1T-2

Gridにおけるネットワーク負荷を考慮したタスク分散システムの設計と実装

橋本 浩二† 西山 裕之† 溝口 文雄†
所 属† 東京理科大学理工学部経営工学科

1 はじめに

近年，グリッドコンピューティング（以下，グリッド）という技術が注目を集めている．これは，ネットワークを通じて複数の計算機を接続し，メモリ・ストレージ・CPUといった資源を共有することで，仮想的に１つの高性能コンピュータとして扱うことを可能にする技術である．

グリッドではSETI@home[1]などのように，オフィスや家庭に散らばる遊休状態にある計算機の資源を共有し，高性能なコンピュータとして扱ことが一般的である．しかし，このプロジェクトのようにインターネットを通じて個人向けの計算機を利用することは，各計算機資源のネットワーク環境に差を出す結果となる．また，膨大なデータを異なる場所や計算機に分割して保存することで保存場所を確保し，それらのデータを共有して利用するデータグリッドと呼ばれるものがあるが，このような形態のグリッドの運用はネットワークの負荷に多大な影響を受ける．また，ネットワーク負荷は時々刻々と変化するため，静的な計測では対応できず，頻繁に計測を行うことによる動的な対応が求められる．このネットワーク負荷の計測・予測を行う研究は，Network Weather Service(以下，NWS)[2]などがあり，定期的にデータ転送を行うことによりネットワーク負荷を計測し，１ステップ先の負荷値を予測する．しかし，負荷計測の際にネットワークに負荷を与えてしまうという問題がある．

本研究では，上述のようなヘテロジニアスなネットワーク環境において，各計算機のグリッド参加時に負荷計測を行うことで，最適な計算機にタスクを投入する．さらにタスク投入時に各計算機の予定処理時間を求めることにより，動的に変化するネットワーク負荷に対応できるグリッドシステムの設計と実装を行った．

Design and Implementation of task distributed system that considers network load in Grid
Kouji Hashimoto† , Hiroyuki Nishiyama† , Humio Mizoguti† ,
†Industrial Administration , Faculty of Science and Technology, Tokyo University of Science

2 設計

2.1 システムの概要

本システムにはマスタとワーカがあり，マスタは全ワーカとのネットワークの負荷状況を把握，スケジューリング，システム全体の視覚化を行う．ワーカは与えられたタスクの実行を行う．図１．にシステムの概要を示す．システムは，特定のOSやマイクロプロセッサに依存することがなく，基本的にどのようなプラットフォームでも動作するという汎用性の高さから，Javaによって開発した．そのため，マスタ・ワーカ共にOSやスペックなどの制限はないが，Javaの実行環境がインストールされていることが条件となる．

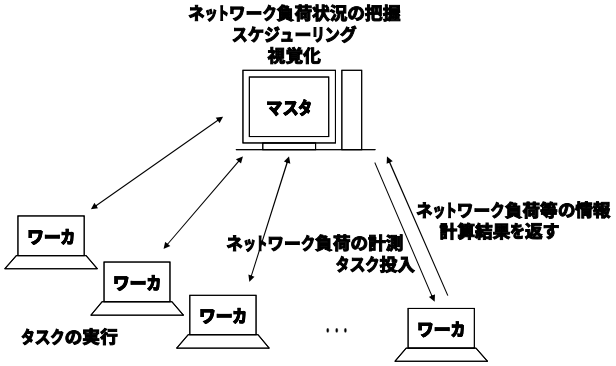

図１．システム概要

2.2 ネットワーク負荷の計測

本システムは，マスタがグリッドに参加している各ワーカとのネットワーク負荷を計測する．計測には，実際にデータを流すことにより測定する手法を取る．NWSでは，初期設定として64kbytesのデータを定期的に転送することにより，その遅延時間や経過時間を測定している．本システムにおいては8kbytesと1bytesの２種類のデータを送受信することにより，その経過時間を測定してネットワークの負荷を計測する．計測はワーカの初回接続時とタスク投入時に行い，計測によって発生するグリッドへのネットワーク負荷を減らし，グリッド実行の妨げにならないようにする．実際には，8kbytesのデータ転送に要した時間を

Btime, 1bytes のデータ転送に要した時間を Stime とし，ネットワーク負荷：NetworkLoad を次式により定める．

$$latency = \frac{Stime}{2}$$

$$EffectiveThroughput = \frac{DataSize(8kbytes)}{Btime - Stime}$$

$$NetworkLoad = \frac{latency}{EffectiveThroughput}$$

2.3 スケジューリング

ネットワーク負荷を考慮し，グリッドの運用を円滑に，最適にすることができるスケジューリングを行う．

2.4 視覚化

グリッドの現在の状況を瞬時に把握し制御するため，接続している計算機資源の情報や，各計算機間のネットワーク負荷の情報，タスク分散の情報を視覚化する．

3 実装

3.1 実装方法

2.2 によって得られたネットワーク負荷の値を用いてタスク分散を行う．ワーカの初回接続時に算出したネットワーク負荷を LF とし，LF とタスク投入の際に送受信するデータの量により送受信にかかる予定処理時間 PTime を求め，実際の経過時間 ETime との遅延 ΔT を求める．これらの値により，各ワーカの最新のタスク予定処理時間 PTime を求めることができる．実際には，前回までの処理時間の平均 $\overline{ETime}$ に，誤差を考慮する．次式に最新の PTime を求める式を定める．ここで n は遅延を求めた回数とする．

$$PTime = \overline{ETime} \pm \frac{\sum_{i=1}^{n} \Delta T_i}{n}$$

この予定処理時間 PTime が最小なワーカからタスクの投入を行う．

3.2 視覚化

本システム全体の現在の状況を把握・制御するため，3D による視覚化を行った．図２．にその外観を示す．球体でワーカを表し，OS を色で区別する．マスタからワーカへ伸びる線の太さでタスク量を表し，その色でネットワーク負荷を示す．

4 評価

評価はヘテロジニアスなネットワーク環境において，大容量のデータを各ワーカへ分散して保

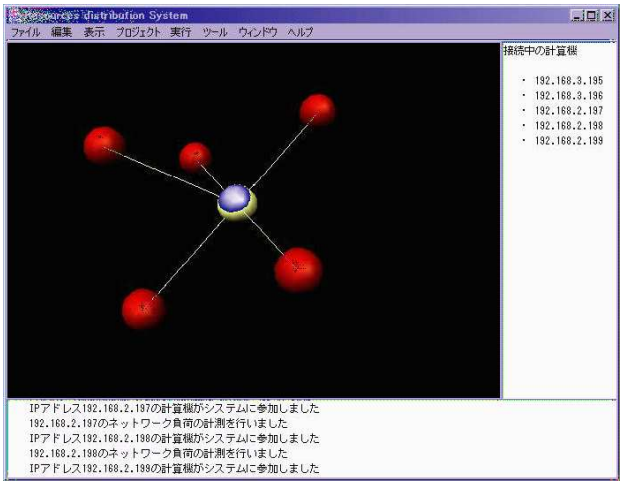

図２．システム外観

存するアプリケーションを実行し，その経過時間を計測することにより行った．比較対象は，120 秒間隔で定期的にネットワーク負荷を計測する方法とした．表１．に結果を示す．本研究のシステムは，比較研究のシステムよりも処理時間が短く，グリッドに対するネットワーク負荷も軽減できる結果を得た．

	処理時間（ミリ秒）
比較研究	26495
本研究	20649

表１．計算機５台による処理時間

5 おわりに

ネットワークがヘテロジニアスな環境において，ワーカの初回接続時にネットワーク負荷を測定し，マスタからのタスク投入時にワーカの予定処理時間を動的に算出し，その値が最小となるワーカからタスクを投入するシステムの設計と実装を行い，処理時間を短縮する結果を得た．

参考文献

[1]SETI@home http://setiathome.ssl.berkeley.edu/
[2]Rich Wolski, Neil Spring, and Jim Hayes, “The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing”, Journal of Future Generation Computing Systems, Volume 15, Numbers 5-6, pp. 757-768, October, 1999
[3]秋岡明香，村岡洋一，”Grid におけるネットワーク負荷予測”，計算機アーキテクチャ 147-5 ハイパフォーマンスコンピューティング 89-5, 2002.3.7