

発表概要

Python・Ruby 記述の分散多重スレッドプログラム用 部分停止型デバッガの開発

伊藤 泰[†] 永井 和宏[†] 小菅 圭介^{††}
小倉 正充^{††} 佐藤 規男^{†,††}

主要スクリプト言語は、ネットワークプログラミング等での応答性の向上等のため多重スレッド機能を提供している。そこで我々は、部分停止 (low-intrusion) モデルにより拡張した分散型多重スレッドデバッガを提案する。このモデルでは、デバッグ操作ごとにプロセス全体の実行を停止させる既存の世界停止 (stop-the-world あるいは high-intrusion) モデルとは異なり、プロセス全体は実行状態のまま個々のスレッドを制御できる。本発表では Python 用に開発し Ruby 用に拡張したデバッガの機能と実装を述べる。本デバッガはネットワーク上で協調する複数プロセスを扱うため、クライアント部と言語対応サーバ部に分離し、両者を言語共通形式にコード化された非同期メッセージで結合している。前者は同時に複数のプロセスを捕捉でき、複数スレッド操作を容易にするため GUI を提供する。サーバ部の実装はデバッグ API 補強部、クライアント部からデバッグ用コマンドを受け取る専用スレッド、およびデバッグ対象スレッドがコールバックする関数群から成る。また、部分停止モデルが有効となるような場合をスレッド協調パターンを用いて示す。さらに世界停止モデル、部分停止モデル両者の得失を議論し、両者および実時間トレースの融合環境を提案する。

Low-intrusion Debugger for Python and Ruby Distributed Multi-thread Programs

YASUSHI ITOH,[†] KAZUHIRO NAGAI,[†] KEISUKE KOSUGA,^{††}
MASAMITSU OGURA^{††} and NORIO SATO^{†,††}

Major scripting languages provide with multi-thread features that could improve response time as for network programming etc. We propose a new multi-thread debugger extended with “low-intrusion” model. With this model, we can control individual threads without suspending a whole process, whereas existing debuggers with “high-intrusion” or “stop-the-world” model cannot. We present the features and implementation of a debugger we have developed for Python and extended for Ruby. The debugger consists of server and client parts to handle communicating processes via network. Both parts are coupled with asynchronous messages encoded by a common format. This allows for developing the client part in common, while the server parts are implemented for individual languages. The client part can catch more than one process at the same time, and provides users with full GUI support to facilitate the handling of multiple threads inside of the processes. We have implemented the server parts with extension modules: a native code reinforcement, a dedicated thread listening to debug commands coming from the client part, and functions call-backed by individual debugged threads. We present typical threading patterns for which low-intrusion is effective. We compare high- and low-intrusion models, propose an integrated environment of both models including realtime tracing.

(平成 16 年 3 月 18 日発表)

[†] 金沢工業大学大学院情報工学専攻
Graduate School of Information and Computer Science,
Kanazawa Institute of Technology

^{††} 金沢工業大学情報工学科
Department of Information and Computer Science,
Kanazawa Institute of Technology