[ポスター発表] 研究報告

ブロックチェーンの待ち行列的シミュレーションの構築

鈴木 龍[1]　大場 春佳[1]　水野 信也[1,a)]

Development a blockchain queueing simulation

1. はじめに

ブロックチェーンは 2008 年にインターネットに登場して [1]，主に仮想通貨を扱うプラットフォームとして成長している．ブロックチェーンの利用範囲は仮想通貨だけに留まらず，完全性を強く必要とするアプリケーションへの利用が期待されている．例として製造業や流通業では，作業が階層化されている場合があり，扱う部品数も膨大となる．製品の不良が確認された場合，膨大な部品情報から構成を確認する必要がある．ブロックチェーンを利用していれば，トレーサビリティが明確であり，追跡や遡及の可能性が上がる．しかし，ブロックチェーンの技術的課題もあげられている．ファイナリティ（決済完了性）の問題では，ブロックが認証される過程でチェーンが分岐した場合，自分のトランザクションが確定されるまでに未確定の情報が反映してしまう可能性がある [2]．これはクラウド・コンピューティング上のシステムの特徴のひとつとして，CAP 定理 [3] があり，可用性と分散性を同時に満足させるには Consistency（一貫性）を犠牲にし，トランザクション処理で言及される「BASE」の 結果整合性と同種の問題と言える．またブロックチェーンでは，システムのスケーラビリティも課題となっている．ビットコインでは一つのブロックを処理するのに平均 10 分程度となるように調整されている．また 1 ブロックは 1MByte 程度であり，単位時間あたりのトランザクションの処理量は限られる．そのため，ビットコインの利用者が増えた場合，ブロック処理が追いつかなくなるという可能性があり，そのための対応策も必要となっている [4]．

ブロックチェーンの数理モデル化については，笠原 [5] が待ち行列モデルで表現をしており，定常性を仮定した評価が可能となっている．ブロックチェーンを使ったアプリケーションは現在様々なものがあり，数理モデルの基盤の上に，動的なシミュレーションを連携させることで，ブロックチェーンの様々な仕様に対応でき，時系列的な変動を確認することができる．また上記で挙げた課題に対しても，柔軟なアプローチが可能である．本研究では，ブロックチェーンの待ち行列モデルを基本とした数理モデルに対応するシミュレーションを構築して，ブロックチェーンが抱える課題に対してアプローチを図る．

2. ブロックチェーンの待ち行列を利用したモデル化

ブロックチェーンの課題を検証するために，ブロックチェーンの処理過程を待ち行列を利用してモデル化を行う．以下でモデル化の仮定を行う．

(1) トランザクションの到着は率 λ のポアソン過程に従う．
(2) 到着したトランザクションは先着順で n 個目のブロックに格納され，その格納数を c_n とする．
(3) 1 ブロックに格納できるトランザクションの総和最大容量を b とする．
(4) n 個目のブロックに格納されたトランザクションの総和容量を b_n とする．
(5) ブロックの容量が最大容量に達する，または処理中で格納できない場合，トランザクションはバッファに格納される．
(6) このバッファ容量は無限大とする．
(7) 形成されたブロックの処理時間は平均 μ の指数分布に従うとする．
(8) ブロックの処理が終了すると，新たなブロックがトランザクションの先着順で形成される．
(9) ブロックの処理作業は 1 つのサーバでの処理に相当するため，窓口数は 1 つとなる．

これらの仮定は基本的な待ち行列モデルとして，集団サービス型待ち行列 $M/M^{[b_n,b]}/1$ [5–8] と考えることも可能である．しかし，ブロックを形成する基準がブロック容量であり，格納されるトランザクション数ではないように，すべてを集団サービス型待ち行列として表現することが難しい．そこで，次の仮定を取り入れる．

(1) トランザクションの大きさは平均 α, 標準偏差 σ の正規分布に従う．
(2) 1 ブロックの上限容量が b を超えない最大となるトランザクション総和容量 b_n でブロックが形成される．

[1] 静岡理工科大学
Shizuoka Institute of Science and Technology
[a)] mizuno.shinya@sist.ac.jp

(3) 1 ブロックの容量は変動するが，ブロックの処理時間は b_n にかかわらず，平均 μ の指数分布に従う．

このような待ち行列モデルを基本とした仮定の中で，数学的な枠組みを維持し，シミュレーションを実施することで，ブロックチェーンの処理の挙動を把握していく．

3. ブロックチェーンシミュレーションの数値計算例

今回のブロックチェーンシミュレーションにおいて，表 1 の値を用いて実施した．

表 1 ブロックチェーンシミュレーションのパラメタ

項目	値
トランザクションの到着率 λ(個/秒)	1.0
トランザクションの容量平均 (byte)	550
トランザクションの容量標準偏差 (byte)	50
1 ブロックの最大容量 (Mbyte)	1.0
1 ブロックの平均処理時間 (秒)	600
シミュレーション時間	100000

この条件での平均系内トランザクション数は 1495.27, 平均待ちトランザクション数は 151.33 となった．今回は約 1800 個のトランザクションが一つのブロックにまとめられ，同時にサービスを受け退去することになる．図 1 は系内のトランザクション数を示している．

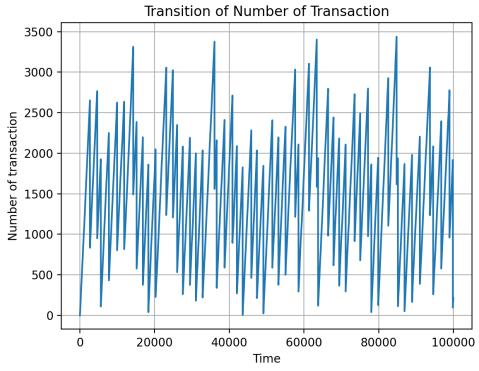

図 1 系内トランザクション数の変化 ($\lambda = 1.0$)

トランザクションの到着率を 2.0 にしてみると，平均系内トランザクション数が 2860.33, 平均待ちトランザクション数が 1273.19 となり，系内のトランザクションの変化は図 2 のようになった．

システムの利用率 ρ は，$\rho = \frac{\lambda}{\mu \bar{c}_n}$ となる．$\lambda = 1.0$ のとき，$\rho = 0.33$ となる．$\lambda = 3.0$ のとき，$\rho = 0.99$ となり，トランザクションの到着率が増えると，ブロックチェーンの処理が遅延し，ブロックチェーンシステムの破綻が起きる可能性がある．このようにブロックチェーンシステムはスケーラビリティへの対応が必要になる．

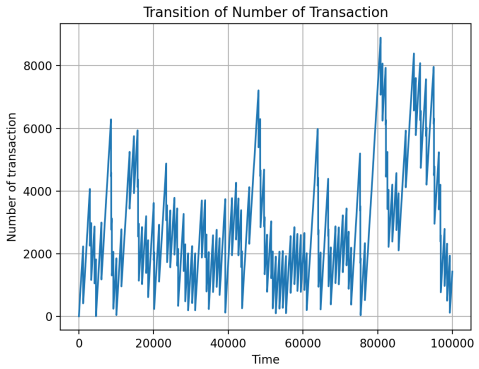

図 2 到着率が増えた場合の系内トランザクション数の変化 ($\lambda = 2.0$)

4. おわりに

本研究では，今後導入が期待されるブロックチェーンの仕組みに対し，待ち行列理論をベースとしたシミュレーションを構築して，系内のトランザクション数などを観察した．実験からも現在のブロック処理の仕組みでは，利用者が増え到着率が増加すると，未処理のトランザクションが増え，実際の決済にも影響を及ぼすことがわかる．今後はこのシミュレーションを基本として，複数種のブロックチェーンを検証して，最適な基本パラメタの算出などを試みていく．

参考文献

[1] S. Nakamoto, ”Bitcoin: A Peer-to-Peer Electronic Cash System”, https://bitcoin.org/bitcoin.pdf, 2008
[2] コンセンサス・ベイス株式会社, ”図解即戦力 ブロックチェーンのしくみと開発がこれ 1 冊でしっかりわかる教科書.”, 技術評論社 (2019/9/2)
[3] Brewer, Eric A. ”Towards robust distributed systems.” PODC. Vol. 7. No. 10.1145. 2000.
[4] coindesk. ”Is Segregated Witness the Answer to Bitcoin’s Block Size Debate?.” https://www.coindesk.com/segregated-witness-bitcoin-block-size-debate
[5] 笠原正治. ”ビットコインと待ち行列モデル (特集 第 34 回待ち行列シンポジウム).” オペレーションズ・リサーチ = Communications of the Operations Research Society of Japan: 経営の科学 63.8 (2018): 474-479.
[6] Ross, Sheldon M. Introduction to probability models. Academic press, 2014.
[7] Bolch, Gunter, et al. Queueing networks and Markov chains: modeling and performance evaluation with computer science applications. John Wiley & Sons, 2006.
[8] Krishnamoorthy, A., and P. V. Ushakumari. ”A queueing system with single arrival bulk service and single departure.” Mathematical and computer modelling 31.2-3 (2000): 99-108.