

Vývojářské nástroje pro SPL testování

Softwarový projekt

Motivace, kontext

Testování je nedílnou součástí vývoje současných softwarových systémů. Testování typicky probíhá na několika úrovních, kde na nejnižší úrovni je „unit“ testování zaměřené na ověřování správnosti jednotlivých metod a tříd. Toto testování ale nebere v potaz výkonnostní požadavky.

Pochopitelně existují nástroje, které přidávají požadavky na výkon k existujícím unit testům, aby umožnily testovat i výkon automaticky (jako například rozšíření JUnitPerf [1] pro JUnit). Nicméně, tyto jsou spíše zaměřené na „absolutní“ výkon („funkce $f()$ běží méně jak 2s na tomto vstupu“) než na relativním porovnání („funkce $f()$ je dvakrát tak rychlá jako funkce $g()$ na vstupu velikosti 200“).

SPL logika (stochastic performance logic)[2] je zaměřena právě na zachycení výkonnostních požadavků s cílem umožnit automatické testování bez ohledu na platformu. Cílem SPL je testování/měření malých částí kódu a jejich relativní porovnávání.

V SPL lze například zapsat, že očekáváme, že implementace šifrovacího algoritmu Rijndael bude maximálně 200krát pomalejší než kopírování paměti pro vstup velikosti 256, 1024 a 4096:

$$\forall n \in \{256, 1024, 4096\} : \text{Rijndael}(n) \leq_{(id,x.200x)} \text{MemCopy}(n)$$

SPL vznikla nedávno a postrádá jakékoliv nástroje, které by umožnily integraci do existujících projektů. Cílem tohoto softwarového projektu je proto vytvořit tyto podpůrné nástroje, které by měly usnadnit práci vývojářům/uživatelům SPL.

Cíle projektu

Vlastní projekt se skládá z několika částí, které na sebe navazují a které spolu souvisí.

Implementace projektu bude v Javě a velká část implementace bude rozšiřovat již existující nástroje. Projekt bude vyvíjen pro platformy Windows a GNU/Linux.

- Testovací framework, který spustí testy, změří jejich výkon a zobrazí výsledky.
- Integrace s Eclipse v podobném duchu jako existuje současná integrace s JUnit [3].
- Integrace s nástrojem pro kontinuální testování (např. Hudson).
- Rozšíření SPL formulí, aby byly snadno použitelné i pro regresní testování.
- Demonstrace použitelnosti vytvořených nástrojů.

Testovací framework

Úkolem testovacího frameworku je spustit SPL testy, změřit jejich výkon a spočítat příslušné statistiky, aby bylo možné rozhodnout, zda implementace splňuje zadané formule.

Na rozdíl od „klasických“ unit testů, měření výkonnosti může být ovlivněno mnoha faktory (např. neočekávaná zátěž sítě) a framework musí umět uživateli poskytnout informace o závažnosti chyby (porušení požadavků formule).

Výstup měření musí být navíc rozumně seřazený podle důležitosti, aby se vývojář mohl více soustředit na závažnější problémy.

Řešitelé budou moci využít jako základ již existující nástroj připravený vedoucím projektu.

Integrace s Eclipse

Integrace s vývojovým prostředím Eclipse se bude skládat ze dvou částí – rozšíření editoru zdrojových kódů a zobrazování výsledků měření.

Editor bude rozšířen o podporu syntaxe SPL formulí. Tato podpora bude obsahovat funkcionalitu typickou pro editor v Eclipse: zvýrazňování syntaxe, navigaci v kódu či automatické doplňování.

Zobrazení výsledků umožní jednoduchý přechod mezi výsledky měření a odpovídajícími metodami a řazení podle závažnosti chyb. Výsledky budou získány buď spuštěním testů přímo z vývojového prostředí nebo interpretací dříve naměřených hodnot (u komplikovanějších formulí může sběr dat trvat poměrně dlouhou dobu a proto bude kladen hlavní důraz na druhý případ).

Regresní testování

Současný návrh SPL formulí nepočítá s regresním testováním, tj. s případem, že by se porovnávala rychlost téže metody v různých revizích (časech). Bude proto nutné syntax rozšířit o zápis revize (minimálně pro Subversion a nějaký distribuovaný VCS, např. Git) a dodefinovat jeho sémantiku.

Kontinuální testování pro SPL

Cílem tohoto podúkolu bude rozšíření existujícího nástroje pro „continuous integration“ (CI) o podporu SPL.

Důraz bude kladen na ukládání naměřených dat (popř. jejich statistik), aby bylo možné provádět regresní testování v rozumném čase.

Předpokládá se, že plugin do Eclipse – pokud nespustí testy přímo – získá výsledky měření právě z CI systému. Z tohoto důvodu musí modul do CI umět nabídnout výsledky (statistiky, popř. jejich grafické znázornění) ve strojově zpracovatelné formě pro stažení.

Případová studie

Předpokládá se, že řešitelé použijí SPL k anotování částí vlastního kódu (minimálně pro účely prvotního testování). Je ale nutné, aby použitelnost vyvinutých nástrojů ukázali i na jiné aplikaci. Nepředpokládá se anotace veškerých metod, ale očekává se, že řešitelé prozkoumají i starší verze vybrané aplikace a pomocí SPL zpětně „odhalí“ (popř. potvrdí) problémy s výkonem a jeho vyřešení.

Informace o projektu

Vedoucí: Vojtěch Horký (horky@d3s.mff.cuni.cz)

Počet řešitelů: 4

Doba trvání: 9 měsíců

- Porozumění SPL logice, její rozšíření pro regresní testování: 1 měsíc
- Testovací framework: 3 měsíce
- Integrace do Eclipse a Hudsonu: 2 měsíce
- Demonstrace na běžné aplikaci: 2 měsíce
- Ladění chyb, dokumentace: 1 měsíc

Odkazy

[1] <http://clarkware.com/software/JUnitPerf.html>

[2] <http://d3s.mff.cuni.cz/publications/download/SPL-2011.pdf>

[3] <http://help.eclipse.org/helios/topic/org.eclipse.jdt.doc.user/gettingStarted/qs-junit.htm>