社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2009－ARC－182 (3)
2009－HPC－119 (3)
2009／2／26

四種プロセッサからなるヘテロ型スーパーコンピュータにおけるLinpackチューニング

遠藤敏夫[†,††] 額田彰[†,††] 松岡聡[†,†††,††]
丸山直也[†,††] 實本英之[†,††]

汎用プロセッサ二種とSIMD型アクセラレータ二種を備えたヘテロ型スパコンであるTSUBAMEにおけるLinpackベンチマークの実行について報告する．アクセラレータ型アーキテクチャは電力・設置面積を抑えつつ計算機システムの性能向上のために重要であるが，大規模並列計算を行った報告は筆者らの報告を除き依然少ない．TSUBAMEの約10000のOpteronコア，約500のXeonコア，約640基のClearSpeedアクセラレータ，約620基のNVIDIA Tesla GPUを全て用いたLinpack実行において，77TFlopsを達成した．この結果を得るためには，アクセラレータの性能を保持するためのプログラムの改変やチューニングが必要不可欠であった．なお今回の結果は最新Top500スパコンランキングに29位にランクされ，ヘテロ型システムとしては世界二位である．

Linpack Tuning on a Heterogeneous Supercomputer with Four Types of Processors

Toshio Endo,[†,††] Akira Nukada,[†,††] Satoshi Matsuoka,[†,†††,††]
Naoya Maruyama[†,††] and Hideyuki Jitsumoto [†,††]

We report Linpack benchmark results on the TSUBAME supercomputer, a large scale heterogenous system with two types of general processors and two types of accelerators. Although accelerator architectures are promising for performance improvement of computer systems while keeping power consumption and footprint low, there are only few reports about large scale computations on a large number of accelerators, except our previous trials. With all of about 10,000 Opteron cores, 500 Xeon cores, 640 ClearSpeed accelerators and 620 NVIDIA Tesla GPUs, we have achieved 77TFlops in Linpack. Keys for obtaining this result are modification to the program code and careful tuning that preserve performance of accelerators. With this result, TSUBAME is ranked as 29th in the latest Top500 supercomputer ranking, and it is the second largest heterogeneous system in the world.

1. はじめに

高性能計算を実現するアーキテクチャとして，限られた電力やスペースで高い演算性能を実現可能なアクセラレータアーキテクチャが注目されている．多くはSIMDアーキテクチャ，もしくはそれらの複数から構成されており，近年のアクセラレータとしては，Sony/IBM/東芝のCell Broadband Engine，ClearSpeed SIMDアクセラレータ，NVIDIAやAMD(ATI)のグラフィックプロセッサ(GPU)などが挙げられる．アクセラレータ上の計算技術の研究は近年急速に増加しており，複数のGPUを用いた並列流体演算の性能[8]などについてもすでに報告されている．またCPUとGPUを併用することにより単体より性能向上させる研究もおこなわれてきた[6),9)]．しかし，数百台以上のオーダのアクセラレータを大規模計算に用いた研究については，我々の以前の報告[4),10)]をのぞき，ほとんど見られない．

これまでに我々は東京工業大学のスーパーコンピュータTSUBAMEを用い，汎用プロセッサであるOpteronとClearSpeedアクセラレータを併用してLinpackを実行し，56TFlopsを達成したことを報告した．その後2008年秋に東京工業大学学術国際情報センター(GSIC)ではNVIDIA社のGPUに基づくアクセラレータであるTesla S1070を170機(680デバイス)導入し，それらはTSUBAMEへ装着されている．またTSUBAMEと別クラスタとして，90

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

ノード720Xeonコアからなる TSUBASA クラスタが設置され，TSUBAME と InfiniBand で接続されている．本稿では，以上のほぼ全ての計算資源である，TSUBAME の約 10000 の Opteron コア，約 640 基の ClearSpeed アクセラレータ，約 620 基の NVIDIA Tesla GPU，および TSUBASA のうち約 500Xeon コアを併用して Linpack を実行した結果について報告する．アクセラレータの性能を保持するためのプログラムの改変やチューニングにより，77.48TFlops を実現した．この結果は最新の Top500 スパコンランキング[3] で 29 位にランクされ，異種のプロセッサを用いたヘテロ型システムとしては世界二位である．

2. TSUBAME システム

TSUBAME では，655 ノードの計算サーバ SunFire X4600 と，合計 1.6PBytes のストレージサーバが InifiniBand により接続されている．以下，本論文に関連の深い部分について概要を示す．

TSUBAME 計算ノード: TSUBAME の各計算ノードは，dual core 2.4GHz Opteron 880 を 8 個を持ち，16 CPU core が 32GB のメモリを共有する☆．またノードは InifiniBand host channel adapter (HCA) を 2 つ持つ．オペレーティングシステムは 64bit 対応 SuSE Linux Enterprise Server 10 である．主要な I/O スロットは PCI-X および PCI-Express 1.0 x8 であり，ここに HCA および後述のアクセラレータが接続されている．

InfiniBand インターコネクト: 各ノードは 2 本の 10Gbps SDR InfiniBand により 288 ポートの Voltaire ISR9288 スイッチ群に接続される (図 1)．スイッチ間は，InifiniBand 24 本により接続される．並列プログラムを動作させるために，Message passing interface(MPI) の一実装である Voltaire MPI が利用可能となっており，本稿でもそれを用いる．

ClearSpeed アクセラレータ: 655 ノードのうち 648 ノードが，PCI-X バスによって接続される ClearSpeed アクセラレータボード X620[1] を一枚ずつ持つ．各アクセラレータは 2 つの CSX600 SIMD プロセッサと 1GB の DRAM(メモリバンド幅 6.4Gbytes/s) を持つ．各プロセッサには，420MFlops(倍精度) の演算能力を持つ 96 個の PE が含まれ，アクセラレータの理論性能は 80.64GFlops となる．なお，アクセラレータ上の演算の入出力データは，1.06GBytes/s の PCI-X バスを介してホストと通信する必要がある．アクセラレータあたりの消費電力は約 25W，648 枚で約 16kW となっており，これはシステム全体のピーク消費電力である約 1MW の 2%以下に相当する．

アクセラレータを利用する手段として，SIMD 並列プログラミング言語 C^n，基本線形演算を行う CSXL ライブラリ，高速フーリエ変換を行う CSFFT ライブラリなどが提供されている．本稿ではこれらのうち CSXL ライブラリを利用する．

Tesla アクセラレータ: NVIDIA Tesla S1070 は，1U の筐体に 4 つのアクセラレータデバイス (グラフィックボードと同等) を含んでいる．システム全体では 170 筐体，680 デバイスが導入されている．316 ノードの計算ノードが Tesla と接続されており，基本的にノードあたり 2 デバイスが接続されている☆☆．ノードと Tesla 筐体は PCI-Express gen1 x8 のインタフェースカードおよびケーブルで接続される．残りの 339 ノードは Tesla を持たない．

各 Tesla デバイスは Tesla T10GPU を持ち，その中にはストリーミングマルチプロセッサ (SM) が 30 基存在する．また SM から共有され，102Gbytes/s のメモリバンド幅を持つ 4GB のデバイスメモリが搭載されている．各デバイスのピーク速度は，倍精度浮動小数演算では 86.4GFlops，単精度では 1.04TFlops である．消費電力については筐体あたり約 700W であり，デバイスあたり約 175W となる．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 を使わずに独自に後述するカーネルを作成した．

TSUBASA クラスタ: TSUBAME とは別のシステムとして，TSUBASA と呼ばれる Xeon クラスタが設置されており，TSUBAME とは 20 本の InfiniBand(計 200Gbps) で接続されている．図 1 の網掛け部が TSUBASA クラスタである．本研究ではこのクラスタも TSUBAME と協調させて Linpack を実行する．

このクラスタは 90 ノードからなり，各ノードは Quad core Xeon E5440 (2.83GHz) を 2 つ，計 8CPU

☆ 一部のノードは 2.6GHz の Opteron885 を持つ，またメモリ容量が 64GB または 128GB のノードもあるが，本稿ではその差異は利用しない

☆☆ 一部のノードは 4 デバイスと接続されているが，今回はノードあたり最大 2 デバイス利用している

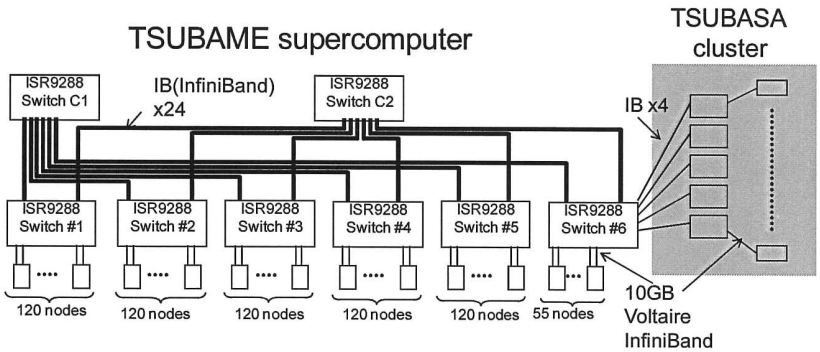

図 1 TSUBAME のネットワーク構成．右方の網掛け部は TSUBASA クラスタ

コアを持つ．メモリ容量はノードあたり 8GB または 16GB である．各ノードは SDR InfiniBand(10Gbps) でネットワークと接続される．

2.1 ヘテロ性について

本研究では以上のような計算資源をほぼ全て併用して並列プログラムを動作させる．Opteron, ClearSpeed, NVIDIA, Xeon という 4 種類のプロセッサが混在することになる (ノード内ヘテロ性)．また，ノード毎のコンフィグレーションに注目すると，以下の 3 種類が混在している (ノード間ヘテロ性)．

Tesla あり TSUBAME ノード： 16 コアの Opteron, 1 つの ClearSpeed，2 つの Tesla デバイスを持つ．

Tesla なし TSUBAME ノード： 16 コアの Opteron と 1 つの ClearSpeed を持つ．

TSUBASA ノード： 8 コアの Xeon を持つ．

なおより正確には，CPU 周波数やメモリ容量などの違うノードもあるが，本稿ではその差異は利用しない．また若干台の ClearSpeed を持たない TSUBAME ノードについては，今回は Linpack 実行に含めなかった．

3. HPL の概要

本稿では Linpack の良く知られた並列実装である High performance Linpack (HPL)[7] 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 2)．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．

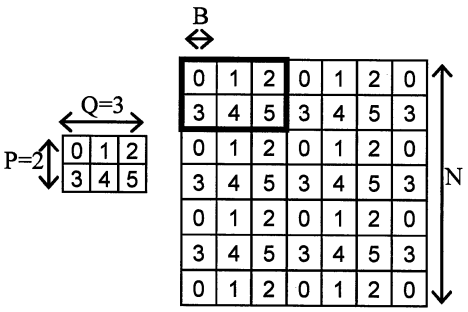

図 2 (左)$P \times Q = 2 \times 3$ プロセスのプロセス格子例，(右)6 プロセスによる $N \times N$ 行列の二次元ブロックサイクリック分割

パネル分解： 第 k ブロック列はパネル列と呼ばれ，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．

更新計算： パネル列と，行交換後の第 k ブロック行の内容を用い，行列の未分解部分の更新計算を行う．

以上のうち，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．HPL は更新計算のために BLAS の DGEMM 関数 (行列積) と DTRSM 関数 (三角行列求解) を用いるので，TSUBAME においてはこれらの計算をアクセラレータを併用して高速化することが重要である．

4. カーネル演算性能

HPL の性能に最も影響を与える DGEMM 関数 (行列積) の性能について述べる．HPL の性質から，以下のような行列サイズについて実験した: B を前述のブロックサイズ，M を B より十分大きなサイズとし，$M \times B$ 行列と $B \times M$ 行列の積の性能を，いくつかの B, M について調べた☆．

まず Opteron 1 コアを用い，GotoBLAS 1.26[5] の DGEMM 性能を測定したところ，約 4.4GFlops であった．また後述のアクセラレータの場合と異なり常にほぼ一定であった．

ClearSpeed 上の DGEMM 性能を図 3 に示す．CSXL ライブラリ 3.11 を用いている．行列サイズ M, B が大きくなるにつれ性能が向上しているが，以下の理由によると考えられる．CSXL の DGEMM 関数を呼び出す際には行列データはホストメモリ上にあり，ホスト-アクセラレータ間の PCI-X 通信コストはグラフの性能に含まれている．相対的な通信コスト (通信量/計算量) は M, B が大きくなるにつれ小さくなるため性能が向上する．$M = 11520, B = 1152$ のとき 63.4GFlops(理論値の 79%) である．

同様に Tesla 1 デバイス上の DGEMM 性能を図 4 に示す．これは我々が作成した DGEMM カーネルを用いており☆☆，やはりホスト-アクセラレータ間の通信コストを含む．M, B と性能の関係については ClearSpeed と似た傾向を示す．$M = 11520, B = 1152$ のとき 78.4GFlops(理論値の 91%) である．

図 5 は，Tesla あり TSUBAME ノードにて，Opteron と ClearSpeed 1 デバイスと Tesla 2 デバイスを全て用いた場合の DGEMM 性能を示す．複数の CPU, アクセラレータを利用するために，行列を一定の割合で分割して負荷分散を行っている．Opteron, ClearSpeed, (2 つの)Tesla に 20%, 24%, 56%の割合で仕事を分散した．なお Opteron については以下の

☆ ClearSpeed の CSXL ライブラリの特性のため，B, M は 288 の倍数としている

☆☆ 純正ライブラリの CUBLAS は，演算とホスト-アクセラレータ間通信のオーバラップを許さず，我々の文脈では十分な性能が得られないため，利用しなかった．

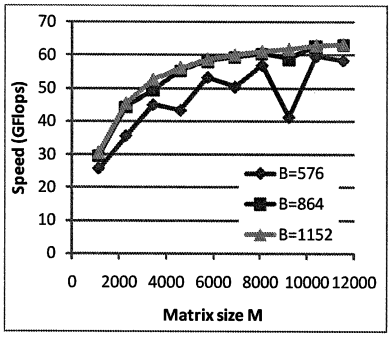

図 3 ClearSpeed の DGEMM 性能 (CSXL ライブラリ 3.11 利用)

理由により，ノードの 16 コアのうち 12 コアのみを行列演算に用いている．アクセラレータを用いた演算のためにホスト-アクセラレータ間通信を行う必要があるが，そのためにも CPU が必要である．予備実験により，ClearSpeed との通信に 2 コア，Tesla との通信にデバイスあたり 1 コアを専用に確保する場合が性能が良いとわかった．

$M = 23040, B = 1152$ のとき 241.3GFlops を達成している．これはノード内の全 CPU, アクセラレータの合計ピーク性能の 330.2GFlops の 73%である．

アクセラレータ単体で測定したときは，$B = 1152$ と $B = 864$ の性能差は小さく，ほとんどの場合 1%以内の差であった．しかしノード全体を用い，アクセラレータを計 3 デバイス用いる図 5 ではその差はひろがり，M が 13824 以上の場合に 16%以上の差となっている．これは $B = 864$ のときに，アクセラレータ単体であれば通信コストは大きな問題にならなかったものが，3 デバイスになったことにより PCI-Express や Hypertransport への負荷が増え，通信コストが相対的に大きくなり，$B = 1152$ との差が大きくなったと考えられる．この結果から，TSUBAME 全体を用いた Linpack 実験においてもブロックサイズ $B = 1152$ を採用した☆☆☆．

5. ヘテロ性への対応と Linpack 性能

5.1 ノード毎のチューニング

異種プロセッサ混在というノード内ヘテロ性，ノードのコンフィグレーションが異なるノード間ヘテロ性へ対応するために，HPL ソースコードの改造などを行った．前報告[4),10)] の方針を基本的に踏襲するため，

☆☆☆ これより B を大きくしても (1440 など)DGEMM 性能はほとんど上がらない一方，HPL の他の箇所のボトルネックが大きくなり全体性能は下がる

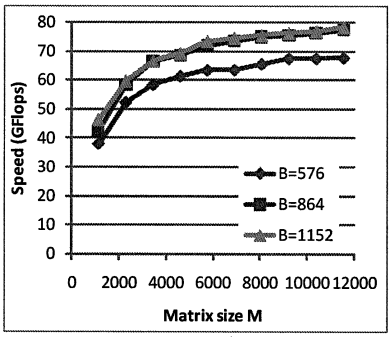

図 4　Tesla の DGEMM 性能 (新たに実装したカーネル利用)

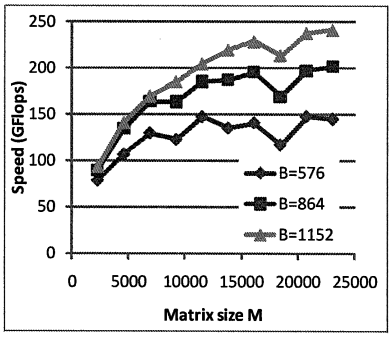

図 5　Opteron 12 コア，ClearSpeed 1 デバイス，Tesla 2 デバイスを用いた DGEMM 性能

ここでは概略のみ述べる．

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)．そしてアクセラレータはカーネル演算のためにのみ利用する．そのため，ノード内ヘテロ性については BLAS ライブラリの層で吸収することになり，これにより各プロセスは CPU とアクセラレータの双方を利用する．

ノード間ヘテロ性については以下のように対応する．HPL では各プロセスに均一の部分行列を持たせ，各プロセスがほぼ均一の速度でそれへの処理を行うことを仮定している．この構造を保つために以下のようにする．性能の高いノードではより多くのプロセスを起動し，結果としてどのプロセスもほぼ均等の性能のプロセッサ群 (CPU, アクセラレータ) を利用するようにする．プロセス数はノードごとの合計 DGEMM 性能になるべく比例するようにし，今回の実験においては，Tesla あり TSUBAME ノードでは 4，Tesla なし TSUBAME ノードでは 2，TSUBASA ノードでは 1 とした．

図 6 は Tesla あり TSUBAME ノードのプロセッサを，4 つのプロセスがカーネル実行時にどう使い分け

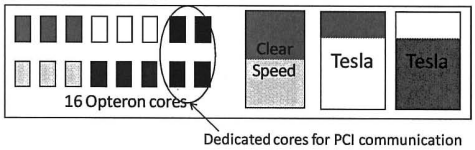

図 6　Tesla あり TSUBAME ノードにおけるプロセスとプロセッサ (Opteron, ClearSpeed, Tesla) の対応

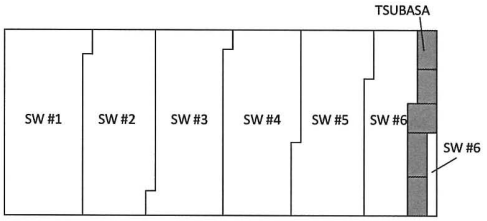

図 7　採用したプロセス格子の概略

るかの概念図である．色がプロセスに対応する．OS による予期しないスケジューリングを最小限にするため，sched_setaffinity システムコールでプロセスと CPU コアの対応付けを行っている．

5.2　システム全体のチューニング

TSUBAME, TSUBASA のほぼ全ての資源を利用しつつ，かつプロセス格子がうまく構成できるノード数を決定する必要がある．最終的には，Tesla あり TSUBAME ノードを 314 ノード，Tesla なし TSUBAME ノードを 330 ノード，TSUBASA ノードを 68 ノード，計 712 ノードを利用することとした．このときの総プロセス数は 1984 であり，32×62 の大きさのプロセス格子とした．さらに，どのノードのプロセスを格子のどこに配置するか決定する必要がある．基本的にはネットワーク的に近いプロセスどうしを近づけるのがよく，図 7 のようにした．さらに，TSUBAME ネットワークのスイッチが静的ルーティングを行っており，単純にノード番号順に格子上に並べるのでは，ネットワーク上流が必要以上に混雑してしまうことが分かった．それを緩和するため，図 7 の配置は保持しつつも，同一スイッチ内のノードのプロセスをランダムに入れ替えることを行っている．

5.3　測定結果

以上のようなチューニングのもと，TSUBAME 全体を用いた Linpack 測定を，2008 年 10 月末のシステムメンテナンス時に行った．利用したソフトウェア環境は，Voltaire MPI，GCC 4.1.2 である．BLAS ライブラリとしては，Opteron, Xeon においては GotoBLAS 1.26，ClearSpeed においては CSXL 3.11，Tesla においては新規に実装した DGEMM/DTRSM

表 1 Top500 における TSUBAME の Linpack 性能と順位の経緯．利用プロセッサにおいて O は Opteron，C360 は ClearSpeed(360 枚)，C648 は ClearSpeed(648 枚)，X は Xeon，T は Tesla を表す

年月	速度 (TFlops)	順位	利用プロセッサ
2006/6	38.18	7	O
2006/11	47.38	9	O, C360
2007/6	48.88	14	O, C360
2007/11	56.43	16	O, C648
2008/6	67.70	24	O, C648, X
2008/11	77.48	29	O, C648, T, X

関数を用いた．

前述の 712 ノードを用いた Linpack 実行により，77.48TFlops を達成した．これは Opteron のみを用いた場合の 38.18TFlops に比べ 2.02 倍の性能である．これは 2008 年 11 月の Top500 ランキングにおいて世界 29 位，国内では東大 T2K に次ぐ 2 位にランクされている．また，異種プロセッサを混在させたシステムとしては，LANL RoadRunner に次ぐ 2 位である．

TSUBAME 導入以来の Linpack 実行結果を表 1 に示す．これらの結果は全て Top500 ランキングに掲載されており，6 回連続性能向上を果たしている．アクセラレータや Xeon プロセッサを増加させるごとに性能向上を行っているが，向上の理由はそれだけでなく，再チューニングや BLAS 性能の向上の影響も含まれている．

前節に述べたチューニングのうち，プロセス格子内のプロセス入れ替えを行わない場合，約 66TFlops しか得られず，Tesla を使わない場合と大差なかった．プロセス入れ替えを行うことにより 77TFlops を達成したが，それでもピーク性能比は 48%となっており，他の上位スパコンに比べて低い．この理由としては，ホスト-アクセラレータ間の通信のために割かれる CPU コア，ノード内バスにおけるアクセラレータの通信と InfiniBand 通信の干渉，(上記チューニングを行ってさえも) ノード間ネットワーク上流における通信混雑など，複数考えられる．今後，それらの影響の分析を行っていきたい．

6. おわりに

TSUBAME スーパーコンピュータにおいて，10,000 コア以上の汎用プロセッサと，Tesla と ClearSpeed という異種アクセラレータを合計 1200 デバイス以上用いて Linpack を実行し，77.48TFlops の性能を達成した．これにより 1000 デバイスのオーダの異種アクセラレータを搭載したシステムのスケーラビリティを実証した．

今後は通信の干渉の影響の解析を行う予定である．また，計算量に対するデータ量の比がより高く，ホスト-アクセラレータ間通信の負荷が高いアプリケーションにおけるスケーラビリティの調査を行いたい．

謝辞 実験にあたって日本電気 (株)，サン・マイクロシステムズ (株)，NVIDIA Corp.，ClearSpeed Inc.，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．TSUBASA クラスタは東工大グローバル COE「計算世界観の深化と展開」により導入されました．また GOTO BLAS ライブラリの作者であるテキサス大学 後藤和茂氏に感謝致します．本研究の一部は JST CREST 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) ClearSpeed Technology Inc. `http://www.clearspeed.com/`.
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*, 2008.
5) Kazushige Goto. Goto BLAS. `http://www.tacc.utexas.edu/resources/software/`.
6) Yasuhiko Ogata, Toshio Endo, Naoya Maruyama, and Satoshi Matsuoka. An efficient, model-based CPU-GPU heterogeneous FFT library. In *Proceedings of International Heterogeneity in Computing Workshop (HCW '08)*, 2008.
7) 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/`.
8) 小川 慧 and 青木 尊之. CUDA による定常反復 poisson ベンチマークの高速化. In **情報処理学会研究報告** *2008-HPC-115*, pages 19–23, 2008.
9) 大島 聡史, 吉瀬 謙二, 片桐 孝洋, and 弓場 敏嗣. CPU と GPU を用いた並列 GEMM 演算の提案と実装. In **先進的計算基盤システムシンポジウム** *SACSIS2006* **論文集**, pages 41–50, 2006.
10) 遠藤 敏夫, 松岡 聡, 橋爪 信明, and 長坂 真路. ヘテロ型スーパーコンピュータ TSUBAME の Linpack による性能評価. **情報処理学会論文誌コンピューティングシステム**, 48(SIG 8 (ACS 18)):62–70, 2007.