高速なリーダー選出アルゴリズムの提案

田中 昌宏[1,a)]　川島 英之[2,b)]

低遅延環境における分散データベースにおいて，故障によるサービスの停止時間の短縮が課題である．リーダー故障から次のリーダー選出までの時間が短かく，スプリットブレインや故障誤検知にも耐性がある手法として，リーダーの次に大きい ID を持つプロセスが過半数の票を集めたときにリーダーとなるアルゴリズムを提案する．暫定的な評価実験を行った結果，提案手法はサービス停止時間を短縮できることを確認した．

1. はじめに

低遅延環境における分散データベースにおける分散トランザクション技術の開発が進められている．低遅延環境を活かし，故障によるサービスの停止時間をできるだけ短くできるリーダー選出アルゴリズムが課題である．

2. 既存のリーダー選出アルゴリズム

既存のリーダー選出手法のうち，リーダー選出が含まれる分散合意プロトコルとして，Raft [1], ZooKeeper [2] がある．また，古いリーダー選出アルゴリズムとして Bully アルゴリズム [3] がある．

2.1 ZooKeeper

ZooKeeper [2] には，FastLeaderElection (FLE) と呼ばれるリーダー選出アルゴリズムが実装されている．FLE のアルゴリズムは以下のとおりである．

(1) リーダーの故障を検知して選挙フェーズに入ると，自分の投票 v をブロードキャスト．
(2) 受信した投票 v' が自身の投票 v より強い場合，投票を v から v' に更新し，投票 v' を再ブロードキャスト．投票の強さが同じ場合，その票をカウント．受信した投票 v' が弱い場合，その票を無視．
(3) finalizeWait ms の間に投票を受信しなかった場合，選挙結果を確定．

2.2 Bully アルゴリズム

Bully アルゴリズム [3] は，リーダー選出を目的としたアルゴリズムとして 1982 年に提案された．

(1) リーダー停止を検知したプロセスは，自分より大きい ID のプロセスに選挙メッセージを送信．
(2) 自分より小さい ID のプロセスから選挙メッセージを受信した場合，OK メッセージを返信し，自分が選挙を実行．
(3) OK メッセージを受信したプロセスは，選挙を終了．
(4) OK メッセージを一定時間経過しても受け取らなかったプロセスがリーダーとなり，他のプロセスに自分の選出を通知．

Bully アルゴリズムの特徴は，大きい ID を持つプロセスが常に勝つことである．

この特徴を捉えて，Basu は，リーダー次に大きい ID を持つプロセスにのみ選挙メッセージを送ることにより，メッセージ送信回数を削減できるというアイデアを提案した [4, 5]．しかしこの方式には欠点があり，リーダー故障の誤検知により，リーダーの変更が起こってしまう．また，Bully アルゴリズムにも Basu のアルゴリズムにも，例えば ID=1,2,3 と ID=4,5 でネットワークが分断されると，リーダーが ID=3 のグループと ID=5 のグループの間でスプリットブレインが発生する．

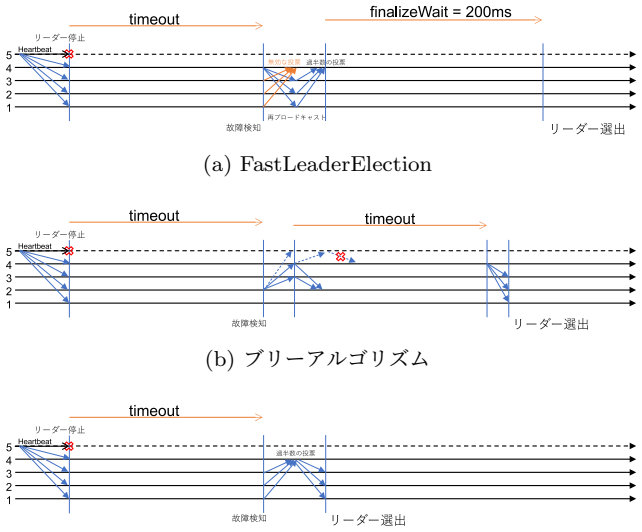
3. 提案手法

提案手法は，Basu のアルゴリズムをベースに，過半数の投票を取り入れたアルゴリズムである．

(1) リーダーの故障を検知したプロセスは，リーダーの次に大きい ID を持つプロセスに選挙メッセージを送信．
(2) 選挙メッセージを受信したら OK メッセージを返信．
(3) 選挙メッセージを過半数のプロセスから受け取った時，リーダーとなり，選出通知をブロードキャスト．
(4) OK メッセージを一定時間受け取らなかった場合，次に大きい ID を持つプロセスに選挙メッセージを送信する．

過半数の投票でリーダーを決めることにより，故障の誤検知やスプリットブレイン対して耐性があるアルゴリズムである．このアルゴリズムの仕様を TLA+で記述し，7 プロセスで 9,184,021 の状態を検査し，パスした．

[1] 慶應義塾大学大学院 理工学研究科
[2] 慶應義塾大学 環境情報学部
Keio University
[a)] masa16.tanaka@keio.jp
[b)] river@sfc.keio.ac.jp



(a) FastLeaderElection

(b) ブリーアルゴリズム

(c) 提案手法

図 1 リーダー選出までの時間経過

FLE，Bully アルゴリズムおよび提案手法におけるリーダー選出までの時間経過を図 1 に示す．(a) FLE では，図には書かれていないが，再ブロードキャストによって通信が錯綜し，それが落ち着くまで finalizeWait=200 ms 待って選挙結果が確定する．(b) Bully アルゴリズムでは，前のリーダーから次のリーダーへの応答がタイムアウトすることによってリーダーが決まる．そのため，故障検知後にさらにタイムアウトによる待ちが発生する．(c) 提案手法では，故障検知したプロセスから過半数の票が集まることによって，直ちにリーダーが決まる．これにより，既存手法よりリーダー選出時間が短いという利点がある．ただし，次のリーダーも同時に故障している場合にはタイムアウトが発生する．

4. 実装

リーダー選挙アルゴリズムの実装には，ZooKeeper の実装を利用した．FastLeaderElection.java を書き換えて，Bully アルゴリズムと提案手法のアルゴリズムを実装した．通信の実装も ZooKeeper のものをそのまま用いた．

5. 性能評価

実験では，1 台のマシンに複数のプロセスを起動し，それらのプロセス間で通信を行った．本稿では暫定的な実験結果を掲載する．CPU は Intel Xeon Gold 6130 2.10GHz である．

リーダーが停止してから新リーダーが選出されるまでの，サービス停止時間を測定した結果を図 2 に示す．模擬的にリーダー故障を起こすため，リーダーのプロセスを STOP シグナルで停止させた．これにより，リーダーの故障検出は，ハートビートのタイムアウトによって引き起こされる．タイムアウトは 100 ms に設定した．プロセス数

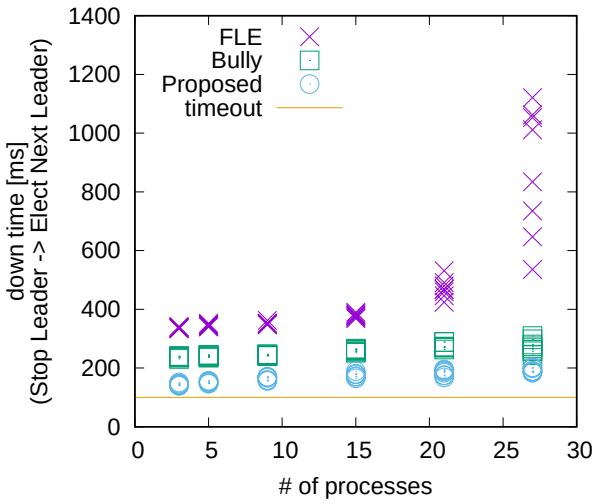


図 2 リーダー故障から次のリーダーが選出されるまでの時間．

を 3,5,9,15,21,27 と変えて測定した．同一の設定で 8 回測定した．図 2 の結果から，提案手法においてサービス停止時間が最も短いことがわかる．

6. まとめ

低遅延環境における分散データベースにおいて，故障によるサービスの停止時間を短縮するためのリーダー選出アルゴリズムとして，次に大きい ID を持つプロセスが過半数の票を集めたときにリーダーとなるアルゴリズムを提案した．暫定的な評価実験により，提案手法は既存手法よりもサービス停止時間を短縮できることがわかった．今後は実際の低遅延ネットワーク環境での実験を行う予定である．

謝辞 本研究の成果は，国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務「ポスト 5G 情報通信システム基盤強化研究開発事業」(JPNP20017) 及び (JPNP16007)，日本学術振興会科研費（22H03596）及びセコム科学技術振興財団の助成により得られたものである．

参考文献

[1] Ongaro, D. and Ousterhout, J.: In Search of an Understandable Consensus Algorithm, *USENIX ATC14*, pp. 305–319 (2014).
[2] Hunt, P., Konar, M., Junqueira, F. P. and Reed, B.: ZooKeeper: wait-free coordination for internet-scale systems, *USENIX 2010*, pp. 11–11 (2010).
[3] Garcia-Molina, H.: Elections in a Distributed Computing System, *IEEE TRANSACTIONS ON COMPUTERS*, Vol. C-31, No. 1, pp. 48–59 (1982).
[4] Basu, S.: An Efficient Approach of Election Algorithm in Distributed Systems, *Article in Indian Journal of Computer Science and Engineering* (2011).
[5] Surolia, J. and Bundele, M. M.: Design and Analysis of Modified Bully Algorithm for Leader Election in Distributed System, *ICAIAA 2019*, pp. 337–347 (2020).