

実時間グリッド計算のための資源割当システムの設計 Design of Resource Allocation System for Real-Time Grid Computing

伊野 文彦[†]
Fumihiko Ino

土屋 博之[‡]
Hiroyuki Tsuchiya

萩原 兼一[†]
Kenichi Hagihara

1. はじめに

グリッド技術とは、異なる組織に属する複数の計算機環境を統合し、仮想的な1つの高性能計算機環境（グリッド）として動作させるための技術である。グリッドの形態は、計算機センタなどで常時運用しているPCクラスタを基にするサーバグリッドや、一般家庭や企業内LANなどで遊休状態にある不特定多数のPCを基にするデスクトップグリッド[1]まで様々である。

本研究が対象とするグリッドの形態は後者である。特に、企業内LANに接続する数百台以上からなるPC群において、遊休状態にあるPC（ノード）を用い、数十秒以内で実時間処理できる仕事（ジョブ）を対象とする。この実時間グリッド計算を実現するためには、(1) 遊休状態にあるノードを発見し、(2) 適切なノードへジョブを割り当てるなどを、少なくともジョブの実行時間よりも短い時間で終えることが必要である。

しかし、既存の資源割当システムは数時間から数日を要する大規模アプリケーションを対象としていて、実時間処理を考慮した設計が欠落している。例えば、グリッドにおける単位時間当たりの処理能力の向上を目的とするもの[2]や、汎用性の向上を重視するもの[3]などが挙げられる。これらのシステムでは、ユーザからのジョブを受け付けてから(2)を終えるまでの応答時間が長く、ジョブを実時間処理できる可能性が低い。この問題は、ノード数の増大とともに顕著になる。例えば、100ノードにおいて30秒以上の応答時間を要する[4]。さらに、応答時間の増大は、ノードが遊休状態であるか否かという情報が新鮮でなくなることを意味するため、遊休状態でないノードにジョブを割り当てる可能性が高くなる。

本稿では、これらの問題を解決することを目的として、デスクトップグリッドにおいて実時間処理を実現するための資源割当システムの設計を示す。提案システムは応答時間を短縮するために、分散処理により資源評価を高速化し、通信量を削減する。この工夫により、実時間処理に堪える応答時間を実現し、より新鮮な情報に基づいてノードにジョブを割り当てるなどを狙う。

2. 提案する資源割当システム

提案システムのアーキテクチャを図1に示す。提案システムが既存システムと異なる点は、資源評価部が資源割当部から独立し、各ノード内に存在する点である。ここで、資源評価部はそのノードが遊休資源としてどの程度適合するかということを評価値として判定する。各ノードが資源評価を担当することにより、以下に示す2つの利点がある。

- 資源評価の高速化：各ノードが資源評価を並列処理

[†]大阪大学大学院情報科学研究科コンピュータサイエンス専攻
[‡]東芝情報システム（株）

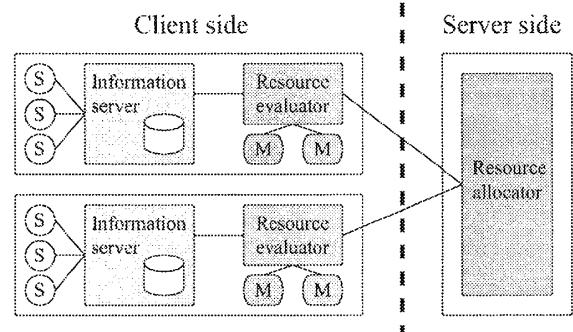


図1：提案システムのアーキテクチャ（SおよびMは、各々センサおよび評価モジュールを表す）

するため、高速化が期待できる。また、サーバの計算負荷を軽減できるため、ノード数の増大に起因する性能低下を抑制できる。特に、過去の測定データに基づいて計算資源を割り当てる場合、そのための計算負荷を分散できることは重要である。

- 通信量の削減：センサが蓄積するCPU利用率や主記憶使用量などの測定データをサーバへ集める必要がないため、通信量を削減できる。通信量の削減は応答時間の短縮のみならず、グリッド計算のためのネットワーク帯域を温存できる点からも有益である。

図2(a)に、提案システムにおける処理の流れを示す。

1. 評価方法の指定：サーバは、計算資源を選択するために用いる評価モジュールのIDを各ノードへ送信する。
2. 計算資源の評価：各ノードは、受信したIDに対応する評価モジュールおよびセンサが蓄積する測定データに基づいて、各々の遊休状態を評価する。
3. 評価値の収集：各ノードは、計算した評価値をサーバへ送信する。
4. 計算資源の選択：サーバは、受信した評価値を降順に整列し、その先頭から順にジョブを割り当てる。

既存システムにおける処理の流れ（図2(b)）と比較すると、各ノードがサーバの替わりに計算資源を評価していることが分かる。

3. 評価実験

本章では、実機を用いて提案システムの応答時間を評価した結果を示す。実験に用いたPCは、CPUとしてPentium II 450MHzを持ち、通信バンド幅100Mb/sのイーサネットで相互接続している。

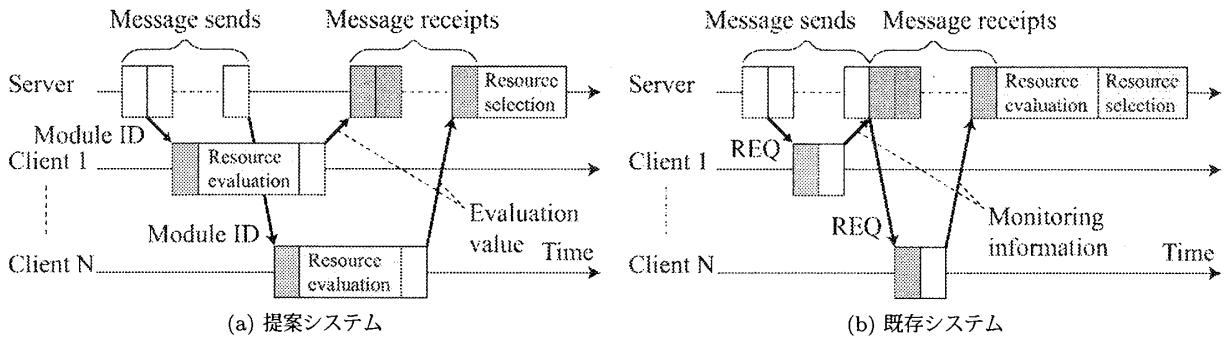


図 2: 資源割当のための処理の流れ

実験では、ノード数および通信量を変化させ、応答時間の特性を調べた。表1に、測定した応答時間を示す。ここで、 D_n は過去 n 回の測定データを参照して計算資源を選択することを表し、1回あたりの測定データはCPU利用率、空き主記憶量や空きディスク量などの情報を含む33バイトからなる。

表1より、参照データの増大とともに応答時間を削減できている。過去600回の測定データを参照する場合、応答時間をおよそ30分の1に短縮できている。したがって、過去の測定データに基づいて計算資源を割り当てる場合、参照するデータのサイズが大きいほど、提案システムによる改善効果は大きい。ゆえに、過去の測定データに基づいてより適切な計算資源を選択するために、提案システムは有用である。

表2に、8ノードかつ D_{10} のときのサーバにおける応答時間の内訳を示す。既存システムでは、メッセージ受信および計算資源の評価が応答時間のおよそ96%を占めている。一方、提案システムでは測定データの替わりに評価値のみをサーバに集めるため、通信量を $33 \cdot n$ から4バイトに削減でき、メッセージ受信のための時間を64.8から4.3ミリ秒に短縮できる。さらに、各ノードが計算資源の評価を担当するため、サーバにおける評価のための時間25ミリ秒を除去できている。なお、各ノードにおける評価のための時間はおよそ8分の1(3ミリ秒)であった。このように、提案システムにおける工夫が迅速な応答時間を実現している。

4. まとめ

本稿では、LAN環境における遊休資源を用いて実時間グリッド計算を実現するための資源割当システムの設計を示した。提案システムは、分散処理により資源評価を高速化し、通信量を削減する。提案システムを8台のPCを用いて評価した結果、既存システムと比較して応答時間を最大で30分の1に削減できた。過去の測定データに基づいて計算資源を割り当てる場合、より適切な計算資源を選択するために、提案システムは有用である。

今後の課題としては、より大規模な環境における性能評価が挙げられる。

表1: 応答時間の比較（単位はミリ秒、 D_n は過去 n 回の測定データを参照することを表す）

ノード 数	提案システム			既存システム		
	D_{10}	D_{60}	D_{600}	D_{10}	D_{60}	D_{600}
1	2.4	2.6	5.3	10.0	15.3	64.5
2	2.5	2.8	5.6	19.7	27.3	93.4
3	4.7	5.2	14.8	31.6	65.8	243.0
4	7.3	7.9	16.4	46.0	70.9	282.4
5	7.5	8.5	17.9	55.7	86.1	363.9
6	7.8	9.6	17.7	68.9	99.8	442.8
7	7.9	9.8	19.8	81.5	113.2	457.6
8	8.8	11.0	19.9	93.4	119.4	634.6

表2: サーバにおける応答時間の内訳（単位はミリ秒）

内訳	提案システム	既存システム
メッセージ送信	1.1	1.1
受信待ち	3.2	2.3
メッセージ受信	4.3	64.8
計算資源の評価	—	25.0
計算資源の選択	0.2	0.2
計	8.8	93.4

参考文献

- [1] Chien, A., Calder, B., Elbert, S. and Bhatia, K.: Entropia: architecture and performance of an enterprise desktop grid system, *J. Parallel and Distributed Computing*, Vol.63, No.5, pp.597–610 (2003).
- [2] Raman, R., Livny, M. and Solomon, M.: Matchmaking: An extensible framework for distributed resource management, *Cluster Computing*, Vol.2, No.2, pp.129–138 (1999).
- [3] Czajkowski, K., Fitzgerald, S., Foster, I. and Kesselman, C.: Grid Information Services for Distributed Resource Sharing, *Proc. 10th IEEE Int'l Symp. High Performance Distributed Computing (HPDC'01)*, pp.181–194 (2001).
- [4] Zhang, X., Freschl, J. L. and Schopf, J. M.: A Performance Study of Monitoring and Information Services for Distributed Systems, *Proc. 12th IEEE Int'l Symp. High Performance Distributed Computing (HPDC'03)*, pp.270–282 (2003).