

GraphJIT: a dynamic graph bytecode JIT compiler

Shijie Xu, David Bremner, Daniel Heidinga
University of New Brunswick, IBM Canada
Faculty of Computer Science
{sxu3, dbremner}@unb.ca, daniel_heidinga@ca.ibm.com

Background

- Graph structures are prevalent--A number of hops from root to leaves are involved for a traversal.
- Traversal on a graph consumes a number of computing resources;
- Traversal is still necessary even if the graph has been JITted on the JVM.

Our Solution

- GraphJIT: a dynamic graph bytecode JIT compiler that translates a graph on the bytecode level, for graph simplification.
- The main idea is to move leaves closer to the graph root by fusing graph internal nodes.

Performance

GraphJIT is applied to compile method handles, which are created by dynamic JVM language interpreters for dynamic method invocations.

- Ahead-Of-Time (AOT) compilation:
The execution time of the common MHG from JRuby micro-indy benchmark is reduced by 31%.
- Just-In-Time (JIT) compilation
 - JRuby micro-indy benchmark: 7% speedup on average.
 - JavaScript Octane benchmark on Nashorn: 6% speedup, with maximal 30%.

