スーパーコンピュータ「富岳」向けのアプリケーション開発環境

渡辺　健介　　野瀬　貴史　　鈴木　清文　　千葉　修一

あらまし

スーパーコンピュータ「富岳」（以降,「富岳」）の開発の目標は，スーパーコンピュータ「京」に対して最大100倍のアプリケーション実行性能を実現すること，そして幅広いアプリケーションの動作に対応する汎用性を確保することである。そのためには，ハードウェアの性能を最大限に引き出すアプリケーションの開発から実行までを支援する，統合的な開発環境が必要であるが，実現には大きな課題があった。まず，コンパイラにおいては,「富岳」で採用したCPU向けの最適化開発と様々なアプリケーションへの対応が必要となる。また，MPI（Message Passing Interface）通信ライブラリーにおいては，新ハードウェアおよび総プロセス数の増大によるメモリー使用量への対応が必要となる。更に，アプリケーション開発支援ツールにおいては，実用性の向上が挙げられる。

本稿では，アプリケーション開発環境に対するこれらの取り組みについて述べる。

1. まえがき

スーパーコンピュータ「富岳」（以降,「富岳」）[1] は，Arm Limited社の命令セットアーキテクチャー（以降，Armアーキテクチャー）を採用して開発した高性能CPU「A64FX」を搭載し，様々なアプリケーションに対応する汎用性や，後述するTofuインターコネクトD（以降，TofuD）による超並列性などを実現する。この「富岳」の開発目標は，スーパーコンピュータ「京」（以降,「京」）[2] に対して最大100倍のアプリケーション実行性能を実現することである。そのため，ハードウェアの性能を最大限引き出すアプリケーションの開発から実行までを支援する，統合的な開発環境が必要である。開発した「富岳」は，ISC 2020 [3] で発表されたスーパーコンピュータの性能を示すランキングにおいて，TOP500 [4]，HPCG [5]，HPL-AI [6]，Graph500 [7] の4部門で世界1位を獲得し，高い性能を証明している。

本稿では,「富岳」を対象としたアプリケーション開発環境のコンパイラ，MPI（Message Passing Interface）通信ライブラリー，アプリケーション開発支援ツールの新機能や性能改善への取り組みについて紹介する。

2. コンパイラ

本章では,「富岳」向けコンパイラの技術開発について述べる。

「富岳」向けコンパイラは，Fortran/C/C++の3種類の言語に対応する。そして，その開発目標は,「富岳」で重点的に取り組むべき社会的・科学的課題に関するアプリケーション（以下，重点課題アプリ）について,「京」に対して100倍の実行性能を達成すること，そして様々なアプリケーションを動作可能にすることである。この目標を実現するためには，A64FXの特性を理解し，適した機能を強化しなければならない。アプリケーションとの協調設計（コデザイン）によって抽出した課題は，次の2点である。

・Armアーキテクチャーに対応した最適化の強化

・近年利用が拡大しているオブジェクト指向型言語や整数型演算の最適化への対応

以下では，これらの課題に対する取り組みを説明する。

2.1 Armアーキテクチャーへの対応

高度なコンパイラの実現には，アプリケーションを高速に動作させるためベクトル命令の適用促進と，命令の並列性を高める最適化の適用が必要不可欠である。

まず，ベクトル命令の利用促進として,「富岳」で新たに採用されたSVE（Scalable Vector Extension）を活用する。SVEでは，ベクトル命令の各要素に対して演算の実行有無を指示する，プレディケートレジスタが利用可能になった。これによって，IF文を含むような複雑なループのベクトル化が可能になり，高速に実行することが可能になった。

次に，命令の並列性を高める最適化として，富士通コンパイラの強みであるソフトウェアパイプライニング（SWP）が重要となる。「京」には128本のベクトルレジスタがある一方で,「富岳」では32本と省レジスタ化している。このような「富岳」であっても，多くのレジスタを必要とするSWPを効果的に動作させるために，ループ分割最適化の強化を行った。ループ分割は，文数の多いループをレジスタ数，メモリーアクセス数などを考慮して，SWPが適用できるループに分割する最適化である。図-1に重点課題アプリの一つであるNICAM [8] の物理過程カーネル，および力学過程カーネルの性能評価結果を示す。前者は計算量や分岐処理が多い

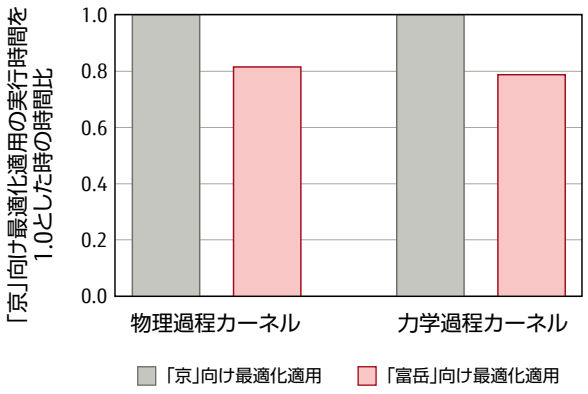

図-1　NICAMのカーネルにおける最適化効果

ループ構造が特徴であり，後者はメモリースループットが高いことが特徴である。このように性質の異なるカーネルに対しても，ループ分割を適用しSWPを適用することで，約20％の性能向上が実現でき，「京」に対して100倍のアプリ実行性能の目標達成に向けて大きく寄与した。

2.2 オブジェクト指向型言語や整数型演算への対応

「富岳」では利用者の裾野を広げるために，従来のシミュレーションプログラム以外にも様々な分野のアプリケーションを実行できることが目標とされた。そのため，整数系演算やオブジェクト指向プログラムへの最適化を強化する必要があった。そこで，C/C++言語のプログラムの高速化に実績があるOSS（Open Source Software）コンパイラのClang/LLVM [9] をベースとして採用し，富士通がこれまで培ってきたHPC向け最適化技術を移植した。更に，「京」で使われていたプログラムや伝統的なプログラムとの互換を確保するために従来の機能を併用可能とし，「京」からの容易な移植を可能としている。

3. 通信ライブラリー

HPC（High-Performance Computing）分野の高並列アプリケーションで用いられる通信API（Application Programming Interface）はMPIがデファクトスタンダードとなっており，MPIを実装した通信ライブラリーの性能はHPCシステムの実用性に大きく影響する。「京」では，MPI規格を実装したOSSの一つであるOpen MPIをベースとして独自に改良を加え，高い性能が得られた。「富岳」でもこれを継承しているが，新ハードウェアへの対応，および総プロセス数の増大によるメモリー使用量に課題があった。

本章では，この課題に対する「富岳」向け通信ライブラリーの取り組みについて述べる。

3.1 新ハードウェアへの対応

「京」では，多数のプロセッサを高次元で接続する6次元メッシュ/トーラス構成のノード間インターコネクトであるTofu（Torus Fusion）インターコネクト（以降, Tofu）[10] を採用した。更に「富岳」では，Tofuと比べて同時通信数，耐故障性，バリア同期通信などを強化したTofuインターコネクトD [11] を採用した。

「富岳」に搭載されるTofuDは，「京」のTofuに比べて1リンクあたりの通信速度が5 GB/sから6.8 GB/sに強化されている。しかし，その向上率は計算性能の向上率には及ばないため，各ランクが協調してデータの放送や縮約演算を行う集団通信アルゴリズムがそのままでは計算性能を最大限に活かすことができない。「富岳」では，同時通信可能リンクを「京」の4本から6本に増加させているため，これを活用して集団通信アルゴリズムを改良し，TofuDの通信性能をフルに活用することが重要である。「富岳」では，五つの集団通信関数MPI_Bcast, MPI_Reduce，MPI_Allreduce，MPI_Allgather, MPI_Alltoallにおいて，6本同時通信化が行われている。特に，大メッセージ長向けのMPI_Bcastの従来アルゴリズムは同時通信数が3本に制約されていたため，6本への強化で大きく通信バンド幅が向上している（図-2）。

また，TofuDではバリア通信 [12] の加速に用いられる通信機能が強化されている。これを活用し，バリア通信の加速をノード内複数プロセスによる通信が発生するパターンにも適用可能となっている。TofuDのバリア通信（以降，Tofuバリア通信）は「京」に比べ，適用可能なデータ長が1要素から3要素（浮動小数点数）または6要素（整数）に拡大している。ソフトウェア実装に比べ大幅に高速であるため，ハードウェアが対応するこれらの要素数を超える範囲でも，ソフトウェアで繰り返しTofuバリア通信を呼び出すことで，純粋なソフトウェア実装より更に高速な通信を実現できる範囲がある（図-3）。繰り返しTofuバリア通信を行うオプション機能をMPIライブラリーに付加することで，ユーザーがプログラム改変なしに適用データ長を拡大する最適化を利用できるようにしている。

3.2 メモリー使用量のコントロール

「京」では，1ノードにメモリーが16 GiB搭載されており，また1プロセスが全ての領域を専有して

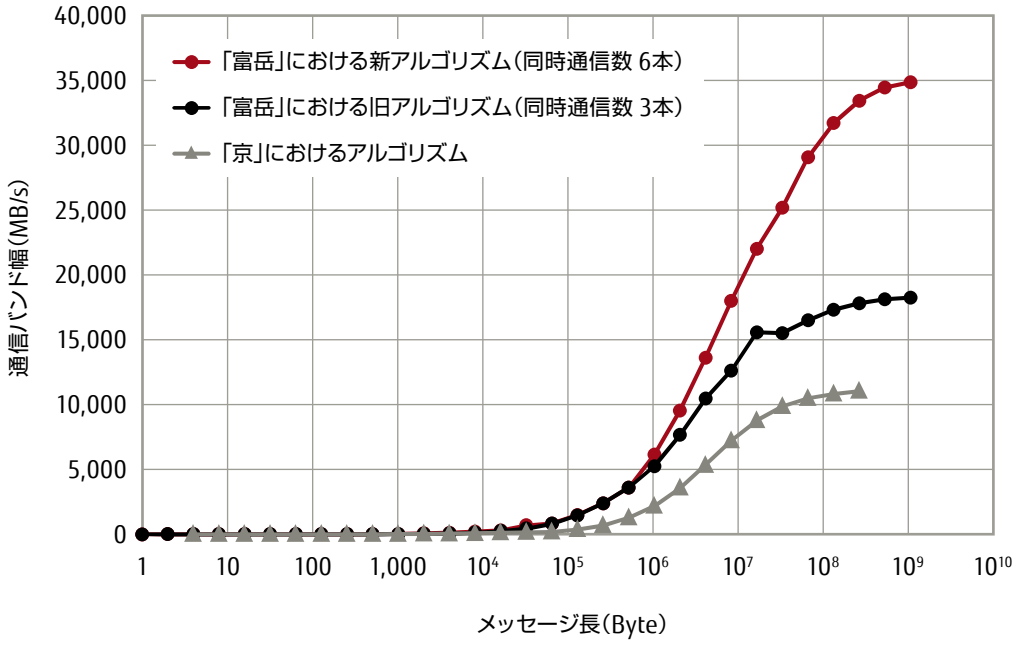

図-2　6方向対応した専用集団通信アルゴリズムの性能（Bcast）

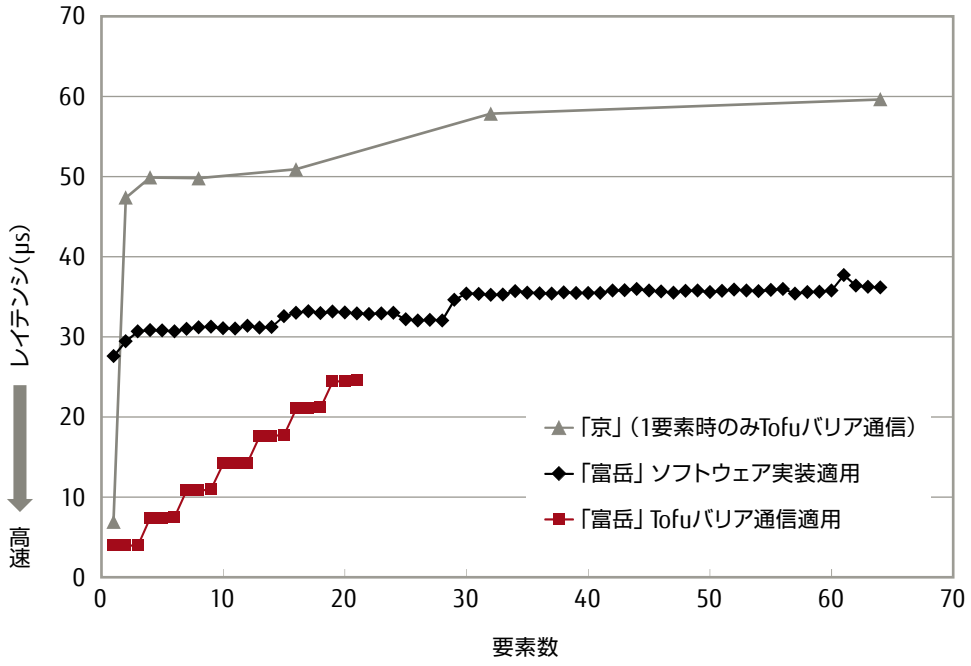

図-3　バリア通信の高速化（MPI_Allreduce）

いた。しかし「富岳」では，1ノードのメモリー容量は32 GiBに増強されたものの，CMG（Core Memory Group）と呼ばれるノード内構造により4分割されている。そして，複数のプロセスをそれぞれ別のCMGに割り当てることが推奨されるため，「富岳」では32 GiBを4プロセスで分け合うこととなる。このため，1プロセスあたりのメモリー使用量は8 GiBに減少している。更に，「富岳」のシステム全体のプロセス数で見た規模は，約15万のノードにノード内4プロセスが掛け合わされるため，

トータルで約60万プロセスとなり，「京」の約8万プロセスから大幅に増大している。一般に，MPI通信の接続相手となるプロセスの数に比例して，MPIライブラリー内部で必要な管理情報も増加するため，システム全体の規模の増大はメモリー容量に対する圧迫要因となり，アプリケーション性能へ大きく影響する。

この問題に対して，「富岳」のMPIライブラリーでは，MPI_Init関数による初期化時に全通信相手分全て確保するのではなく，通信相手ごとに通信を初めて行う時に動的にメモリーを確保するようにした。これによって，単純に全体のプロセス数のみに依存して増加するのではなく，通信パターンに応じて必要最小限に抑えられるよう改善している。また，TofuDの機能を活用した更にメモリー消費が少ない通信モードを実装することで，プロセス規模とメモリー消費量の両立が必要なアプリケーションに対する柔軟性が向上している。この通信モードでは，デフォルトの通信モードと比較して，27,648ノード110,592プロセス時の最大メモリー消費量を，約34％削減することが可能である。

4. アプリケーション開発支援ツール

本章では，「富岳」に向けて開発したアプリケーション開発支援ツールについて述べる。

アプリケーション開発支援ツールは，目的に応じた3種類の機能から構成される。

- 性能チューニングのための情報を収集し，分かりやすく表示するプロファイラ
- 大規模並列処理で発生するデッドロックや異常終了といった事象の調査を支援するための並列実行デバッガ
- GUIを通してコンパイルやジョブ投入などの操作を行うことのできる統合開発環境

以下では，最も多く用いられるプロファイラについて，特徴と「富岳」向けに改善した点について紹介する。

4.1 「富岳」への対応

「富岳」に対応したプロファイラは，実用性の観点で評価の高かった「京」のプロファイラを継承・発展させて開発した。図-4に示すように，「基本プロファイラ」「詳細プロファイラ」「CPU性能解析レポート」から成り，状況に応じてこれらを段階的に使用することで，効率的な性能解析を行うことができる。

4.2 基本プロファイラによる性能傾向把握

基本プロファイラは，サンプリング方式でアプリケーションのコスト分布情報を収集し，手続き単位・ループ単位・行単位で表示することで，アプリケーション全体の性能の概要を把握することを支援する。コスト分布からは，コストが集中しているホットスポットを特定することが可能である。

「京」向けのプロファイラでは，収集した情報を，グラフ作成や統計処理など加工しやすい汎用的なフォーマットで入手したい，という要望が挙がっていた。しかし，実際に提供されたのはテキスト形式とCSV形式のみであり，「富岳」では汎用性の高い出力形式をサポートすることが課題であった。「富岳」のプロファイラでは，基本プロファイラと詳細プロファイラの双方でXML形式の出力をサポートすることで，この課題に対応した。

4.3 詳細プロファイラとCPU性能解析レポートによる詳細情報把握

基本プロファイラの結果から特定されたホットスポットに対し，その部分のより詳細な性能情報を得るために，詳細プロファイラおよびCPU性能解析レポートを使用する。

詳細プロファイラは，特定区間のMPI通信コスト情報やCPU性能解析情報を収集し，ホットスポットの詳細な性能分析を可能とする。CPU性能解析レポートは，CPUに内蔵され演算性能に関するCPUの動作状況を計測するPMU（Performance Monitoring Unit）カウンタの情報を，グラフや表の形式で体系立てて分かりやすく表示する（図-5）。PMUカウンタの情報を分析することにより，演算性能に関するボトルネックを詳細に把握することが可能となる。

「京」では，「富岳」のCPU性能解析レポートと同様の機能を，Excel形式の精密PA可視化機能で実現していた。しかし，情報収集のためにはアプリ

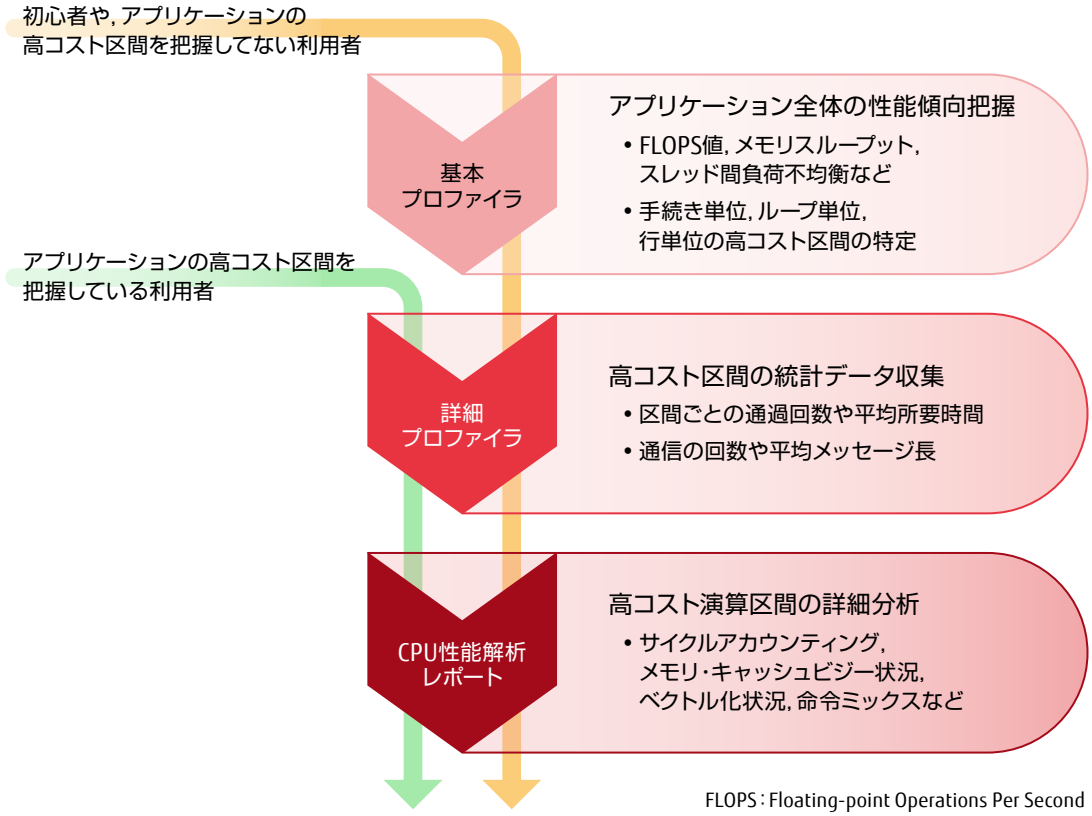

図-4　性能解析の手順

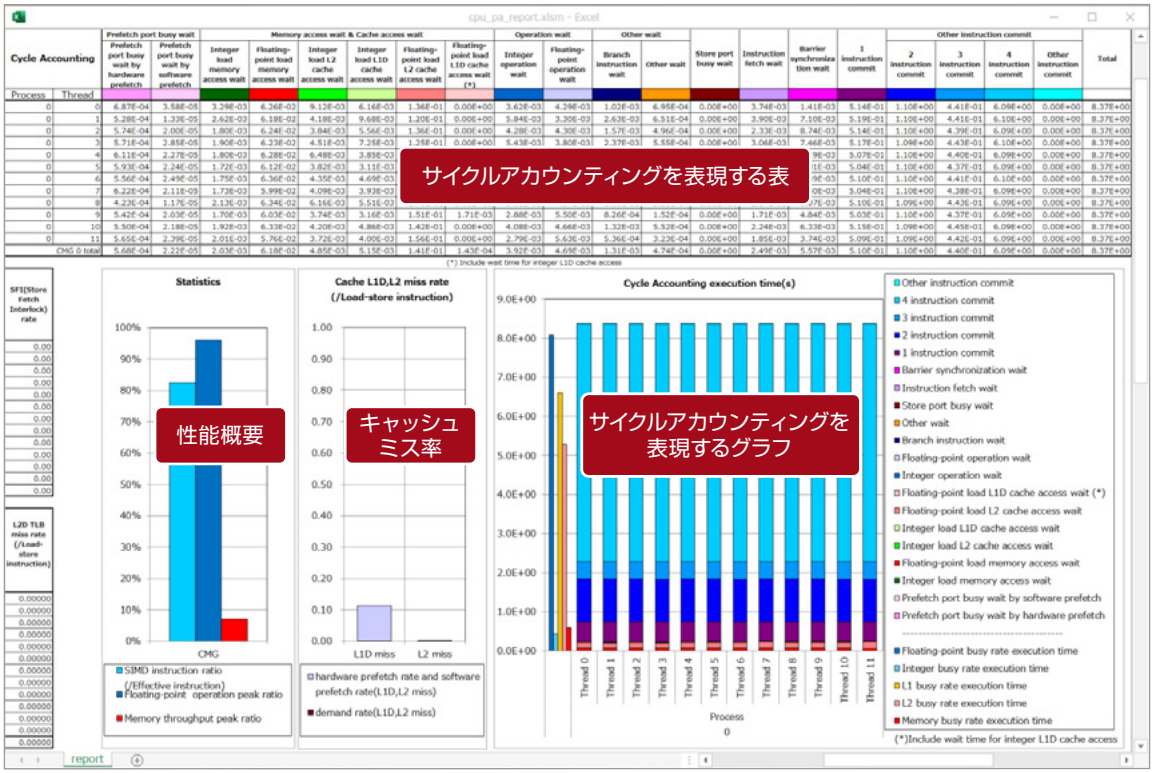

図-5　CPU性能解析レポートの例

ケーションを必ず7回実行する必要があったため，利便性が低いという課題があった。また，「京」の技術を応用した商用機であるPRIMEHPC FX100においても，11回実行する必要があった。

「富岳」のCPU性能解析レポートでは，アプリケーションを1回，5回，11回，または17回実行するごとにレポートを作成することが可能であり，実行回数が多くなるほど情報量が多くなる。そのため，必要な情報量に応じて実行回数を変えるような柔軟な使い方ができるとともに，「富岳」で種類が増えたPMUカウンタの情報を活かして，より詳細な分析が可能となっている。

5. むすび

本稿では，「富岳」をターゲットとしたコンパイラ，MPI通信ライブラリー，アプリケーション開発支援ツールで開発した機能や性能改善への取り組みについて述べた。「京」およびその後継商用機であるPRIMEHPC FX10，FX100，そして「富岳」に至るまでには，アーキテクチャーの変更やハードウェアの変更などがあり，その度にソフトウェアとして大きな変化への対応が求められた。しかし，それぞれのソフトウェアが工夫を取り入れ課題を解決することで，「富岳」の開発目標である「京」に対して100倍のアプリ性能達成に向けて大きく寄与した。

「富岳」は，ISC 2020においてTOP500などで世界1位を獲得した。今後も最先端の研究開発基盤として「富岳」およびその商用機であるPRIMEHPC FX700，FX1000が利用されていくためにはソフトウェアの改善が必要不可欠である。ユーザーの利便性に重点を置きつつ，高性能な機能を提供し，革新的な科学技術の成果につなげていきたい。

参考文献・注記

[1] 理化学研究所が2019年5月に決定したポスト「京」スーパーコンピュータの正式名称。

[2] 理化学研究所が2011年7月に決定したスーパーコンピュータの正式名称。

[3] ISC 2020.
https://www.isc-hpc.com/

[4] 行列計算による連立一次方程式の解法プログラムであるLINPACKの実行性能を指標とし，その結果をもって最も高速なコンピュータシステムの上位500位を示すランキング。

[5] 実際のアプリケーションでよく使われる疎な係数行列から構成される連立一次方程式を解く計算手法である共役勾配法を用いたベンチマークテストのランキング。

[6] LINPACKベンチマークを改良し低精度演算で実施するようにしたベンチマークテストのランキング。

[7] 大規模グラフ（頂点と枝によりデータ間の関連性を示したもの）の解析に関するベンチマークテストのランキング。

[8] Nonhydrostatic ICosahedral Atmospheric Model（非静力学正20面体格子大気モデル）の略称。「富岳」の重点課題アプリの一つとして採用された。力学過程では格子上の風速・気温などを解き，構造格子ステンシル計算やメモリーへの要求が高いループ構造を持つ。物理過程では雲の相変化などの現象を解き，計算量が多く分岐処理が多いループ構造を持つ。

[9] Clang/LLVM.
https://clang.llvm.org/

[10] 富士通：革新的な「6次元メッシュ／トーラス」ネットワーク技術.
https://www.fujitsu.com/jp/about/businesspolicy/tech/k/whatis/network/

[11] Y. Ajima et al.：The Tofu Interconnect D. IEEE International Conference on Cluster Computing, p. 646-654 (2018).

[12] 複数のMPIプロセスの同期を行うための通信。TofuDでは同期のみならず，少量のReduce, Allreduce演算を同時に行うことも可能。

著者紹介

渡辺　健介（わたなべ　けんすけ）
富士通株式会社
プラットフォームソフトウェア事業本部
スーパーコンピュータ「富岳」向けコンパイラの開発に従事。

野瀬　貴史（のせ　たかふみ）
富士通株式会社
プラットフォームソフトウェア事業本部
スーパーコンピュータ「富岳」向けMPI通信ライブラリーの開発に従事。

鈴木　清文（すずき　きよふみ）
富士通株式会社
プラットフォームソフトウェア事業本部
スーパーコンピュータ「富岳」向けアプリケーション開発支援ツールの開発に従事。

千葉　修一（ちば　しゅういち）
富士通株式会社
プラットフォームソフトウェア事業本部
スーパーコンピュータ「富岳」のアプリケーション開発環境向けソフトウェアの開発責任者として従事。