Souhrnná výzkumná zpráva k projektu *Zlepšování kvality software* za rok 2017

Objednatel: Red Hat Czech, s.r.o.
Zhotovitel: Fakulta informačních technologií, Vysoké učení technické v Brně
Koordinátor projektu na FIT VUT: prof. Ing. Tomáš Vojnar, Ph.D.

1. Úvod

Projekt *Zlepšování kvality software* zahrnuje výzkumné, vývojové a experimentální práce v několika vzájemně komplementárních oblastech týkajících se různých aspektů kvality software. V roce 2017 byly zahrnuty jak aspekty spolehlivosti, výkonnosti, tak také bezpečnosti software. Předmětem projektu byly mimo jiné následující oblasti:

- Vývoj technik a nástrojů pro automatizovanou statickou i dynamickou analýzu programů s důrazem na analýzu výkonnosti, resp. analýzu spotřeby výpočetních zdrojů programy.
- Vývoj metod a nástrojů pro automatizovanou podporu migrace aplikací.
- Refaktoring nástroje mkfs xfs a jeho verifikace s využitím různých nástrojů pro statickou analýzu.
- Návrh a implementace nástrojů umožňujících zobrazovat stav průběhu testů ve vývojovém prostředí Eclipse.
- Využití regresní a korelační analýzy dat naměřených během výkonnostního testování s cílem rozpoznat ve výsledcích odchylky od normálního chování a z nich automaticky identifikovat výkonnostní problémy.
- Analýza nástrojů a procesů průběžné integrace používaných ve společnosti Red Hat a návrh možnosti jejich sjednocení s využitím automatizované podpory.
- Návrh a implementace nástroje pro třídění, filtrování a spojování údajů o chybových hlášeních shromažďovaných nástrojem Automatic Bug Reporting Tool.
- Analýza zranitelnosti SSL/TLS protokolu a implementace vybraných útoků v nástroji tlsfuzzer pro testování SSL/TLS implementací.
- Vývoj internetové služby umožňující vzdálené použití InfiSpectoru, tedy aplikace, která graficky zobrazuje komunikaci mezi uzly Infinispan serverů.

Níže jsou blíže zmíněny dvě nejvýznamnější oblasti, které byly v rámci projektu v roce 2017 rozvíjeny.

2. Statická a dynamická analýza výkonnosti programů

V rámci tohoto tématu byla navržena, v nástroji Ranger prototypově implementována a experimentálně ověřena nová metoda statické formální analýzy konečnosti běhu a analýzy spotřeby výpočetních zdrojů programů s dynamickými datovými strukturami. Tato metoda staví na využití výsledků získaných pomocí statické formální analýzy dosažitelných tvarů dynamických datových struktur (tzv. „shape analysis“). V nástroji Ranger je za tím účelem použito konkrétně výstupu z nástroje Forester založeného na lesních automatech (tzv. „forest automata“). Na základě výsledků analýzy dosažitelných tvarů převádí Ranger daný program s dynamickými datovými strukturami na čistě numerický program, a to s využitím originální sady měr nad dynamickými datovými strukturami a pravidel popisujících, jak se hodnoty těchto měr mění při aplikaci různých programových konstrukcí. Výsledný numerický program je analyzován nástrojem pro formální amortizovanou analýzu spotřeby zdrojů (tedy „amortized resource bounds analysis“). Nástroj Ranger za tím účelem konkrétně staví na nástroji Loopus. Vytvořený nástroj byl otestován na řadě krátkých, ale složitých programů, u nichž formální analýza složitosti byla dříve nad možnosti automatických nástrojů.

Dále pak pokračoval vývoj nástroje Perun zaměřeného na dynamickou analýzu spotřeby zdrojů programy. Perun je spojen s nástrojem pro správu verzí (aktuálně git). Umožňuje sbírat pomocí různých sond informace o spotřebě zdrojů programy a tyto údaje ukládat spolu s jednotlivými verzemi programů. Perun samozřejmě podporuje možnost vývoje a nasazení vlastních sond uživatelem. Shromážděná data je možno následně vizualizovat, analyzovat různými statistickými metodami a také detekovat automaticky problémy s degradací výkonu programů při přechodu mezi verzemi.

3. Vývoj metod a nástrojů pro automatizovanou migraci aplikací

V druhé polovině roku 2017 byly zahájeny práce na vývoji metod a nástrojů pro automatickou migraci aplikací mezi různými verzemi operačního systému a v souvislosti s tím i pro podporu automatizované tvorby kontejnerů. Jedním z problémů, jehož řešení bylo v souvislosti s tím zahájeno, je automatizované posouzení, jaké přepínače jádra jsou použity ve výchozí verzi operačního systému, jaké jsou k dispozici v cílové verzi a posouzení, zda a jak je možno tyto přepínače na sebe namapovat. Za tím účelem je studováno mj. využití technik vyřezání relevantních částí zdrojového kódu („code slicing“) a jeho symbolické exekuce.

4. Závěr

Výstupy projektu dosažené v roce 2017 byly objednateli předány v jím požadované podobě zahrnující (dle konkrétních témat) zdrojové kódy, zprávy, či experimentálně získaná data. V řadě z uvedených oblastí přitom probíhá a bude probíhat další výzkum i v roce 2018.