

## 3-5 Grid 環境におけるネットワーク特性がアプリケーションの実効性能へ及ぼす影響

### 3-5 Influence of Network Characteristics on Application Performance in a Grid Environment

北辻佳憲 田頭秀樹 山崎克之 鶴 正人 尾家祐二  
KITATSUJI Yoshinori, TAGASHIRA Hideki, YAMAZAKI Katsuyuki,  
TSURU Masato, and OIE Yuji

#### 要旨

多様なアプリケーションが並行して実行される Grid 環境では様々な Grid 資源が共用される。アプリケーションの性能を十分に引き出すために求められる様々な資源は、アプリケーションによって異なり、限られた資源の効率的な割当てが課題である。ネットワーク資源に着目すると、アプリケーションが通信を行うとネットワーク資源の状態が変化し、その変化がネットワークを共用する他のアプリケーションの通信に影響して、ときにはアプリケーションの実行性能が劣化する恐れがある。そこで、本報告では Grid で動作可能なアプリケーションのトラフィック特性を調査し、ネットワーク特性がアプリケーションの実行性能(実行時間)に与える影響を解析する。解析から、アプリケーションがネットワーク資源に与える影響とネットワーク資源の状態がアプリケーションの実行性能に与える影響が、共にアプリケーションによって大きく異なることを示す。そして、グリッドコンピューティングでの効率的なネットワーク資源の割当てには、個々のアプリケーションのネットワーク特性に対する実行特性を考慮することが不可欠なことを示す。

In grid computing, a key issue is how limited network resources can be shared by communications by various applications more effectively in order to improve application-level performance, e.g., by reducing the completion time for an individual application and/or set of applications. Communication by an application changes the condition of the network resources, which may, in turn, affect communications by other applications, and thus may degrade their performance. In this paper, we examine the characteristics of traffic generated by typical grid applications, and the effect of the round-trip time and bottleneck bandwidth on the application-level performance (i.e., completion time) of these applications. Our experiments show that the impact of network conditions on the performance of various applications and the impact of application traffic on network conditions differed considerably depending on the application. These results suggest that effective allocation of network resources must take into account the network-related properties of individual applications.

#### [キーワード]

Grid, ネットワーク資源, 実行性能, 分散コンピューティング, トラフィックエンジニアリング  
Grid, Network resource, Application-level performance, Distributed computing, Traffic engineering

## 1 まえがき

汎用計算機の高性能化と広域ネットワーク(WAN)の高速化に伴って、広域分散計算環境が驚く速さで整いつつある。WAN を経由して膨大

な数の計算機が接続されることで、飛躍的な計算性能を得ることができる。このような広域分散計算(Grid コンピューティング)環境では、地理的に離れた組織の間で多様な計算資源(CPU、メモリ、ストレージ、アプリケーションプログラム、

データなど)が様々な特性を持つネットワーク資源を通して接続され、要求に合わせて計算資源が動的に組み合わせられる[1][2]。Grid コンピューティングでは、限られた、または異質な計算機資源及びネットワーク資源を共有して複数のアプリケーションの全体の実行性能を向上させることが課題である。特に、離れた計算機間でアプリケーションを実行する場合、全体の実行時間に対してデータ転送にかかる時間が無視できないほど大きくなるため、複数の分散計算が同時に実行されるネットワークでは、限られたネットワーク資源の適切な配分が大きな課題である。

これまで、ネットワーク資源の利用効率を向上させるトラヒックエンジニアリング技術の研究は多く行われてきているが、アプリケーションの実行性能に着目した、ネットワーク資源の割当てに関する研究は少ない。例えば、[3]では、MPLSなどによる複数の通信経路(パス)が確立されたネットワークにおいてパスごとにアクティブ計測を行い、パスの輻輳状態に基づいてトラヒックフローを複数パスへ振り分ける手法が提案されている。また、[4]では、同じく複数パス環境においてパッシブ計測による可用帯域の計測を行って、その情報を基にトラヒックフローを振り分ける手法が提案されている。これらの研究は、フローの振り分けの目的がネットワークの利用率の向上であり、アプリケーションの通信性能や通信を行うアプリケーションの実行性能は考慮されていない。

一方、[5]では、パスに割り当てられた複数のTCP コネクションの全体のスループット(各ファイル転送時間の総和)を改善するため、同一のパスに確立されるTCP コネクションのスループットを等しくする方式が提案されている。さらに、[6]では、インタラクティブ性の高い分散計算アプリケーションを対象として、通信パスの遅延時間に基づくトラヒックフローの振り分けが提案されている。Grid コンピューティングでは、同時に実行される複数のアプリケーションのネットワーク資源の要求が多様であるため(例えば、遅延は問題とならず、広帯域パスや、逆に帯域は少なくとも遅延が短いパス)、これらの提案を適用した場合に、同時に実行されるアプリケーション全体の実行性能を向上できない場合が考えられる。

本報告では、複数の分散計算アプリケーションが同一のネットワークを共有し、各アプリケーションが計算パラメータを少しずつ変更して何度も実行される状況を考える。このとき、もし、アプリケーショントラヒックとネットワーク状態の関係が事前に明らかになっていれば、複数のアプリケーションを同時に実行するときに、全体の実行時間を最小にするネットワーク資源の割当てが可能と考えられる。アプリケーショントラヒックとネットワーク状態の関係には、アプリケーショントラヒックが発生することでネットワークの状態が変化する点と、ネットワーク状態が変化したときに他のアプリケーションのトラヒック特性が変化する二点がある。このようなアプリケーショントラヒックとネットワーク資源の関係が考慮されれば、計算資源の割当てスケジューリングに、ネットワーク資源の割当てスケジューリングを組み合わせることで、より効果的なGrid コンピューティングが実現できると考えられる。

このような問題に取り組むため、Grid で動作可能なアプリケーションについて、アプリケーショントラヒックとネットワークの状態の相互の影響を解析する。特に、様々なネットワーク資源の状態がアプリケーションの実行性能へ及ぼす影響について分析する。これらを基に、複数の分散計算アプリケーションを同時に実行するGrid 環境においては、アプリケーションのトラヒックパターン、ネットワークの状態、実行特性の関係を明らかにすることが重要であることを示す。

## 2 分散計算アプリケーション

本論文では、次の四つの分散計算アプリケーションを実験に用いた。

- Nクイーン  
N×Nマスの中に、縦、横、斜めに重ならないようにN個のクイーンを並べる組合せを探索するプログラム。
- ジグソーパズル  
2001年ACM国際大学対抗プログラミングコンテストアジア予選函館大会の問題Cの計算機用ジグソーパズル問題[9]に対して、ピースの大きさ及び回転も考慮する拡張を行った問題を解くプログラム。

- LU 分解

正方行列の Upper 行列 (上三角行列) と Lower 行列 (下三角行列) を求めるプログラム。NAS Parallel Benchmarks [10] に含まれるプログラムの一つ。

- タスクスケジューリング

標準タスクグラフ集 [11] を題材として、計算機の空きメモリを考慮したクリティカルパス法を用いて実際にタスクをスケジューリングするプログラム。クリティカルパス法は、タスクの重みと依存関係から最も長い時間がかかるタスクフロー (クリティカルパス) を求め、そのタスクフロー上にあるタスクを優先的に実行するアルゴリズム。

N キーン、ジグゾーパズル、LU 分解の分散計算はタスクフレーム型に、タスクスケジューリングはワークフロー型に分類される [12]。タスクフレーム型は、まずマスタに該当する計算機が複数のスレーブとなる計算機へ、処理に必要なデータを転送する。スレーブはタスクごとに設定されている計算パラメータを用いて処理を実行し、計算結果をマスタへ転送する。通常、計算開始直後又は終了直前に多量なデータ転送が発生する。

ワークフロー型は、一連の処理を複数の段階に分けてパイプラインに処理する。各段階は一つあるいは複数のタスクで構成され、タスクの計算結果は次の段階を構成するタスクの入力データとして扱われる。通常、タスク間でデータ交換が行われ、演算処理の間に頻繁に入力データ・計算結果の転送が発生する。

### 3 実験環境

図 1 に、4 及び 5 の実験で用いた実験構成を

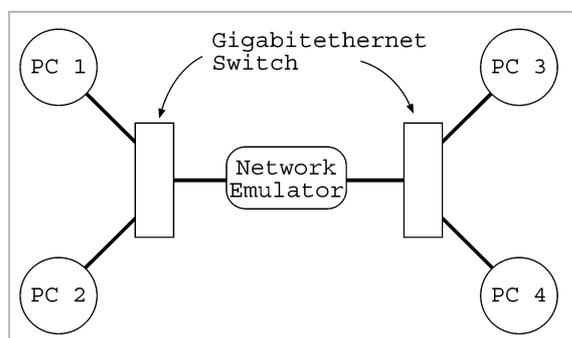


図 1 実験構成

示す。

2 で説明したアプリケーションは、PC1 から PC4 で実行し、起動は PC1 から行った。タスクフレーム型アプリケーションでは、PC1 をマスタとし、すべての計算機 (PC1 を含む) をスレーブとする。各計算機の仕様は、インテル社製 Pentium III 3.06GHz CPU、2GB メモリ、インテル社製 PRO/1000 NIC、PCI-X バス、RedHat 9 Linux kernel 2.4.22 である。ギガビットイーサネットスイッチはネットワークエミュレータ [13] を用いて接続し、通過するパケットへの遅延の挿入や、ボトルネック回線として帯域制限を行う。機器はすべてギガビットイーサネットで接続されている。同一のスイッチに接続される PC 間の往復遅延時間 (RTT) の平均は 0.141 ミリ秒、エミュレータを介して接続される PC 間の RTT の平均は、遅延を挿入しない場合で、0.331 ミリ秒であった。4 及び 5 に示す各アプリケーションのトラフィックパターン、フロー数、パケット間隔の解析には、TCPDUMP を用い、各計算機を出入りするすべてのパケットのキャプチャデータを使用した。

### 4 通信特性の解析

2 で説明したアプリケーションのトラフィック特性の調査を行った。具体的には、トラフィック量、トラフィックパターンの変化、トラフィックフローの構成について解析を行った。

表 1 に、ネットワークを最良の (エミュレータによる遅延の挿入及び帯域制限を行わない) 状態で、アプリケーションを実行したときに発生するトラフィック量、平均スループット、実行時間を示す。計測は PC3 で行い、各値は 20 回の試験の平均である。

すべてのアプリケーションにおいて、問題の規模が大きくなると実行時間が増加し、転送データが増加している。一方、転送データ量と実行時間の間の関係は、アプリケーションごとに異なる特徴を示している。

N キーンでは、転送データが 5 倍に増加し実行時間は 2.5 倍に拡大している。ジグゾーパズルでは、転送データが 1000 倍に増加し実行時間の拡大は 10 倍である。LU 分解では他のアプリケーションと異なり、問題の規模が大きくなると平均

表1 各アプリケーションのトラヒックの特徴

アプリケーション	入力トラヒック		出力トラヒック		実行時間 (秒)
	総トラヒック (MB)	平均スループット (Mbit/s)	総トラヒック (MB)	平均スループット (Mbit/s)	
Nクイーン (15×15マス)	0.926	0.878	39.2	37.2	40.78
Nクイーン (16×16マス)	4.00	0.167	242	10.1	191.33
ジグソーパズル (4×4ピース)	0.0780	0.0443	0.117	0.0666	13.70
ジグソーパズル (36×36ピース)	101	4.56	194	8.79	176.48
LU分解 (36×36行列)	67.5	10.4	67.2	10.3	50.18
LU分解 (64×64行列)	179	5.24	179	5.23	258.64
タスクスケジューリング (300タスク)	116	7.68	132	8.78	131.83
タスクスケジューリング (500タスク)	163	10.1	188	11.6	132.52

スループットが減少しているが、転送データ量と実行時間は、それぞれ3倍と5倍に増大している。タスクスケジューリングにおいては、転送データが1.5倍に増加しているが、実行時間はほとんど拡大していない。ジグソーパズルとタスクスケジューリングは、問題の規模が大きくなると平均スループットが増加しているため、転送データの増加に比べて実行時間の拡大が小さくなっていると考えられる。

以下の実験では、アプリケーションごとに、問題の規模を変えても同様の特徴が得られたため、各アプリケーションで問題の規模が大きい計算について報告する。

20回の計測の一つについて、PC3で送受されたトラヒックのスループット変化を図2に示す。図のx軸はアプリケーションを実行してからの経過時間を、y軸はスループット(10ミリ秒ごとの平均速度)を示している。PC2及びPC4においても同様の変化を示していた。また、各アプリケーションで問題の規模を変えても変化のパターンは類似していた。

アプリケーション間でスループットのパターンは大きく異なっていることが分かる。Nクイーンでは、処理が終了する直前にスレーブからマスタ

(PC3からPC1の方向)へ非常に多くのデータが転送されている。ジグソーパズルでは、絶えず5から10Mbit/sの一定レートで、データが交換されている。LU分解も絶えずトラヒックを送受しているが、スループット変化はon-offパターンを示している。タスクスケジューリングでは、間欠的に多量のデータが交換されている。図3は図2のスループット変化を累積分布で示したものである。

すべてのアプリケーションにおいて、入力トラヒックと出力トラヒックの間で同様の分布を示していることが分かる。Nクイーンでは、処理中の95パーセント以上の時間でトラヒックが発生しておらず、残りの時間で最大速度に近いスループットがある。ジグソーパズルでは、最も多いスループットは、入力トラヒックの場合で5Mbit/s、出力トラヒックの場合で10Mbit/sとなっており、図2で確認されるレートと一致する。LU分解では、入力・出力トラヒックの半分が1Kbit/s以下となっており、on-offトラヒックの特徴が現れている。タスクスケジューリングでは、30%の時間でトラヒックが発生しておらず、40%の時間で20Mbit/sから50Mbit/sのスループットがある。

次に、アプリケーショントラヒックのフローの

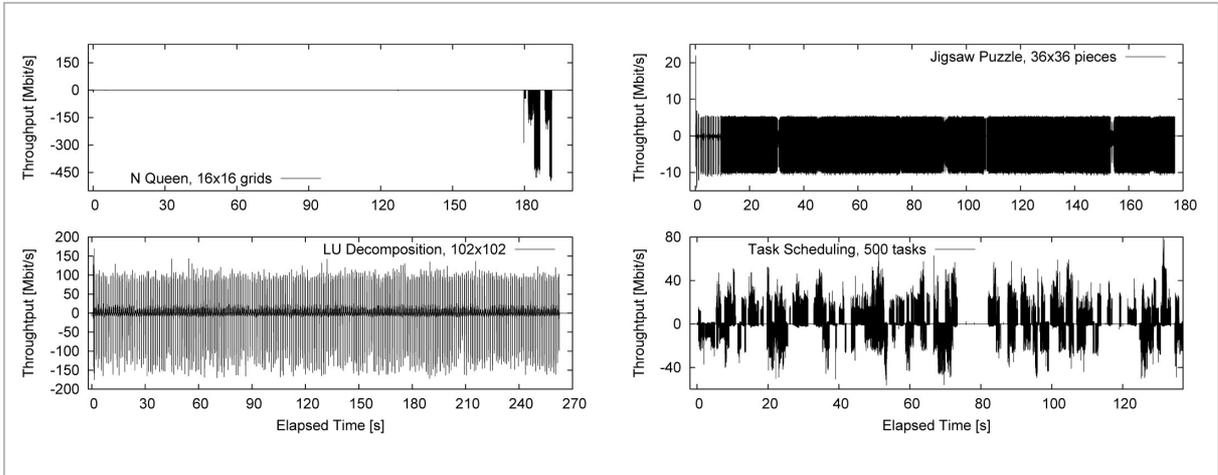


図2 スループットの変化

構成について調査した。各アプリケーションのトラフィックは TCP のみであったため、フローを、方向別で、SYN フラグで始まり FIN フラグで終了する、送受信アドレス及びポート番号で区別されるパケットの集合とした。各アプリケーションは、複数のフローを同時に発生させていた。

図 4 は N キューン (a) とタスクスケジューリング (b) の各フローの継続時間と転送データ量を示している。図中の直線は 1Gbit/s の限界速度である。ジグゾーパズル及び LU 分解は N キューンと同様の特徴を示したため図を割愛する。

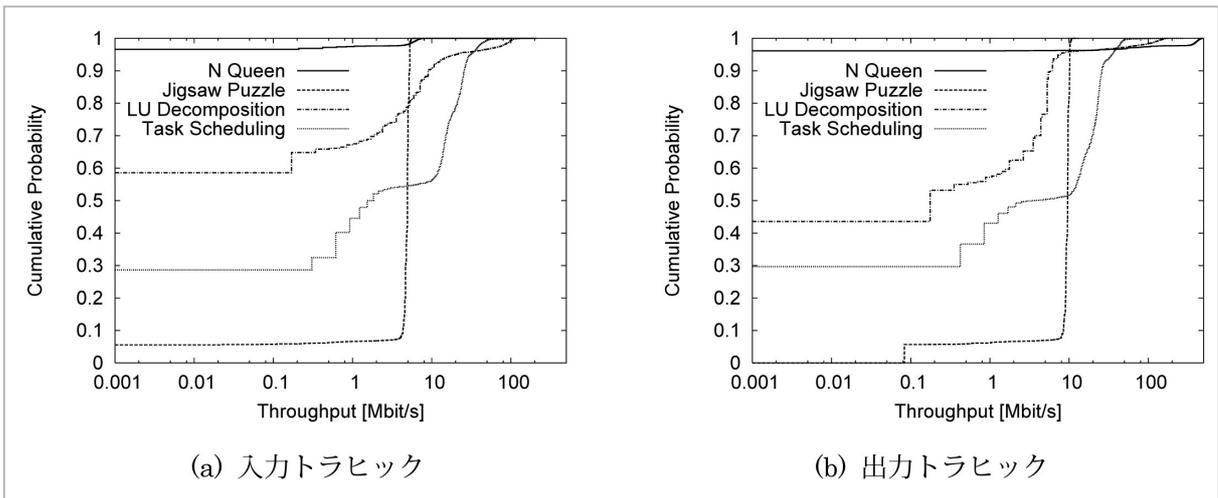
N キューンでは、タスクスケジューリングに比べフロー数が少なく、短時間で終了するフローと長時間継続するフローに区別される。長時間継続するフローは転送データ量が様々で、処理終了直前の多量なトラフィックもこの長時間継続するフロ

ーによって運ばれたと考えられる。タスクスケジューリングでは、フローの継続時間及び転送データ量が共に様々なことが分かる。N キューン、ジグゾーパズル、LU 分解の分散計算タイプがタスクフレーム型で、タスクスケジューリングがワークフロー型に分類されることから、このようなフロー構成の特徴の違いは、分散計算のタイプの違いから来るものと考えられる。

以上、トラフィック量、スループットパターン、フロー構成の解析から、次のことが分かった。

- アプリケーションによって異なるトラフィックパターンを示す。
- 問題の規模が大きくなって実行時間が延びても、通信の特徴はアプリケーションごとに同様の特徴を示す。

さらに、各アプリケーションのトラフィックの特



(a) 入力トラフィック

(b) 出力トラフィック

図3 スループットの累積分布

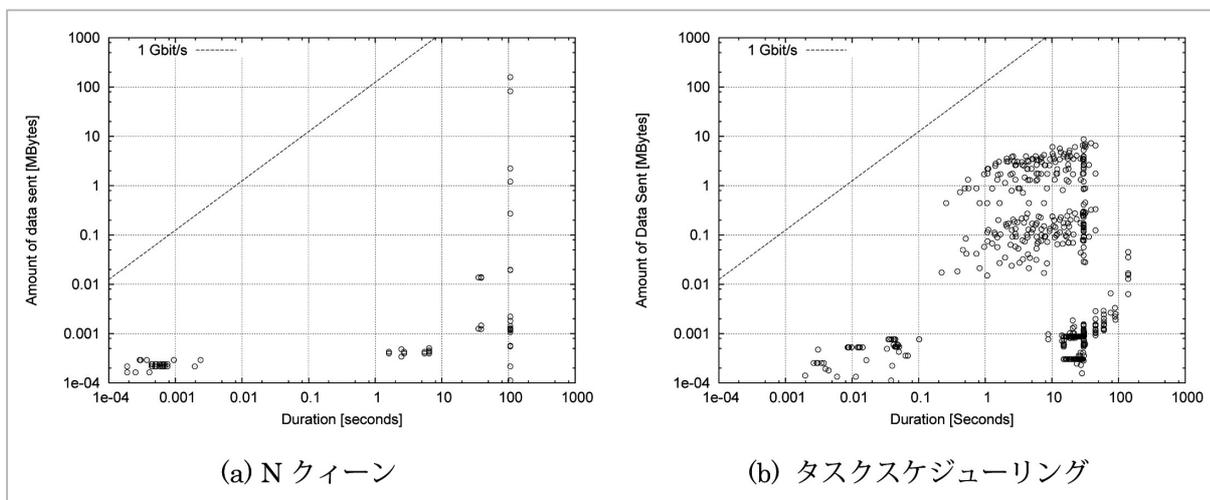


図4 各フローの継続時間と転送データ量

徴から、Nクイーンのような多量のデータを一気に送信するアプリケーションは、遅延が大きくても広帯域を確保できれば、通信時間を抑えて実行性能を良好に保てることが予想される。ジグゾーパズルのような通信は、要求帯域が少なく、多重したときに他のアプリケーションへの影響が小さいと考えられる。LU分解のような通信は、on-offパターンを示すことから、トラフィックが多重されたときに他のアプリケーショントラフィックへの影響が比較的大きいと考えられる。タスクスケジューリングのような通信は様々なホストの間で間欠的な通信を行うため、他のアプリケーショントラフィックとの多重効果が比較的可なりと考えられる。

## 5 ネットワークの状態がアプリケーションの実行性能に及ぼす影響

この章では、ネットワーク状態がアプリケーションの実行性能に及ぼす影響を調査する。具体的には、RTTが大きいとき、ボトルネック帯域が狭いときの実行性能の劣化について調べる。

### 5.1 RTTの増大がアプリケーションの実行性能へ及ぼす影響

図1の実験構成において、ネットワークエミュレータを通過するすべてのパケットに片道1から32ミリ秒の遅延を挿入し、アプリケーションの実行性能の劣化を調べた。遅延の影響に注目するた

め、すべての回線の帯域は1Gbit/sとした。NクイーンとジグゾーパズルではTCPコネクションのソケットバッファの大きさを16KBと128KBに設定し、LU分解では128KBと1024KBに設定し、タスクスケジューリングでは64KBと256KBに設定して調査した。

図5はRTTを大きくしたときの実行性能の劣化の様子を示している。図のx軸はRTTを、y軸は劣化の度合いを示している。劣化の度合いは、挿入遅延が0のときの実行時間を1として正規化した。各実行時間は20回の試験の平均とした。実験では、Nクイーンとジグゾーパズルが、ソケットバッファの大きさを変えても同じ劣化の特性を示したため、128KBについては図を割愛する。

すべてのアプリケーションにおいて、RTTが増加すると性能がより劣化するが、その中でジグゾーパズルは他のアプリケーションに比べて劣化が小さい。LU分解とタスクスケジューリングでは、遅延の影響を小さくするためにソケットバッファの拡大が大変効果的なことが分かる。それに対して、Nクイーンはソケットバッファの拡大の効果を得られていない。Nクイーン、ジグゾーパズル、LU分解は分散計算タイプが同じであるにもかかわらず、このように遅延の影響が異なった。その理由を調べるため、RTTが小さいときと大きいときの場合で、フローごとのパケット間隔(時間の差)と転送データ量を計測した。

図6にNクイーン(a)とジグゾーパズル(b)のパケット間隔の累積分布を示す。LU分解はNク

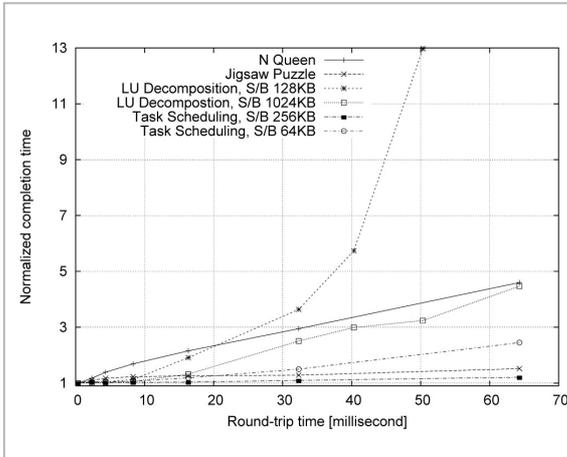


図5 RTTの影響を受けたときの実行性能の劣化の様子

ーンと同様の分布を示したため図を割愛する。

Nクイーン(a)では、RTTが小さいとき(0.331ミリ秒)、ほとんどのパケット間隔がRTT以下となっている。1500バイトのパケットが、間隔をあげずにギガビットの速度で通過するときのパケット間隔は0.012ミリ秒となるが、Nクイーンでは全体の70%がこのような短い間隔になっていたことから、パケットがバースト的に送信されていたことが分かる。RTTが大きく(32.3ミリ秒)なると、短い間隔のパケットが60%に減って、拡大したRTTの間隔に移ったことから、RTTの増大によってTCPの送信ウィンドウの円滑な拡大が妨げられ、データ送信に時間を要すようになり、スループットが減少したと考えられる。

ジグゾーパズル(b)では、遅延が小さいとき、

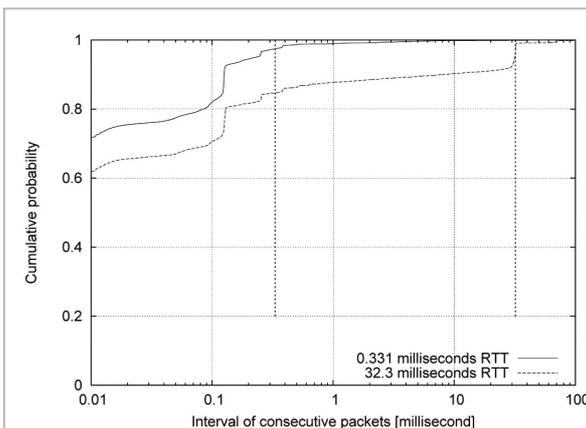
短いパケット間隔が全体の40%を占め、次に多いパケット間隔がRTT(0.331ミリ秒)に近い値であった。RTTの大きさのパケット間隔は、インタラクティブ性のある通信であると考えられる。RTTを大きくすると、Nクイーンと異なり、もともとRTTの大きさのパケット間隔が新しいRTT(32.3ミリ秒)へ移ったことから、インタラクティブ性の通信が遅延の影響を受けたことが分かる。バースト性の高い通信が遅延の影響を受けていないことから、ジグゾーパズルの場合、一度に送信されるデータ量がウィンドウサイズ(16KB)より少ないことが頻繁にあったと考えられる。

表2は各アプリケーションの、RTTと双方向の転送データ量の関係を示している。ジグゾーパズルは、他のアプリケーションと異なり、RTTが大きくなるほど転送データ量を減少させることが分かる。インタラクティブ性を示す通信が遅延の影響を受けつつも、全体の実行性能が劣化しにくい理由は、この転送データ量の減少によるものと考えられる。

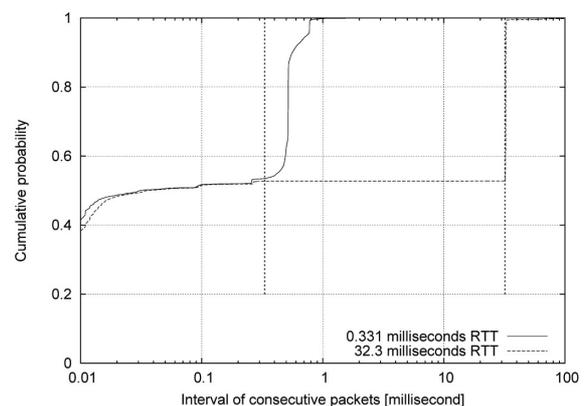
以上から、RTTの大きさに対するアプリケーションの実行性能の感度の情報は、アプリケーショントラフィックに複数のパスの一つを割り当てるような状況において、大変有効であると考えられる。

## 5.2 狭帯域のボトルネックがアプリケーションの実行性能へ及ぼす影響

本節では、図1実験構成のネットワークエミュレータにおいて帯域を80Kbit/sから1Gbit/sに絞



(a) Nクイーン



(b) ジグゾーパズル

図6 パケット間時刻差の累積分布

表2 RTT と転送データ量の関係

アプリケーション	RTT: 0.331 ミリ秒		RTT: 4.3 ミリ秒		RTT: 16.3 ミリ秒		RTT: 64.3 ミリ秒	
	パケット	バイト	パケット	バイト	パケット	バイト	パケット	バイト
N クイーン	233K	242M	234K	242K	227K	238M	225K	237M
ジグソーパズル	2372K	297M	992K	118M	293K	35.1M	84K	9.99M
LU 分解	674K	475M	716K	513M	739K	537M	748K	542M
タスクスケジューリング	823K	338M	898K	362M	720K	313M	690K	352M

り、ボトルネック回線の帯域が各アプリケーションの実行時間に及ぼす影響を調査する。

ボトルネック回線の帯域を 1Gbit/s から縮小させたときの実行時間の変化を図 7 に示す。図では、ボトルネック回線の帯域が 1Gbit/s のときの実行時間を 1 として正規化を行っている。ボトルネックの影響に注目するため、遅延は挿入しなかった。ソケットバッファは、N クイーンとジグソーパズルでは 16KB に、LU 分解では 1024KB に、タスクスケジューリングでは 256KB に設定した。各値は 20 回の試験の平均を示している。

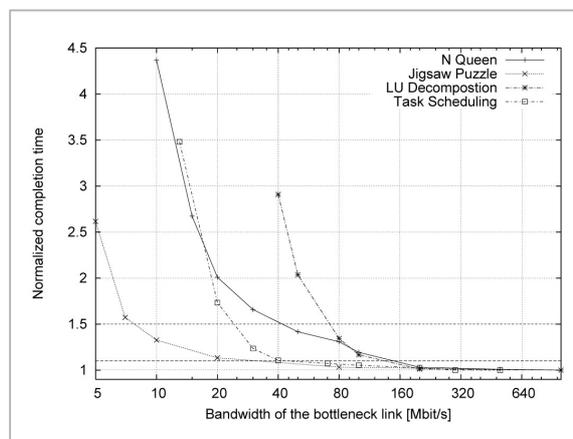


図7 ボトルネックの帯域が縮小したときの実行性能の劣化の様子

図から、実行時間がある値(例えば、1.1)に達するまでは、ボトルネック回線の帯域の縮小に対する実行時間の増大は緩やかなことが分かる。このような実行時間への影響が小さいボトルネック回線の帯域はアプリケーションによって異なる。さらにボトルネック回線の帯域を縮小すると、実行時間がある値(例えば、1.5 倍)を超えてからは、ボトルネック回線の帯域の縮小に対して実行時間の増大が急激になることが分かる。このような実行時間への影響が大きくなるボトルネック回線の

帯域もアプリケーションによって異なる。それぞれの違いを表 3 に示す。

表3 緩急の実行性能の劣化を引き起こすボトルネック帯域

アプリケーション	緩やかな劣化を示す帯域	急激な劣化を示す帯域
N クイーン	140~500 Mbit/s	< 40 Mbit/s
ジグソーパズル	30~200 Mbit/s	< 7 Mbit/s
LU 分解	140~200 Mbit/s	< 70 Mbit/s
タスクスケジューリング	40~200 Mbit/s	< 25 Mbit/s

このようなボトルネックの制約に対するアプリケーションの実行特性の情報は、多様なアプリケーションが並行して実行されるネットワークにおいて、アプリケーショントラフィックに実行性能を考慮した帯域を割り当てる際に有効と考えられる。

## 6 むすび

Grid アプリケーションのトラフィックがネットワーク資源の状態に与える影響と、逆にネットワーク資源の状態がアプリケーションの実行性能に与える影響を明らかにするため、Grid 上で動作可能な分散計算アプリケーションの、ネットワーク特性に対する実行特性を調査した。

まず、各アプリケーションについて生成されるトラフィックの特徴を解析し、アプリケーションによって転送データ量と実行時間は異なるものの、すべてのアプリケーションにおいて問題の規模が拡大すると転送データと実行時間が増大することが確認された。さらに、トラフィックフローについて解析し、N クイーン及びジグソーパズル、LU

分解(タスクフレーム型アプリケーション)では比較的少数のフローを発生させ、各フローの継続時間は短いものと、比較的長いものに分類された。一方、タスクスケジューリング(ワークフロー型アプリケーション)では非常に多くのフローが発生し、各フローの継続時間と転送データ量は共に様々な値を示した。

次に、ネットワーク資源の状態がアプリケーションの実行時間に及ぼす影響を調査した。すべてのアプリケーションにおいて RTT の増大に伴って実行時間の増大が確認されたが、その劣化の度合はアプリケーションによって大きく異なることが分かった。さらに、LU 分解及びタスクスケジューリングでは、送信ウィンドウの拡大によって

遅延増大による性能劣化を抑制できることが分かった。次に、すべてのアプリケーションにおいて、ボトルネック回線の帯域がある値以下になると性能が急激に劣化し、その帯域はアプリケーションによって大きく異なることも分かった。

本研究は、多様な Grid アプリケーションが同一のネットワークで実行される環境で、アプリケーションの実行性能を考慮したトラヒックエンジニアリングの実現を目指している。本報告では、アプリケーションの実行性能を考慮したトラヒックエンジニアリングを実現するためには、ネットワーク特性と分散計算アプリケーションの実行性能の関係をあらかじめ明らかにすることが有効であることを示唆した。

## 参考文献

- 1 I. Foster and C. Kesselman, "The GRID Blueprint for a New Computing Infrastructure", Morgan Kaufmann Publishers, 1998.
- 2 I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International Journal of Supercomputer Applications, Vol. 15, No. 3, pp. 200-222, 2001.
- 3 A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS adaptive traffic engineering", Proc. of the Infocom, pp. 1300-1309, 2001.
- 4 T. Guven, C. Kommareddy, R. La, M. Shayman, and S. Bhattacharjee, "Measurement Based Optimal Multi-path Routing", Proc. of the Infocom, 2004.
- 5 R. Kawahara, "An Adaptive Load Balancing Method for Multiple Paths Using Flow Statistics and Its Performance Analysis", IEICE Transactions on Communications, Vol. E87-B, No. 7, pp. 1993-2003, 2004.
- 6 N. S. V. Rao, "NetLets: End-To-End QoS Mechanisms for Distributed Computing in Wide-Area Networks Using Two-Paths", Proc. of the first International Conference on Internet Computing, pp. 475-478, 2001.
- 7 K. Aida and T. Osumi, "A Case Study in Running a Parallel Branch and Bound Applications", Proc. of the 2005 International Symposium on Application and the Internet (SAINT2005), pp. 164-173, 2005.
- 8 A. Plaat, H. E. Bal and R. F. H. Hofman, "Sensitivity of Parallel Applications to Large Differences in Bandwidth and Latency in Two-Layer Interconnects", Proc. of the 5th High Performance Computer Architecture, pp. 244-253, 1999.
- 9 The ACM International Collegiate Programming Contest Japan Domestic, Problem C, Jigsaw Puzzle for Computers, [http://www.fun.ac.jp/icpc/domestic\\_problems.html](http://www.fun.ac.jp/icpc/domestic_problems.html), 2001.
- 10 D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagurm, R. A. Fatoohi, P. O. Federickson, T. A. Jasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrisnan, and S. K. Weeratunga, "The NAS Parallel Benchmarks", International Journal of Supercomputer Applications, Vol. 5, No. 3, pp. 66-73, 1991.

- 11 T. Tobita and H. Kasahara., "A standard task graph set for fair evaluation of multiprocessor scheduling algorithms", Journal of Scheduling, Vol. 5, issue 5, pp. 379-394, 2002.
- 12 R. Buyya, "High Performance Cluster Computing : Programming and Applications", Vol. 2, Prentice Hall PTR, 1999.
- 13 Empirix Packet Sphere, <http://www.empirix.com/>

きたつじよしのり  
**北辻佳憲**

拠点研究推進部門北九州 JGN II リサーチセンター専攻研究員  
トラヒック制御技術

たがしらひでき  
**田頭秀樹**

拠点研究推進部門北九州 JGN II リサーチセンター特別研究員(九州電力株式会社)  
トラヒック計測

やまざきかつゆき  
**山崎克之**

株式会社 KDDI 研究所 博士(工学)  
通信品質、ネットワーク設計

つる まさと  
**鶴 正人**

拠点研究推進部門北九州 JGN II リサーチセンター専攻研究員 博士(情報工学)  
ネットワーク性能計測、ネットワークモデリング、ネットワーク解析

おいえゆうじ  
**尾家祐二**

拠点研究推進部門 JGN II 研究開発プロジェクト総括責任者 工学博士  
情報ネットワーク工学