L-013

RTTを用いたグリッドシステムにおける計算資源配分の改善

Improvement of Computational Resource Allocation based on RTT in Grid System

小山 敦† 長坂 康史†
Tsutomu Koyama Yasushi Nagasaka

1. はじめに

近年、コンピュータの発展に伴い、ヒトゲノム解析やゲノム創薬、分子科学計算、加速器科学などに代表される、大容量ストレージや強力な計算能力が必要とされる研究・開発が増えてきた。高額なスーパーコンピュータなどの導入に代えて、既存の散在するマシンパワーを集約し、遊休資源を有効に活用することにより、それに匹敵するパフォーマンスを得ることができる技術、グリッドコンピューティングが注目され、研究・開発されている。

このグリッドコンピューティングはシステムの概念としてクラスタと似た物であるが、実装の方法が異なっている。通常グリッドコンピューティング環境を構築する際には、OS 等が異なった環境へ用いることが前提となる。そのため、異機種間で適用可能なミドルウェアが用いられるが、現在業界標準として利用されているミドルウェアとして、Globus Toolkitがある。Globus Toolkitは1995年に現 Globus Alliance 社によって開発が開始され、現在に至るまで様々なツールやライブラリが開発されてきた。

グリッドコンピューティングが我々の身近な環境で利用される状況として、各種センサーから動画や計測値など大量の観測データを収集する分散型気象観測システムへの適用が考えられる。この際には、ユーザの要求程度にもよるが、システムが統計的データを提供するには年単位のデータを処理する必要がある。そのためには、グリッドの機能によって処理を分担化し、処理待ち等によるオーバーヘッドを軽減することで、システムの性能を向上させることが出来る。しかし、RTT を考慮した場合、タスク振り分けにおいて処理効率を高める手法が求められる。

本論文では、既存のグリッドシステム上で RTT を考慮した場合の、処理分散におけるタスク配分を最適化する手法についての説明と実装方法について述べる。

2. システムの前提

通常、多地域気象観測システムでは多数のセンサーの配置を必要とし、気象庁のアメダスを例にとると、1300もの観測地点を設けている。本システムでは処理ノード数が増えすぎると、処理ノードの情報管理が煩雑になってしまうため、センサー数が数百、処理ノードが数十程度の規模を想定している。データは気温などの数値を5分毎に取得し、センサー管理マシンに保存する。データの大きさは1年で数百 MB 程度となる。データの受け渡しはユーザからの要求が発生した時点のみで行われる。

本研究ではグリッド環境上にシステムを構築している。グリッドシステムを用いるメリットは、資源情報の管理機能を使うことにより、各マシンのリソース状況を把握し、リソースに空きのあるマシンに処理を依頼することができる。ここでいうリソースとはそのマシンの持つ遊休状態の計算資源のことである。環境構築に使用するミドルウェアとしては、GlobusToolkitを用いている。

†広島工業大学大学院 工学研究科

3. RTT考慮による処理分散の効率化

従来のグリッドシステムでは、各処理マシンのリソースを管理するだけで、ホストマシンからセンサーまでの往復遅延時間(RTT)は考慮されていない。そのため、処理マシンとセンサー側の組み合わせによっては、RTT が大きくなり、スループットが低下してしまう。これまでの研究では RTT を考慮することにより若干の性能の向上が見られることが分かっているが[1]、リソースの余裕と RTT のどちらを重視するかはユーザの要求によって異なってくる。本研究における処理分散環境を図 1 に示す。1つのデータ収集処理につき、複数のセンサーからのデータ収集が必要な場合、個別のタスクにおいて、RTT が最も小さくなるように処理マシンを選ぶことにより、スループットを向上させることができる。

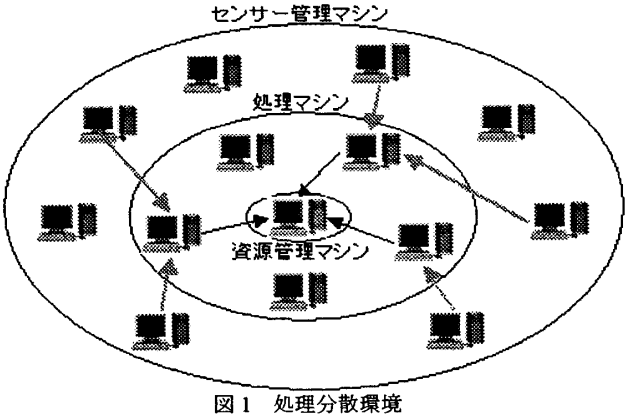

図1 処理分散環境

しかし、タスクの組み合わせが偏る場合、処理の集中するマシンでは処理待ち時間が長くなり、全体のスループットに影響を与えることになる。

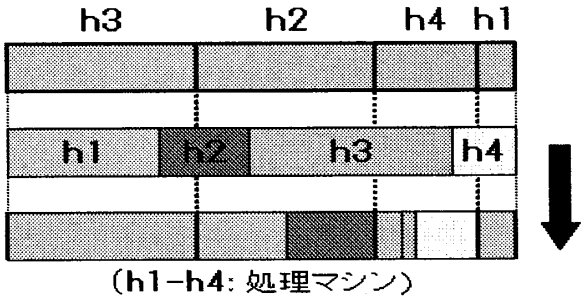

図2 タスクの振り分け

そこで本研究では、図2のように管理情報を元にしたタスクの振り分けを行った。図はタスク振り分けの手順を示し、各バーの長さはタスクの数を表しており、下方向

に向かって処理が進行む。まず最初に処理マシンの遊休資源比率に比例してタスク割り当て数を決定する。次いで各タスクと、RTT が最小となるような処理マシンの組を決定し、処理マシン毎にグループ化する。そして、第 3 手順では、タスクの最も少ない物から該当グループのタスクを割り当てていく。溢れたタスクが発生した場合、少ないものから順に、まだ処理タスクに空きがあり、割り当て数の小さい処理マシンから割り当てる。

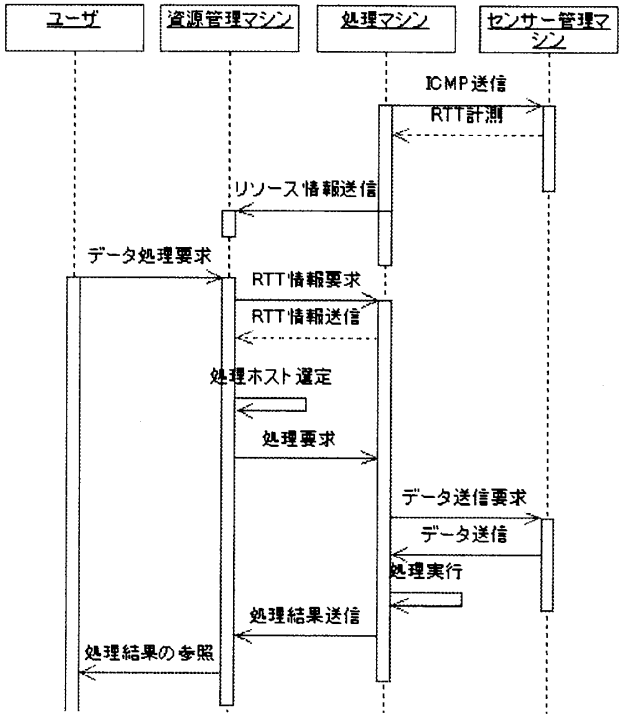

図 3　処理の流れ

4. 性能評価のためのテストシステムの構築

4.1 システムの概要

本システムでの主要な処理の流れを図 3 に示す。独立した処理として、処理マシンが 1 秒毎の RTT 取得、20 秒ごとの自身のリソース情報取得を行い、毎回資源管理マシンに情報を送信する。この処理はあまり短い時間間隔で繰り返されると、アプリケーション処理がマシンにかける負担が大きくなってしまうので上記のように設定した。次にユーザの要求が発生すると、資源管理マシンでは 2 段階の処理が行われる。まずユーザから受けた処理内容をもとに、複数の地点の気象情報が必要であるか否かにより、処理を分散させるかどうかを判断し、その時点までに受け取った処理マシンの管理情報を元に、前章で述べた手法により処理を分散させる。処理マシンは受け取ったタスクに従って目的のセンサー管理マシンのデータを集めて資源管理マシンにデータを送信する。資源管理マシンは、各処理マシンから集められたデータを元に統計計算を行い、結果を保存する。処理結果が出ると、データベースに保存され、ユーザに通知される。ユーザは WEB 上で処理結果を見ることが出来る。

GlobusToolkit は WSRF 標準の提案を受け、Ver.3.2 から Ver.4.0 に移行すると共に、Web サービス上での構築を前提とするようになった。Web サービスを用いることで、システムを一般に公開し、利用しやすく出来る。本研究でも将来の利用性を考えて Web サービス技術を用いることとした。これにより、ユーザは処理の依頼や、処理結果の参照を、Web 上で行うことができる。

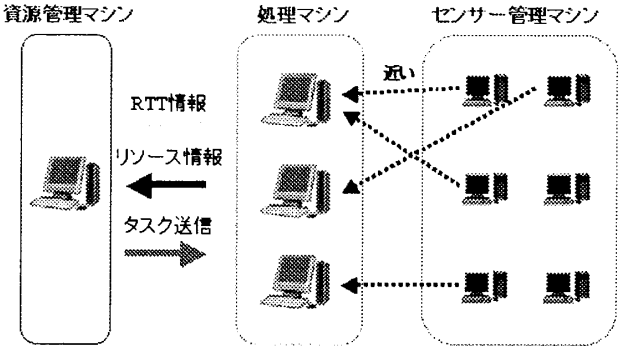

図 4　システム構成

本システム(図 4)は資源管理マシン 1 台、処理マシン 3 台、センサー管理マシン 6 台で構成される。資源管理マシン、処理マシンは全て同じ機種であり、OS が FedoraCore4、CPU が Pentium3(1.40GHz)、HDD が 40GB である。センサー管理マシンは HDD が 80GB で OS に Linux を用いている。

4.2 システムの処理の流れ

まず、中央管理マシンおよびホストマシンではコンテナと呼ばれるアプリケーションが起動することにより、定期的に中央管理マシンにリソース情報が集められる。リソースと RTT 情報(合わせてマシン情報とする)を取得すると、Web サービス上で情報を公開することにより、ユーザもその値を確認できる。マシン情報を取得し終わると、まず選定可能な処理能力を持つホストマシンが選ばれる。そして、ユーザの要求した処理の分割を行った後、各ホストマシンの RTT 情報を考慮の上、処理に必要なデータをセンサーから取得するための最適なホストマシン選定が行われる。選定が終了すると処理がホストマシンに渡され、センサーからデータを収集する。このデータに必要な処理を加えて、結果を中央管理マシンに送信する。結果を受けた中央管理マシンはデータを WEB サービスに公開する。すべての処理が終了した時点でユーザは情報を参照することができる。この処理情報はデータベースに格納され、WEB 上で何処からでも利用できる。

5. まとめ

本論文では、これまで研究を行ってきた、グリッドミドルウェアによるグリッドシステム上での、RTT の考慮によるホストマシン選定補助に加えて、さらにシステムの処理効率を高める方法として処理の複数マシンへのタスク配分手法について 3 章で述べた。処理が複雑で大規模になるほどホストマシンへの負担は増加するため、処理負荷の分散は必要である。ネットワーク経路の差と共に処理分散処理を最適化することにより、グリッドシステムの処理効率を向上することが出来ると考え、テストシステムを構築した。

今後はシステムの性能評価のためのテストを行うことによって実際の可用性について確認する。

参考文献

[1] 小山敦：“グリッドシステムにおけるコンピューティングリソース割り当て手法への RTT 導入の有効性の検証”、第 6 回情報科学技術フォーラム、2007/09/05