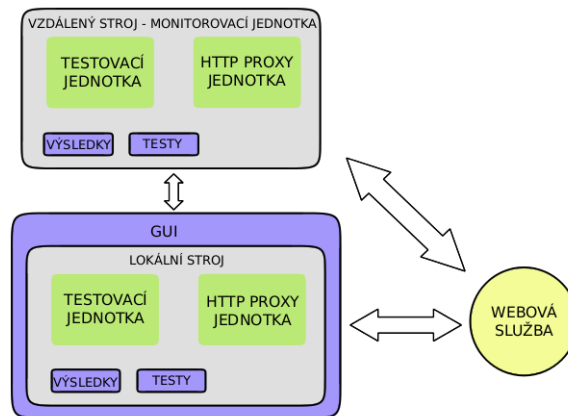
Základní použití nástroje znázorňují očíslované šipky na obrázku 1, jejichž význam bude nyní popsán. Nejdříve si uživatel v grafickém uživatelském rozhraní vytvoří testovací data (1). Poté spustí testovací jednotku (2), která vyzvedne data ze specifikovaného úložiště (3). Ta se následně přes proxy HTTP jednotku spojí (4) s webovou službou (6). Mezi tím proxy jednotka za určitých okolností vloží chybu do komunikace (5). Testovací jednotka přijme odpověď, která také putuje přes proxy jednotku a všechny výsledky uloží do specifikovaného úložiště (7). Uživatel pak bude moci z grafického uživatelského rozhraní vyhodnotit výsledky (8). Proxy jednotka se bude spouštět zvlášť a to pro případ, že ji charakter testování nebude vyžadovat. Právě proxy jednotka obsahující injektor poruch je výsledkem předchozí práce [1] a bude vylepšena o další možnosti vkládání poruch



Obrázek 1: Základní koncept.

4.1 DISTRIBUOVANÝ MODEL

Distribuovaný model odesílání požadavků je znázorněn na obrázku 2. Tento způsob odesílání umožní reálnější simulaci vysokého počtu uživatelů, protože požadavky budou odesílány z různých fyzických strojů. Pro implementaci správy jednotlivých procesů na vzdálených strojích je možno použít technologii RMI, kterou nabízí platforma Java. Na vzdáleném stroji musí běžet monitorovací jednotka, která bude zprostředkovávat vzdálené příkazy pro testovací a proxy jednotky. V grafickém uživatelském prostředí bude možnost spojit se s již běžícími instancemi monitorovacích jednotek a centralizovaně spravovat vzdálené stroje.



Obrázek 2: Model distribuované architektury.

4.2 AUTOMATIZACE TESTŮ

Testovací jednotka bude mít i rozhraní pro ovládání z příkazové řádky, aby bylo možné spouštět vytvořené testy pomocí automatizačních nástrojů jako je například Apache Ant. Rozhraní pro příkazovou řádku bude mít podobné možnosti jako grafické rozhraní. Hlavním rozdílem v ovládání přes příkazovou řádku bude absence vytváření testovacích scénářů. Nicméně, bude možné předat jednotce umístění již připravených testů v parametru a zajistit tak jejich zpracování. Testovací případy budou nastaveny ve formátu XML. Z tohoto důvodu bude možná jejich editace libovolným textovým editorem.

5 ZÁVĚR

Tento příspěvek představil testování webových služeb a možná úskalí, jež se této problematice týkají. Byl zaměřen především na představení budoucí aplikace pro zátěžové testování s využitím injekce chyb. Proxy jednotka s injektorem poruch je výstupem z předchozí diplomové práce o injekci chyb pro webové služby [4]. Tato proxy jednotka byla převzata a následně zpracována do představeného návrhu. Výsledkem projektu bude nástroj, díky němuž bude moci uživatel vytvářet testy pomocí grafického uživatelského rozhraní a následně je automatizovaně spouštět. Práce bude doplněna o převzatou proxy jednotku, která bude v rámci této práce vylepšena o další funkcionality.

REFERENCE

- [1] GRAHAM, Dorothy, Erik VAN VEENENDAAL, Isabel EVANS a Rex BLACK. *Foundations of software testing: ISTQB certification*. Australia: Thomson, 2007. ISBN 978-184-4803-552.
- [2] SMARTBEAR SOFTWARE. *SoapUI* [online]. 2013, 2013-01-10 [cit. 2013-03-25]. Dostupné z: <http://www.soapui.org>
- [3] ASTON, Philip a Calum FITZGERALD. *The Grinder* [online]. 2001-01-17, 2012-01-01 [cit. 2013-03-25]. Dostupné z: <http://grinder.sourceforge.net/>
- [4] ŽOUŽELKA, Martin. *Injekce poruch pro webové služby*. Brno, 2012. Dostupné z: <https://www.fit.vutbr.cz/study/DP/DP.php.cs?id=12072&file=t>. Diplomová práce. Vysoké učení technické, Fakulta informatiky. Vedoucí práce Marek Rychlý.