

マルチプロセス／マルチスレッド対応 GUI デバッガ TotalView®

**HPC システム
世界標準デバッガ**

ローレンスリバモ国立研究所や
ロスアラモス国立研究所、
東工大 TSUBAMEをはじめ
世界の大手研究所・教育機関で
幅広く採用！

あなたの開発したプログラムは、思った通りに動いていますか？

プログラム開発にバグはつきものです。特に並列処理の失敗やメモリエラーは再現性が低く、欠陥箇所と問題発生箇所が異なることが多いなど、一般に修正が困難です。バグは、機能追加やパフォーマンスチューニング作業を妨げ、開発スケジュールを遅らせます。さらに、稼働後に発見されると組織に大きな損害をもたらす、研究成果の妥当性が疑われ、信頼を損ないます。

バグはできるだけ素早く効率的に発見し適切に取り除かなければなりません、そのための効果的で強力なツールは多くありません。

TotalView でバグを根絶！

TotalView は、Macbook から大規模 UNIX スパコンまで幅広い環境に対応し、世界各地の企業や研究所、計算機センターで利用されています。使いやすい GUI と高機能なバッチ処理機能を備え、常に手元に置いておきたい高機能デバッガです。

C/C++、Fortran に対応し、複雑なデータ構造や GPU 上のデータ、メモリ状態を可視化します。MPI/OpenMP などの大規模並列環境や、CUDA や Xeon-Phi などのヘテロジニアスな環境での処理やデータの流れなど、複雑なコードの動きが手に取るようになります。スレッドごとに動作を変更したり横断的に監視することも可能で、並列プログラムの理解やバグ除去の効率が飛躍的に向上します。

シミュレーション開発や大規模データ分析に：

研究者や開発者はますます大容量化するデータに対し高速に精度よく成果を出すよう求められています。大規模並列や GPU などの先進的なハードウェア環境を使いこなしたコードを書くのは至難の技です。

TotalView を使えばリソースのチューニングも容易になり、手に取るようにアルゴリズムを検証できるようになります。

並列計算における身近なバグ：

▶ レースコンディション (データ競合)

アルゴリズムが複数のプロセスの実行順序に依存しており、その都度結果が変わってしまうバグです。処理速度やネットワーク速度に暗黙の仮定を設けてしまうと配列やデータファイルの書き換えが同時に発生します。

▶ デッドロック

リソース解放、イベント、他のプロセスからのメッセージなど、必要なきっかけを相互に待ち続けている状態です。データ通信の順番に細心の注意を払わないとプログラムは容易にフリーズします。

メモリ関連の身近なバグ：

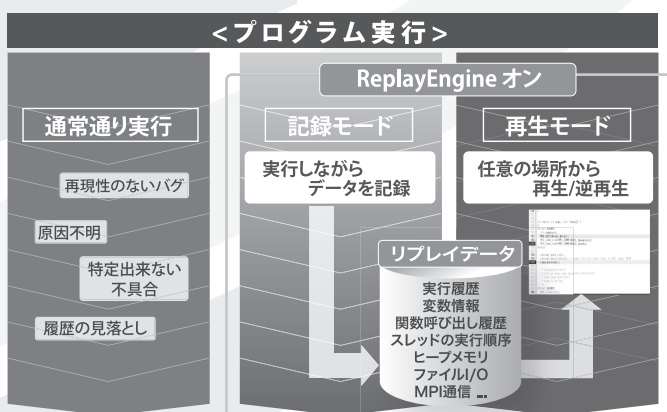
▶ メモリリーク

C/C++ や Fortran は C# や Java のようなガベージコレクタ (GC) を持った言語と異なり、動的に確保したメモリを自動では解放しません。自在にリソースを制御できる反面バグの温床にもなります。その結果、長時間稼働するアプリケーションの場合システムのリソースを使い尽くして異常終了や速度の低下を招きます。

▶ メモリクラッシュ

コード自身によるメモリの破壊です。配列の領域外アクセスや書き込みによるバッファオーバーランは、スタック破壊による異常終了や、任意のプログラムを実行させるなどの脆弱性を呼び込みセキュリティリスクにもなります。

TotalView のリバースデバッグ機能 - ReplayEngine



プログラム実行状態の記録・再生が可能なリバースデバッグツールです。

いつも通り実行しながら保存し、マルチスレッド実行を決定論的に再現します。任意の場所から自由に前後に再生や逆再生が可能なので、コードを深く理解したりエラーからさかのぼるなど、トラブルシューティングに費やされる時間を大幅に短縮できます。

英国ケンブリッジのジャッジ・ビジネススクールが2013年に行った調査によると、リバースデバッグを行うことで開発者はデバッグにかかる時間を平均13% (開発時間の26%)削減することができるそうです。