6ZA-7

大規模ネットワークにおける効率的なバンド幅マップ構築アルゴリズム

長沼 翔† 田浦 健次朗†

†東京大学大学院情報理工学系研究科

1 はじめに

多数の計算機をネットワーク接続し計算処理を分散させて行う並列分散計算が盛んに行われるようになって以来，大容量データに対する大規模な計算を必要とする言語処理や画像処理などの様々な分野では並列分散処理は必須の技術となっている．並列分散システムを扱う際，ネットワークリンクのバンド幅の情報を知るということはシステム管理者だけでなくユーザにとっても重要である．特に上に挙げたような大容量データを扱う計算を実行したいユーザは，計算機間の通信が全体の処理のボトルネックとならないように，バンド幅の値とよく照らし合わせて負荷分散やデータレプリカの仕方を考える必要がある．他にも集合通信などの，並列計算における基本的な操作であってもバンド幅を考慮して最適化べきだという研究も多くなされている [6][2].

しかし，分散環境の規模が大きくなるに従いネットワーク構成が不均質になりがちで複雑な接続形態や大小のバンド幅が混在することが一般的であり，そのような環境上では既存のバンド幅測定プログラムで特定のリンクのバンド幅の値を把握することは難しい．

本研究ではバンド幅マップ構築手法を提案する．バンド幅マップとはネットワークトポロジーの全てのエッジにバンド幅の情報を振ったものである．ところで，ある環境におけるバンド幅の別の表現法としてよく用いられるものにバンド幅行列がある．これは単に全ホストペアの End-to-End バンド幅を列挙したものであり，取得は容易である．しかしこれでは，特に WAN にまたがるヘテロな分散環境では，通信の衝突や中間のボトルネックリンクを知ることができない．本手法は，複数ペアの測定を協調させることで中間エッジのバンド幅測定を可能にし，衝突しない測定を並列に行うことで効率性を実現する．

2 関連研究

Burger らによる TopoMon は，環境中の全ホストペア間でのバンド幅測定の結果と traceroute の結果を組み合わせることによって，グリッド環境におけるバンド幅マップに近いものを得ている [1]．TopoMon では基本的に 1 対 1 のバンド幅測定しかしておらず，したがってそれらのホスト間に複数のスイッチやリンクが存在する場合にそれらの最小のバンド幅の値しか結果として得られない．そのような測定をベースとして出力されたバンド幅マップは，我々の目標とするバンド幅マップより情報量の少ないものである．また，TopoMon では測定を逐一進めていく仕組みになっているため，環境の拡大に対してスケールしない．

それらの問題を解決するために，多対多のバンド幅測定を基本とし全体の処理を並列に進めることができる bhtree を我々は過去に実装した [3]．実験では約 300 ホストの広域分散環境において 100 秒程度でバンド幅マップの構築ができている．しかし，bhtree では想定するネットワーク構造を無向のツリーに限定しており，特に WAN のような非対称バンド幅や循環構造を持つネットワークにおいては意味のある結果を出すことはできない．

3 提案手法

提案する手法はネットワークトポロジーデータを入力として，そのバンド幅マップを出力とする．トポロジーデータは与えられていると仮定して話を進めるが，traceroute を用いた手法やホスト間遅延から接続形態を推定する手法などが既に提案されており，トポロジーデータは容易に取得することができる [5][4].

3.1 バンド幅測定

まず，基本となるバンド幅測定について述べる．この結果とトポロジーを組み合わせ，結果を出力する．

我々は，一つのリンクの一方向のバンド幅を以下のように定義する．あるリンクの一方向に着目する．パケットがそのリンクをその方向に流れるような全ての通信ペアらでバンド幅測定を一斉に実行し，それぞれが観測したバンド幅の和を，それのバンド幅とする（図 1）.

具体的には以下のようである．分散システム中に N のホストが存在していれば，向きも考えた通信ペアは $N(N-1)$ 組あり，つまりデータストリームのパスは $N(N-1)$ 通りある．データストリームパスがどのようになっているかは，通信ペアと入力のトポロジーデータを参照すれば抽出することができる．それらのパスのうち，パス中に着目しているリンクが含まれるパスを列挙する．それらパスを流れるデータストリームを発生させるような複数通信ペアのバンド幅測定を一斉に行い，それぞれの結果の和が着目しているリンクのバンド幅である．

我々がこのように定義した意味としては，1 対 1 のバンド幅測定だけでは注目しているリンクのバンド幅

An Efficient Algorithm for Building Bandwidth Map of Large-Scale Networks
Sho Naganuma† and Kenjiro Taura†
†University of Tokyo

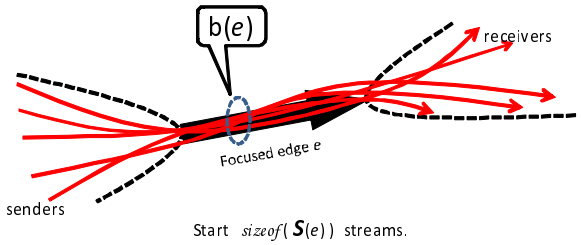

図 1: バンド幅の定義:
そのリンクを通るストリームを全て束ね，
観測されたバンド幅の和をそのリンクのバンド幅とする．

を使いきれず正しい測定が行えないため，多対多のバンド幅測定を行ってそのキャパシティを飽和させるということである．ここで，列挙された全ての通信ペアでストリームを発生させたとしても飽和しないような，大きいバンド幅のリンクが存在する可能性もある．そのようなリンクは，システム全体でネットワークに負荷をかけたとしても使い切ることのできないバンド幅として，バンド幅を無限大とするかまたは上述の定義で得られた和を答えとしても実用上問題はない．

この定義に従って，システム中のリンクを一つずつ決定して行けば，目標とするバンド幅マップを得ることができる．しかし，バンド幅一つを決定する度に過度なネットワーク負荷をシステムにかけてしまうことになるので，実装では列挙したペアのストリームを一つずつ流し，バンド幅が飽和したと判断できた時点でそのリンクのバンド幅を決定することにしている．どのような場合に飽和したと判断するかは，あるストリームを流そうとしても全体の和に変化が起きなかった場合である．それ以降の列挙されたペア間のストリームを加えたとしても和に変化が起きないことは明らかであるので，そこで飽和したとし，以降のストリームは仮想的に流しているとすれば上述の定義に合致した測定と言うことができる．

3.2 並列化

上述のバンド幅測定を入力トポロジーの全てのリンク対して適用すれば，我々の求める結果が得られるが，環境の拡大に対してスケールする手法とは言えない．既存研究ではシステム中のホスト数 N に対して全体実行時間が $O(N^2)$ であり，この問題を解決するためには並列化が不可欠である．

我々の提案では，全体のリンクをいくつかの集合に分け，それぞれを並列に測定するという方針である．いくつかの集合に分ける際に必要な条件は，それぞれが閉じたネットワークであり，自分の集合中のストリームが，ある別の集合中に発生するいかなるストリームとも衝突しない保証がされていることである．このような条件の下，できるだけ並列度を上げられる様，細かい集合にリンク集合を分割することが望ましい．実際には，ヒューリスティックな考えから，エンドホストから近く，ホップ数の少ないリンク同士を優先してリンクのグループ分けを行う．

このようにして分割されたリンクの集合，言い換えれば分割されたネットワークに対して，それぞれに含まれる全リンクに上で定義したバンド幅測定を行っていくことで，並列化が実現される．分割された各ネットワークについてバンド幅マップが構築し終わったなら，それを元の通りに組み合わせれば，プログラム全体の出力となるバンド幅マップを得ることとなる．

4 おわりに

本稿では大規模ネットワークに適用できる効率的なバンド幅マップ構築アルゴリズムの概念を紹介した．このアルゴリズムでは，多対多のバンド幅測定の和を用いることで，これまでの 1 対 1 のバンド幅測定では知ることのできなかった情報を得，実用時間で完了させるよう並列化の仕組みを取り込んでいる．今後は，このアルゴリズムの実装を実際の広域分散環境で実験・評価をし，実用的であることを示し，有用なソフトウェアを提供することを目指す．

参考文献

[1] Mathijs den Burger, Thilo Kielmann, and Henri E. Bal. Topomon: A monitoring tool for grid network topology. *In ICCS '02: Proceedings of the International Conference on Computational Science-Part 2*, pp. 558–567, 2002.

[2] Thilo Kielmann, Rutger F. H. Hofman, Henri E. Bal, Aske Plaat, and Raoul A. F. Bhoedjang. Magpie: Mpi's collective communication operations for clustered wide area systems. *Symposium on Principles and Practice of Parallel Programming*, pp. 131–140, May 1999.

[3] Sho Naganuma, Kei Takahashi, Hideo Saito, Tkeshi Shibata, Kenjiro Taura, and Takashi Chikayama. Improving efficiency of network bandwidth estimation using topology information. *Symposium on Advanced Computing Systems and Infrastructures (SACSIS)*, pp. 359–366, June 2008.

[4] Tatsuya Shirai, Hideo Saito, and Kenjiro Taura. A fast topology inference — a building block for network-aware parallel computing. *In Proceedings of the 16th IEEE International Symposium HPDC 2007*, pp. 11–21, June 2007.

[5] Rich Wolski, Neil T. Spring, and Jim Hayes. Implementing a performance forecasting system for metacomputing: the network weather service. *Proceedings of the 1997 ACM/IEEE conference on Supercomputing*, pp. 1–19, 1997.

[6] Shota Yoshitomi, Ken Hironaka, and Kenjiro Taura. An adaptive gather algorithm avoiding contention. *Symposium on Advanced Computing Systems and Infrastructures (SACSIS)*, pp. 71–78, May 2009.