分散ストリーム処理システムにおける効率的なデータ配信基盤に関する研究

宮城亮太[†]　川島英之[††, †††]　北川博之[††, †††]
筑波大学第三学群情報学類[†]　筑波大学大学院システム情報工学研究科[††]
筑波大学計算科学研究センター[†††]

1. はじめに

センサデバイスや監視カメラから連続的に配信されるデータストリームを処理する基盤技術としてストリーム処理エンジンが注目を集めている．ストリーム処理エンジンを高性能化，高信頼化するために，それを分散環境に適応させた分散ストリーム処理エンジン(DSPE)についての研究も進められている．ストリーム処理エンジンは実行時にオペレータから構成されるオペレータ木を構築する．DSPE では負荷分散のために，それらのオペレータを異なるマシンに配置する．このとき，オペレータ間通信方式を工夫することにより，その性能を高められる可能性がある．そこで本研究では DSPE 内部において，応用層で動作する，効率的なデータ配信基盤を提案する．

2. 既存手法

2.1. 直接リンク法

直接リンク法とは，マシンに配置されたオペレータ間でコネクションを張ることである．マシン内に同時に動作するオペレータ数が少ない場合，優れた性能を示す．しかし，DSPE では同一マシンに多数のオペレータが配置されることがある．その場合，特定のマシン間において多数のコネクションが張られるため，負荷が増大し，処理能力が低下する．即ち大規模オペレータ環境では性能が劣化する．

2.2. 集約リンク法

集約リンク法は複数の送信要求をまとめて行う手法である．これは応用層データ配信システムの一つである Pub/Sub システム[2]などで用いられている．この手法ではコネクション数を減らすことが可能となるため，大規模なオペレータに対して性能が劣化しにくい．一方，オペレータ数が少数の場合には直接リンク法に対して性能が劣化する．

3. 提案

本研究では，従来手法の利点を兼ね備えた，DSPE に組込む専用配信基盤を構築する．

3.1. 専用配信基盤

応用層配信システムでは，クライアントと配信システムが別プロセスになるため，両者の通信にはプロセス通信が必要になる．この処理は，多量のデータを絶えず処理し続け，高速な処理が求められる DSPE においては望ましくない．そこで，本研究では DSPE と同一プロセス空間で実行される，専用の応用層配信基盤を構築する．配信基盤とオペレータの通信はメモリアクセスにより実現される．

提案配信基盤は初期化処理において，オペレータが動作する全マシン間に TCP コネクションを確立する．次に，グループ通信（3.2 節）に使うためのデータ送受信用バッファ領域を確保する．マシン数が n の場合，各マシンにおけるコネクション数と送信用共有バッファ領域数は n-1 でなる．データ送受信は共有バッファ領域を利用し，送受信処理は専用スレッドが非同期的に実行する．オペレータと送受信処理は排他制御を行う．オペレータは送受信のために，図 1 に示される関数を用いる．各関数では，オペレータ木 ID とオペレータノード番号を指定する．各関数において，tid はオペレータツリーの ID，dst は送信先オペレータ，me はデータを受信するオペレータ，tpl は実際に送信されるタプル，szt はタプルのデータサイズを示す．

3.2. グループ配信

データ配信時には効率化のためにグループ配信を行う．各マシン間で送信 TCP コネクション

An Efficient Data Delivery Infrastructure for Distributed Stream Processing Systems
Ryota Miyagi[†], Hideyuki Kawashima[††, †††], Hiroyuki Kitagawa[††, †††]
[†]College of Information Sciences University of Tsukuba
[††]Graduate School of Systems and Information Engineering University of Tsukuba
[†††]Center for Computational Sciences, University of Tsukuba

```
ssend(int tid, int dst, char *tpl, int szt)
srecv(int tid, int me, char *tpl, int szt)
```

図1:配信基盤インターフェース

は一つのみ確立されている．このオペレータ共有のコネクションを利用し，配信先が同一マシンである，複数のオペレータによる処理結果を一括して配信する．配信データは図 1 で示した ssend 関数により，配信先ごとに用意されているバッファ領域に保持される．保持されているデータの総計が閾値を超えるか，あるいは一定時間が経過すると，共有バッファ領域内のデータはまとめて送信される．受信されたデータは，オペレータ木ごとの受信用バッファ領域に保持される．その後，受信側は図 1 の srecv 関数を利用することで，指定したオペレータ木，ノードあてのデータを取得する．送受信の際には通信速度の向上のため zlib[3]を利用し，圧縮および解凍処理を実装した．

4. 実験

提案した配信基盤の有用性を示すために，提案した配信基盤と，直接リンク法（2.1 節）の比較実験を以下の 2 つ環境で行った．タプルサイズは 100 バイト固定とし，送信タプル数はオペレータ毎に 1000 件とした．実験環境には下記の 2 つの場合を用いた．

- 実行環境 1（近距離通信）：InTrigger[4]千葉拠点内の 2 マシン間での通信
- 実行環境 2（遠距離通信）：InTrigger 千葉拠点のマシンと InTrigger 京都拠点のマシン間での通信

また，配信基盤のパラメータとして，送信スレッショルドは約 1.3KB，送信バッファサイズは 108KB とした．オペレータ数を 1, 10, 100 と配置した場合に，上記のデータ全てを配信するのに経過した時間を測定した．配信基盤はノード間で 1 本のコネクションのみを使用した．一方，比較対象である直接リンク法では，配置オペレータの数と同数のコネクションがノード間で張られた．例えばオペレータ数が 100 の場合には TCP コネクションは 100 本確立された．

実験結果を図 2 と図 3 に示す．縦軸は処理時間(秒)を表し，横軸は登録オペレータ数を表す．青線は直接リンク法の結果を表し，赤線が提案方式を利用した場合の結果を表す．いずれの実行環境においても，オペレータ数 1, 10, 100 全ての場合において提案配信基盤のほうが処理時間を圧倒的に短縮した．実行環境 2 では，どちら

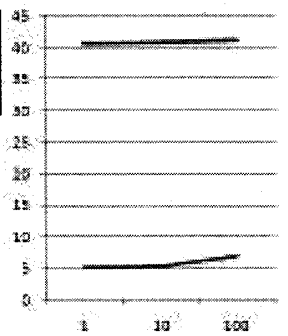

図2:実行環境1

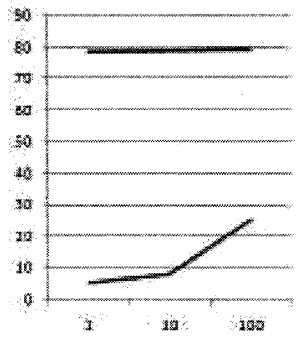

図3：実行環境2

の通信手法の場合も処理時間の増加がみられた．これは地理的距離が原因と考えられる．
処理時間が短縮された理由は，グループ配信で送受信するデータをまとめて扱い，送受信の際に圧縮を行ったことが主たるものであろうが，TCP コネクション数を減らして OS への負荷を下げたことも影響したと考えられる．また，配信基盤はオペレータ数が 100 の場合に処理時間が増加することが示された．今後はコネクション数をオペレータ数により動的に調整する等の改善手法に今後取り組むつもりである．

5. まとめと今後の課題

本研究では，DSPE における効率的配信基盤を提案し，直接リンク法に対する優位性を示した．今後はより大規模な実験を行うと共に，コネクション数の動的変化をさせることでさらなる処理能力の向上について検討したい．

謝辞

本研究の一部は科学研究費補助金基盤研究(A)(#21240005)，科学研究費補助金若手研究(B)(#20700078)による

参考文献

[1] Arasu, A. and Babcock, B. and Babu, S. and Datar, M. and Ito, K. and Nishizawa, I. and Rosenstein, J. and Widom, J. STREAM: The Stanford Stream Data Manager. Proceedings of the 2003 ACM SIGMOD international conference on Management of data.
[2] Antonio Carzaniga, David S. Rosenblum, Alexander L.Wolf. Design and Evaluation of aWide-Area Event Notification Service. ACM Transactions on Computer Systems, 19(3):332-3382, 2001.
[3] zlib: http://www.zlib.net/
[4] InTrigger プラットフォーム http://www.intrigger.jp