サーバシステムの性能データ転送における効率化手法の実装と評価

理学専攻・情報科学コース 飯山 知香 (指導教員：小口 正人)

1 はじめに

近年，多数台サーバの共有利用などに関する需要が増加している．これらのサーバの負荷分散などをリアルタイムで行うためには，各サーバの性能データ分析を低オーバヘッドで行う必要がある．そこで本研究では，ターゲットサーバからデータ解析用サーバへのデータ転送時のオーバヘッドを考慮した効率的な転送手法の開発を目的としている．性能データ収集の手法として一般的な perf[1] に合わせた形式で，転送する全ての CPU コアのデータを 1 つのコアで転送する 1 コア転送版を使用した予備実験では，コア数が増えると転送処理が追い付かなくなることが分かった．本研究では，転送効率化手法として，転送処理並列化，データ解析用サーバ性能改善，転送タイミング制御を行い，その効果を評価する．

2 実験

2.1 実験概要

環境構築として，データ収集用サーバ (以下収集サーバ) とデータ解析用サーバ (以下解析サーバ) を用意した．評価環境概要を図 1 に示す．収集サーバには，Linux Kernel の標準的なイベントデータ収集/トレース機能のフレームワークである perf を使用した．解析サーバには，収集した時系列データを管理する時系列 DB として InfluxDB[2] を使用した．事前に収集した stress[3] 実行時の perf record のデータをデータ転送プログラムを用いて転送し，解析サーバ上の InfluxDB に格納した．解析サーバには，構成が異なる 2 台のサーバを用いて分析を行った．以降，CPU コア数，メモリ容量が多いサーバを高性能解析サーバ，少ないサーバを低性能解析サーバと記載する．各サーバのネットワーク構成図を図 2 に示す．収集サーバと高性能解析サーバの構成を表 1，低性能解析サーバの構成を表 2 に示す．各使用技術のバージョンは，perf が 3.10.0，stress が 1.0.4，InfluxDB が 1.8.3 である．

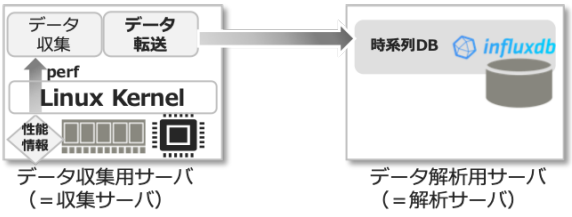

図 1: 評価環境概要

各実験の詳細を説明する．転送タイミング制御は，CPU コア数 n での転送において各コア i で転送前に i/n 秒 sleep し，コアごとの転送タイミングを 1/n 秒ずつずらすことで実装した．なお，各実験でのデータ転送時間の計測は 6 回行い，初回転送のオーバヘッドによる影響を避けるため，2〜6 回目の結果のうち，全コアの平均のデータ転送時間が中間のものを用いた．

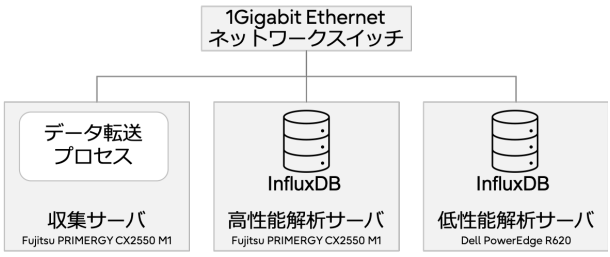

図 2: ネットワーク構成図

表 1: 収集サーバ・高性能解析サーバ環境

機種名	Fujitsu PRIMERGY CX2550 M1
CPU	Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz (× 2CPU)
コア数	14
メモリ	128GB
OS	CentOS 7

表 2: 低性能解析サーバ環境

機種名	Dell PowerEdge R620
CPU	Intel(R) Xeon(R) CPU E5-2603 v2 @ 1.80GHz (× 1CPU)
コア数	4
メモリ	16GB
OS	CentOS 7

2.2 結果

2.2.1 データ転送時間

1 コア転送版でのデータ時間の結果を図 3 に示す．図 3 より 1 コア転送版では，8 コア以上のデータ転送に 1 秒以上かかっていることが読み取れる．

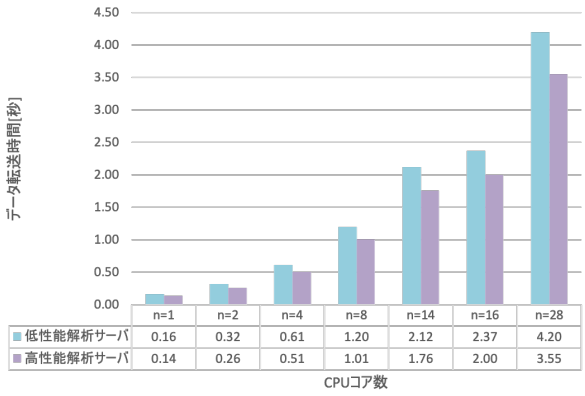

図 3: データ転送時間 (1 コア転送版)

並列転送版でのデータ転送時間の結果を図 4 に示す．

図中の各点がコアを表している．転送処理並列化を行った結果，28 コアのデータ転送でも平均 0.51 秒で転送可能であり，並列化による効果が得られていることが分かる．しかし，各 CPU コア数 n において，並列転送版のデータ転送時間が 1 コア転送版の 1/n よりも長くなっていることから，並列化にはオーバヘッドがあることが推察される．例えば n=8 において，並列転送版の平均転送時間 (0.24 秒) は 1 コア転送版の 1/8(0.15 秒) よりも長く，約 1/5 となっている．また，並列転送版では，転送を実行するコア間で転送時間にばらつきがあり，CPU コア数が増えるとばらつきも同時に大きくなっていることが読み取れる．この要因として，コア数が増えて InfluxDB 側の CPU 負荷率が増加した場合に，InfluxDB 側の処理が追いつかなくなっている可能性が挙げられる．

そこで，解析サーバ性能改善を行なった．低性能解析サーバにて，転送を実行するコア間で転送時間にばらつきがあり，CPU コア数の増加に伴いばらつきも増加していたが，高性能解析サーバではばらつきが減っていることが読み取れる．これは，InfluxDB サーバのコア数が増加したことにより，InfluxDB 側の CPU 負荷率が平準化されたためと推察できる．ただし，高性能解析サーバにおいてもばらつきが残っていることから，同時転送するコア数が増加した場合は，解析サーバ側の増強だけでは改善できないボトルネックがあると考えられる．

そこで，転送タイミング制御を加えて計測を行なった結果，低性能/高性能解析サーバ使用時ともに，CPU コア数の増加に伴う転送時間および転送時間のばらつきの増加が抑えられていることが読み取れる．また，低性能解析サーバ使用時においても，転送タイミング制御を行うことで，高性能解析サーバ使用時と同等の転送時間で転送できることが分かる．

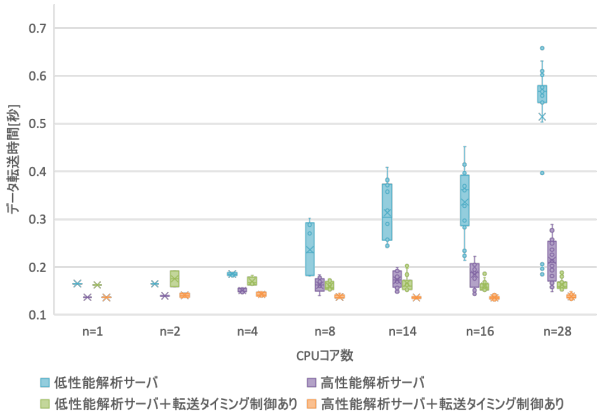

図 4: データ転送時間 (並列転送版)

2.2.2 データ転送処理部分の CPU 負荷率

データ転送処理部分の CPU 負荷率は CPU コア数 n=14 で計測を行った．これは，コア数が多いほどオーバヘッドは分かりやすく，また表 1 に示すように収集サーバの 1CPU あたりのコア数は 14 であるためである．結果を図 5 に示す．図中の各点がコアを表している．図 5 より，解析サーバ性能改善を行なった結果，高性能解析サーバ使用時では低性能解析サーバ使用時と比べてデータ転送時間は短く，CPU 負荷率は高くなっていることが読み取れる．また，低性能/高性能解析サーバ使用時ともに，コアごとに転送時間にばらつきがあり，転送時間が長くなるほど CPU 負荷率は低くなっている，つまり Idle 時間が発生していることが推測できる．さらに，高性能解析サーバにて，CPU 負荷率が最大 67.4%程度と高くなっている．これによる同一 CPU コア上で動作する処理への影響が大きければ，転送用のコアを分けるなどの対応が必要になると考えられる．

そこで転送タイミング制御を加えた結果，低性能/高性能解析サーバともに，転送時間および CPU 負荷率のばらつきは減少していることが読み取れる．ただし，転送時間が短くなるにつれ CPU 負荷率が高くなる傾向には変化はない．このことから，転送タイミング制御によって転送中の Idle 時間が減少し，転送効率が改善されたことが推測できる．

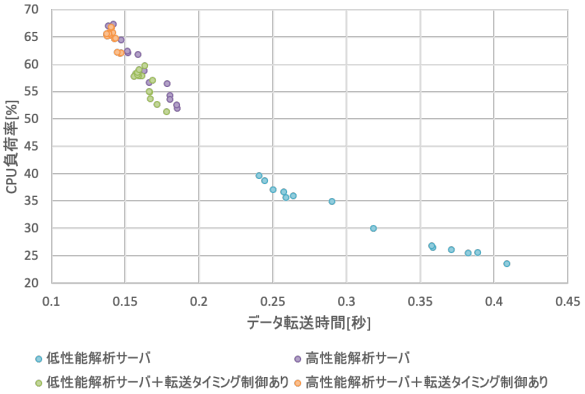

図 5: データ転送処理部分の CPU 負荷率 (n=14，並列転送版)

3 まとめと今後の予定

本報告では，データ収集を行うサーバと解析を行うサーバが分割された環境を想定したデータ転送の効率化手法として，転送処理並列化と解析サーバ性能改善，転送タイミング制御を行なった．その結果，転送タイミング制御はデータ転送時間の短縮において有用な手法であることが分かった．今後は，転送プログラムの改善として，転送側で CPU 負荷の低いコアを検出してそのコアに転送させる処理の実装を行う予定である．また，転送タイミング制御におけるより効率の良いずらし方についても検討していく．

謝辞

本研究の一部はお茶の水女子大学と富士通株式会社との共同研究契約に基づくものであり，JST CREST JPMJCR22M2 の支援を受けたものである．

参考文献

[1] perf. https://perf.wiki.kernel.org/index.php/Main*page*.

[2] influxdb. https://www.influxdata.com/products/influxdb/.

[3] stress. https://linux.die.net/man/1/stress.