

# 安価な自作サーバーによる大規模乱流解析データのポスト処理

## Post-Processing of Large-Scale Turbulent Simulation Data on a cheap DIY server

○ 松山 新吾, JAXA, 東京都調布市深大寺東町 7-44-1, E-mail: smatsu@chofu.jaxa.jp  
Shingo Matsuyama, Japan Aerospace Exploration Agency, 7-44-1 Jindaiji-higashimachi, Chofu, Tokyo

In this paper, we report tips for post-processing large-scale turbulent simulation data on a single cheap DIY server cheaper than one million yen. The memory and storage sizes required for post-processing turbulent simulation data on one billion grid points are discussed for ideal-gas flow and hydrogen combustion simulations as examples. The storage performance for reading huge size turbulent simulation data is evaluated. The visualization of DNS result of a planar turbulent jet are also attempted on commodity GPUs. The results of performance evaluations indicated that the post-processing of turbulent simulation data on one billion grid points can be performed on a single cheap DIY server.

### 1. はじめに

京コンピューターをはじめとしたスーパーコンピューターによる近年の計算速度性能の向上は目覚ましく、最新(2016年6月)のTOP500<sup>(1)</sup>において1位を獲得した中国の神威太湖之光(Sunway TaihuLight)の実測性能は93 Pflopsに到達している。京コンピューターでは1.855兆点(12288<sup>3</sup>点)という超大規模な乱流DNS解析<sup>(2)</sup>が実アプリケーションとして実施されている。このような超大規模解析は計算結果の可視化といったポスト処理も研究対象になるような解析であり、京コンピューターならではの研究である。

著者が在籍している国立研究開発法人宇宙航空研究開発機構(JAXA)では、2016年4月より理論性能3.49 PflopsのJAXAスーパーコンピューターシステム<sup>(3)</sup>(JSS2)を稼働させている。JSS2は3240ノードのFujitsu PRIMEHPC FX100を中核としたシステムであるが、著者の場合は100ノード程度(全ノードの約3%)を利用して1~10億点の非定常乱流解析を日常的に実施している。京コンピューターで実施される超大規模解析では専用のポスト処理環境に頼らざるを得ないが、著者が日常的に行うような計算の場合、自分のデスク上で手軽に可視化が出来るといった、ポスト処理の迅速性も重要な要求の一つではないかと認識している。

本講演ではDNS・LESなどの非定常乱流解析による大規模データ(格子点で1~10億点を想定)を比較的安価な(ここでは100万円以下とする)自作サーバー上でポスト処理するために必要なノウハウを事例とともに紹介したい。1~10億点という解析規模は最先端の超大規模解析と比較するとかなり小規模かもしれないが、これから経験と実績を積もうとする若手研究者にとっては個人レベルで様々な研究の試行が出来、新たなモデルの提案や現象の解明など、論文執筆に値するレベルの研究成果を出すことができる規模であると考えられる。本講演で紹介する事例は、著者が実際に科研費の若手Bや基盤Cなど数百万円の研究予算で実施した研究において蓄積された経験によるものである。同規模の限られた研究予算で研究を行う研究者にとって、本稿の内容が研究を効果的に推進するための参考となれば幸いである。

### 2. 解析例

本稿では非定常乱流解析の事例として、(1)スカラー混合を伴う平面乱流噴流のDNSと(2)水素乱流拡散火炎のLESを挙げる。Fig.1にスカラー混合を伴う平面乱流噴流に対するDNS解析の例を示す。 $Re = 10^4$ の平面乱流噴流について、9次精度の高次精度化を施したSLAUスキーム<sup>(4)</sup>により圧縮性Navier-Stokes方程式を解いている。解析に用いた総格子点数は約2.5億点(78×85×65格子点×572ブロック)である。図の結果はスカラー

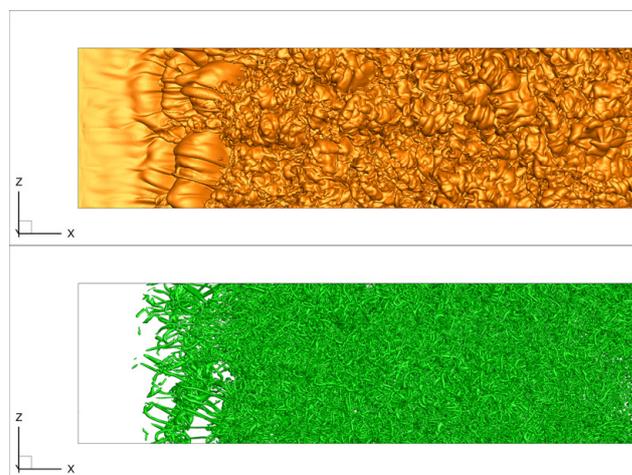


Fig. 1 An example of DNS result of a planar turbulent jet with scalar mixing. Iso-surfaces of mass fraction of passive scalar (upper) and Q-criteria (lower) are shown.

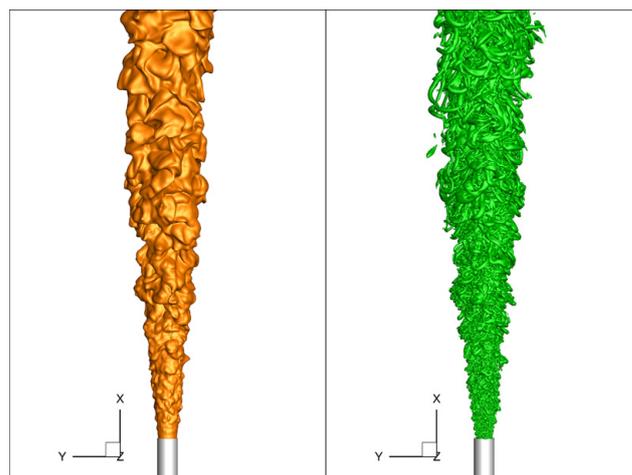


Fig. 2 An example of LES result of a hydrogen turbulent non-premixed flame. Iso-surfaces of mixture fraction (left) and Q-criteria (right) are shown.

の質量分率と速度勾配テンソルの第二不変量の等値面を示す。Fig.2に水素乱流拡散火炎のLES解析の例<sup>(5)</sup>を示す。この問題では $Re = 10^4$ の乱流拡散火炎について、5次精度の高次精度化を施したSLAUスキームにより水素・空気燃焼で生じる9化学種

を考慮した圧縮性 Navier-Stokes 方程式を解いている。解析に用いた総格子点数は約 7000 万点 (41×41×101 格子点 ×414 ブロック) である。図の結果は混合分率と速度勾配テンソルの第二不変量の等値面を示す。

### 3. 単体サーバー上で 10 億点までの乱流解析データのポスト処理を行うために必要な性能の検討

筆者がこれまでに研究に使用してきたポスト処理サーバーのスペックと価格を Table 1 に示す。購入価格は部品価格のみでサーバーの組み立ては筆者が自ら行った。サーバーの基本的な構成を以下に述べる。

まず、ワークステーション用のシングル CPU 用マザーボードに 4 もしくは 6 コアの CPU を搭載している。シングル CPU 用マザーボードのほとんどはメモリスロット数が 4, 6, 8 であるから、スロットを全て埋めるように 2~16GB のモジュールを搭載する。1 枚あたりの容量が大きいメモリはレジスタードメモリになるため、購入価格が大きく跳ね上がる。

ハードディスクを大量に搭載するためにはサーバー用の大型ケースが必要であるが、5 インチベイ 3 段に 5 台の HDD を搭載可能なリムーバブルラックを 3 つ搭載することで最大 15 台のハードディスクを搭載している。HDD 交換などのメンテナンス性を考えるとリムーバブルラックを使用するのが良いが、価格を抑えるために 12 段の HDD 用 3.5 インチベイを内蔵するケースを選択した場合もある。マザーボードに搭載される SATA ポート数は 6~12 しかないため、不足分は RAID 機能の無い SATA ポート拡張カードを使用している。ハードウェア RAID カードを購入して全ての HDD を RAID カードに接続することも考えられるが、購入価格を抑えるために Linux のソフトウェア RAID を使用している。

グラフィックカードはグラフィックワークステーション用の高価なカードではなく、ゲーム用途の数万円程度のグラフィックカードを搭載している。グラフィックカードに搭載されるメモリは可能な限り大容量なモデルが望ましい。

Table 1 に示すように 2009 年からサーバーを構築しているが、おおむね 50 万円以下で 12 台以上の HDD を搭載したサーバーを構築可能である。高価な CPU・グラフィックカード・RAID カードを使用しない限り、購入価格は搭載する HDD の容量・台数とメモリの容量に依存する。最も新しいサーバーではメモリ容量を増やすためレジスタードメモリを利用しており、過去のサーバーと比べて 15 万円程度価格が跳ね上がっている。

#### 3. 1 メモリ容量およびディスク容量

Table 2 に乱流解析データのポスト処理に必要とされるメモリ使用量と出力されるデータ容量を、理想気体を仮定した解析と水素の燃焼解析を事例にして示す。

理想気体を仮定した解析の場合、保存変数は 5 つ ( $\rho, \rho u, \rho v, \rho w, E$ ) であるからすべての保存変数を読み込んだ場合に必要なメモリ容量は格子点 1 億点で 3.72529 GB、10 億点では 37.2529 GB になる。乱流統計量を解析するために最低でも  $10^3$  個程度の時系列データが必要だと仮定すると、データを保存するために必要なディスク領域は 10 億点の場合に 36.3798 TB になる。

水素の燃焼解析の場合、9 化学種の質量分率を含んだ保存変数は 14 ( $\rho, \rho u, \rho v, \rho w, E, \rho Y_s$ ) となるため、10 億点の解析データのポスト処理に必要なメモリ容量は 104.308 GB、時系列データの容量は 101.863 TB になる。

メモリ容量については保存変数以外にも時間平均量などのデータを読み込む場合があるため、Table 2 での見積もり量に対して倍程度のメモリ容量を見込んだ方が無難である。最も新しいサーバー (2016 年 1 月購入) ではメモリサイズが 128 GB、12 台の 5TB

HDD で RAID6 の RAID ディスクを構築するとディスク領域は 50 TB になる。したがって、理想気体を仮定した解析の場合、10 億点の乱流解析データを単体のサーバー上でポスト処理することが十分可能なスペックになっている。水素の燃焼解析の場合、1 億点の解析データは十分にポスト処理が可能であるが、10 億点の場合ではスペックが不足するため、メモリを 256 GB へ拡張する必要が生じる。ただし、必ずしも全計算領域のデータを同時に読み込む必要は無く、マルチブロックコードの場合、ブロックごとにポスト処理を行うことでメモリ使用量を大幅に削減することも可能であることを付け加える。ディスク容量の不足については、時系列データの個数を半分の 500 程度に抑えるか、10 TB の HDD を利用して 100 TB の RAID ディスクを構築する必要がある。10TB の HDD は 5TB HDD に対して倍以上の価格であるため (2016 年 11 月現在)、12 台購入した場合にサーバーの購入価格は 100 万円以内に収まらないものと推測される。

Table 1 Specifications and prices of DIY post-processing servers

購入時期	価格	スペック
2009 年 4 月	38.5 万円	CPU 2.66 GHz (4 core), Memory 12GB, HDD 1.5TB×12, GPU 1792MB
2009 年 6 月	43.0 万円	CPU 2.66 GHz (4 core), Memory 12GB, HDD 1.5TB×15, GPU 1024MB
2009 年 12 月	48.9 万円	CPU 2.93 GHz (4 core), Memory 16GB, HDD 2TB×12, GPU 1792MB
2010 年 8 月	47.2 万円	CPU 3.33 GHz (6 core), Memory 24GB, HDD 2TB×12, GPU 1280MB
2011 年 5 月	47.0 万円	CPU 3.4 GHz (4 core), Memory 16GB, HDD 3TB×15, onboard VGA
2012 年 8 月	47.2 万円	CPU 3.3 GHz (6 core), Memory 32GB, HDD 3TB×15, GPU 512MB
2015 年 12 月	49.6 万円	CPU 3.3 GHz (6 core), Memory 64GB, HDD 5TB×12, GPU 1024MB
2016 年 1 月	66.8 万円	CPU 3.5 GHz (6 core), Memory 128GB, HDD 5TB×12, GPU 4096MB

Table 2 Estimated minimum memory and storage sizes required for ideal-gas flow and hydrogen combustion simulations

理想気体解析 (保存変数 = 5, $\rho, \rho u, \rho v, \rho w, E$ )		
格子点数	メモリ使用量	データ容量 (時系列データ $10^3$ 個)
$10^8$	3.72529 GB	3.63798 TB
$10^9$	37.2529 GB	36.3798 TB
水素燃焼解析 (保存変数 = 14, $\rho, \rho u, \rho v, \rho w, E, \rho Y_s$ )		
格子点数	メモリ使用量	データ容量 (時系列データ $10^3$ 個)
$10^8$	10.4308 GB	10.1863 TB
$10^9$	104.308 GB	101.863 TB

#### 3. 2 ディスク I/O 性能

時系列データを保存するために必要な HDD 容量は 10 億点の解析まで確保できそうであることは確認できた。次に、ディスクの I/O 性能について検討する。これまでの経験上、ポスト処理の結果として大きなサイズのデータを出力することはほとんどないため、ここでは主に読み込み性能について検討をする。

最新の HDD 単体のシーケンシャル読み込みの性能は 100 MB/s 以上に達するが、仮に 100 MB/s の速度で  $10^3$  個の時系列データを読み込んだ場合、1 億点および 10 億点の理想気体解析ではそれぞれ 10.5964 時間と 105.964 時間になり、データを読み込むだけで 半日と 4.5 日の時間を費やすことになる。さらに、水素

燃焼解析の場合では 29.6697 時間と 296.697 時間となり、データ読み込みには 1 日強と 12 日強が必要となる。

ここまでで既に説明したように、著者が構築するサーバーでは Linux のソフトウェア RAID を利用して RAID ディスクを構築している。これまでの経験では利用できるディスク容量 (HDD 2 台分の容量が減るのみ) と信頼性のバランスにおいて RAID6 が使いやすい。RAID6 では冗長性が確保されるだけでなく、データが複数のディスクに分散されて保存されるため、単体 HDD よりも読み込み性能が向上する。仮に 1 GB/s の読み込み性能が実現できるのであれば、10 億点の解析であっても、時系列データの読み込みに必要な時間は理想気体解析で半日、燃焼解析では 1 日強まで短縮されることになる。

以下では Table 1 に示したサーバーを利用して RAID の I/O 性能に影響を与えるパラメーターと環境変数 (チャンクサイズ, read\_ahead\_kb, stripe\_cache\_size) を変化させて性能を評価する。I/O 速度の計測には Linux の hdparm コマンドに加えて自作の Fortran プログラムを使用した。自作の Fortran プログラムではデータサイズを変化させるとともに、単一ファイルを読み込む場合と 10 分割したファイルを読み込む場合とで性能評価を実施した。速度計測には 5TB HDD (MD04ACA500) ×12, 1.5TB HDD (ST31500341AS) ×15, および 2TB HDD (MD04ACA200) ×12 という HDD 構成による RAID6 ディスクを搭載した 3 種類のサーバーを使用した。Table 1 で示した最も新しいサーバーは 5TB HDD ×12 の構成である。OS などの環境は、openSUSE leap 42.1 (kernel 4.1), mdadm v3.3.1, Intel Fortran バージョン 16.0.2.181 である。

まず、Fig. 3 に hdparm コマンドによりディスク読み込み速度を評価した結果を示す。RAID6 のチャンクサイズについて 32 KB から 8192 kB まで変化をさせ、環境変数では read\_ahead\_kb を 32 kB から 131072 kB まで変化させて読み込み速度を評価した。いずれのサーバー構成についても read\_ahead\_kb を大きな値に設定することで読み込み性能が向上する。また、チャンクサイズは小さな値の場合に読み込み性能が高く、256 kB 以下であればほぼ同程度の読み込み性能が実現されている。最大の読み込み速度は 5TB HDD ×12 による 1400 MB/s であった。

Fig. 4~7 に、自作の Fortran プログラムにより単一ファイルのデータ読み込み速度を評価した結果を示す。データサイズは、S (16<sup>3</sup>×256), M (26<sup>3</sup>×256), L (51<sup>3</sup>×256), XL (101<sup>3</sup>×256) と変化をさせて評価を行った。得られた結果は、データサイズが小さい S と M サイズでは性能のばらつきが目立つが、おおむね hdparm による評価結果と同様の傾向が得られる。ここでも read\_ahead\_kb を大きくすることで性能が向上し、256 kB 以下のチャンクサイズで性能が高い。高い読み込み速度が必須である L と XL サイズのデータでは最大で 1200 MB/s の読み込み性能が得られている。

次に、10 のファイルに分割して出力された 2.4 億点 (78×85×129 格子点 ×286 ブロック) のデータに対して読み込み性能の評価を行った。これは、筆者が使用している乱流解析コードの中で使用しているデータ出力形式を模擬したものである。Fig. 8 に読み込み性能の評価結果を示す。10 分割されたファイルを読み込む場合でも、総格子点数がほぼ同じになる XL サイズの単一ファイル読み込み (Fig. 7) と同程度の性能が実現されていることがわかる。最大の読み込み速度は 5TB HDD ×12 のサーバー上で 1100 MB/s に到達する。

最後に、S, M, L, XL サイズのデータを出力する場合の書き込み速度を評価した結果を Fig. 9~12 に示す。書き込みの場合には環境変数 stripe\_cache\_size を 32 kB から 32768 kB まで変化させた。得られた傾向として、データサイズが小さい場合 (S, M) は stripe\_cache\_size が大きいほど書き込み性能が高く、データサイズ

が大きい場合 (L, XL) には stripe\_cache\_size への依存性が小さい。また、チャンクサイズについては値が大きすぎなければ、いずれも同様の性能が得られる。読み込み性能と比べて使用する HDD の種類によって少し書き込み速度のばらつきが見られるが、小さいサイズのファイル出力は元々時間がかからない上に、XL サイズのデータ出力では 1000 MB/s 程度の書き込み性能が得られており、十分に高速な書き込み性能が実現されている。

以上のベンチマークの結果から、RAID6 ディスクのチャンクサイズとしては 256 kB 以下とすることを推奨する。また、環境変数の read\_ahead\_kb と stripe\_cache\_size についてはいずれも 2048 kB 以上の大きな値に設定することを推奨する。ベンチマーク結果が示すように、5TB HDD ×12 の構成による最新のサーバーでは 1 GB/s 以上の読み込み性能が達成されており、10 億点レベルの乱流燃焼解析であっても 10<sup>3</sup> 程度の時系列データを 1 日強で読み込むことが可能である。

Table 3 CPUs and GPUs equipped on tested post-processing servers

	CPU	GPU
(1)	Xeon E5-1650v3 3.50 GHz	GeForce GTX 960 4096 MB
(2)	Core i7-870 2.93 GHz	GeForce GTX 275 1792 MB
(3)	Core i7-920 2.66 GHz	GeForce GT 520 1024MB
(4)	Core i7-920 2.66 GHz	GeForce 210 512 MB
(5)	Core2 Quad Q9550 2.83 GHz	GeForce 8300 GS 256 MB

### 3. 3 グラフィック描画性能

グラフィック性能の評価には少し任意性があるが、ここでは Fig.1 に示した約 2.5 億点の乱流解析データの可視化をスムーズに行うことができるかで評価した。性能評価を行ったサーバーに搭載されている CPU と GPU の構成を Table 3 に示す。GPU だけでなく CPU の構成も示す理由であるが、後述するようにソフトウェアレンダリングも可視化手法として有効なためである。Fig. 1 に示した等値面の可視化は Table 3 のサーバー (1) により Tecplot 360 EX 2016 R1 を使用して実施したものである。NVIDIA が提供する Linux Display Driver のコントロールパネルによる情報を見ると、可視化を行っている最中のグラフィックカード上のメモリ使用量は約 3500 MB (87%) であった。さらに描画する等値面の値などを変化させて再描画などを行ったところメモリ使用量は最大で約 3700 MB (91%) にまで到達した。Fig. 1 に示した結果以上に微細な等値面を描画する場合には、グラフィックカードのメモリ容量が不足する可能性があり、8192 MB のメモリを搭載するモデルを利用することを検討したほうが良さそうである。ただし、サーバー (1) 上で可視化を行った際にはマウス操作に対する可視化ソフトウェアの反応は悪くなく、再描画の際に少し時間が必要であったがフリーズすることも無く、十分実用に足るものと考えられる。また、Tecplot で描画を行っている際には CPU も利用しているようで、主メモリ (マザーボードに搭載されているメモリ) の使用量は最大で 25~30 GB 程度であった。

サーバー (2)~(5) に搭載されるグラフィックカードはサーバー (1) と比較するとかなり貧弱なモデルであり、スムーズな可視化ができないものと想像される。実際に可視化を行ってみると、再描画の際にグラフィックカードのメモリ使用量が 100% を超えてしまい、主メモリも不足してスワップが発生しフリーズに近い状態に陥ってしまった。残念ながら、2.5 億点の乱流解析データを可視化するためには最低でも 4096 MB のメモリを搭載したグラフィックカードが必須なようである。2016 年 11 月の時点で販売されているグラフィックカードでは NVIDIA GTX 1070 を搭載したモデルが 8192 MB のメモリを搭載しており、価格も 5,

6 万円で購入可能なようである。さらに大規模な 10 億点レベルのデータを可視化するためには 8192 MB のメモリを搭載したグラフィックカードの使用を検討すべきだろう。

最後にソフトウェアレンダリングを利用した可視化についても触れておきたい。筆者が良く利用する Tecplot の機能の一つとして、マクロとソフトウェアレンダリングを利用した完全なバックグラウンド実行による描画機能がある。これは、非定常解析結果の動画を作成するために良く利用している機能であるが、レイアウトを定義するレイアウトファイルを作成すれば、あとはそのレイアウトにあわせて時系列データを連番で読み込んで描画した結果を出力していくものである。この機能の面白いところは、マクロを利用してバックグラウンドで描画を行う場合、完全に CPU によるソフトウェアレンダリングのみで描画が行われている点である。そのため、レイアウトファイルを作成するために一台だけ高機能のグラフィックカードを搭載したサーバーを用意しておけば、後は貧弱なグラフィックカードしか搭載していないサーバー上でも完全にソフトウェアレンダリングのみで動画を作成することも可能である。また、粗い格子解像度の解析データを利用して可視化を行ってレイアウトファイルを作成し、それを高解像度のデータに利用することもできるだろう。実際に Table 3 のサーバー (1) で作成したレイアウトファイルを利用して、サーバー (2) ~ (5) 上で 2.5 億点の乱流解析の時系列データ 100 個を連番で読み込んで動画を作成してみた。Fig. 13 にソフトウェアレンダリングによる描画中の主メモリ使用量を示す。いずれのサーバーにおいても最大メモリ使用量は 12 GB 以下であり、主メモリの搭載量が十分であればグラフィックカードが貧弱であっても 2.5 億点データの可視化が可能である。描画作業に必要とした時間はほぼ CPU の性能に比例しており、グラフィックカードの機能は全く利用していないようである。

ここではグラフィックカードの描画性能について検討を行ったが、マウス操作に対する反応という主観的な評価に頼らざるを得なかった。今後、マクロを利用した定型パターンの描画処理にかかる時間といったような、客観的な評価ができるベンチマークも検討したい。今回の性能評価の結果からは、2.5 億点の乱流解析データの可視化は 4096 MB 以上のメモリを搭載したグラフィックカードを利用すれば十分可能である。より大規模な可視化データについてはさらに大容量のメモリを搭載したグラフィックカードの利用が必要であろう。また、レイアウトファイルを用意することができれば、そのレイアウトにしたがってマクロとソフトウェアレンダリングを利用した描画が可能である。主メモリの搭載量が十分であれば 10 億点レベルの可視化データを描画することも可能であると思われる。

#### 4. まとめ

本稿では 1~10 億点規模の乱流解析データを 100 万円以下の自作サーバー単体でポスト処理するために必要なノウハウを紹介した。必要なメモリ容量・ディスク領域の検討結果、ディスク I/O 性能の評価結果、また、グラフィック描画性能の評価結果から 1~10 億点規模の乱流解析データを安価な単体のサーバーでポスト処理することは十分に可能であるといえる。

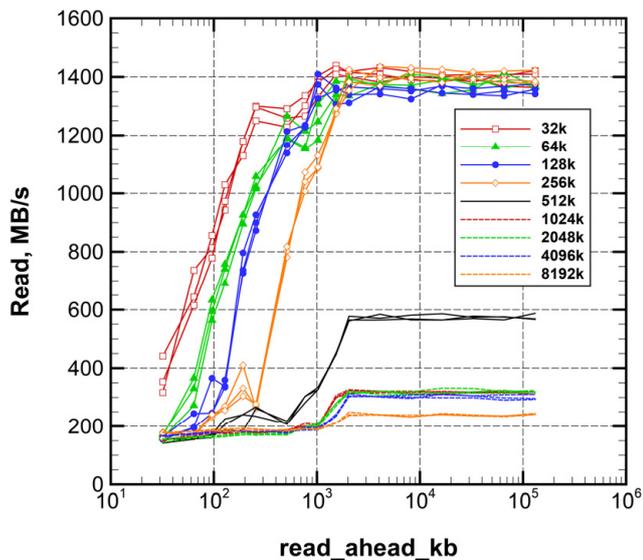
#### 謝辞

本研究は JSPS 科研費 JP15K05817 の助成を受けたものです。

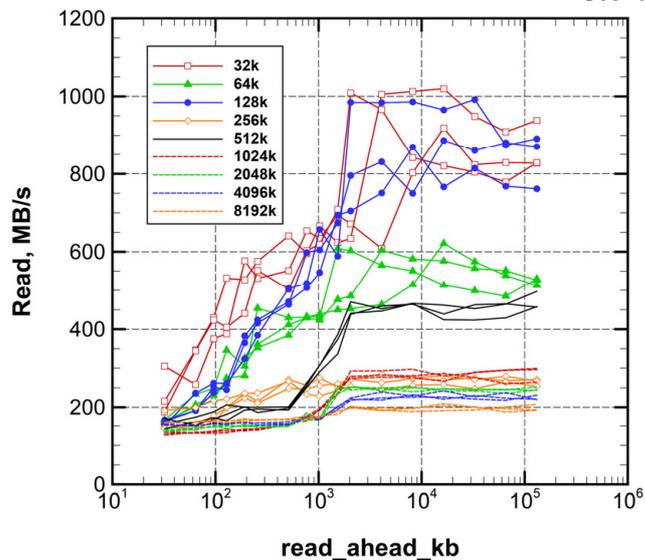
#### 参考文献

- (1) <https://www.top500.org/lists/2016/06>
- (2) <https://www.hpci-office.jp/output/hp150174/outcome.pdf>
- (3) <https://www.jss.jaxa.jp/>

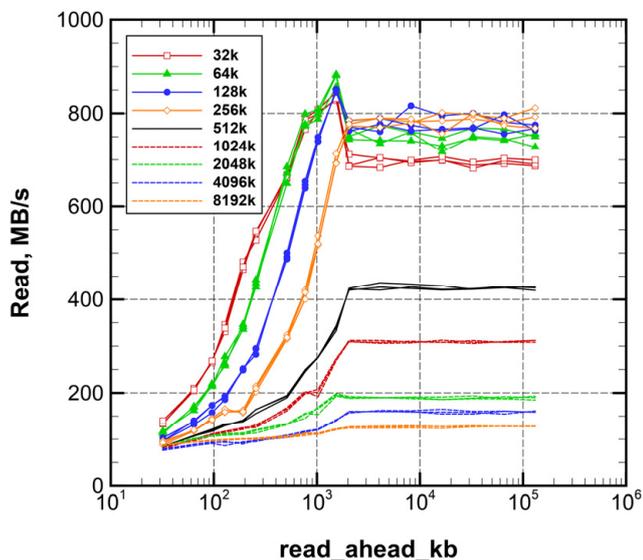
- (4) Shima, E., and Kitamura, K., AIAA Journal 49: 1693-1709 (2011).
- (5) 松山, “水素乱流拡散火炎の Implicit Large-Eddy Simulation”, 第 54 回燃焼シンポジウム講演論文集, C-121, 2016.
- (6) Kang, H. M., Kim, K. H., and Lee, D. H., J. Comput. Phys. 229: 7102-7128 (2010).



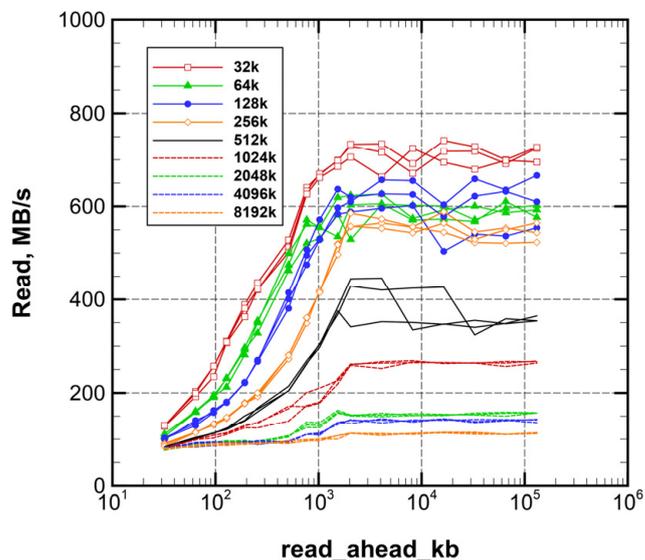
(a) RAID6 by MD04ACA500×12



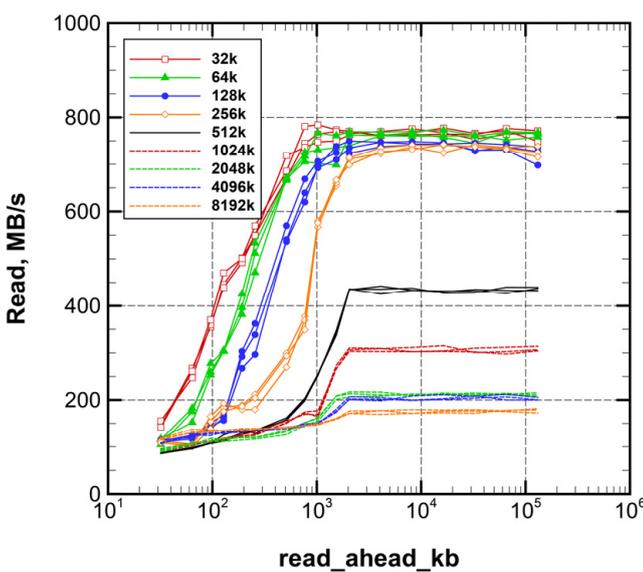
(a) RAID6 by MD04ACA500×12



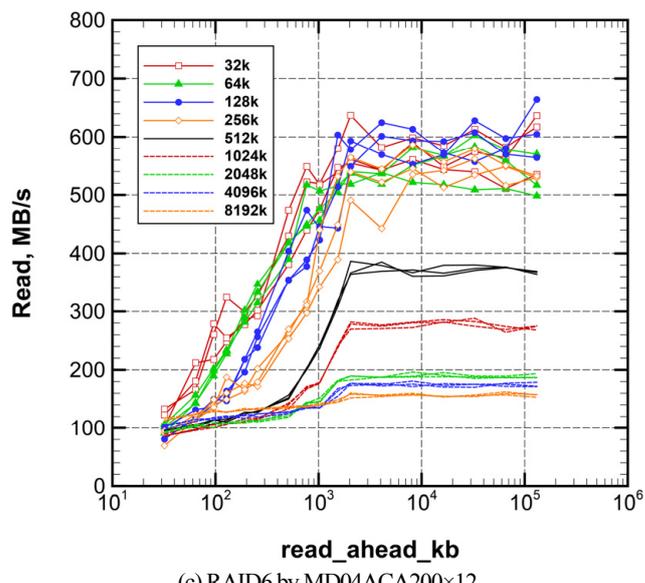
(b) RAID6 by ST31500341AS×15



(b) RAID6 by ST31500341AS×15



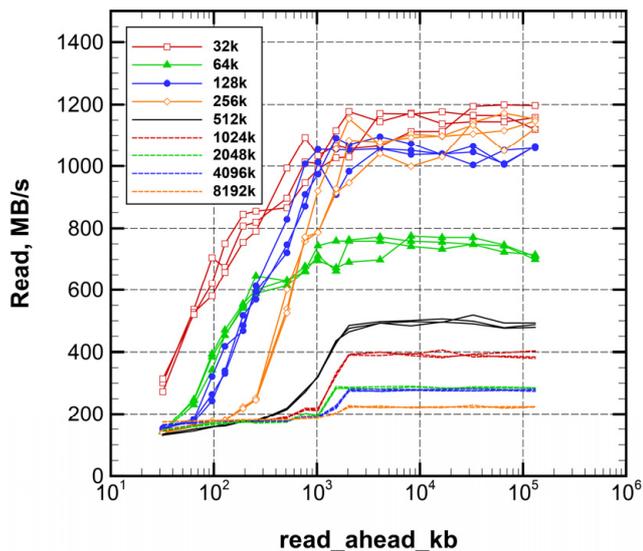
(c) RAID6 by MD04ACA200×12



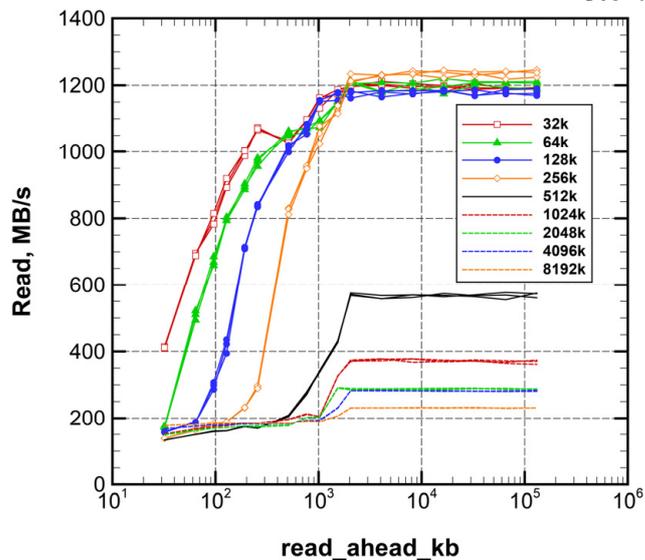
(c) RAID6 by MD04ACA200×12

Fig. 3 Read performance by hdpam

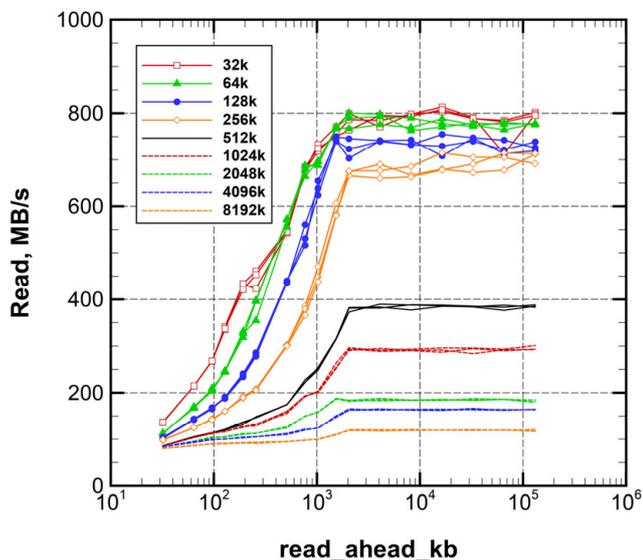
Fig. 4 Read performance by Fortran program (S size)



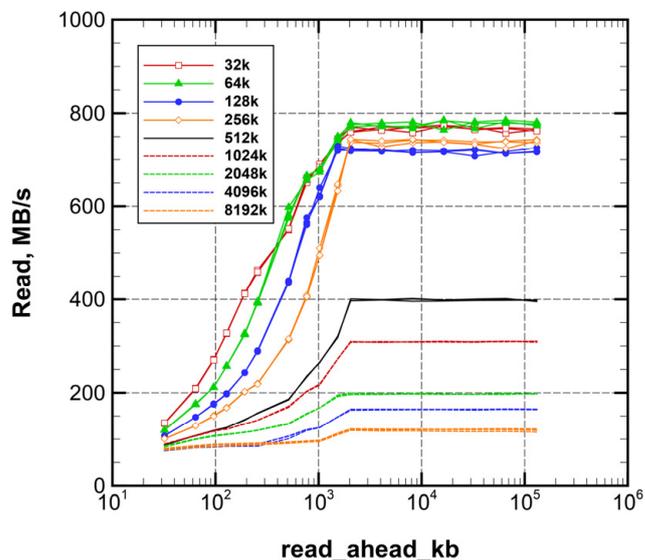
(a) RAID6 by MD04ACA500×12



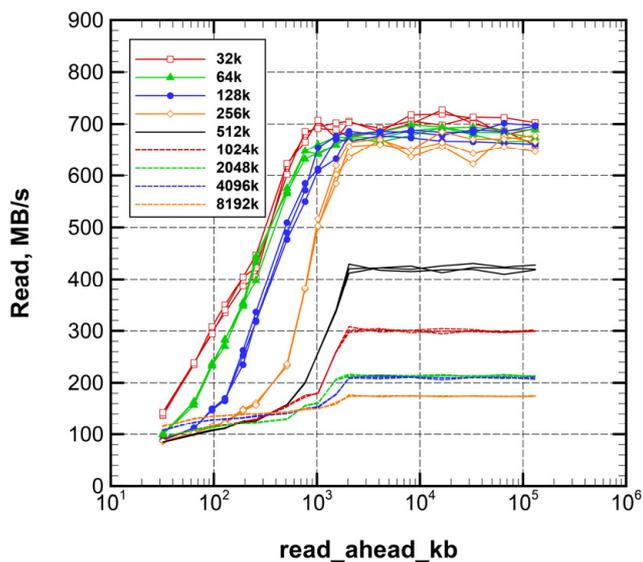
(a) RAID6 by MD04ACA500×12



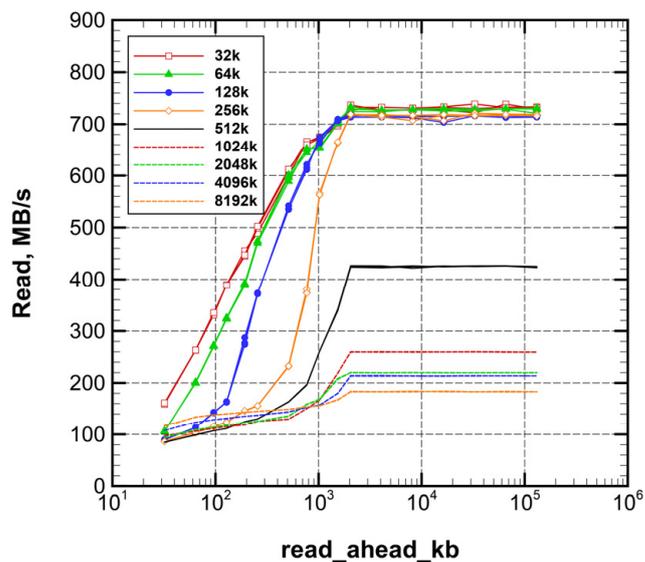
(b) RAID6 by ST31500341AS×15



(b) RAID6 by ST31500341AS×15



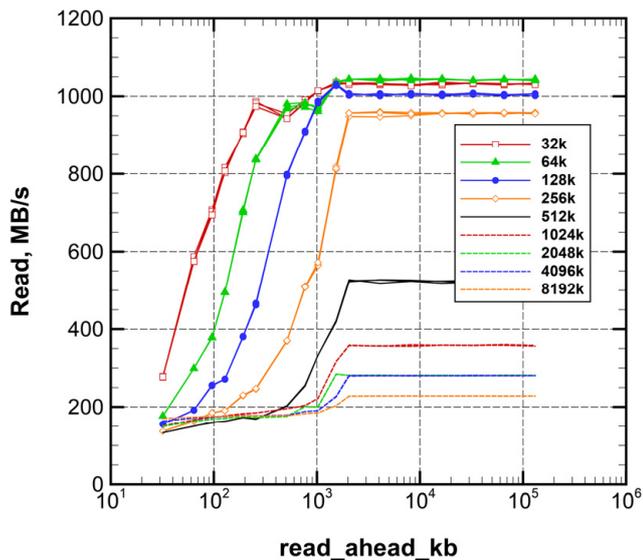
(c) RAID6 by MD04ACA200×12



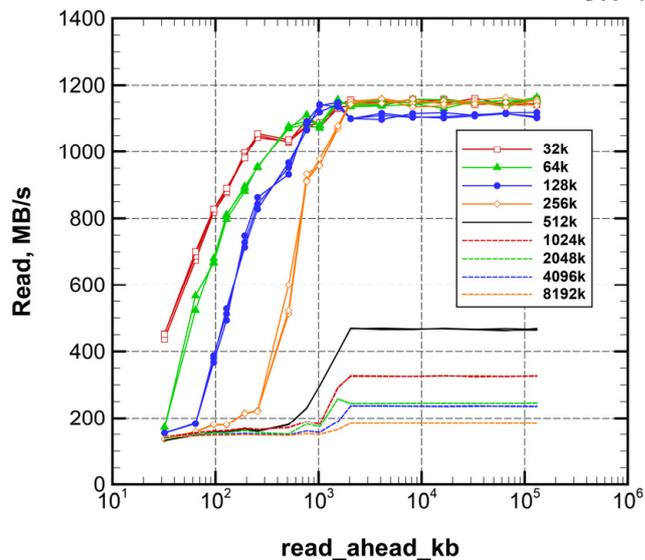
(c) RAID6 by MD04ACA200×12

Fig. 5 Read performance by Fortran program (M size)

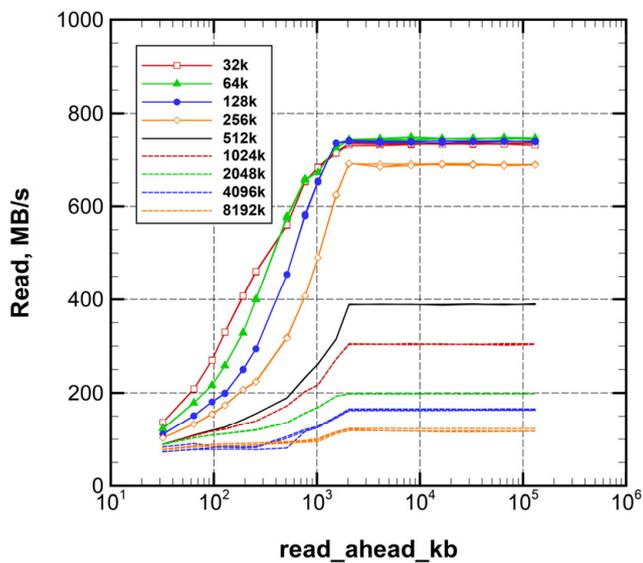
Fig. 6 Read performance by Fortran program (L size)



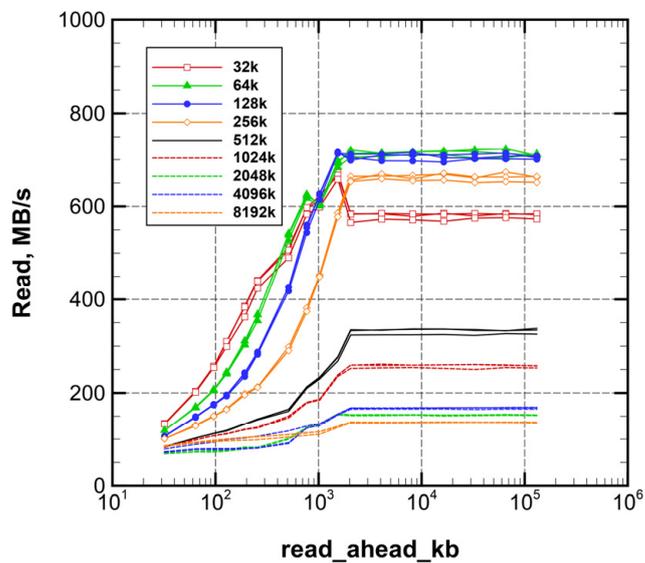
(a) RAID6 by MD04ACA500×12



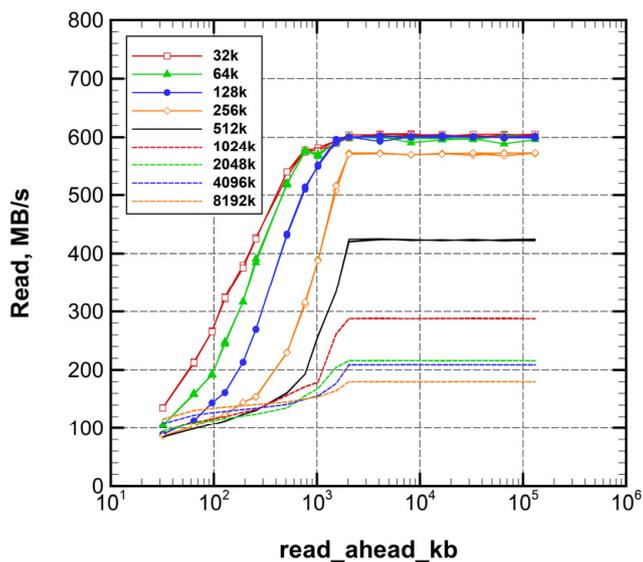
(a) RAID6 by MD04ACA500×12



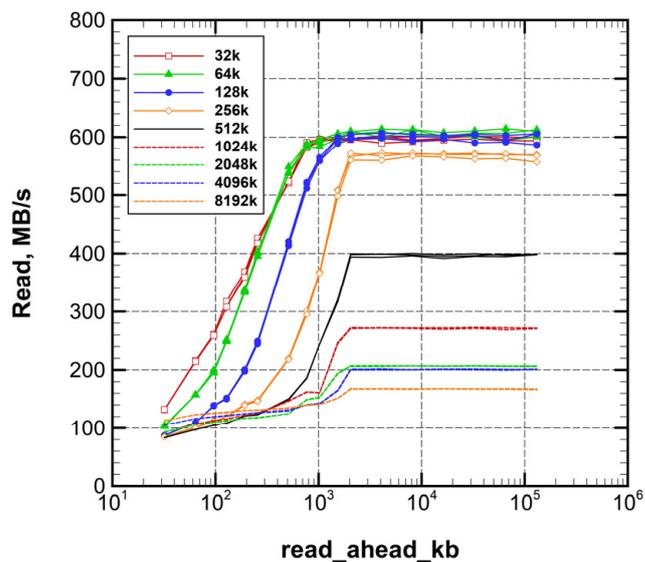
(b) RAID6 by ST31500341AS×15



(b) RAID6 by ST31500341AS×15



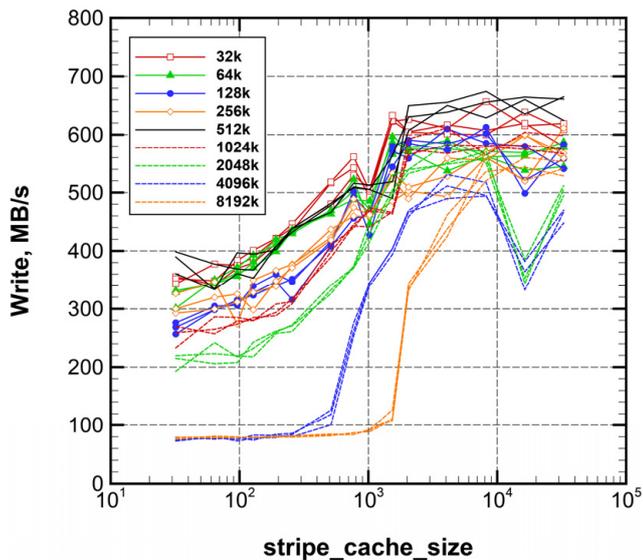
(c) RAID6 by MD04ACA200×12



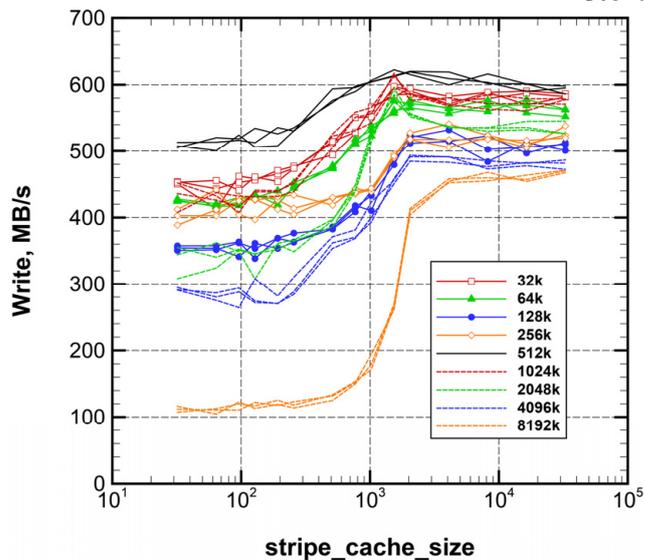
(c) RAID6 by MD04ACA200×12

Fig. 7 Read performance by Fortran program (XL size)

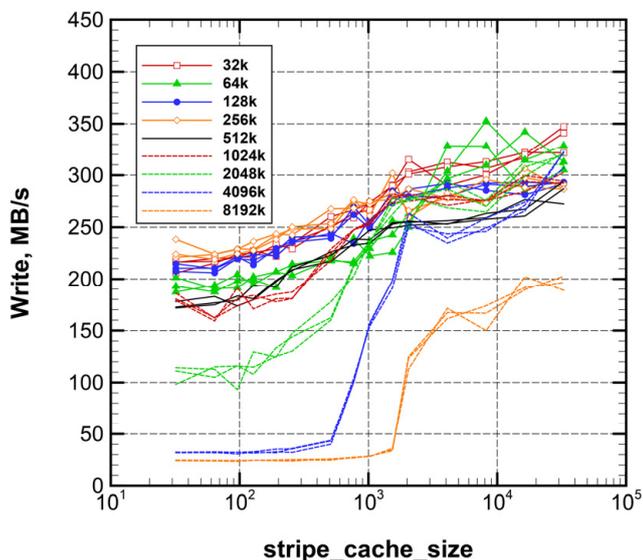
Fig. 8 Read performance by Fortran program (multiple data file)



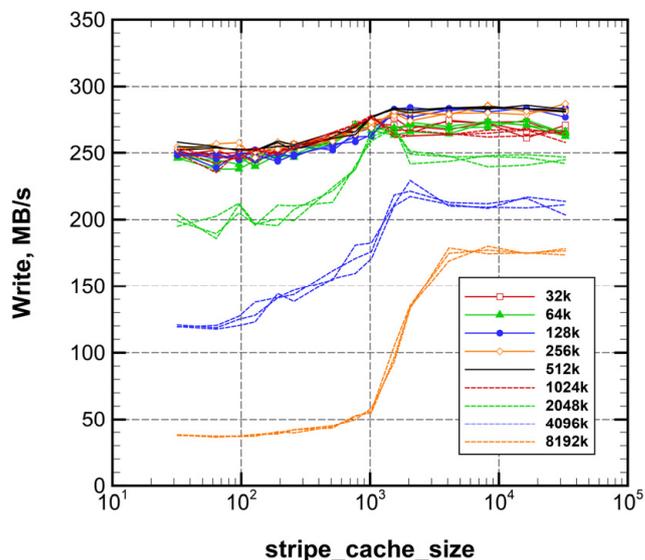
(a) RAID6 by MD04ACA500×12



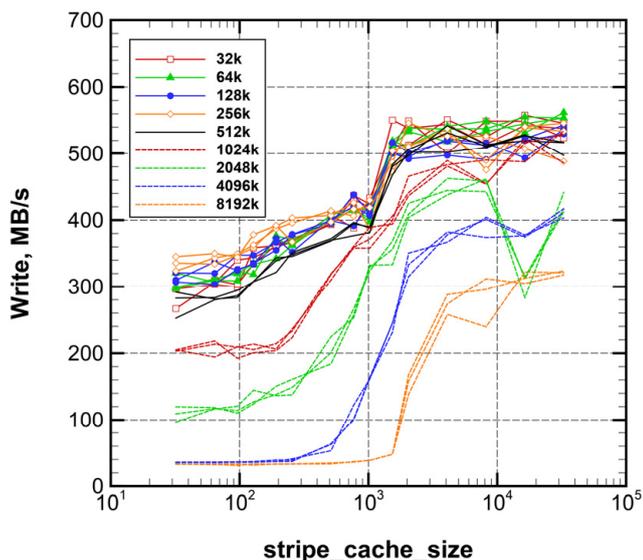
(a) RAID6 by MD04ACA500×12



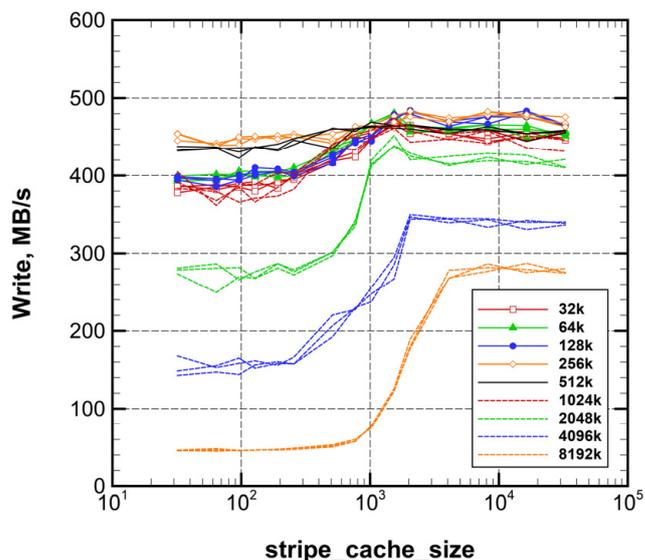
(b) RAID6 by ST31500341AS×15



(b) RAID6 by ST31500341AS×15



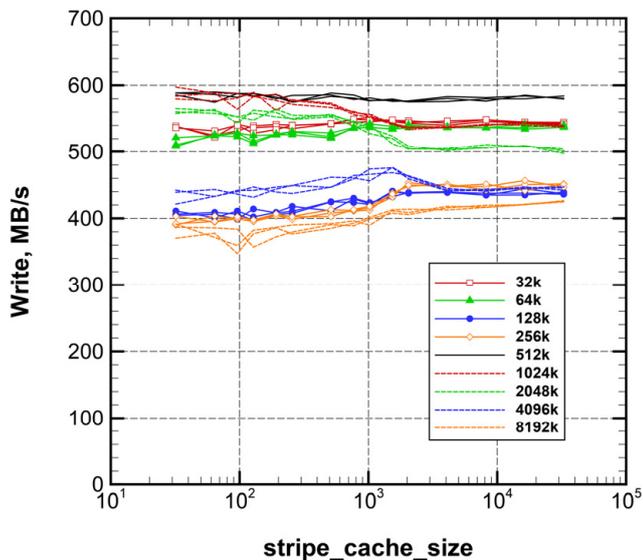
(c) RAID6 by MD04ACA200×12



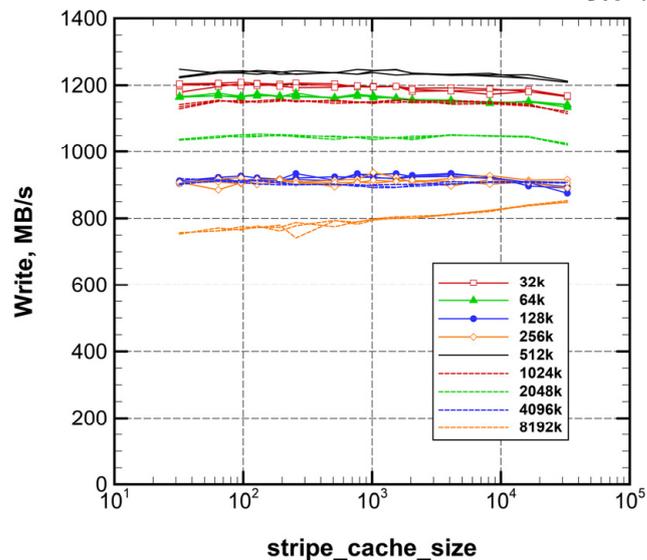
(c) RAID6 by MD04ACA200×12

Fig. 9 Write performance by Fortran program (S size)

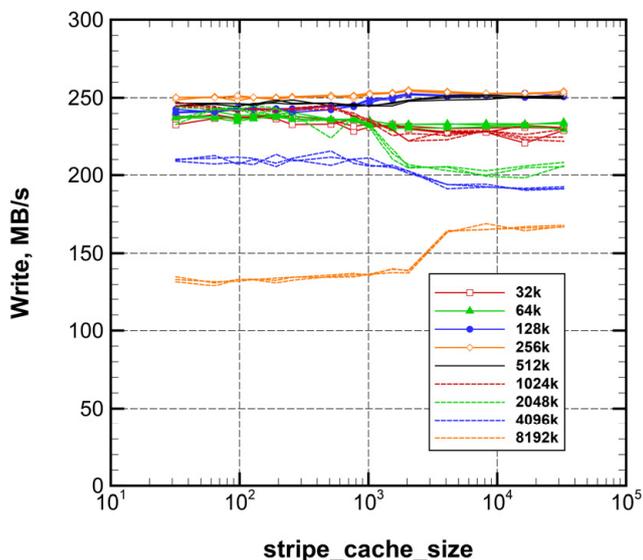
Fig. 10 Write performance by Fortran program (M size)



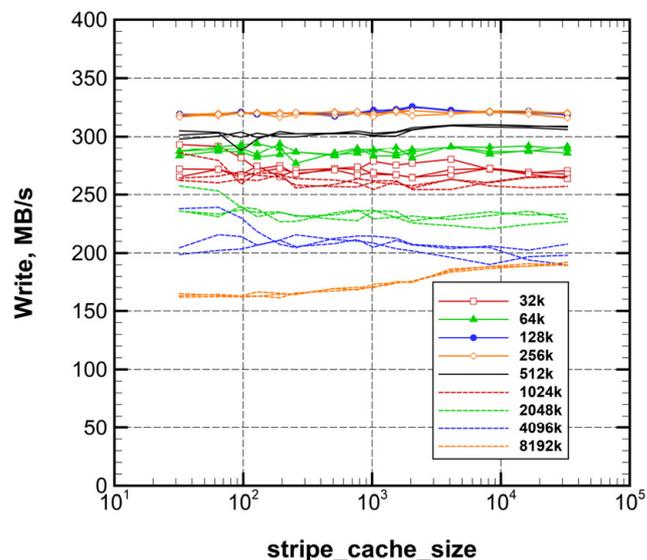
(a) RAID6 by MD04ACA500×12



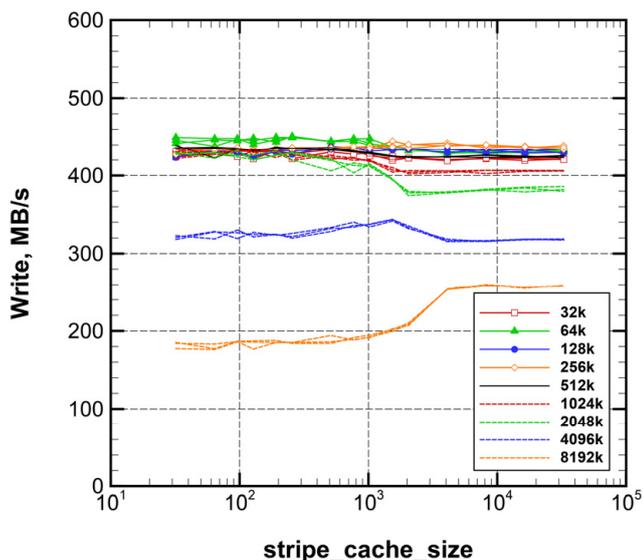
(a) RAID6 by MD04ACA500×12



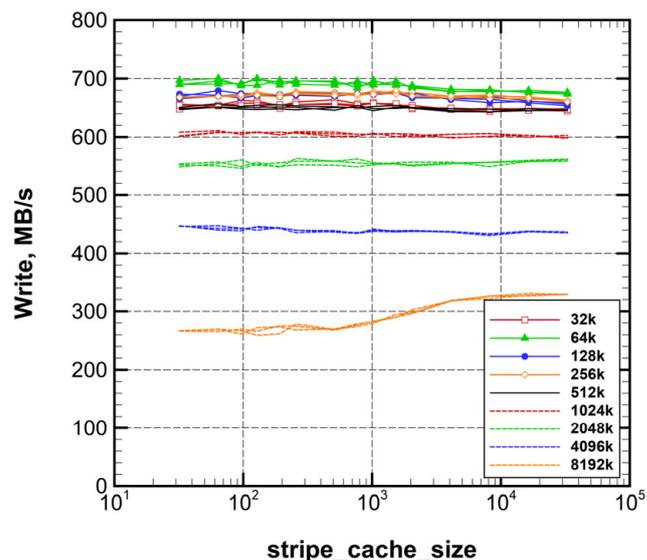
(b) RAID6 by ST31500341AS×15



(b) RAID6 by ST31500341AS×15



(c) RAID6 by MD04ACA200×12



(c) RAID6 by MD04ACA200×12

Fig. 11 Write performance by Fortran program (L size)

Fig. 12 Write performance by Fortran program (XL size)

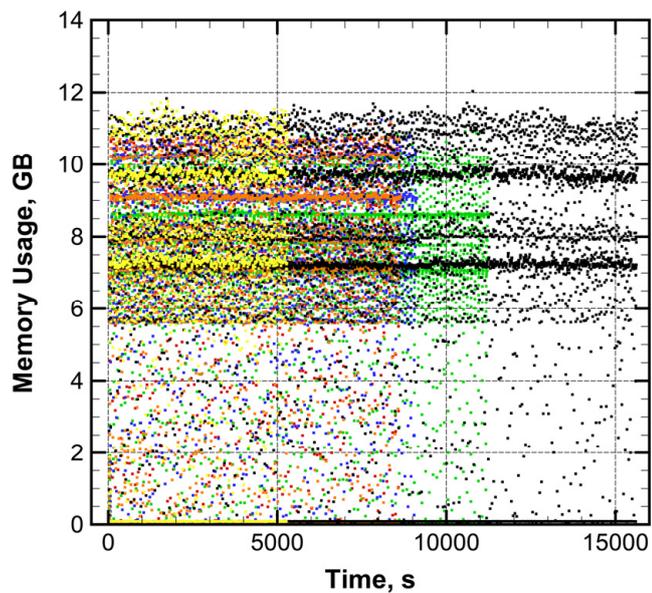


Fig. 13 Memory usages during the visualization by software rendering.