



チュートリアル: MPI Performance Snapshot で MPI アプリケーションを解析する

MPI Performance Snapshot (パフォーマンス・スナップショット)



著作権と商標について

本資料は、明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず、いかなる知的財産権のライセンスも許諾するものではありません。

インテルは、明示されているか否かにかかわらず、いかなる保証もいたしません。ここにいう保証には、商品適格性、特定目的への適合性、知的財産権の非侵害性への保証、およびインテル製品の性能、取引、使用から生じるいかなる保証を含みますが、これらに限定されるものではありません。

本資料には、開発の設計段階にある製品についての情報が含まれています。この情報は予告なく変更されることがあります。最新の予測、スケジュール、仕様、ロードマップについては、インテルの担当者までお問い合わせください。

本資料で説明されている製品およびサービスには、不具合が含まれている可能性があり、公表されている仕様とは異なる動作をする場合があります。

Intel、インテル、Intel ロゴ は、アメリカ合衆国および / またはその他の国における Intel Corporation の商標です。

* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

© 2016 Intel Corporation.

最適化に関する注意事項

インテル® コンパイラーでは、インテル® マイクロプロセッサに限定されない最適化に関して、他社製マイクロプロセッサ用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサ依存の最適化は、インテル® マイクロプロセッサでの使用を前提としています。インテル® マイクロアーキテクチャに限定されない最適化のなかにも、インテル® マイクロプロセッサ用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。

注意事項の改訂 #20110804

内容

1. 概要	4
1.1. 必要条件.....	4
2. MPI アプリケーションの解析	5
2.1. 解析の準備.....	5
2.2. アプリケーションの統計情報を見る.....	6
2.2.1. Summary ページを参照する.....	6
2.2.2. Function Summary (関数サマリー) を参照する.....	7
2.3. 問題の解決.....	7
2.3.1. SendRecv 関数をチューニング.....	7
2.3.2. Allreduce 関数をチューニング.....	8
2.4. 結果を表示.....	8
3. 重要な用語	10



1. 概要

MPI Performance Snapshot は、MPI アプリケーション向けのスケーラブルで軽量なパフォーマンスツールです。通信、アクティビティ、負荷バランスなどの MPI アプリケーションの特性を収集し、分かりやすい形式で表示します。アプリケーションのスケーラビリティとパフォーマンスを詳しく解析するため収集された情報を使用します。

このチュートリアルについて	<p>チュートリアルでは、アプリケーションに適用できる、エンドツーエンドのワークフローを説明します。</p> <ul style="list-style-type: none">• アプリケーションのパフォーマンスの問題を特定• 通信ホットスポットを特定• アプリケーションのレビュー <p>このチュートリアルでは、ポワソン・アプリケーションのサンプルを使います。アプリケーションのソースコードは、製品のインストール先に格納されています。</p> <pre><installdir>/examples/poisson</pre>
所要時間	10-15 分
目的	<p>このチュートリアルでは、以下のトピックについて説明します。</p> <ul style="list-style-type: none">• MPI Performance Snapshot を使用して、アプリケーションを素早く解析します• アプリケーションのパフォーマンスを改善します
その他の情報	MPI Performance Snapshot の詳細については、『 MPI Performance Snapshot ユーザーズガイド 』(英語)をご覧ください。

1.1. 必要条件

MPI Performance Snapshot を使用する前に、必要なソフトウェア/ライブラリーがインストールされ、環境が設定されていることを確認してください。

1. インテル® Fortran コンパイラー 15.0.1 以降をインストールして、環境を設定します。

```
$ source <compiler_installdir>/bin/compilervars.sh
```

2. インテル® MPI ライブラリー バージョン 5.0.3 以降をインストールして、環境を設定します。

```
$ source <IMPI_installdir>/intel64/bin/mpivars.sh
```

3. MPI Performance Snapshot 向けの環境を設定するため、スクリプトを source します。

```
$ source <ITAC_installdir>/bin/mpsvars.sh
```

2. MPI アプリケーションの解析

ステップ 1: 解析の準備	アプリケーションをビルドして、統計ファイルを生成します。
ステップ 2: パフォーマンス問題を特定	<ul style="list-style-type: none">• Summary ページを参照します。• Function Summary ページを参照します。
ステップ 3: 問題の解決	<ul style="list-style-type: none">• ブロッキング通信を行う SendRecv 関数を非ブロッキングの lcomm 関数に置き換えます。• Allreduce 関数をチューニングします。
ステップ 4: ワークの確認	アプリケーションを再ビルドして結果を参照します。

2.1. 解析の準備

[必要条件](#)の節で説明したステップを完了します。

<installdir>/examples/poisson ディレクトリーの内容を作業ディレクトリーにコピーします。

inp ファイルを次のように変更します。

```
3200  
2 16
```

make コマンドでアプリケーションをビルドし、2 ノードのクラスターでアプリケーションを実行します。

```
$ make  
$ mpirun -mps -n 32 -hosts <node1>,<node2> ./poisson
```

2 つの統計ファイルが生成されます: stats.txt と app_stat.txt。



2.2. アプリケーションの統計情報を見る

2.2.1. Summary ページを参照する

mps コマンドで生成されたファイル进行处理して、アプリケーションのサマリーを表示します。

```
$ mps ./stats.txt ./app_stat.txt
```

結果は以下のようになります。

```
| Summary information (サマリー情報)
|-----
| Application (アプリケーション): ./poisson
| Number of ranks (ランクス数): 32
| Used statistics(使用された統計ファイル): app_stat.txt, stats.txt
|
| WallClock time (ウォールクロック時間):          6.37 sec
| Total application lifetime (アプリケーションの総時間). The time is elapsed time for the slowest process (この時間は
| 最も遅い プロセスの経過時間です).
| This metric is the sum of the MPI Time and the Computation time below (このメトリックは、MPI 時間と以下の計算
| 時間の合計です).
|
| MPI Time (MPI 時間):                2.36 sec          37.56%
| Time spent inside the MPI library (MPI ライブラリー内部で費やされた時間) High values are usually bad
| (通常高い値が悪い傾向を示します。). High values are usually bad.
| This value is HIGH (この値は HIGH です). The application is Communication-bound (アプリケーションは
| 通信依存です).
| This might be caused by (次の原因が考えられます):
| - High wait times inside the library - see the MPI Imbalance metric below
|   (ライブラリー内部の高い待機時間 - 以下の MPI インバランス・メトリックを参照).
| - Active communications - see the diagrams 'MPI Time per Rank' (key '-m' or '-m -D' for per MPI-
|   function details) & 'Collective Operations Time per Rank' (key '-t' or '-t -D' for per MPI-function
|   details) (アクティブな通信 - ダイアグラムの MPI Time per Rank(MPI 関数ごとの詳細 -m または -m -D
|   キー) と Collection Operations Time per rank (MPI 関数ごとの詳細向けの -t または -t -D キーを参照).
| - Unoptimized settings of the MPI library (最適化されていない MPI ライブラリーの設定). You can tune
|   Intel(R) MPI Library for your application and cluster configuration using the mpitune utility available as
|   part of the library package (ライブラリーのパッケージに含まれる mpitune ユーティリティを使用して、
|   アプリケーションとクラスター向けのインテル® MPI ライブラリーの設定をチューニングできます).
|
| MPI Imbalance (MPI インバランス):          2.34 sec          37.24%
| Mean unproductive wait time per-process spent in the MPI library calls when a process is waiting for
| data. This time is part of the MPI time above. High values are usually bad. This value is HIGH.
| The application workload is NOT well balanced between MPI ranks (MPI ランク間のアプリケーション・
| ワークのバランスが取れていません).
| For more details about the MPI communication scheme use Intel(R) Trace Analyzer and Collector
| available as part of Intel(R) Parallel Studio XE Cluster Edition (詳細は、インテル® Parallel Studio XE
| Cluster Edition に同梱されるインテル® Trace Analyzer & Collector を使用して、MPI 通信のスキームを
| ご覧ください).
|
| ...
```



MPI アプリケーションの解析

MPI 時間と MPI インバランス時間の値が非常に近いのが分かります。これは、アプリケーションのワークが少なく最適化すべきことを示します。

次のステップでは、最も時間のかかる関数を特定するため関数のサマリーを確認します。

重要な用語

[MPI インバランス](#)

2.2.2. Function Summary (関数サマリー) を参照する

関数サマリーを見るには、mps コマンドに -f オプションを追加して統計ファイル进行处理します。

```
$ mps -f ./stats.txt ./app_stat.txt
```

結果は以下のようになります。

Function	Time(sec)	Time(%)	Volume(MB)	Volume(%)	Calls
SendRecv	50.37	59.96	39.06	99.96	9200
Allreduce	24.20	28.81	0.01	0.03	1600
Init	8.83	10.51	0.00	0.00	32
Bcast	0.55	0.66	0.00	0.00	32
Gather	0.03	0.06	0.00	0.01	32
Finalize	0.00	0.00	0.00	0.00	32
TOTAL	84.01	100.00	39.08	100.00	10928

SendRecv と Allreduce 関数は、潜在的なホットスポットであり最適化が必要であることを確認できます。

重要な用語

[Hotspot \(ホットスポット\)](#)

2.3. 問題の解決

2.3.1. SendRecv 関数をチューニング

ポワソン・アプリケーションのパフォーマンスを改善するには、ブロッキング SendRecv 関数を非ブロッキング lcomm 関数に置き換えます。

変更が必要なソースコードは、poisson ディレクトリーにあります。pardat.f90_icommm ファイルを pardat.f90 にリネームします。



2.3.2. Allreduce 関数をチューニング

インテル® MPI ライブラリーの引数を使用して、Allreduce 関数をチューニングできます。
I_MPI_ADJUST_ALLREDUC 環境変数に適切な値を設定します。ポワソン・アプリケーションの最適な値は 2 であることが分かります。しかし、値はアプリケーションや設定に依存するため、最適な値は手動でチェックする必要があります。

2.4. 結果を表示

結果を見るには、アプリケーションを再ビルドして、統計ファイルを再度生成します。最適により、次のような結果を得られました:

- アプリケーションのライフタイム時間は、6.37 秒から 5.72 秒 (~9%) に減少しました。
- MPI 時間レートは、37.56% から 30.28% (7.28%) に減少しました。

以下のダイアグラムをご覧ください。

```
| Summary information (サマリー情報)
|-----
| Application (アプリケーション): ./poisson
| Number of ranks (ランズ数): 32
| Used statistics (使用する統計情報): app_stat.txt, stats.txt
|
| WallClock time(ウォールクロック時間): 5.72 sec
| Total application lifetime (アプリケーションの総時間). The time is elapsed time for the slowest process (この
| 時間は最も遅いプロセスの経過時間です).
| This metric is the sum of the MPI Time and the Computation time below (このメトリックは、MPI 時間と以下
| の計算時間の合計です).
|
| MPI Time (MPI 時間): 1.70 sec 30.28%
| Time spent inside the MPI library (MPI ライブラリー内部で費やされた時間).
| High values are usually bad (通常高い値が悪い傾向を示します).
| This value is HIGH (この値は HIGH です). The application is Communication-bound
| (アプリケーションは通信依存です).
| This might be caused by (次の原因が考えられます):
| - High wait times inside the library - see the MPI Imbalance metric below (ライブラリー内部の高い待機
| 時間 - 以下の MPI インバランス・メトリックを参照).
| - Active communications - see the diagrams 'MPI Time per Rank' (key '-m' or '-m -D' for per MPI-
| function details) & 'Collective Operations Time per Rank' (key '-t' or '-t -D' for per MPI-function
| details)(アクティブな通信 - ダイアグラム 'MPI Time per Rank' (MPI 関数のごとの詳細向け '-m' または
| '-m -D' キー) と 'Collective Operations Time per Rank' (MPI 関数のごとの詳細向け '-t' または '-t -D' キー)).
| - Unoptimized settings of the MPI library (最適化されていない MPI ライブラリーの設定). You can tune
| Intel(R) MPI Library for your application and cluster configuration using the mpitune utility available as
| part of the library package (ライブラリーのパッケージに含まれる mpitune ユーティリティを使用して、アプリケーションとクラスター向けのインテル® MPI ライブラリーの設定をチューニングできます).
```




MPI アプリケーションの解析

MPI Imbalance (MPI インバランス): 1.66 sec 29.49%

Mean unproductive wait time per-process spent in the MPI library calls when a process is waiting for data (プロセスがデータを待機する時に、MPI ライブラリー呼び出しでプロセスごとに費やされる非生産的な待機時間). This time is part of the MPI time above (この時間は上記の MPI 時間の一部です). High values are usually bad (通常高い値が悪い傾向を示します). This value is HIGH (この値は HIGH です). The application workload is NOT well balanced between MPI ranks (MPI ランク間のアプリケーション・ワークのバランスが取れていません). For more details about the MPI communication scheme use Intel(R) Trace Analyzer and Collector available as part of Intel(R) Parallel Studio XE Cluster Edition (詳細は、インテル® Parallel Studio XE Cluster Edition に同梱されるインテル® Trace Analyzer & Collector を使用して、MPI 通信のスキームをご覧ください).

結果は以下のようになります。

```
Function summary for all ranks
-----
| Function          | Time(sec) | Time(%) | Volume (MB) | Volume(%) | Calls |
|-----|-----|-----|-----|-----|-----|
| Recv              | 26.94     | 42.94   | 138.53      | 78.00     | 9200  |
| Allreduce         | 25.76     | 41.06   | 0.01        | 0.01      | 1600  |
| Init              | 9.06      | 14.44   | 0.00        | 0.00      | 32    |
| Broadcast         | 0.61      | 0.97    | 0.00        | 0.00      | 32    |
| Send              | 0.31      | 0.49    | 39.06       | 21.99     | 9200  |
| Gather            | 0.04      | 0.07    | 0.00        | 0.00      | 32    |
| [skipped 2 lines]
|-----|-----|-----|-----|-----|
| TOTAL             | 62.74     | 100.00  | 177.60      | 100.00    | 23328 |
```



3. 重要な用語

MPI Imbalance (MPI インバランス):プロセスが MPI 呼び出しでデータの待機に費やす非生産時間を示します。

Hotspot (ホットスポット): 実行に長い時間を要しているコード領域。ホットスポットがボトルネックを示している場合は排除できますが、そうでないホットスポットは処理の性質上実行に長い時間を必要とするものです。