NTT DATA
Trusted Global Innovator

ETHTerakoya スケーリング公開WS

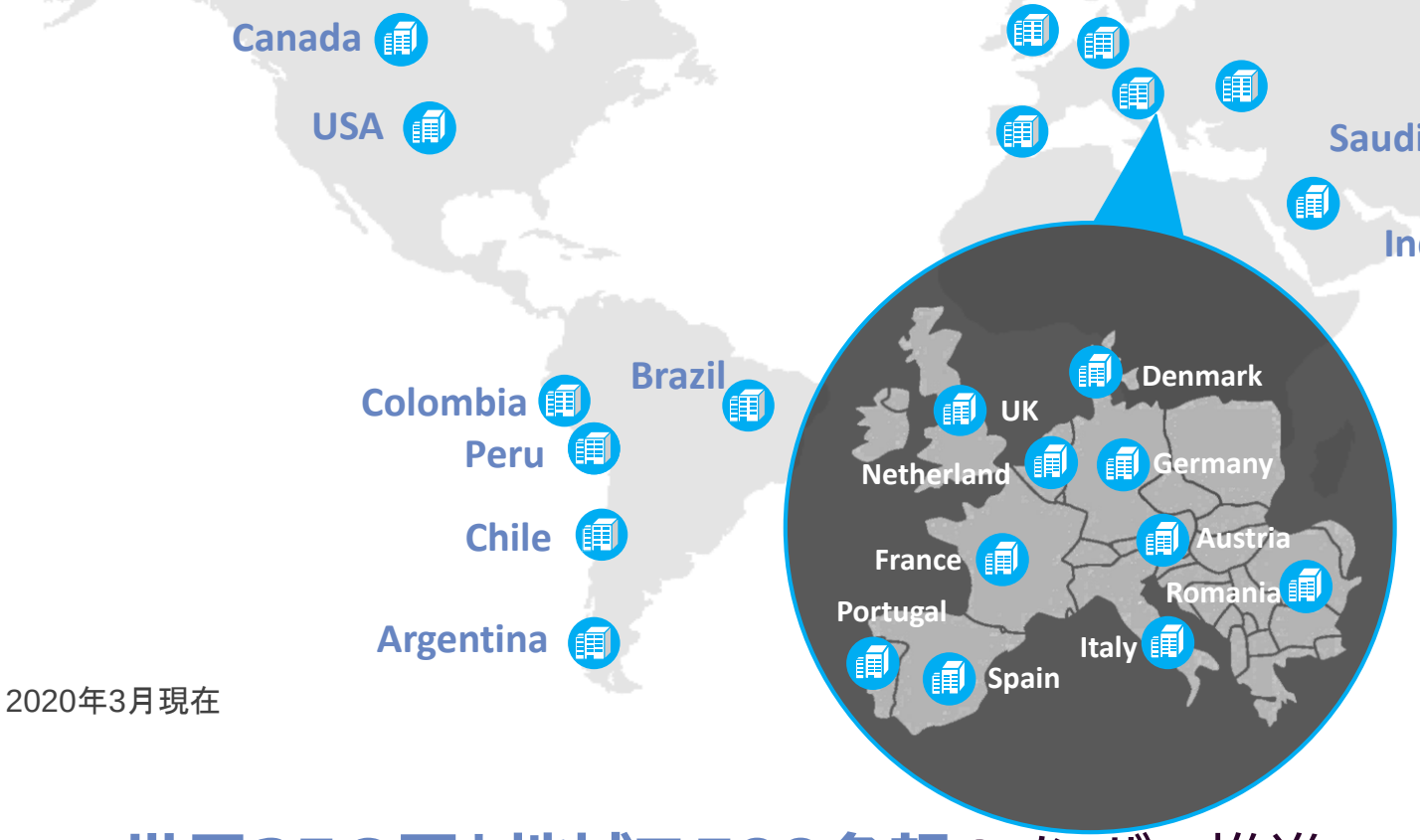
Hyperledger Caliperを用いたQuorum性能検証について

2021年10月29日
株式会社NTTデータ
清水 俊平

NTTデータの取り組み

ブロックチェーンへの取り組み：Blockchain CoE（Center of Excellence）

様々なバックグラウンドを持つ世界中のチームメンバがビジネス動向・技術ナレッジを共有
お客様にとって最適なプラットフォームを提供できる体制を確立



2020年3月現在

各国の多様な価値

社会インフラ

新しい技術を用いた研究開発力

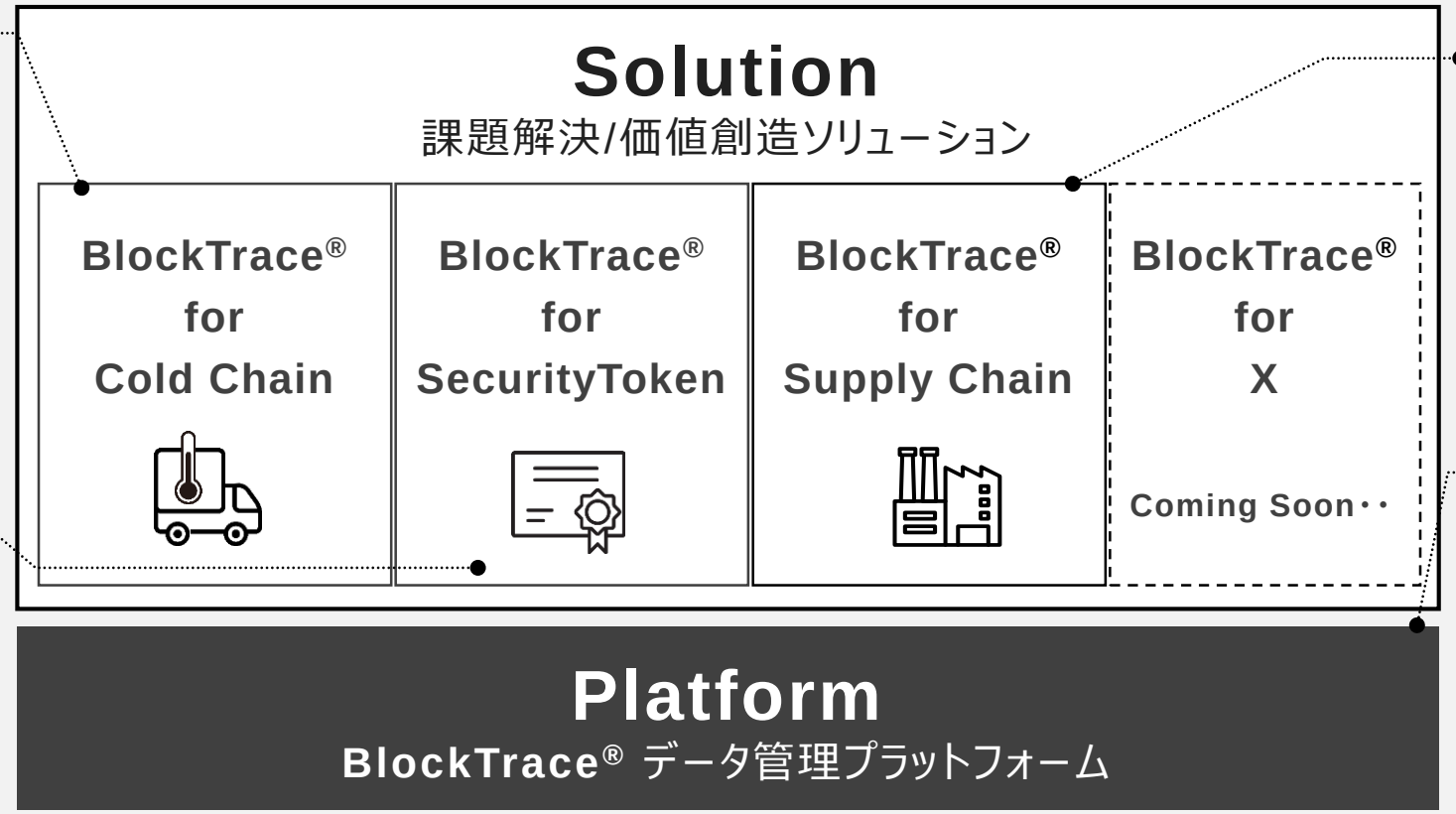
世界25の国と地域で500名超のメンバで推進

ブロックチェーン技術を活用したDX推進ソリューション“BlockTrace®”

ビジネスシーン/ユースケースに応じたブロックチェーンプラットフォーム上に、お客さまの適用シーンに応じたアプリケーションを構築・提供するソリューション「BlockTrace®」を展開

BlockTrace® for Cold Chain
- 食品の品質保証ソリューション

BlockTrace® for Security Token
- 証券デジタル化ソリューション



BlockTrace® for Supply Chain
- サプライチェーンにおける企業間情報共有ソリューション

BlockTrace® データ管理プラットフォーム
- エンタープライズBlockchain基盤に対応(Hyperledger Fabric、Corda、Quorum)
- NTTの研究所技術活用により、「大容量ファイル登録」、「ユーザ毎にデータアクセス制御」が可能

Quorum性能検証

NTT DATA

目次

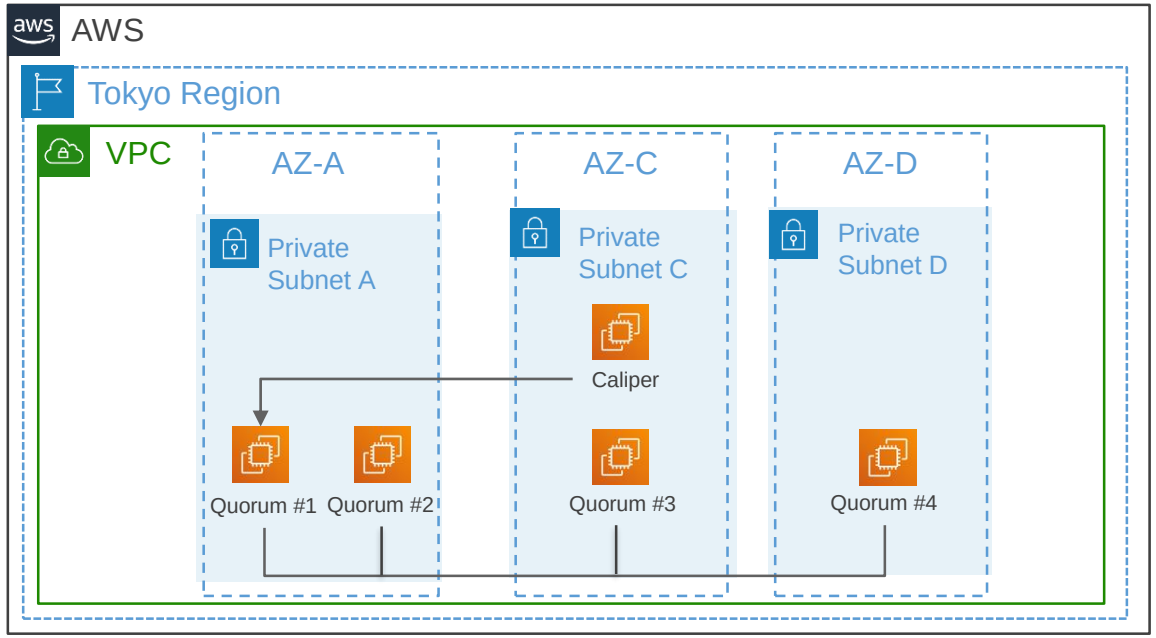
1. 検証環境構成図①

Quorumの構成情報は以下とする。
なお、本構成情報はホワイトペーパーに準ずる。

種別	項目	定義	値
ブロックチェーンに関する情報	Blockchain Client Name	ブロックチェーンのクライアント名	Go quorum (v21.7.1)
	Consensus Algorithm	ブロックチェーンにおけるネットワーク全体の合意を行うための方式。	IBFT
	Transaction Method	ブロックチェーンのデータをある値から別の値に変更する状態遷移法。性能測定の対象となる処理の内容。	Open Query Transfer
	Network Size (Node数)	合意形成に参加しているバリデータノードの数。	4台
	ブロック生成間隔	ブロックを生成する間隔	10s

1. 検証環境構成図②

計測環境としてAWS上に下記の構成を構築した。



種別	項目	Quorum インスタンス	Caliper インスタンス
ハードウェア情報	OS	Ubuntu20.04LTS	Ubuntu20.04LTS
	インスタンスタイプ	m5a.large	m5a.large
	vCPU（コア数）	2	2
	メモリ（GB）	8	8
	プロセッサ	Intel Xeon® Platinum 8175 /3.1 GHz	Intel Xeon® Platinum 8175 /3.1 GHz
	インスタンスストレージ	EBSのみ	EBSのみ
	ネットワーク帯域幅	最大10GiB/s	最大10GiB/s
	EBS帯域幅	最大 4,750	最大 4,750
	ボリュームタイプ	汎用SSD（gp2） IOPS:1200	汎用SSD（gp2） IOPS:1200
	主なソフトウェア	Quorum v21.7.1	Caliper v4.0.0

システム構成上のポイント：

- Quorumは最低台数である4台構成とする
- Caliperは1台とする
  ※Dockerは利用しない
- サブネットはAZ単位で用意するが、各サブネット間通信は無条件に通信可能な状態とする

2. Caliper について

Caliperは、残高設定・残高参照・送金のサンプル処理を実行し、性能を計測することができる。
検証別に下記の処理(スマートコントラクト)を用いた。

Open： 残高を設定(書き込み)する
Query：残高を参照(読み込み)する
Transfer：残高を送る(書き込み＋読み込み)する

本検証で設定したCaliperの主な設定項目：

項目	Caliper設定名称	説明
目標TPS	tps	各処理における目標とするTPSの数値。 (例) 50とした場合、50tpsを目標に負荷をかける
トランザクション送信数	txNumber	ラウンド中にCaliperが送信するトランザクションの数。目標TPSに達するまでトランザクションを送信する回数。 (例) 1000とした場合、1000回トランザクションを送信したら試験は終了する
ウォーカー数	workersNumber	負荷をかけるプロセス数。

参考：Caliper公式ドキュメント https://hyperledger.github.io/caliper/v0.4.2/ethereum-config/

3. 計測条件 – 測定項目

Caliperを実行して、取得する項目を下記に定める。
なお、本測定項目はホワイトペーパーに準ずる。

種別	項目	説明
ブロックチェーンに関する情報	Read Latency	読み取り要求を送信し、その応答を受信するまでの合計時間。 実行関数のQueryの結果より取得する。
	Read Throughput	1秒あたりの読み取り数。 実行関数のQureryの結果より取得する。
	Transaction Latency	ネットワーク全体がトランザクションを検証するためにかかる時間。 実行関数のOpenとTransferの結果より取得する。
	Transaction Throughput	定義された期間にブロックチェーンによって有効なトランザクションがコミットされる割合。 実行関数ごとにReadとWriteを合わせたThroughputより取得する。
システムのリソース使用率に関する情報	CPU負荷(Max, Min, Avg)	CPUに対する負荷。dstatコマンドで取得する。
	ディスク容量負荷 (Max, Min, Avg)	ディスク容量に対する負荷。dstatコマンドで取得する。 Dstatの取得コマンドは下記で実施する。 $ dstat -tcd -o <outputファイル名>5 ※5秒間隔で時刻、CPU使用率、ディスク使用率を記録

3. 計測条件 – 検証項目

方針：目標TPSは50(固定)、ウォーカー数は1(固定)とする。
トランザクション送信数を＋1000増やしながら検証を繰り返す。

本検証では下記のパターンで計測を実施する。

検証No	種別	実行関数	マイニング間隔	ウォーカー数	目標TPS	トランザクション送信数
1-1	fixed-rate	open	10s	1	50	1,000
1-2	fixed-rate	query				
1-3	fixed-rate	transfer				
2-1	fixed-rate	open				2,000
2-2	fixed-rate	query				
2-3	fixed-rate	transfer				
・・・ 以降繰り返し（目標TPS 50固定、トランザクション送信数＋1000）						

※上記の設定値はbenchmarks/config.jsonに定義する

4-1. 計測結果

計測結果（サマリー）：

検証No.	実行関数	Read Latency			Transaction Latency			Transaction Throughput ※(Read,Write合計)	リソース最大使用率	
		最大値 [s]	最小値 [s]	平均値 [s]	最大値 [s]	最小値 [s]	平均値 [s]		CPU 平均値 [%]	Disk Write I/O 平均値 [byte]
1-1	Open	-	-	-	12.18	1.94	7.05	94.5	21.33335	0
1-2	Query	0	0	0	-	-	-	50.1	0.333333333	0
1-3	Transfer	-	-	-	12.03	1.22	6.42	50	6.000142857	0
2-1	Open	-	-	-	16.53	1.21	6.7	66.9	22.83191667	800.6414167
2-2	Query	0.01	0	0	-	-	-	50	1.222222222	376.539
2-3	Transfer	-	-	-	12.03	1.22	6.42	50	11.54527273	30.18190909
3-1	Open	-	-	-	34.92	1.99	17.37	60.3	29.94282353	2.823117647
3-2	Query	0.01	0	0	-	-	-	50	1.038384615	115.1046923
3-3	Transfer	-	-	-	12.23	1.21	6.58	50	10.53433333	5.868266667

4-2. 計測結果

計測結果（サマリー）：

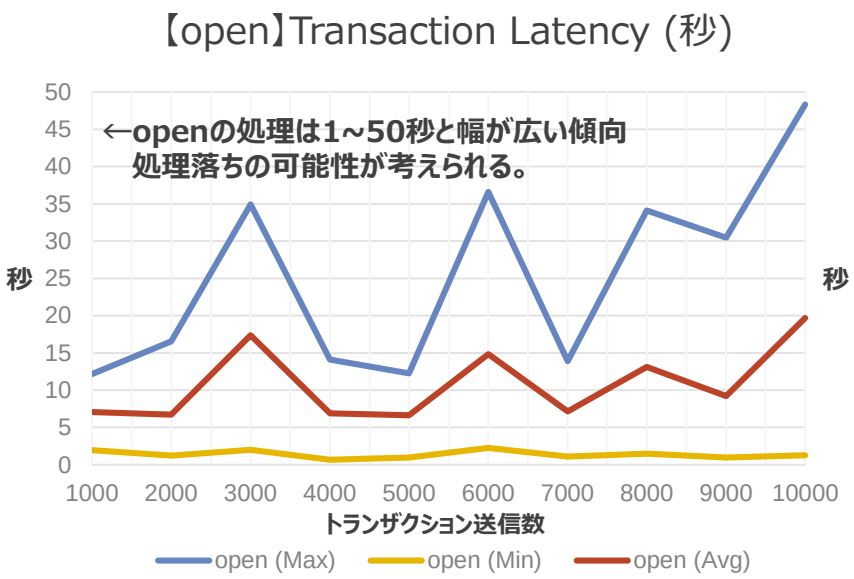
検証No.	実行関数	Read Latency			Transaction Latency			Transaction Throughput ※(Read,Write合計)	リソース最大使用率	
		最大値 [s]	最小値 [s]	平均値 [s]	最大値 [s]	最小値 [s]	平均値 [s]		CPU 平均値 [%]	Disk Write I/O 平均値 [byte]
4-1	Open	-	-	-	14.1	0.64	6.9	49.8	20.99794737	0
4-2	Query	0.01	0	0	-	-	-	50	0.027777778	54.22455556
4-3	Transfer	-	-	-	12.24	0.64	6.46	49.9	2.789473684	0
5-1	Open	-	-	-	12.25	0.96	6.63	49.9	57.85386957	169.5736522
5-2	Query	0.01	0	0	-	-	-	50	3.272727273	704.006
5-3	Transfer	-	-	-	12.23	1.16	6.67	49.9	48.34991304	3.304173913
6-1	Open	-	-	-	36.59	2.23	14.83	54.1	64.44280769	413.0867308
6-2	Query	0.01	0	0	-	-	-	50	0.942538462	0
6-3	Transfer	-	-	-	12.23	1.03	6.69	49.9	7.574074074	367.4096667

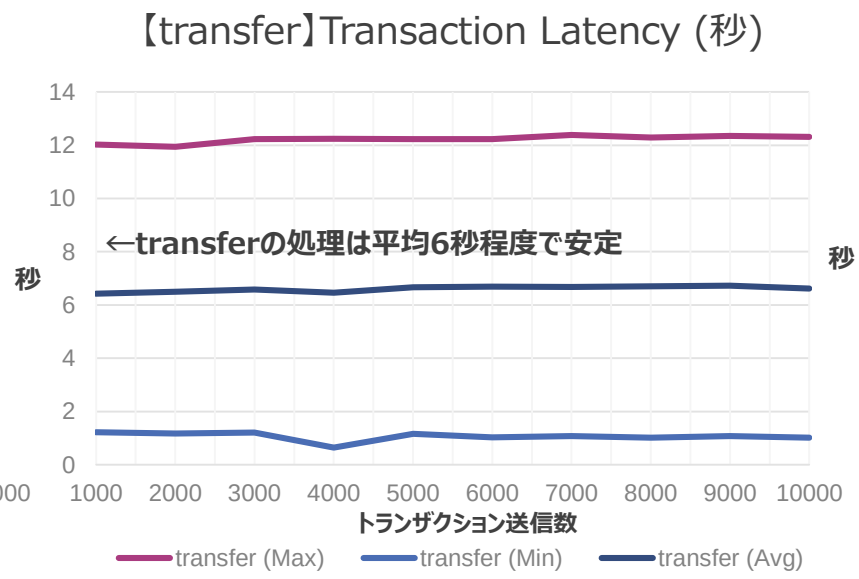
検証No10(トランザクション送信数：10000)まで測定を実施

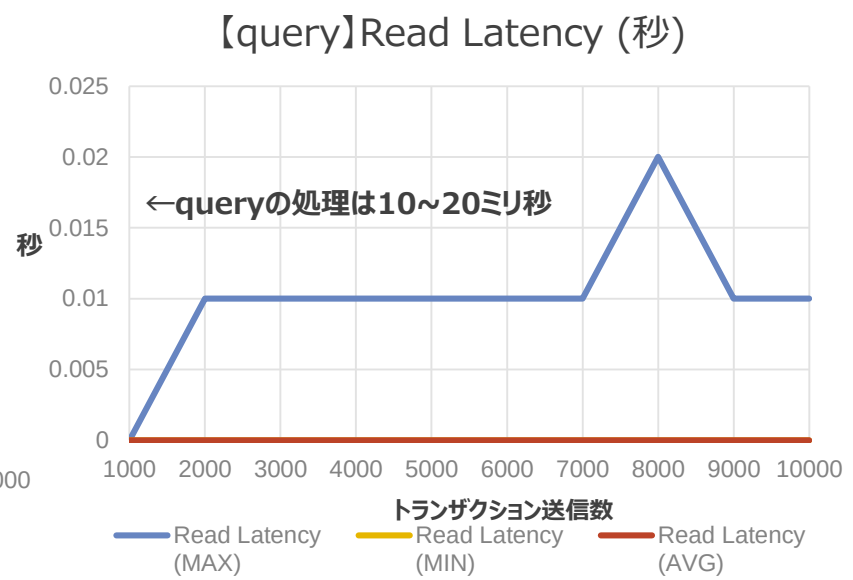
4-3. 計測結果

■Latencyの傾向について

下記のグラフの通り、トランザクション送信数の大小に関わらず、平均してLatencyは低い傾向にある。
また、query処理におけるRead Latencyは常に高速であるのに対し、openおよびtransfer処理における Transaction のLatencyは高くなる傾向にある。



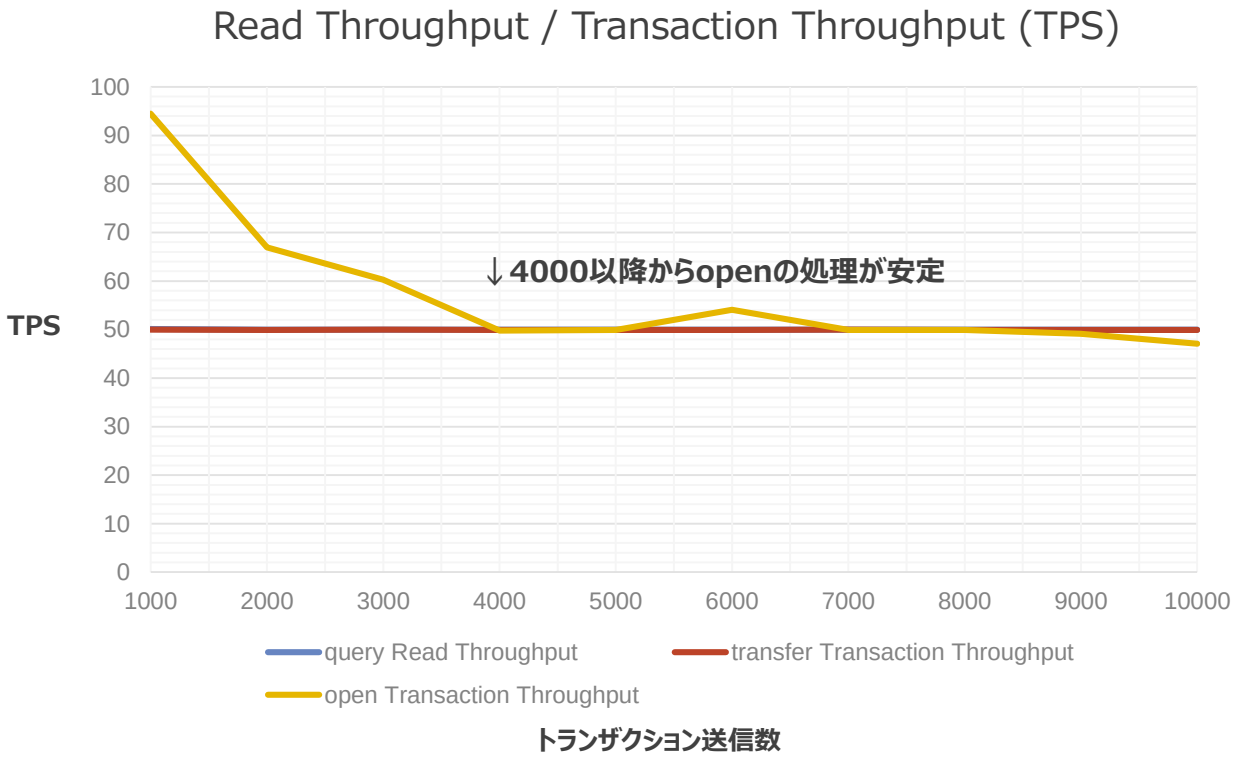




4-4. 計測結果

■Throughputの傾向について

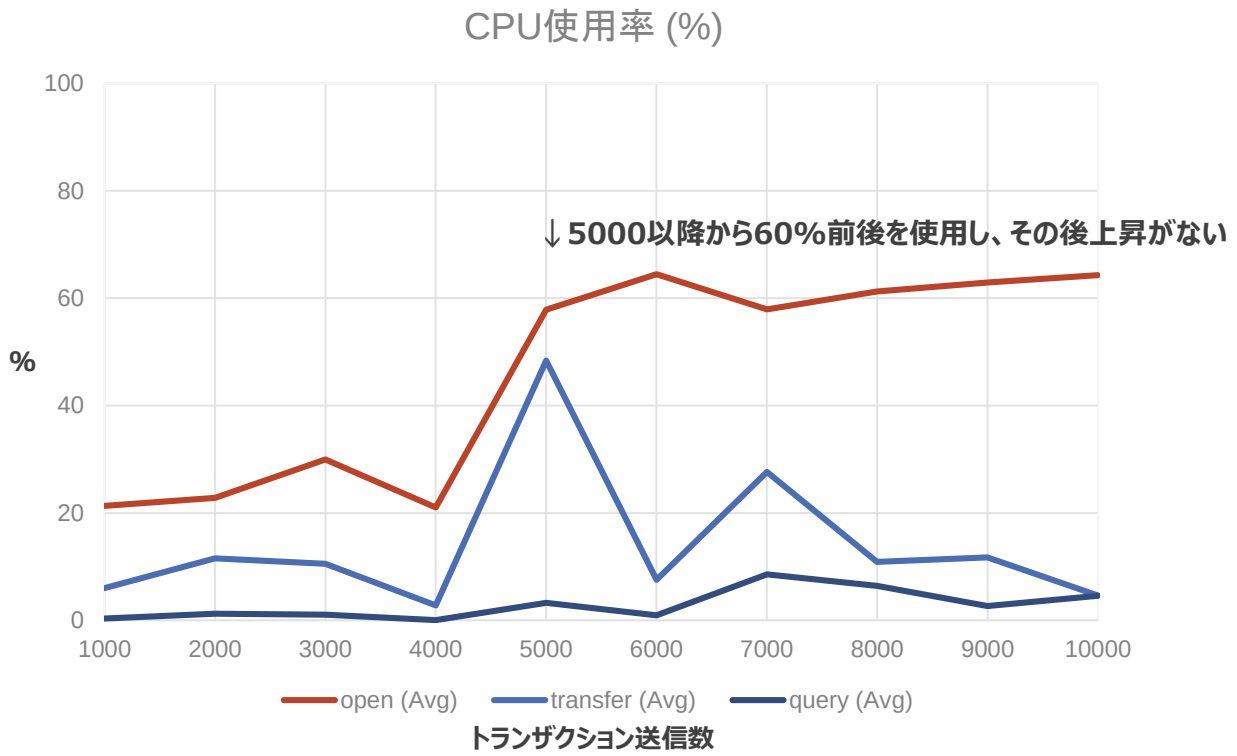
下記のグラフの通り、トランザクション送信数が多くなるにつれてThroughputが目標TPS(50tps)近くへ推移し、トランザクション送信数が4000回以上で安定する傾向にある。



4-5. 計測結果

■CPU使用率の傾向について

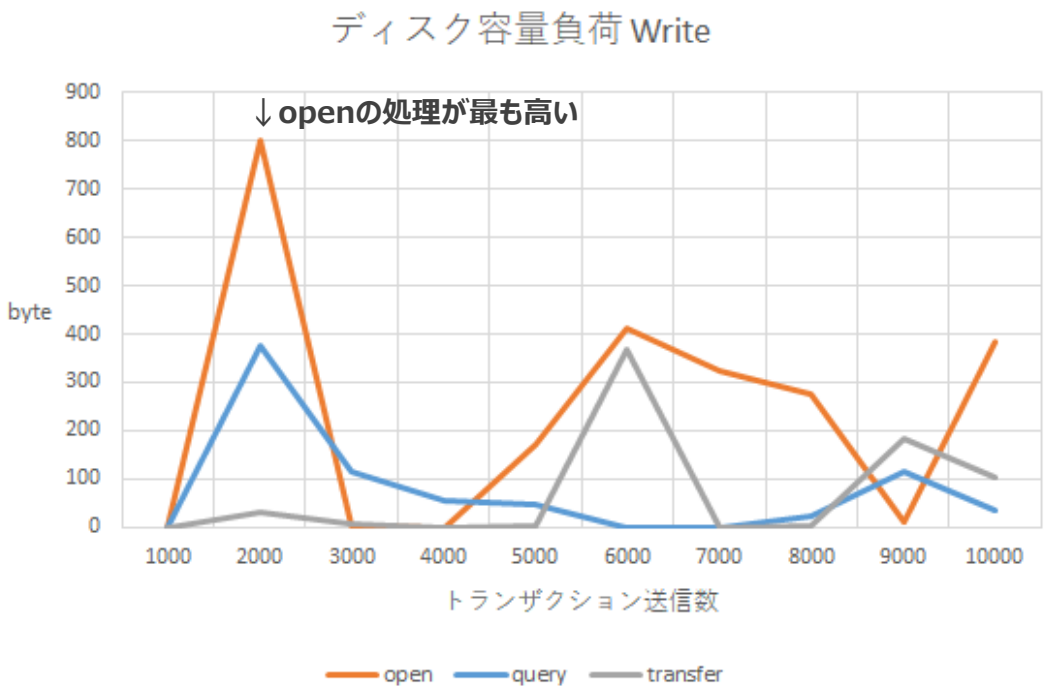
下記のグラフは、5秒おきの平均値の値をdstatで取得し、その値からトランザクション送信数ごと(検証ごと)に平均値を割り出してグラフ化したものである。
Open処理が60%に到達し、70%を超えていない状況からCPU使用率は余裕がある状況である。また、Open処理が最もCPUを使用している処理であることがわかる。

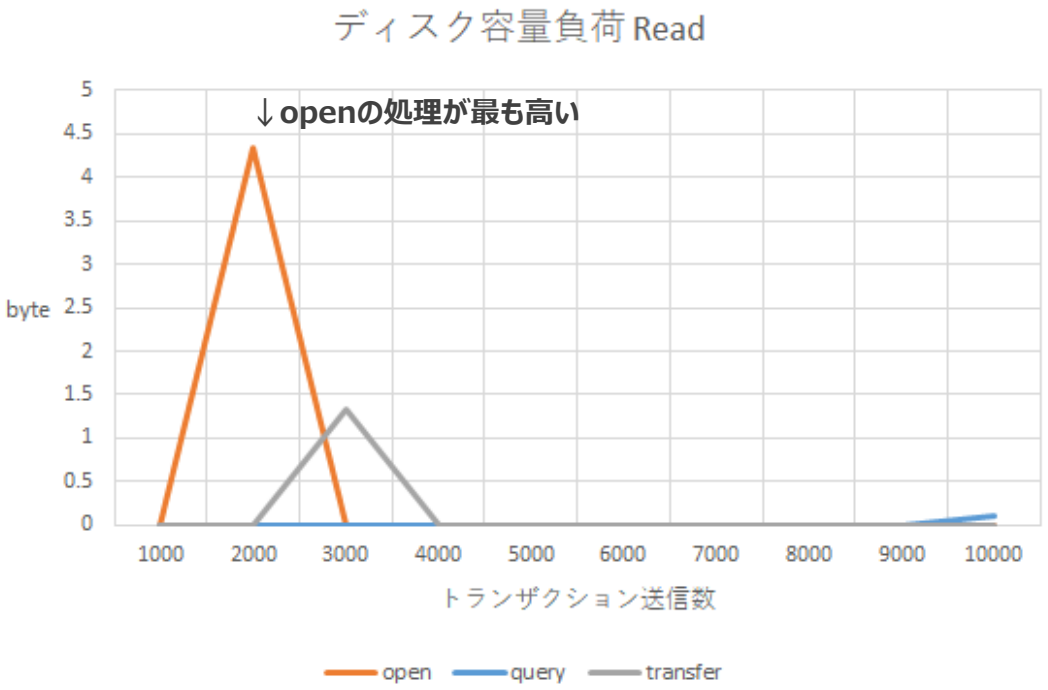


4-6. 計測結果

■Disk負荷の傾向について

下記のグラフは、5秒おきの平均値の値をdstatで取得し、その値からトランザクション送信数ごと(検証ごと)に平均値を割り出してグラフ化(ReadとWriteで分けて作成)したものである。
Openの処理が最も書き込みも読み込みも負荷が高い処理であるということがわかる。
ただし、本測定以外のプロセスによるディスク容量負荷が含まれているため、より正確な測定には負荷かけ対象のディスクを用意する必要がある。



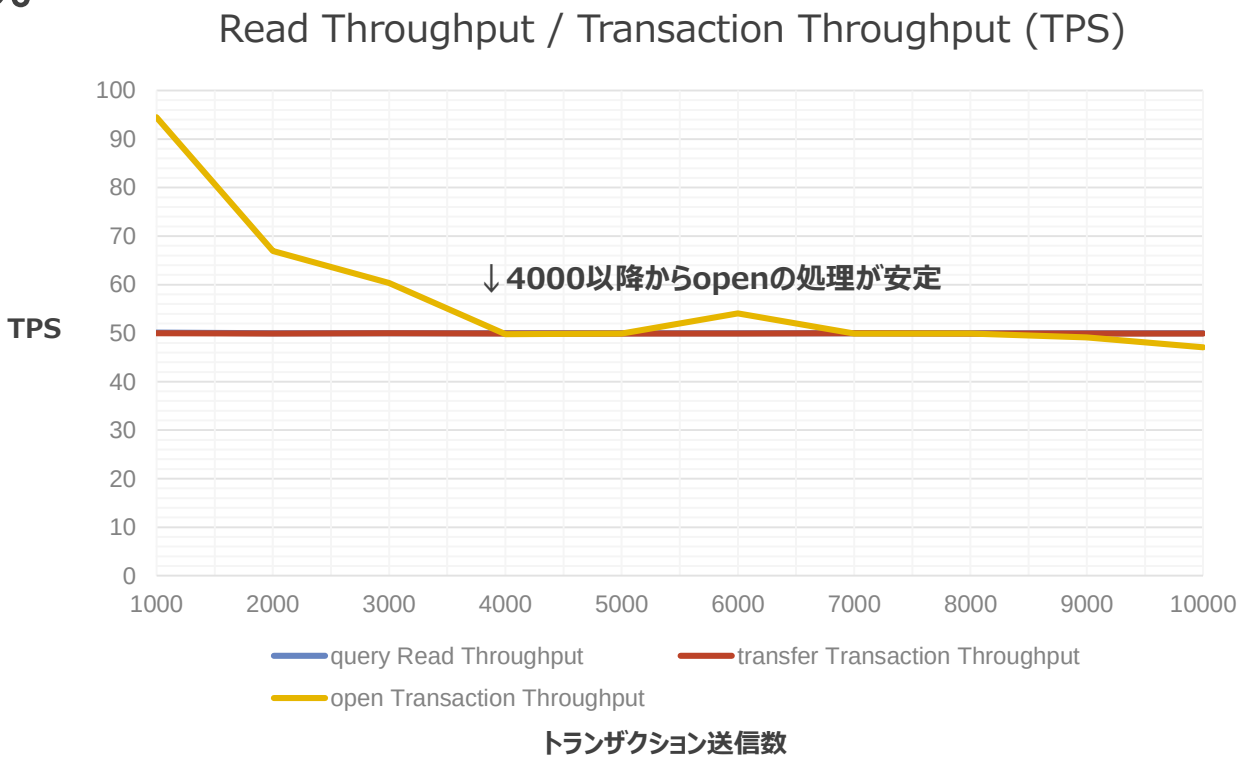


5. 考察

■目標TPS × トランザクション送信数の関連性

トランザクション送信数を1000から18000まで増やす検証を実施したが、目標TPSおよびCPU使用率はトランザクション送信数4000以上で安定する傾向であった。これより、目標TPS50の本検証では適切な負荷を安定的にかけるために、トランザクション送信数は4000以上に設定する必要があることが分かった。

本検証では目標TPSを固定としたが、高負荷の性能検証を行う場合には目標TPSとトランザクション送信数を適切に設定する必要がある。



NTT DATA

Trusted Global Innovator

BlockTrace® はNTTデータの登録商標です。

本資料に掲載の会社名、製品名、サービス名は、それぞれ各社の登録商標です。