

並列処理プログラム用リモートデバッガ

1 L - 5

横山 法明 大谷 雄史 橋ヶ谷 誠

(株)日立製作所 ソフトウェア開発本部

1. はじめに

近年、並列処理を行うプログラムの開発が進んでくるとともに、デバッグ時に対象とする被テストプログラムの数が増大してきている。従来は、並列処理プログラムの被テストプログラム毎にデバッガを起動していた。この方法では、起動したデバッガが互いに独立しているため、複数のプログラムを関連させながらデバッグすることは難しい作業であった。

そこで、複数の被テストプログラムを一括して扱えるようなアプリケーション用デバッガ DBR (DeBugging system for Remote environment)を開発した。

2. デバッガへの要求事項

並列処理を行うプログラムでは、同時にデバッグ対象とする被テストプログラムの数が増加する。それらのプログラムのためのデバッガへの要求事項は次のとおりである。

(1) リソース使用量削減

被テストプログラムのデバッグ情報は、多くのリソースを必要とする。そのため、次の問題点を解決しなければならない。

(a) 起動したデバッガ毎にデバッグ情報をメモリ上に展開しているため、システムが大きくなれば所要メモリ量が増大する。

(b) 複数ワークステーション（以下、WS と称す）で動作する並列処理プログラムをデバ

ッグする場合、被テストプログラムとデバッグ情報を WS 毎のディスクにインストールする。そのため、所要ディスク容量が増大する。

(2) 被テストプログラム実行環境の一括管理

並列処理プログラムは同一の WS 上だけではなく、複数の WS 上でも動作する。デバッグ中はプログラムの実行環境をテスト項目に合わせて変更することもあり、その実行環境の構築・変更が簡単に行えることが望まれる。

(3) 複数のプログラムへの同時デバッグ操作

並列処理プログラムのデバッグを効率良く行うには、プログラム実行制御や情報表示等のコマンドに、複数の被テストプログラム間の連携が必要となる。

3. リモートデバッガ DBR

複数 WS 上で動作する複数のプログラムをまとめて対象とできるデバッガであれば、2. に示した要求事項を満足することができる。そのために、リモート操作でのデバッグが可能な並列処理プログラム用リモートデバッガ DBR を開発した。図1は、今回開発した DBR のシステム構成図である。

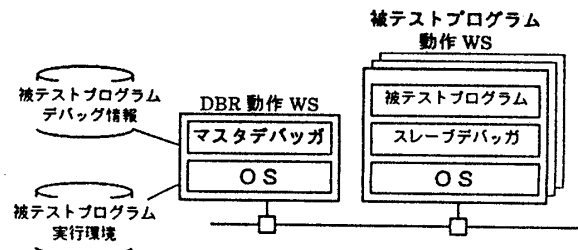


図1 リモートデバッガシステム構成

Remote Debugger for Parallel Programs

Noriaki Yokoyama Yuji Ohtani Makoto Hashigaya
Software Development Center, Hitachi, Ltd.

5030 Totsuka, Totsuka, Yokohama, Kanagawa 245, Japan

リモートデバッガの特徴は次のようになる。

(1) リモートデバッグ

図2のようなDBR動作WS上で動作する1つのDBRから、リモート操作により複数のWS上で動作する複数のプログラムのデバッグを可能とした。さらに、シンボル解析等は、DBRの動作WS上に置かれたデバッグ情報をもとに行われる。そのため、被テストプログラムが動作しているWS上にはデバッグ情報が必要ないため、そのWSの使用メモリ量とディスク容量を大幅に削減することが可能となる。

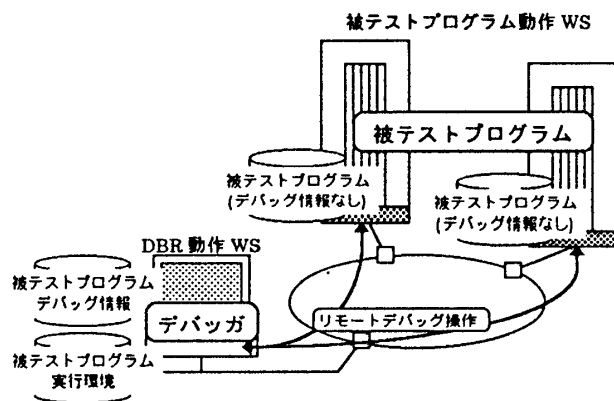


図2 リモートデバッグ操作

(2) 被テストプログラム実行環境リモート構築

3. (1)に示したようなDBR操作WSからのリモート操作を利用して、被テストプログラムの実行環境の構築も可能となる。この方式では、DBR動作WSから被テストプログラムの動作環境をそのプログラムが実行されるWSに配布する。そして、配布された環境を受け取ったWSは実行環境を整えた後、被テストプログラムの実行を行う。したがって、実行環境はDBR動作WS上で構築・変更すればよい。

(3) ユーザインタフェース

今回開発したDBRでは、被テストプログラム毎にウィンドウを用意するマルチウィンドウをサポ

ートしている。マルチウィンドウのため見掛け上は複数に見えるが、1つのDBRで処理を行っている。そのため、次のような複数の被テストプログラム間のデバッグ操作や情報の連携をとることができ、デバッグの操作性が向上している。

(a)被テストプログラム同時実行再開

(b)複数の被テストプログラム情報表示

4. 評価

大規模な並列処理プログラムに対して、従来のデバッガではデバッグに必要な情報を扱うために、多くのメモリやディスクの量を必要とした。それらの所要リソース量を削減することにより、限られたマシン環境の中でのデバッグ効率を向上させることができる。今回開発したリモートデバッガDBRは、次のとおりデバッグ効率を向上させることができた。

- ①使用メモリ量の削減 (60%減)
- ②使用ディスク量の削減 (60%減)
- ③リモート操作による実行環境構築
- ④デバッグ操作の連携

一方、リソース使用量の削減を行った結果、次のような2次的な効果を得ることができた。

- (a)複数の開発者が共用しているテスト用WSの共用率が増加する。
- (b)被テストプログラムが使用できるリソース量が増加する。

5. まとめ

並列処理プログラム用リモートデバッガについて述べた。リモート操作を可能としたデバッガとすることにより、リソースの使用効率がよく、並列処理プログラムのデバッグ操作性のよいデバッガを実現できた。

参考文献

- [1]山本、他：SR2001におけるカーネルデバッガ、情報処理学会第50回全国大会論文集（1995）