インテルのFortranへの取り組み

インテル株式会社ソフトウェア＆ソリューションズグループ

池井　満

Agenda

- How does Intel support coarray and will enhance it in the future?
- What is Intel thinking for support for many core CPU now and in the future?
- Where will Intel drive/take Fortran people to?
- What's New in Intel® Fortran 16.0

How does Intel support coarray and will enhance it in the future?

Intel Fortran's coarray support is based on Intel MPI

There are three driving forces behind coarray enhancements:

a. The draft Fortran 2015 Standard will have many new coarray features, like teams, atomics, collectives, events, and failed images, all based on the "almost finished" Technical Specification 18508. Intel Fortran will implement these coarray features plus the rest of Fortran 2015 in future releases.

b. Continuous improvement of coarray performance is essential to customers and the continued future success of coarrays. Much of that improvement will come in future releases from optimizing generated code that handles coarrays, like removing unnecessary locks, handling contiguous data in large chunks, and faster synchronization of images. Intel MPI currently supports the MPI 3 standard which contains features that may help us make other improvements in performance in future releases.

c. We are investigating the possibility that, in future releases, Intel Fortran could also support 3rd-party MPI implementations, like OpenMPI, MPICH, IBM MPI, or SGI MPI, as the base for coarrays.

What is Intel thinking for support for many core CPU now and in the future?

Many-core CPU support

Intel Fortran will continue to support Intel's many-core, multi-core, and Xeon Phi CPUs through OpenMP constructs like TARGET and Intel directives like OFFLOAD.

Intel Fortran's parallel support is a hierarchy of features from finest to coarsest grain:

a. DO CONCURRENT from the Fortran 2008 Standard

b. Auto-parallel and PARALLEL, VECTOR, and SIMD directives

c. Coarrays

d. OpenMP

e. MPI

All of these features are areas for improvement in future releases.

Where will Intel drive/take Fortran people to?

Where to drive

Intel Fortran will continue to support new and emerging Fortran and OpenMP standards in future releases while supporting deprecated and deleted standard features and legacy features from VAX FORTRAN and Digital and Compaq Visual Fortran. Intel specific Fortran features may be added in future releases when hardware, operating system, or optimizer needs require them.

What's New in Intel® Fortran 16.0

New and Changed Features in Fortran 16.0

- Submodules features from Fortran 2008
- Further C Interoperability from Fortran 2015
- OpenMP 4.1 TARGET ENTER | EXIT DATA
- OpenMP 4.1 TARGET NOWAIT | DEPEND
- !DEC$ BLOCK_LOOP directive
- -init enhancements
- -fpp-name option
- VS2013 Shell

Submodules (Fortran 2008)

The Problem:

- Any edit to a module, no matter how trivial, requires recompilation of all sources that USE that module, directly or indirectly
- Can cause a “recompilation cascade” in builds, greatly lengthening build time

The Solution:

- Submodules separate interface from implementation
- Changes in a submodule don’t force recompile of module or sources that use the module (unless interfaces change)

Further C Interoperability (Fortran 2015)

TS29113 on "Further Interoperability of Fortran with C" to be part of Fortran 2015. Motivations include:

- Support the needs of MPI3
- Provide Fortran equivalent of C's "`void*`" – assumed type and rank
- Enable C code to see array bounds, manipulate pointers and allocatables
- Extend interoperable interfaces to ALLOCATABLE, POINTER, assumed shape, CHARACTER(*) - all passed by new "C Descriptor"
- OPTIONAL allowed in interoperable interface
- Extend ASYNCHRONOUS beyond I/O
- Relax restrictions

EXIT from BLOCK (Fortran 2008)

- When we first implemented BLOCK in Fortran 15, we didn't support EXIT from a BLOCK – now we do
- EXIT from other named constructs still in the future

```
outer: block
do i = 1, num_in_set
if ( x == a(i) ) exit outer
end do
call r
end block outer
```

New and Changed Features in Fortran 16.0

- Submodules features from Fortran 2008
- Further C Interoperability from Fortran 2015
- OpenMP 4.1 TARGET ENTER | EXIT DATA
- OpenMP 4.1 TARGET NOWAIT | DEPEND
- !DEC$ BLOCK_LOOP directive
- -init enhancements
- -fpp-name option
- VS2013 Shell

intel®

構想

- 数コアのプロセッサから始まって一貫したモデル、言語、ツールや手法でメニーコアに対応することで、持続的な価値を生み出すことができる。
- アプリケーションは利用できるすべての並列性を活用する。
  - 命令レベル：コアの特性を知り、考慮
  - データレベル：SIMD命令を用いるようにベクトル化
  - スレッドレベル：OpenMP等の標準ツールで並列化
  - クラスタレベル：MPI等の標準ツールで並列化
- 専門家がプロセッサに最適化した標準ライブラリや言語を利用する。
- ヘテロジニティまで考慮した最適化を検討する。

高速なコードを迅速に開発
インテル® Parallel Studio XE 2015

高速なコード

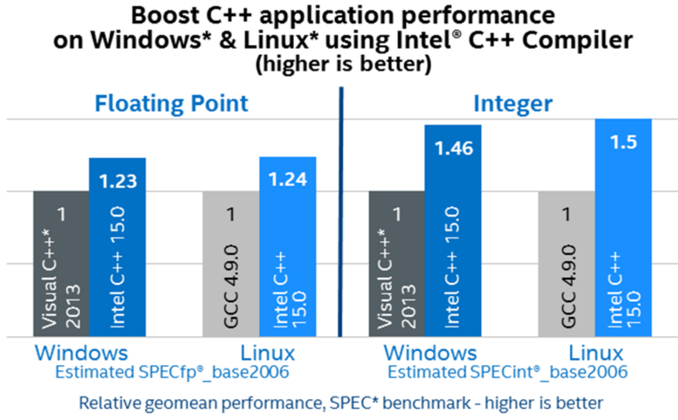
- 明示的なベクトル・プログラミングでより多くのコードをスピードアップ
- インテル® Xeon Phi™ コプロセッサー、Skylake † およびBroadwell † マイクロアーキテクチャー向けの最適化
- 最新のMPI-3 標準をサポートするインテル® MPI ライブラリー
- 小さな行列と大きな演算問題を高速処理
- クラスタ用の並列疎行列ソルバー

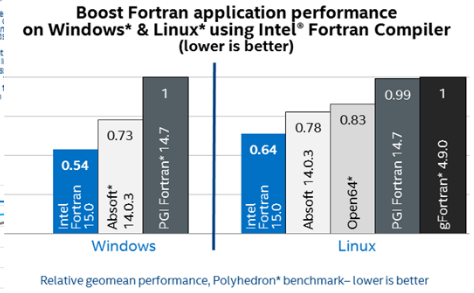
迅速な開発

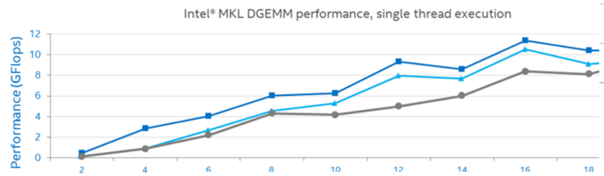
- 総合的なコンパイラーによる最適化レポート
- Mac* 上でもWindows* またはLinux* のプロファイル・データを解析

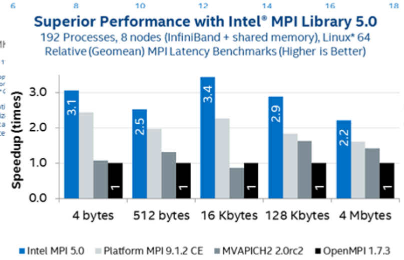
最新の標準規格をサポート

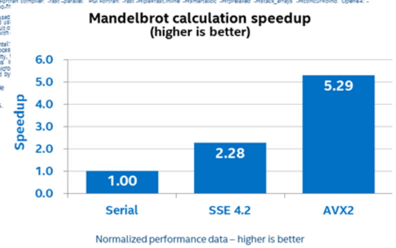
- MPI-3, OpenMP 4, C++11およびFortran 2003 (フルサポート)









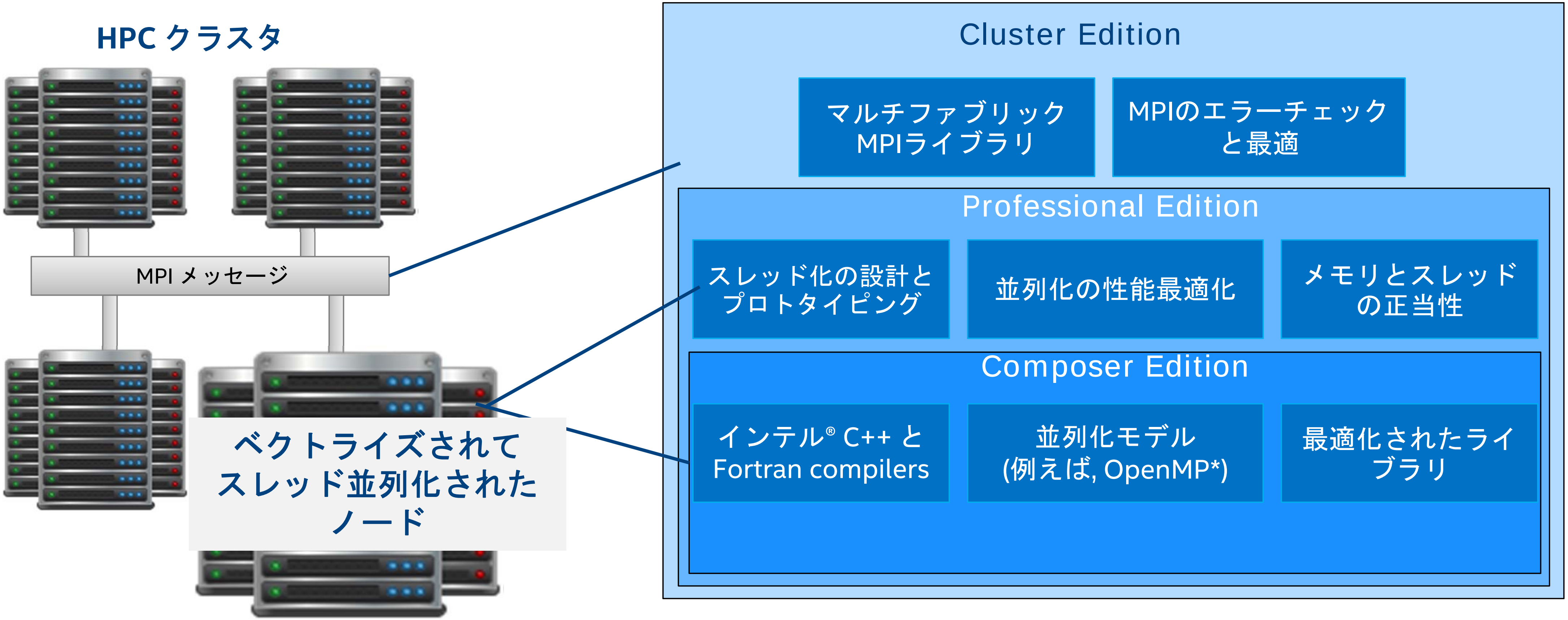


インテルの並列化プログラミングモデル

PSXE 2015 で提供され，開発者が選択

言語と実行時システムの拡張	分野に特化したライブラリ	標準化されたツール	その他の開発ツール
Intel® compiler directives for offload and vectorization ループのベクトル化を促進したりプログラムの一部をIntel® Xeon Phi™ コプロセッサにオフロードするためのスイッチや指示文	**Intel® Integrated Performance Primitives (Intel® IPP)** 暗号化，データ圧縮，信号処理やマルチメディアの処理等，複数の分野にわたる汎用関数群	**Message PassingInterface (MPI)** 分散メモリ環境化実行する並列プログラムで広く使われる送信／受信モデルを主にした標準ライブラリ	**Open Computing Language (OpenCL)** is ヘテロジニアスな計算環境を使うための C99-ベース言語標準
Intel® Cilk™Plus 新しいkeywordや表記法並列性の記述や配列の並列処理の実行を可能にするC/C++ 言語の拡張	**Intel® Math Kernel Library (Intel® MKL)** HPCコミュニティで用いられる 数学関数群． BLAS, LAPACK, ScalaPACK1, space solvers, FFT, とベクトルの数学関数を含む	**Open Multi-Processing (OpenMP)**l 指示文をベースにしたC, C++ やFortranに順次書き加えて並列化するための標準化言語拡張	
Intel® Threading Building Blocks (TBB) タスク，並列アルゴリズムやコンテナを提供するC++ のtemplate library		**CoarrayFortran** Fortran 2008 標準の機能の一つで，Fortranで並列実行を行えるようにする拡張	

どのように Intel® Parallel Studio XE 2015 は HPCの“高速なコードの迅速な開発”を行うか



構成

インテル® Parallel Studio XE 201 Composer Edition	インテル® Parallel Studio XE 201 Professional Edition	インテル® Parallel Studio XE 201 Cluster Edition
インテル® C++ コンパイラー インテル® Fortranコンパイラー インテル® TBB インテル® IPP インテル® MKL インテル® Cilk™ Plus インテル® OpenMP*	Iインテル® C++ コンパイラー インテル® Fortranコンパイラー インテル® TBB インテル® IPP インテル® MKL インテル® Cilk™ Plus インテル® OpenMP*	インテル® C++ コンパイラー インテル® Fortranコンパイラー インテル® TBB インテル® IPP インテル® MKL インテル® Cilk™ Plus インテル® OpenMP*
	インテル® Advisor XE インテル®Inspector XE インテル®VTune™ Amplifier XE	インテル®Advisor XE インテル®Inspector XE インテル®VTune™ Amplifier XE インテル®MPI Library インテル®Trace Analyzer and Collector
アドオン: ローグウェーブIMSL* ライブラリー	アドオン: ローグウェーブIMSL* ライブラリー	アドオン: ローグウェーブIMSL* ライブラリー

フローティング・ライセンスおよびアカデミック・ライセンスを含むその他の構成については、http://intel.ly/perf-toolsを参照してください。

Legal Disclaimer & Optimization Notice

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804