



SafeTSA - Mobiler Code der Zukunft

Im Zeitalter des Internets treffen wir zunehmend auf mobilen Code. Das sind Programme, die in einer einzigen Version an verschiedenartige Rechnersysteme gesendet und mit der gleichen Semantik auf jedem dieser Rechner ausgeführt werden können. Solch mobiler Code wird über ein Netz, beispielsweise das Internet, geladen und unter Verwendung einer Ausführungsumgebung oder nach einer dynamisch durchgeführten Übersetzung auf dem Zielrechner ausgeführt.

Die dynamische Übersetzung mit Hilfe eines adaptiven Compilers stellt ein hoch komplexes System dar, in dem unter anderem hohe Anforderungen an den mobilen Code gestellt werden. Um diesen Anforderungen zu genügen, muss bei der Entwicklung eines Systems für mobilen Code besonders auf die verwendete Zwischencoderepräsentation geachtet werden. Ein geeignetes Zwischencodeformat sollte sich durch Eigenschaften wie hohe Portabilität, geringe Manipulationsanfälligkeit und Unterstützung einer effizienten Ausführung auszeichnen.

SafeTSA wurde als eine Alternative zu Java-Bytecode entworfen. Es ist die erste absolut referenz- und typsichere, auf Static-Single-Assignment-Form basierende Zwischencoderepräsentation für mobilen Code. Gegenüber Java-Bytecode hat sie mehrere entscheidende Vorteile:

- direkte Verwendung der Zwischencoderepräsentation innerhalb eines optimierenden Compilers
- maschinenunabhängige Codeoptimierungen bei der Erzeugung des mobilen Codes sind durchführbar
- Referenz- und Typsicherheit der Programme ist garantiert
- Verifikationsarbeit auf dem Zielrechner wird minimiert
- kompakte binäre Encodierung der zu übertragenden Programme

Die Entwicklung der mittlerweile patentierten Zwischencoderepräsentation SafeTSA wurde in den Jahren 2002 bis 2009 von der Deutschen Forschungsgemeinschaft mit einem Gesamtvolumen von nahezu 600.000 Euro gefördert.



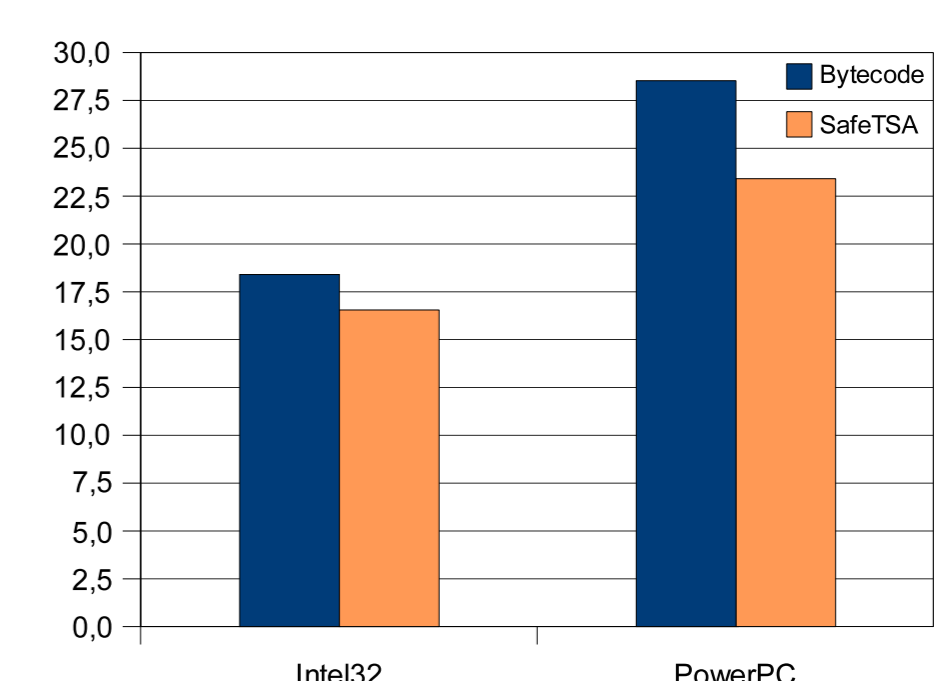
Evaluierung

SafeTSA bietet grundlegende Voraussetzungen für die Laufzeitverbesserung von Programmen. Der Programmcode liegt in einer für Optimierungen günstigen Form vor: der Static-Single-Assignment-Form. Darüber hinaus ist die verwendete Programmstruktur, im Gegensatz zu Java Bytecode, baumorientiert und mit Hochspracheninformationen angereichert. Die darin enthaltenen Instruktionen sind atomar und können damit ohne zusätzliche Transformationen optimiert werden.

Aufgrund dieser Eigenschaften werden Programmoptimierungen begünstigt und sind auf einer hohen Sprachebene anwendbar. Die Optimierungen können statisch während der Übersetzungszeit und dynamisch zur Laufzeit erfolgen. Messungen der Programmlaufzeit ergaben eine Verbesserung von 10-18% gegenüber herkömmlichem Bytecode.

Optimierungen können ebenfalls genutzt werden, um die Größe des mobilen Codes zu reduzieren. Je kleiner und kompakter der mobile Code ist, umso schneller kann er von einem Rechner zu einem anderen übertragen werden. Die Binärcodierung von SafeTSA unterstützt genau diese Eigenschaft. Die Größe der Programmdateien konnte um durchschnittlich 43% gegenüber entsprechenden Bytecode-Klassendateien gesenkt werden.

Durchschnittliche Ausführungszeit in Sekunden



Durchschnittliche Dateigröße in Byte

