Abstract

Fast Debugger Features to Trace Forked Processes of Ruby and Python Parallel Applications

Norio Sato[1,a)]

Presented: August 2, 2013

This manuscript reports an extended feature and speed-up of Dionea, a visual debugger having been developed by us for Ruby and Python. By this extension, applications using process-based parallel libraries such as Ruby's grosser/parallel and Python's multiprocessing are made debuggable. Such libraries wrap fork, and let processes look like threads. Dionea is a thread-aware debugger for distributed processes. It consists of one debugger (UI) and many debuggee processes. Both are asynchronously connected via non-blocking sockets. The newly developed portion in the debuggee is a wrapper of existing fork to make the forked process connectable from the debugger. In the debugger (UI), we added copying of debugger data such as breakpoints, from parent to child proxy objects that interact debuggee processes. The effects of the proposed features are demonstrated using typical but non-trivial parallel patterns such as "map-fold" and "map-reduce". It proved, however, that fast tracing of script code under debugging is required to make the proposed feature effective in realistic debugging of such applications. We have developed, therefore, two native code modules (so-called extension modules), each for Ruby and Python, which hook line by line based callbacks and suppress callbacks to script except when debug events really occur. Using these native code modules, the slow-down to trace Ruby and Python code can be reduced from hundreds of times to only twice or so compared with normal running speed.

[1] Department of Information and Computer Science, Kanazawa Institute of Technology, Hakusan, Ishikawa 924–0838, Japan

[a)] satoh@infor.kanazawa-it.ac.jp