

マルチキャスト通信を利用した
ネットワーク分散型データ収集システムの高効率化に関する研究
Research on a High Efficiency Network Distributed Data Acquisition System
with Multicast Communication

繁元 貴大† 長坂 康史†
Takahiro Shigemoto Yasushi Nagasaka

1. はじめに

高エネルギー物理学実験などの測定器実験で使用するデータ収集システムでは、大規模なものでは測定器に接続された数百台ものコンピュータから測定データを収集する。さらに近年、実験規模の拡大に伴い、測定器の大規模化、収集データの大容量化が進んでおり、膨大な量のネットワークトラフィックが発生している^[1]。その結果、データ収集システムでは、システム制御やモニタリングのデータ等、膨大な通信データを同時に処理しなければならなくなった。しかし、この複雑なトラフィックは輻輳を招くことになり、システム全体の効率を下げる原因となっている。

データ収集システムの開発の効率化を目的としたデータ空間共有型通信フレームワーク^[2]では、データの収集におけるネットワーク通信システムに一对一のユニキャスト通信を利用している。本研究ではデータ空間共有型通信フレームワークの機能を拡張し、収集データ以外の一部のデータ通信にマルチキャスト通信を導入することで、データ収集時の通信の高効率化を目的とする。

2. データ空間共有型通信フレームワーク

本フレームワークはリポジトリスタイルをベースとしており、処理を行う単位であるコンポーネントがデータを管理するリポジトリにアクセスすることで通信を行う。リポジトリ内にはデータとともに、そのデータと関連付けられたキーが保存されており、コンポーネントからはそのキーを指定してデータ操作を行う。この操作には **Get**, **Put**, **Delete** が用意されており、データの取得、作成、更新、削除等を行うことが出来る (図 1)。

Get 機能はデータの取得を行う機能である。キーを指定してリポジトリへリクエストを送ると、そのキーに対応するデータをレスポンスとして返す。

Put 機能はデータの作成と更新を行う機能である。キーの指定と共にデータをリポジトリへ送り、指定したキーに対応するデータの作成、または更新が行われる。

Delete 機能はデータの削除を行う機能である。キーを指定してリポジトリへリクエストを送り、キーに対応するデータを削除する。

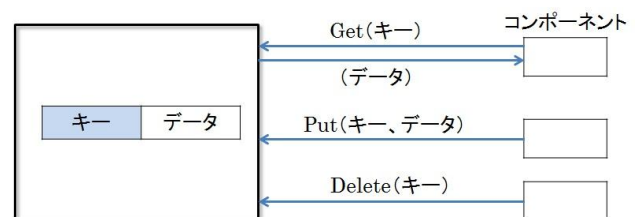


図 1 Get, Put, Delete 機能

リポジトリは、データの作成、更新、削除のいずれかが起こると、それをイベントとして捉える。イベントが発生した際に通知、同期を行う機能として、**Notify**, **Link** が用意されている (図 2)。

Notify はイベントが発生した際に通知を行う機能である。必要なデータのキーを指定してリポジトリに通知設定リクエストを送ると、**NotifyID** をレスポンスとして返す。リクエストしたリポジトリでデータの作成、更新、削除のいずれかが発生した場合、そのイベント内容と **NotifyID** が通知される。また、発生したイベントが作成、更新の場合は、そのデータも通知される。

Link はキー同士を関連付け、データの同期を行う機能である。同期は、リンク元キーからリンク先キーへの一方向で行われる。リンク元となるデータのキーを指定してリポジトリにリンク先リポジトリ、リンク先キーを同期設定リクエストとして送る。リンク元キーに対応するデータに作成、更新、削除のいずれかが発生した場合、それと同様のイベント処理がリンク先キーに対応するデータにも行われる。

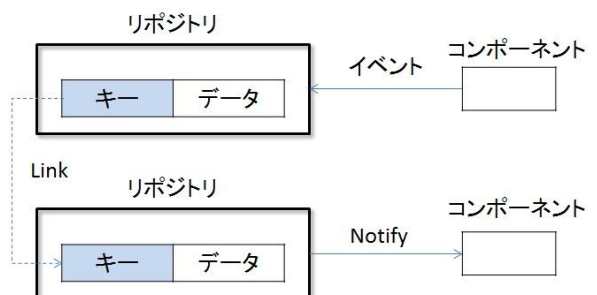


図 2 Notify, Link 機能

† 広島工業大学 大学院工学系研究科,
Hiroshima Institute of Technology
Graduate School of Science and Technology

3. マルチキャスト通信のための拡張

本研究では、新たに **Multi** という通知機能を追加する。従来のフレームワークはユニキャスト通信を用いているため、異なるコンピュータ間では複数のコンポーネントへイベントによるデータの変更を同時に通知することが出来ない。そこで、**Multi** ではマルチキャスト通信を用い、複数のコンポーネントへの同時通知を可能とする。

具体的には、受信側コンポーネントがキーを指定してリポジトリに **Multi** リクエストを送る。そのリポジトリにイベントが発生すると、リクエストを受け付けたコンポーネントに対してマルチキャスト通信を行い、同時にイベントの発生を通知する。

図 3 に、**Multi** による通知設定と実際の通知の流れをシーケンス図で示す。リポジトリからのイベント通知を必要とするコンポーネントからリポジトリに **Multi** リクエストが送信され、通知設定が行われる。このリポジトリに何らかのイベントが発生した場合、通知設定を行ったコンポーネントへマルチキャスト通信により、同時にイベント内容が通知される。

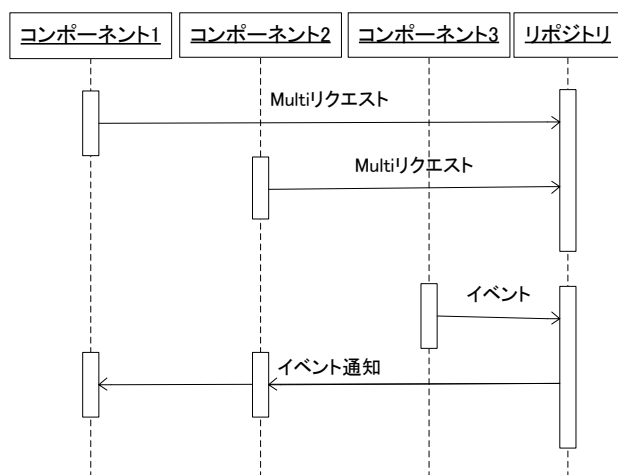


図 3 Multi 機能によるデータの流れ

4. 性能評価

従来のフレームワークのユニキャスト通信を用いた **Notify** 機能による通知機能の遅延を測定した。ネットワークを介してコンポーネントからリポジトリへの操作を行い、一つのコンポーネントへ通知を返す場合と、二つのコンポーネントへ通知を返す場合を比較した。また、この時のデータ操作のイベントは **Put** 機能によるデータの更新を行う。データサイズは、**Put** によって更新されたデータの容量を示す。

図 4 にこの性能評価の二つのコンポーネントへ通知を返す場合の通信を示す。コンポーネント 1, 2 の二つからリポジトリへ通知設定した状態で、コンポーネント 1 からリポ

ジトリへ **Put** リクエストを送る。リポジトリは **Put** によるデータの更新をイベントとして検出し、二つのコンポーネントへ通知する。

図 5 に測定結果のグラフを示す。このグラフにより、従来の **Notify** による通知機能では、通知対象となるコンポーネントが増えると通信効率が低下することがわかる。また、イベントデータサイズが大きくなるごとに遅延時間も増えていくことがわかる。

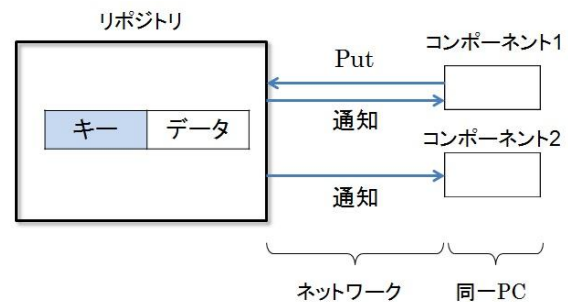


図 4 遅延測定時のデータ通信

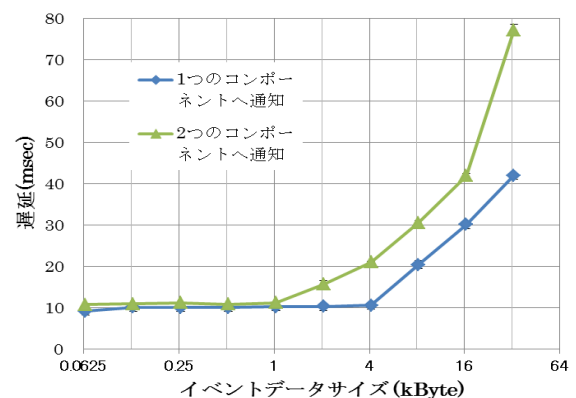


図 5 Notify の通知遅延時間

5. まとめ

本研究では、データ空間共有型通信フレームワークにマルチキャスト通信機能を拡張し、**Notify** によるリポジトリのデータ操作の検知を、複数のコンピュータ間で同時に行える **Multi** 機能を追加する。測定では現状のフレームワークでの通知機能の性能を調べたが、ユニキャスト通信を利用しているため、複数のコンポーネントへ通知を行う際に通信効率が低下していることがわかった。今後、マルチキャスト通信を用いた **Multi** 機能によるデータ転送試験を行い、提案システムの検証を行う。

参考文献

- [1] 元山裕基, 長坂康史: 「動的帯域制御を用いたデータ収集システム」, 情報科学技術フォーラム一般講演論文集, Vol. 4, No. 4 pp. 3-4 (2005)
- [2] 長坂康史, 元山裕基: 「データ収集システムのためのデータ空間共有通信フレームワーク」, 広島工業大学紀要研究編, 第 42 巻, pp. 221-225 (2008)