

分散台帳を利用したデータ検証可能な分散データベースの性能に関する検討

坂本 明穂 (指導教員：小口 正人)

1 はじめに

検証済みのデータを分散データベース上に保存することで、複数のユーザ間でのデータの安全な共有が可能になる。このときデータ検証プラットフォームとしてプライベートブロックチェーンの利用が考えられる。

しかしプライベートブロックチェーンによるデータ検証ではブロックチェーンネットワークを構成するノード間での合意形成が必要となるため、データの処理速度は通常のデータベースと比べて遅くなる。

本研究では、プライベートブロックチェーンの1つである Hyperledger Iroha の性能について通常のデータベースとの比較を行う。またネットワークサイズを変化させて、処理するトランザクション数が増加することによるレイテンシやスループットへの影響を計測する。

2 Hyperledger Iroha

Hyperledger Iroha ではデータ検証をピアが担っている。クライアントから送信されたトランザクションはオーダリングサービスに集められ、複数トランザクションから成るプロポーザルが作成される。各ピアへプロポーザルが送られると、各ピアはプロポーザルの検証を行う。その後ピア間でメッセージのやり取りが行われ、プロポーザルに含まれるトランザクションに対する合意が形成されると、正しいと検証されたトランザクションは各ピアのブロックに格納される。

3 関連研究

Hyperledger Iroha のネットワークサイズが小さいとき投票ステップ遅延値を短く設定することがスループットの向上に有効である一方、ネットワークサイズが大きくなると投票遅延値が小さいとタイムアウトが発生してコンセンサスの形成に失敗する可能性がある [1]。

Hyperledger Iroha ネットワークにおいて、トランザクション送信レートを増加させても最小レイテンシおよび平均レイテンシは影響を受けないが、ネットワークサイズを大きくすると平均レイテンシは増加する。またトランザクション送信レートが小さい (20tps) ときスループットは概ね一定であるが、トランザクション送信レートが高い (100tps) とき、スループットはネットワークサイズが大きくなるにつれて減少する [2]。

4 実験

4.1 実験環境

実験に使用したサーバの性能は Ubuntu20.04LTS の OS, Intel(R) Xeon(R) Silver 4314 CPU @ 2.40GHz の CPU, コア数 32, スレッド数 64, メモリは 192GB である。

また PostgreSQL 単体および Hyperledger Iroha ネットワークの各ピアは Docker コンテナを用いて構成される。Hyperledger Iroha の実行環境は図 1 の通りで、各ピアは Hyperledger Iroha プロセスを実行するコンテナと PostgreSQL データベースを実行するコンテナか

ら構成される。ピア同士は仮想ネットワークによって接続される。

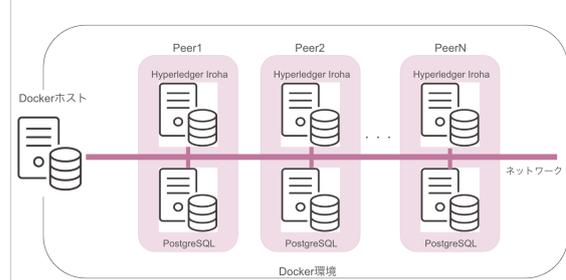


図 1: Hyperledger Iroha 実験環境

4.2 実験概要

PostgreSQL と Hyperledger Iroha で書き込み処理 1 件にかかる時間を比較する。Hyperledger Iroha ではネットワークを構成するピア数を 5, 10, 20, 30 と変化させ、スケラビリティに関する評価も行う。

次にブロックチェーン基盤の性能測定ツールである Hyperledger Caliper を用いて、Hyperledger Iroha の平均レイテンシとスループットの測定を行う。使用した Hyperledger Caliper のバージョンは 0.3.2 である。5, 10, 20, 30 ピアから構成される Hyperledger Iroha ネットワークを対象とし、1 秒あたりに送信するトランザクション数を 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200 と変化させる。測定は 10 回行い、最大値と最小値を除いた 8 回の結果を箱ひげ図に示す。

4.3 実験結果

PostgreSQL と Hyperledger Iroha での 1 つのトランザクションの処理にかかる時間は表 1 のようになった。両者の実行時間の比較から、Hyperledger Iroha での PostgreSQL へのデータの書き込みにかかる時間は処理全体の約 0.2% であり、Hyperledger Iroha ではトランザクション処理時間のほとんどをコンセンサス形成に費やしていることがわかる。また Hyperledger Iroha の構成ピア数が増加しても、処理時間に大きな差が見られなかった。

表 1: PostgreSQL と Hyperledger Iroha での処理時間

測定対象	測定結果 [ms]
PostgreSQL	10.3
Hyperledger Iroha 5peer	5696.27
Hyperledger Iroha 10peer	5641.27
Hyperledger Iroha 20peer	5777.56
Hyperledger Iroha 30peer	5804.01

次に Hyperledger Caliper を用いて、トランザクション送信レートを変化させたときの Hyperledger Iroha ネットワークの平均レイテンシとスループットを計測した。

トランザクション送信レートを変化させたときの平均レイテンシの変化を図 2 に示す。毎秒 50 トランザク

ションまではトランザクション送信レートの増加とともに平均レイテンシが増加する傾向が見られた。一方、50トランザクション以降はトランザクション送信レートを増加させても平均レイテンシの値は横ばいとなった。したがって単位時間あたりにオーダーリングサービスからピアに送信されるプロポーザル数には上限が定められていると推測できる。すなわちオーダーリングサービスが受け取るトランザクション数が一定値を超えると、超えた分のトランザクションはすぐにピアに送信されるのではなく一定時間経過後ピアに送信される。その結果送信されるトランザクション数が増加しても、単位時間あたりにピアが処理するプロポーザル数は一定となるため平均レイテンシの値が一定になったと考えられる。またトランザクション送信レートが20tps以下のときにネットワークサイズを大きくすると、20ピアまでは平均レイテンシは増加したが30ピアでは減少した。一方トランザクション送信レートが30tps以上のとき、ネットワーク構成ピア数の増加とともに平均レイテンシも増加した。ネットワークを構成するピア数が増えるとプロポーザルのコンセンサス形成に必要なメッセージの総数が増加するため、ピア数が増えるにしたがって平均レイテンシも増加したと考えられる。

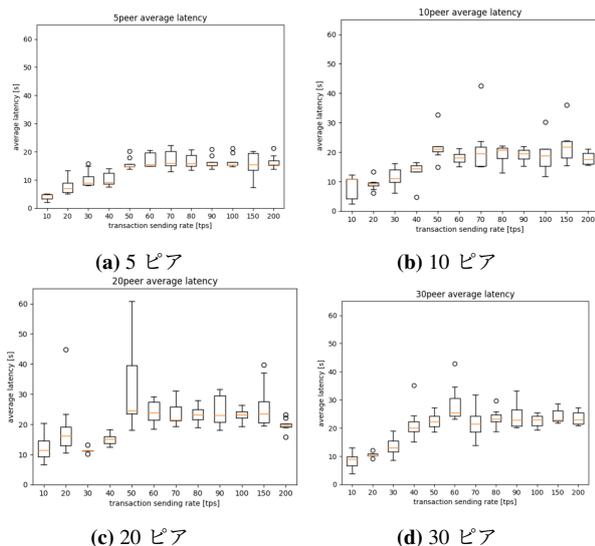


図2: トランザクション送信レートを変化させたときの平均レイテンシ

トランザクション送信レートを変化させたときのスループットの変化を図3に示す。ネットワークを構成するピアが5つのとき、単位時間あたりのスループット数は70tpsまでは減少し、その後横ばいになるが100tps以降は増加した。10ピア以降のとき、トランザクション送信レートが30tpsまたは40tpsまではスループットは減少し、30tpsから40tps以降ではスループットの値に大きな変化は見られなかった。スループットの値が横ばいとなった原因としては、ピアに送信されるトランザクション数が一定になった可能性が考えられる。またネットワーク構成ピア数を変化させたとき、スループットは概ね低下した。ただしトランザクション送信レートが10tps、20tpsのときは20ピアのときスループットが最低となった。平均レイテンシのとき

と同様、ピア数が増えるとコンセンサス形成時に送信されるメッセージ数が増加するため、ネットワークサイズが大きくなるとスループットが低下すると考えられる。

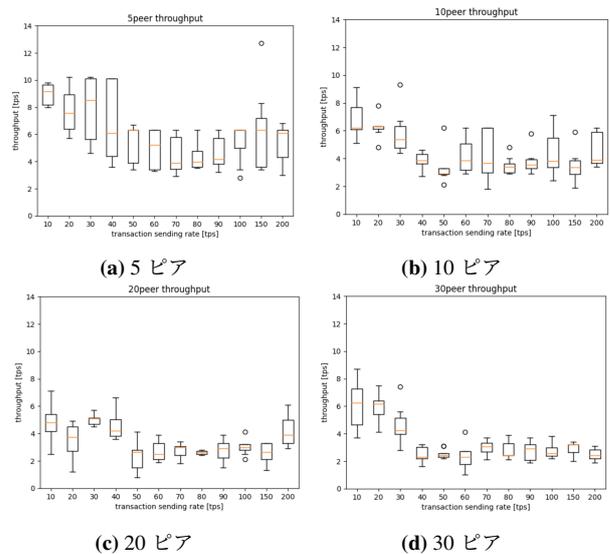


図3: トランザクション送信レートを変化させたときのスループット

Hyperledger Iroha を構成する各コンテナはCPUの各スレッドに割り当てられるためコンテナ内の処理は他のコンテナの影響を受けない。しかしネットワークやI/Oは共有しているため、それらの負荷が高い場合は他のコアによる遅延が発生する可能性があり、測定された性能にある程度ランダム性が存在すると考えられる。

5 まとめと今後の課題

データベースにおけるデータ検証プラットフォームとしてのHyperledger Irohaの活用を目的として、一般的なデータベースとHyperledger Irohaの性能比較および、Hyperledger Irohaのネットワークサイズや送信トランザクション数による性能の変化について評価を行った。

今後は、Hyperledger Iroha ネットワーク内に異なる処理能力を持つピアが存在したときのネットワークの振る舞いの評価や、実ネットワークを用いてピアを接続したときの性能の評価を行う。

謝辞

本研究は一部、JST CREST JPMJCR22M2の支援を受けたものである。

参考文献

- [1] Fedor Muratov, Andrei Lebedev, Nikolai Iushkevich, Bulat Nasrulin, and Makoto Takemiya. Yac: Bft consensus algorithm for blockchain. 2018.
- [2] Arnold Woznica and Michal Kedziora. Performance and scalability evaluation of a permissioned blockchain based on the hyperledger fabric, sawtooth and iroha. *Computer Science and Information Systems*, 19:659–678, 2022.