



メモリーエラーとスレッド化エラーチェッカー

インテル® Parallel Inspector 2011

製品紹介

インテル® Parallel Inspector 2011



「インテル® Parallel Inspector とインテル® Parallel Amplifier により、hotspot やメモリーリークの検出作業が驚くほど簡単になりました。全体で 2 倍ほどパフォーマンスが向上し、以前は特定できなかったメモリーリークを排除することができて、非常に満足しています。」

OpenCascade S.A.S.

ソフトウェア開発マネージャー

Vlad Romashko 氏

メモリーエラーとスレッド化エラーを早期に発見してアプリケーションの信頼性と品質を向上

インテル® Parallel Inspector 2011 は、シリアルおよび並列アプリケーションを動的に分析する、メモリーエラーとスレッド化エラーの検証を組み合わせた強力なツールです。Microsoft® Visual Studio® C/C++ アプリケーションの信頼性と品質の向上を支援します。動的インストルメントにより、コードのテストをより頻繁に、そして簡単に行えます。特別なテストビルドやコンパイラーは不要です。

- シリアル / 並列コードのメモリーエラーおよびスレッド化エラーを簡単で便利な 1 つのツールで検出
- 信頼性の高い、高品質なアプリケーションを提供することにより顧客満足度を向上
- 初心者、上級者を問わず、スレッドコードの動作に対して鋭い洞察を提供
- スレッド化の潜在的な不具合を簡単に検出
- 開発とサポートのコストを抑え、生産性を向上して、製品の開発期間を短縮

簡単で便利な 1 つのツールでメモリーチェックとスレッドチェック

メモリーチェック機能とスレッドチェック機能の両方が、使いやすいインターフェイスで Microsoft® Visual Studio® に完全に統合されます。インテル® Parallel Inspector は、クラッシュを引き起こすスレッド化やメモリーの不具合について、根本的な原因を素早く分析します。関連する不具合を要約した問題の分析とこれらの機能を組み合わせることにより、メモリーエラーやスレッド化エラーを検出する包括的なツールとなります。

標準ビルドとバイナリーで動作する動的インストルメンテーション

インテル® Parallel Inspector は、テストデータの取得に動的インストルメントを利用します。特別なビルド、アドイン、コンパイラーは不要です。実行されるコードのみをインストルメントするため、分析の実行時間が少なく、大規模なアプリケーションで作業できます。ソースコードがなくてもバイナリーのエラーを検出できます。

メモリーとスレッドの動的な分析ツール

インテル® Parallel Inspector は、シリアル / 並列コードの包括的で動的な分析ツールで、メモリーとスレッド化エラーを容易かつ迅速に発見します。メモリーリーク、不正なメモリーの読み取り / 書き込み問題、ダンダリング・ポインター、初期化されていないデータの使用、データ競合、デッドロックを検出できます。

優れた価値

インテル® Parallel Inspector の分析機能は、開発コストを抑えて製品の開発期間を短縮しながら、信頼性の高いソフトウェアを開発できるように支援します。インテル® Parallel Inspector は、インテル® Parallel Studio に含まれる、C/C++ スレッド・アプリケーションの開発、デバッグ、チューニングの包括的なツールスイートです。

| ID | Problem | Sources | Modules | Object Size | State |
|-----|------------------------------------|----------|-------------------|-------------|-----------|
| P1 | Uninitialized memory access | main.cpp | worstcodeever.exe | | Not fixed |
| P2 | Uninitialized memory access | main.cpp | worstcodeever.exe | | Not fixed |
| P3 | Mismatched allocation/deallocation | main.cpp | worstcodeever.exe | | Not fixed |
| P4 | Mismatched allocation/deallocation | main.cpp | worstcodeever.exe | | Not fixed |
| P5 | Invalid memory access | main.cpp | worstcodeever.exe | | Fixed |
| P6 | Invalid memory access | main.cpp | worstcodeever.exe | | Not fixed |
| P7 | Invalid memory access | main.cpp | worstcodeever.exe | | Not fixed |
| P8 | Invalid memory access | main.cpp | worstcodeever.exe | | Not fixed |
| P9 | Memory leak | main.cpp | worstcodeever.exe | 5 | Not fixed |
| P10 | Memory leak | main.cpp | worstcodeever.exe | 12 | Not fixed |

シングルスレッド・アプリケーションとマルチスレッド・アプリケーションで、リークや破壊を含むメモリーエラーを素早く検出します。これにより、アプリケーションの出荷前に問題を識別することができ、サポートコストを抑えることが可能です。

| ID | Problem | Sources | Modules | Object Size | State |
|----|--------------------------|----------|-------------------|-------------|-----------|
| P1 | Data race | main.cpp | worstcodeever.exe | | Not fixed |
| P2 | Lock hierarchy violation | main.cpp | worstcodeever.exe | | Not fixed |

| ID | Description | Source | Function | Module | Object Size | State |
|-----|-----------------|--------------|----------|-------------------|-------------|-------------|
| X10 | Allocation site | main.cpp:176 | DeadLock | worstcodeever.exe | | Information |
| X13 | Allocation site | main.cpp:174 | DeadLock | worstcodeever.exe | | Information |

デッドロックやデータ競合を含む潜在的なスレッド化エラーを正確に指摘し、デバッガーやその他のツールでは発見できない、スレッド化エラーが原因のストールやクラッシュを防止するのに役立ちます。

Analysis completed successfully **Interpret Result**

Event Log → Sources

| Time | Description | Modules | Sources |
|----------|---------------------------------------|-------------------|--------------------------|
| 11:06:05 | Error:Invalid memory access | worstcodeever.exe | main.cpp:52 |
| 11:06:06 | Error:Invalid memory access | worstcodeever.exe | main.cpp:54 |
| 11:06:06 | Error:Uninitialized memory access | worstcodeever.exe | main.cpp:51; main.cpp:56 |
| 11:06:06 | Error:Uninitialized memory access | worstcodeever.exe | main.cpp:51; main.cpp:57 |
| 11:06:06 | Error:Invalid memory access | worstcodeever.exe | main.cpp:61 |
| 11:06:06 | Error:Mismatched allocation/deallo... | worstcodeever.exe | main.cpp:64; main.cpp:66 |
| 11:06:06 | Error:Mismatched allocation/deallo... | worstcodeever.exe | main.cpp:63; main.cpp:67 |
| 11:06:06 | Error:Invalid memory access | worstcodeever.exe | main.cpp:79 |
| 11:06:06 | Error:Memory leak | worstcodeever.exe | main.cpp:76 |
| 11:06:06 | Error:Memory leak | worstcodeever.exe | main.cpp:192 |

[Interpret Result (結果の解釈)] ボタンをクリックすると、関連する問題がまとめてグループ化され、問題を直感的に把握できます。1つの問題を解決する場合に、インテル® Parallel Inspector で、同じ修正が必要な関連するすべての場所を表示できます。

Build | Debug | Tools | Window | Co

- Build Solution F7
- Rebuild Solution Ctrl+Alt+F7
- Clean Solution
- Build worstcodeever
- Rebuild worstcodeever
- Clean worstcodeever
- Project Only
- Profile Guided Optimization
- Batch Build...
- Configuration Manager...
- Compile Ctrl+F7

インテル® Parallel Inspector は、標準のデバッグビルドだけでなくバイナリーでも動作します。特別なコンパイラーやアドイン、特別なビルドは不要です。

Configure Analysis

Intel® Parallel Inspector

Analysis type: Memory Errors Always show this dialog box before running memory error analyses

2x-20x | Does my target leak memory?
 10x-40x | Does my target have memory access problems?
 20x-80x | Where are the memory access problems?
 40x-160x | Where are all the memory problems Inspector can find?

Analysis Time Overhead

簡単な分析設定で、分析レベルと収集時間を指定できます。

- L1 分析:メモリーリークとデッドロックを発見します。
- L2 分析:既存の問題を特定します。
- L3 分析:問題を解決する根本的原因を表示します。
- L4 分析:最も包括的なレベルの問題を特定し、詳細を提供します。

Overview → Sources ← Details

Focus Observation: main.cpp:86 - Write

```

85 for (int i=0;i<TotalBlueTroops;i++)
86     BantamBridge+=Blue;
87     return NULL;
88 }

```

Related Observation: main.cpp:92 - Write

```

91 for (int i=0;i<TotalGreyTroops;i++)
92     BantamBridge+=Gray;
93     return NULL;
94 }

```

特定された問題をクリックして、ソースコードを表示し、問題を発生させているコードに移動して、素早く変更します。

Private suppressions: Delete problems

- Delete problems
- Mark problems
- Do not use suppressions

特定された問題のうち、不要な問題をマークまたは削除して、表示を制御できます。

機能

- Microsoft* Visual Studio* との完全な統合
- シングルスレッド・アプリケーションとマルチスレッド・アプリケーションの両方でメモリーエラーを検出
- 初期化されていないロードの検出、不正なメモリー参照の使用、メモリー割り当てと解除の不一致、メモリーリークの検出、スタックメモリーのチェック、深さを指定したスタックとレースの検証を含むメモリーチェック
- スレッド化エラーの検出
- データ競合の検出、デッドロックの検出、深さを指定したコールスタックの分析、診断ガイダンス、ビルトインナレッジ（インテル® スレディング・ビルディング・ブロック（インテル® TBB）、OpenMP*、Windows* スレッド）
- 標準デバッグビルドで動作
- 特別なテストビルドやコンパイラーは不要。コードのテストをより頻繁に、そして簡単に行えます。
- 動的インストルメントにより、ソースがなくてもコードのテストが可能。実行コードのみがインストルメントされるため、メモリー消費が少なく、大規模なアプリケーションのテストが可能

評価版のダウンロード

評価版のダウンロードについての詳細は、次の Web サイトを参照してください。

<http://www.intel.co.jp/jp/software/products/intel-parallel-studio-home/>

システム要件

- Microsoft* Visual Studio* 2005、2008 または 2010（Express Edition を除く）
- 最新のシステム要件については、次の Web ページを参照してください。
www.intel.com/software/products/systemrequirements/（英語）

互換性

- コンパイラー：Microsoft* Visual C++* コンパイラー 2005、2008、2010 およびインテル® C++ コンパイラー
- スレッド化手法：インテル® TBB、インテル® Cilk™ Plus、OpenMP*、Windows* スレッド
- プロセッサー：インテル® Core™ 2 プロセッサーとインテル® Core™ i7 プロセッサーを含む IA-32 プロセッサーおよびインテル® 64 アーキテクチャー対応プロセッサー向けに設計、動作確認済み。互換プロセッサーで使用することも可能ですが、専用命令によっては正常に機能しないことがあります。インテル® Parallel Composer（コンパイラーとライブラリー）は、IA-32 およびインテル® 64 アーキテクチャー・ベースのインテル® プロセッサー、およびすべての互換プロセッサーをサポートしています。

サポート

インテル® Parallel Studio 製品をご購入いただくと 1 年間のサポートサービスが提供されます。このサポートには、インテル® プレミアサポートへの問い合わせ、製品の最新版のダウンロードおよび、テクニカル・ドキュメントへのアクセスが含まれています。

詳細は、<http://software.intel.com/sites/support/>（英語）を参照してください。

究極のオールインワン・パフォーマンス・ツールキット：インテル® Parallel Studio 2011

今日のシリアル・アプリケーション、そして明日のソフトウェア・イノベーションのための設計

インテルは、シリアル・アプリケーションおよびマルチコアやメニーコア向けの新しいスレッド・アプリケーション用に設計された生産性ソリューション・セットにより、Microsoft* Visual Studio C++* 開発者に簡素化されたスレッド・プログラミング環境を提供します。

革新的なスレッド化アシスタント

インテル® Parallel Advisor 2011: スレッド・アプリケーション設計を明確化および迅速化

コンパイラーとスレッド化ライブラリー

インテル® Parallel Composer 2011: C/C++ コンパイラーと高度なスレッド化ライブラリーで効率的なアプリケーションを開発

メモリーエラーとスレッド化のエラーチェッカー

インテル® Parallel Inspector 2011: 並列メモリーエラー / スレッドエラーを未然に防ぐ検証機能でアプリケーションの信頼性を確保

スレッドとパフォーマンス・プロファイラー

インテル® Parallel Amplifier 2011: スケーラブルなマルチコア・パフォーマンスが得られるようにボトルネックを迅速に検出してスレッド・アプリケーションをチューニング

