地球シミュレータ用ジョブスケジューリングアルゴリズムの評価

宇野篤也[†]　板倉憲一[†]

地球シミュレータは 640 台の計算ノードを単段のクロスバネットワークで接続した大規模分散メモリ型並列計算機である．この巨大なシステムを効率良く運用するために，専用のジョブスケジューラが開発されている．本稿では，地球シミュレータ用ジョブスケジューリングアルゴリズムの性能をソフトウェアシミュレータを用いて評価した．また，現在導入中のハードディスクストレージシステム導入後の性能予測も行った．

Evaluation of job scheduling algorithm for the Earth Simulator

Atsuya Uno[†] and Ken'ichi Itakura[†]

The Earth Simulator is a distributed-memory parallel system which consists of 640 nodes connected via a fast 640×640 single-stage crossbar network. The job scheduler customized for the Earth Simulator is developed to control this huge system efficiently. This paper discusses the performance evaluation of the job scheduling algorithm for the Earth Simulator using a software simulator. The performance of a new hard disk data storage system is also estimated.

1. はじめに

地球シミュレータ[1)2)]は，大気大循環シミュレーションなど地球変動研究に代表される超大規模科学技術計算向けプラットフォームとして開発された超高速並列計算機で，640 台の計算ノードが単段クロスバネットワークで接続された大規模分散メモリ型並列計算機である．地球シミュレータは平成 14 年 2 月末に完成し，同年 3 月より運用が開始されている．

本稿では，大規模並列計算機である地球シミュレータ用に開発されたジョブスケジューリングアルゴリズムの評価を行った．評価には，実際に地球シミュレータ上で実行されているジョブの特性を考慮したジョブを使用した．また，現在使用されているカートリッジテープライブラリをハードディスクドライブで構成されるデータストレージシステムに置き換えた場合の性能予測も行った．

2. 地球シミュレータのアーキテクチャ

地球シミュレータ (ES) の構成を図 1 に示す．ES は 640 台の計算ノード (PN:Processor Node) を単段のクロスバネットワークで結合したメモリ分散型並列計算機である．PN はピーク性能 8Gflop/s の計算用ベクトルプロセッサ (AP:Arithmetic Processor) 8 台，16GB の主記憶ユニット (MMU:Main Memory Unit)，リモートアクセス制御装置 (RCU:Remote Access Control Unit) 及び 入出力プロセッサ (IOP:I/O Processor) から構成されている．地球シミュレータ全体では AP 5120 台，ピーク性能 40Tflop/s，主記憶 10TB である．

2.1 クラスタ

ES ではシステムを効率よく運用するために，640 ノードを 16 ノードづつ 40 個に分割して管理している．これをクラスタと呼ぶ．図 2 にクラスタおよび周辺機器構成を示す．

40 個のクラスタのうち，0 番目のクラスタを S 系と呼ぶ．S 系 16 ノードのうち，2 ノードはインタラクティブノードとして使用する．S 系の残り 14 ノードで，NQS を利用して小規模ジョブ (ユーザは最大 8AP まで使用可能) を処理する．残り 39 クラスタは L 系と呼び，大規模ジョブで使用する．各クラスタには CCS (Cluster Control Station) と IOCS (Input Output

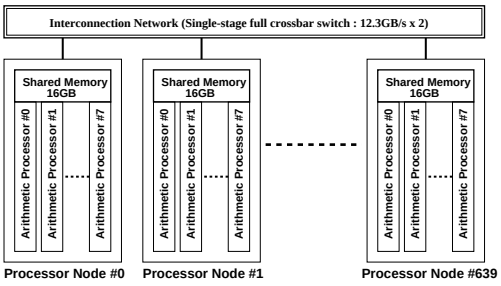

図 1　地球シミュレータの構成

† 海洋科学技術センター 地球シミュレータセンター
Japan Marine Science and Technology Center

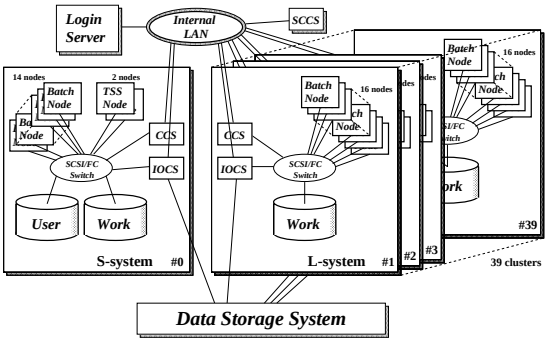

図 2 地球シミュレータの周辺機器構成

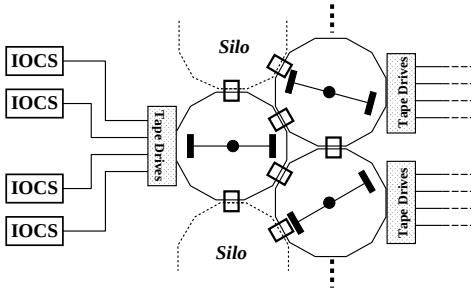

図 3 カートリッジテープライブラリ

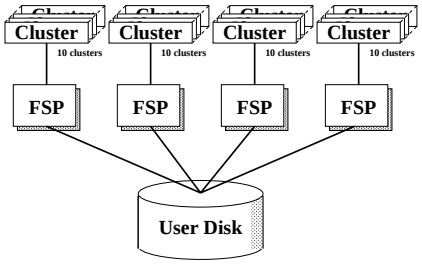

図 4 ディスクストレージシステム

Control Station) の 2 つのワークステーションが置かれている．CCS は各ノードの起動制御やノード間のリソースの共有等を管理していて，SCCS (Super Cluster Control System) が全ての CCS をコントロールしている．IOCS はジョブ実行準備段階でのノード間のデータ転送や，ノードのワークディスクとデータストレージ間のファイルのリコールとマイグレーション等を制御している．

L 系では，ES 用にカスタマイズされた NQS (gm-nqs) が使用されている．gm-nqs は L 系でのジョブ実行機能だけを提供していて，ジョブのスケジューリングは ES 用に開発されたジョブスケジューラが行っている．

2.2 データストレージシステム

L 系ノードのワークディスクは，ジョブが実行時に一時的に使用する領域で，ユーザファイルは置かれていない．L 系のユーザファイルはデータストレージシステムに格納されている．これらのファイルはノードから直接参照できないので，ジョブで使用するファイルはジョブ実行前に L 系のワークディスクへ転送する必要がある．ユーザはジョブで使用するファイルをジョブスクリプトに記述する．ジョブスクリプトに記述されたファイルは，ジョブの実行前に自動的にデータストレージから L 系のワークディスクに転送され，ジョブ終了後に L 系のワークディスクからデータストレージへ自動的に転送される．

S 系ではファイル転送機能はサポートされていない．S 系には L 系のデータストレージとは別にユーザディスクがマウントされている．小規模ジョブはユーザディスク上にあるファイルを直接参照することができる．しかし，データストレージを直接参照することはできないので，小規模ジョブでデータストレージにあるファイルを使用する場合には，ユーザがジョブ実行前に S 系のディスクへ書き出しておく必要がある．

2.2.1 カートリッジテープライブラリ

現在の ES では，データストレージシステムとしてカートリッジテープライブラリが使用されている (図 3)．このシステムでは，各 IOCS に 2 台づつのテープドライブがマウントされ，システム全体で計 80 台が使用されている．L 系のワークディスクとデータストレージ間のファイル転送のスループットは 800MB/s である．

L 系のワークディスクとカートリッジテープライブラリ間のスループットがボトルネックになるのを避けるため，I/O の並列化が行われている．I/O の並列化により，システムから見たデータ転送のスループットは向上した．しかし，ユーザが実行結果を他のワークステーションや S 系で利用するためには，カートリッジテープライブラリから S 系にマウントされたディスクへ一度書き出す必要があり，この作業にかなりの時間が必要となる．これらの問題を改善するべく，現在，データストレージシステムとして使用しているカートリッジテープライブラリを，ハードディスクをベースにしたデータストレージシステムに変更する作業が行われている．

2.2.2 ディスクストレージシステム

現在，テープライブラリをハードディスク装置に置き換えたシステムの導入が行われている．このシステムでは，ユーザファイルのあるディスクドライブから L 系のワークディスクへのファイル転送を FSP (File Service Processor) が行う．FSP は計 4 台あり，各 FSP は 10 クラスタ，160 ノードを担当する．

ディスク装置だけでは，テープドライブを使用している現在のシステムと比べて容量が不足する．そのため，テープライブラリを利用した階層型ファイルシステムを構築する予定である．

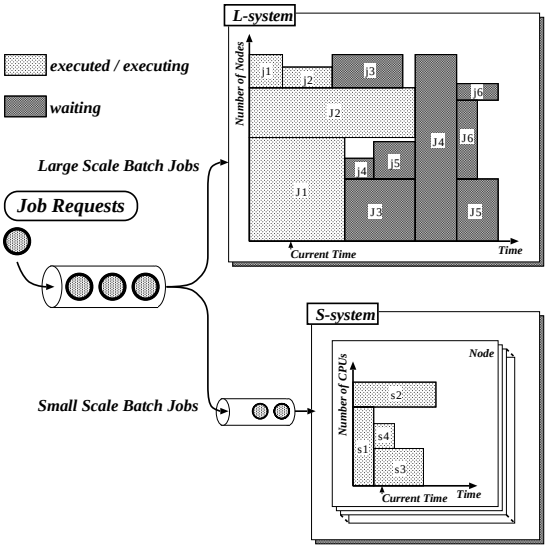

図 5 キュー構成

3. ジョブスケジューラ

ES はバッチジョブシステムであり，ES を効率良く運用するためにはジョブスケジューラの役割は重要である．ES 用ジョブスケジューラは以下のポリシーを持って開発された．

- ユーザはノードを占有する
- CPU 時間ではなく，経過時間でジョブをコントロールする

ES では，シングルバッチキューシステムが採用されている (図 1)．キューに投入されたジョブは，使用するリソース (経過時間，ノード数，ディスクスペース等) に従ってスケジューラにより実行順序が制御される．L 系では 624 ノードまでのジョブを実行することができるが，ユーザのジョブは最大 512 ノードまでに制限されている．割り当てられたノードはユーザが占有し，他のユーザのジョブの影響をうけることはない．

ジョブが使用するファイルは，基本的にデータストレージに格納されていて，ジョブの実行前に自動的に L 系のワークディスクへ転送される．そのため，データ転送のスループットがジョブのスケジューリングに大きく影響を及ぼすことになる．ES 用ジョブスケジューラは，データ転送がスケジューリングに及ぼす影響が最小限になるように設計されている．

ES 用ジョブスケジューラの主な処理を以下に示す．

3.1 リコール

ジョブが投入されると，ジョブスケジューラはジョブが使用するファイルを L 系のワークディスクへ転送する．これをリコールと呼ぶ．スケジューラは，ジョブが使用するディスクスペースを L 系のワークディスクに確保し，データストレージからファイルを転送する．この時，転送先のワークディスクはジョブを実行するノードとは独立に決定される．

3.2 スケジューリング

リコールが終ったジョブはスケジューリングの対象となる．ジョブの優先度は，ユーザの宣言経過時間と使用ノード数から決定される．ジョブが必要とするリソースが少ない程，優先度は高くなる．

ジョブスケジューラは優先度の高いジョブから順次スケジューリングを行う．リコールに使用されたディスクのあるノードから順次割当を行うが，全てのリコール済のノードをジョブに割り当てることができなかった場合には，他の空いているノードを割り当てる．この場合，リコール先のディスクから割り当てられたノードへファイルの移動を行う．この時，転送に必要な時間を計算し，ジョブの実行開始予定時刻までに移動が終了しない場合にはそのノードは割り当てない．

3.3 ファイル移動

ジョブスケジューラは，ノードスケジューリング時にユーザファイルが転送されていないノードを割り当てられた場合，そのノードへファイルを移動する．この移動は IOCS の LAN を経由して行われる．

3.4 ジョブ実行

ジョブが必要とするノードが全て確保され，そのノードのワークディスクにファイルが全て準備できたら，ジョブスケジューラはそのジョブを実行する．ジョブスケジューラはジョブの経過時間を監視し，ユーザの宣言経過時間を超過した場合には，そのジョブを強制終了させる．

3.5 マイグレーション

ジョブスケジューラは，ジョブが終了すると，ノードを解放し，ワークディスク上のファイルをストレージシステムへ転送する．これをマイグレーションと呼ぶ．この時，ジョブの実行後に更新されたファイルだけがマイグレーションの対象となる．

カートリッジテープライブラリを使用した現在のシステムでは，マイグレーション処理はリコール処理よりも優先度が低くなっている．これは，マイグレーション処理によりリコール処理が後回しになり，ジョブの実行効率が低下するのを防ぐためである．

4. ジョブ特性

ジョブのスケジューリングは，実行されるジョブの特性に大きく影響を受ける．とくに ES では，経過時間をベースにジョブのスケジューリングを行っているため，ユーザの宣言経過時間と実際に使用された実経過時間の関係は重要である．また，ジョブが使用するファイルをデータストレージから転送するため，ファイルサイズも重要な要素となる．

スケジューリングアルゴリズムの評価を行うにあた

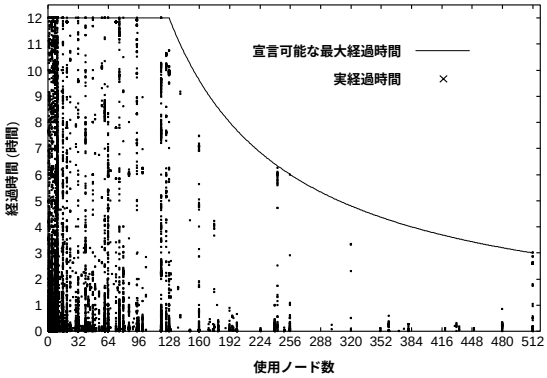

図 6 使用ノード数と経過時間

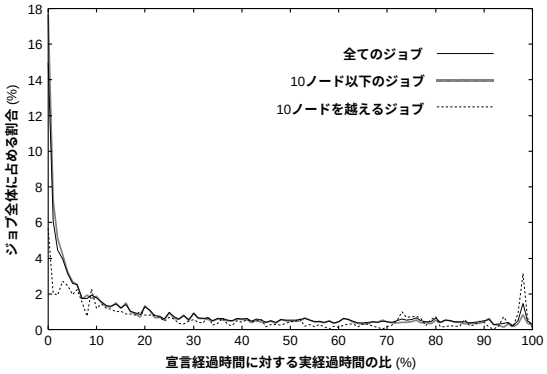

図 7 宣言経過時間と実経過時間の比

り，ES 上で実行されているジョブについて調査を行った．6ヵ月間に ES 上で実行された 45,273 個のジョブを対象とした．これにはハードウェアの障害で実行が中断されたものは含まれていない．

4.1 使用ノード数と経過時間

まず，ジョブで使用するノード数と経過時間について調査した．図 6 に使用ノード数と実経過時間の関係を，図 7 にユーザの宣言経過時間と実経過時間の比をそれぞれ示す．

図 6 から 10 ノード以下のジョブが，大部分を占めていることがわかる．これは，ES では使用開始時には 10 ノードまでしか利用することができないという制限があるためである．10 ノードを越える並列ジョブを実行するためには，ユーザは 10 ノード以下でプログラムを十分にチューニングしなくてはならない．そのため，経過時間の短い 10 ノード以下のジョブが多くなっていると考えられる．10 ノードを越えるジョブには，特定のノード数で実行されるものが多い．これは，大気モデルや海洋モデル等の特定のジョブで，プロダクトランを行っているジョブである．

図 7 から，大部分のジョブは宣言経過時間よりもかなり短い時間で終了していることがわかる．とくに 10 ノード以下のジョブの場合，チューニングを目的とした実行が多いためか，宣言時間の 10%未満で終了するジョブが多い．一方，10 ノードを越えるジョブの場合，宣言時間の 70%を越えるジョブが 10 ノード以下のジョブの場合と比べて多くなっている．

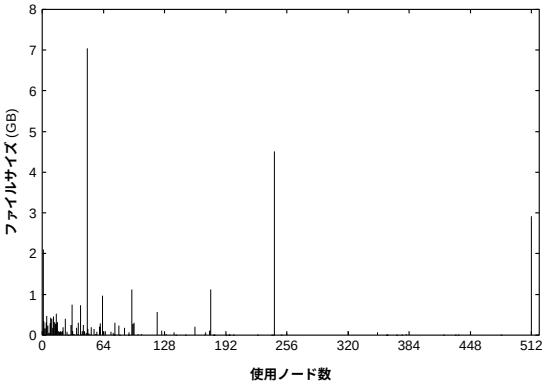

図 8 ノードあたりのファイルサイズ

ES では，ジョブのスケジューリングアルゴリズム上，宣言経過時間が短い方がジョブが実行されるまでの時間が短くなる．また，ノード数と宣言経過時間の積での実行制限もあるため，大規模ジョブを効率良く実行するには，実行可能な最大経過時間ぎりぎりまで計算を行う必要がある．プロダクトランを行っているユーザは，これらのことを理解し，効率よくジョブを実行しているものと思われる．

4.2 ファイルサイズ

次に，ジョブで使用するファイルサイズについて調査した．図 8 にジョブの使用ノード数とノードあたりのファイルサイズを示す．ここでは，リコールとマイグレーションの対象となるファイルについて集計している．図からわかるように，使用するノード数とファイルサイズの関連は低く，アプリケーション依存になっている．

以上のことから，ES 上で実行されているジョブには，宣言経過時間と実経過時間に大きな差があり，使用するノード数にも偏りがあることがわかる．とくに，大規模ジョブは特定のアプリケーションに偏っており，その多くはプロダクトランを行っている．ES 用ジョブスケジューリングアルゴリズムの評価では，これらの特性を考慮したジョブを使用する．

5. スケジューリングアルゴリズム評価

我々は，ES 用のジョブスケジューリングアルゴリズムを評価するために，ソフトウェアジョブシミュレータを開発している[3]．ジョブスケジューリングアルゴリズムの評価にあたり，ES で実行されているジョブ特性を考慮したジョブを使用した．表 1 に使用したジョブの特性を示す．宣言経過時間と実経過時間の関係は 4.1 の調査結果を用いた．

表 1 評価で用いたジョブの特性

ジョブ種	ノード数	実行時間	入力 (MB)	出力 (MB)	投入割合
J0	1 – 4	1 sec – 12 hour	353 – 4,235	247 – 2,960	45.3%
J1	5 – 10	1 sec – 12 hour	354 – 2,121	581 – 3,483	31.5%
J2	11 – 64	1 sec – 12 hour	557 – 9,581	920 – 15,962	14.8%
J3	65 – 128	1 sec – 12 hour	4,591 – 26,074	7,109 – 41,102	5.9%
J4	129 – 256	1 sec – 9 hour	14,383 – 83,460	10,641 – 57,920	1.9%
J5	257 – 512	1 sec – 4 hour	71,523 – 389,545	127,120 – 712,713	0.6%

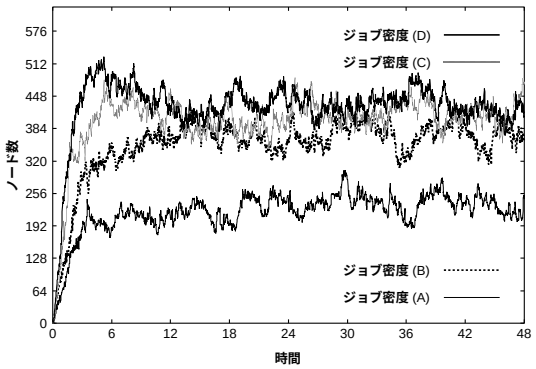

図 9 テープライブラリを使用した場合のノード使用率

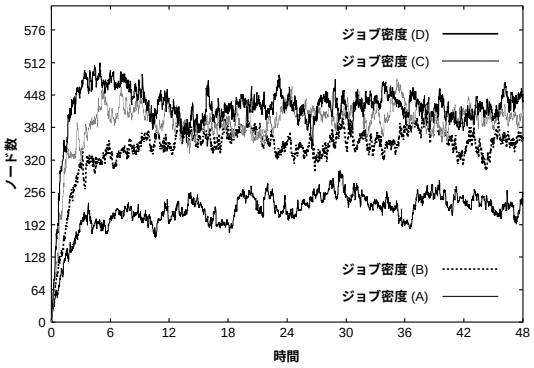

図 10 ハードディスク装置を使用した場合のノード使用率

表 2 ノード使用率 (%)

ジョブ密度		MT 48H	MT (24H)	HD 48H	HD (24H)
(A)	37.74	35.13	(37.86)	35.20	(37.91)
(B)	68.39	56.20	(59.69)	55.18	(57.30)
(C)	105.96	63.82	(66.74)	63.19	(65.73)
(D)	144.10	68.45	(69.05)	67.75	(67.97)

ジョブスケジューリングアルゴリズムを評価するにあたり，ジョブの実経過時間がノード時間に占める割合 (ジョブ密度) が (A)37.74%，(B)68.39%，(C)105.96%，(D)144.10%の 4 通りの場合についてシミュレーションを行った．システム全体が安定する 48 時間までをシミュレーション時間とし，乱数による揺らぎを補正するため，30 回実行した結果の平均値を求めた．

ユーザ側から見たアルゴリズムの評価指標としてジョブ待ち時間を定義する．

ジョブ待ち時間 $= Finshed - Submited - CPU$

ここで，$Finshed$ はジョブがマイグレーションまで完了した時刻，$Submited$ はジョブが投入された時刻，CPU はジョブの実経過時間をそれぞれ表している．この値が小さい程，ユーザからみたジョブのターンアラウンドタイムが小さくなる．

表 2 に 48 時間および後半の 24 時間におけるノード使用率を示す．また，図 9 にデータストレージとしてテープドライブを使用した場合のノード使用率の時間変化を，図 5 にハードディスク装置を使用した場合のノード使用率の時間変化をそれぞれ示す．ノード使用率からみた場合，テープドライブを使用した場合とハードディスクを使用した場合との間に大きな差は見られない．

ジョブ密度が低い場合，投入されたジョブはほぼ全て実行されている．ジョブ密度が高くなるにつれ，実行されたジョブ数は減少している．これはユーザの宣言経過時間と実経過時間との差に起因するものと考えられる．ES 用ジョブスケジューラはユーザの宣言経過時間をベースにスケジューリングを行う．そのため，ジョブが宣言経過時間よりも早く終了すると，その部分にスケジュールの空きができてしまう．ジョブスケジューラは，可能な限りスケジュール済のジョブを再配置したり，未実行のジョブをスケジュールすることでこのスケジュールの空きを埋めようとする．ジョブの密度が低い場合は先に終了したジョブの影響を受けるジョブは少ないが，ジョブの密度が高い場合には影響を受けるジョブ数は多くなり，その影響を隠すことが難しくなっているものと考えられる．また，大規模ジョブ (ジョブ種 J4，J5) の実行時には，スケジューリング時の隙間が大きくなる傾向がある．ジョブ密度が高い程この現象は顕著に現れるため，ノードの使用効率が低くなる原因にもなっている．

表 3 に，ジョブ種毎の平均待ち時間を示す．この表には，シミュレーション時間内に終了しなかったジョブは含まれていない．実行ノード数の少ないジョブの場合，ディスクシステムの場合の待ち時間がテープドライブの場合よりも短くなっている．これは，実行

表 3 ジョブ平均待ち時間 (sec)

	ジョブ密度 (A)		ジョブ密度 (B)		ジョブ密度 (C)		ジョブ密度 (D)	
ジョブ種	MT	HD	MT	HD	MT	HD	MT	HD
J0	426	73	433	76	487	79	631	83
J1	485	83	586	208	828	327	1,317	725
J2	1,050	306	2,521	1,927	4,736	3,848	6,740	5,620
J3	1,520	466	4,210	3,428	7,578	6,940	10,861	9,456
J4	2,479	1,268	6,890	5,335	13,776	10,970	15,324	12,553
J5	8,768	9,799	21,796	22,844	36,771	33,542	41,427	40,104

表 4 ジョブ処理状況 (%)

	ジョブ密度 (A)		ジョブ密度 (B)		ジョブ密度 (C)		ジョブ密度 (D)	
ジョブ種	MT	HD	MT	HD	MT	HD	MT	HD
J0	98.36	98.28	63.59	63.55	46.17	45.91	35.25	35.12
J1	98.00	97.92	62.93	62.45	44.86	44.71	33.98	34.22
J2	96.27	96.54	63.00	62.99	43.67	43.82	33.01	33.12
J3	94.74	96.03	61.17	61.35	41.54	41.72	31.01	31.02
J4	95.72	96.50	58.58	59.63	37.46	38.45	28.16	28.08
J5	90.00	87.78	52.59	52.96	31.91	32.10	18.95	19.84

ノード数が少ない場合には，ディスクのスループットがテープドライブのスループットより大きくなるためである．一方，実行ノード数が多くなると，テープドライブのスループットの方が大きくなり，ディスクドライブ使用時の待ち時間との差は小さくなっている．

テープドライブ使用時には，リコール処理が優先されマイグレーション処理が後回しになる．そのため，ジョブ密度が高くなるにつれ，ジョブ待ち時間が増大する．この現象は，実行ノード数が多いほど顕著に現れる．ディスク使用時には，リコール処理とマイグレーション処理は同時に行われているため，一時的にワークディスクとユーザディスク間のスループットは落ちるものの，テープドライブ使用時程待ち時間は増大しない．

表 4 に投入されたジョブ種毎の処理状況を示す．ディスク使用時とテープドライブ使用時のジョブ処理状況に大きな差は見られない．ジョブの待ち時間に差はあるが，ジョブの処理状況には差がないことからも，ジョブシュケジューラがリコールやマイグレーションのスケジューリングへの影響を最小限に抑えていることがわかる．使用ノード数が多いジョブ種 J4 や J5 は，ジョブ密度が高くなるにつれ他のジョブ種に比べて実行されにくくなってはいるが，全く実行されない状況になっているわけではないこともわかる．

以上から，ES 用ジョブスケジューリングアルゴリズムはジョブ密度が高くなっても投入されたジョブを偏り無く処理することができるといえる．テープドライブ使用時とハードディスク使用時のノード利用率には大きな差は無いが，ディスクドライブを導入することにより，ジョブのターンアラウンドタイムは小さくできることがシミュレーションから判明した．現在のテープドライブを使用したシステムでは，外部へ処理した結果を出力するためにさらに時間が必要になることを考えると，ユーザの利便性は，ディスクドライブを導入することにより大幅に向上するものと予想される．

6. おわりに

本稿では，地球シミュレータ用ジョブスケジューリングアルゴリスムの評価を行った．アルゴリズムの評価には，地球シミュレータで実際に実行されているジョブの特性を考慮したジョブを使用した．シミュレーションによりノード時間に対する実行ジョブの実経過時間が約 100%の場合，ノード使用率は 63%程度になることがわかった．

また，現在の地球シミュレータのデータストレージシステムにディスクドライブ装置を導入した場合のスケジューリング性能についても評価を行った．シミュレーション結果からノードの使用率については，現システムと大きく変化することはないが，ユーザ側からみたジョブのターンアラウンドタイムの改善と，ユーザファイルの利便性の向上が期待できることがわかった．

謝辞 本報告を発表する機会を与えてくださった佐藤 哲也 地球シミュレータセンター長に感謝致します．また，本研究を進めるにあたり，日頃から御討論頂く地球シミュレータセンターの諸氏に感謝致します．

参考文献

1) 谷 啓二, 横川 三津夫：地球シミュレータ計画, 情報処理, Vol 41, No.3, pp.249–254 (2000).
2) 横川 三津夫, 谷 啓二：地球シミュレータ計画, 情報処理, Vol 41, No.4, pp.369–374 (2000).
3) Atsuya Uno, Tatsuya Aoyagi, Keiji Tani : Job-scheduling on the Earth Simulator, *NEC Research & Development*, Vol. 44, No. 1, pp. 47–52 (2003).