Vol.2017-HPC-159 No.9
2017/4/17

GPU搭載スーパーコンピュータ Reedbush-H の性能評価

塙 敏博[1,a)]　星野 哲也[1]　中島 研吾[1]　大島 聡史[1]　伊田 明弘[1]

概要：東京大学情報基盤センターでは，データ解析・シミュレーション融合スーパーコンピュータシステム Reedbush を導入し，2017 年 3 月より全系での稼働を始めている．そのうち、Reedbush-H サブシステムは、計算ノードとして Intel Xeon E5 (Broadwell-EP) プロセッサに加えて NVIDIA Tesla P100 (Pascal) GPU を 2 基ずつ搭載しており，GPU との間で直接通信が可能なように，InfiniBand FDR インタフェースを 2 枚搭載している．本稿では Reedbush-H サブシステムの性能について報告する．

1. はじめに

高い演算性能を限られた電力制約の中で実現するため，様々なプロセッサの開発が行われている．その中で，GPU は高い電力あたり性能，高いコスト性能比により，高性能計算向けのアクセラレータとして用いられてきた．また近年，最新の GPU において単精度演算の 2 倍のスループットで実行できる半精度演算をサポートしたことから，それほど精度を必要としない，機械学習や Deep Learning 向けのプロセッサとしても注目を集めている．

東京大学情報基盤センター（以下，当センター）では，2016 年 7 月より「データ解析・シミュレーション融合スーパーコンピュータシステム」を導入した [1]．

本システムは，計算ノード群の一部に GPU が搭載されている．当センターで演算アクセラレータを搭載したシステムを導入するのは今回が初めてである．従来，GPU 向けのプログラムを作成するためには，CUDA のような専用のプログラミング言語を用いて記述する必要があり，多数のユーザにそのような言語を習得してもらうのは困難であると考えてきた．しかし近年，OpenACC[2] といった指示文ベースのアクセラレータ用並列プログラミング言語が標準的に使われるようになり，実用に耐えうる十分な性能が得られるようになってきた．さらに，これまでの計算科学や工学向けのユーザに加えて，データ科学や機械学習など，従来のユーザとは異なる分野からも，GPU 搭載スパコンへの期待やニーズが高まっている．

データ解析・シミュレーション融合スーパーコンピュータシステムは，「Reedbush システム」の愛称で呼ばれており，2017 年 3 月より全系での稼働を始めている．CPU のみを搭載した計算ノードで構成される「Reedbush-U サブシステム」については，すでに 2016 年 7 月より稼働を開始しており，性能評価については文献 [1] で報告した通りである．

本稿では，2017 年 3 月に稼働を開始し，GPU を搭載した計算ノード群からなる「Reedbush-H サブシステム」について性能評価を行った結果を報告する．

なお，Reedbush システムは，現在試験運転期間中であり，性能向上・安定のため，ドライバやソフトウェアなどの更新を頻繁に行っている．従って，本稿と本サービス開始後の実システムとでは性能が一部異なる可能性がある．

2. Reedbush システムの紹介

2.1 概要

Reedbush システムは、**図 1** に示すように、CPU のみのノードからなる **Reedbush-U** と、演算アクセラレータとして GPU を搭載したノードからなる **Reedbush-H** の 2 つのサブシステムから構成され、それぞれ独立のシステムとして運用される．計算ノードに搭載されるプロセッサや GPU については次節以降で詳しく述べるが、2017 年 3 月時点で最新の製品である．さらに、計算ノード間インタコネクトには InfiniBand EDR、ストレージとして Lustre ファイルシステムに加えて高速ファイルキャッシュシステムを採用している．これらコモディティ技術をベースにした最新製品を導入することにより、運用期間中のハードウェアの陳腐化を抑制する一方で、運用に向けた準備にはこれまでのソフトウェア資産や経験を活かすことができる．

図 2 に Reedbush システムの外観を示す．また、**表 1** にシステム全体の仕様を示す．システム全体は、フルバイセクションバンド幅を持つ 1 つの Fat-tree 網として構成され、ノード当たり 100 Gbps を超える InfiniBand ネット

[1] 東京大学 情報基盤センター
[a)] hanawa@cc.u-tokyo.ac.jp

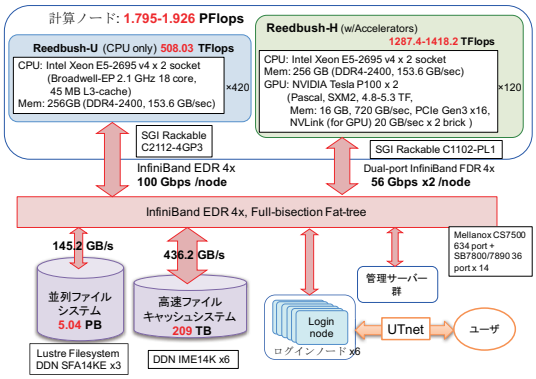

図 1 Reedbush システムの概要

図 2 Reedbush システムの外観

表 1 Reedbush システム全体仕様

Reedbush-U	総理論演算性能	508.03 TFLOPS
	総ノード数	420
	総主記憶容量	105 TByte
	インタコネクト I/F	InfiniBand EDR 4x 1 リンク / ノード (100 Gbps)
Reedbush-H	総理論演算性能	1417.15 TFLOPS (うち GPU: 1272.0 TFLOPS)
	総ノード数	120
	総主記憶容量	30 TByte
	インタコネクト I/F	InfiniBand FDR 4x 2 リンク / ノード (56 Gbps x2)
ノード間インタコネクト		InfiniBand EDR 4x フルバイセクションバンド幅 Fat-tree
並列ファイルシステム	種類	Lustre ファイルシステム DDN SFA14KE
	サーバ (OSS) 数	3 セット (6 ノード、12 サーバ)
	ストレージ容量	5.04 PByte
	バンド幅	145.2 GB/秒
高速ファイルキャッシュシステム	サーバ	DDN IME14K
	サーバ数	6 セット (12 ノード)
	容量	209 TByte
	バンド幅	436.2 GB/秒

ワークにより接続される．

Reedbush システムは空冷システムであり、消費電力は 368.4 kW(冷却除く) の見込みである．

2.2 汎用計算サブシステム：Reedbush-U

各計算ノードは、**表 2** に示すように、各ソケットに 18 コアの Intel Xeon E5 プロセッサ (開発コード名: Broadwell-EP) を 2 ソケット搭載し、256 GB の DDR4 メモリを搭載する．ノードあたり性能は 1.2 TFLOPS、メモリバンド幅は 153.6 GB/sec である．

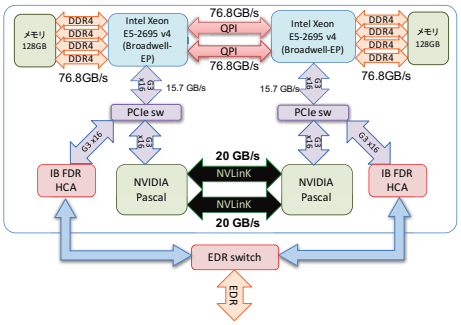

図 3 Reedbush-H ノードの構成

表 1 に示すように、Reedbush-U サブシステム全体は 420 台の計算ノードからなり、各ノードは 100 Gbps の InfiniBand EDR によりフルバイセクションバンド幅を持つ Fat-tree トポロジで接続されている．ピーク演算性能は 508.03 TFLOPS, 総メモリ容量は 105 TByte である．

2.3 演算加速サブシステム：Reedbush-H

各計算ノードは、表 2 に示すように、Reedbush-U と同じ CPU、メモリを搭載しており、加えて**表 3** に示す通り、ノード当たり 2 基の NVIDIA Tesla P100 GPU (開発コード名: Pascal) を搭載する．この GPU は 1 基あたり、最大 5.3 TFLOPS と極めて高い性能を持ち、また 16 GByte の HBM2 (High Bandwidth Memory) を搭載し、メモリバンド幅は 732 GByte/秒に達する[*1][3]．

Reedbush-H サブシステムの計算ノードの構成を**図 3** に示す．Reedbush-H の構成として特徴的なのは、

- 新しい高速インタコネクトである NVLink により 2 基の GPU 間が 40 GByte/秒のバンド幅で接続されていること
- 各 GPU に近接した InfiniBand FDR の HCA (Host Channel Adapter) を用意し、GPU メモリの内容を CPU を介さず InfiniBand で直接送受信できるように工夫されていること

である．

表 1 に示すように、Reedbush-H サブシステム全体は 120 台の計算ノードからなり、各ノードは 56 Gbps の InfiniBand FDR を 2 リンク持ち、ノード当たりでは 112 Gbps を超えるフルバイセクションバンド幅を持つ Fat-tree トポロジで接続されている．ピーク演算性能は 1418.2 TFLOPS、総メモリ容量は 30 TByte である．

3. ベンチマークによる性能評価

本章では，Reedbush-H 計算ノード上の GPU に対するベンチマークとして，STREAM ベンチマーク, 疎行列ベクトル積の結果を示す．さらに，GPU メモリを転送するた

*1 当初は 720 GByte/秒と発表されたが，その後 732 GByte/秒に引き上げられた．

表 2 計算ノードの仕様 (Reedbush-U, H 共通)

CPU	プロセッサ	Intel Xeon E5-2695v4 (Broadwell-EP) × 2 ソケット
	周波数・コア数	2.1 GHz, 18 コア × 2 ソケット
	ピーク性能	1209.6 GFLOPS
メモリ	種別・構成	DDR4-2400, 4 チャネル × 2 ソケット
	容量	256 GB
	バンド幅	153.6 GB/秒

表 3 Reedbush-H 計算ノードの GPU 仕様

プロセッサ		NVIDIA Tesla P100 (Pascal)
搭載数		2 基
GPU 間接続		NVlink × 2 brick (40 GB/sec)
CPU-GPU 間接続		PCI Express Gen3 x16 レーン (16 GB/sec)
GPU 単体	演算ユニット	56 SM (Symmetric Multiprocessor) × 64 CUDA コア (単精度), 32 CUDA コア (倍精度)
	ピーク性能	5.3 TFLOPS
	メモリ種別	HBM2
	メモリ容量	16 GByte
	メモリバンド幅	732 GByte/秒

表 4 P100 における STREAM 性能

種類	GB/s
Copy	514.6
Scale	514.6
Add	537.2
Triad	536.9

めの MPI ベンチマーク，HPL, HPCG の結果を示す．

3.1 STREAM ベンチマーク

STREAM ベンチマーク [4], [5] を用いて P100 単体のメモリバンド幅を測定した．STREAM ベンチマークでは，

Copy: 配列のコピー $a(i) = b(i)$

Scale: 配列のスカラ倍 $a(i) = q \times b(i)$

Add: 2 つの配列の加算 $a(i) = b(i) + c(i)$

Triad: Scale と Add の組み合わせ $a(i) = b(i) + q \times c(i)$

の 4 種を測定することができる．

ここでは，C 版のプログラムをベースに，OpenACC の指示文を追加した．

コンパイルには PGI Compiler 17.1 を用い，コンパイルオプションには `-O2 -acc -ta=tesla:cc60` を指定した．ここで，`-acc` が OpenACC を有効化，`-ta=tesla:cc60` が P100 向けの最適化のためのオプションである．

結果を**表 4** に示す．ここでは 5 回実行した中での最大値を示している．最大の Add の場合で，理論メモリバンド幅 732 GB/s の 73.4%を示した．

3.2 疎行列ベクトル積の性能

疎行列とベクトルを掛け合わせる疎行列ベクトル積計算は、CG 法など様々な数値計算問題にて使われる基本的な演算である。疎行列ベクトル積計算の性能は、今日の多くの科学技術計算の性能に大きな影響を及ぼしている。本節では Tesla K40 (Kepler アーキテクチャ) と Tesla P100 (Pascal アーキテクチャ) を用いてそれぞれ疎行列ベクトル積計算を行い、性能を比較する。対象とする行列は Florida Sparse Matrix Collection[6] に含まれる 202 の行列を用いた。疎行列ベクトル積計算の実装は CUDAtoolkit 8.0 に含まれる CUSPARSE の cusparseDcsrmv 関数を使用し、100 回計算を行ったうちの最速最速時間について比較を行った。疎行列格納形式は CRS 形式 (Compressed Row Storage 形式)、データ型は倍精度浮動小数点型 (double 型) である。

K40 GPU の実行時間に対する P100 GPU の実行時間の比を**図 4** に示す。X 軸は左ほど総非ゼロ要素数の少ない行列、右ほど総非ゼロ要素数の多い行列である。全体的な傾向としては、総非ゼロ要素数の少ない行列では実行時間比 0.7 程度、総非ゼロ要素数の多い行列では実行時間比 0.3 程度であり、総非ゼロ要素数の小さい行列よりも総非ゼロ要素数の多い行列の方が Kepler GPU に対する Pascal GPU の性能割合が高いことがわかる。

K40 GPU と P100 GPU の演算性能比およびメモリ転送性能比と照らし合わせてみると、演算性能比は 1430GFLOPS / 5300GFLOPS ≒ 0.27、メモリ転送性能比 (STREAM Triad 相当) は 218GB/s / 534GB/s ≒ 0.41 である。同程度の総非ゼロ要素数を持つ疎行列よりも性能比が大きく異なる行列がいくつか存在しているが、これは非ゼロ要素の配置と GPU による問題分割位置の相性によるものであると考えられる。疎行列ベクトル積計算はメモリバンド幅の影響が大きな計算であるが、P100 GPU は K40 GPU に対して、対象とする行列の非ゼロ要素数が多い場合には、メモリ転送性能比以上の性能向上を得られていることがわかる。

3.3 GPU 間通信性能

本節では，GPU 間でのデータ転送・通信性能の測定を行う．ノード内での GPU 間でのデータ転送性能・通信性能，および異なるノードの GPU 間での通信性能について評価する．

2GPU 間の通信については，いくつかのデータ転送・通信の手段が存在する．

(1) ノード内: cudaMemcpy による方法 (P2P)
  (a) ホストメモリを間接的に経由する方法
  (b) NVLink(+PCIe) を使って直接転送する方法
(2) ノード内・ノード間: MPI による方法
  (a) ホストメモリを間接的に経由して通信する方法
  (b) GPUDirect for RDMA により直接 (InfiniBand のみで) 通信する方法

これらについて，測定を行った．ここで，CUDA は 8.0 (8.0.44)，ドライバ 375.20 を用いた．MPI については，GPUDirect for RDMA (GDR) が利用できる，MVAPICH2-GDR 2.2, Open MPI 2.0.2 を用いた．

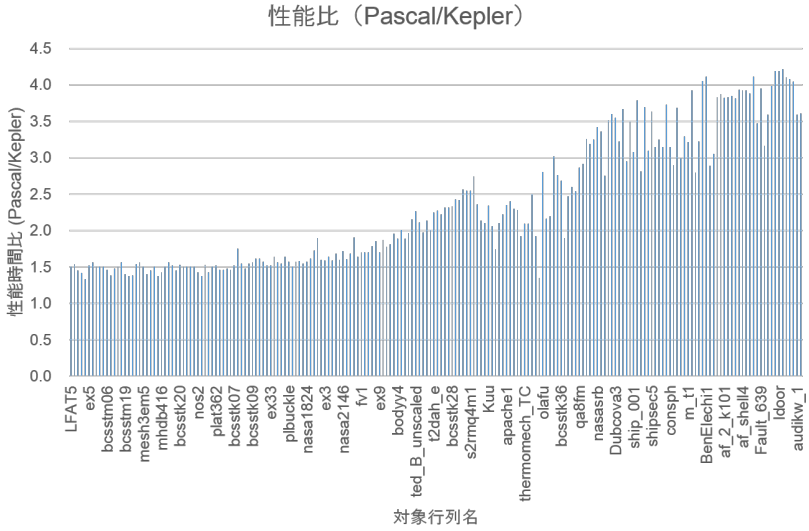

図 4 Tesla P100 (Pascal) と Tesla K40 (Kepler) の疎行列ベクトル積実行時間比

(1) については，自作のコードを用い，(2) については，オハイオ州立大学で開発された OSU Micro-Benchmarks [7] に含まれる osu_latency および osu_bw ベンチマークを用いて測定した．

図 5 に転送・通信遅延時間，**図 6**, **図 7** に転送・通信バンド幅の測定結果を示す．

今回の評価では，(2) の一部として，同一ノードに 2 MPI プロセスを起動し，GPU 間での転送を試したが，メッセージ長が長くなると正常に通信できなくなる問題が発生した．しかし実際には，異なるノードでの GPU 間転送の結果と大きな相違はないと考えられる．

また，Open MPI において，2 ポートを有効にする設定を試したが，1 ポートと同等の性能しか得られなかったため，結果は載せていない．

(1)(a) については，11.0 GB/s で飽和しており，PCIe Gen3 x16 のバンド幅 16 GB/s を考えると，理論ピークに対して 68%程度の性能である．

(1)(b) について，図 6 から，37.0 GB/s の性能が得られている．理論ピークバンド幅は，40 GB/s であり，理論ピークの 93%程度と高い性能が得られた．

一方，(2) MPI の結果を見ると，メッセージサイズが 64 KB 以下の場合には，同一ホスト内のデータ転送よりも高速な結果になっている．

ノード内のデータ転送では，内部で P2P プロトコルを使用しており，これは基本的に write ベースのプロトコルであるため，事前にネゴシエーションを行ってから書き込みを行う動作になる．従って，オーバヘッドが大きく，最小の遅延時間は 10μs である．

一方，MPI での最小遅延時間は，Open MPI 2.0.2 では，4.8 μs, MVAPICH2 GDR 2.2 では，2.5 μs であった．MVAPICH2 では，GDRcopy ライブラリ [8] を使用して，小メッセージサイズの場合に，CPU の AVX 命令を用いて GPU メモリと IB-HCA 間でのデータコピーを行う．実際には 8 KB 程度まで有効になっている．

MVAPICH2 においては，16 KB でバンド幅が落ち込み，1 リンクの場合には 6.3 GB/s，512 KB から 2 リンク使う場合には 9.5 GB/s まで性能が向上している．なお，2 リンク目は，GPU から遠い HCA になるため，GPU メモリを直接転送することができず，ホストメモリ経由になる．そのため，ある程度大きなメッセージサイズでないと効果がない．

Open MPI においては，GDRcopy を使用しないため，8 KB までは MVAPICH2 よりも低い性能であるが，それ以上のメッセージサイズでは MVAPICH2 1 リンクを上回っており，8 GB/s 程度の性能が得られている．

MVAPICH2 において，パラメータを変更したところ，図 7 の Opt に示す性能が得られた．ここでは，`MV2_GPUDIRECT_LIMIT = 16384`, `MV2_USE_GPUDIRECT_RECEIVE_LIMIT = 32768` に設定した．16 KB まで GDR の送受信が有効になることで，16 KB 付近のバンド幅の落ち込みがなくなった．また GDR による受信を 32 KB までに制限することにより，64 KB 以上で 2 リンク利用可能になり性能が改善した．

3.4 HPL ベンチマーク

LU 分解による連立一次方程式の求解を行うベンチマーク [9] であり，倍精度行列積演算 (Level-3 BLAS DGEMM) の性能がベンチマーク結果に大きな影響を与えることが知られている．

NVIDIA より提供された P100 用のバイナリ 2.13.17 のうち，Open MPI 1.10.2 向けにコンパイルされたものを用いた．

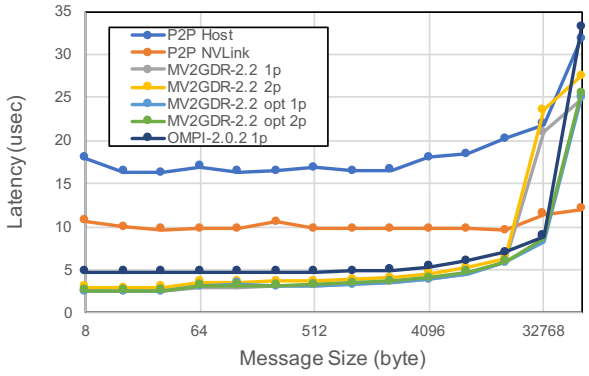

図 5 GPU 間転送・通信遅延時間

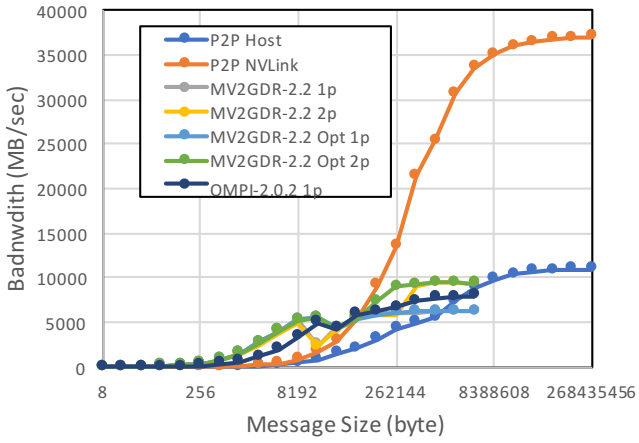

図 6 GPU 間転送・通信バンド幅性能

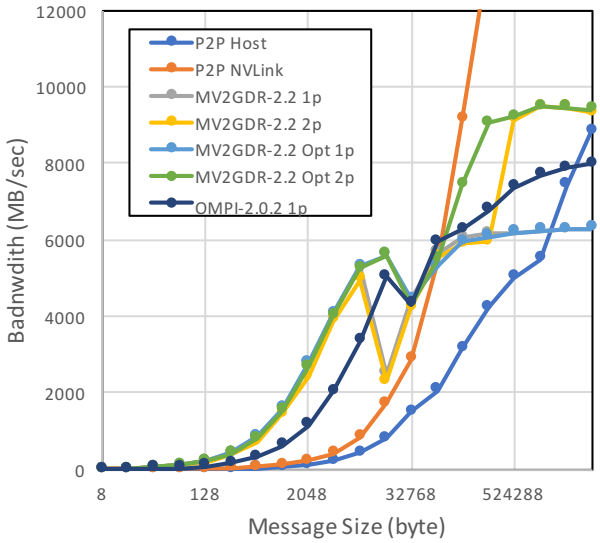

図 7 GPU 間転送・通信バンド幅性能（拡大）

3.4.1 ノード単体性能

計算ノード 1 ノードにおいて，2 MPI プロセスを用い，それぞれに P100 を 1 つずつ割り当てた．MPI ライブラリは，Open MPI 2.0.2 を用い，GDR を有効にした．問題パラメータ (HPL.dat) は以下の通り設定した．

- $N = 67200$, $NB = 384$, $P = 2$, $Q = 1$

実行の結果，10.04 TFLOPS の性能を得た．

アルゴリズムの一部は CPU で実行されているため，計算ノード全体での理論ピーク演算性能を計算すると，CPU については 1.2 TFLOPS, GPU については 10.6 TFLOPS であり，合計で 11.8 TFLOPS となる．従って，ピーク性能の 85% を得られたことがわかる．

表 5 複数ノードにおける HPL 実行

ノード数	N	NB	P	Q	性能 (TF)	ピーク性能比
2	96000	384	2	2	20.3	86.0%
4	130176	384	2	4	35.8	75.8%
8	192000	384	2	8	56.3	59.6%

3.4.2 複数ノードでの性能

計算ノード 2, 4, 8 ノードを用い，ノード単体性能と同様に実行を行った．MPI ライブラリは Open MPI 1.10.5 を用いた．（GDR は有効にしていない．）問題パラメータと性能は**表 5** の通りである．

3.5 HPCG ベンチマーク

HPCG は，HPC システムのための，より実アプリケーションに近いベンチマークとして提案されているもので，有限要素法から得られる疎行列を対象として共役勾配法 (Conjugate Gradient, CG 法) を用いて連立一次方程式を解く部分の演算性能を求めるものである．[10]

NVIDIA から提供されたバイナリ (v2) のうち，Open MPI 1.10.2 用にコンパイルされたものを用いた．

3.5.1 ノード単体性能

計算ノード 1 ノードにおいて，2 MPI プロセスを用い，それぞれに P100 を 1 つずつ割り当てた．MPI ライブラリは，Open MPI 2.0.2 において GDR を有効にした．問題パラメータ (hpcg.dat) は以下の通り設定した．

- $nx = ny = nz = 256$

実行の結果，226.2 GFLOPS の性能を得た．これは理論ピーク性能の 1.9%である．

3.5.2 複数ノードでの性能

計算ノード 4, 8, 32 ノードを用い，ノード単体の場合と同様に，1 ノードあたり 2 MPI プロセスを用い，それぞれに P100 を割り当てた．問題パラメータはノード単体の場合と同じである．

MPI ライブラリには，Open MPI 2.0.2 において GDR を有効にした．

実行の結果，856.7, 1704.9, 6711.5 GFLOPS の性能を得た．これはいずれも理論ピーク性能の 1.8%である．

4. おわりに

本稿では，2017 年 3 月に稼働を開始した Reedbush-H サブシステムを用いて性能評価を行った．Reedbush システムは，様々な最新技術を導入しており，コモディティ技術をベースにしていることから安定動作はしているものの，様々な性能最適化の余地がある．特に，今回導入された P100 に関連して，デバイスドライバ，GPUDirect for RDMA と，CUDA 開発環境，MPI ライブラリ，OpenACC コンパイラなど，様々な開発が続けられている最中である．これらの組み合わせによって正常に動作しない場合があったり，最適に実行可能なオプションの組み合わせがわかりにくいなど，今後のユーザ利用に向けて改善すべき点が多数残されている．

運用開始から日が浅く本報告では十分な実験期間が取れなかったが，今後は Reedbush-H 上で動作する GPU アプリケーションの最適化について研究を進めていく予定である．

また，機械学習やビッグデータのアプリケーションに対しても，性能評価を行い，Reedbush システムの適合性と，今後の研究開発につなげていく．

謝辞 Reedbush システムの実験にご協力いただいた，日本 SGI 株式会社，NVIDIA Japan, および東京大学情報基盤センタースーパーコンピューティング研究部門の皆様に感謝します．

参考文献

[1] 塙 敏博，中島研吾，大島聡史，伊田明弘，星野哲也，田浦健次朗：データ解析・シミュレーション融合スーパーコンピュータシステム Reedbush-U の性能評価，情報処理学会研究報告，Vol. 2016-HPC-156 (2016).
[2] : OpenACC, OpenACC-standard.org (online), available from ⟨http://www.openacc.org⟩ (accessed 2017-03-17).
[3] : Whitepaper: NVIDIA Tesla P100, NVIDIA (online), available from ⟨https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf⟩ (accessed 2016-08-15).
[4] McCalpin, J. D.: Memory Bandwidth and Machine Balance in Current High Performance Computers, *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pp. 19–25 (1995).
[5] McCalpin, J. D.: STREAM Sustainable Memory Bandwidth in High Performance Computers, University of Virginia (online), available from ⟨http://www.cs.virginia.edu/stream/⟩ (accessed 2016-08-15).
[6] Davis, T. A. and Hu, Y.: The University of Florida Sparse Matrix Collection, *ACM Transactions on Mathematical Software*, Vol. 38, No. Issue 1, pp. 1:1–1:25 (online), available from ⟨http://www.cise.ufl.edu/research/sparse/matrices⟩ (2011).
[7] Panda, D. et al.: OSU Micro-Benchmarks 5.3.2, Ohio State Univ. (online), available from ⟨http://mvapich.cse.ohio-state.edu/benchmarks/⟩ (accessed 2017-03-17).
[8] Rossetti, D.: *NVIDIA/gdrcopy: A fast GPU memory copy library based on NVIDIA GPUDirect RDMA technology*, NVIDIA Corp.
[9] Petitet, A., Whaley, R. C., Dongarra, J. and Cleary, A.: HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers, ICL, University of Tennessee (online), available from ⟨http://www.netlib.org/benchmark/hpl/⟩ (accessed 2016-08-15).
[10] Dongarra, J., Heroux, M. and Luszczek, P.: HPCG Benchmark, ICL UT, SNL (online), available from ⟨http://www.hpcg-benchmark.org⟩ (accessed 2016-08-15).