利用者へのストレージ性能情報公開の取り組み

中川 剛史 [1)], 辻田 祐一 [2)], 宇野 篤也 [2)], 板倉 憲一 [1)]

1) 海洋研究開発機構 地球情報基盤センター
2) 理化学研究所 計算科学研究センター

tnakagawa@jamstec.go.jp

Activity to provide I/O information of supercomputer for users

Tsuyoshi Nakagawa [1)], Yuichi Tsujita [2)], Atsuya Uno [2)], Kenichi Itakura [1)]

1) Center for Earth Information Science and Technology (CEIST),
Japan Agency for Marine-Earth Science and Technology (JAMSTEC)
2) RIKEN Center for Computational Science (R-CCS).

概要

シミュレーションのみならず、データ駆動型サイエンスへの対応が運用側に求められる中で、スーパーコンピュータのストレージの役割がますます重要になっている。ストレージの I/O 性能やファイルシステムの特性、効果的な利用方法などの情報公開は、ユーザの利便性向上だけなく、新たな科学的知見およびイノベーションを生み出す可能性がある。海洋研究開発機構における I/O 情報の提供に関する取り組みを紹介する。

1 はじめに

海洋研究開発機構（以下、機構）は、地球シミュレータ[1]（以下、ES）や大型計算機システムなどのスーパーコンピュータ（以下、スパコン）および大規模ストレージを設置し、相互に連携させながら運用している（図 1）。機構ユーザのスパコンに対するニーズは年々多様化してきており、従来のシミュレーション用途以外にも、データ解析（ML・DL も含む）・可視化など多岐に渡り、データそのものだけではなく、データハンドリングの最適化やデータ保存管理に対しても重要性は広がっている。

スパコンに付随するストレージシステムに関しては、システム毎にハードウェア構成や並列ファイルシステムの種類が異なっていることも多く、機器更新時に特性が大きく変化することも多い。しかし、演算処理や MPI 通信の最適化、コンパイラや言語仕様などのトピックに比べて、ユーザへの I/O 情報提供については、十分なされていない傾向があった。たとえば、シミュレーションの場合では、初期条件の読み込みやリスタートファイルの作成以外での I/O 時間は、計算メインルーチン部分と比較して短時間である場合が多く、全体の計算時間に与える影響が限定的であることも理由の一つである。また、ユーザが I/O 部分のチューニングを実施したくても、I/O プロファイリングツールや並列ファイルシステムの特性に関しての情報が少なく、理解が不十分となっているだけでなく、誤った使われ方をしている場合もある。

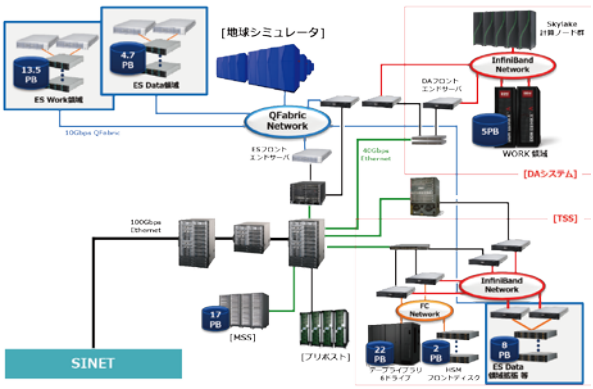

図 1 ：JAMSTEC 計算機システム環境

本稿では、ユーザへのスパコンストレージ I/O 情報の提供および利便性向上を目的とした、I/O プロファイル取得環境整備と I/O モニタリング環境整備の取り組みを紹介する。そして、スパコンのファイルシステム特性と I/O パフォーマンス情報のスパコンセンター間共有について、理化学研究所 計算科学研究センターと実施したので、こちらも併せて紹介する。

2 I/O プロファイル分析への取り組み

機構所内向け大型計算機システムの更新に伴

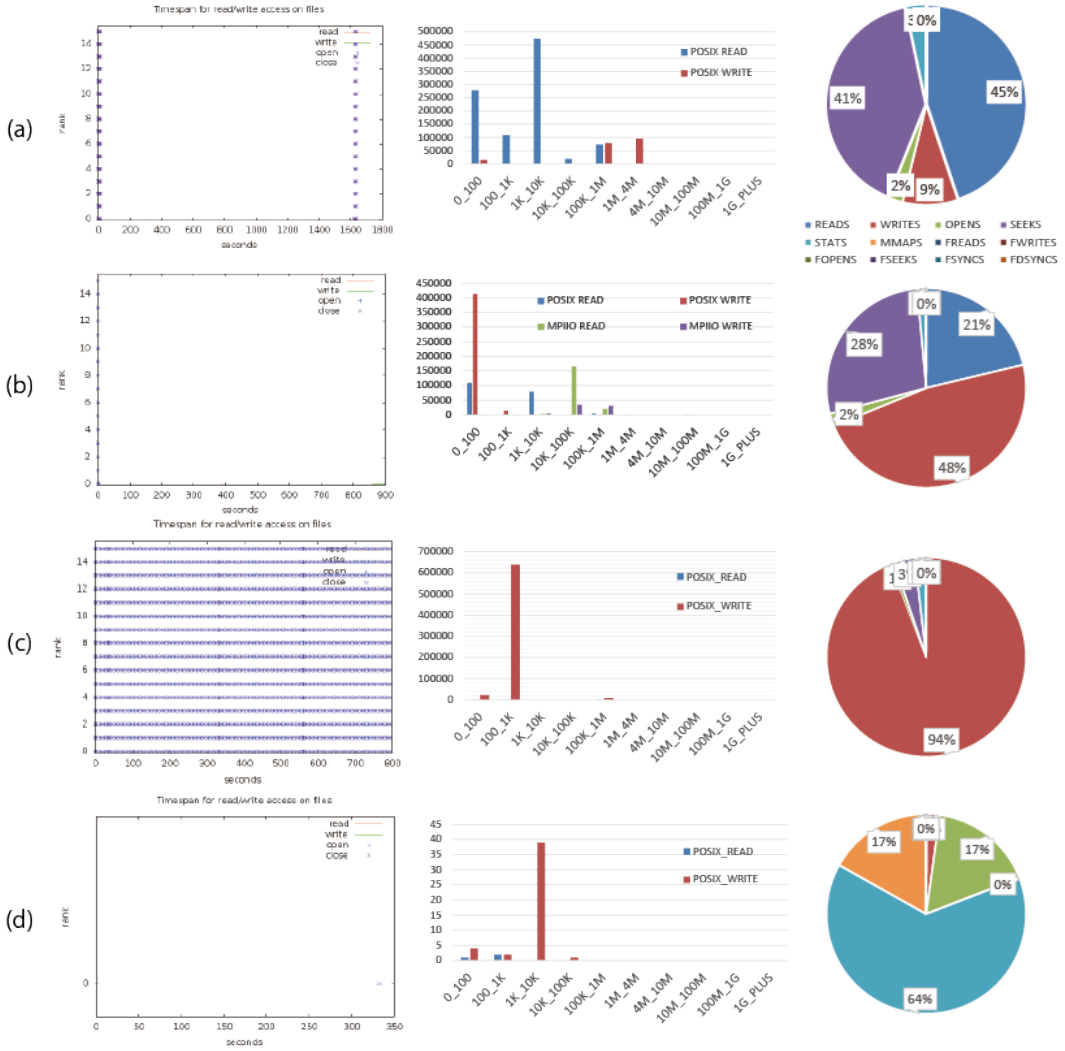

図 2:アプリケーションの I/O プロファイルデータ. (a)領域大気モデル (b)海洋大循環モデル (c)津波浸水シミュレーションコード (d)ゲノム相同性解析コード.
左図：プロセス毎のタイムヒストリ. 中図：POSIX および MPI-IO による IO サイズ分布.
右図：メタデータオペレーション種別分布

う要求要件およびベンチマークセット作成の際に、将来的なワークロードとしての参考情報とするために、複数の機構アプリケーションに対して I/O プロファイルの取得・分析を行った。

本取り組みでは、オープンソースで導入しやすいアルゴンヌ国立研究所が開発した Darshan[2]の内、ver.2.3.0 をベースに理化学研究所 計算科学研究センターが、非 MPI アプリケーションおよび I/O パターン（タイムヒストリ）に対応可能なように機能強化を施した Darshan-riken [3]を利用した。スパコンで良く用いられている I/O プロファイラとしては、ベンダーが用意している専用のツールやコンパイラ付属の機能（ES の場合、FILEINF オプション）に加えて、商用では Intel 社 Vtune Amplifer でも可能であるが、darshan は、環境変数 LD_PRELOAD のみで関数をフックでき、アプリケーションのリコンパイルも不要で容易に利用でき、且つオーバヘッドも小さい。

I/O プロファイリング例として、領域大気モデル・海洋大循環モデル・津波シミュレーションコード、ゲノム相同性解析コードの 4 つのプロファイルを紹介する（図 2）。現実的な I/O ワークロードを反映させるために、それぞれのモデル設定およびインプットデータは、スパコンでの実行時と同様のものを利用している。I/O プロファイルでのアプリケーション毎の分析値は、表 1 に示す。

	(a)領域大気モデル	(b)海洋大循環モデル	(c)津波浸水コード	(d)ゲノム相動性コード
#Running Process	1024MPI	1024MPI	16MPI16SMP	4SMP
Total I/O Size	320 GB	50 GB	5 GB	300 KB
# I/O File	15000	1000	4000	400
% I/O Time / Calc Time	1 %	5 %	1 %	0.1 %
I/O Speed	38GB/s	1GB/s	0.6GB/s	4MB/s
# Meta Operation	2 Mop	0.9 Mop	0.7 Mop	2 Kop
% Meta Time / I/O Time	46 %	21 %	14 %	99 %

表 1：I/O プロファイルデータ

2.1 領域大気モデル（a）および海洋大循環モデル (b)

どちらのモデルも計算アルゴリズムや解像度が異なるものの、ストレージに与えるファイル I/O の観点では、ほぼ同じ特性である。つまり、実行の初めに初期モデル設定のために大容量のリスタートファイルを読み込み、ある一定の計算ステップ毎にモデルパラメタのバイナリアウトプットの出力を実施し、実行の最後に大容量のリスタートファイルを書き出す I/O パターンとなっている。よって、瞬間的に高い I/O スループット性能を必要とする。また、海洋大循環モデルの方は、MPI-IO を採用しており、MPI のマスターランクのみがファイル I/O していることもプロファイラから判別できる。なお、I/O 量の割合としては、インプットデータ＝アウトプットデータとなっている。また、アンサンブル計算の場合を除いて、ジョブはほぼシーケンシャルに実行されるため、複数のジョブが同時に実行される必要性は小さい。

2.2 津波浸水シミュレーションコード

本コードは領域モデルであるため高解像度ではあるが、インプットデータ量は比較的小さい。そして、動画作成用アウトプットファイルが一定間隔で頻繁に作成され、その総量はインプットデータ量の数十倍になるが、個々のファイルサイズは小さく、さほど大きな I/O スループットは必要としていない。ただし、地理的に限定された領域モデル毎にシミュレーションが実行されるため、全体的な結果を得るためには、同時に数 100 の地域に関して同等のシミュレーションの実行が必要とされる場合があり、そのワークロードを考慮する必要がある。

2.3 ゲノム相同性解析コード

本コードは、問い合わせ DNA 配列に類似した配列をデータベース中から検索するツールである。特徴としては、I/O 時間におけるメタデータ時間割合がかなり高いことである。また、主なアウトプットファイルは、テキストファイルが 1 つだけであるので、I/O スループットはほとんど必要としない。

しかしながら、本アプリケーションは実行のされ方に特徴がある。大規模データ実行の際には、本来のインプットデータとデータベースが細分化され網羅的に総当たりで実行されることが必要となるため、同時並行的に単位時間あたりのメタデータオペレーション数がかなり高くなることが推測される。プロファイルでは、1 実行あたり約 60ops と見積もることができるので、システム全体で 10000 ジョブを処理する場合には、最大で 60 万 ops を処理することを想定する必要がある。

2.4 考察

今回例示した 4 つのアプリケーションが必要とする I/O 時間は、実計算時間においては 0.1～5%の範囲にまとまっているものの、I/O 量・サイズ・総ファイル数・メタデータオペレーションの種類と総数にも違いがあることが I/O プロファイルから判明した。ストレージシステム全体でこれらの I/O をカバーするには、I/O スループット能力の強化のみならず、特定のアプリケーションに対してのメタデータ性能へ焦点を絞ったシステム要件や運用方法が必要となる。

また、アプリケーション側で高速化のためのワークロード分割が実施されている場合、1 つの結果を出すために必要とするジョブ数はそれぞれ異なる。よって、単独の I/O プロファイルだけでなく、総合した解釈が必要となる。将来的には、実運用時を想定したワークロードセット作成を目的とし、より多くのアプリケーションに対して、I/O

プロファイルを取得していく必要がある。

3 I/O モニタリングの整備

2 項で述べた機構所内向け大型計算機の更新の際に、ファイルシステム側で I/O 情報を取得できるように、モニタリングツールを採用した。本システム（DA システム；HPE 製 Apollo6500 および Apollo2000 から構成）のストレージは、Data Direct Networks 社の EXAScaler が採用されると共に、Web ベースの I/O 情報モニタリングツール（DDN Lustre Mon）が組み込まれている（図 3）。

DDN Lustre Mon は、オープンソース（Collectd, Graphite, Grafana）をベースとした Web ベースのモニタリングツールであり、Lustre ユーティリティコマンド(lctl)[4]のアウトプットを収集することで、ストレージの利用状況・負荷状況や障害原因特定への利用のみならず、ユーザ毎・グループ毎・計算ノード毎・リクエスト毎に、inode 数や使用容量、I/O サイズ・I/O 種別・I/O 量・メタデータオペレーション数など、アプリケーションの多彩な I/O 情報をデータベース化し分析することが可能である。

旧システムでの I/O モニタリングとしては、sar を用いた I/O サーバおよびネットワークの負荷状況を、障害対応用に利用することに加えて、軽量の I/O ベンチマークセットで定常的に負荷を監視していたのみであり、実際のノード単位・システム単位の I/O スループットやメタデータオペレーション数などの情報は、モニタリングすることができていなかった。よって、I/O に関する問い合わせがユーザサポートに来た場合、調査に時間がかかるだけでなく、ユーザにも追実験をお願いする場合があった。そこで、ジョブの統計情報として取得されている CPU 時間や使用メモリ量、そして FLOPS の様なより詳細なパフォーマンス情報と同様に、I/O 情報に関しても取得及び統計情報への追加することを目標とした。

図 4 はストレージ全体に対する 1 週間の I/O 情報をモニタリングした一例である。特定のジョブが実行されると瞬間的にスループットに負荷を与えている一方で、解析的ジョブが大量実行されている期間は、メタデータオペレーションに台形のピークや瞬間的な山が散見され、一目で混雑状態が判別できている。

現在は、ユーザサポートに対する個別の問い合わせの原因切り分けのみに利用しているが、長期

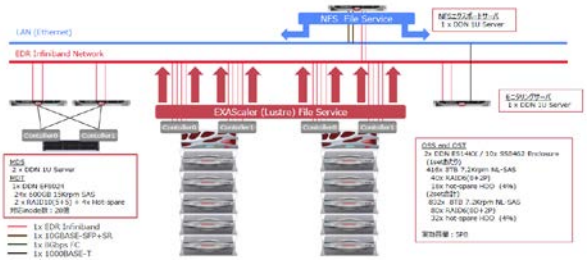

図 3：DA システムストレージ構成図

図 4：Lustre モニタリングシステムの web 画面の一例. (上)I/O スループット. (下)メタデータオペレーション数.

間の I/O 分析からの情報開示についても今後検討している。また、本モニタリングで取得した I/O スループット最大値・メタデータオペレーション数を、ジョブ実行ノードおよび日時を用いて、ジョブ統計情報に結び付けると共に、ユーザが直接アクセス可能な I/O 情報に変換する仕組みを現在検討中である。

その際に I/O モニタリングサーバ側で問題になっていることも数点あり、たとえば、現在のシステムバッチキュー設定では、最大 3 日の長時間ジョブ実行が可能であるが、I/O モニタリングサーバのデータベース部分で想定以上のメモリを使用しており、挙動が不安定な影響でユーザ単位の満足いくデータが取得できていない。また、スループットピーク値に関しては、I/O 情報のサンプリング周期をかなり短くしないと正確な値の取得が難しいが、それに関しても I/O 情報取得の仕組みに依存するため、データベースによるメモリ使用量が増大してしまう。これらの問題点については、現在対処にあたっており、引き続きベンダーと共に最適化を施す予定である。

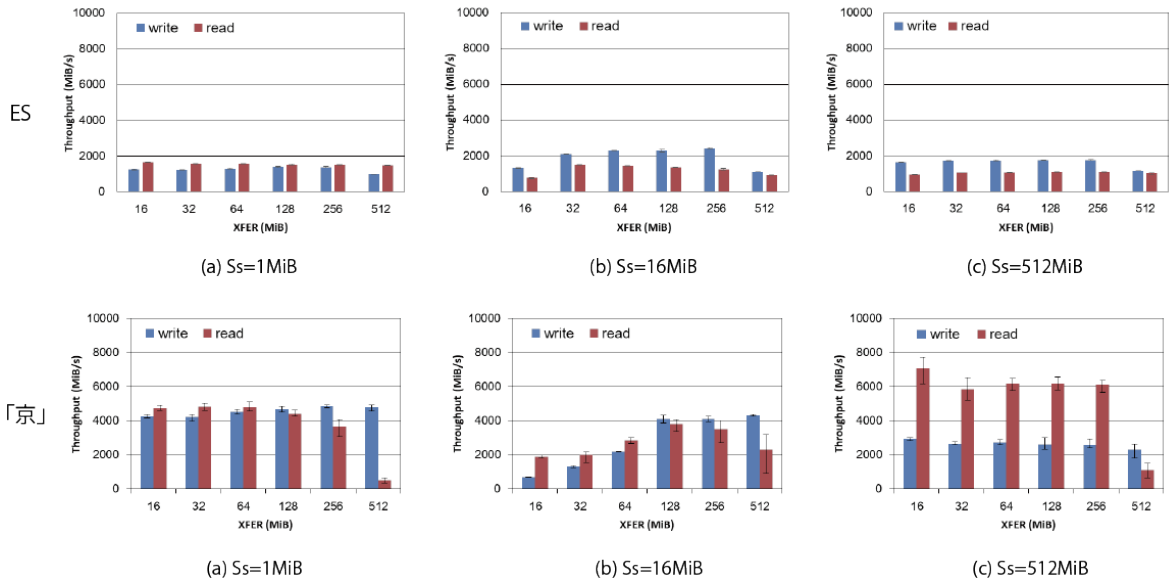

図 5 : IOR ベンチマーク MPI-IO 性能の比較. 上側: ES の 96 ノードに 384 プロセス起動. 下側:「京」の 96 ノード（形状：4×3×8）に 384 プロセス起動.

4 ファイルシステム特性のスパコンセンター間共有とユーザへの情報公開について

複数スパコンサイトのシステムを利用し研究を進めているユーザも多く、例えば、気象分野の課題に関しては、スーパーコンピュータ「京」(以下、「京」) および ES にて、解像度の異なるシミュレーション実験を実施している場合がある。その場合に、それぞれのストレージ構成および並列ファイルシステム特性の差異に起因したアプリケーションの移植性、並びに、ユーザで誤ったパラメタの設定が見受けられる場合があり、そのユーザサポートにはある程度の期間が必要となっている。

そこでユーザに資する情報提供を行う枠組み作りを目標とし、異なるスパコンの並列ファイルシステムの性能比較や挙動をスパコンセンター間で共有する取り組みとして、「京」と ES において、各々の並列ファイルシステムにおける共通 I/O ベンチマーク性能評価を中心とした情報共有を行った。具体的な項目は以下のとおりである。

- ストレージ構成の情報共有
- ファイルシステム詳細の情報共有
- POSIX-IO の性能評価
- MPI-IO 実装方法およびパラメタの情報共有、Two-Phase I/O の実装

図 5 は、この取り組みの MPI-IO 実装における集団型 MPI-IO 関数における IOR ベンチマークの結果の一例である[5]。ES と「京」において各々96 ノード(=384 プロセス)を利用し、ユーザが比較的変更しやすいストライピングサイズ(Ss)を変更した際の、計算ノード側で利用されるそれぞれの並列ファイルシステム ScaTeFS 並びに FEFS の I/O 特性を取得した。ScaTeFS および FEFS のデフォルトストライピングサイズは、256MiB（＝ノンストライピング）と 1MiB となっており、ユーザ側で変更可能となっている。実行した IOR のコマンドオプションは以下のとおりである。

```
./IOR –i 5 –a MPIIO –c –k –m
  –U ${HINTS_FILE} –H –w(-r) –t ${XFER}m
  –b ${BLK}m –s 1 –o ${fname_prefix} –d 0.1
```

ここで、ブロックサイズ${BLK}を 512MiB に設定し、転送サイズ${XFER}を変化させ傾向を調査した。

ベンチマーク結果によると、ES では、Ss をある程度大きくとることで性能を改善できることが分かるが、転送サイズ＝ブロックサイズとなる場合（512MiB）で性能が落ちる場合があることがわかる。一方、「京」では、Ss が小さいほうが書き込み性能が良く、読み込みでは Ss が大きく方が高い性能を示すという特徴を持っている。

このような I/O パターンに依存した挙動および性能評価に関して、その他に POSIX や MPI-IO での Two-Phase I/O 実装方法・特性などに関しても、共通パラメタによるベンチマークを通して調査お

よび情報共有できている。また、MPI-IO 実装においてパラメタ名に違いがあることも取り組みの中で判明しており、ユーザ側ではこのような特性も参考とすることで、アプリケーションの性能改善が行える可能性があると考えられる。

5 まとめと今後の計画

本稿では、機構における I/O 情報の見える化についての取り組みを紹介した。

I/O プロファイル情報およびファイルシステム特性の情報共有は、ユーザ目線に立つと、最適な I/O パフォーマンスのための性能改善が可能となれば、今まで出力していなかったモデルパラメタやより短いタイムステップでのスナップショットなどの新しいアウトプットを増やすことにもつながり、その結果、新たな知見を生み出す可能性があると考えられる。また、集団型 MPI-IO の情報を参照することにより、性能面だけでなく、ファイル管理や自在な実行並列数変更といった面での利便性を向上させることにもつながると考えられる。運用部門にとっては、ユーザサポートや障害対応の際に有効な基礎データとなり得る。また、システム更新の調達時の要件作成時にも、アプリケーション毎のファイル I/O のプロファイルや必要スループットの情報は、I/O ベンチマークテスト策定のみならず、計算能力・ネットワーク能力・ストレージ能力のバランスを考える上で、重要な指針にもできると考えている。

ファイルシステム特性に関してのスパコンセンター間の情報共有の取り組みについては、共通ベンチマークパラメタを用いた定量的な比較のみならず、お互いのファイルシステムやストレージに関する詳細を理解することができたため、ストレージだけでなくシステム全体の効果的な利用方法・運用方法に関しても、有用な情報共有ができると確認した。今後、メタデータ性能やジョブリクエスト統計情報などに関しても情報共有を進めていく計画を進めており、他のスパコンセンターとも是非このような取り組みで連携していきたいと考えている。

参考文献

[1] 国立研究開発法人海洋研究開発機構：地球シミュレータ，https://www.jamstec.go.jp/es/jp/

[2] Darshan: http://www.mcs.anl.gov/research/projects/darshan/

[3] Darshan-riken: https://www-sys-aics.riken.jp/releasedsoftware/ksoftware/darshan/

[4] Lustre manual Chapter 37.3 “Monitoring Lustre File System I/O”: http://lustre.org/documentation/

[5] 辻田祐一,中川剛史,板倉健一,宇野篤也，ファイルシステムの利用高度化に向けたスパコンセンター間での情報共有の取り組み，情報処理学会研究, Vol.2018-HPC-165, NO. 6 (2018).