2007－HPC－111 (44)
2007／8／3

経路が動的に接続/解放されるネットワークにおけるユーザ単位の経路切替手法

岡崎史裕† 工藤知宏†
中田秀基† 竹房あつ子†

同一宛先との通信に複数の経路がある IP ネットワークでは、実際に通信に使用する経路を選択する仕組みが必要である。例えば、遠隔の計算機間の帯域を動的に確保して並列処理を行う場合や、VLAN を用いてレイヤ 2 ネットワーク上に複数の経路を実現する VLAN ルーティング法では、ユーザごとに経路を切替えることが有効であると考えられる。そこで本稿では、これを実現する手法として、Linux の netfilter と iproute2 を用いる方法と、ロードバランサ装置である BIG-IP を用いる方法を比較検討する。また、これらの切替手法が並列計算アプリケーションの性能面に与える影響を調べるため、NPB3.2 による性能評価を行う。

Per-user Path Selection Methods for Multiple Paths IP Network

FUMIHIRO OKAZAKI ,† TOMOHIRO KUDOH ,† HIDEMOTO NAKADA †
and ATSUKO TAKEFUSA†

When multiple paths exist between two end-points in an IP network, a mechanism to select a path is required. For some applications like dynamic path provisioning network and VLAN-based routing, it is desirable to select a path per-user and per-application, according to the user's decision. In this report, we propose two such methods, use of a combination of netfilter and iproute2 of Linux and use of a load balancer hardware such as BIG-IP, and compare these two methods in terms of performance and functions. NAS parallel benchmark results are shown to compare the performance of the methods for real applications.

1. はじめに

我々は、広域ネットワーク上で、帯域が保証された経路を動的に確保、開放し、必要時にのみ必要な帯域を提供する仕組みの開発を行っており、他組織との共同研究である G-lambda プロジェクト[1),2)] において、帯域確保のためのインタフェースの策定を行うと共に、実証実験を行ってきた。この実証実験を通じて、従来型のネットワークによって遠隔装置間の接続性は常に確保しつつ、新たな経路が提供されている間だけ、その経路を要求したユーザのトラフィックのみを当該経路にルーティングする仕組みが必要なことが明らかになった。

また、我々はクラスタ内のネットワークとしてレイヤ 2 のイーサネットを用いる場合に、複数経路を実現する手法として VLAN ルーティング法を提案している[3)]。これは、物理トポロジ上に、複数の異なる木構造のトポロジを持つ VLAN を設定することにより、木構造状でないループを含む物理トポロジを利用することを可能にしたものである。VLAN ルーティング法では、トラフィックを VLAN に振り分けることにより負荷分散を行う。アプリケーションの通信パターンにより最適な振り分け方が異なるため、ユーザやアプリケーション単位に経路を切換える仕組みが有効であると考えられる。実際のクラスタの運用では、管理などのための通常の通信が利用する常に確保されている経路（これは、特定の VLAN を用いても良いし、物理的に別のネットワークを用意しても良い）に加えて、ユーザによるジョブの実行時に新たな経路を VLAN により確保することになる。

このような用途での経路切替では以下に示す要件が考えられる。ここではユーザの要求により提供される経路を新たな経路と呼び、常に接続している経路を従来の経路と呼ぶことにする。

- 動的な経路切替
  新たな経路は、ユーザの要求により、ジョブの実行開始時に設定され終了時に削除されなくてはならない。経路の切替はジョブの実行時間に相当する比較的長い時間間隔で行われることになる。
- ユーザと宛先アドレスの組による経路の選択
  新たな経路は、要求したユーザが占有して利用できることが望ましい。広域ネットワークにおける動的帯域提供では、帯域遅延積の大きな通信が想定されるため、ユーザが他のトラフィックの影響を受けずに安定して新たな経路の帯域を利用できることが重要であるし、確保した経路について個々のユーザが課金される使い方も考えられる。また、VLAN ルーティング法では、アプリケーションごとに通信のパターンが異なるため、それぞれに経路を選択できる

† 産業技術総合研究所 グリッド研究センター
Grid Technology Research Center, AIST

必要がある。また、同一ジョブでも宛先アドレスごとに異なる経路を使用できなくてはならない。

- 単一 IP アドレスでの異なる経路の利用

新たな経路が確保された場合に、アプリケーションの設定変更は不要であることが望ましい。このためには、ユーザプロセスは複数の経路に対して同一の IP アドレスで通信できる必要がある。特に、並列処理のための MPI[4] などの通信ミドルウェアでは、各プロセスは単一の IP アドレスしか持つことが出来ず、他のどのプロセスからも、そのアドレスで通信可能でなくてはならない場合がある。

- ジョブマネージャによる経路切替の管理

多くのユーザにより利用されるクラスタ計算機システムなどでは、キューイングシステムなどのジョブマネージャによりユーザジョブの実行が管理されるのが普通である。この場合、経路切替は、各ジョブの実行開始前と終了後に行う必要があるため、ジョブマネージャがサポートすることが望ましい。このためには、ジョブ開始前に新たな経路を設定し (PRE 処理)、終了後にこの経路を削除する (POST 処理) 機能がジョブマネージャに必要とされる。特に、ユーザジョブが異常終了した時にはユーザジョブの一部として経路の削除処理を実行することはできないため、ジョブマネージャによる POST 処理が必要となる。

そこで本稿では、ユーザ単位に経路切替を実現する手法として、拠点内部のノードにおける切替手法 (ノード内切替手法) と拠点の接続点における切替手法 (接続点切替手法) を比較検討する。ノード内切替手法は Linux の netfilter と iproute2 を用いる手法を、接続点切替手法は接続点にロードバランサ装置 (ここでは F5 社の BIG-IP[5]) を用いる手法を採用した。どちらの切替手法でもユーザ単位の経路切替が可能である。これらの切替手法が並列計算アプリケーションの性能面に与える影響を調べるため、NPB3.2[6] による性能評価を行った。

なお、切替の対象となる各経路間は、ネットワークの中間で経路制御されていてはならない。インターネットでは、各ドメイン間で BGP(Border Gateway Protocol) などにより経路を交換することで経路制御されている。従って、経路切替の対象となる経路のうち複数がインターネットにルーティングされていると、提案方式は用いることが出来ない。

2. 経路切替手法

2.1 ノード内切替手法

ノード内切替手法として、Linux の netfilter と iproute2 を組合わせて経路切替を行う手法を説明する。

netfilter はパケットフィルタリング機能や NAT、IP マスカレード機能の他にパケット情報を書換える機能を提供するフレームワークである。netfilter は、iptables コマンドによって作成されたルールに従ってパケットのマッチングを行い、その後の処理を決定する。マッチングルールには、パケットのヘッダ情報の他にパケットを生成したプロセスのユーザ情報も指定できる。さらに、マッチングしたパケットの ToS(Type of Service) 値や FWMARK 値などを変更できる。ToS は IP ヘッダの中にある情報で、ネットワーク内でパケットの優先度の識別に利用することができる。FWMARK は、sk_buff 構造体のメンバ nfmark で管理されており、ノード内でパケットの識別に利用できる。

iproute2 はパケットに対するクラス分け、優先度、帯域や経路などを制御するトラフィックエンジニアリングを提供するためのフレームワークである。昔から Linux 等では route コマンドにより 1 個の経路表で経路を管理してきたが、iproute2 を使用することで、複数の経路表の中からポリシーに適合した経路表を用いてルーティングすることが可能となる。ポリシーは送信元アドレス、送信先アドレス、ToS 値や FWMARK 値を用いたルールにより指定する。これにより、ポリシーに適合したパケットは、他のパケットとは別の経路表によりルーティングされ、route コマンドで設定した経路とは別の経路が選択される。

ノード内切替手法では、特定のユーザが生成したパケットを対象に、netfilter により ToS や FWMARK をマーキングし、iproute2 でこの値に適合するポリシーの経路表を適用することで、ユーザ単位の経路切替を実現する。

2.2 接続点切替手法

接続点切替手法として、接続点にロードバランサ装置である BIG-IP を用いる場合を例に説明する。BIG-IP は、インターネットサービスのパフォーマンスや可用性の向上を目的としたトラフィックコントロールを行うレイヤ 7 スイッチである。これを実現するため、BIG-IP は負荷分散、NAT やトラフィック制御などの機能を持っている。

BIG-IP は、iRule と呼ばれるスクリプトで記述されたルールを用いることで、フロー単位の柔軟なトラフィック制御を可能にしている。iRule はフローの開始や終了などイベント毎に実行され、パケットの TCP/IP ヘッダやアプリケーションヘッダの情報によりトラフィック制御を行うことができる。iRule のトラフィック制御にはフロー単位に経路を切替える機能がある。

BIG-IP はパケットのユーザを直接には判断できない。ToS にマーキングした情報によりパケットのユーザを間接的に判断することはできる。ユーザの指示に従って通信ミドルウエアが ToS へマーキングすることは簡単であるが、本機能をサポートしている物は現在のところほとんど存在しない。しかし、各ノードで netfilter により ToS にマーキングすることは可能である。

例えば、ToS 値が 8、宛先が 192.168.201.0/24 なら、新たな経路のルータ 192.168.203.253 に切替える設定を以下に示す。

- pool の設定

新たな経路のルータを定義する pool gw203 の設定をする。

```
pool gw203 {
   monitor all gateway_icmp
   member 192.168.203.253:any
}
```

- iRule の設定

ToS 値が 8 であれば pool gw203 を使用するように rule203 を設定をする。

```
rule rule203
when CLIENT_ACCEPTED {
```

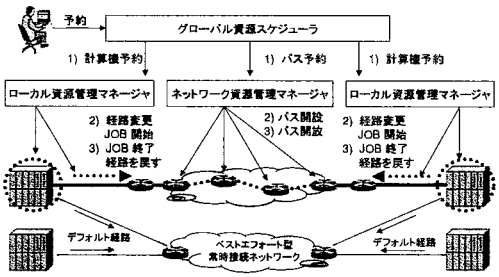

図 1 事前予約システムの例

```
  set client [IP::client_addr]
  if { [IP::tos] == 8 ) } {
    if { [IP::addr $client equals
    192.168.201.0/255.255.255.0] } {
      pool gw203
} } }
```

3. 経路切替の適用

3.1 動的帯域提供ネットワークへの適用

例えば、あるユーザが MPI などで記述された並列処理を行うために帯域を確保する場合、G-lambda ではネットワークも資源の 1 つとして扱い、他の計算機資源などと共に事前予約して利用する仕組みを提供している。**図 1** に示すように様々な資源の管理・予約を行うグローバル資源スケジューラが中心となり、これと個々のクラスタ内の計算資源を管理するローカル資源管理マネージャや個々のネットワークドメインを管理するネットワーク資源管理マネージャが連携して、事前予約を行う。

予約した時刻になると、各ネットワーク資源マネージャが帯域保証されたネットワークを提供し、各クラスタのローカル資源管理マネージャが計算ジョブの実行を開始する。ローカル資源管理マネージャはジョブ実行開始前の処理 (PRE 処理) として、このジョブのパケットを対象に常時接続ネットワークから帯域保証ネットワークへ経路の切替えることで実行環境を整える。また、ジョブの終了時や予約した時刻の終了時の処理 (POST 処理) として経路を元に戻す処理を行う。経路切替には帯域保証されたネットワークで接続される計算機資源のアドレスの情報が必要である。ローカル資源管理マネージャはグローバル資源スケジューラからこの情報を入手する。

PRE/POST 処理に本切替手法を適用することで、他のユーザとは別の経路表による経路切替が可能となる。これにより、ノードで複数ユーザのジョブが同時に実行されたとしても、帯域保証ネットワークを要求したユーザのトラフィックのみ当該経路にルーティングすることが可能である。また、個別の経路表を用いるため、同じ宛先アドレスでも別の経路を選択することができ、複数のネットワークを同じアドレスで運用することが可能となる。

3.2 VLAN ルーティング法への適用

VLAN ルーティング法は Fat Tree などの複数パスを持つトポロジのネットワークをイーサネットを用いて構築する手法である。**図 2** に示すように、ノード

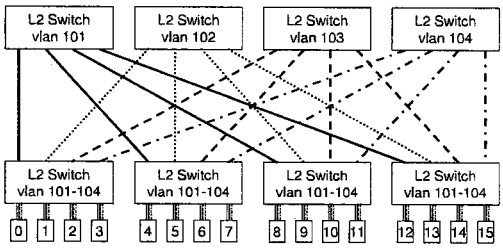

図 2 VLAN ルーティング法による Fat Tree の構成例

図 3 VLAN で接続されたモデル

間の複数パスを VLAN で構成し、送信ノードはどの VLAN を使用するかで経路を選択する。

VLAN ルーティング法のすべての VLAN に対して同じアドレスを設定して、経路を切替える方法について説明する。簡単なモデルを**図 3** に示す。2 つの VLAN で接続されたノードがあり、これらの VLAN には同じアドレスを設定する。Linux 等の経路設定では次にパケットを転送すべきルータのアドレス (ゲートウエイ) と共にネットワークデバイスが指定できる。レイヤ 2 接続ではゲートウエイは存在しないから、ネットワークデバイスのみを指定すれば送信する VLAN を選択できる。ノード A でノード B への経路を切替える例を以下に示す。

- VLAN 101 を使用する場合

```
# route add -net 192.168.0.2/32 \
        -dev eth0.101
```

- VLAN 102 を使用する場合

```
# route add -net 192.168.0.2/32 \
        -dev eth0.102
```

VLAN ルーティング法を用いたクラスタをジョブマネージャを用いて運用する場合には、PRE/POST 処理に本切替手法を適用することで、ユーザ単位に個別な経路表を用いた VLAN の選択ができる。各 VLAN を同じアドレス空間で運用するため、単一アドレスで異なる経路が利用できるようになる。

ユーザがアプリケーションの種類によって複数の経路表を使分けたい場合には、ユーザ情報の他にどの経路表を適用するかを ToS にマーキングする必要がある。ユーザの指示に従って通信ミドルウエアが ToS にマーキングできれば、同一ユーザのアプリケーションでも、各々異なる最適な経路の選択できる。

4. 評価実験

評価実験として各切替手法のルール数の違いにより、切替手法が通信やアプリケーションの性能に与える影響を調べる。ノード内切替手法は、パケットを発生させたプロセスのユーザを識別し、FWMARK をマーキングするルールとした。接続点切替手法は送信先アドレスにより経路を切替えるルールとした。

実験結果中のルール数は経路切替のために記述した

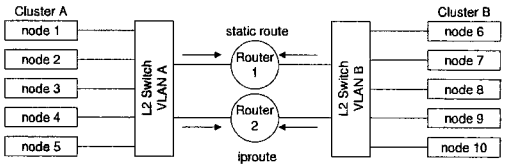

図 4 ノード内切替手法の評価環境

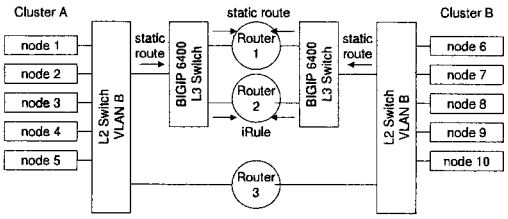

図 5 接続点切替手法の評価環境

ルール数を示す。ルール数が 0 の場合には各切替手法を無効とし、静的経路を使用した場合を指す。

実験中、RTT は GridMPI[8),9)] を使用して 8bytes メッセージを 10 万回往復させた結果の平均を示す。スループットは送受信のバッファサイズを 128KB に設定した iperf を使用して 120 秒間の平均スループットを示す。iperf は version 2.0.2 を用いて、gettimeofday 関数による UDP のスループット調整部分をキャンセルしている。iperf の示すスループットは TCP/UDP のペイロード部分の転送速度である。

4.1 評価環境

ノード内切替手法の評価環境を**図 4** に示す。2 つのクラスタ A、B を L2 スイッチ A、B に接続して、これらの L2 スイッチ間をルータで接続した構成である。それぞれのクラスタへのゲートウエイに静的経路設定で Router-1 を指定し、ノード内切替手法により Router-2 に変更している。L2 スイッチ A、B と Router-1、2 は物理的に 1 台の L3 スイッチ内で構築している。

接続点切替手法の評価環境を**図 5** に示す。図 4 の L2 スイッチとルータの間に BIG-IP を挿入した構成になっている。クラスタで BIG-IP をゲートウエイに設定し、BIG-IP ではそれぞれのクラスタへのゲートウエイに静的経路設定で Router-1 を指定し、iRule により Router-2 に変更している。

図 4 と図 5 の構成で L2 スイッチと各ルータ間の帯域に大きな違いがある。図 4 では L3 スイッチの内部接続となるため 5 ノードに対して 5Gbps 以上あるが、図 5 では 1000BASE-T 接続となるため 1Gbps しかなくボトルネックとなる。このため、1000BASE-T で接続された Router-3 を追加することで、接続点切替手法の性能を正しく比較できるようにした。また、BIG-IP はルータとして動作しているので、この環境ではクラスタ間のホップ数は 3 となる。

評価環境で使用した機器の諸元を**表 1** に示す。

4.2 L3 スイッチとしての BIG-IP の事前評価

接続点切替手法の経路切替評価の前に BIG-IP を L3 スイッチとして使用した場合のスループット性能を計測した。複数の UDP フローによるスループットの計測結果を**表 2** に示す。複数の UDP フローはそれぞれ異なるホスト間で転送されるようにし、各々のフローは独立な 1000BASE-T を経由する。フローが 1 本の場合にはワイヤーレイトで転送できるが、複数のフローではその合計が 1000BASE-T の転送レートとほぼ一致する。フローの一部の方向を変えても同じ傾向を示す。これは、BIG-IP 内部の L3 転送機能に双方向合わせて 1Gbps のボトルネックがあることを示している。この原因はライセンスや設定の問題の可能性もあるが、現在のところわかっていない。なお、本機のカタログ上のスループットは 2Gbps となっている。

表 1 評価環境で使用した機器の諸元

PC クラスタ ノード	OS	Linux 2.6.20-1.2962.fc6
	CPU	Intel Pentium4 2.40GHz
	NIC	Intel PRO/1000
	Memory	512MB
L3 スイッチ L2 Switch-A,B, Router-1,2	Machine	CISCO WS-C3750G-24T
	Version	IOS 12.1(19)EA1c
L3 スイッチ Router-3	Machine	PowerConnect 6248
	Version	image 1.0.0.27
BIG-IP	Machine	BIG-IP 6400
	Version	9.1.2 40.6

表 2 BIG-IP の L3 スイッチの性能

フロー数	スループット (Mbps)				合計 (Mbps)
1	955				955
2	478	478			956
4	241	238	237	237	953

表 3 ノード内切替手法の通信基本性能

ルール数	RTT(msec)	スループット (Mbps)	
		UDP	TCP
0	0.119	955	938
1	0.136	955	938
10	0.141	955	938
50	0.151	955	938
100	0.168	955	913

表 4 接続点切替手法の通信基本性能

ルール数	RTT(msec)	スループット (Mbps)	
		UDP	TCP
0	0.201	955	914
1	0.201	955	914
100	0.201	955	914

4.3 通信基本性能の評価

基本通信性能としてラウンドドリップタイム (RTT) とスループットを各クラスタの 1 ノード間で計測した結果を**表 3**、**表 4** に示す。

ノード内切替手法ではルール数が増加すると RTT も増加する。TCP フロー転送時の CPU 使用率を測定し

表 5 ノード内切替手法の CPU 使用率

ルール数	Reciever		Sender	
	user(%)	system(%)	user(%)	system(%)
0	1.4	19.3	1.1	25.8
1	1.4	21.8	1.3	37.3
10	1.3	22.7	1.2	41.4
50	1.5	25.6	1.6	73.2
100	1.4	29.5	1.6	98.4

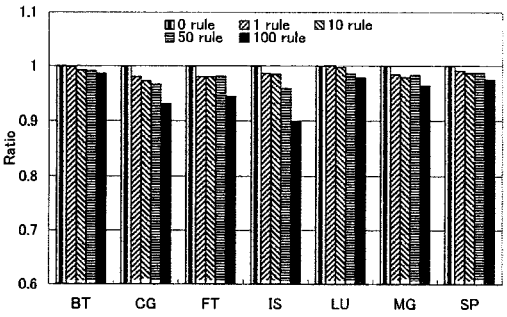

図 6 ノード内切替手法による NPB の性能

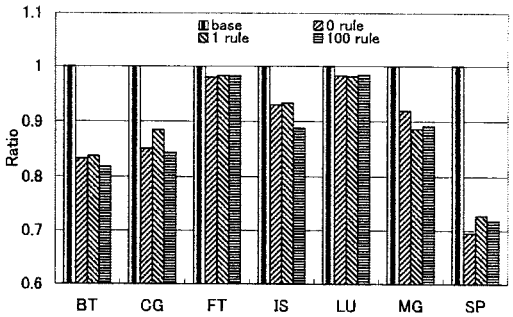

図 7 接続点切替手法による NPB の性能

た結果を**表 5** に示す。ルール数の増加に伴ってパケット毎のマッチング処理が多くなり、System CPU の使用率が増加することがわかる。ルール数 100 の TCP スループットが低下しているが、送信側の CPU 負荷が 100%になっているのが原因である。netfilter を動作させた場合、すべてのパケットに対してマッチングを行うので、経路切替しないユーザのパケットも CPU 負荷を増加させ、その影響を受ける。経路切替したユーザと、しないユーザ間では公平性が保たれており、同時にフローを発生させた場合のスループットは半分ずつであった。

接続点切替手法ではルール数に関係なく RTT やスループットは一定である。iRule ではフローの最初のパケットに対してのみマッチングを行い、その後の同一フローのパケットはハードウエアにより処理されていると考えられるので、ルール数には影響されない。RTT はホップ数が 3 なのでノード内切替手法の場合より大きくなる。TCP の結果はワイヤレートと比べて 4%程度低いスループットとなっている。これは ACK 応答パケットの転送量に相当するものと考えられる。BIG-IP の L3 スイッチ全体のスループットが 1Gbps 程度に抑えられているため、ACK 応答パケットの転送量分だけスループットが減少する。

4.4 NPB3.2 による評価

アプリケーションへの影響を調べるため、NPB3.2 のクラス B によるベンチマークを計測した。NPB3.2 は GridMPI を使用して実行し、2 つのクラスタ間を IMPI[10] で通信している。CG、FT、IS、LU、MG はクラスタ A、B の各 4 ノードで起動し計 8 プロセスで実行した。BT、SP は n^2 のプロセス数が必要なので、クラスタ A の 4 ノード、クラスタ B の 5 ノードで起動し計 9 プロセスで実行した。各々のベンチマークを 3 回実行して、その中で最も良かった数値を計測結果としている。

計測結果を**図 6**、**図 7** に示す。図 6 はノード内切替手法を用いた場合の結果で、ルール数 0 の値で正規化している。同様に図 7 は接続点切替手法を用いた場合の結果を示す。図 7 は図 5 の構成で Router-3 で L3 接続した測定値で正規化している。これを図 7 の中で base と示す。それぞれの基準となる Mop/s 値を**表 6** に示す。

ノード内切替手法で 10 ルールを設定した場合、ベンチマーク値の低下は全体で 1 から 3%程度であった。1 ルールで 1 ユーザの経路切替えができるので、PC クラスタの場合には 10 ルールで十分なユーザ数の経路切替が可能と考える。100 ルールを設定した場合は IS で最大 10%の性能低下が起こり、他のベンチマークでも性能低下が目立つようになる。ノード内切替手法では、全パケットを対象にマッチング処理が発生するために、CPU の負荷が増加した影響と考える。

表 6 ルール数 0 の NPB 性能 (Mop/s)

	BT	CG	FT	IS	LU	MG	SP
ノード内切替	5485	1710	2161	105	3648	3679	2365
接続点切替	5343	1629	1645	70	3656	3398	2241

接続点切替手法を用いた場合には、BIG-IP の L3 転送に 1Gbps のボトルネックが存在するので性能悪化は避けられない。この影響でベンチマークの種類により、性能が 20%から 30%も悪化している。クラスタを接続している L2 スイッチと BIG-IP でパケットのドロップが大量に発生しているので、ベンチマーク性能は安定しない。ルール数の増加とベンチマーク性能は無関係であると考える。

5. 議　　論

従来、複数のネットワークに接続するノードは、ネットワーク毎に異なるアドレス空間を割付け、インタフェースに異なるアドレスを設定していた。単純な経路設定を用いているので、クラスタ内の全ジョブが同じ経路を持つことになり、ユーザ単位のルーティングが出来なかった。また、送信側ノードは送信先ノードのどのアドレスに送信するかでネットワークを選択していた。利用するネットワークを考慮してアプリケーションの設定を変更する必要があった。

これに対し本切替手法では、ユーザ個別の経路表を適用することで、ユーザ単位のルーティングを実現している。複数のジョブが実行され同じ宛先アドレスに同時に送信したとしても、ユーザ毎に異なる経路を選択できる。また、経路を変更しても宛先アドレスを変更する必要がないため、複数ネットワークを単一のアドレスで運用できるようになる。つまり、アプリケーションがどのネットワークを利用するかをユーザは意識しなくてもよい。

ノード内切替手法では、接続点に新たな機器を必要としない。しかし、ノードの CPU 負荷を上げることでアプリケーションに影響を与える欠点がある。切替手法に netfilter と iproute2 を用いたが、これによりユー

ザ単位に経路切替した例はこれまで見たことがなった。なお、本切替手法を利用する場合には、以下の2点に注意が必要である。

- sshd などデーモンプロセスとの通信
  デーモンで起動しているプロセスは、オーナが異なるので経路切替の対象にならない。例えば、sshd は root で起動されており、接続を受付けて子プロセスを生成し、ユーザの認証が完了した後にオーナをユーザに変更する。この間の通信は root がオーナなので経路切替の対象とならず、従来の経路が適用される。
- ping による疎通確認
  疎通確認のために使用される ping は ICMP を利用している。ICMP の応答パケットは kernel のプロトコルスタックで生成される。このパケットはプロセス番号やオーナを持たないので経路切替の対象にならない。また、ping コマンドは setuid で root 権限で実行される。従って、経路切替したネットワークでは ping による疎通確認ができない。

一方、接続点切替手法はノードに全く影響を与えない。しかし、接続点に比較的高価な機器が必要である。クラスタの各ノードから出るトラフィックはこの機器に集中する。従って、この機器には広帯域なインタフェースと高性能なパケット処理能力が必要とされる。これらの性能が不足するとこの機器がボトルネックとなり、フローに影響を与えることになる。本手法は、インターネットへマルチホーミング接続する方式と同じである。しかし、マルチホーミング接続は、ユーザやアプリケーションの要求とは関係なく、ネットワークの負荷分散や耐故障性向上を目的として経路切替をしている点が異なる。ユーザの要求に従った経路切替のためにロードバランサ装置を使用した例は筆者らの知る限り他に無い。

6. おわりに

本報告では、経路切替手法として、netfilter でマーキングし、この値に従って iproute2 で個別の経路表を用いるノード内切替手法と、BIG-IP のフロー単位のトラフィック制御機能による接続点切替方法を比較検討した。ユーザジョブの実行開始前と終了時に本手法で経路を切換えることで、ユーザ単位に経路を選択でき、単一アドレスで異なる経路を利用できることを示した。これらの切替手法が並列計算アプリケーションの性能面に与える影響を調べるため、NPB3.2 による性能評価を行った。

ノード内切替手法では、全パケットを対象にマッチング処理が発生するために CPU の負荷が増加する。10 ルールを設定した場合には NPB ベンチマークは 1 から 3%低下した。PC クラスタで同時に経路切替が必要なユーザ数は 10 程度に収まると考えられるので、経路切替えによる影響はこの程度と推測する。ただし、経路切替が必要ないユーザジョブの通信でも CPU 負荷を増大させる。また、経路切替が必要ないユーザジョブも CPU 負荷の影響を受ける。通信ミドルウエアでマーキングできれば、マッチングによる CPU 負荷は問題なくなると考える。

接続点切替手法ではノードに影響を与えないが、BIG-IP の L3 スイッチ性能に 1Gbps のボトルネックがあるために、NPB ベンチマークは大きく低下した。接続点切替手法でユーザ単位に経路切替するには、パケットがノードから出る前に ToS へマーキングする必要がある。これは、ノード内で netfilter や通信ミドルウエアでマーキングすることで対応できる。

今後は、動的帯域提供ネットワークに接続されたクラスタの経路切替に本手法の適用を検討していく。

謝　辞

BIG-IP の使用に関してご助言をいただいた NTT 未来ねっと研究所の平野章氏、築島幸雄氏に深く感謝します。本研究にご助言、ご協力していただいた数理技研の大久保克彦氏、産総研の児玉祐悦氏、高野了成氏に深く感謝します。

本研究の一部は、文部科学省科学技術振興調整費「グリッド技術による光パス網提供方式の開発」による。

参考文献

1) 竹房, 林, 築島, 岡本, 柳田, 宮本, 平野, 鮫島, 中田, 谷口, 工藤. ミドルウェア連携による計算・ネットワーク資源の日米間グリッドコアロケーション実験. 情報処理学会研究報告 2007-HPC-109, HOKKE'07, pp. 281-286, 2007.
2) G-lambda Project. http://www.glambda.net
3) 工藤, 松田, 手塚, 児玉, 建部, 関口. VLAN を用いた複数パスを持つクラスタ向き L2 Ethernet ネットワーク. 情報処理学会論文誌, Vol.45, No.SIG6(20040515), pp. 35-44, 2004.
4) Message Passing Interface Forum. MPI: A Message-Passing Interface Standard, May 5, 1994. University of Tennessee, Knoxville, Report CS-94-230, 1994.
5) BIG-IP. http://www.f5.com
6) D. Bailey, T. Harris, W. Saphir, R. van der Wijngaart, A. Woo, and M. Yarrow. The NAS Parallel Benchmarks 2.0. International Journal of Supercomputer Applications, 1995. http://www.nas.nasa.gov/Software/NPB
7) 中田, 竹房, 大久保, 工藤, 田中, 関口. グローバルスケジューリングのための計算資源予約管理機構. HPCS2007 論文集, pp. 127-134, 2007.
8) 松田, 石川, 鐘尾, 枝元, 岡崎, 鯉江, 高野, 工藤, 児玉. GridMPI Version 1.0 の概要. 情報処理学会 2005-HPC-103, SWoPP'05, pp. 281-286, 2005.
9) GridMPI Project. http://www.gridmpi.org
10) W. L. George, J. G. Hagedorn, and J. E. Devaney. IMPI: Making MPI Interoperable. Journal of Research of the National Institute of Standards and Technology, Vol.105, N.3, May 2000.