

有限射影平面を利用したメッセージ数 $2n\lfloor\sqrt{n}\rfloor$ の 分散合意プロトコル

中島 周

日本アイ・ビー・エム(株) 東京基礎研究所

ネットワーク内の各ノードが持つ値を使用して計算を分散的に行なう種々の分散アルゴリズムを総称して分散合意プロトコルと呼ぶ。分散合意プロトコルの処理は、各ノードの計算と通信の2つに分けられる。そのうち、通信の部分は異なる分散合意プロトコルで共通である。この通信処理での各ノードの宛先の集合を通信構造と呼ぶ。本論文では、2ラウンドで計算を実行する分散合意プロトコルの通信構造について検討する。有限射影平面の点と線の位置関係の対称性を利用した効率の良い通信構造を使用すると、Lakshman と Agrawala の方法に比べて半分のメッセージ数、 $2n\lfloor\sqrt{n}\rfloor$ 個、で動作する分散合意プロトコルが作成できることを示す。

$2n\lfloor\sqrt{n}\rfloor$ Decentralized Consensus Protocols Using a Finite Projective Plane

Amane Nakajima

Tokyo Research Laboratory, IBM Japan, Ltd.
5-19 Sanbancho, Chiyoda-ku, Tokyo 102

Decentralized consensus protocols are distributed algorithms that compute a function whose arguments are distributed in nodes of a network. Processing of decentralized consensus protocols is divided into two parts: computation in each node and communication. The communication is common in all decentralized consensus protocols. A set of destination nodes defined for each node is called a communication structure. This paper discusses two-round communication structures of the protocols and creates an efficient communication structure using symmetric relation of points and lines in a finite projective plane.

With the proposed communication structure, decentralized consensus protocols that require $2n\lfloor\sqrt{n}\rfloor$ messages are obtained.

1 はじめに

近年、高性能のワークステーションを高速のネットワークで接続した計算環境が一般的となってきた。このように物理的に離れた複数の計算機と通信ネットワークから構成される分散システムでは、従来は1台の計算機による集中処理で実現されてきた機能を分散的に実現しなければならない。しかし、データの分散や非同期に動作する複数の計算主体のために、計算機間での通信が必要となる。分散システムでは、通信に要する時間は、計算機内の処理時間に比べるとかなり大きく、そのために処理に要するメッセージ数を減らすことが重要である。本論文では、2ラウンドの分散合意プロトコルの通信方法について論じ、従来の方法と比較して半分のメッセージ数で動作する通信方法を記述する。

分散合意プロトコルとは、ネットワーク内の各ノードが持つ値をデータとし、各ノードが通信を行ないながら同じ処理を実行してある関数を計算したり、ある述語が真であるかを判定したりする分散アルゴリズムの総称である。分散合意プロトコルの例としては、最大値の計算、合計の計算、分散コミットプロトコル、分散チェックポインティングなどがある[1]-[11]。分散合意プロトコルは、複数のラウンドから構成され、最終ラウンドの実行後、つまりプロトコルの終了時には、すべてのノードが同じ計算結果を有する。1つのラウンドは、各ノードでのメッセージの送受信と受信したメッセージ中のデータを使用した計算から構成される。各ノードは1ラウンド中に複数のノードにメッセージを送信し、また、複数のノードからメッセージを受信する。各ノードが各ラウンドでメッセージを送る宛先ノードの集合を通信構造と呼ぶ。種々の分散合意プロトコルにおいて異なっているのはノードでの計算であり、通信構造はすべての分散合意プロトコルで共通である。よって、効率のよい通信構造を構築することは、すべての分散合意プロトコルの効率を向上させることを意味する。

分散合意プロトコルの効率を判定する主要な尺度はメッセージ数とラウンド数である。メッセージ数は通信の回数を表すために最も重要である。分散合意プロトコルでは、各ノードは各ラウンドで自分が受信する予定のメッセージをすべて受信するまで待ち、その後計算を行ない、次のラウンドへと進む。このように、各ラウンドで受信待ちを行なうため、ラウンド数が少

ないほどプロトコルの実行時間が短くなる。このような理由から、メッセージ数とラウンド数の積を評価基準にすることもある[8]。

分散合意プロトコルは、分散コミットプロトコルとして、Lakshman と Agrawala によって最初に提案された[1], [2]。Lakshman と Agrawala は、有限射影平面を利用した通信構造を採用し、ノード数が n のとき、2ラウンドで $4n\lfloor\sqrt{n}\rfloor$ 個のメッセージを要するプロトコルを作成した。その後、ラウンド数を任意の整数 k に拡張し、メッセージ数を減少させるために種々の通信構造の研究が行なわれた[4]-[11]。一般に、分散合意プロトコルの通信構造は有限射影平面、generalized hypercube、de Bruijn networkなどの数学的な構造に基づいている。2ラウンドの通信構造に関しては、これまでの研究で、generalized hypercubeを利用すると $2n(\sqrt{n}-1)$ 個のメッセージで、de Bruijn networkを利用する場合は $2n\sqrt{n}$ 個のメッセージで分散合意プロトコルが実行できることが示されている[6], [7], [9], [10]。本論文では、有限射影平面を利用する場合も、 $2n\lfloor\sqrt{n}\rfloor$ 個のメッセージで分散合意プロトコルが実行できることを示す。

2 2ラウンドの通信構造

2.1 2ラウンドの分散コミットプロトコル

分散合意プロトコルでは、以下のことを仮定する。

- ネットワーク内にはグローバルクロックなどの集中化されたコントローラは存在せず、処理はすべて分散的に実行され、すべてのノードは対等である。
- ネットワーク内には共有メモリは存在せず、ノード間のすべての通信はメッセージの送受信によって行なわれる。
- 通信は信頼性を有する。ノードからのメッセージは内容を変更されず、紛失や重複なしで有限時間内に宛先に到達する。
- ノードや通信リンクは故障しない。通信はすべて1対1で行なわれ、マルチキャストやブロードキャストは使用しない。

このような仮定のもとで効率良い通信構造を考えることが問題となる。通信時間の上限を仮定しない場合に

は、ノードに故障があれば分散的に合意に達することが不可能であることが証明されている [12], [13]。また、通信時間の上限を仮定し、故障や悪意のあるノードを許して分散的に合意する問題はビザンチン將軍問題となる [14]。

本節では分散合意プロトコルの例として2ラウンドの分散コミットプロトコルを説明する。複数のデータベースにデータを複製する場合には、これらの複製間のデータの一貫性を保たなければならない。このため、トランザクションの原子性を保証することが必要になる。集中制御を用いる方法では、2相コミットプロトコルによってこの問題を解決できる [15]。分散コミットプロトコルでは、トランザクションの原子性を分散的に実現することを目指す。これを実直に1ラウンドで実現すると、各ノードにデータが複製されている場合、各ノードが他のすべてのノードと通信することが必要になり、全体で $n(n-1)$ 個のメッセージが必要となる。

2ラウンドの分散合意プロトコルの有限状態オートマトンを図1に示す [1]。各ラウンドで各ノードは複数のノードに対してメッセージを送信する。ノードが第 i ラウンドでメッセージを送る宛先のノードの集合を第 i 送信集合と呼ぶ。各状態の処理を以下に記す。

- s: 各ノードは、トランザクションを受信したら、そのトランザクションをコミットするかアボートするかを自分自身で決める。アボートするなら、“no” メッセージを自分のノードの第1送信集合に送り、トランザクションをアボートし、状態 a に移る。コミットするなら、“yes¹” メッセージを自分のノードの第1送信集合に送り、状態 w_1 に移る。
- w_1 : 自分のノードが第1ラウンドで受信することになっている“yes¹” メッセージをすべて受信したら、“yes²” メッセージを自分のノードの第2送信集合に送り、状態 w_2 に移る。もし、1つでも“no” メッセージを受信したら、“no” メッセージを自分のノードの第2送信集合に送り、トランザクションをアボートし、状態 a に移る。
- w_2 : 自分のノードが第2ラウンドで受信することになっている“yes²” メッセージをすべて受信したら、トランザクションをコミットし状態 c に移る。もし、1つでも“no” メッセージを受信したら、トランザクションをアボートし、状態 a に移る。

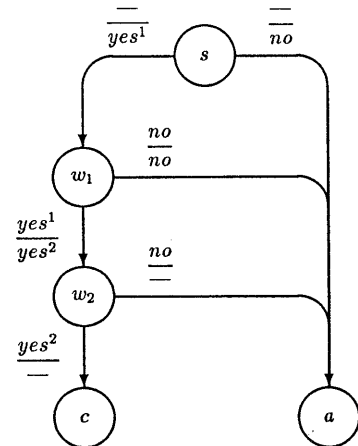


図 1: 2ラウンドの分散コミットプロトコルの有限状態オートマトン (1)

c: コミット状態

a: アボート状態

このプロトコル実行後は、すべてのノードが同じ状態、すなわちコミット状態かアボート状態、に達する。この有限状態オートマトンを実行したときに、トランザクションの原子性を保持できる効率良い通信構造、すなわち各ノードの第1、第2送信集合、を有限射影平面を利用して作成することが本論文の目的である。

2.2 有限射影平面

Lakshman と Agrawala の通信構造を記述し、さらに効率の良い通信構造を議論する前に、有限射影平面について簡単に説明する。

定義 1 射影平面とは、以下の公理 [16] を満たす点の集合と線（線とは点の集合である）の集合である。

公理 1 異なる 2 点を通る直線はただ 1 本存在する。

公理 2 異なる 2 直線はただ 1 点で交わる。

公理 3 4 点で、そのどの 3 点も 1 直線上にないようなものが存在する。

公理 1 と公理 2 は互いに双対的であり、これらの 3 つの公理から公理 3 の双対命題

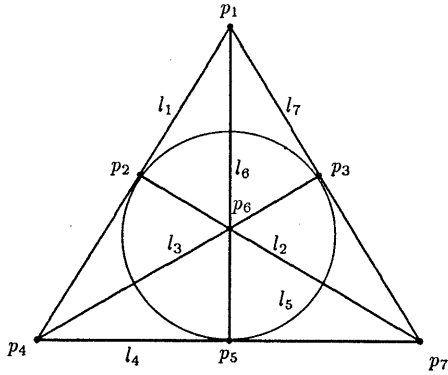


図 2: 位数 2 の有限射影平面 (1)

公理 4 4 直線で、そのどの 3 直線も 1 点で交わらないものが存在する。

が導かれる。よって射影平面は点と線に関して自己双対的である。公理 3 は 1 直線と 2 点のような縮退した場合を除外するために存在する。

定義 2 射影平面の中で、点の数が有限なものを有限射影平面と呼ぶ。

有限射影平面に関して以下の定理が存在する [17], [18]。

定理 1 m は $m \geq 2$ なる整数とする。射影平面 π において次の各性質はすべて互いに同値である。

1. 1 直線はちょうど $(m+1)$ 個の点を含む。
2. 1 点はちょうど $(m+1)$ 個の線上にある。
3. すべての線はちょうど $(m+1)$ 個の点を含む。
4. すべての点はちょうど $(m+1)$ 個の線上にある。
5. π にはちょうど $(m^2 + m + 1)$ 個の点がある。
6. π にはちょうど $(m^2 + m + 1)$ 個の線がある。

m を有限射影平面 π の位数という。

定理 2 q が素数、 α が正整数のとき、位数 $m = q^\alpha$ の有限射影平面が存在する。

位数 2、つまり 7 点と 7 線から構成される有限射影平面の例を図 2 に示す。

2.3 Lakshman と Agrawala の通信構造

Lakshman と Agrawala は有限射影平面を利用して 2 ラウンドの通信構造を構成した。このようにある特定の組合せ論的性質を持つ有限集合族をアルゴリズムに利用することは他にも行なわれている。有限射影平面は分散合意プロトコルでの利用以前には、分散排他制御に用いられていた [19]。また、属性からレコードを検索するためのファイル構成法では有限幾何学が応用された [20], [21]。組合せ論でデザインと呼ばれる集合族を分散アルゴリズムに利用した例もある [22]。

有限射影平面 π の点 i を p_i 、線 i を l_i と表すことにしよう ($1 \leq i \leq m^2 + m + 1$)。Lakshman と Agrawala は有限射影平面の点と線を

$$l_i \ni p_i \quad (1 \leq i \leq m^2 + m + 1) \quad (1)$$

となるように配置できることを証明し、この性質を持った有限射影平面をもとにして通信構造を作成した [1]。また、同じ通信構造を使用した分散チェックポイントイングのアルゴリズムも発表されている [3]。図 2 に示した有限射影平面も式 (1) を満たしている。

ノード i の第 j 送信集合を $S_j(i)$ と書くことにすると、Lakshman と Agrawala の通信構造は、

$$\begin{aligned} S_1(i) = S_2(i) = \{a : l_i \ni p_a, i \neq a\} \\ \cup \{b : l_b \ni p_i, b \neq i\} \\ (1 \leq a, b, i \leq m^2 + m + 1), \end{aligned} \quad (2)$$

となる。ノード i を p_i と l_i に対応させると、ノード i は各ラウンドで l_i 上の点に対応するノードと、 p_i を通る線に対応するノードのうち自分以外のノードにメッセージを送ることになる。このとき、

$$|S_1(i)| = |S_2(i)| = 2m,$$

$$n = m^2 + m + 1,$$

なので、プロトコルに必要となる総メッセージ数は、

$$\sum_{i=1}^n (|S_1(i)| + |S_2(i)|) = 4mn = 4n\lfloor\sqrt{n}\rfloor, \quad (3)$$

となる。

図 2 に示した有限射影平面を使用したときの Lakshman と Agrawala の通信構造を表 1 に示す。

表 1: Lakshman と Agrawala の通信構造の例

i	$S_1(i)$	$S_2(i)$
1	2, 4, 6, 7	2, 4, 6, 7
2	1, 5, 6, 7	1, 5, 6, 7
3	4, 5, 6, 7	4, 5, 6, 7
4	1, 3, 5, 7	1, 3, 5, 7
5	2, 3, 4, 6	2, 3, 4, 6
6	1, 2, 3, 5	1, 2, 3, 5
7	1, 2, 3, 4	1, 2, 3, 4

表 2: 本論文の通信構造の例

i	$S_1(i)$	$S_2(i)$
1	1, 2, 4	1, 6, 7
2	2, 6, 7	1, 2, 5
3	3, 4, 6	3, 5, 7
4	4, 5, 7	1, 3, 4
5	2, 3, 5	4, 5, 6
6	1, 5, 6	2, 3, 6
7	1, 3, 7	2, 4, 7

3 効率の良い通信構造

本章では, Lakshman と Agrawala の通信構造に比べて, メッセージ数を半減する通信構造を構成する。まず通信構造とそれに伴う分散コミットプロトコルの一部の変更を記述し, その後でそれらの正当性を証明する。

3.1 通信構造の構成法

本論文の通信構造は, 式 (1) を満たす有限射影平面を使用し, 以下のように定義される。

$$S_1(i) = \{a : l_i \ni p_a\} \quad (1 \leq a, b, i \leq m^2 + m + 1), \quad (4)$$

$$S_2(i) = \{a : l_a \ni p_i\} \quad (1 \leq a, b, i \leq m^2 + m + 1), \quad (5)$$

となる。ノード i を p_i と l_i に対応させると, ノード i は第 1 ラウンドで l_i 上の点に対応するノードに, 第 2 ラウンドで p_i を通る線に対応するノードにメッセージを送ることになる。このとき,

$$|S_1(i)| = |S_2(i)| = m + 1,$$

であるが, 自分自身に対するメッセージは処理が自ノード内で行なわれるため, メッセージ数の計算時には数えないことになっている。式 (1) が成立するので, プロトコルに必要となる総メッセージ数は,

$$\sum_{i=1}^n (|S_1(i)| - 1 + |S_2(i)| - 1) = 2mn = 2n\lfloor\sqrt{n}\rfloor, \quad (6)$$

となる。

図 2 に示した有限射影平面を使用したときの, 本論文の通信構造を表 2 に示す。

有限射影平面は公理 1, 2, 3 からわかるように点と線は双対である。よって, 有限射影平面の点と線を入れ換えたものも有限射影平面になる。つまり, ある点がある線上にある, ある線はある点を通過する, といった位置関係を逆にして, p_i を l_i に, l_i を p_i にしたものも有限射影平面となる。図 2 に示した有限射影平面をこのようにして変換した有限射影平面を図 3 に示す。図 2 では l_1 上に p_1, p_2, p_4 が存在していたのが, 図 3 では p_1 を l_1, l_2, l_4 が通過するように変換されている。よって,

$$S_1(i) = \{a : l_a \ni p_i\} \quad (1 \leq a, b, i \leq m^2 + m + 1), \quad (7)$$

$$S_2(i) = \{a : l_i \ni p_a\} \quad (1 \leq a, b, i \leq m^2 + m + 1), \quad (8)$$

のように式 (4), (5) の $S_1(i)$ と $S_2(i)$ を入れ換えたものも本論文で記述した通信構造の 1 つとなる。表 2 の $S_1(i)$ と $S_2(i)$ を入れ換えて作った通信構造は, 図 4 の有限射影平面を基にして式 (4), (5) を使って作成したのと同じになる。

3.2 分散コミットプロトコルへの適用

本論文の通信構造を分散コミットプロトコルに使用する場合は, 図 1 で示した処理を図 4 のように変更する必要がある。この変更は, あるノードがアボートしたときに "no" メッセージが他のすべてのノードに到達するために必要となる。同様の変更は, 複数ラウンドの分散コミットプロトコルにおいても文献 [9] で行

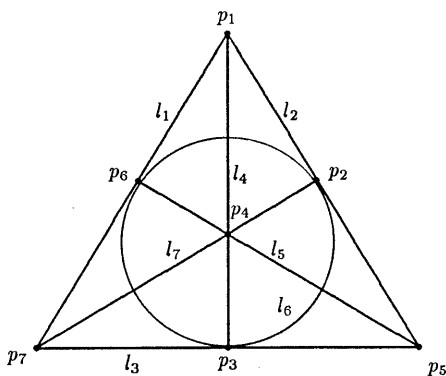


図 3: 位数 2 の有限射影平面 (2)

なわれている。図 1 の処理との変更、追加の部分のみを以下に記す。

s : 各ノードは、トランザクションを受信したら、そのトランザクションをコミットするかアボートするかを自分自身で決める。アボートするなら、“no” メッセージを自分のノードの第 1 送信集合に送り、状態 t_1 に移る。コミットするなら、“yes¹” メッセージを自分のノードの第 1 送信集合に送り、状態 w_1 に移る。

t_1 : “no” メッセージを自分のノードの第 2 送信集合に送り、トランザクションをアボートし、状態 a に移る。

次に、本章で記述した通信構造と分散コミットプロトコルの正当性を証明する。

補題 1 各ラウンドで、ノード i は $(m+1)$ 個のメッセージを、第 1 ラウンドでは p_i を通過するすべての線に対応するノードから、第 2 ラウンドでは l_i 上のすべての点に対応するノードから受信する。

証明 ノード i は第 1 ラウンドで l_i 上の点に対応するノードに、第 2 ラウンドで p_i を通る線に対応するノードにメッセージを送ることになる。これをすべてのノードが実行する。よってノード i は第 1 ラウンドでは p_i を通過するすべての線に対応するノードから “yes¹” メッセージを、第 2 ラウンドでは l_i 上のすべての点に対応するノードから “yes²” メッセージを受信する。公理 1, 2 により 1 つの点を通る線は $(m+1)$ 個、1 つの線の上には $(m+1)$ 個の点が存在する。よって各ノード

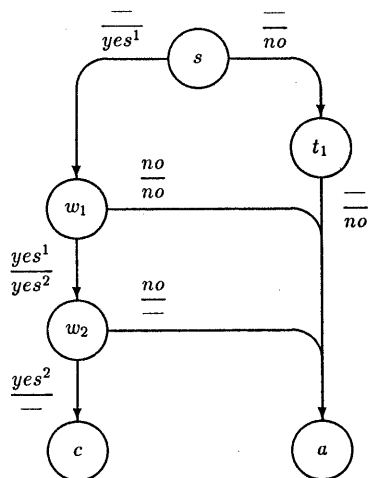


図 4: 2 ラウンドの分散コミットプロトコルの有限状態オートマトン (2)

ドが各ラウンドで受信するメッセージの個数は $m+1$ である。 □

補題 2 あるノードがコミット状態にあれば、すべてのノードは “yes¹” メッセージを送信している。

証明 コミット状態にあるノードをノード i とする。このとき “yes¹” メッセージを送信していないノードが存在すると仮定し、このノードをノード j とする。ノード i は補題 1 により、 l_i 上の $(m+1)$ 個の点から “yes²” メッセージを受信している。これらの点を $p_{i1}, p_{i2}, \dots, p_{i,m+1}$ とする。このとき、ノード $i_1, i_2, \dots, i_{m+1}$ は “yes²” メッセージを送信しているので、状態 w_1 にいるか状態 w_1 を通過したことになる。状態 w_1 に遷移するためには第 1 ラウンドで受信予定のすべての “yes¹” メッセージを受信していることが必要となる。よってこれらのノードは補題 1 により、自分に対応する点を通るすべての線に対応するノードから “yes¹” メッセージを受信していることになる。

ノード j は “yes¹” メッセージを送信していないので、 l_j は $p_{i1}, p_{i2}, \dots, p_{i,m+1}$ のどれをも通過しないことになる。すると、 $p_{i1}, p_{i2}, \dots, p_{i,m+1}$ は l_i 上のすべての点であるので、 l_j は l_i と交わらないことになる。これは公理 2 に矛盾する。よって仮定は誤りである。 □

定理 3 あるノードがアボートを決意すれば、すべてのノードが最終的にはアボートする。

証明 補題 2 により、あるノードがアボートを決意し

たならば、どのノードもコミット状態にはいない。よってすべてのノードが最終的に“no”メッセージを受信することを証明すればよい。

アボートを決意したノードをノード i とする。するとノード i は式 (4) により、第 1 ラウンドで自分に対応する線 l_i を通るすべての点に対応するノードに“no”メッセージを送信する。これらの点を $p_{i_1}, p_{i_2}, \dots, p_{i_{m+1}}$ とする。第 1 ラウンドで“no”メッセージを受信したノード $i_1, i_2, \dots, i_{m+1}$ は式 (5) により、第 2 ラウンドで自分に対応する点を通る線に対応するノードに“no”メッセージを送信する。このとき、式 (1) により $p_{i_1}, p_{i_2}, \dots, p_{i_{m+1}}$ の内の 1 点は p_i であるが、状態 t_1 が存在するので、この点に対応するノード i も第 2 ラウンドで送信を行なう。よって第 2 ラウンドでは、 $p_{i_1}, p_{i_2}, \dots, p_{i_{m+1}}$ を通過するすべての線に対応するノードに“no”メッセージが送られる。

このとき、“no”メッセージを受信しなかったノードがあると仮定し、そのノードを j としよう。すると l_j は $p_{i_1}, p_{i_2}, \dots, p_{i_{m+1}}$ のどれをも通過しないことになる。すると、 $p_{i_1}, p_{i_2}, \dots, p_{i_{m+1}}$ は l_i 上のすべての点であるので、 l_j は l_i と交わらないことになる。これは公理 2 に矛盾する。よって、“no”メッセージを受信しなかったノードがあるという仮定は誤りである。つまり、すべてのノードが“no”メッセージを受信する。□

定理 4 どのノードもアボートを決意しなければ、すべてのノードは最終的にはコミットする。

証明 どのノードもアボートを決意しなければ、すべてのノードは“yes¹”メッセージを各ノードの第 1 送信集合に送信し、状態 w_1 に遷移する。すべてのメッセージは有限時間内に宛先に到達するので、すべてのノードはそれぞれ第 1 ラウンドで受信する予定の“yes¹”メッセージをすべて受信し、“yes²”メッセージを各ノードの第 2 送信集合に送信して、状態 w_2 に遷移する。第 1 ラウンドと同様に各ノードは受信予定の“yes²”メッセージをすべて受信し、コミット状態 c に達する。□

4 おわりに

本論文では、有限射影平面を利用して、メッセージ数 $2n\lfloor\sqrt{n}\rfloor$ の分散合意プロトコルの通信構造を構成した。その通信構造を分散コミットプロトコルに適用した場合のプロトコルを記述し、その正当性を証明した。有限射影平面を利用した通信構造では、メッセージ数

$4n\lfloor\sqrt{n}\rfloor$ のものが Lakshman と Agrawala によって示されていたが、本論文では同じく有限射影平面を利用しながらメッセージ数を半分に減らせることを示した。ラウンド数を一般の整数 k に拡張した複数ラウンド用の通信構造をラウンド数 2 に対して適用すると、generalized hypercube を利用する場合には $2n(\sqrt{n}-1)$ 個のメッセージを要する通信構造が、de Bruijn network を利用する場合には $2n\sqrt{n}$ 個のメッセージを要する通信構造が構成できることが知られていたが、本論文により有限射影平面を基にした場合もほとんど同じメッセージ数で分散合意プロトコルが実現できることが明らかとなった。

完全分散ですべてのノードの処理が等しい場合の各ノードの各ラウンドでの送信メッセージ数、受信メッセージ数の下限が $\sqrt{n}$ になることは理論的に証明できる。このとき、分散合意プロトコルに必要となるメッセージ数は $2n\sqrt{n}$ となる。これを実現する通信構造で、

$$S_j(i) \ni i \quad (1 \leq i \leq n, 1 \leq j \leq 2)$$

を満たすもののメッセージ数が $2n(\sqrt{n}-1)$ となる。

本論文では、ブロッキングプロトコルである 2 相コミットプロトコルを分散化して分散コミットプロトコルを構成したが、文献 [1] などにあるようにノンブロッキングプロトコルである 3 相コミットプロトコル [23] を分散化して分散コミットプロトコルを構成することも可能である。ただしこの場合はラウンド数、メッセージ数ともブロッキングの場合の 2 倍になる。

参考文献

- [1] T. V. Lakshman and A. K. Agrawala, “ $O(n\sqrt{n})$ decentralized commit protocols,” *Proc. 5th Symp. Reliability Distrib. Softw. Database Syst.*, pp. 104-110, Jan. 1986.
- [2] T. V. Lakshman and A. K. Agrawala, “Efficient decentralized consensus protocols,” *IEEE Trans. Softw. Eng.*, vol. SE-12, no. 5, pp. 600-607, May 1986.
- [3] S. H. Son, “An algorithm for efficient decentralized checkpointing,” *Comput. Syst. Sci. Eng.*, vol. 4, no. 1, pp. 27-34, Jan. 1989.

- [4] T. V. Lakshman and A. K. Agrawala, "Communication structure of decentralized commit protocols," *Proc. 6th Int. Conf. Distrib. Comput. Syst.*, pp. 100-107, May 1986.
- [5] A. A. Farrag and R. J. Dawson, "On designing efficient consensus protocols," *Proc. IFIP WG 10.3 Working Conf. Distrib. Processing*, pp. 413-427, Oct. 1987.
- [6] J-C. Bermond, J-C. König, and M. Raynal, "General and efficient decentralized consensus protocols," *Proc. 2nd Int. Workshop Distrib. Algo.*, pp. 41-56, July 1987.
- [7] J-C. Bermond and J-C. König, "General and efficient decentralized consensus protocols II," *Proc. Int. Workshop Paral. Distrib. Algo.*, pp. 199-210, Oct. 1988.
- [8] A. Ghafoor and P. B. Berra, "An efficient communication structure for distributed commit protocols," *IEEE J. Select. Areas Commun.*, vol. 7, no. 3, pp. 375-389, Apr. 1989.
- [9] S. -M. Yuan and A. K. Agrawala, "A class of optimal decentralized commit protocols," *Proc. 8th Int. Conf. Distrib. Comput. Syst.*, pp. 234-241, June 1988.
- [10] C. S. Hsieh, "Decentralized evaluation of associative and communicative functions," *Proc. 9th Int. Conf. Distrib. Comput. Syst.*, pp. 9-11, June 1989.
- [11] S. -M. Yuan, "The communication complexity for decentralized evaluation of functions," *Info. Proc. Let.*, vol. 35, no. 4, pp. 177-182, Aug. 1990.
- [12] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, no. 2, pp. 374-382, Apr. 1985.
- [13] G. Taubenfeld, "One the nonexistence of resilient consensus protocols," *Inf. Proc. Let.*, vol. 37, no. 5, pp. 285-289, Mar. 1991.
- [14] L. Lamport, R. Shostak, and M. Pease, "Byzantine generals problem," *ACM Trans. Prog. Lang. Syst.*, vol. 4, no. 3, pp. 382-401, July 1982.
- [15] J. N. Gray, "Notes on data base operating systems," in *Operating Systems: An Advanced Course*, R. Bayer, R. M. Graham, and G. Seegmüller, Eds. New York:Springer-Verlag, 1978, pp. 393-481.
- [16] 長尾汎, 群とデザイン. 東京:岩波書店, 1974.
- [17] M. Hall, Jr., *Combinatorial Theory*. Blaisdell Publishing Company, 1967. (岩堀信子訳, 組合せ理論. 京都:吉岡書店, 1971.)
- [18] S. S. Anderson, *Graph Theory and Finite Combinatorics*. Chicago IL: Markham Publishing Company, 1970.
- [19] M. Maekawa, "A $\sqrt{n}$ algorithm for mutual exclusion in decentralized systems," *ACM Trans. Comput. Syst.*, vol. 3, no. 2, pp. 145-159, May 1985.
- [20] C. T. Abraham, S. P. Ghosh, and D. K. Ray-Chaudhuri, "File organization schemes based on finite geometries," *Inf. Control*, vol. 12, no. 2, pp. 143-163, Feb. 1968.
- [21] S. P. Ghosh and C. T. Abraham, "Application of Finite Geometry in File Organization for Records with Multiple-Valued Attributes," *IBM J. Res. Dev.*, vol. 12, no. 2, pp. 180-187, March 1968.
- [22] A. Nakajima, "Fault-tolerant distributed match-making with any resiliency," *IEICE Trans.* (電子情報通信学会英文論文誌), vol. E 74, no. 2, pp. 427-434, Feb. 1991.
- [23] D. Skeen, "Nonblocking commit protocols," *Proc. ACM SIGMOD*, pp. 133-142, Apr.-May 1981.