

広域分散環境における Pub/Sub 通信機能の試作と評価

1240336 多田 輝

【分散処理 OS 研究室】

1 はじめに

コンテンツ指向ネットワークを支える方式として Publisher/Subscriber モデルが存在する。データの送信者 (Publisher) と受信者 (Subscriber) の間に仲介者 (Broker) を配置することで、構造のシンプル化、拡張性向上が期待できる。本研究では、Pub/Sub 通信を広域分散環境を対象として実現する。具体的には、HTTP を用いた Pub/Sub 通信を実現すると共に、Broker 間の負荷分散機能を実現し評価する。

2 広域分散環境での Pub/Sub 通信の実現

本研究で実現した Pub/Sub 通信のシステム構成を図 1 に示す。まず、Publisher-Broker-Subscriber 間の通信プロトコルについて説明する。Pub/Sub 通信の代表的なメッセージングシステム Apache Kafka(以下、Kafka)では、独自の Kafka プロトコルを用いている。しかし、広域分散環境では、セキュリティの観点から通信プロトコルを制限する機会が多いため、Kafka をそのまま適用することが難しい。このため、本研究では Publisher-Broker-Subscriber 間の通信を HTTP で実現した。

次に Broker での負荷分散機能について説明する。最近の Pub/Sub 通信では、Broker がデータの配信だけでなく加工をする場合が多く、Broker の負荷が問題となる。本研究では、動画配信・加工を対象の Broker 間の負荷分散機能を実現した。図 1 に示すように、Publisher は、Broker1 へ動画を送信し、Broker1 は、Yolov8 を用いて動画解析を行う。Publisher から動画が送信されるたび、Broker1 上に動画解析処理を実行するプロセスが増加し、負荷がかかる。そこで、Broker1 は自身の CPU 使用率を監視し、閾値を超えた場合、別拠点の Broker2 に負荷を分散する。Broker2 は動画を解析し、解析した結果を元の Broker1 に返却することで、Broker1 の負荷を軽減する。

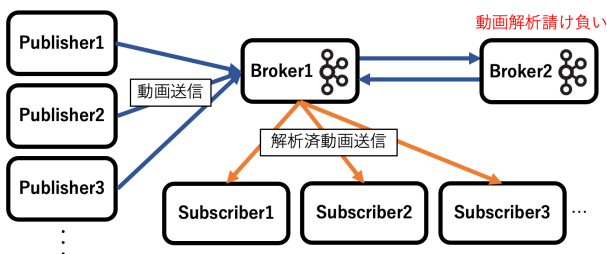


図 1 本研究での Pub/Sub システム構成

3 評価

プロトコル別のテキスト送受信において、送信処理側はデータを送信し終えるまでの時間、受信処理側はデータを受信し、保存し終えるまでの時間とし、1K、10K、

100K、1M、10Mbyte のテキストデータごとに検証する。結果は図 2 の通りである。

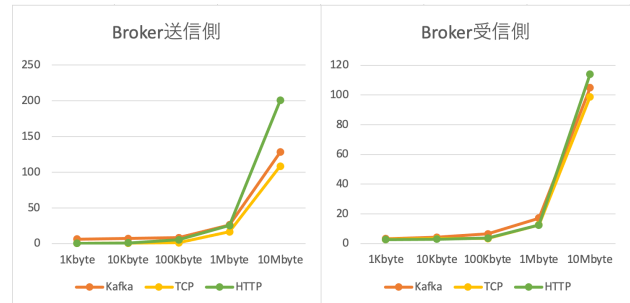


図 2 プロトコル別テキスト送受信処理時間

結果として、HTTP が最も処理時間が長い、大きな差ではなく、Kafka において HTTP の特性を活かした汎用性の高い通信を微小の遅延時間で使えることができると思われる。

次に、Broker 間の動画解析の負荷分散において、Publisher は FullHD(1920 × 1080)、5FPS の動画データを送信し続けるものとする。CPU 使用率が 20 % 台の時を安定時、80 % 以上の時を高負荷時とする。高負荷時に Broker は別 Broker に動画解析の負荷分散を行う。解析した動画の 60 秒間の平均 FPS を算出した結果が以下の図 3 である。

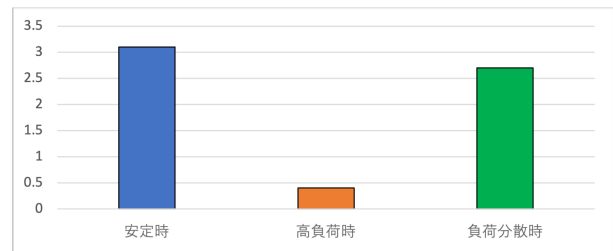


図 3 リアルタイム解析動画の平均 FPS

負荷分散時には安定時と比較し FPS が減少しているが、高負荷時と比べると安定して解析動画を提供できていると思われる。

4 まとめ

本研究では、Pub/Sub 通信における広域分散環境での実現によるサービスの維持、プロトコル別のデータ処理時間の比較を行い、汎用的に使えることを証明した。

参考文献

- [1] Neha Narkhede, Gwen Shapira, Todd Palino, :Kafka, 株式会社オライリー・ジャパン, 東京 (2018).