Vol.2020-HPC-176 No.7
2020/9/25

ファイルシステムの利用動向を中心とした「京」のジョブの特徴分析と「富岳」の運用に対する一検討

辻田　祐一[1,a)]　宇野　篤也[1]　関澤　龍一[2]　山本　啓二[1]　末安　史親[2]

概要：スーパーコンピュータ「京」（以下，「京」）は 2019 年 8 月に約 7 年間の共用利用を終了した．「京」のシステム運用に関わる様々な運用ログを用い，システム運用の改善に運用中から寄与してきた経緯があるが，共用終了後も，スーパーコンピュータ「富岳」（以下，「富岳」）の運用に活かすことも念頭に置いた分析を行っている．「京」のファイルシステムの利用動向の分析もその一環で行っており，ジョブのディスク利用やファイル I/O の傾向などに加え，それらと演算性能，利用メモリバンド幅や電力との関連性を含めた特徴の抽出・分析を進めている．この分析から，ジョブのノード形状毎に異なった特徴を持つ傾向が確認されると共に，ファイル I/O や電力のジョブ毎の特徴付けによる「富岳」の運用高度化の可能性についても検討を行った．

キーワード：スーパーコンピュータ「京」, ファイルシステム, 電力, 相関係数, スーパーコンピュータ「富岳」

1. はじめに

2019 年 8 月に「京」は共用を終了し，現在は，「富岳」の共用開始に向けた取り組みを，「京」の運用で得た知見等も活用しながら精力的に行っている．この取り組みの一環として，「京」の運用中に蓄積された様々な運用ログを用い，「富岳」の運用に有用な知見を集めるための分析も行っている．

「京」の運用においては，消費電力の超過防止や，ファイルシステムの高負荷防止などの観点で，最適なジョブスケジューリングを実現すべく，様々な観点でジョブの利用動向の分析ならびに実運用へのフィードバックを進めてきた経緯がある [1], [2], [3]．スケジューリングの改善におけるファイル I/O に関連する項目としては, ジョブ実行中に使用予定のディスク容量を過剰に設定し，ジョブスケジューリング効率の低下につながりやすいジョブの監視・抽出と，当該ジョブのユーザへの適切な設定依頼を行ってきた．この取り組みの有効性の検証の一環で行った各ジョブのファイル I/O 等の動向分析を通して，実行されたジョブのファイル I/O を中心とする種々のパラメタ間の相関関係の分析結果が，「富岳」の運用の高度化にも有用であると考え，分析ならびに検討を進めている．

[1] 理化学研究所　計算科学研究センター
[2] 富士通株式会社
[a)] yuichi.tsujita@riken.jp

以下，第 2 章では，「京」のファイルシステム周りの構成とジョブ実行におけるファイル I/O について概説し，第 3 章において，ファイル I/O に関する運用改善取り組みの検証から見られたファイル I/O を中心とするジョブ実行状況を報告する．次に第 4 章において，個々のジョブのファイル I/O および関連するメトリックス間の相関の分析結果を報告すると共に，分析結果を基にした「富岳」の運用高度化への寄与の可能性を第 5 章で検討する．関連研究について第 6 章で報告し，最後に第 7 章で本稿のまとめを行う．

2. 「京」におけるファイルシステム構成とジョブのファイル I/O

ジョブの実行効率の向上やジョブのファイル I/O の高速化を目的に Lustre [4] ver. 1.8 をベースに富士通により開発された FEFS（Fujitsu Exabyte File System）[5] を用いた二階層のストレージ構成により，「京」は運用された．ユーザプログラムやデータは Global File System（GFS）に格納されており，ジョブの実行開始前にユーザのジョブスクリプトに指示子と共に指定されたプログラムやデータ等がファイルステージングにより Local File System（LFS）にコピーされるステージイン処理（以下，SIN）が実施される．ジョブの終了後には，ジョブスクリプトの記述に従い，対象のデータ等が LFS から GFS へコピーされるステージアウト処理（以下，SOT）が実施される．なお，SIN に際し，

ジョブスケジューラは個々のジョブに対して，ジョブスクリプト内に記載された LFS 利用におけるノードあたりの要求ディスク容量（node-quota：デフォルトは 14 GB/ノードで最大で 100 GB/ノード）を満たす計算ノード数が確保されるまで再スケジューリングを定期的に実施し，ノード数の割り当てを行う．

node-quota の設定に関しては，特にジョブ側で指示が無い場合には，デフォルトの設定である 14 GB/ノードでディスク容量を確保する．各計算ノードは約 150 GB のディスク領域を有しており，非同期ステージングにより，ディスク容量並びにノード数に空きがあれば，ジョブ開始予定時刻に対してシステム側で定めた一定時間前の段階で後続のジョブの SIN が実施される．node-quota の設定を大きくするほど空きディスク容量のあるノード群の確保が難しくなり，計算ノードの利用率の低下に繋がる．その結果，後続ジョブも含めて SIN 開始時刻やジョブ実行開始時刻が遅れるため，利用者には適切な設定を依頼していた経緯がある．

3. ファイル I/O を中心としたジョブ全体での動向

「京」で実行されたジョブのログを用いてファイル I/O を中心とした動向分析を行うにあたり，前述の過大な node-quota 設定の問題に対して行った運用改善の状況分析を行うと共に，ジョブの特徴分析を行った．以下，個々の分析状況について説明する．

3.1 過大な node-quota 設定の問題と運用改善の経緯

「京」のジョブスケジューリングにおける SIN 開始時刻および実行開始時刻の遅延の理由としては，node-quota 設定よりも，ノード数や最大経過時間，さらにはノード形状の固定などが主たる要因であったが，2015 年度にジョブの待ち時間の長期化が顕著化したことから，運用ログ情報の分析を行ったところ，要因の一つとして，再スケジューリングあたりの node-quota 不足となっていたジョブ数が，2015 年度に入り無視できないレベルで増加していたことが分かった．これを受けて 2016 年度からは，利用実績から大きく乖離した node-quota 設定を行ったジョブの監視を強化した経緯がある．個々のジョブに関して，node-quota 設定値により割り当てられたディスク容量に対する「京」のジョブ統計情報で集計しているファイル I/O 量の割合を判別に利用した．なお，ファイル I/O 量は，あくまでジョブのディスクに対する I/O 処理量であり，ディスク上のデータ量にはなっていない．また，FEFS 側の I/O 統計情報では，loopback ファイルシステムによるランク番号ディレクトリ内での I/O 量は集計できないため，ジョブ統計情報で集計されたファイル I/O 量を用いた判別を行っていた．そのためにアクセスパターンによっては，割り当てられたディスク容量を超えたファイル I/O 量を計上しているジョブも散見されているが，明らかに利用率の低いジョブはこの方法により検知できるため，非効率なディスク利用のジョブの監視と運用改善をこの手法により行ってきた．

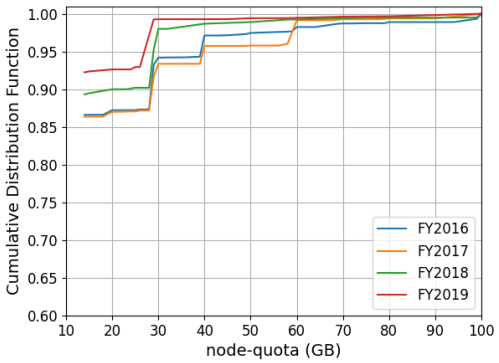


図 1: 2016 年度以降（FY2016 から FY2019）の年度毎の node-quota 設定値に関するジョブ数の累積分布関数

図 1 に 2016 年度以降の年度毎の node-quota 設定値に関するジョブ数の累積分布関数を示す．node-quota 設定に対する監視強化以外の要因もあったとは思われるが，年度が進むにつれて，より低い node-quota 設定を行うジョブの割合が増加していた．また，より少ない設定値への移行が進むことで，上述のような実行開始までの待ち時間の長期化の問題も起きていなかったことを確認している．

3.2 ジョブ全体でのファイル I/O を中心とした利用動向

大きな node-quota が設定されていても，ジョブのファイル I/O に必要な量を確保する限り問題は無いが，その一方で，割り当てられたディスク領域を殆ど利用しないジョブも散見されていた．また，電力に関連する CPU 負荷やメモリアクセス量の大小がファイル I/O 処理量の大小と関係する傾向があり，各パラメタの利用動向を確認した．

なお，通常の運用時において，large キュークラスのジョブ（以下，large ジョブ）が，最も多くの計算ノード数が使えることや，計算ノード区画割り当ても全計算ノードの約 90%であることから，large ジョブの動向が運用上での影響が最も大きいと判断し，本稿では large ジョブにおける動向分析を行った．node-quota の設定値に関して以下に記載した 3 段階に分類し，2016 年度後期から共用終了までの期間で運用ログを用いて分析を行った．

- 20 GB/ノード未満（以下，Q_{low}）
- 20 GB/ノード以上 80 GB/ノード未満（以下，Q_{mid}）
- 80 GB/ノード以上（以下，Q_{high}）

1 番目の Q_{low} は，ほぼデフォルトの設定で実行され，スケジューリングにおいて最も影響の少ないジョブ群として，2 番目の Q_{mid} は，スケジューリングへの影響が Q_{low} よりは大きいが，制限容量近くまでは設定されていないジョブ

群として分類した．最後の Q_{high} は，スケジューリングへの影響が最も大きい高めの node-quota 設定を行うジョブ群として分類した．

large ジョブのうち，ジョブが 10 分以上実行されたものについて，以下に示すように，使用ノード数の累計値に加えて，ファイル I/O と関連性があると考えられる数種類の項目に関して日毎の推移を確認した．

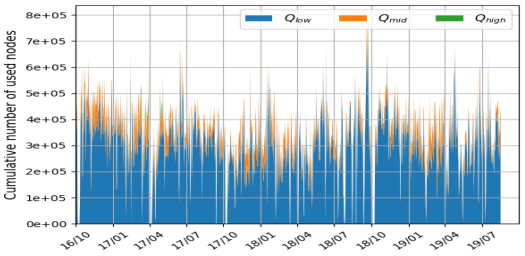
- 使用ノード数の累計値
- node-quota 設定による割り当て済みのディスク容量に対するファイル I/O の正規化された利用率（以下，ディスク利用率：R_D）
- 最大 FLOPS 値に対する到達率の最大値（以下，FLOPS 到達率：R_F）
- 最大メモリバンド幅に対する到達率の最大値（以下，メモリバンド幅到達率：R_M）
- ベース電力分を除くノードあたりの最大電力（以下，ノード単位最大電力：P_{node}^{max}）

R_D は，上述の node-quota の監視における指標になっており，ファイル I/O 量が割り当て済みディスク容量を超えた段階で 1 として扱った．一方，R_F 並びに R_M は，[6] における算出方法に従い，ジョブ毎に集計しているハードウェアモニタ情報等を用い，前者は理論性能上の FLOPS 値に対する到達率の最大値，後者は最大メモリバンド幅に対する到達率の最大値を求めた．「京」のシステムラックや計算ノードには電力計は設置しておらず，各計算ノードに設置されていた温度センサーによる CPU の温度変化とシステムボードの排気の温度変化から電力を求めている [1]．この手法に基づいて，「京」本体のベース電力となる 10 MW [1] を除いた CPU 負荷やメモリアクセス等により変動するノードあたりの最大電力を P_{node}^{max} として算出した．

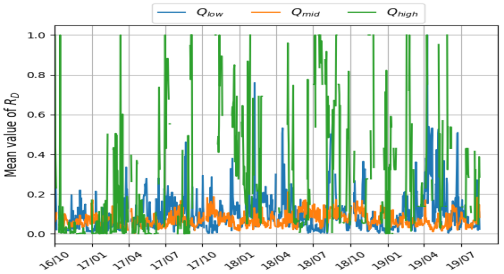
各項目の日毎の推移を**図 2** に示す．図 2(a) に示した日毎の使用ノード数累計値に関しては，Q_{low} に分類されたものが殆どを占めているが，Q_{mid} で分類されているものも一定の割合で存在している．Q_{high} に分類されたものは少ないが，大きな node-quota 設定のジョブが投入されると，容量不足による割り当て計算ノード決定の遅れにより，計算ノード利用率の低下に繋がる可能性がある．

図 2(b) に示した R_D の日毎の平均値については，Q_{high} に分類されたケースにおいては他のケースと比較して R_D の平均値が高かった傾向があるが，Q_{mid} から Q_{low} と node-quota 設定値が低くなるにつれて，R_D の平均値が低下する傾向が確認できた．なお，Q_{high} のケースでは，R_D の平均値の高い日が時間と共に多くなってきた様子が伺えるが，適切な node-quota 設定の周知に基づいて，R_D が高めのジョブで主体的に占められるように推移していたものと考えている．

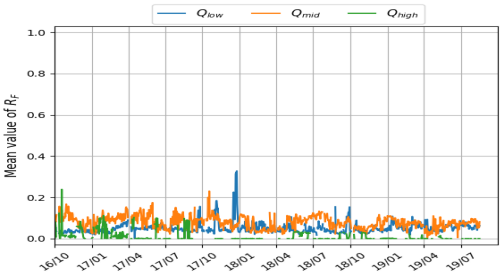
次に，図 2(c) および (d) に示した R_F 並びに R_M の平均値の推移では，高めの node-quota 設定である Q_{high} に比べ，Q_{mid} や Q_{low} のケースの方が到達率が高い傾向が伺



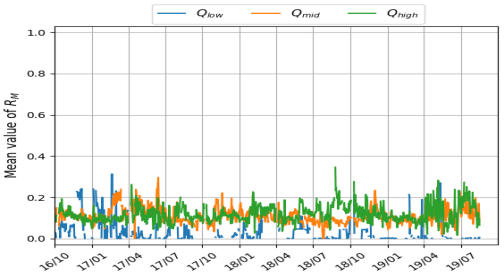
(a) 日毎の使用ノード数累計値



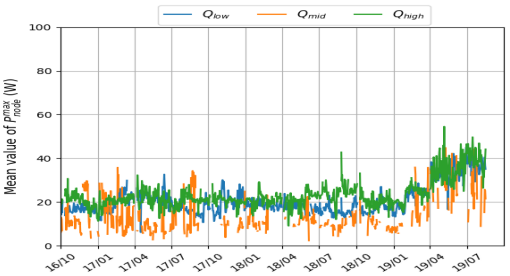
(b) R_D の日毎のジョブ間平均値



(c) R_F の日毎のジョブ間平均値



(d) R_M の日毎のジョブ間平均値



(e) P_{node}^{max} の日毎のジョブ間平均値

図 2: 10 分以上実行された large ジョブの動向（2016 年度後期以降）

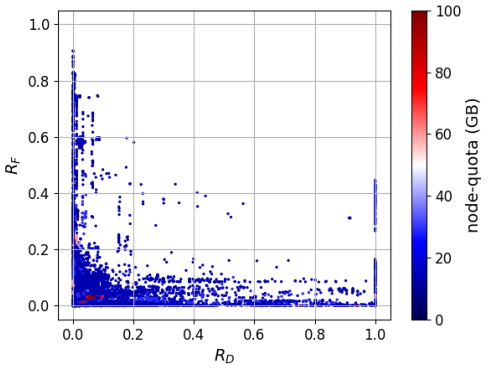
える．Q_{mid} や Q_{low} に分類されるケースは Q_{high} のケースと比較して I/O よりも計算処理やメモリアクセスが主体的なジョブが多かったものと推測される．各計算ノードにおける消費電力の主な要因は CPU とメモリであり，図 2(e) に示す P_{node}^{max} の平均値でも同様の傾向が出ていたと考えている．なお，2019 年 2 月以降の P_{node}^{max} の平均値が時間と共に段階的に増加傾向を示しているが，これは冷却施設での検証試験の一環で，2019 年 1 月 29 日から共用終了まで，冷却水温度を段階的に上昇させた影響と考えている．よって，冷却水温度の影響を排除した上で電力も含めた関連性を検証するために，以降の分析では，半期単位できりの良い 2018 年度上期までのログを用いた．

4. 個々のジョブのファイル I/O 等の動向と相関関係の分析

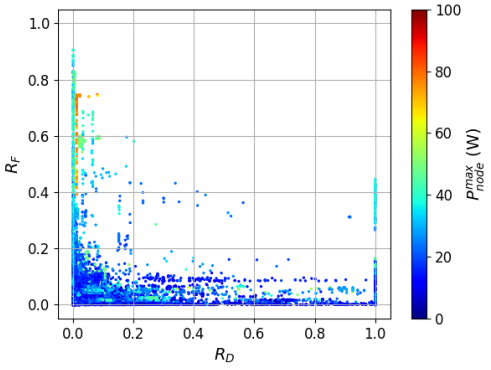
検証を行った各項目間の関連性を検証するために，個々のジョブにおける項目間の関連性の分析を 2016 年度下期から 2018 年度上期までの期間で行った．

「京」の LFS におけるファイル I/O に関しては，他のジョブからの I/O への外乱防止やファイル I/O 帯域の確保などのために，I/O ノードを占有するようにノード形状を 3 次元で固定にしてジョブを投入する事例も散見されている．よって，ファイル I/O 処理の動向はノード形状と密接な関係が想定されるため，ここでは形状毎に分けて分析を行った．ノード形状は 1 次元，2 次元，ならびに 3 次元（以下，それぞれ 1D ジョブ，2D ジョブ，ならびに 3D ジョブ）の 3 種類があり，3D ジョブの場合，ノード形状の固定指示子が付与されている場合，指定形状でのジョブ実行が，付与されていない場合には，スケジューラ側でノード数を確保しつつ適宜形状を変更してジョブを実行させる運用になっていた．ただし，システム側で集計しているジョブ統計情報等からは，ノード状の固定指示子の有無を記録していないため，3D ジョブにおいて，ジョブ形状が投入時から実行時で変更があったもの（以下，3D ジョブ（変更あり））と無かったもの（以下，3D ジョブ（変更なし））で分類した．スケジューラは可能な限り柔軟に形状を変更させてスケジューリング効率の向上を狙う設計のため，後者には形状固定指示子を付与されたジョブが相当数含まれていたものと考えている．2D ジョブの件数は非常に少なく統計的な分析が難しいため，本稿では 1D ジョブ，3D ジョブ（変更あり），および 3D ジョブ（変更なし）の 3 種類について分析を行った．

相関係数の算出には，*Scipy* で提供されているピアソンの積率相関係数，スピアマンの順位相関係数，並びにケンドールの順位相関係数を用いた．なお，ピアソンの積率相関係数においては，他の研究 [7], [8] でも行っているように，データの外れ値の影響を極力軽減するために，対象データを Log スケールに変換して評価を行った．



(a) node-quota 設定値



(b) P_{node}^{max}

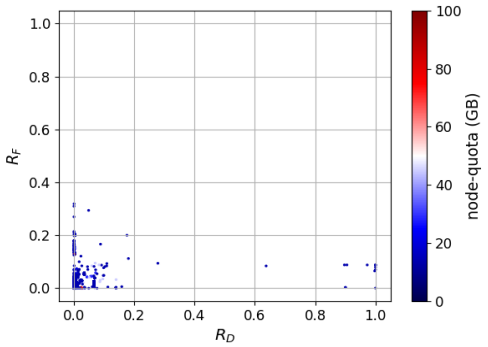
図 3: 1D ジョブにおける R_D と R_F の関係における（a) node-quota 設定値および (b) P_{node}^{max} の分布

4.1 ディスク利用率と FLOPS 到達率の関連性

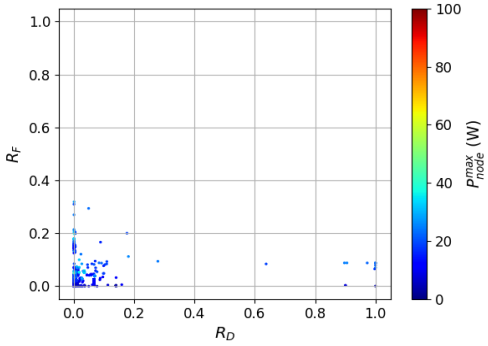
large ジョブに関して，ノード形状毎に R_D と R_F の関連性を node-quota 設定値ならびに電力と合わせて個々のジョブにおけるパラメタ間の関連性を分析した．

1D ジョブに関する結果を**図 3** に示す．全体的な特徴として，R_D が低い程，R_F が高くなる傾向が見られる．図 3(a) からは，R_D および R_F が共に 0.1 以下の領域で比較的高めの node-quota 設定を行っていたジョブが散見されている．このようなジョブでは適切な node-quota 設定に変更することによって，計算ノード利用率向上の可能性があったと考えられる．図 3(b) に示す P_{node}^{max} の状況からは，R_D に関係なく，R_F が低いジョブ群では P_{node}^{max} が低めになっている傾向がある一方で，R_D が 0 付近かつ，R_F が 0.4 以上のあたりに P_{node}^{max} が高いジョブ群が見られた．なお，この高電力傾向を持つジョブ群では図 3(a) から分かるように，node-quota 設定が低い設定であるため，ディスク I/O やスケジューリングの観点から，適切な使い方をされていたと思われる．

次に，3D ジョブ（変更あり）に関する状況を**図 4** に示す．こちらは先程の 1D ジョブとは状況が異なり，R_D および R_F 共に低い領域にジョブが局在している．図 4(a) で示

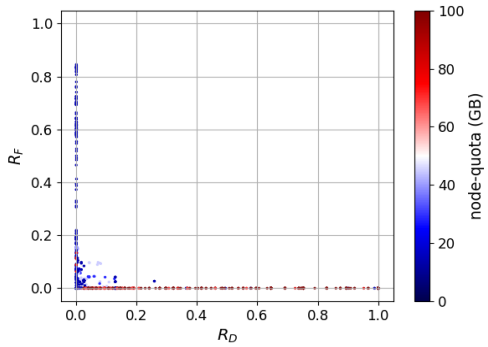


(a) node-quota 設定値

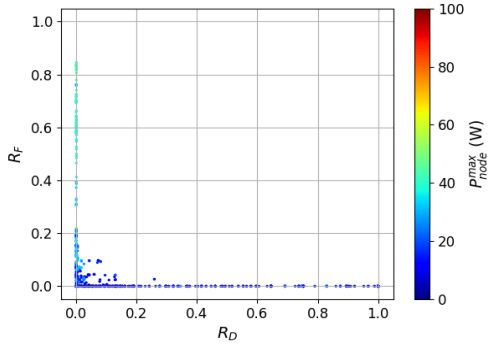


(b) P^{max}_{node}

図 4: 3D ジョブ（変更あり）における R_D と R_F の関係における（a) node-quota 設定値および (b) P^{max}_{node} の分布



(a) node-quota 設定値



(b) P^{max}_{node}

図 5: 3D ジョブ（変更なし）における R_D と R_F の関係における（a) node-quota 設定値および (b) P^{max}_{node} の分布

した node-quota 設定値の状況からは，高めの node-quota 設定をしているにも関わらず R_D が低いジョブは確認できなかった．また，図 4(b) で示した P^{max}_{node} からは，全体的に R_F が低いこともあり，P^{max}_{node} も低めの傾向を示していたと考えられる．

最後に**図 5** に 3D ジョブ（変更なし）の状況を示す．3D ジョブ（変更あり）のケースと比較すると，こちらのケースでは，R_D が 0 から 1 までの間に分布し，かつ R_F が 0 付近にあるジョブ群と R_D は 0 付近だが，R_F が 0 から 1 の間に分布するジョブ群に分かれている点が異なる．前者のジョブ群においては，図 5(a) から，高めの node-quota 設定をしているジョブが多く見られるが，後者のジョブ群では逆に低めの node-quota 設定が多い．また，図 5(b) に示す P^{max}_{node} の分布から，前者は低い電力傾向を示しているが，後者ではやや高めの電力傾向を示しており，非常に対照的である．ノード形状が変更されていないジョブ全てが形状固定指定された 3D ジョブとは限らないが，前者はファイル I/O を重視した 3 次元形状指定と共に高めの node-quota 設定をしたジョブ群が多く，後者では計算・ノード間通信を重視した 3 次元形状指定を行ったジョブ群が多く占めていた可能性が考えられる．

分析した各ノード形状毎に，R_D と R_F との間の相関係数の分析結果を**表 1** に示す．この表にある”P”, ”S”および”K”は，それぞれピアソンの積率相関係数（ただしデータを Log スケールに変換して分析），スピアマンの順位相関係数，並びにケンドールの順位相関係数を表す．3D ジョブ（変更なし）（表中の”3D（変更なし）”）では，比較的強い負の相関が確認されたが，これは，前述の通り，このケースでの特徴的な I/O に重点を置いたノード形状固定に起因するもので，I/O 処理量が多いほど R_F の低下に繋がっていた状況を表しているものと考えている．一方，1D ジョブや 3D ジョブ（変更あり）では有意な相関は確認できなかった．

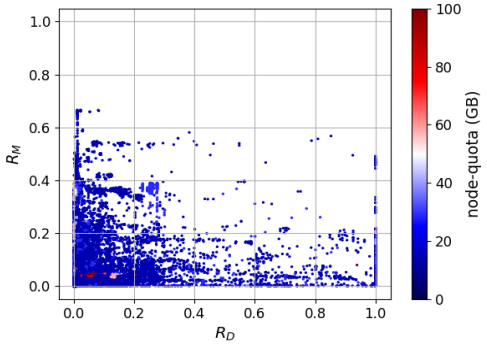
表 1: R_D と R_F の相関係数（上段）と p 値（下段）

ノード形状（ジョブ数）	相関分析の手法		
	P	S	K
1D (127,439)	-0.315 ($p < 0.001$)	-0.345 ($p < 0.001$)	-0.224 ($p < 0.001$)
3D（変更あり） (1,512)	0.135 ($p < 0.001$)	-0.0424 ($p > 0.05$)	-0.0491 ($p < 0.01$)
3D（変更なし） (985)	-0.717 ($p < 0.001$)	-0.696 ($p < 0.001$)	-0.489 ($p < 0.001$)

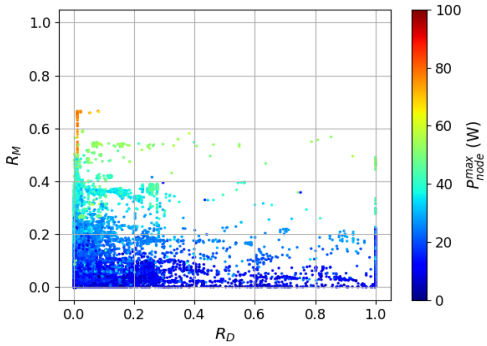
次に R_F と P^{max}_{node} との相関関係の状況を**表 2** に示す．ノード形状に関係なく全体的に高めの相関が確認されているが，FLOPS 値が高くなるほど電力も上昇することが表

表 2: R_F と P_{node}^{max} の相関係数（上段）と p 値（下段）

ノード形状 (ジョブ数)	相関分析の手法		
	P	S	K
1D (127,439)	0.649 ($p < 0.001$)	0.693 ($p < 0.001$)	0.499 ($p < 0.001$)
3D（変更あり） (1,512)	0.613 ($p < 0.001$)	0.662 ($p < 0.001$)	0.459 ($p < 0.001$)
3D（変更なし） (985)	0.745 ($p < 0.001$)	0.693 ($p < 0.001$)	0.505 ($p < 0.001$)



(a) node-quota 設定値



(b) P_{node}^{max}

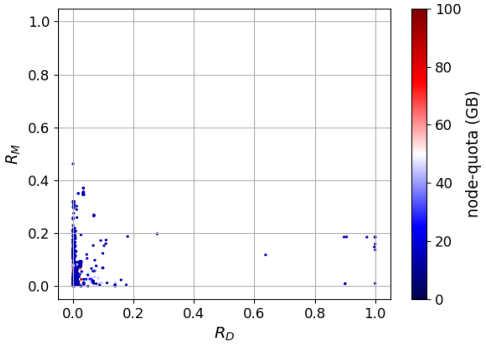
図 6: 1D ジョブにおける R_D と R_M の関係における (a) node-quota 設定値および (b) P_{node}^{max} の分布

されていると考えられる．

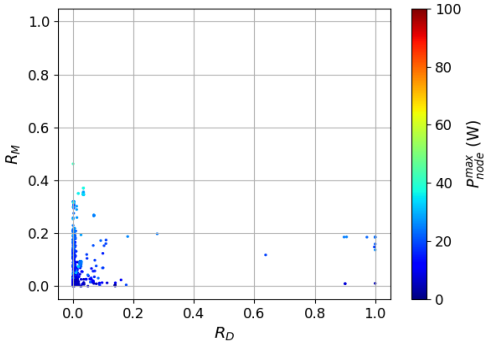
4.2 ディスク利用率とメモリバンド幅到達率の関係

電力の増減に寄与するもう一つの大きな要因であるメモリバンド幅到達率 R_M とディスク利用率 R_D との関係を，同様にノード形状毎に関連性を確認した．

1D ジョブでの結果を**図 6** に示す．R_D が低い方に R_M が高めのジョブ群が多く集まっている様子が伺える．図 6(a) で示す node-quota 設定値では，R_D と R_M 共に低い領域に高めの node-quota 設定を行っていたジョブがあったことが分かる．また，図 6(b) で示した P_{node}^{max} では，R_M が上昇するにつれて P_{node}^{max} も高くなっている傾向が伺え，R_M の高いジョブ群では低目の node-quota 設定が行われていたことも図 6(a) から確認できる．



(a) node-quota 設定値

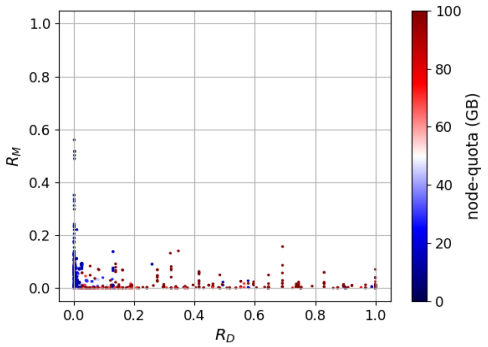


(b) P_{node}^{max}

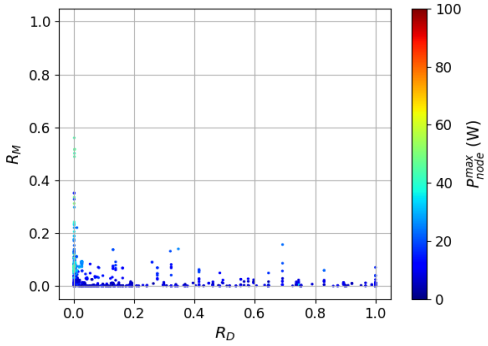
図 7: 3D ジョブ（変更あり）における R_D と R_M の関係における（a) node-quota 設定値および (b) P_{node}^{max} の分布

次に 3D ジョブ（変更あり）における状況を**図 7** に示す．R_D および R_M 共に 0.2 以下の低めの領域に多くのジョブが集まっており，形状に配慮しなくても計算性能や I/O 処理等で特に問題が無いジョブが多数占めていたとも考えられる．図 7(a) で示す node-quota 設定値において，R_D および R_M 共に 0 に近い低い利用率のジョブ群の一部にやや高めの node-quota 設定を行っているジョブが散見されるが，全体としては低めの node-quota 設定のジョブ群で占めれていることが分かる．P_{node}^{max} に関しても，R_M が 0.4 付近でやや高めのジョブ群が見られるが，ほとんどのジョブは低めの値であった．

最後に 3D ジョブ（変更なし）の状況を**図 8** に示す．図 8(a) で示す node-quota 設定値の状況から，R_D が 0 から 1 までの間で，かつ R_M が 0 に近い領域に高めの node-quota 設定のジョブ群が点在している．一方で，R_D が 0 に近く，かつ R_M が 0 から 0.5 あたりまでに点在するジョブ群は低めの node-quota 設定を行っていたジョブで占められている．P_{node}^{max} を示した図 8(b) においても，前者のジョブ群は多数の低めの値のジョブで占められているのに対し，後者のジョブ群はやや高めの値のジョブで占められていた．この特徴は R_D と R_F の関係を示した図 5(a) および (b) で見られた特徴に起因するものと考えられる．



(a) node-quota 設定値



(b) P^{max}_{node}

図 8: 3D ジョブ（変更なし）における R_D と R_M の関係における（a) node-quota 設定値および (b) P^{max}_{node} の分布

表 3: R_D と R_M の相関係数（上段）と p 値（下段）

ノード形状（ジョブ数）	相関分析の手法		
	P	S	K
1D (127,439)	0.0262 ($p < 0.001$)	0.0499 ($p < 0.001$)	0.0621 ($p < 0.001$)
3D（変更あり）(1,512)	-0.126 ($p < 0.001$)	-0.187 ($p < 0.001$)	-0.158 ($p < 0.001$)
3D（変更なし）(985)	-0.509 ($p < 0.001$)	-0.511 ($p < 0.001$)	-0.326 ($p < 0.001$)

分析した 3 種類のケースにおける R_D と R_M の相関係数を**表 3** に示す．3D ジョブ（変更なし）における負の高めの相関係数からも，この分類に含まれる多数のジョブが，ファイル I/O あるいはメモリ帯域の利用率のいずれかに重点が置かれたものであったと言える．一方，他のケースでは有意な相関は確認出来なかった．

次に，**表 4** に R_M と P^{max}_{node} の相関係数を示す．いずれの形状においても高い正の相関が確認されたが，メモリ帯域の利用率が高いほど電力も高めになることによる結果と考えられる．

4.3 FLOPS 到達率とメモリ帯域到達率の関係と電力

最後に FLOPS 到達率 R_F とメモリ帯域到達率 R_M の間

表 4: R_M と P^{max}_{node} の相関係数（上段）と p 値（下段）

ノード形状（ジョブ数）	相関分析の手法		
	P	S	K
1D (127,439)	0.749 ($p < 0.001$)	0.756 ($p < 0.001$)	0.565 ($p < 0.001$)
3D・変更あり (1,512)	0.770 ($p < 0.001$)	0.773 ($p < 0.001$)	0.571 ($p < 0.001$)
3D・変更なし (985)	0.722 ($p < 0.001$)	0.736 ($p < 0.001$)	0.520 ($p < 0.001$)

表 5: R_F と R_M の相関係数（上段）と p 値（下段）

ノード形状（ジョブ数）	相関分析の手法		
	P	S	K
1D (127,439)	0.555 ($p < 0.001$)	0.427 ($p < 0.001$)	0.337 ($p < 0.001$)
3D（変更あり）(1,512)	0.570 ($p < 0.001$)	0.567 ($p < 0.001$)	0.361 ($p < 0.001$)
3D（変更なし）(985)	0.821 ($p < 0.001$)	0.834 ($p < 0.001$)	0.634 ($p < 0.001$)

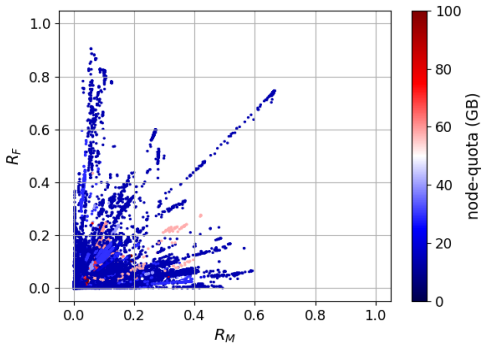
の関連性について，node-quota 設定値，R_D および P^{max}_{node} に着目して分析を行った．

1D ジョブにおける状況を**図 9** に示す．R_M および R_F 共に 0 のところから R_F と R_M が共に線形に増加する方向に伸びた分布が確認できるが，特に両者間の増加比率がほぼ 1 に近い直線に近くなるほど P^{max}_{node} が高くなる傾向が図 9(c) から分かる．この直線に最も近づいたジョブ群では，図 9(a) および (b) から，node-quota 設定かつ R_D が共に低いことからも計算に重点を置いたジョブ群であったと推測できる．

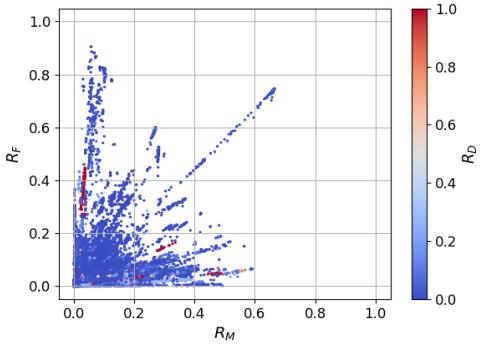
次に 3D ジョブ（変更あり）の状況を**図 10** に示す．これまでの分析からも分かるように R_F および R_M ともに低い領域にジョブ群が集まっている状況が伺えた．既に述べているように，計算やファイル I/O などに対してノード形状の固定化の配慮を必要としないジョブ群が多数占めていたと思われる状況がここにも出ていたと考えている．

最後に 3D ジョブ（変更なし）の状況を**図 11** に示す．ここからは，R_M に比べてやや R_F 側が高くなる傾向のジョブ群が図 11(c) に示すように高めの値を呈していた．またそのようなジョブでは，これまでの分析で述べたように node-quota 設定ならびに R_D 共に低い傾向にある．一方で，図 11(a) および (b) において，R_M が 0 から 0.2 の間で，かつ R_F が 0 付近の領域に着目すると，node-quota 設定値が高く，かつ R_D もそれなりに高いジョブ群が点在していることが分かる．このようなジョブ群はファイル I/O を重視したジョブ群であるものと思われる．

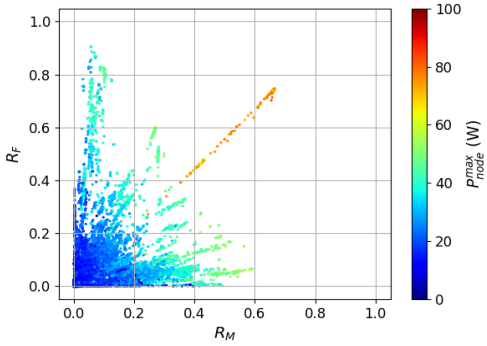
R_F と R_M の間の相関係数についてまとめたものを**表 5** に示す．全般的に正の相関が確認できており，計算処理に重点的なジョブ群における R_F と R_M の関係が表れている



(a) node-quota 設定値

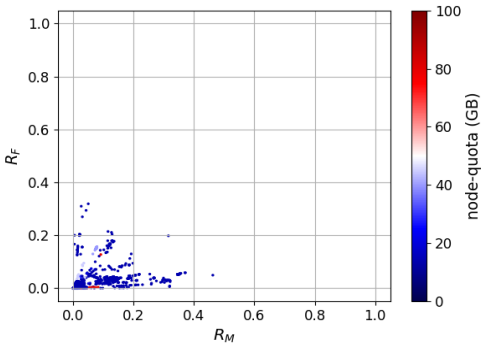


(b) R_D

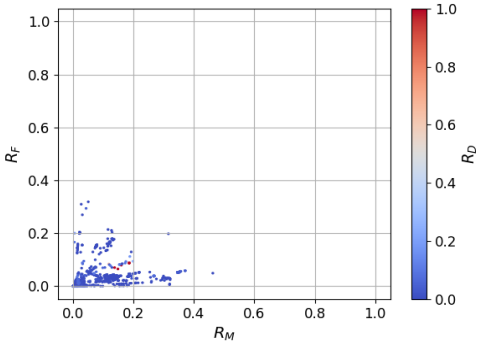


(c) P^{max}_{node}

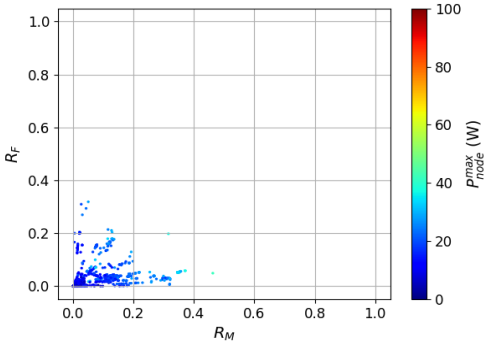
図 9: 1D ジョブにおける R_F と R_M の関係における (a) node-quota 設定値，(b) R_D，および (c) P^{max}_{node} の分布



(a) node-quota 設定値



(b) R_D



(c) P^{max}_{node}

図 10: 3D ジョブ（変更あり）における R_F と R_M の関係における（a) node-quota 設定値，(b) R_D，および (c) P^{max}_{node} の分布

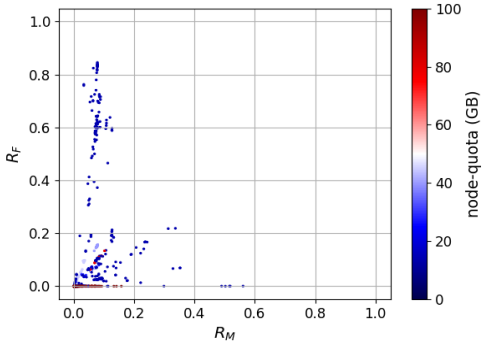
と考えている．

5. 「京」における分析結果に基づいた「富岳」の運用高度化に向けた一検討

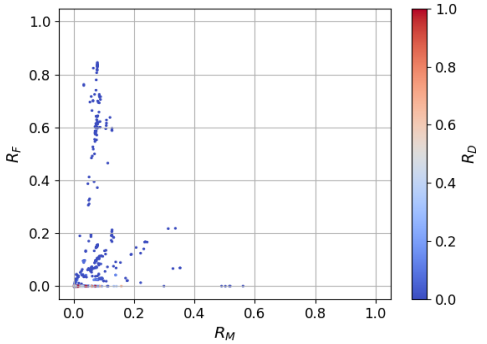
ファイルシステムの利用動向を中心とした「京」のジョブの特徴分析から，ファイル I/O あるいは計算処理のいずれかに重点が置かれたジョブの特徴がノード形状別に確認できた．今回の分析では，ファイル I/O の状況を示す R_D に加えて，電力の指標となる R_F，R_M，並びに P^{max}_{node} との間の相関係数の分析から，以下の 3 種類のジョブ群への分類を試みた．

- $R_F \geq 0.4$ かつ $R_M \geq 0.4$：高電力系ジョブ（P_{high}）
- ($R_F \geq 0.2$ かつ $R_M < 0.4$) あるいは ($R_F < 0.4$ かつ $R_M \geq 0.2$)：電力がやや高めのジョブ（P_{mid}）
- $R_F < 0.2$ かつ $R_M < 0.2$：ファイル I/O 量が多いジョブを含む低電力系ジョブ（P_{low}）

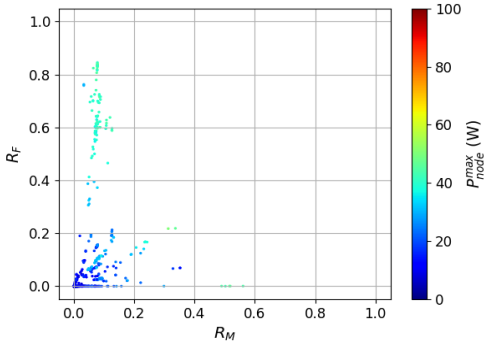
本稿で分析した large ジョブを，この分類に基づいて対象メトリックスの累積分布関数を改めて集計したところ，**図 12** に示すようになった．図 12(a) に示す node-quota 設定値に関しては，一方，P_{low}，P_{mid}，P_{high} と進むにつれて，より少ない node-quota 設定値側にシフトしている様子が伺え，P_{low} では，ほぼデフォルトの 14 GB で占められている．図 12(b) に示す R_D においては，P_{low}，P_{mid}，P_{high}



(a) node-quota 設定値

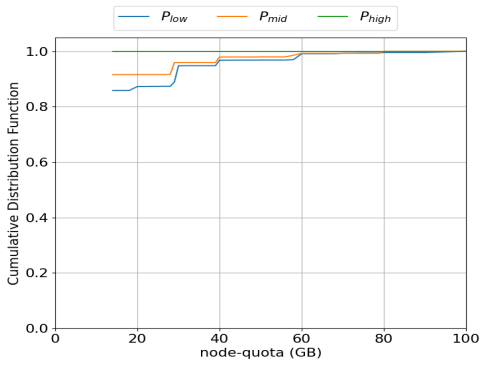


(b) R_D

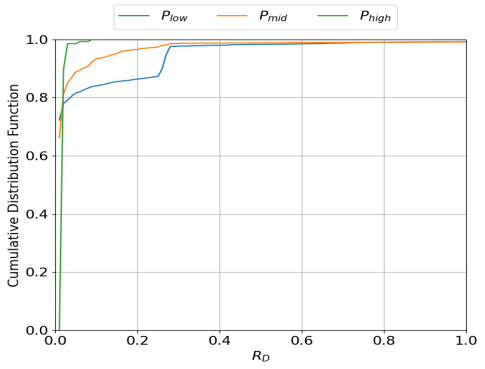


(c) P_{node}^{max}

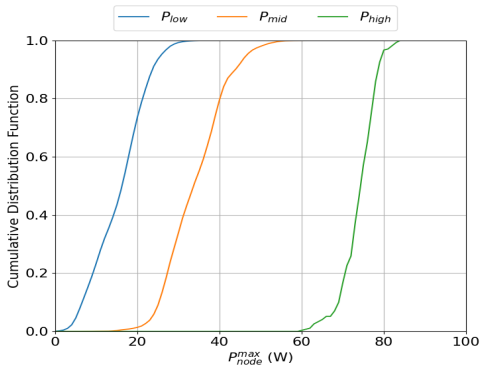
図 11: 3D ジョブ（変更なし）における R_F と R_M の関係における (a) node-quota 設定値，(b) R_D，および (c) P_{node}^{max} の分布



(a) node-quota 設定値



(b) R_D



(c) P_{node}^{max}

図 12: 分析に基づく分類毎の (a) node-quota 設定値，(b) R_D，および (c) P_{node}^{max} に関する累積分布関数

と進むにつれて，低い R_D への局在化が顕著になっている様子が伺える．図 12(a) および (b) からは，P_{high} とそれ以外との間でファイル I/O 処理量の違いが表れたと考えられ，ファイル I/O 量観点で，ある程度の分類ができると考えられる．一方，図 12(c) に示す P_{node}^{max} に関しては，P_{low} のジョブ群は 0 から 30 W にかけて，P_{mid} は 20 W 付近から 60 W あたりにかけて分布しており，P_{high} は 60 W から 80 W 付近あたりまでに分布しているため，60 W を境に，電力の高い P_{high} とそれ以外の 2 つに容易に分類できることが分かる．

なお，検証に用いた数値は「京」における large ジョブ分析での結果であり，「京」とは異なるストレージ構成等を備える「富岳」においては，「京」とは異なった I/O ワークロードにより，状況が異なってくる可能性はある．さらに「富岳」ではユーザのジョブ側から電力制御も行えるため，ジョブ毎の多様な電力傾向も想定される．また「京」では経験の無かった深層学習系などの新しい分野のアプリケーションも「富岳」での利用が見込まれており，これらも含めた分析から異なった動向が見えてくることも想定される．しかしながら，「京」からの継続利用アプリケーションも多く，本稿で行った分析結果も有効に活用することで，適宜調整を加えながら「富岳」の運用高度化に向けた以下のような検討・対策が行えると考えている．

- I/O 処理が重たいジョブ同志で I/O ノードを共有させないスケジューリング
- 電力供給等に配慮したスケジューリング

前者は I/O ノードが担当するファイル I/O での性能ブレや，I/O ノードの負荷を低減することによる計算処理への影響の低減などの効果を期待している．一方，後者は電力供給並びに冷却系の適切な運用に資するための最適なジョブスケジューリング等に寄与することを想定している．「京」で実行されたジョブスクリプト群から学習したアプリケーション予測モデルの提案 [2] や，機械学習等による電力予測の取り組み [9] も行っており，これらの手法との連携により，実行前の投入ジョブへの事前の特徴付けによる最適な計算ノードの割り当てによる効果的なジョブスケジューリングの実現可能性があると考えている．

6. 関連研究

多くの HPC システムにおいて運用ログを用いたジョブの動向分析が行われており，ファイル I/O を中心とした分析も多々報告されている [10], [11], [12]．[10] では，アプリケーションの I/O を含む特徴解析ツールを提案し，これを基に I/O に配慮したジョブスケジューリングを実現している．[11] においては，I/O の大量のログから多角的に動向分析や障害原因の究明を支援するフレームワークを提案しており，I/O 性能情報収集には Darshan [13] を用いている．[12] では，Darshan 等のツール群を組み合わせた HPC システム全体の分析フレームワークを提案している．我々の取り組みも，これらの既存研究と同様に様々なログを活用し，ジョブのファイル I/O を中心とした分析を行っているが，HPC システムのジョブの特徴をファイル I/O に限らず，CPU やメモリ等の消費電力に関係するパラメタも含めた多角的な分析を行っている．

7. おわりに

「京」のジョブのファイルシステムの利用動向を中心に，理論 FLOPS や最大メモリバンド幅に対する到達率や電力等も含めたジョブの特徴分析を行った．システム全体で最も影響の大きいキュークラスのジョブについてノード形状別に分析を進めた結果，形状毎の特徴を確認すると共に，「富岳」の運用高度化に向けた一検討も行った．なお，本稿では触れていないが，計算ノード群と Tofu で結合された I/O ノード群もファイル I/O に関わる重要な要素であり，I/O ノード群における Tofu の通信状態を含めたログ解析による I/O 最適化の可能性を示している [14]．このような解析も組み合わせた多角的なジョブの動向分析が今後期待できる．「富岳」の共用開始に向けて，「京」における運用ログ等の分析から得られる知見も活かしながら高度なシステム運用に向けた取り組みを進める所存である．

謝辞 本稿での分析結果は, スーパーコンピュータ「京」の運用ログおよびジョブ統計情報を用いて得られたものである．

参考文献

[1] 宇野篤也，肥田　元，井上文雄，池田直樹，塚本俊之，末安史親，松下　聡，庄司文由：消費電力を考慮した「京」の運用方法の検討，情報処理学会論文誌コンピューティングシステム（ACS），Vol. 8, No. 4, pp. 13–25 (2015).

[2] Yamamoto, K., Tsujita, Y. and Uno, A.: Classifying Jobs and Predicting Applications in HPC Systems, *High Performance Computing - 33rd International Conference, ISC High Performance 2018, Frankfurt, Germany, June 24-28, 2018, Proceedings*, Lecture Notes in Computer Science, Vol. 10876, Springer, pp. 81–99 (2018).

[3] 古谷吉隆，辻田祐一，山本啓二，宇野篤也，末安史親，肥田　元，岡本光央：計算ノードの使用効率向上を目指した「京」のファイルシステムの運用改善，情報処理学会研究報告，Vol. 2019-HPC-168, No. 21, pp. 1–5 (2019).

[4] Lustre: `http://lustre.org/`.

[5] Sakai, K., Sumimoto, S. and Kurokawa, M.: High-Performance and Highly Reliable File System for the K computer, *Fujitsu Sci. Tech. J.*, Vol. 48, No. 3, pp. 302–309 (2012).

[6] FUJITSU LIMITED: Parallelnavi Technical Computing Language プロファイラ使用手引書利用者向け公開ソフト (2017).

[7] You, H. and Zhang, H.: Comprehensive Workload Analysis and Modeling of a Petascale Supercomputer, *Job Scheduling Strategies for Parallel Processing, 16th International Workshop, JSSPP 2012, Shanghai, China, May 25, 2012. Revised Selected Papers*, Lecture Notes in Computer Science, Vol. 7698, Springer, pp. 253–271 (2012).

[8] 滝澤真一朗，小川宏高，高野了成：大規模 GPU クラスタにおける深層学習ワークロードの傾向把握，情報処理学会研究報告，Vol. 2019-HPC-170, No. 40, pp. 1–8 (2019).

[9] 宇野篤也，末安史親，山本啓二，肥田　元，池田直樹，辻田祐一：ジョブの時系列消費電力の推定，情報処理学会研究報告，Vol. 2019-HPC-170, No. 1, pp. 1–7 (2019).

[10] Liu, Y., Gunasekaran, R., Ma, X. and Vazhkudai, S. S.: Server-side Log Data Analytics for I/O Workload Characterization and Coordination on Large Shared Storage Systems, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC'16, IEEE, pp. 819–829 (2016).

[11] Wang, T., Snydery, S., Lockwood, G. K., Carns, P., Wright, N. J. and Byna, S.: IOMiner: Large-Scale Analytics Framework for Gaining Knowledge from I/O Logs, *2018 IEEE International Conference on Cluster Computing (CLUSTER)*, IEEE, pp. 466–476 (2018).

[12] Lockwood, G. K., Wright, N. J., Snyder, S., Carns, P., Brown, G. and Harms, K.: TOKIO on ClusterStor: Connecting Standard Tools to Enable Holistic I/O Performance Analysis, *2018 Cray User Group Meeting (CUG)* (2018).

[13] DARSHAN: HPC I/O Characterization Tool, `http://www.mcs.anl.gov/research/projects/darshan/`.

[14] 辻田祐一，古谷吉隆，肥田　元，宇野篤也：「京」におけるファイルシステムと I/O ノードのログ情報を用いたファイル I/O の最適化支援の取り組み，情報処理学会研究報告，Vol. 2019-HPC-169, No. 6, pp. 1–8 (2019).