Grid Datafarm におけるスケジューリング・複製手法の性能評価

竹房 あつ子[†1] 建部 修見[†2]
松岡 聡[†3,†4] 森田 洋平[†5]

グリッド技術を基盤にした大容量データに対する遍在するアクセスを可能にする技術をデータグリッドと呼び，複数のシステムの設計・実装が行われている．しかしながら，それらは実験段階にあり，データグリッドアーキテクチャの設計方針の妥当性や性能に関する議論は不十分である．本稿では，Bricks グリッドシミュレータにデータグリッドシステムに対する拡張を行い，Grid Datafarm アーキテクチャに基づくデータグリッドモデルとその性能について比較・調査した．データグリッドモデルでは，Central モデルと Tier モデルを比較し，Tier モデルでは様々なスケジューリングと複製アルゴリズムを適用し，2006 年に行われる CERN の高エネルギー物理実験を想定し，その性能を評価した．

Performance Analysis of Scheduling and Replication Algorithms on Grid Datafarm Architecture for High Energy Physics Applications

Atsuko Takefusa,[†1] Osamu Tatebe,[†2] Satoshi Matsuoka[†3,†4]
and Youhei Morita [†5]

Data Grid is a Grid environment for ubiquitous access and analysis of large-scale data. Due to its early research status, the performance of petabyte-scale Data Grid models in a realistic data processing setting have not been well investigated. By enhancing our Bricks Grid simulator to be able to simulate Data Grid scenarios, we investigate and compare the performance of different Data Grid models in the Grid Datafarm architecture, mainly categorized into the central and the tier models but with varying scheduling and replication strategies, under realistic assumptions of job processing for the CERN LHC experiments.

1. はじめに

グリッド技術を基盤にした大容量データに対する遍在するアクセスを可能にする技術をデータグリッドと呼び，複数のシステムの設計・実装が行われている．Grid Datafarm[1)~3)]，EU DataGrid[4)]，GriPhyN[5)]，PPDG[7)] はその代表的なプロジェクトであり，特に，2006 年開始予定の CERN Large Hadron Collider (LHC) 実験をターゲットとして開発されている．LHC 実験では数十各国数千人規模の物理学者により大規模加速器から得られるペタバイト規模のデータ解析が行われる．大容量ディスクおよび計算資源を要するため，MONARC(Models of Network Analysis at Reginal Centres)[6)] プロジェクトでは各国に地域センタを配置し，地球規模の多階層データグリッドモデルが提案されている．一方，それらは現在開発・実験段階にあり，提案されているデータグリッドシステムアーキテクチャの設計方針の妥当性や実用性，実アプリケーションを想定した性能評価に関する議論は不十分である．

本稿では，Bricks グリッドシミュレータに対しデータグリッドのためのディスクシミュレータの拡張と複製機構を組み込み，データグリッドシステムモデルとその性能について比較・調査した．評価では，1 つのサイトで集中的にデータ解析を行う Central モデルと MONARC で提案されている Tier モデルを比較した．また，Tier モデルでは様々なスケジューリングと複製アルゴリズムを提案・適用し，LHC 実験を想定してその性能をシミュレーションにより評価した．

2. データグリッドプロジェクト

GriPhyN では，データグリッドのシミュレーションとして，アプリケーションのいくつかのアクセスパターンを想定し，多様な複製 / キャッシング手法を評価している[8)]．評価では，“データのダウンロード” に要する応答時間を比較し，“fast spread” 手法ではネットワークバンド幅が節約可能であり，“cascading” 手法では応答時間を短縮できることを示した．しかしながら，こ

†1 お茶の水女子大学 Ochanomizu University
†2 産業技術総合研究所 National Institute of AIST
†3 東京工業大学 Tokyo Institute of Technology
†4 国立情報学研究所 National Institute of Informatics
†5 高エネルギー加速器研究機構 High Energy Accelerator Research Organization

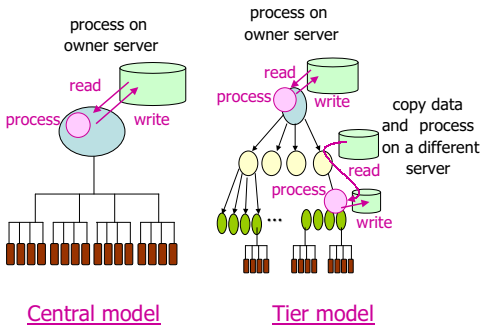

図 1　Central モデル（左）と Tier モデル（右）．

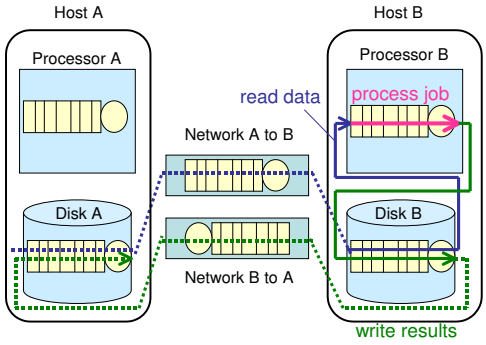

図 2　データグリッドコンピューティングにおける Bricks シミュレーションの流れ．

の評価は既存のデータ処理システムを仮定したデータアクセスに対する評価であり，実際の大規模アプリケーションを想定したデータ処理を含めた評価にはなっていない．

Grid Datafarm では，ペタバイト規模のオンライングローバル並列ストライピングファイルシステムを提供する．これにより，数万ノードにおよぶグリッド上のクラスタを利用し，スケーラブルな I/O バンド幅，並列処理を実現可能とする．一般に，データグリッドにおけるデータ処理システムではペタバイト規模のデータを扱うため，計算に必要なデータを HPSS 等の高性能ディスクシステムに格納し，適宜計算ホストにロードして処理する．一方，Grid Datafarm では高 I/O バンド幅を達成するため，クラスタノードのディスクに分割・格納されたデータに対し，owner-computes ルールでプロセスをスケジュールし，並列実行する．本稿の評価では，データグリッドシステムにとして Grid Datafarm アーキテクチャを想定する．

3. シミュレーションモデル

データグリッドに対し極端な要求をする LHC 実験をアプリケーションモデルとして評価を行う．

3.1 データグリッドアプリケーションモデル: CERN LHC 実験

LHC 実験では，粒子の衝突実験から測定されるペタバイト規模のデータ（イベント）を収集し，以下の段階的な処理が（）内の頻度で行われる[6]．

Large: **RAW→ESD:** RAW データを再構成し，ESD(Event Summary Data) を生成する（2-4/ 年）

Medium: **ESD→AOD:** ESD を用い，AOD(Analysis Object Data) を生成する（1/ 月）

Small: **AOD→TAG:** AOD を用い，TAG データを生成する（1/4 時間）

LHC 実験での典型的なジョブ Large，Medium，Small は，いずれも数百万もの物理イベント処理の集合からなり，それぞれのイベント処理は独立なためイベント単位で並列データ処理が可能である．各ジョブはデータグリッドシステム上で次のように処理される．

(1)　クライアント計算機でユーザ（物理学者）がジョブをデータグリッドシステムに投入する

(2)　データグリッドスケジューラがジョブに対して適切なサーバ群を選択する

(3)　各サーバは割り当てられたタスクを処理する

(4)　サーバは指定されたディスクに出力データを送る（クライアントには統計情報のみが返される）

選択された各サーバはそのサーバ上で処理されるタスク（ジョブの一部）に要するデータがローカルディスクにない場合，ネットワーク経由でロードされる．

ジョブの処理に要する時間 $T_response$ は，T_read，$T_process$，T_write を入力データの読み込み時間，ジョブの処理時間，出力結果の書き出し時間としたとき，以下のように表される．

$$T_response = T_read + T_process + T_write \quad (1)$$

3.2 データグリッドアーキテクチャ

MONARC[6] では，単一サイトで構築可能な計算・ディスク資源に制限があるという前提で，多階層地域センタモデルを提案した．一方，近年のコモディティ PC およびクラスタリング技術の発展は目覚ましい．Grid Datafarm では，それらを利用し大規模ディスククラスタを設計・構築しており，6) で必要とされている計算資源を単一サイトで確保できる可能性は十分にある．

本研究では Grid Datafarm アーキテクチャを仮定して単一サイトで全てのジョブを処理する Central モデルと，多階層の地域センタでジョブを処理する MONARC 型の Tier モデルを比較する（図 1）．Tier モデルでは効率的なデータ処理のため，適切なユーザジョブの割当てと適切なデータ複製の必要がある．それらスケジューリングおよび複製手法の詳細は 4節で述べる．

3.3 Bricks シミュレータのデータグリッド拡張

データグリッドアプリケーションの性能を評価するため，Bricks グリッドシミュレータ[9] を拡張した．Bricks は Java で実装された離散イベントシミュレータ

であり，典型的なグリッドのスケジューリングモジュール群 (Scheduling Unit) と動的なシミュレーショングリッド環境を提供し，様々なスケジューリングアルゴリズムの解析を可能にする[10),11)]．

データグリッドにおける精密なシミュレーションではディスクアクセスは無視できないため，ディスクアクセスの挙動を待ち行列を用いて表現するよう Bricks システムを拡張した．各ジョブは Bricks では図 2のように処理される．図中の実線はローカルホストでジョブが処理される場合のワークフローであり， Disk B からデータを読み， Processor B でそのジョブを処理し，結果を Disk B に格納する．一方，破線 + 実線のワークフローはジョブ処理に要するデータが計算ホスト上に格納されておらず，結果も計算ホストとは異なるホストに格納する場合を示す．この場合，データを Disk A から Network A を介して Disk B に格納した後読み込み， Processor B でジョブを処理して結果を Disk B，Network B を介して Disk A に格納する．

本稿のシミュレーションでは， Processor と Disk の待ち行列は時分割処理され， Network の待ち行列は FCFS で処理する．ただし，ネットワークではデータは指定された論理パケットサイズに分割・転送される．また，図 2で示すように， Disk では read 時， write 時の遅延とも， 1 つの待ち行列により表すことにする．

また，データグリッドのための拡張として Bricks Scheduling Unit モジュールで複製マネージャを提供するようにした．複製マネージャの詳細は 4節で述べる．

4. Tier モデルのスケジューリングと複製管理

多階層分散データグリッドシステムでは，ユーザのジョブを効率よく処理するために適切なスケジューリングと複製手法を用いなければならない．多くの複製を生成すると効率的な負荷分散が実現でき，応答時間の短縮が望めるが，ディスクサイズの制限やネットワークバンド幅への圧迫により性能に悪影響を与える．

4.1 Grid Datafarm における複製管理

2節で述べたように， Grid Datafarm システムでは拡張ストライピングクラスタファイルシステムを提供し，データグリッドアプリケーションに必要なデータを断片化してメタデータにより管理する．メタデータはファイルステータス，ファイル断片ステータス，ディレクトリ，複製カタログ，ファイルシステムノードステータスからなり，ファイルシステムメタサーバが管理する．これらの情報により，分散データグリッドシステム上で耐故障性，広バンド幅，低レイテンシ，負荷分散の実現が可能となる．本稿のスケジューリング・複製手法は，これらのメタデータを利用することを前提とする．

4.2 オンラインスケジューリングアルゴリズム

スケジューラは発行されるジョブに対して入力データのオーナーである DataSourceHost，ジョブを処理する ComputeHost，結果が格納される DataDestinationHost を適切に選択する． DataSourceHost!=ComputeHost / ComputeHost!=DataDestinationHost の場合，入力 / 出力データの複製がオンデマンドに生成される．一方，複製マネージャは定期的にグリッド環境情報を収集し，複製の生成，移送，削除をバックグラウンドで管理する．

シミュレーションでは，次のオンラインスケジューリングアルゴリズムを比較・評価する．

Greedy: 処理時間の最短にすることを目指すアルゴリズムであり， MCT(Minimum Completion Time) として知られている[12)]．スケジューラは式 (1) に示す応答時間が最短になるように DataSourceHost， ComputeHost， DataDestinationHost を割り当てる．

OwnerComputes: 発行されたジョブの処理に必要な入力データを格納しているホストの中から処理時間が最短となるホストを計算ホストとして選択する．この場合， DataSourceHost， ComputeHost， DataDestinationHost は全て同じホストとなる．

LoadBound-Read: スケジューラは MCT で以下を満たすホストから計算ホストを選択する．

$$Perf_{specified} > Perf_{estimated} \quad (2)$$

$$Perf_{estimated} = Perf/(LoadAvg + 1) \quad (3)$$

$Perf$ はサーバの性能， LoadAvg は負荷平均値，$Perf_{estimated}$ はある時点でのサーバの処理性能の見積もり値，$Perf_{specified}$ はあらかじめ指定した性能値を示す．ジョブは適切なホストから入力データを読み込み， ComputeHost に出力結果を格納する．

LoadBound-Write: スケジューラは以下の $T_{duration}$ を最小にする ComputeHost を選択する．

$$T_{duration} = T_{read} + T_{process} \quad (4)$$

この際，選択された ComputeHost が式 (2) を満たさなければ，式 (3) が最大となるホストに出力データを送信する．これにより，処理能力のあるホストへの負荷の分散を図る．

全てのアルゴリズムにおいて，$Perf_{specified}$ があるホストの性能 $Perf$ より大きい場合，そのホストはスケジューリングの対象から外れる．

4.3 複製アルゴリズム

複製マネージャは定期的にデータグリッドシステム上の計算ホストの状況を調べ，次のように適宜複製の生成，移送を実施する．

LoadBound-Replication: 複製マネージャは定期的に全てのホストに対して式 (3) で $Perf_{estimated}$ を算出する．あるホストの $Perf_{estimated}$ が式 (2) を満たす場合，$Perf_{estimated}$ が最小のホストから最大のホスト， 2 番目に小さいホストから 2 番目に大きいのホスト，...，へと複製を生成して転送する．

複製マネージャはアクセス率 AR が最も多いデータの複製を生成する．

$$AR = N_{accesses}/(T_{current} - T_{stored}) \quad (5)$$

表 2　シミュレーション環境のパラメータ．

モデル	ディスク容量 [PB]	サイト性能 [MSI95]	サイト内ノード数
Central	2	0.5-1.8	10000
Tier	Tier0(x1): 2	0.6/0.5/0.4	10000
	Tier1(x4): 1	0.3/0.25/0.2	5000
	Tier2(x16): 0.1	0.03/0.025/0.02	500

$N_{accesses}$ はデータへの総アクセス数，$T_{current}$ と T_{stored} は現在の時刻とそのデータがディスクに格納された時刻を示す．

4.4　評価でのスケジューリングと複製手法の組合せ

評価では，表1に示す組合せでスケジューリングと複製手法を適用する．いくつかのホストでは生成されたデータのオリジナルコピーを管理し，それらのデータの複製が他のホストに転送される．すなわち，データは移動するのではなく，コピーされる．例えば，DataSourceHost と ComputeHost が異なる場合，そのジョブに要するデータが DataSourceHost から ComputeHost のディスクにコピーされる．もし，そのディスクの空き領域が不十分だと判明した場合（スケジューラが検出），またはデータグリッドシステム上のホストのうち x% のホストのディスクの使用領域が y% 以上となった場合（複製マネージャが検出），”複製の削除”を行う（パラメータ x，y は実行時に指定）．本シミュレーションでは複製の削除のために，以下のアルゴリズムを用いる．

(1)　データグリッドシステム上に複製をもつデータのリストを作る

(2)　(1) のデータを最後にアクセスされた時刻が古い順に並べる

(3)　(2) のリストの最初から N 個のデータを対象にし，式 (6) でデータのアクセス率 AR_{elim} を計算する

$$AR_{elim} = N_{accesses}/(T_{current} - T_{stored})/N_{copies} \quad (6)$$

N_{copies} はあるデータの複製の総数を示す

(4)　以下の条件を満たすまで式 (6) の小さいデータを順に削除する．$Size_{total}$，$Size_{available}$ はディスクの総容量と利用可能容量，$Compactness$ は複製削除の頻度調節パラメータである．

$$Size_{total} \times Compactness > Size_{available} \quad (7)$$

(5)　式 (7) が満たされない場合は，次の N 個のデータに対して (3) 以降のステップを繰り返す

本シミュレーションでは，N を 10 とした．

複製削除の命令は複製マネージャが発行するが，各サイトでのディスク空き領域の調査，複製削除アルゴリズムの実行はローカルディスクマネージャが行うため，スケーラブルにデータグリッドシステム上のディスク領域管理が可能である．

5.　シミュレーションによる評価実験

評価では，ジョブの応答時間を比較する．

表 3　LHC 実験のジョブパラメータ．各ジョブのイベント数は全て 1G 個．

Job	計算量 [GSI95*sec]	平均頻度	入力 [TB]	出力 [TB]
Large	1000	1/4[months]	1000	100
Medium	25	1/1[month]	100	10
Small	5	1/4[hours]	10	0.1

表 4　LHC 実験データ RAW(1PB)，ESD(100TB)，AOD(10TB)，TAG(10GB) の平均増加量．

フェイズ	データとその個数
0mth	1PB(1)
4mth	1PB(2), 100TB(1)
8mth	1PB(3), 100TB(2), 10TB(4)
12mth	1PB(4), 100TB(3), 10TB(8), 10GB(720)
16mth	1PB(5), 100TB(4), 10TB(12), 10GB(1440)
20mth	1PB(6), 100TB(5), 10TB(16), 10GB(2160)

5.1　シミュレーションシナリオ

3節で述べたように図 1の 2 つのモデルを比較する．

Central モデル:　全てのジョブリクエストが処理できる十分な計算性能をもつ 1 つのサイトに全てのデータが格納されており，そこで全てのジョブを処理する．安定したジョブ処理が可能となる計算性能は，待ち行列理論よりを見積もることができる．

Tier モデル:　1 サイトの負荷が増加すると，他の地域センタにデータの複製を生成・転送し，そこでジョブを処理する．Tier モデルでは，表 1に示す 5 種類のスケジューリング・複製手法の組合せを用いる．

評価では，データグリッドシステム上に 1 つのデータグリッドスケジューラを想定し，サイトに対してジョブを割り当てるものとする．サイト内ではローカルスケジューラが各ホストにジョブを割り当てる．

表 2にシミュレーション評価実験環境のパラメータを示す．これらのパラメータは GriPhyN のシミュレーション[8] における設定パラメータをもとに決定した．Tier モデルでは，Tier0 が 1 サイト，Tier1 が 4 サイト，Tier2 が 16 サイトとし，Tier3 にはユーザの各計算機があるものとする．Tier 間の性能比は (Tier0, Tier1, Tier2) = (0.6, 0.3, 0.03), (0.5, 0.25, 0.025), (0.4, 0.2, 0.02) [M SI95(10^6SpecINT95)] とした．Central の最低性能を 0.5[MSI95] としたのは，0.453318[MSI95] より性能が低い場合飽和して想定する LHC ジョブを処理できないことが待ち行列理論で明らかなためである．

WAN とローカル I/O のバンド幅は 2006 年の時点で実現する技術を想定し，それぞれ 10[Gbps] と 100[MB/sec] とした．また，各ジョブは 1 つのサイト内で処理されるものとし，Grid Datafarm システムを想定して各サイトでは並列 I/O，並列処理することにする．一般のクラスタ並列ファイルシステムでは，I/O ノード数を増やすとディスク I/O バンド幅が LAN のバンド幅により制限されるが，Grid Datafarm アーキテクチャではデータアクセス局所性のあるファイルに対し

表 1　評価でのスケジューリングと複製手法の組合せ．

Scheduling Policies	Replication Policies	ComputeHost Selection	Timing of Replication	Object of Replica Creation	Destination of Replica
Greedy	-	MCT	Read	Input Data	Compute Host
OwnerComputes	-	Owner + MCT	-	-	-
LoadBound-Read	-	MCT + Load	Read	Input Data	Compute Host
LoadBound-Write	-	MCT + Load	Write	Output Data	Arbitrary
OwnerComputes	LoadBound-Write	MCT	Read/Anytime	Input Data / Arbitrary	Compute Host / Arbitrary

数千ノードのスケーラブルな I/O バンド幅が期待できる．すなわち，Tier0 の各ホストのローカル I/O バンド幅が 100[MB/sec]，ノード数が 10000 の場合，Tier0 での総バンド幅は 1[TB/sec] となる．

3.1節で述べたように，シミュレーションでは実際の LHC 実験での 3 つの異なるレベルの解析ジョブ (表 3) を複数同時に実行する．表 3のデータの粒度・頻度の場合，表 2の Central モデルでは全てのジョブの平均応答時間が待ち行列理論で 38.575-1.337[hours] になると予測できる．また，表 3より LHC 実験の RAW(1PB)，ESD(100TB)，AOD(10TB)，TAG(10GB) のデータは表 4のように増加する．表中の () 内の数値は各フェイズでの複製を含めない各データの総数を示す．

評価では，8mth から 23mth 終了までの 1 年分のシミュレーションを数千回行う．シミュレーションの実行には，東京工業大学の Presto IV クラスタ (Dual Athlon MP 1900+，768MB Memory，256 nodes) を用いた．全シミュレーションの開始時には全てのデータ (1PBx1，100TBx2，10TBx4) は Tier0 に格納しておく．また，1PB の RAW データは HPSS のような異なるディスク領域に格納されることを想定し，各シミュレーションの間 RAW データはシミュレーション環境中のディスク領域で増加しないものとする．

GriPhyN のシミュレーション[8] では，データアクセスに関して時間的，地域的，空間的局所性を挙げている．本シミュレーションでは，ランダムアクセス (局所性なし) と時間的局所性 (最近アクセスされたデータは再びアクセスされやすい) を持つアクセスパターンを想定した．LHC 実験では新しく加速器から得られた観測情報データや興味深い解析結果が得られるデータに対して頻繁にアクセスが発生する傾向があるため，時間的局所性を適用する．一方，地域的，空間的局所性の重要性が明らかでないため本シミュレーションでは用いない．

5.2　シミュレーションによる評価結果

図 3，4，表 5に Central と Tier の実験結果を示す．これらは，各モデル，時間的局所性をもつアクセスパターン，スケジューリング・複製手法の組合せに対する 10 回のシミュレーションでの平均応答時間を算出したもので，シミュレーション中に発行されたジョブ Large，Medium，Small の総数はそれぞれ 30，102，21693 であった．

図 3では Central での平均応答時間を示す．グラフ

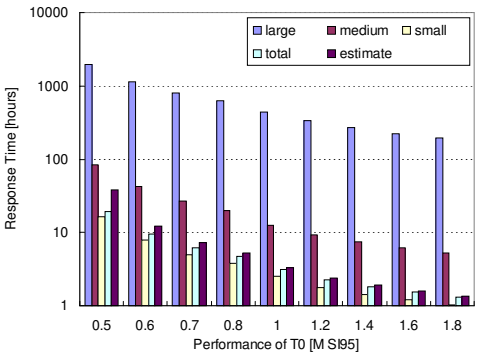

図 3　Central の異なる処理性能における応答時間の比較．Tier0 の性能は 0.5-1.8[MSI95] ．

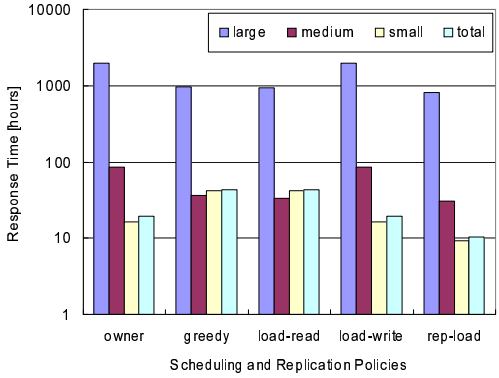

図 4　Tier で異なるスケジューリング・複製手法を用いたときの応答時間．(Tier0, Tier1, Tier2) = (0.5, 0.25, 0.025)[MSI95] ．

中の large，medium，small は表 5のジョブ Large，Medium，Small であり，total は全ジョブの平均応答時間，estimate は待ち行列理論で算出した平均応答時間の見積値を示す．グラフの x 軸には Tier0 サイトの総計算性能を示し，y 軸にはログスケールで平均応答時間を [hours] で示す．図 3より，サイトの総性能の低下に伴い応答時間が急激に増加していくことが分かる．

図 4は Tier で表 1に示した異なるスケジューリングと複製手法を適用したときの平均応答時間をログスケールで表している．各 Tier サイトでの総計算性能は (Tier0, Tier1, Tier2) = (0.5, 0.25, 0.025)[MSI95] である．(Tier0, Tier1, Tier2) = (0.6, 0.3, 0.03)[MSI95] の場

表 5　Central と Tier（OwnerComputes と LoadBound-Replication を適用）の応答時間の比較．

	Tier Model [hours]			Central Model [hours]				
Job	Tier0=0.4	Tier0=0.5	Tier0=0.6	0.4	0.5	0.6	0.7	0.8
Large	1107.225	821.429	631.699	na	1952.214	1149.828	809.387	629.545
Medium	39.884	30.912	23.859	na	84.502	42.633	26.639	19.731
Small	14.286	9.262	6.258	na	16.211	7.783	5.007	3.799
total	15.907	10.479	7.199	na	19.186	9.514	6.213	4.732
estimate	na	na	na	na	38.575	12.277	7.300	5.194

合もほぼ同じシミュレーション結果を示したため，本稿では Tier0=0.5 の場合のみを示す．'Owner'は，複製手法は用いず OwnerComputes を用いた時のシミュレーション結果であり，図 3の Tier0 の総性能が 0.5[MSI95] の時の結果と等しい．図 4では，LoadBound-Read を用いた場合に平均応答時間が増大している．これは，LHC 実験解析ジョブの入力データはテラバイトからペタバイトに及ぶため，LoadBound-Read で負荷分散のために入力データをオンデマンドにコピーしてもそのオーバーヘッドが大きすぎてしまうためである．一方，OwnerComputes と LoadBound-Replication を用いた場合では非常によい性能を示しており，入出力データサイズが非常に大きいデータグリッドアプリケーションでは各サイトの負荷を考慮してバックグラウンドでデータの複製を作る手法が有効であることが分かる．

表 5は Central と Tier の平均応答時間の比較結果である．Tier では，OwnerComputes と LoadBound-Replication を適用している．図 3で示したように，Central は Tier0 サイトの総計算性能が高くなるにつれ性能が向上するが，計算要求に対して総計算性能が十分にないと応答時間が急激に増大する．よって，電力的，空間的，経済的な要因等で 1 つのサイトに配置できる計算・ディスク資源に制限がある場合，Central では性能低下が著しい．一方 Tier の場合，Tier0 の性能が 0.6，0.5[MSI95] の場合のシステム全体の総計算性能はそれぞれ 2.28，1.9[MSI95] であり，同性能の Central と比較すると Tier でバックグラウンド複製手法を用いた場合でもその性能差は著しい．しかしながら，1 サイトに十分な計算性能があればシステムの安定性を維持してジョブを非常に効率よく処理できるが，Tier では Central より各 Tier の総性能が低く構成することができる．すなわち，Tier0 の性能が 0.4[MSI95] の場合のように，Tier0 の総性能が Central での処理限界より小さくなる場合でも，OwnerComputes と LoadBound-Replication のように適切なスケジューリング・複製手法を用いていれば安定したジョブ処理が可能であることが分かる．

6. まとめと今後の課題

本稿では，Bricks グリッドシミュレータにデータグリッドシステムに対する拡張を行い，データグリッドシステムモデルとその性能についてシミュレーションで比較・評価した．評価では，単一サイトで集中的にデータ解析を行う Central モデルと MONARC の Tier モデルを比較し，十分な計算性能を保持できれば Central で効率よくデータグリッドジョブを処理可能であるが，1 サイトの性能に制限がある場合でも，Tier モデルで適切なスケジューリング・複製手法を適用すれば，著しく性能低下しないことが分かった．

より効率的なスケジューリング・複製アルゴリズムを提案し，スケーラブルかつ様々な環境を想定した評価を行うことが今後の課題である．

謝辞　日頃よりご討論頂く Grid Datafarm プロジェクトの皆様に感謝致します．

参 考 文 献

1) Grid Datafarm: `http://datafarm.apgrid.org/`.
2) Tatebe, O. et al: Grid Datafarm Architecture for Petascale Data Intensive Computing, *CCGrid2002*, pp. 102–110 (2002).
3) 建部修見ほか: ペタバイトスケールデータインテンシブコンピューティングのための Grid Datafarm アーキテクチャ, 情報処理学会論文誌： HPCS (to appear) (2002).
4) EU DataGrid: `http://www.eu-datagrid.org/`.
5) GriPhyN: `http://www.griphyn.org/`.
6) Aderholz, M. et al: Models of Networked Analysis at Regional Centres for LHC Experiments, Monarc phase 2 report (2000).
7) PPDG: `http://www.ppdg.net/`.
8) Ranganathan, K. and Foster, I.: Identifying Dynamic Replication Strategies for a High Performance Data Grid, *Grid Computing* (2001).
9) Bricks: `http://ninf.is.titech.ac.jp/bricks/`.
10) Takefusa, A. et al: Overview of a Performance Evaluation System for Global Computing Scheduling Algorithms, *Proc. of HPDC-8*, pp. 97–104 (1999).
11) Takefusa, A. et al: A Study of Deadline Scheduling for Client-Server Systems on the Computational Grid, *Proc. of HPDC-10*, pp. 406–415 (2001).
12) Maheswaran, M. et al: Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems, *Journal of Parallel and Distributed Computing*, Vol. 59, pp. 107–131 (1999).