

Analýza paralelních C++ programů

Vladimír Štill

Anotace

Pokud chceme plně využívat výpočetních možností dnešních počítačů, je třeba využívat paralelní software. To s sebou přináší mnohé dodatečné problémy, se kterými se programátoři nesetkají při vývoji sekvenčních programů. Tato práce prezentuje několik vylepšení v oblasti analýzy paralelních programů v programovacím jazyce C++, obzvláště pak v oblasti hledání těžko odhalitelných chyb.

První přínos této práce je v oblasti analýzy vysokoúrovňových programovacích jazyků se všemi jejich pokročilými vlastnostmi a standardními knihovnami. Toto téma výrazně ovlivňuje praktickou použitelnost nástrojů pro analýzu programů programátory. Bohužel, kompletní podpora daného programovacího jazyka není snadný cíl. I přes to však ukazujeme, že tento cíl je dosažitelný s pomocí vhodné kombinace existujících komponent určených k použití ve spustitelných programech a komponent vytvořených speciálně pro účely analýzy programů. V této práci se konkrétně zaměříme na podporu programovacího jazyka C++ v nástroji DIVINE a na podporu výjimek v C++ v tomto nástroji.

Druhý přínos této práce spočívá v návrhu nového postupu pro analýzu programů běžících pod relaxovaným paměťovým modelem procesorů x86. Tyto procesory mohou zpožďovat zápisy do paměti a provádět je až po provedení nezávislých čtení. Tento přístup však může vést k podivnému chování paralelních programů, s čímž se programátorům špatně pracuje, a proto může docházet ke špatně odhalitelným, leč závažným chybám. Náš přínos spočívá v novém způsobu simulace relaxovaného chování v nástroji pro analýzu programů, konkrétněji v tom, že minimalizujeme množství nedeterminismu, který je třeba do programu přidat, aby bylo možné posoudit jeho chování v systému s relaxovanou pamětí. Toto omezení nedeterminismu snižuje celkovou náročnost analýzy.

Posledním přínosem této práce je metoda detekce lokální neterminace paralelních programů, tedy částí programu, které by měly skončit, ale neděje se tak. Tato metoda umožňuje detekovat neterminaci v programech, které nekončí, ale mají části, které musí skončit, tedy například v serverech a jiných službách, které obsluhují dotazy – takový dotaz musí být vyhodnocen v konečném čase i přes to, že služba jako taková běží nepřetržitě. Naše metoda využívá jednoduché anotace k označení částí programu, které musí skončit, a dokáže pracovat s libovolnými synchronizačními mechanismy. Omezením naší metody je, že dokáže pracovat jen s programy, které mají konečný stavový prostor (takové programy však mohou mít nekonečné chování, v němž se některé stavy programu opakují). I přes toto omezení však věříme, že je naše metoda užitečná při návrhu paralelních algoritmů a datových struktur.

Přínosy prezentované v této práci jsou rovněž implementovány v open-source nástroji DIVINE a jsou podloženy experimentálním vyhodnocením.