TSUBAME3.0 における計算資源分割の検証

阿部 公一 [1)], 香月 稔 [1)], 藤田 和宏 [2)], 鶴見 慶 [1)], 安良岡 由規 [2)], 根本 忍 [2)],
梁井 善行 [1)], 野村 哲弘 [3)], 三浦 信一 [3)], 渡邊 寿雄 [3)], 額田 彰 [3)], 遠藤 敏夫 [3)]

1) 東京工業大学 研究推進部
2) 東京工業大学 技術部
3) 東京工業大学 学術国際情報センター

computer@o.cc.titech.ac.jp

Resource Segmentation in TSUBAME3.0 Supercomputer

Masakazu Abe [1)], Minoru Katsuki [1)], Kazuhiro Fujita [2)], Kei Tsurumi [1)], Yoshinori Yasuraoka [2)],
Shinobu Nemoto [2)], Yoshiyuki Yanai [1)], Akihiro Nomura [3)], Shin'ichi Miura [3)],
Toshio Watanabe [3)] , Akira Nukada [3)], Toshio Endo [3)]

1) Research Promotion Department, Tokyo Institute of Technology
2) Technical Department, Tokyo Institute of Technology
3) Global Scientific Information and Computing Center, Tokyo Institute of Technology

概要

東京工業大学学術国際情報センターにおいて 2017 年 8 月より運用を開始したスーパーコンピュータ TSUBAME3.0 では、効率的な運用のために、計算ノードを論理的に分割することにより、より多くのジョブを同時実行するよう設計されている。本稿では、過去 1 年間の運用実績をもとに、計算ノードの論理分割が実際にどのように使われてきたかを検証する。

1 はじめに

東京工業大学学術国際情報センターは TSUBAME3.0 を 2017 年 8 月より稼働を開始した[1]。TSUBAME3.0 は HPE SGI ICE XA を基にカスタマイズされ、540 台の計算ノードには CPU1,080 基、GPU2,160 基が搭載され、理論最大性能は倍精度で 12.15 ペタフロップス、半精度(以上)で 47.2 ペタフロップスになる。

各計算ノードには 2TB の NVMe 対応 SSD が搭載され、合計容量 1.08PB のスクラッチ領域を備える。また、全計算ノードから参照可能なストレージとして容量 15.9PB、データ転送速度 150GB/s の高速ストレージが用意されている。

システムの冷却方式は TSUBAME2.0/2.5 の間接水冷および TSUBAME-KFC[2]の液浸・温水冷却等で培った経験から、主要な熱源である CPU と GPU のみを直接水冷、他のコンポーネントを間接水冷とする等の工夫を加えることにより、冷却塔による水冷効果を高め省電力性に寄与している。

本稿では、TSUBAME3.0 がより多くのユーザに効率的に利用されるために行った資源分割方式の概要と、その利用状況について報告する。

図 1: TSUBAME3.0 計算ノードラック

2 TSUBAME3.0 のノード構成

TSUBAME3.0(図 1)の計算ノード 540 台全ての構成は同一である。 1 台のノードには、CPU として Intel Xeon E5-2680 V4（14 コア, 2.4GHz）を 2 基搭載し、主記憶装置は 256GiB、バンド幅 154GB/s

を持つ。GPU としては NVIDIA Tesla P100 を 4 基搭載する。ネットワークインターコネクトは Intel Omni-Path を 4 ポート搭載し、ノードあたり片方向 400Gbps、双方向 800Gbps のインジェクションバンド幅を持つ。各ノードに搭載された 2TB の大容量 SSD は、NVMe サポートにより高速なアクセスが可能である。特にネットワークインターコネクトと SSD はビッグデータや AI 分野での処理の高速化に重要な要素である。

図 2 に TSUBAME3.0 の計算ノードのブロック図を示す。ノード内の各プロセッサ数が偶数で、上下でほぼ対称な構造であることから後述するとおり資源分割の点で有利となっている。

表 1 は TSUBAME2.5 と TSUBAME3.0 の計算ノードの比較である。

表 1: TSUBAME2.5 と TSUBAME3.0 の計算ノードの比較（数値はノード内合計）

指標	T2.5	T3.0	倍率
CPU			
コア数(HT を除く)	12	28	2.33
コア数×周波数(GHz)	35.16	72.8	2.07
メモリ容量(GiB)	54	256	4.74
メモリバンド幅(GB/s)	64	153.6	2.40
GPU			
CUDA コア数	8064	14,336	1.78
FP64(TFlops)	3.93	21.2	5.39
FP32(TFlops)	11.85	42.4	3.58
FP16(TFlops)	11.85	84.8	7.16
メモリ容量(GiB)	18	64	3.56
SSD			
容量(GB)	120	2,000	16.67
READ(MB/s)	550	2,700	4.91
WRITE(MB/s)	500	1,800	3.60
ネットワーク			
転送速度(Gbps)	80	400	5.00
ノード数	1408	540	0.38

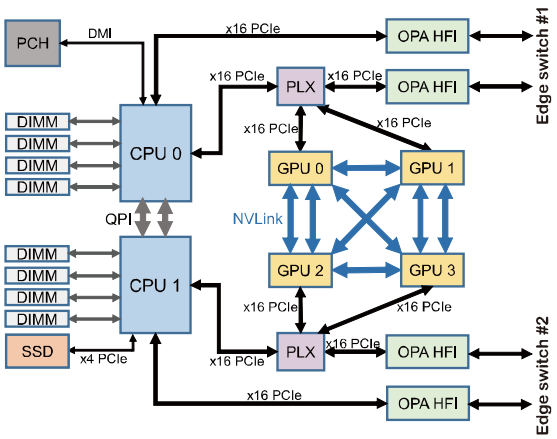

図 2: 計算ノードのブロックダイアグラム

このように、TSUBAME3.0 の計算ノードはあらゆる面で TSUBAME2.5 の計算ノードより強化されているが、ノードの総数は TSUBAME2.5 の約 38%しかなく、学内外の多数のユーザの計算需要を満たすためにより効率的な運用が求められる。

このため、より先進的な資源分割を実施し、より多くのユーザが効率的に計算ノードを利用できるようにすることが求められている。

3 TSUBAME におけるノード動的分割

前節に示したとおり、TSUBAME シリーズの計算ノードは、多数のコアを持つ CPU や複数の GPU など、ノード単体で見ても強力な計算資源である一方で、TSUBAME の利用者はスーパーコンピューティングの初心者である学生や、ISV アプリケーションユーザ、GPU コンピューティングの研究者など多岐にわたり、すべてのユーザが TSUBAME の計算ノードを余すことなく利用できるとは考えにくい。たとえば、GPU を利用しないプログラムの実行中はそのノードの GPU は利用されない遊休資源となる。逆に GPU で計算を行うプログラムは一般的には多数の CPU コアを必要としない。また、1 つのプログラムからノード内の複数の GPU を同時に使用するには複数 GPU に対応したプログラムを記述する必要があり、プログラミングの難度が上がるため、複数の GPU の利用を望まないユーザも存在する。このような要求資源の性質が異なるユーザ・プログラムを同一の計算ノードに割り当て、計算ノードを共有させることにより限られた TSUBAME の計算資源をより多くのユーザへ提供可能とする。

TSUBAME2.0/2.5 では全体の約 1/3 のノードにおいて VM 技術を用い計算ノードを GPU 部分と CPU 部分に分割し、それぞれを仮想的に別の計算ノードとして資源提供することで、GPU を中心に利用し CPU はさほど利用しないユーザや、GPU を利用せず CPU のみを利用するユーザの需要に応えてきた。しかしながら、VM およびスケジューラの資源分割が固定であるため、利用者の需要と合致しない資源が一時的に遊休化するなど、資源利用率の面の課題があった。また、当時の VM 技術の制約上、性能を犠牲にすることなく GPU を仮想化することが出来なかったため、ノードの GPU 部分は仮想化の対象とせず、GPU ジョブはベアメタルノードで実行しつつ、そのノードの中に

CPU ジョブのための VM を作成する形をとる必要があり、ノード内の 3 基の GPU を分割することはできなかった。

TSUBAME3.0 では、TSUBAME2.5 で利用してきた VM 技術に代わり、Linux cgroup を用いた資源分割を導入した。分割された各資源からは GPU および Omni-Path HFI に直接アクセスすることが可能であり、それぞれから利用できるデバイスを制限することもできるため、TSUBAME2.5 より柔軟に資源の分割が可能となる。例えば図 3 に示すように、ノード内の一部の GPU のみを切り出してノードの分割を行うことができる。

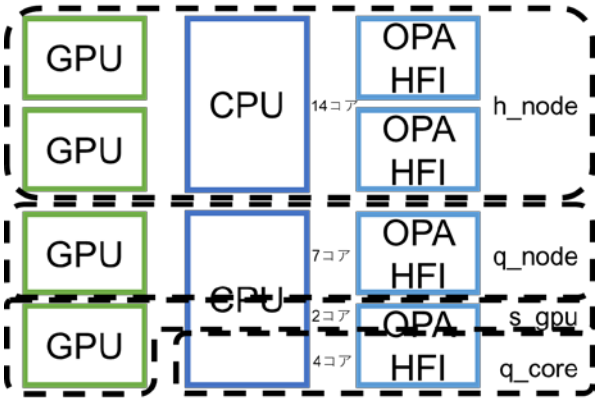

図 3: TSUBAME3.0 計算ノードの資源分割イメージ

ユーザは実行するプログラムの性質に基づき、下記の資源タイプから利用したい資源タイプを選択する。ユーザへの課金(トークンの消費)も、利用する資源タイプにもとづいて行われる。なお、下記 CPU コア数は物理コア数であり、HyperThreading により 2 倍の論理コア数を利用可能である。

- f_node: ノード全体を使用、4GPU, 28 コア
- h_node: 1/2 ノードを使用、2GPU, 14 コア
- q_node: 1/4 ノードを使用、1GPU, 7 コア
- s_gpu: GPU をメインに使用、1GPU, 2 コア
- q_core: CPU コアをメインに使用、4 コア
- s_core: CPU1 コアのみを使用、1 コア

以下、本稿では分割する前の計算ノードのことを物理ノード、分割後の各資源タイプの仮想的なノードのことを論理ノードと呼ぶこととする。

スケジューラである Univa Grid Engine[3]と連携して Linux cgroup 機構を用いた資源分割を動的に行うことにより、TSUBAME2.5 で行われていた静的なパーティショニングは不要であり、あるときは h_node として提供していた部分を次のジョブでは q_node と s_gpu と q_core に分割して提供するなど、需要に応じた物理ノードの切り出しが行えるようになったために、システム全体の計算資源が有効に活用可能となった。

4 ノード分割における制約事項

TSUBAME3.0 でノードを分割するにあたり、以下のような制約が発生する。これらの制約により実行できなくなるプログラムのユーザは、資源分割を行わない f_node 資源を使用する必要がある。

- 計算ノードへの SSH の利用
  デバッグ目的、もしくは X アプリケーションの利用のために、TSUBAME ではジョブ実行中の計算ノードへの SSH を許可しているが、cgroup による資源分割で複数のジョブが実行されている物理ノードへの SSH は、どの論理ノードへの接続かを区別できないため、利用することができない。
  対応策として、X アプリケーションの実行については、インタラクティブジョブに限定して、SSH を利用せずに実行する方法を整備し、2018 年 1 月に公開した[4]。これにより X アプリケーションのユーザも資源分割の利点を享受することができるようになった。
- ISV アプリケーションの cgroup 対応
  ISV アプリケーションによっては、cgroup によって利用できる CPU・GPU 等に制約を受けている状態では動作しないものがある。これらを利用する際には、cgroup による制約がかからない f_node を使用する必要がある。

5 ノードの利用状況

TSUBAME3.0 における 2017 年 9 月から 2018 年 7 月における資源タイプごとのジョブの論理ノード時間積の積み上げを図 4 に示す。各月の積み上げグラフの左側には実際に使用された物理ノード時間積を、右側には当月に提供していた物理ノード時間積が示されている。なお、others とされている計算時間は、TSUBAME グランドチャレンジ制度[5]によるスケジューラの利用を伴わないノード占有利用を示している。

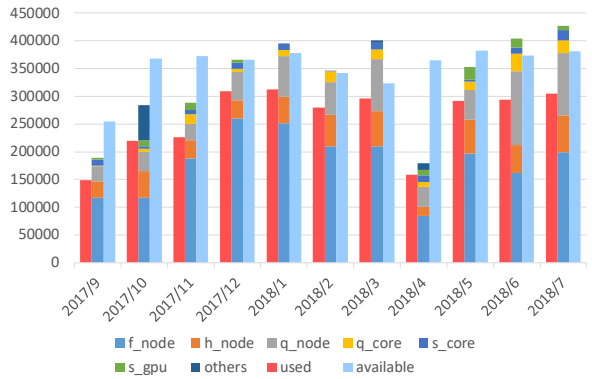

図 4: TSUBAME3.0 の資源タイプ別使用論理ノード時間および使用物理ノード時間

TSUBAME3.0 が提供可能な物理ノード時間は 30 日あたり 388,800 ノード時であるため、たとえ障害などによるノード提供不能時間がなかったとしても、図中に示す全てのジョブを資源分割することなく TSUBAME3.0 上で実行することは不可能である。計算資源を分割することにより、このようなワークロードを論理ノードで実行することが可能となった。

2018 年 7 月時点でも、論理ノード時間積で 47%のジョブは物理ノードを占有して実行しているが、その割合は徐々に減少してきている。これは、課金における優位性だけではなく、X アプリケーションなどの資源分割に伴う制約事項を解消していったことによるものと考えられる。また、ISV アプリケーションのなかに、運用期間中に GPU 対応版がリリースされたもの(例: Gaussian)があり、それらによる需要の変動も考えられる。

なお、2017 年 1 月から 3 月の間、スケジューラの資源分割に関する不具合があり、s_gpu の資源が適切に分離できなかったため、この期間は s_gpu の利用を禁止しており、利用量が 0 となっている。

5　今後に向けて

TSUBAME3.0 では現在、cgroups によるアクセス制限によりノードを論理的に分割しているが、これをより推し進めて、コンテナ技術によるノードの仮想化を行うことを計画している。コンテナによる仮想化を行うことで、ユーザは自らのプログラムに最適な環境を持ち込むことが可能になることに加え、コンテナで実現されるアプリケーション用ユーザランドと基盤となる物理ノードのユーザランドを分離することにより、それぞれを独立に更新することができるようになり、セキュリティ問題の対処が行いやすくなると期待している。

TSUBAME3.0 導入当初は、スケジューラと連携させて Docker を利用できるように計画していたが、こちらはテスト段階においてスケジューリングに関する不具合が発生したため、ユーザへの提供に至っていない。また、Docker のセキュリティモデルと複数のユーザが計算ノードや並列ファイルシステムを共有するスパコン環境の親和性は高くなく、データセキュリティのため、計算ノード上で実行できる Docker コンテナを何らかの形で制限することが必要となり、コンテナの利点を最大限に活用できる形でユーザに提供できる道筋が立っていない。

一方、コンテナの実行に root 権限を常用しない Singularity がこのような環境におけるコンテナ実現手段として有力となっており、本稿執筆時点でユーザへの提供を開始したところである。コンテナ内から OmniPath の高速な通信を利用する方法など、多少の困難はあるものの、ユーザが任意の環境を持ち込んでプログラムを TSUBAME で実行できるというコンテナ技術の利点を十分に果たすものとして期待している。

6　おわりに

本稿では、東京工業大学のスーパーコンピュータ TSUBAME3.0 において、ノードの論理分割を用いた限りある計算資源の有効活用の取り組みと、その利用状況について報告した。計算資源の仮想化を行うことにより、ユーザはより扱いやすい形で計算ノードにアクセスすることができるとともに、ジョブスケジューリングの観点からも物理的なノード数を超えるジョブを実行することが可能となり、ジョブの実行開始時間の減少も期待される。また、ノードの仮想化を推進することによりユーザがプログラムを実行環境ごと持ち込めるようにするコンテナ化の展望もあり、実装を推し進めているところである。

謝辞

TSUBAME3.0 の設計および運用には、東京工業大学学術国際情報センターが推進してきた文部科学省「スパコン・クラウド情報基盤におけるウルトラグリーン化技術」および「スマートコミュニティ実現のためのスパコン・クラウド情報基盤のエネルギー最適化の研究推進」、JST CREST(JPMJCR1303, JPMJCR1501) などのプロジ

ェクトの研究成果が活用されている。

参考文献

[1] 藤田 和宏 ほか 「新スーパーコンピュータ TSUBAME3.0 の概要」大学 ICT 推進協議会 2017 年度年次大会 (2017)

[2] T. Endo, A. Nukada, S. Matsuoka. TSUBAME-KFC: a Modern Liquid Submersion Cooling Prototype towards Exascale Becoming the Greenest Supercomputer in the World. IEEE ICPADS 2014, pp.360-367 (2014)

[3] Univa Grid Engine, http://www.univa.com/products/

[4] f_node 以外でも X 転送(GUI)が利用できるようになりました(TSUBAME 計算サービス お知らせ), http://www.t3.gsic.titech.ac.jp/node/149

[5] 東京工業大学学術国際情報センター : TSUBAME グランドチャレンジ大規模計算制度、 http://www.gsic.titech.ac.jp/GrandChallenge