Vol.2014-ARC-213 No.13
Vol.2014-HPC-147 No.13
2014/12/9

分散ファイルシステムGfarmにおけるMTCアプリケーションの性能予測モデルの構築

キョウ ユ[1,a)]　建部 修見[2,3,4]　田中 昌宏[2,3]

概要：天文学，生命科学などの様々な科学分野において，膨大なデータに対する並列分散処理性能の向上は大きな課題になっている．並列処理の性能を向上させるためには，I/O 性能に影響を与える要因の調査が必要である．しかしながら，現状では，I/O 性能調査のためのデータが不足している．本研究では，ワークフロー実行中の各プログラムの I/O 性能を測定できる方法について検討し，分散ファイルシステム Gfarm における MTC アプリケーションの性能予測モデルを構築する．

キーワード：分散ファイルシステム，MTC アプリケーション，並列処理，性能予測

1. はじめに

MTC(Many-Task-Computing) アプリケーションはプロシージャ呼び出しのような簡単なタスクまたはスタンドアロンアプリケーションのような複雑なタスクから構成されるアプリケーションである [1]．天文学，生命科学などの様々な科学分野において多くのデータインテンシブアプリケーションが MTC アプリケーションである．

天文学，生命科学などの様々な科学分野において，扱うデータ量は年々増加している．膨大なデータに対してデータ解析を行うには，並列分散処理が必要となる．複数のプロセスを並列に実行するために，処理内容や依存関係を記述した「ワークフロー」を記述し，それに基づいてクラスタやグリッド上で並列分散処理を行う．こうした並列処理の性能を向上させるためには，I/O 性能調査が必要とする．性能調査の現状では，ワークフロー全体の実行時間や各プログラムの実行時間の測定に止まっている．しかし，これらの実行時間だけでは，I/O 性能の詳しい情報が不明であり，I/O 性能を影響する要因を特定するためのデータが不足している．

本研究では，ワークフロー実行中の各プログラムの I/O 性能を測定する方法について検討し，分散ファイルシステム Gfarm における MTC アプリケーションの性能予測モデルを構築する．このモデルでは，まず，MTC アプリケーション処理の I/O 挙動を解析し，Gfarm に適した I/O 性能の特性を示す MTC envelope を定義した．そして，Gfarm ファイルシステムの構造に基づき，I/O の消耗時間を予測する式を提案した．最後に，筑波大学のクラスタを使用し，定義した MTC envelope の各指標性能をベンチマークで測定し，Montage の一部分である mProjectPP の実行時間を予測した．

[1] 筑波大学大学院システム情報工学研究科
[2] 筑波大学計算科学研究センター
[3] 独立行政法人科学技術振興機構
[4] 筑波大学システム情報系
[a)] gong@hpcs.cs.tsukuba.ac.jp

2. 背景

2.1 Gfarm ファイルシステム

Gfarm ファイルシステム [2] は，1 台のメタデータサーバ (gfmd) と複数のファイルシステムサーバ (gfsd) クライアントで構成される．メタデータサーバは，共通の階層的名前空間，実際のファイルの所在などのメタデータを管理する．ファイルシステムサーバは，ローカルのファイルシステムへのアクセスのために利用されるサーバである．**図 1** は Gfarm 広域ファイルシステムの構成を表している．

Gfarm ファイルシステムは以下の特徴がある．

- ファイルのデータを管理するファイルシステムノードが計算ノードも兼ねることができ，ファイルが置かれたノード，またはネットワーク的に近いノードで処理を行うことにより，ファイルアクセスの効率を高めることが可能である．
- Gfarm ファイルシステムを Fuse でマウントすることが可能で，通常のプログラムから Gfarm によって管理されているファイルを直接読むことができる．
- 複製管理をファイルシステムで行うことで，アクセスの局所性を利用できる．ファイルの複製は，ファイル

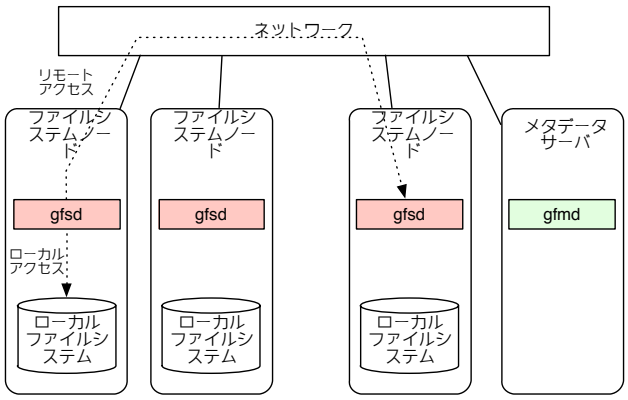

図 1 Gfarm ファイルシステムの概要

参照時の負荷分散，遠隔からの低遅延，高バンド幅のアクセスおよび耐故障性のために利用される．

2.2 並列分散ワークフローシステム Pwrake

Pwrake[3] は，Rake という記述力が高い Ruby 版ビルドツールをベースに，並列分散実行の機能を拡張したワークフローシステムである．Pwrake では，指定されたコア数分だけプロセスを並列に実行することができ，タスクの依存関係を基にして並列実行可能なタスクを自動的に並列実行することが可能となる．Prawke は Gfarm と連携し，自動的にすべてのリモートノードに接続して Gfarm ファイルシステムノードをコアの数だけマウントし，参照されるファイルをワーキングディレクトリに自動的に移動してタスクを実行することができる．また，Pwrake は，Gfarm に付属コマンドを用いて，入力ファイルが格納されているノードの情報を取得し，適切なタスク配置を行う．このタスク配置ではデータ移動を最小化して，Gfarm ファイルシステムの高速なローカルアクセスを生かし，高い並列 I/O 性能を達成することが可能である．

2.3 天文画像処理ソフトウェア Montage

Montage[4] は Many-Task-Computing アプリケーションであり，複数の画像を一つの画像に合成 (モザイキング) を行う汎用ソフトウェアである．

2.4 現状

田中ら (2012)[5] は Montage のワークフローを Rake で記述し，Pwrake による並列実行性能を測定した．性能調査の現状では，Pwrake で Montage ワークフロー全体の実行時間や各プログラムの実行時間の測定に止まっている．しかし，これらの実行時間だけでは，I/O 性能の詳しい情報が不明であり，I/O 性能を影響する要因を特定するためのデータが不足している．

3. 先行研究

Zhang(2013)[6] は Montage を含む 3 種類の Many-Task-Computing アプリケーションを GPFS ファイルシステムで実行した場合の I/O 性能を測定し，性能予測モデルを構築した．

彼らが提案したモデルでは，まず MTC アプリケーションのプロファイリングを行い，I/O 挙動を追跡した．その結果，MTC アプリケーションの主なファイル操作は open, create，read，write で，I/O 性能のボトルネックは並行性，メタデータスループット，小容量ファイルの I/O スループット，大容量ファイルの I/O バンド幅のいずらかであることを明らかにした．そこで，彼らはそれらの指標を一括して記述する MTC envelope を定義し，以下の 8 個のパラメータに限定した：

- create 操作スループット
- open 操作スループット
- 1-to-1 読み込みデータスループット
- 1-to-1 読み込みデータバンド幅
- N-to-1 読み込みデータスループット
- N-to-1 読み込みデータバンド幅
- 書き込みデータスループット
- 書き込みデータバンド幅

また彼らの研究で，Montage の特徴は multi-read-single-write I/O パターンであることを明らかにし，multi-read パターンを 1-to-1 読み込みと N-to-1 読み込みに分類した．1-to-1 読み込みは各タスクが異なるファイルを読み込むことであり， N-to-1 読み込みは複数のタスクが一つの共有ファイルを読み込むこである．

次に，ファイルサイズや I/O ノードの数など様々な条件をつけ，ファイルシステム GPFS においての MTC envelope の性能をベンチマークで測定した．

最後彼らは，I/O の消耗時間を測定する式を提案し，ベンチマークで測定した各指標を代入することで I/O 性能を示すヒートマップを作成した．

図 2 は GPFS 上でのアプリケーションの書き込みバンド幅を表しているヒートマップである．

このヒートマップにより，アプリケーションの I/O 性能を測定することができ，I/O 性能の Bounding Factors を予測することが可能となる．

4. 提案手法

そこで，Gfarm ファイルシステムで同様の性能評価・モデル構築ができると考えられる．

本研究では，天文画像処理ソフトウェア Montage のワークフロー実行中の各プログラムの I/O 性能を測定できる方法について検討し，Gfarm ファイルシステムにおける

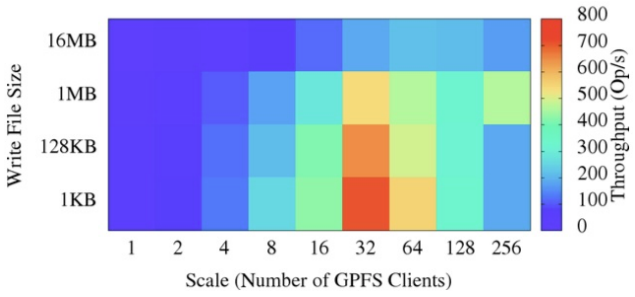

図 2 Heat map of write bandwidth
Cited from [6]

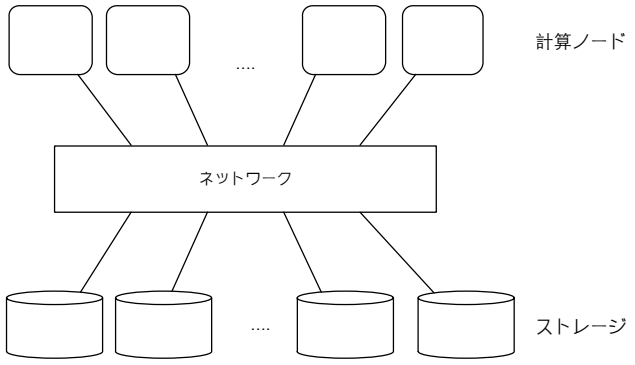

図 3 GPFS の構成

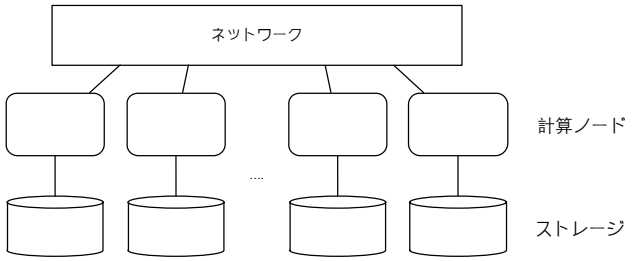

図 4 Gfarm の構成

MTC アプリケーションの性能予測モデルを構築する．

4.1 MTC envelope 性能の定義

MTC envelope の性能と定義するとき，Gfarm の構造により，基準が変わる．

- GPFS(**図 3**) においては，計算ノードとストレージが分かれていて，計算ノードはネットワークを経由し，ファイル操作を行う．アクセスパターンは同じである．
- Gfarm(**図 4**) においては，ストレージの実体を持つファイルシステムノードが計算ノードを兼ねることができ，ネットワークを経由しなく，ローカルストレージの I/O を利用できる．

そのため，読み込み性能測定をローカルとリモートに分類される．また，Gfarm では，プロセスが実行された計算ノードのストレージに出力ファイルが書き込まれるという特徴があるため，書き込み性能測定をローカルのみにする．

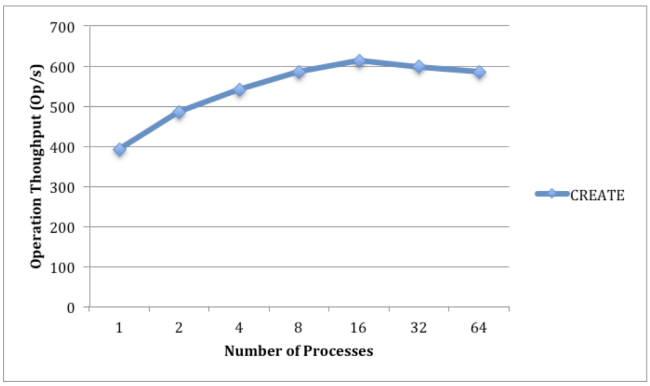

図 5 メタデータ操作スループット

MTC envelope の性能と定義した：

- ファイル create 操作スループット
- 1-to-1 読み込みデータスループット (ローカル＆リモート)
- 1-to-1 読み込みデータバンド幅 (ローカル＆リモート)
- N-to-1 読み込みデータスループット
- N-to-1 読み込みデータバンド幅
- 書き込みデータスループット (ローカル)
- 書き込みデータバンド幅 (ローカル)

4.2 ベンチマーク

ファイルサイズや I/O ノード数など様々な条件をつけ，Gfarm においての MTC envelope の性能をベンチマークで測定する．

	測定環境
CPU	Intel(R) Xeon(R) CPU E5620 @ 2.40GHz (8cores) x2
Memory	24GB
OS	Linux version 2.6.32-431.3.1.el6.x86_64
Filesystem	Gfarm file system
	metadata server*1, filesystem node*8
Benchmark	IOR[7], mdtest[8], iozone[9]

図 5 はプロセス数の増加とメタデータ操作である create 処理のスループットの関係を表している．

図 6 は読み込みファイルのサイズがそれぞれ 1KB, 128KB, 1MB, 16MB の場合, プロセス数の増加と 1-to-1 ローカル読み込み操作のスループットの関係を表している．

図 7 は読み込みファイルのサイズがそれぞれ 1KB, 128KB, 1MB, 16MB の場合，プロセス数の増加と 1-to-1 ローカル読み込み操作のバンド幅の関係を表している．

図 8 は読み込みファイルのサイズがそれぞれ 1KB, 128KB, 1MB, 16MB の場合，プロセス数の増加と 1-to-1 リモート読み込み操作のスループットの関係を表している．

図 9 は読み込みファイルのサイズがそれぞれ 1KB,

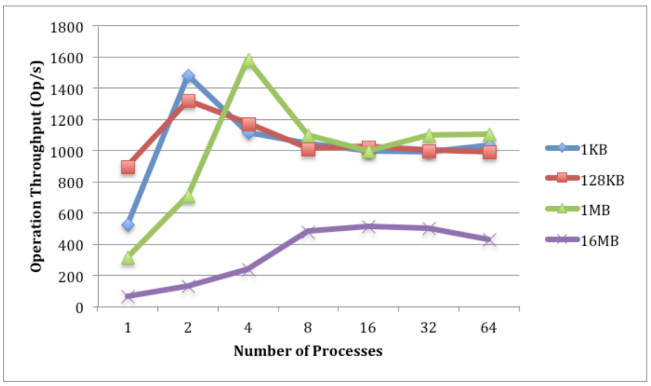

図 6 1-to-1 読み込みスループット (ローカル)

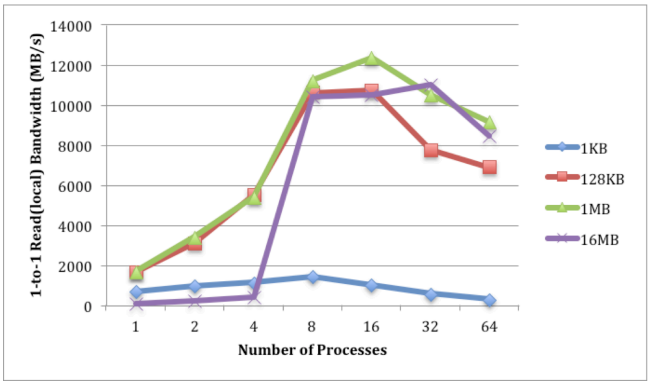

図 7 1-to-1 読み込みバンド幅 (ローカル)

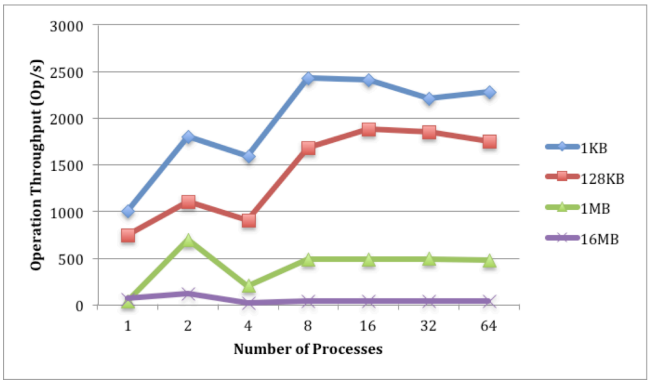

図 8 1-to-1 読み込みスループット (リモート)

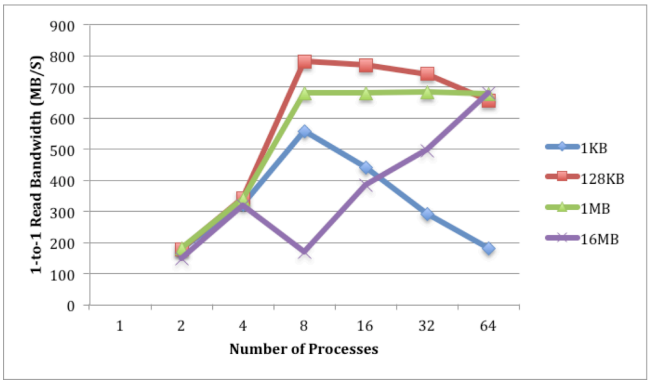

図 9 1-to-1 読み込みバンド幅 (リモート)

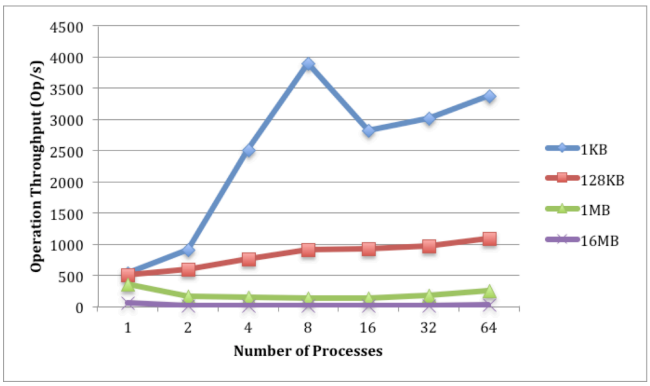

図 10 N-to-1 読み込みスループット

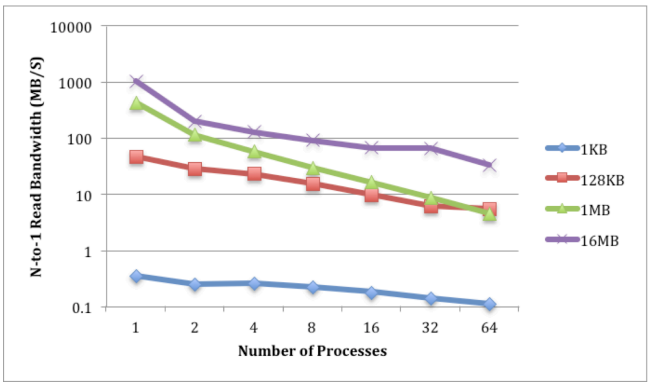

図 11 N-to-1 読み込みバンド幅

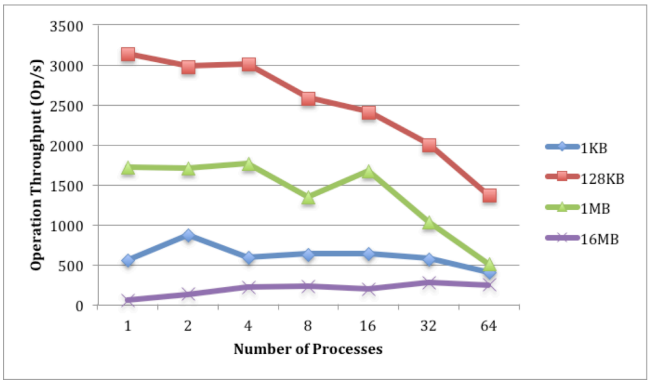

図 12 書き込みスループット

128KB, 1MB, 16MB の場合，プロセス数の増加と 1-to-1 リモート読み込み操作のバンド幅の関係を表している．

図 10 は読み込みファイルのサイズがそれぞれ 1KB, 128KB, 1MB, 16MB の場合，プロセス数の増加と N-to-1 読み込み操作のスループットの関係を表している．

図 11 は読み込みファイルのサイズがそれぞれ 1KB, 128KB, 1MB, 16MB の場合，プロセス数の増加と N-to-1 読み込み操作のバンド幅の関係を表している．

図 12 は書き込みファイルのサイズがそれぞれ 1KB, 128KB, 1MB, 16MB の場合，プロセス数の増加と書き込み操作のスループットの関係を表している．

図 13 は書き込みファイルのサイズがそれぞれ 1KB,

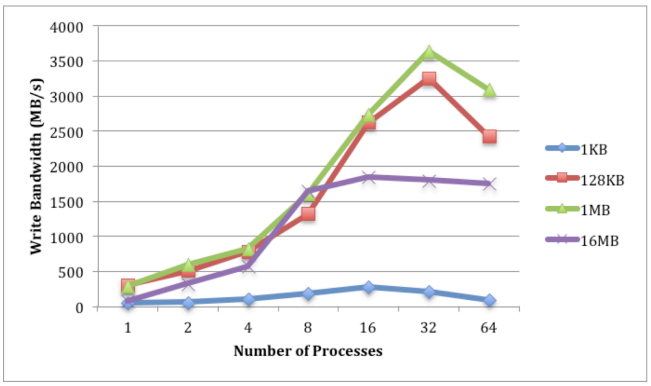

図 13 書き込みバンド幅

128KB, 1MB, 16MB の場合，プロセス数の増加と書き込み操作のバンド幅の関係を表している．

4.3 プロファイリング

Gfarm 上で並列分散ワークフローシステム pwrake で，ソフトウェア Montage を実行し，Gfarm ライブラリへの I/O 挙動をトレースし，各段階の主なプログラムはローカルで実行する割合を調査した．

- 入力ファイル　：956 枚の画像
- ファイルサイズ：1 枚あたり約 2MB

	mProjectPP	mDiff	mBackground
ローカル割合	94%	62%	63%

4.4 タイムコンサンプションの予測

Gfarm においての I/O のタイムコンサンプションを予測する式を提案する．

4.4.1 書き込みタイムコンサンプション

バンド幅バウンドの場合:

$$Time = \left\lceil \frac{N}{C} \right\rceil * (\frac{1}{T_m} + \frac{C * D}{B_L}) \quad (1)$$

スループットバウンドの場合：

$$Time = \left\lceil \frac{N}{C} \right\rceil * (\frac{1}{T_m} + \frac{C}{T_L}) \quad (2)$$

ここのスループットは I/O 操作を実行する速度という意味であり，単位は $operation/s$ である．バンド幅はデータ転送速度で，単位は $bytes/s$ である．

N はアプリケーションの毎段階のタスク数，C は計算ノード数で，$\lceil \frac{N}{C} \rceil$ は全部のタスクを書き込みのラウンド数である．T_m は create のスループットで，Gfarm ではメタデータサーバが 1 台のみのため，毎ラウンドのメタデータ操作がかかる時間は 1 個の操作わる操作の速度 T_m である．

バンド幅バウンドの場合，各書き込みの出力ファイルサイズは D bytes，ローカル書き込みのバンド幅は B_L メガバイト毎秒で，毎ラウンドの全部の書き込み操作がかかる時間は C 個の操作かける各書き込みの出力ファイルサイズであり，この全部の出力ファイルサイズわるデータ転送速度 B_L である．

スループットバウンドの場合，T_L はローカル書き込み操作のスループットで，毎ラウンドの全部の書き込み操作がかかる時間は C 個の操作 ÷ 操作の速度 T_L である．
メタデータ操作がかかる時間プラス書き込み操作がかかる時間，そしてラウンド数をかけると，アプリケーション毎段階の書き込みタイルコンサンプションを予測できる．

4.4.2 読み込みタイムコンサンプション

$$Time = \lceil \frac{N}{C} \rceil \times [max(\frac{\alpha C}{T_{1L}}, \frac{C * \alpha D}{B_{1L}}) + max(\frac{(1-\alpha)C}{T_{1R}}, \frac{C * (1-\alpha)D}{B_{1R}}) + max(\frac{C}{T_N}, \frac{C * D}{B_N})] \quad (3)$$

N はアプリケーションの毎段階のタスク数，C は計算ノード数で，$\lceil \frac{N}{C} \rceil$ は全部のタスクを書き込みのラウンド数である．

Gfarm ではファイル配置により性能が変わるため，1-to-1 読み込みがローカルで実行する割合を α にする．

T_1 は 1-to-1 読み込みのスループット，同時に各読み込みの入力ファイルサイズは D_1 bytes，1-to-1 読み込みのバンド幅は B_1 メガバイト毎秒で，毎ラウンドの 1-to-1 読み込みがかかる時間はデータ転送時間と読み込み操作時間の長い値である．

T_N は N-to-1 読み込みのスループット，共有ファイルサイズは D_Nbytes，N-to-1 読み込みのバンド幅は B_N メガバイト毎秒で，毎ラウンドの N-to-1 読み込みがかかる時間は，データ転送時間と読み込み操作時間の長い値である．

1-to-1 読み込みがかかる時間 + N-to-1 読み込みがかかる時間，そしてラウンド数をかけると，アプリケーション毎段階の読み込みタイルコンサンプションを予測できる．

4.5 評価

Montage のプログラム mProjectPP を例として，書き込みタイムコンサンプションの評価を行った．

- タスク数　　：955
- ノード数　　：8
- 出力ファイル：4.2MB

実測 (秒)	予測 (秒)	誤差 (秒)
2.28	2.64	15.7%

5. まとめと今後の課題

本研究は分散ファイルシステムにおいてプログラムのI/O 性能を測定できる方法について検討し，Gfarm ファイルシステムにおける MTC アプリケーションの性能予測モデルを構築した．そして，Gfarm 上での MTC アプリケーション処理の I/O 挙動のトレースと MTC envelope 性能のベンチマーク測定を行った．実測と比べ，提案手法の予測は 15.7%の誤差があることが明らかになった．

しかし，今回の評価では，タスク全体の実行時間は約 168 秒と比較的短かいものであった．今後では，テストケースを拡大し，より正確に評価できることを課題とする．また，次の段階では図 2 で示したようなヒートマップを作成し，I/O 性能のボトルネックを発見し，性能を向上させるための最適化の方法を見つけることが目標である．

参考文献

[1] Raicu, I., Foster, I. T. and Zhao, Y.: Many-Task Computing for Grids and Supercomputers, *IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS08) 2008*.
[2] Tatebe, O., Hiraga, K. and Soda, N.: Gfarm Grid File System, *New Generation Computing*, Vol. 28, pp. 257–275 (2010).
[3] Tanaka, M. and Tatebe, O.: Pwrake: A Parallel and Distributed Flexible Workflow Management Tool for Wide-area Data Intensive Computing, *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, New York, NY, USA, ACM, pp. 356–359 (2010).
[4] : Montage, NASA California Institute of Technology (online), available from ⟨http://montage.ipac.caltech.edu⟩
[5] 田中昌宏，建部修見：並列分散ワークフローシステム Pwrake による大規模データ処理 (宇宙科学情報解析論文誌 第一号)，宇宙航空研究開発機構研究開発報告， Vol. 11, pp. 67–75 (2012).
[6] Zhang, Z., Katz, D. S., Wilde, M., Wozniak, J. M. and Foster, I.: MTC Envelope: Defining the capability of large scale computers in the context of parallel scripting applications, *Proceedings of the 22nd international symposium on High-performance parallel and distributed computing*, ACM, pp. 37–48 (2013).
[7] Shan, H. and Shalf, J.: Using IOR to Analyze the I/O performance for HPC Platforms, *Lawrence Berkeley National Laboratory* (2007).
[8] Welch, B. and Unangst, M.: Clustered and Parallel Storage System Technologies, *7th USENIX Conference on File and Storage Technologies (FAST'09)* (2008).
[9] Norcott, W. D. and Capps, D.: Iozone filesystem benchmark, *URL: www. iozone. org*, Vol. 55 (2003).