

簡易 SSP (Simplified Sustained System Performance) 指標の提案

辻 美和子^{1,a)} 佐藤 三久¹

概要：LINPACK や HPCG などのベンチマークはシステムの性能指標として広く用いられるが、これらのスコアは、必ずしも実アプリケーション実行時の性能に関する知見を与えない。一方、実アプリケーションやミニアプリを用いた性能評価は、最適化ノウハウが蓄積されている伝統的なベンチマークと比較すると、チューニングに労力を要する。そこで、本稿では、実アプリケーション・セット上でのシステム性能を予測するための、ベンチマーク・セットを用いた性能評価指標を提案する。

1. はじめに

よりよいシステムを構築するためには、システムのよさを評価する指標が必要である。たとえば、スーパーコンピュータの調達や設計の際には、複数の候補システムを比較・検討するために指標が用いられる。また、指標が存在することで、スーパーコンピュータのランク付け — たとえば Top500[16] — が可能となり、ランキングを通して、HPC システムのトレンドや将来の課題を議論することができる。

ベンチマークとしてもっともよく用いられるものは HPL (High Performance Linkack) [1] であり、上述のスーパーコンピュータのランキング Top500 も HPL のスコアに基づいてランク付けがなされる。しかしながら、近年のアプリケーションの複雑化と多様化に伴い、HPL のスコアと実際のアプリケーションとの性能には強い相関がない、という主張もある [4][2]。これを改善するために、新たなベンチマークとして HPCG (High Performance Conjugate Gradient) [4] などが提案されている。しかし、HPCG はメモリバンド幅に大きく依存することがすでに知られており、HPCG でスコアの低いシステムであっても、アプリケーションによっては十分な性能を示すこともある。

システムを評価するためのもうひとつの方法は、アプリケーション・セットやミニアプリケーション・セットを使うことである。実際のアプリケーションに即した評価を平易に行うためのミニアプリケーション・セットを構築する試みは、理化学研究所などが開発する Fiber[14] など、近

年では積極的に行われている。

Kramer ら [10] は、複数のアプリケーションおよびデータ・セットに対するシステムの性能を比較するための指標として、Sustained System Performance (SSP) を提案した。SSP については 2 章で詳しく述べる。アプリケーションの実行性能をシステム性能の指標とすることで、アプリケーションの要求を広く満足するシステムを選択、あるいは設計することが可能になる。しかしながら、アプリケーションを用いた性能評価は、HPL や HPCG などの最適化ノウハウが蓄積されているベンチマークと比較して、チューニングに多くの労力を要する。

本稿では、実アプリケーション・セット上でのシステム性能を予測するための、ベンチマーク・セットを用いた性能評価指標である SSSP (Simplified SSP) を提案する。SSSP は、SSP の手法を (ミニ) アプリケーション・セットの性能ではなく、ベンチマーク・セットの性能に対して適用する。本研究の目的は、複数のベンチマークを効果的に組み合わせることで、実アプリケーションを用いる場合よりも容易に、実アプリケーションの性能を予測できる、指標を構築することである。

提案指標 SSSP を SSP と比較して評価するために、2 つのシステムにおいてベンチマーク・セットおよびミニアプリケーション・セットを実行し、それぞれの指標を計算した。実験の結果、SSSP は SSP に対して一貫性を持つスコアを与えた、すなわち SSP でより優れたシステムは、SSSP でもより高いスコアを示した。しかしながら、両者の数字には乖離があり、重み付けなどを用いた指標の改善が必要であることがわかった。

本稿の構成は以下である：次章において、既存のシステ

¹ 理化学研究所計算科学研究機構
RIKEN Advanced Institute for Computational Science
^{a)} miwako.tsuji@riken.jp

ム評価指標である SSP 指標について述べる。3 章で新たな指標である SSSP 指標を提案する。4 章で予備実験を行い、提案指標と SSP 指標を比較するとともに、提案指標の改善について議論する。5 章で関連研究について延べ、6 章でまとめと今後の課題について述べる。

2. Sustained System Performance (SSP)

2.1 SSP の概要

本章では、アプリケーションのスループットを用いてシステムを評価する指標である Sustained System Performance (SSP) [10] について述べる。

計算機センターなどにおけるシステムの調達においては、システムが各センターの典型的なアプリケーションをいかに効率的に実行するかという観点で、複数のシステムを公平に評価する指標が求められてきた。このためのアプローチとして、米国立エネルギー研究科学計算センター (National Energy Research Scientific Computing Center, NERSC) らは、SSP 指標を開発した。SSP 指標は、調達時のみならず、システムのライフタイムおよび、将来システムにおいても以下のような役割を果たすことを目的として設計されている：

- 調達するシステムの選択のための評価
- 選択されたシステムの性能の検証
- 稼働期間中のシステムの保証
- 将来のシステム設計の指標

また、SSP の設計においては、以下の性質を満たすことが留意された：

- Proportionality：推定されるシステム性能と、実際の性能とが比例すること
- Reliability：システム A がシステム B より優れた指標値のとき、稼働後のアプリケーション実行においてもシステム A がシステム B よりも優れた性能を示すこと
- Consistency：すべてのシステムに対して、指標の定義が一定であること
- Independence：特殊な命令などによって、実アプリの性能に影響がないにも関わらず、指標の値のみが底上げされないこと
- Ease of use：より多くの人を使いやすい指標であること
- Repeatability：複数のテストを繰り返しても同じ、もしくは似た結果がでること
- Avoid misleading conclusions：誤解を招かないこと

2.2 SSP 指標の定義

本節では、SSP 指標の定義について簡単に述べる。運用期間中のシステムのアップデートやノード数の増加などを

表 1 SSP で用いられる記法

I	ターゲット・アプリケーション i の集合
J_i	アプリケーション i のデータセットの集合
$f_{i,j}$	アプリケーション i がデータセット $j \in J_i$ を実行する時の総リファレンスオペレーションカウント。本稿では Flop を用いる
$m_{i,j}$	アプリケーション i がデータセット $j \in J_i$ を実行する時に用いられるプロセッサ数 (ノード数)
$a_{i,j}$	プロセッサごとの仕事量
	$a_{i,j} = f_{i,j}/m_{i,j}$
$t_{i,j}$	アプリケーション i がデータセット j を実行するのに要する時間
$p_{i,j}$	アプリケーション i がデータセット j を実行する場合のプロセッサごとの性能 (per processor performance)
	$p_{i,j} = a_{i,j}/t_{i,j} = f_{i,j}/(m_{i,j} \times t_{i,j})$
w_i	アプリケーション i の重み
N	システムの総プロセッサ数

考慮したよりくわしい定義は、Kramer ら [9][10] を参照されたい。

表 1 に、SSP で用いられる数式および記号を示す。

I をターゲット・アプリケーション i の集合とし、 J_i をアプリケーション i のデータセットの集合とする。ここで、 $1 \leq i \leq |I|$ である。アプリケーション i がデータセット $j \in J_i$ を実行する時の総リファレンスオペレーションカウントを $f_{i,j}$ とする。 $f_{i,j}$ は、コード解析もしくはリファレンスシステムにおいてパフォーマンスカウンタを用いることで得られる。 $m_{i,j}$ は、アプリケーション i がデータセット j を実行するときの並列数である。なお、SSP では、 $m_{i,j}$ の決定法について、厳密に規定していない。これらを用いて、アプリケーション i がデータセット j を実行するときのプロセッサごとの仕事量 $a_{i,j}$ は以下のように計算される：

$$a_{i,j} = \frac{f_{i,j}}{m_{i,j}}. \quad (1)$$

アプリケーション i がデータセット j を実行するのにかかる時間が、 $t_{i,j}$ であるとき、プロセッサごとの性能 (per processor performance) $p_{i,j}$ は、以下のように計算される：

$$p_{i,j} = \frac{a_{i,j}}{t_{i,j}} = \frac{f_{i,j}}{m_{i,j} \times t_{i,j}}. \quad (2)$$

システム s の SSP の値は、すべてのアプリケーションおよびデータセットに対するプロセッサごとの性能の重み付き和にプロセッサ数を乗じて、以下のように求められる：

$$SSP(s) = N \frac{\sum_{i \in I} w_i \sum_{j \in J_i} p_{i,j}}{\sum_{i \in I} w_i} \quad (3)$$

$$= N \sum_{i \in I} w_i \sum_{j \in J_i} \frac{f_{i,j}}{m_{i,j} \times t_{i,j}} / \sum_{i \in I} w_i. \quad (4)$$

w_i は、アプリケーションの重みであり、過去のシステムにおいて、各アプリケーションが利用したノード時間などから算出される。 N は、システムの総プロセッサ数である。

表 2 Blue Waters における CPU ノードでのアプリケーションセット [8] より

カテゴリ	アプリ名
分子動力学	NAMD v2.0
量子モンテカルロ法	QMCPACK v52
量子色動力学	MILC 7.6.3
量子化学	NWChem 6.1
気象・天気	WRF 3.3.1
地震学	SpecFEM3D 5.13
恒星大気・超新星	VPIC
プラズマ・磁気圏	PPM-7/2/12

表 3 Blue Waters における GPU ノードでのアプリケーションセット [8] より

カテゴリ	アプリ名
分子動力学	NAMD
量子モンテカルロ法	QMCPACK
量子色動力学	CHROMA
量子化学	GAMESS 6.1

上式が示すように、SSP 指標は、 N プロセッサからなるシステムにおける、さまざまな並列度を持つアプリケーションのスループットの期待値として定義される。

上記の重み付き相加平均の代わりに、調和平均や幾何平均を用いることも可能である。たとえば、CPU と GPU からなる Heterogeneous System であった Blue Waters に対して Bauer ら [2] は、両者の幾何平均の和を用いた：

$$^8\sqrt{\prod_{i=1}^8 p_i} \cdot N_{CPU} + ^4\sqrt{\prod_{i=1}^4 p_i} \cdot N_{GPU}. \quad (5)$$

ここで、 N_{CPU} は CPU ノード数、 N_{GPU} は GPU ノード数である。

2.3 Sustained Petascale Performance (SPP)

Sustained Petascale Performance (SPP) 指標は、SSP のインスタンスであり、アメリカ国立科学財団 (National Science Foundation, NSF) とイリノイ大学が Blue Waters を調達する際に用いられた [2][18][8][12]。

サイエンスチームの要求を反映するアプリケーションおよびデータセットが選択され、候補システムにおける性能を推定され、SPP 指標が計算された。ここで、Blue Waters は、CPU と GPU からなるシステムであり、それぞれにアプリケーションが想定された。表 2-3 に SPP で用いられたアプリケーションを示す。

これらのアプリケーション・セットは、構造格子、非構造格子、密行列、疎行列、FFT、N-Body、高い I/O 要求、などさまざまな計算機的特徴をカバーする。結果として得られた Blue Waters の SPP スコアは 1.31 PFlops であり、同様のアプリケーション・セットを用いた場合の Titan の SPP スコアの推定値 0.64 PFlops よりも高いとしている [8]。

3. Simplified Sustained System Performance (SSSP)

本章では、本稿で提案する簡易 SSP (Simplified Sustained System Performance) 指標について述べる。

3.1 SSSP 指標の目的

2章で述べたように SSP/SPP 指標を用いることで、アプリケーション・セットのスループットの観点から、システム性能を数値化し、比較・検討・改善することが可能になる。アプリケーション・セットを用いた指標の利点は、それらのアプリケーションの性能を指標に直接反映できることである。

一方で、システムの適切な指標を得るためには、アプリケーションをシステムに移植し、場合によっては最適化を行う必要がある。しかしながら、複数の分野にまたがるアプリケーションのボトルネックをただしく理解し、要求するライブラリなどを調査し、性能を見積もることは、多くの労力を要する。特に、複数システムを比較する際には、これらの作業を複数回にわたって繰り返す必要がある。

そこで、本稿では、実アプリケーション・セット上でのシステム性能を予測するために、ベンチマーク・セットを用いた性能評価指標である Simplified Sustained System Performance (SSSP) 指標を提案する。SSSP 指標は、SSP 指標と同様に複数の問題と複数のデータセットから計算されるが、問題として SSP 指標で用いられたアプリケーション・セットにかわって伝統的なベンチマークのセットが用いられる。HPL や HPCG に代表されるベンチマークは、計測がしやすく、移植が容易で、最適化についてもすでに方法論が確立していることから、複数のシステムの比較がより平易に行えるようになることが期待される。

SSSP 指標は、アプリケーション・セットを用いた SSP 指標と異なり、ダイレクトにアプリケーションの性能を反映するわけではない。しかし、本研究においては、複数の性質の異なるベンチマークを適切に組み合わせることで、アプリケーション・セット上でのシステム性能を予測することを目的とする。

3.2 SSSP 指標の定義

SSSP 指標においては、SSP 指標と同様の式を、アプリケーション・セットやミニアプリ・セットではなく、ベンチマーク・セットを用いて計算する。

実アプリケーション・セット上でのシステム性能を予測するために SSSP 指標は以下を満たす必要がある

$$\begin{aligned} \forall s, s' \in S, \text{SSP}(s) < \text{SSP}(s') \\ \implies \text{SSSP}(s) < \text{SSSP}(s') \end{aligned} \quad (6)$$

ここで、 S は考慮されるシステムの集合とする。すなわ

ち、実アプリケーション・セット上での性能指標 SSP において、システム s' がシステム s を上回るとき、提案する簡易指標である SSSP 指標においてもシステム s' がシステム s よりも優れた値を示すように SSSP 指標を構築する必要がある。

より正確な指標を与えるためには、

$$\min \sum_{s \in S} |\text{SSSP}(s) - \text{SSP}(s)| \rightarrow 0 \quad (7)$$

のようにすべての対象システム s において、SSP 指標と SSSP 指標が同じ値を示すことが望ましい。

4. 実験

4.1 テストシステム

本実験では、京コンピュータおよび FX10 を用いて、アプリケーション・セットおよびベンチマーク・セットを実行し、それぞれの SSP および SSSP を計算した。

表 4 に、京コンピュータおよび FX10 の仕様を示す。京コンピュータは 8 万ノード以上を持つシステムである一方で、本実験で用いた FX10 は 96 ノードの小規模なシステムである。比較のために、本稿では、京コンピュータの 96 ノードのみを考える。

4.2 Fiber Miniapp Suite

Fiber Miniapp Suite [14] は、将来の HPCI システムのあり方の調査研究・アプリケーション分野のロードマップ [5] を解決するためのアプリケーション (フルアプリケーション) から抽出したミニアプリ集である。

本実験では、Fiber から表 5 にしめす 7 種類のアプリケーションを用いて SSP を計算した。表 6 にそれぞれのアプリケーションおよびデータで用いたノード数、MPI プロセス数、OpenMP スレッド数を示す。

4.3 ベンチマーク・セット

SSSP 指標計算のためのベンチマークとしては表 7 に示すものを用いた。本実験では、各ベンチマークおよびライブラリの開発者によるリファレンス実装をそのまま実行し、個別の最適化は行っていない。このため、HPL などの結果は、すでに公表されている Top500 でのスコアとは異なる。

4.4 予備実験の結果

アプリケーション・セットおよびベンチマーク・セットに関して、性能を測定し、式 (3) および (5) を用いて、SSP 指標と SSSP 指標を計算した。すべての重みを等しく $w_i = 1$ とすると、使用したデータ・セットが他よりも少ない NGS Analyzer と FFB が式全体に与えるインパクトが小さくなってしまふ。これを補正するために、これらのアプリケーションについては $w_i = 2$ とし、他のアプリケーションについては $w_i = 1$ とした。

表 8 に各アプリケーションの per processor performance $p_{i,j}$ を、表 9 に各ベンチマークの per processor performance $p_{i,j}$ を、それぞれ示す。また、これらの $p_{i,j}$ を用いて計算された京コンピュータおよび FX10 の SSP 値と SSSP 値を表 10 に示す。

表 10 から、SSSP 指標は条件 (6) をみたとす、すなわち、アプリケーション・セットに対する指標である SSP で京コンピュータを上回った FX10 は、ベンチマーク・セットに対する指標である SSSP でも京コンピュータよりも高い数字を示した。しかしながら、SSP(K) と SSSP(K) の数字、および SSP(FX10) と SSSP(FX10) のスコアには開きがある。またスコアの比率を見ると、相加平均の場合で、SSSP(FX10)/SSSP(K) がより理論ピーク性能比 (1.65) に近い値 1.49 を示すいっぽうで、SSP(FX10)/SSP(K) は 1.19 であった。また、相乗平均においても、SSSP(FX10)/SSSP(K) = 1.28 である一方で、SSP(FX10)/SSP(K) = 1.11 となった。これは、SSSP 指標の構成要素のうちでも、HPL が高い per processor performance $p_{i,j}$ を示し、全体に関して支配的であるからであると考えられる。

4.5 考察

4.4 で明らかになったように、Flops から計算される SSSP 指標は、要求 B/F 値が小さく高い Flops を出しやすい HPL のスコアに大きく影響される。SSSP 指標の計算には要求 B/F 値が大きい HPCG も含まれるが、HPCG の Flops は HPL の 1/50 ~ 1/75 程度であることから、仮にシステムのメモリバンド幅が大幅に改善され Flops が数倍になったとしても、SSSP スコア全体に与える影響はわずかである。ただし、同様の問題 — Flops の出しやすいアプリケーションがスコアに対して支配的になる — は、SSP にも存在すると考えられる。

予備実験から明らかになった SSSP 指標の課題のひとつは、各ベンチマークのスコアに対する影響を正規化することである。2 章で述べたように、SSP 指標の方法論では、過去のシステムにおけるそのアプリケーションの実行時間などの基準としたアプリケーションの重み付けをすることも提案されているが、正規化は重要なアプリケーションやベンチマークを重み付けする作業とは異なり、その前段階として実施されるべき処理であると考えられる。

また、予備実験では少数のベンチマークしか用いなかったが、これはいくつかの重要なベンチマークが Flops とは異なる指標を用いているためである。たとえば、グラフ構造の幅優先探索処理性能を競う Graph500 [6] では、Traversed Edges Per Second (TEPS) という、1 秒間にたどるグラフの枝の数を基準としている。メモリバンド性能を計測する HPCG の Stream では TB/s が単位となっている。また、近年では、ビッグデータと呼ばれる大規模な

表 4 Specifications for K and FX10 supercomputers

		K	FX10
Node	CPU	SPARC64 TM VIIIfx 2GHz	SPARC64 TM IXfx 1.65GHz
	Performance	128 GFlops	211.2Gflops
	Number of cores	8 cores	16 cores
	Memory	16GB, 64GB/s	36GB 85GB/s
Total number of nodes		82,944	96
Number of nodes considered		96	96
Network		Tofu Interconnect 5GiB/s x 2	Tofu Interconnect 5GiB/s x 2

表 5 本実験で用いられたアプリケーション

アプリ名	カテゴリ	計算機の特徴
CCS-QCD 1.2.1	量子色力学	構造格子モンテカルロ
FFVC-MINI 1.0.1	熱流体解析	3次元キャビティフロー
NICAM-DC-MINI 1.0	気象	構造格子ステンシル
mVMC-MINI 1.0	物質科学	多変数変分モンテカルロ法
NGS-Analyzer-MINI 1.0.1	次世代シーケンサー出力解析	複数タスクからなるワークフロー
NTChem-MINI 1.1	量子化学	分子軌道法
FFB-MINI 1.0.1	熱流体解析	非構造有限要素法

データを扱うアプリケーションへの需要が高まっており、I/O 性能も考慮して SSSP 指標を構築する必要がある。このような Flops 以外のスコアをどのように SSSP に取り込んでいくかも今後の重要な課題である。

5. 関連研究

1 章でも述べたように、実際のアプリケーションに即した評価を平易に行うためのミニアプリケーション・セットは、さまざまな機関により構築され、提供されている。たとえば、理化学研究所などが開発する Fiber[14]、Sandia National Laboratories が開発する Mantevo[15]、DoE が主導する ExaCT[3]、Lawrence Livermore National Laboratories の LULESH[11]、米イリノイ大学のプログラミング言語 Charm++ のためのミニアプリケーション・セット [17] などである。これらのミニアプリケーション・セットは、SSP や SSSP と異なり、ミニアプリケーション・セット全体に対するシステムの評価を数値として与えるわけではない。また、ミニアプリケーション・セットのコードやデータサイズは特定のシステムに対してとくに最適化されている場合があり、複数の機関の複数のシステムを比較する際にはポーティングが難しい場合もある。

複数のベンチマークをセットにしてシステムを評価する試みは、HPC Challenge Benchmark[7] や Nas Parallel Benchmark [13] などがあるが、ミニアプリケーション・セットと同様に SSP/SSSP のように単一の指標を与えないほか、実アプリケーション性能との関連性の議論が十分ではない。

これらの既存研究に対して、本稿で提案する SSSP の新たな試みは、複数のベンチマーク・セットに対して

- (1) 1 システム、1 指標を与える
- (2) SSP 指標の近似することで、実アプリケーションの性能予測を可能にする
ことである。

6. おわりに

本稿では、実アプリケーション・セット上でのシステム性能を予測するための、ベンチマーク・セットを用いた性能評価指標である SSSP 指標を提案した。アプリケーションよりも移植や最適化が容易なベンチマークを用いる提案指標によって、複数のシステムの比較が容易になることが期待される。

予備実験を行い、提案指標がアプリケーション・セットの性能を適切に予測しているかどうかを評価した。提案する SSSP 指標は、アプリケーション・セット上での性能から計算される SSP 指標と一貫するシステムのランク付けを与えることができた。しかしながら、SSSP 指標と SSP 指標の数字には開きがあり、また、SSSP 指標は性能の出やすいベンチマークの結果に大きく影響されることがわかった。

今後の課題としては以下が挙げられる：

- 4.5 節で述べたように、Flops の出やすいアプリケーションとそうでないアプリケーションを同等に扱うための正規化を導入する。
- Graph500 などの Flops 以外の基準を持ったベンチマークを指標に組み込むことを検討する。
- より多くのベンチマーク、アプリケーション、およびシステムを用いた実験・検討を行う。とくに I/O ベンチマークやデータ・インテンシブなアプリケーション

表 6 The number of nodes, MPI-processes and OpenMP-threads for applications

Application	Data	K			FX10		
		# of nodes	processes	threads	# of nodes	processes	threads
CCS-QCD	Class1	1	1	8	1	1	16
	Class2	64	64	8	64	64	16
FFVC	256 ³	1	1	8	1	1	16
	1024 ³	64	64	8	64	64	16
NICAM-DC	gl05rl00z80pe5	5	5	8	5	5	16
	gl05rl00z40pe10	10	10	8	10	10	16
mVMC	tiny	1	1	8	1	1	16
	middle	64	64	8	64	64	16
NGS-Analyzer	bwa	6	6	8	6	6	16
NTChem	h2o	2	2	8	2	4	8
	taxol	32	32	8	32	64	8
FFB	test	8	8	8	6	8	16

表 7 The number of nodes, MPI-processes and OpenMP-threads for traditional benchmarks

Benchmark	Data	K			FX10		
		# of nodes	processes	threads	# of nodes	processes	threads
HPL 2.2	Ns=80000	4	32	1	2	32	1
	Ns=160000	16	128	1	8	128	1
Himeno BMTxp	M	1	1	8	1	1	16
	L	1	1	8	1	1	16
FFTE 6.0	512 ³	8	8	8	8	8	16
	1024 ³	64	64	8	64	64	16
HPCG 3.0	256 ³	64	512	1	32	512	1
	512 ³	64	512	1	32	512	1

を考慮する

謝辞

論文の結果 (の一部) は, 理化学研究所のスーパーコンピュータ「京」を利用して得られたものである。また, イリノイ大学の William Kramer 博士には貴重なご助言をいただいた。ここに謝意を表する。

参考文献

- [1] <http://www.netlib.org/benchmark/hpl/>.
- [2] G. Bauer, T. Hoefer, W. Kramer, and R. Fiedler. Analyses and modeling of applications used to demonstrate sustained petascale performance on blue waters. In *Proceedings of the Annual Meeting of the Cray Users Groups CUG-2012*, p. online, 2012.
- [3] DoE. ExaCT <http://exactcodesign.org/>.
- [4] J. Dongarra and M. A. Heroux. Toward a new metric for ranking high performance computing systems. Technical Report SAND2013-4744, Sandia National Laboratories, 2013.
- [5] Feasibility Study on Future HPC Infrastructures (Application Working Group). Computational Science Roadmap <http://hpci-aplfs.aics.riken.jp/>, 2014.
- [6] Graph 500. <http://www.graph500.org/>.
- [7] HPC Challenge. <http://www.hpcchallenge.org/index.html>.
- [8] W. T. Kramer. Measuring sustained performance on blue waters with the spp metric, cug-2013.
- [9] W. T. Kramer. *PERCU: A Holistic Method for Evaluating High Performance Computing Systems*. PhD thesis, EECS Department, University of California, Berkeley, Nov 2008.
- [10] W. T. Kramer, J. Shalf, and E. Strohmaier. The NERSC sustained system performance (SSP) metric. Technical Report LBNL-58868, 2005.
- [11] Lawrence Livermore National Lab. LULESH <https://codesign.llnl.gov/lulesh.php>.
- [12] C. L. Mendes, B. Bode, G. H. Bauer, J. Enos, and W. T. K. Cristina Beldica. Deployment and testing of the sustained petascale blue waters system. *Journal of Computational Science*, 10:327–337, 2015.
- [13] NAS Parallel Benchmarks. <http://www.nas.nasa.gov/publications/npb.html>.
- [14] RIKEN Advanced Institute for Computational Science. Fiber Miniapp Suite <http://fiber-miniapp.github.io/>.
- [15] Sandia National Laboratories. Mantevo Project <https://mantevo.org/>.
- [16] Top500 Supercomputer Sites. <https://www.top500.org/>.
- [17] University of Illinois. <http://charmplusplus.org/miniApps/>.
- [18] William TC Kramer. Blue Waters and the Future of @Scale Computing and Analysis, 4th AICS Workshop December 2013.

表 8 Per processor performance (GFlops) for applications

Application	Data	K	FX10
CCS-QCD	Class1	18.4	24.7
	Class2	11.0	11.0
FFVC	256 ³	14.0	21.5
	1024 ³	12.7	17.4
NICAM-DC	gl05rl00z80pe5	5.70	7.11
	gl05rl00z40pe10	4.90	5.18
mVMC	tiny	4.38	3.41
	middle	19.1	25.2
NGS-Analyzer	bwa	0.0106	0.00940
NTChem	h2o	10.9	12.16
	taxol	61.2	67.3
FFB	test	5.72	6.19

表 9 Per processor performance (GFlops)
 for traditional benchmarks

Benchmark	Data	K	FX10
HPL	80000	96.5	149
	160000	97.2	150
Himeno	M	19.2	23.2
	L	7.56	9.70
FFTE	512 ³	3.41	3.51
	1024 ³	2.7	2.93
HPCG	256 ³	1.79	2.44
	512 ³	1.52	2.01

表 10 SSP and SSSP for the K and FX10 supercomputers (GFlops)

相加重平均	K	FX10	FX10/K
SSP (アプリケーション・セット)	1190	1420	1.19
SSSP (ベンチマーク・セット)	2750	4100	1.49
幾何平均	K	FX10	FX10/K
SSP (アプリケーション・セット)	378	418	1.11
SSSP (ベンチマーク・セット)	840	1080	1.28

式 (3) および (5) を計算. ただし, データセット数が少ないアプリケーションが式全体に与えるインパクトを調整するために, データセット数が 1 のアプリ (NGS Analyzer と FFB) は, $w_i = 2$ とし, 他のデータセット数 2 のアプリは $w_i = 1$ とした.