固有の識別子をもたない分散システムにおける耐故障リーダ選挙アルゴリズム

2X−6

松本哲也　若林真一　小出哲士　吉田典可
広島大学工学部

1．まえがき

分散アルゴリズムの基本的問題の1つにリーダ選挙問題がある．これまでに，ネットワークに故障が存在する場合や，各計算機が固有の識別子を持たない場合について個別に研究されている．本稿では，ネットワークが故障を含む場合における従来より効率の良いリーダ選挙アルゴリズムを提案し，さらに固有の識別子を持たない場合におけるリーダ選挙アルゴリズムに拡張する．

2．分散システムのモデル

本稿で考察する分散システムは通信リンクを介して通信するプロセスの集合であると仮定する．各プロセスは独自にアルゴリズムを実行し，共有変数を持たず，プロセス間の通信は2プロセス間にリンクが存在するときのみ直接メッセージを送ることができ，システム全体は非同期である．ネットワークはn個のプロセスと，m本の双方向通信リンクの集合からなり，プロセスをノード，リンクを枝とする任意形状のグラフで表される．リンクは故障していることがあり，故障リンクはメッセージを相手側プロセスに伝えない．すべての故障はアルゴリズム開始時に既に起こっているとし，プロセス故障は考えない．またアルゴリズム開始時，任意のプロセス部分集合が始動プロセスに成り得るとする．各プロセスが持つ情報は，実行するアルゴリズム，n，各自のポートとし，ポートは区別できるとする．

3．問題FLEUとアルゴリズムaFLEU

本節では各プロセスが固有の識別子を持つと仮定する．問題FLEUはリーダと呼ばれるただ1つのプロセスを選び出すことを目的とし，アルゴリズム終了時に全プロセスはリーダを知っており，またアルゴリズムが終了したことも知っていなければならない．

文献[1]ではメッセージ複雑度$O(m+n\log^5 n)$でFLEUを解くアルゴリズムが示されている．本稿で提案するアルゴリズムaFLEUは各プロセスが全故障リンク数の上限値f_l（定数）を知っていると仮定することにより，メッセージ複雑度$O(m+n\log^2 n)$でFLEUを解く．

aFLEUはネットワーク上で根付きスパニング森を構築していき，n/2以上の節点を含む木ができたとき，そのルートがリーダとなる．aFLEUはスパニング森を成長させる手続きGSTと，スパニング森の連結成分である各木のノード数を数える手続きNCからなり，これらは同時に実行される．

スパニング森生成手続きGST

GSTは[1]の終了検知なしのリーダ選挙手法とほとんど同じものを用いる．GSTで生成するスパニング木のリンクをBranchと呼ぶ．Branchの集合は森を構成する．森の各連結成分をフラグメント（以下Fで表す），Fのルートをコアと呼ぶ．最初Branchの集合は空で，各始動プロセスがそのFのコアになる．

GSTにおいて，各Fの動作はコアによって統制され，隣接する他のFと結合するためのリンクを1本選択する．選ばれたリンクはBranchになり，2つのFは合体し，どちらか一方のコアが拡張されたFのコアになる．いずれただ1つのFが選ばれ，そのコアがリーダになる．各Fは（レベル，コアの識別子）というラベルを持ち，各コアはレベル0からスタートする．2つのFが合体するとき，レベルの小さいFが大きいFに吸収され，小さいFのみがラベル更新を行う．同じレベルの場合は両方のFがレベルをインクリメントしてラベルを更新する．

ネットワークは非同期のため，故障リンクを遅延の大きなリンクと区別することは不可能である．各ノードvは隣接リンクを，メッセージを受け取ったか否かによりOperationalかQuietに分類する．Operationalになった枝は集合Linkqueue(v)に加えられる．各ノードはキュー

A Fault-Tolerant Leader Election Algorithm in an Anonymous Distributed System
Tetsuya MATSUMOTO, Shin'ichi WAKABAYASHI, Tetsushi KOIDE, and Noriyoshi YOSHIDA
Hiroshima University

の先頭からリンクを取りだし，同じF内のリンクなら集合Rejected(v)に，他のFを導くなら集合Branches(v)に分類する．

上記のレベル更新の方法を使うと，各ノードのラベル更新回数は高々lognである．また各枝の分類，探索にO(m)メッセージを使うので，ＧＳＴのメッセージ複雑度はO(m+nlogn)となる．

ノード数計算手続きＮＣ

あるFのレベルが上がり，ノードのラベルが更新されると，ＧＳＴの再開と同時にＮＣが開始される．各Fに対し，$2f_1+1$個のトークンを含むメッセージがコアから，Branchリンクを使い，Rejectedリンクを無視しながら深さ優先順にノード数を数えながらF内を探索する．ここでトークンの探索するリンクが未分類ならば，以下の動作を行う．

Linkqueueのとき，まずそのリンクがBranch，Rejected，外向枝のいずれかに分類されるまで待つ．外向枝と分類された場合で，このFのラベル更新が行われないときはトークンを１つ残し，残りのトークンは（存在すれば）次のリンク探索に進む．

またQuietのとき，トークンを１つ残し，残りのトークン（を含むメッセージ）は次のリンク探索に向かう．この後，リンクがOperationalになればLinkqueueのときの動作を行う．

いかなる動作においても増加ノード数が次レベルの必要数に達すれば，処理を中断しＧＳＴのラベル更新手続きを起動する．

次にトークンを割り当てられたリンクＬについて考える．あるＦ$_1$が，Ｌを通してＦ$_2$を吸収するとする．このとき，このトークンをもつノードＡはＦ$_2$のノード数n_{F_2}を知ることができ，必要ノード数に達する場合は，Ｆ$_1$のルートにＧＳＴのラベル更新手続きを起動させる．超えていなくても他のノードで同時にFが成長している可能性があるので全体のノード数を$n_{F_1}+n_{F_2}\times(2f_1+1)$と見積り，見積りが次レベルのノード数に達する場合は全体のノード数を数える．そして実際にしきい値を超えていればラベル更新手続きをFのコアが起動する．超えていなければ現レベルのまま，ＮＣを再起動する．Ｆ$_2$はＦ$_1$に吸収された後，ＡをＦ$_2$部分の仮ルートとしてＮＣを再帰的に起動する．すなわち部分木上を$2f_1+1$トークンを持って深さ優先順で探索し，上記と同様の動作を行う．

Ｆ$_2$のBranch以外の全リンクがRejectedになるとＦ$_2$の探索は終了し，Ａのもつトークンは増加ノード数を記録してＦ$_1$の次のノードの探索へと進む．訪れたノードが既に別のトークンの探索中であればノード数の増加分のみを伝え，さらに次へ進む．故障リンク数の上限はf_1なので，全トークンが故障リンクに割り当てられてデッドロックに陥ることはない．

ある１レベルにおいてＮＣのために使われるメッセージ数は１ノード当たり$(2f_1+1)\times O$(logn)なので，ネットワーク全体ではO(nlogn)となる（f_1は定数）．最大レベル数はlognだから，ＮＣのための総メッセージ数は$O(n\log^2 n)$で抑えられる．

したがって，aFLEUはメッセージ複雑度$O(m+n\log^2 n)$，メッセージサイズO(logn)ビットでFLEUを解く．

4．問題ＦＬＥＡとアルゴリズムａＦＬＥＡ

本節では各プロセスが固有の識別子を持たないとした問題FLEAを考える．

始動プロセスはランダムに識別子を選んだ後，aFLEUを実行する．しかし，複数のコアが同じ識別子を選ぶとアルゴリズムが途中で停止する可能性がある．この場合，識別子をもう一度ランダムに選びなおし，そのＦ中の全ノードをＦの所属からはずし，コアのみのＦからＧＳＴおよびＮＣを再起動させる必要がある．

複数のＦが同じ識別子を持つ確率を全ノードの組み合わせで考えても，識別子を選ぶドメインの要素数を$O(n^2)$以上にすれば増加するメッセージ期待値は$O(m+n\log^2 n)$を超えない．

また，故障リンクの存在により，Ｆの成長が止まったことをコアは知ることはできない．そのため，コアはトークンを回すときに時間を計測する．戻ってきたときにもしノード数に変化がなく，しかも経過時間より大きな遅延を持つリンクの存在確率が1/m以下になるだけの時間が経過していれば識別子の変更を行っても増加メッセージの期待値は$O(m+n\log^2 n)$を超えない．

従って，アルゴリズムaFLEAは問題FLEAを期待メッセージ複雑度$O(m+n\log^2 n)$，メッセージサイズO(logn)ビットで解く．

5．あとがき

今後の課題としては提案アルゴリズムのメッセージ複雑度の改善がある．本研究の成果の一部は文部省科学研究費補助金一般研究（Ｂ）（課題番号04452195）による．

文献 [1]Afek, Y. and Saks, M.:"Detecting global termination conditions in the face of uncertainty," Proc. of the 6th ACM Symp. on PODC, pp.109-124 (1987).