Vol.2010-ARC-192 No.11
Vol.2010-HPC-128 No.11
2010/12/16

# ヘテロ型スーパーコンピュータ TSUBAME 2.0 の Linpack による性能評価

遠藤敏夫[†1,†2] 額田彰[†1,†2] 松岡聡[†1,†3,†2]

Intel プロセッサに加え NVIDIA GPU を備え，2010 年 11 月に稼働開始したヘテロ型スパコンである TSUBAME 2.0 における Linpack ベンチマークの実行について報告する．本システムは 2CPU と 3GPU を備えた計算ノードを約 1400 台持ち，それらはフルバイセクションのファットツリー構造を持つ QDR InfiniBand ネットワークにより接続される．理論演算性能は TSUBAME 1.0 の約 30 倍となる 2.4PFlops であり，それを TSUBAME 1.0 とほぼ同じ規模の電力で実現している．Linpack ベンチマークのコード改良およびチューニングを GPU を用いた大規模システムの特性に合わせ行い，実行速度として 1.192PFlops を実現した．この結果は日本のスパコンとしては初めて PFlops を超えるものであり，Top500 スパコンランキングに 4 位にランクされた．

# Performance Evaluation of TSUBAME 2.0 Heterogeneous Supercomputer with Linpack Benchmark

Toshio Endo,[†1,†2] Akira Nukada[†1,†2]
and Satoshi Matsuoka [†1,†3,†2]

We report Linpack benchmark results on the TSUBAME 2.0 supercomputer, a large scale heterogenous system with Intel processors and NVIDIA GPUs, operation of which has started in November 2010. The main part of this system consists of about 1400 compute nodes, each of which is equipped with two CPUs and three GPUs. The nodes are connected via full bisection fat tree network of QDR InfiniBand. The theoretical peak performance reaches 2.4PFlops, 30 times larger than that of the predesessor TSUBAME 1.0, while its power consumption is similar to TSUBAME 1.0. We conducted improvement and tuning of Linpack benchmark considering characteristics of large scale systems with GPUs, and achieved Linpack performance of 1.192PFlops. This is the first result that exceeds 1PFlops in Japan, and ranked as 4th in the latest Top500 supercomputer ranking.

## 1. はじめに

東京工業大学学術国際情報センターでは 2010 年 11 月にスーパーコンピュータ TSUBAME 2.0 の運用を開始した[8),9)]．このシステムは 2006 年稼働開始した TSUBAME 1 の後継であり，当初の TSUBAME1(1.0 と呼ばれる) の約 30 倍である 2.4PFlops の理論演算性能を持つ，日本初のペタフロップスの性能を実現したシステムである．その浮動小数演算性能・電力効率の向上は最新世代の GPU アクセラレータである NVIDIA Tesla M2050 によるところが大きい．さらに，フルバイセクションファットツリー構造のネットワーク，7.1PB の (raw) 容量の並列ファイルシステム，ノードローカルストレージとしての SSD の採用，水冷の Modular cooling system (MCS) による高効率な冷却などの特徴を持つ．

本稿では TSUBAME 2.0 上の Linpack ベンチマーク実行について報告する．Linpack は密行列連立一次方程式を部分ピボッティングを用いたガウス消去法で解くベンチマークであり，スーパーコンピュータのランキングである Top500[3)] のランク決定に使われることでも知られる．用いた手法は我々が TSUBAME1 用に開発したアルゴリズム[4),5),7)] に基づいたものであり，実装は High-performance Linpack[6)] を改造する形で行った．その実装は GPU の演算性能を有効活用するために，カーネル演算，MPI 通信および PCI-Express 通信のオーバラップを行っている．TSUBAME 2.0 の 1357 ノード，4071GPU を用いたときの実行速度は 1.192PFlops となり，2010 年 11 月の Top500 では世界 4 位にランクされた．ピーク演算性能 (1357 ノードで 2.288PFlops) に対する比は 52.1% であり，ピークとの差についての解析についても報告する．

## 2. TSUBAME 2.0 の概要

TSUBAME 2.0 では，1400 ノード以上の計算ノードと，合計 7.1PBytes のストレージが QDR InifiniBand により接続されている (図 1)．計算ノードは 1408 台の Thin ノード，24 台の Medium ノード，10 台の Fat ノードから成る．本論文の実験では Thin 計算ノード

†1 東京工業大学
Tokyo Institute of Technology
†2 JST, CREST
†3 国立情報学研究所
National Institute of Informatics

Vol.2010-ARC-192 No.11
Vol.2010-HPC-128 No.11
2010/12/16

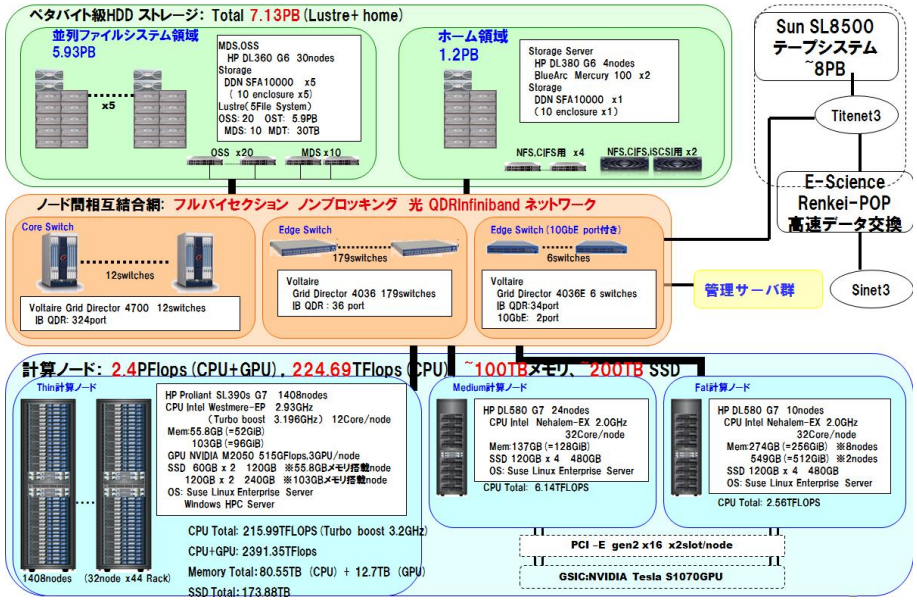


**図 1** TSUBAME 2.0 の全体構成図

**図 2** Thin 計算ノードの外観

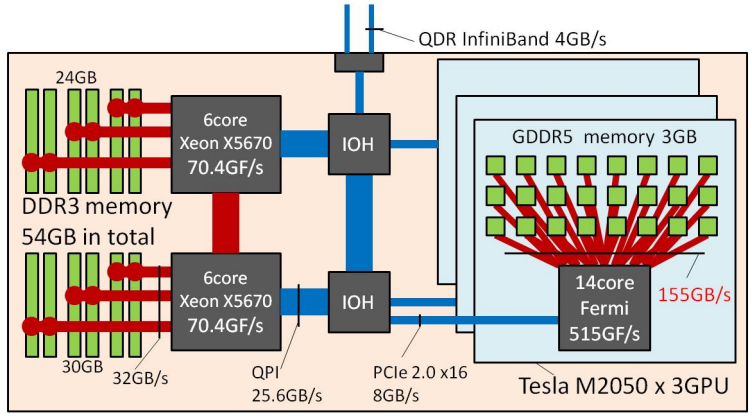


**図 3** Thin 計算ノードの内部構成

を用いるため，以下では単に計算ノードと呼ぶ場合がある．以下，本論文に関連の深い部分について概要を示す．

**Thin 計算ノード:** 各計算ノード Hewlett-Packard Proliant SL390s G7 は 6 コアの Intel Xeon X5670 2.93GHz プロセッサを 2 つ，NVIDIA Tesla M2050 GPU を 3 つ搭載する．図 2 にノード外観を，図 3 にノード内部構成を示す．メインメモリとしては計 54GB の DDR3 メモリを搭載する．また 40Gbps QDR InfiniBand の host channel adapter (HCA) を 2 つ持つ．この HCA は Socket 0 CPU 側 (内部構成図の上側) の IO Hub に，それぞれ PCI-Express (PCIe) Gen 2 x8 で接続される．三つの GPU のうち一つは Socket 0 側の IO hub に，二つは Socket 1 側に，それぞれ PCIe Gen2 x16 で接続される．以上のように，HCA, GPU は PCIe レーンを共有することなく，効率的に通信を行うことができる．

オペレーティングシステムは 64bit 対応 SuSE Linux Enterprise Server 11 および Windows HPC server 2008 R2 である．本論文の実験では Linux を用いる．

**フルバイセクションネットワーク:** インターコネクトは 2 段のスイッチから成るファットツリーであり，フルバイセクション構成である．Dual rail 構成であり，各 rail がファットツリーを構成する．エッジスイッチとして 36 ポートの Voltaire GridDirector 4036 を 185 台持つ．各エッジスイッチのポートのうち 18 は上流のコアスイッチ向け，残り 18 は下流のノード向けである．コアスイッチは 324 ポートの GridDirector 4700 である．各 rail につき 6 台，計 12 台存在する．各ノードは 2 本の 40Gbps QDR InfiniBand によりエッジス

Vol.2010-ARC-192 No.11
Vol.2010-HPC-128 No.11
2010/12/16

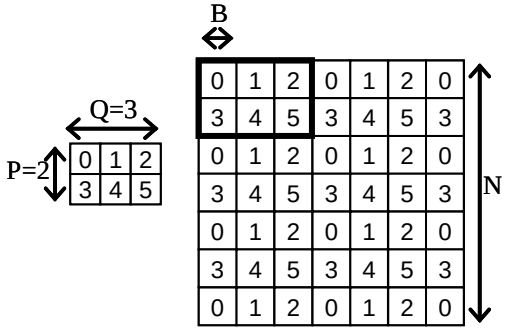


**図 4** (左)$P \times Q = 2 \times 3$ プロセスのプロセス格子例，(右)6 プロセスによる $N \times N$ 行列の二次元ブロックサイクリック分割

イッチに接続される．2 本は Dual rail のそれぞれに接続される．

**Tesla M2050 GPU:** 各ノードは NVIDIA Tesla M2050 と呼ばれる Fermi 世代の GPU を 3GPU 搭載する．各 GPU はストリーミングマルチプロセッサ (SM) が 14 基持ち，各 SM は SIMD 動作する CUDA core を 32 基持つ．また SM から共有され，148Gbytes/s のメモリバンド幅を持つ 3GB の GDDR5 デバイスメモリが搭載されている．GPU の理論演算性能は，倍精度浮動小数演算では 515GFlops，単精度では 1.03TFlops である．アーキテクチャの詳細については NVIDIA の公開情報[2] を参照されたい．

Tesla の利用のためには CUDA プログラミング環境が提供されており，拡張された C 言語によるプログラミングを行うことができる．また BLAS ライブラリである CUBLAS，フーリエ変換ライブラリである CUFFT が提供されているが，本研究では CUBLAS ではなく NVIDIA より提供された内部バージョンのカーネル関数を用いている．なお理論性能は 515GFlops であるが，Fermi 世代 GPU 上の倍精度行列積 (DGEMM) の性能はその 75%が限界となっており[⋆1]，今回用いた DGEMM カーネルのボード内 (PCIe 通信コストを含まない) 性能は 360GFlops 程度となっている．

## 3. High performance Linpack

本稿では Linpack の良く知られた並列実装である High performance Linpack (HPL)[6] を，ソースコードの一部改変して実行に用いる．HPL は正方密行列を係数とする連立一次方程式をブロック化ガウス消去法で解く，MPI 並列ソフトウェアである．指定された行列サイズ $N$ に対して乱数行列を生成し，方程式を解き，その速度を Flops 値で評価する．

計算に参加するプロセス群は概念的にサイズ $P \times Q$ のプロセス格子を形成し，行列はプロセス格子に従って二次元ブロックサイクリック方式で分散される (図 4)．以下，行列サイズを $N$，ブロックサイズを $B$ とする．計算のほとんどの部分をガウス消去法が占め，その各ステップ (ステップ番号 $k$ とする) は，以下のような処理からなる．

**パネル分解:** 第 $k$ ブロック列はパネル列 $L$ と呼ばれ，その箇所の LU 分解を部分ピボット選択を用いて行う．

**パネルブロードキャスト:** パネル列 $L$ の各ブロックの内容を他プロセスへブロードキャストする．ここではプロセス格子の各行内での通信が発生する．

**行交換通信:** 部分ピボット選択の結果に基づき，行交換を行う．ここではプロセス格子の各列内での通信が発生する．これにより，第 $k$ ブロック行が生成され，その箇所を $U$ と呼ぶ．$U$ に対して三角行列求解 (DTRSM) 計算を行う．

**更新計算:** パネル列 $L$ と，行交換後の第 $k$ ブロック行 $U$ の内容を用い，行列の未分解部分の更新計算を行う．行列の未分解部分を $A_k$ とすると，各プロセスは自分の持っている部分行列について，$A_k = A_k - L \times U$ という密行列積 (DGEMM) 計算を行う．

各 MPI プロセスが行う処理を図 5 に示す．なおここでは”パネル分解”は省いている．パネルブロードキャストについては，HPL では lookahead と呼ばれる最適化が採用されている．つまり，ステップ $k+1$ のためのパネル列の通信を，ステップ $k$ のうちに行っておき，通信コストの隠ぺいをしようとするものである．本図に示すアルゴリズムがどのように変更されるかは，後に述べる．

さて上記の処理のうち，パネル分解の計算量総計は $O(N^2B)$，パネルブロードキャストと行交換通信の通信量総計は $O(N^2(P+Q))$，更新計算の計算量総計は $O(N^3)$ である．このことから，最も時間がかかるのは更新計算であり，その傾向は $N$ が大きい程強いと分かる．そのため，並列 Linpack ベンチマークにおいて良い性能を得るためには，$N$ をメモリ量の限界に近づけるように大きくとり，高速な行列積を行う BLAS 数値演算ライブラリを用いることが一般的に行われている．

## 4. TSUBAME 2.0 上の設計と実装

TSUBAME 2.0 上の設計と実装は既報告の TSUBAME1 上のもの[7] を基にする．ここ

⋆1 NVIDIA 技術者からの情報による

Vol.2010-ARC-192 No.11
Vol.2010-HPC-128 No.11
2010/12/16

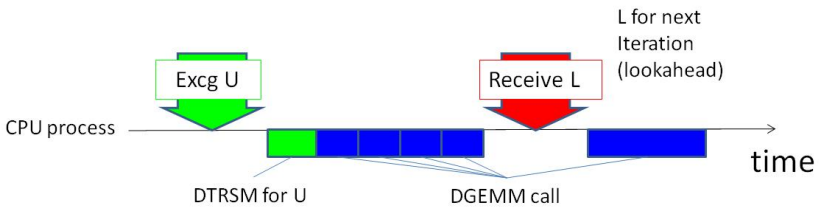


**図 5** オリジナル HPL の 1 ステップのアルゴリズムの模式図

**表 1** 各システムにおけるノード毎の計算性能とノード間通信性能．典型的な x86 クラスタについても概算を示す．ヘテロ型システムにおいては 1 ノードあたりの，ホスト-アクセラレータ間 PCI 通信性能も示す

| | 理論演算性能 (GFlops) | ノード間通信性能 (GB/s) | PCI 通信性能 (GB/s) |
|---|---|---|---|
| x86 cluster | 約 100 ～ 300 | 約 1 ～ 8 | - |
| RoadRunner | 450 | 2 | 4 |
| TSUBAME 1.2 | 157 ～ 330 | 2 | 1 ～ 3 |
| TSUBAME 2.0 | 1685 | 8 | 24 |

ではその設計上の議論を改めて述べ，変更点と実装の詳細について示す．

### 4.1 システムアーキテクチャからの議論

まず TSUBAME 2.0 と，やはりヘテロ型システムである TSUBAME 1 および LANL RoadRunner[?] との比較をシステムアーキテクチャの面から行う．RoadRunner においては，各ノードが Opteron を 4 コアと Cell プロセッサの一種である PowerXCell 8i (以下，Cell) を 4 つ搭載する．TSUBAME 1.2 の各ノードは，Opteron 16 コアと ClearSpeed アクセラレータ，そして一部のノードが Tesla S1070 GPU を 2GPU 持つ．

**カーネル演算の主体:** カーネル演算である行列積 (DGEMM) をどのプロセッサが行うか，各プロセッサ種の演算性能比から議論する．RoadRunner においては理論演算性能の 96%(ピークの割合) を Cell が占めるため，Cell のみがカーネル演算を行う．TSUBAME 1.2 においては，Opteron が 35%，ClearSpeed が 32%，GPU が 33%を占めるために，いずれかのプロセッサ種類の利用をやめると大きな性能ロスが生じる．そのため全ての種類をカーネル演算に用いた．TSUBAME 2.0 においてはそのバランスが大きく変わり，GPU が理論演算性能の 92%, Xeon が 8%であるため，今回の実験では基本的に GPU をカーネル演算に用いることとした．例外として PCIe 通信コストが相対的に高くなる小さい行列の演算は CPU が行うこととした．

なお，DGEMM に GPU と CPU の双方を用いるバージョンも実装したが，予備実験により並列時に性能が微小ではあるが下がることが確認された．MPI 通信との衝突のためと推測されるが，この点については今後改善の余地があると考える．

**行列データの配置場所:** Linpack においては $N \times N$ の行列データを MPI プロセスに分散し保持させる．一方前述の通り，メモリサイズに収まる範囲で $N$ が大きいほうが高性能のために望ましい．RoadRunner においては，CPU のみがアクセスできるホストメモリと Cell のみがアクセスするデバイスメモリの大きさは，どちらもノードあたり 16GB で同じとなっている．そのため行列データをデバイスメモリに置くという方針をとっている．一方 TSUBAME 1.2, 2.0 ともデバイスメモリはホストメモリよりもはるかに小さい．そのため行列データをホストメモリに配置することとした．このときアクセラレータの演算の際に PCIe 通信が必要である．

**ノード間ヘテロ性について:** TSUBAME 1.2 においては，GPU が一部ノードに搭載されていることによりノード間の性能に差異が生じていた．RoadRunner や TSUBAME 2.0 においてはその現象は無いので，実装は比較的容易となる．

**計算性能と通信性能の比:** 表 1 に示すように，一般的にヘテロ型システムは通常のクラスタよりに比べ，通信性能が相対的に低くなる傾向にある．TSUBAME 2.0 でもノードあたり 8GB/s と，絶対通信性能は他の多くのシステムより優れているが，演算性能が約 1.7TFlops と高いため，相対的にはノード間通信のコストは大きくなる．そのために，通信と計算のオーバラップなどの，通信コストを隠ぺいする技術はこれまでよりも重要となる．

### 4.2 TSUBAME 2.0 上の実装

ここでは TSUBAME 2.0 上の HPL ソースコードの改変について述べる．HPL を構成する各 MPI プロセスは，通常通り CPU 上で動作させる (現状ではそれが唯一の選択肢である)．そして GPU はカーネル演算のためにのみ利用する．行列データは前述のように通常はホストメモリに置かれるため，DGEMM/DTRSM 演算の際には一部ずつデバイスメモリに PCIe を介し送信し，GPU 側で計算する．ここではパイプライン処理により，計算と PCIe 通信のオーバラップを行う．さらには MPI 通信もオーバラップ可能とするため，$U$ を列方向分解して行交換処理を細切れに処理可能なように変更した．つまり，図 5 に述べたアルゴリズムは図 6 のように変更された．ここでは，各プロセスが持つ $U$ を列方向分割したものを $U_0, U_1, U_2 \cdots$，$A_k$ を列方向分割したものを $A_0, A_1, A_2 \cdots$ と呼んでいる．また，

Vol.2010-ARC-192 No.11
Vol.2010-HPC-128 No.11
2010/12/16

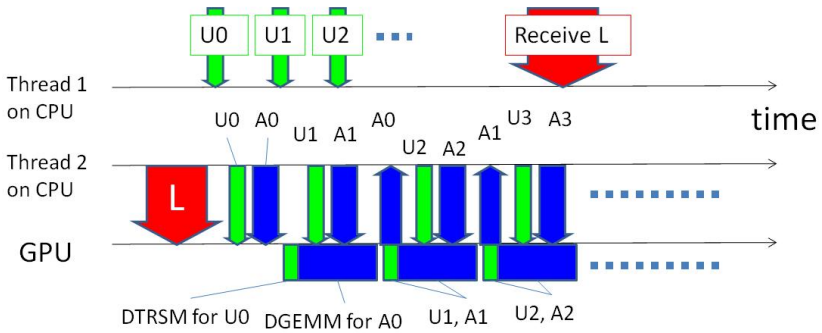


**図 6** TSUBAME 2.0 上の HPL の 1 ステップのアルゴリズムの模式図

MPI 通信を行うスレッド (thread1) と別に，GPU との PCIe 通信，カーネル呼び出しを行うスレッド (thread2) を生成している．この手法においては，オーバラップにより実行時間の多くにおいて GPU 計算が走ることとなる．オリジナル版と異なり，$L$ の MPI 通信中も計算を行う．GPU が動作していないのは $L$ の PCIe 通信，$U_0$ の行交換中および PCIe 通信中など，相対的にはごく一部の時間である．

現在の実装では一つの MPI プロセスが一つの GPU を駆動するようにしているが，複数の GPU を駆動するように変更することは容易である．

### 4.3 チューニング

前節で述べた実装を効率的に動作させるため，以下のようなチューニングを行っている．

現在の実装は一 MPI プロセスが一 GPU を駆動し，各ノードに 3 プロセス (=GPU 数) を起動するが，各プロセスの CPU 上の位置と GPU は近いことが望ましい．そのため，図 3 に応じて Socket 0 CPU に 1 プロセスをバインドし，Socket 1 CPU に 2 プロセスをバインドし，それぞれ近い方の GPU を用いる．なおこのとき 3 プロセス中 2 つは，MPI 通信のために Socket を一度またいだ通信が起こっている．

PCI 通信を必要とするヘテロ型システムにおいては，計算-PCI 通信比を向上させるためにブロックサイズ $B$ のチューニングを必要とする．予備実験をもとに $B = 1024$ とした．これは TSUBAME 1.2 での $B = 1152$ とほぼ同等となった．なお行列データをデバイス側に置く RoadRunner では，PCI 通信が非常に小さく $B$ の影響は少ないため，$B = 128$ となっている．

## 5. 測 定 結 果

TSUBAME2.0 上の Linpack 測定を，システム導入準備期間中である 2010 年 10 月中旬に行った．利用したソフトウェア環境は，SUSE Linux Enterprise 11, OpenMPI 1.4.2, GCC 4.3 である．BLAS ライブラリとしては，Xeon においては GotoBLAS2 1.13，Tesla GPU においては前期の通り NVIDIA 提供の内部バージョンの DGEMM/DTRSM 関数を用いた．Xeon プロセッサの TurboBoost 機能はオフとした．1408 ノード中の 1357 ノードを実行に用いた．このときプロセス数 (=GPU 数) は 4071 となり，このプロセスを $P \times Q = 59 \times 69$ の格子に構成した．利用パラメータは，$N = 2,490,368, B = 1024$ となっている．

この実行により 1.192PFlops を達成した．これは国内で初めて 1PFlops を超えた実行であり，TSUBAME 1.2 の場合の 13.7 倍に相当する．実行時間は 8640 秒であった．この結果は 2010 年 11 月の Top500 ランキングにおいて世界 4 位にランクされた．なお一位の Tianhe-1A，3 位の Nebulae も GPU を用いたヘテロ型システムとなっている．

### 5.1 実行効率の解析

1357 ノードの理論演算性能は 2.288PFlops であるため，Linpack 性能と理論性能の比である実行効率は 52.1%となる．これは TSUBAME 1.2 時の 53%に近いが，その原因は大きく異なることが分かった．原因を解析するために，DGEMM 性能に注目し，その解析結果を図 7 に示す．グラフは，TSUBAME 2.0, TSUBAME 1.2 および，TSUBAME 1.0 の Opteron CPU のみを用いた場合の三通りを比較する．最も左側のプロットは理論性能である 100%を示し，最も右側のプロットは Linpack 性能/理論性能を示す．”DGEMM-1”は，各システムの CPU コアもしくはアクセラレータ単体で DGEMM を実行し，それを合計した値に相当する．また”DGEMM-2”は，各ノードにおいてアクセラレータおよび CPU で，DGEMM を Linpack 実行時と同様のプロセッサで (TSUBAME 1.2 では全プロセッサ種，TSUBAME 2.0 では GPU のみ) 実行した場合に相当する．PCI 通信のコストやバス衝突コストは，”DGEMM-1”と”DGEMM-2” の差に含まれる．

TSUBAME 1.2 と TSUBAME 2.0 における理論性能と Linpack 性能の乖離の原因は大きく異なることが分かる．TSUBAME 1.2 においては DGEMM-2 と Linpack の差が最も重大であるが，これには MPI 通信の影響やノード間ヘテロ性の影響が含まれる．またこの時の実装では，4 節で述べたような細粒度の U 交換のオーバラップを行っていなかったことも原因の一つと考えられる．一方，TSUBAME 2.0 においては Peak と DGEMM-1 の差が最大である．これは 2 節で述べたように，Fermi 世代の GPU において DGEMM の性

Vol.2010-ARC-192 No.11
Vol.2010-HPC-128 No.11
2010/12/16

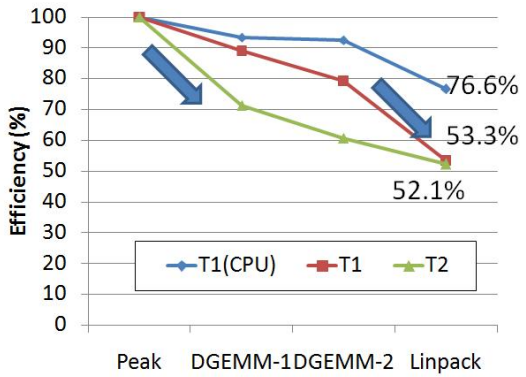


**図 7** TSUBAME 2.0, TSUBAME 1.2, TSUBAME 1.0(CPU のみ) 上の Linpack の実行効率解析結果

能が抑えられていることが原因である．次世代の GPU においてこの制限が緩和されれば，それに応じて Linpack 実行効率も大きく改善されると考えられる．

### 5.2 電力性能

Linpack 実行時の分電盤の記録から消費電力を測定した．分電盤にはアイドルであったノードも含まれているため，記録値からそれらの電力を減算した．その結果，Linpack 実行中のシステムの平均消費電力は 1440kW であった．これには並列ファイルシステムの電力，および MCS 空調，チラーの電力は含まれていない．分電盤レベルの測定であるので，ノードの電源ユニットにおけるロス分は含まれている．

一方でスーパーコンピュータの電力性能比のランキングである Green500[1] には 1243.8kW という値を提出している．この値は Green500 の電力測定ルールを遵守すべく，以下のように求められている．まず電力測定の期間は，Linpack 実行中の 20%以上と定められている．Linpack 実行中の最後の 21.3%の期間の平均電力とした．また，エッジスイッチの電力は含む必要があるが，コアスイッチの電力 (この場合 36kW であった) を含まなくてよいと Green500 委員会から回答を得たのでそのようにした．この時の電力と演算性能の比は 958MFlops/w であり，世界トップクラスの省電力を達成している．

## 6. おわりに

ペタフロップスの演算性能を持つ TSUBAME 2.0 スーパーコンピュータにおいて Linpack を実行し，1.192PFlops の性能を達成した．速度性能と電力性能比の双方において世界トップクラスを実現している．

現在の実装には最適化の余地が残っており，まず GPU と CPU の混合カーネル実行の効率化と，それに対する MPI 通信の影響の軽減を行いたい．また電力性能比を向上させるために CPU/GPU のクロック/電圧と性能の関係に基づいた最適化を行いたい．

**謝辞** システム導入・実験にあたって日本電気，日本ヒューレット・パッカード，NVIDIA，マイクロソフト，Voltaire，DDN，東京工業大学学術国際情報センターの皆様に多大なご協力を頂きました．本研究の一部は科学技術振興機構戦略的創造研究推進事業「Ultra-Low-Powr HPC: 次世代テクノロジのモデル化・最適化による超低消費電力ハイパフォーマンスコンピューティング」, NVIDIA CUDA Center of Excellence および科学研究費補助金 (特定領域研究 課題番号 18049028) の援助による．

## 参 考 文 献

1) The GREEN500 list. `http://www.green500.org/`.
2) NVIDIA CUDA Documentation. `http://www.nvidia.com/object/cuda_develop.html`.
3) TOP500 supercomputer sites. `http://www.top500.org/`.
4) Toshio Endo and Satoshi Matsuoka. Massive supercomputing coping with heterogeneity of modern accelerators. In *Proceedings of IEEE IPDPS08*, page 10pages, 2008.
5) Toshio Endo, Akira Nukada, Satoshi Matsuoka, and Naoya Maruyama. Linpack evaluation on a supercomputer with heterogeneous accelerators. In *Proceedings of IEEE IPDPS10*, page 8pages, 2010.
6) A.Petitet, R.C. Whaley, J.Dongarra, and A.Cleary. HPL - a portable implementation of the high-performance Linpack benchmark for distributed-memory computers. `http://www.netlib.org/benchmark/hpl/`.
7) 遠藤敏夫, 額田彰, 松岡聡, and 丸山直也. 異種アクセラレータを持つヘテロ型スーパーコンピュータ上の linpack の性能向上手法. In **情報処理学会研究報告** *2009-HPC-121*, page 8pages, 2009. 並列/分散/協調処理に関するサマーワークショップ (SWoPP2009).
8) 松岡 聡. TSUBAME 2.0 始まる．*TSUBAME e-Science Journal*, (2):2–8, 2010.
9) 松岡 聡, 遠藤 敏夫, 丸山 直也, 佐藤 仁, and 滝澤 真一朗. TSUBAME 2.0 の全貌. *TSUBAME e-Science Journal*, (1):2–4, 2010.