

AUTOMATIZATION OF TESTING PROCESSES

Petr BOHÁČEK, Bachelor Degree Programme (3)
Dept. of Intelligent Systems, FIT, BUT
E-mail: xbohac02@stud.fit.vutbr.cz

Ivo ŘEZNÍČEK, Bachelor Degree Programme (3)
Dept. of Intelligent Systems, FIT, BUT
E-mail: xrezn15@stud.fit.vutbr.cz

Supervised by: Dr. Zdeňka Rábová

ABSTRACT

The object of this work is to create a process of automated regressive testing of computer applications: mainly focusing on stress testing of antivirus applications. Specifically, this means the massive copying of files, and testing databases under extremely short and extremely large data queries (including the simulation of actual client-side functionality). We developed an information system based on PHP and MySQL as the basic source of information for workers in the testing laboratory. Other parts of our work focus on the operational load on the antivirus core, and simulation of the antivirus clients communicating remotely with the remote administration server.

1 ÚVOD

V mnoha vývojářských firmách je testování správného běhu softwaru poslední instancí prověření kvality, která říká zda má být produkt postoupen zákazníkům. S touto rolí přichází i velká zodpovědnost: Úspěch aplikace a potažmo i firmy přímo závisí na kvalitě jejich softwarových produktů.

Automatizace testovacích procesů může vývoj aplikace značně zlevnit, zrychlit i zpřehlednit. Znovupoužitelné testy se uplatňují hned na počátku etapy implementace a můžou být užívány až do ukončení životního cyklu programu. Šetření finančních prostředků je přímo úměrné době, po kterou je software podporován (mnoho verzí, které musí splňovat stejné podmínky již vytvořených testů) a jeho složitosti (není třeba vysoce kvalifikovaných testerů a čas na provedení testů je značně kratší). Testování je zpřehledněno jednoznačnými výstupy testů. Celý proces může zohlednit i strukturu podniku (informována budou jen ta oddělení, která mohou danou chybu řešit).

K testování je potřeba vytvořit dobré hardwarové, softwarové, dokumentační a lidské zázemí. V rámci práce jsme se věnovali tvorbě informačního systému, který pokrývá potřeby zaměstnanců pracujících v antivirové testovací laboratoři. Tento informační systém je schopen uchovat informace o přístupném hardwaru, softwaru i o dění v laboratoři.

Další části práce se věnují implementaci softwaru sloužícího k testování antivirových programů. První aplikace dokáže testovat jádro antivirového systému hromadným kopírováním souborů (i ze síťových cest a více vláken). Další umožní testovat vzdálenou správu antivirového programu. Toto testování má charakter simulace síťové komunikace mezi stanicemi a serverem vzdálené správy.

2 NÁVRH A IMPLEMENTACE APLIKACÍ

Automatizaci testovacích procesů jsme pojali jako problém týkající se výběru a nastavení podporovaných hardwarových prostředků, jakož i implementace softwaru, který bude užit v testech. Prvním krokem je tedy návrh hardwarového vybavení jednotlivých počítačů v testovací laboratoři. Mají zde být zastoupeny počítače, které jsou podporovány testovaným produktem (s moderním i starším hardwarem). Tím částečně vyloučíme problémy specifické pouze pro jednu produktovou řadu počítačů nebo jejich součástí.

Veškerý hardware je nutné katalogizovat a mít přehled o jeho základních vlastnostech. Těmi jsou např. typy procesorů, disků, síťových karet atd. Rozhodli jsme se tudíž k vytvoření informačního systému, který tyto důležité informace bude uchovávat. Pro implementaci jsme zvolili systém PHP/MySQL vzhledem k multiplat. charakteru webového rozhraní (webové prohlížeče jsou k dispozici v každém operačním systému), server Apache (s PHP modulem) a databázi MySQL.

Testování nebude omezeno na určitý druh operačního systému (podporovaných systémů může být mnoho od různých výrobců). Před testováním bude nutné instalovat různé aplikace (ať již ty testované, testující či podpůrné). Z tohoto důvodu je nutné automatizovat proces znovuoobnovení výchozího stavu celého systému. Protože instalace operačního systému je časově náročná, navrhli jsme jinou metodu obnovy. Ta spočívá v instalaci systému, provedení nastavení a případné instalaci dalších důležitých aplikací. Takto unifikovaný systém se zálohuje softwarem třetí strany, který dovoluje vytvořit kopii disku z různých formátů souborových systémů. Tuto kopii (obraz) je nutné zálohovat a informaci je nutné také katalogizovat s jejími vlastnostmi (typ operačního systému, již předinstalované programy atd.).

Dále jsme se zabývali návrhem a implementací následujících aplikací.

Testování rezidentního štítu

Rezidentní štít je proces, který neustále běží a monitoruje přístup k souborům na disku. Pokud jakákoliv aplikace požaduje otevření souboru, rezidentní štít nejprve otevře soubor sám. Provede test, a pokud je nalezena virová infekce, zablokuje přístup k tomuto souboru (popř. informuje uživatele o dalších proveditelných akcích). Princip testování jádra rezidentního štítu spočívá v aplikaci, která otevírá soubory a kopíruje jejich obsah. Na testovací aplikaci jsou ovšem kladeny i další nároky. Proces rezidentního štítu nesmí při kopírování „příliš“ zatěžovat systém. Zajímá nás tedy také časový rozdíl mezi kopírováním při běhu procesu rezidentního štítu a bez něj. Je také vhodné, aby aplikace umožňovala kopírování více souborů najednou, poté budeme moci simulovat zátěž více aplikací přistupujících k souborům, kterým se musí antivirový program věnovat.

Testování komunikace vzdálené zprávy

Antivirový program je vybaven komunikační knihovnou vzdálené zprávy, která je pomocí TCP spojení připojena k aplikačnímu serveru. Ten je připojen k Databázovému serveru. Testuje se zejména stabilita a výkon Aplikačního a částečně i databázového serveru. Je tedy nutné simulovat komunikační knihovnu vzdálené zprávy. Aplikační server musí protokolovat průběh komunikace. Dále jsme se rozhodli pro konverzi těchto záznamů na formát, ve kterém bude specifikován SQL dotaz a doba, po kterou se musí čekat mezi jednotlivými SQL dotazy. Simulační aplikace bude disponovat analyzátozem tohoto vstupního formátu a možností přepínat mezi simulací reálného a stresového provozu.

Dále jsme vyvinuli program pro uložení obsahu databáze vzdálené správy od zákazníků. Obsahy databází jsou zaslány zákazníky, kteří mají problémy se vzdálenou správou. Tyto chyby mohou vzniknout fyzickým poškozením databázového souboru, ale i chybou ve zpracovávání některých posloupností dotazů správcovskou aplikací. Proto je nutné ji netriviálně analyzovat školenými pracovníky.

Náš systém bude o databázi vzdálené správy uchovávat tyto údaje: datum a čas přidání, čas modifikace záznamu, uživatele vkládajícího záznam, poznámky od testerů či pracovníku, kteří zkoumají poškozenou databázi.

Automatizace testování firewallu

Poslední prací je návrh a zavedení systému automatického testování. V této době je částečně implementováno automatické testování firewallu. V souvislosti se snahou unifikovat vstupní i výstupní soubory přijímá aplikace testující firewall vstup ve formátu XML a výstupem je speciální formát, který je na XML snadno převoditelný (nazvali jsme ho Pseudo XML, neboli PXM). Po převodu PXM na výstupní XML, lze s použitím XSL transformací generovat HTML dokument, který je již snadno čitelný. HTML lze poslat např. jako elektronickou poštu zainteresovaným oddělením. Vstupem do celého procesu je generátor testu ve kterém lze nastavit pravidla i možnost spustit test ihned po vygenerování.

3 ZÁVĚR

V této práci jsme implementovali různé části projektu v jazycích PHP, Javascript, XML, C, C++ či unixovém shellu. Dotkli jsme se problematiky zátěžového testování antivirových aplikací přístupem k souborům i zatížením komunikací v síti.

Veškeré navrhované postupy implementovali ve vyhrazené laboratoři. To nám umožnilo odladit všechny hrubé chyby a nejasnosti, které se při užívání informačního systému a aplikací objevily. Současně s tím jsme byli konfrontováni s realitou běhu zavedené antivirové firmy, což bude pro naši další práci velice přínosné.

Další rozvoj této práce vidíme v prohloubení pracovních metodik a tvorbě testovacích programů. Vstupy a konečné výstupy všech námi vytvořených programů budou uchovávány v bezsémantickém jazyce XML. To dává dalším vývojářům mnoho možností jak znovu interpretovat již naměřené výsledky s jen malým úsilím.