

# Voting を用いた分散システムの可用性の最大化

Maximizing the Availability of Distributed Systems that Use Voting

松井 佑記<sup>†</sup>  
Yuki Matsui

小島 英春<sup>†</sup>  
Hideharu Kojima

土屋 達弘<sup>†</sup>  
Tatsuhiko Tsuchiya

## 概要

本研究では、ネットワーク上に分散したデータのレプリカに対する可用性の最大化について議論する。多数決投票システムは、障害が生じる状況下で、データのレプリカの一貫性を提供する一般的なメカニズムである。多数決投票システムにおけるレプリカへの票割当ては、このシステムにおけるデータ可用性に大きな影響を与える。本研究では、最適な票割当てを MAX-SMT 問題と呼ばれる組合せ最適化問題として定式化する。この定式化によって高速な MAX-SMT ソルバを用いることができ、票割当て問題の解を求められる。ネットワーク下での故障・回復モデルを用いてデータ可用性を求めることで、この手法の効果を評価する。その結果、障害が生じる状況において、提案手法によって求めた最適な票割当てを用いることで、データ可用性が大きく改善できることが分かった。

キーワード 一貫性を保持したデータ; 多数決投票; 可用性; ネットワークパーティション; SMT;

## 1 はじめに

本研究では、障害発生時に一貫性を保持したデータを確保するときの可用性を最適化する手法の提案と評価を行う。障害発生時に一貫性を保持したデータを提供するための一般的な手法としてレプリケーションがある。レプリケーションとは、ネットワーク上の異なるノードにデータのレプリカを与えることである。しかし、データのレプリカの一貫性を保証することは容易ではない。例えば、ネットワークが複数のパーティションに分割されたとき、異なるパーティションに属するデータがそれぞれ変更されると、データの一貫性が失われてしまうことになる。

多数決投票 (*Majority voting*) は多重化したデータの一貫性やパーティション耐性を保証するための一般的なメカニズムである [1, 2]。本研究では、多数決投票によって実現される一貫性を保持したデータの可用性を最適化する。このメカニズムでは、各ノードに非負整数の票を割当てる。それぞれのノードは、自身のノードと他のノードから過半数の票を得られる場合のみ動作してよいものとする。互いに接続されていない 2 つ

のノードグループは同時に過半数の票を得ることが出来ないので、レプリカ間でのデータの一貫性が保持される。過半数投票は 1970 年代の後期に提案されたが [1, 2], Apache ZooKeeper のような有名なソフトウェアシステムで利用され始めたのは近年になってからである。

多数決投票によるデータの可用性は、ノードの票割当てによって異なることが知られている。これは、適切な票割当てを行うことで可用性を最大化可能なことを意味する。実際、票割当てについて多くの文献で研究されている。しかし先行研究では、現実的な状況下で最適な票割当てを必ずしも行えなかった。この要因には次のようなものが挙げられる。

- 1) パーティションのないネットワークが想定されている。
- 2) 得られた票割当てが最適であることが保証されない。
- 3) ネットワークモデルが現実的でない。

本研究では、ネットワークの故障やネットワークの分割の影響がある確率に基づく故障モデルを想定する。このモデルに基いて、最適な票割当て問題を MAX-SMT 問題として定式化する。MAX-SMT 問題は有名な組合せ最適化問題である最大充足可能性 (MAX-SAT) 問題を一般化した問題である。MAX-SAT 問題とは、与えられたブール式に対して充足される項の数が最大となるように真偽値を割り当てを決定する問題である。MAX-SMT 問題は MAX-SAT 問題に背景理論を付加することで一般化したものである。例えば、線形算術によるブール式の制約を用いることができる。

票割当て問題を MAX-SMT 問題に定式化するアプローチには次のような利点がある。まず、近年高速化が著しい MAX-SMT ソルバを利用可能な点である。次に、MAX-SMT 問題では簡潔な定式化を可能にする。それに対して、票割当て問題は従来用いられてきた線形計画問題といった最適化問題に定式化することは困難である。

提案手法の効果は実験を行うことによって示す。その実験では、まずネットワークモデルを構築する。そして、確率に基づく故障モデルを用いてネットワークコンポーネントの故障と回復をシミュレーションする。得られたネットワーク分割の確率を用いて最適な票割当てを MAX-SMT 問題として定式化し、既存の MAX-SMT ソルバを適用してこの問題を解く。最適な票割当てを得るために要した MAX-SMT ソルバ実行時間と、その票割当てによって達成される可用性を評価する。

<sup>†</sup> 大阪大学大学院情報科学研究科, Graduate School of Information Science and Technology, Osaka University

### 1.1 関連研究

Spasojevic と Berman は最適な票割当てを発見するアルゴリズムを提案している [3]。しかし、このアルゴリズムではネットワークに完全グラフを仮定していたり、イーサネットのような媒体を持つネットワークが想定されている、つまり、ネットワークパーティションが発生しないことを想定している。

ネットワークトポロジーを考慮したノードへの票割当ては文献 [4, 5, 6, 7, 8, 9] で議論されている。これらの文献では、ツリー構造やリング構造のような特定のネットワークトポロジーに対する票割当てが行われている。これらのモデルでは、ネットワークトポロジーは無向グラフとしてモデル化されており、グラフの各頂点で表される各ノードがルータとして投票に関与している。このようなモデルは各処理ユニットが互いに直接的に接続されているような並列コンピュータに適している。一方で、広域ネットワーク上に実現されるネットワークアプリケーションには適していない。

広域ネットワーク上の多数決投票システムに関する研究は文献 [10, 11, 12] で行われている。文献 [10] では、ノードの信頼性を既知としたときの票割当て手法 [3] による可用性と均等な票割当てによる可用性を比較している。上記の通り文献 [3] の手法では、ネットワークトポロジーの信頼性が十分高く、イーサネットのような媒体を持つネットワークトポロジーのみを想定している。文献 [11, 12] では、票割当ての可用性に関する議論はされていないが、性能に関する最適化が研究されている。

### 1.2 構成

本研究の構成は次の通りである。2章では、システムのモデルと多重化したデータの可用性の定義について示す。3章では、票割当て問題を MAX-SMT 問題へと定式化する手法を提案する。4章では、シミュレーションによる実験を行う。実験では、提案手法の効果と最適な票割当てによる利点を評価する。最後に5章で、本研究の結論を示す。

## 2 モデル

ネットワークに接続されている  $n$  ノードで構成された分散システムを想定する。 $n$  ノードの集合を  $\Pi = \{1, 2, \dots, n\}$  と表す。ノードは正常、もしくは故障の2つの状態を取りうるものとする。任意の正常な2ノードに対しては、ネットワーク障害に依存して、連結性を持つか、連結性を持たないかの2つの状態を取りうる。互いに連結する正常なノードの集合をパーティショングループと呼ぶ。

分散システムに、多重化したデータの一貫性を保つ手法として多数決投票を適用する。システムの各ノードには非負整数で与えられる票数を割り当てる。以降、ノード  $i$  に割当てる票数を  $v(i)$  と表すことにする。データの一貫性を保持するために、システム的全票数のうち過半数の票数を持つパーティショングループに属するノードのみデータにアクセスできるとする。過半数の票数を持つパーティショングループは最大で1つしか

いたため、データの一貫性が保証される。多重化したデータへの可用性は、データにアクセス可能な確率で表される。

次のような確率に基づく故障モデルを与える。 $\Pi$  の空集合でない部分集合  $g$  に対して、パーティショングループ  $g$  が構成される確率を  $P(g)$  とする。このモデルは、票割当ての先行研究で扱われているネットワーク故障モデルを包括する点で一般的である。例えば、完全グラフのネットワークモデルは、 $g$  に含まれる全てのノードが正常で、その他のノードが故障する確率として  $P(g)$  が与えられる。

票割当て  $v$  による可用性は次の通りである。

$$A(v) = \sum_{g \in 2^\Pi, g \neq \emptyset} \left( P(g) * \left[ \sum_{p \in g} v(p) > \frac{total}{2} \right] \right)$$

ここで、 $2^\Pi$  は  $\Pi$  の冪集合、 $total = \sum_{p \in \Pi} v(p)$  を表し、 $[P]$  は Iverson の記号で論理式  $P$  が真ならば 1、偽ならば 0 を取る。 $A(v)$  が上記の式で与えられる根拠は次の通りである。過半数の票を有するパーティショングループのデータのみが利用可能である。また、2つのパーティショングループが同時に過半数の票を得ることはない。そのため可用性は、過半数の票を有するパーティショングループを構成する確率の総和に等しい。

## 3 票割当て問題の定式化

### 3.1 重み付き部分 MAX-SMT 問題

最適な票割当て問題を MAX-SMT 問題として定式化する。ここでは、MAX-SMT 問題として、最も一般的な重み付き部分 MAX-SMT 問題を考える。この問題のインスタンスは  $(HC, SC, w)$  によって定義される。ここで、 $HC$  はハード制約の集合、 $SC$  はソフト制約の集合、 $w$  はそれぞれのソフト制約に非負整数の重みを割り当てる。問題の解には、次の制約を満たすように値  $a$  の割当てを定める。

- (i) 全てのハード制約を充足する。
- (ii) 充足するソフト制約による重みの総和、つまり、

$$\sum_{c \in SC} w(c) * [c(a) = true]$$

を最大化する。ここで、 $c(a)$  は  $a$  の値が割り当てられているとき、ソフト制約  $c$  の真偽値である。

MAX-SMT 問題の背景理論には、整数線形算術を用いる。この場合には、ハード制約やソフト制約に次のようなブール式の組合せとなる。

- (i) 整数定数と変数、加減演算子 (+, -)、不等号 (=, <, ≤, >, ≥) による整数制約で構成される。
- (ii) ブール定数と変数によるブール式で構成される。
- (iii) (i) と (ii) の両方で構成される。

### 3.2 問題の定式化

ここでは、可用性の最大化問題を重み付き部分 MAX-SMT 問題として定式化する。問題の定式化において、次のような変数を定める。

- $p \in \Pi$  に対して, 非負整数の変数  $v_p$  .
- $\Pi$  の空集合でない部分集合  $g$  に対して, ブール型の変数  $x_g$  .

変数  $v_p$  は  $p$  に割り当てる票数を表す. 変数  $x_g$  はノードの部分集合  $g$  が過半数の票を有するとき真, そうでないとき偽の値を取る. これらの変数に値を割り当てると, 可用性は次のように決定する.

$$\sum_{g \in 2^\Pi, g \neq \emptyset} P(g) * [x_g = true].$$

提案した定式化においては, ハード制約は  $x_g$  に対して上で述べた意味を与える. 空集合でないノードの部分集合  $g$  それぞれに対して, 次のようなハード制約を与える.

$$x_g = true \Leftrightarrow 2 \sum_{p \in g} v_p > \sum_{p \in \Pi} v_p$$

上式はブール変数  $x_g$  が真の値を取るのには, システム全体の票数が  $g$  の有する票数の 2 倍より少ない, すなわち,  $g$  が過半数の票数を有するときであることを表す.

それぞれのソフト制約は空集合でないノードの部分集合  $g$  を表し,  $g$  がパーティショングループを構成することを表す.  $g$  によって, 次のようなソフト制約を与える.

$$x_g = true$$

それぞれのソフト制約  $c_g = (x_g = true)$  に対する重み  $w(c_g)$  は  $P(g)$  を正規化することで得られる整数値を持つ. つまり, ある定数  $const$  を用いて,  $w(c_g) = P(g) * const$  で表される.

次式により, この MAX-SMT 問題の解によって可用性が最大化されることが分かる.

$$\begin{aligned} & \frac{1}{const} * \sum_{c \in SC} w(c) * [c(a) = true] \\ &= \sum_{g \in 2^\Pi, g \neq \emptyset} P(g) * [x_g = true] \end{aligned}$$

### 3.3 定式化の例

定式化を行う例として, 3 ノードで構成されるシステムを考える. この場合は単一ノードが票を割当てられるシングルトン, もしくは全てのノードが 1 票ずつ割当てられることが知られている [13].

簡単のために, 広域ネットワーク上のノードやリンクの信頼性は十分高く, そのネットワークに接続するノードやリンクが故障を受けやすいとする. ノードの信頼性は 0.95, リンクの信頼性は 0.99 であるとする. この仮定のもと, 任意の  $g \in 2^\Pi, g \neq \emptyset$  に対して  $P(g)$  は次のようになる.

$$\begin{aligned} P(\{1\}) &= 0.0128, & P(\{2\}) &= 0.0128 \\ P(\{3\}) &= 0.0128, & P(\{1, 2\}) &= 0.0526 \\ P(\{1, 3\}) &= 0.0526, & P(\{2, 3\}) &= 0.0526 \\ P(\{1, 2, 3\}) &= 0.8319 \end{aligned}$$

ハード制約の集合として, ノードの部分集合  $g$  が過半数の票を有する条件を  $x_g$  の値に与える.

$$\begin{aligned} x_{\{1\}} = true &\Leftrightarrow 2v_1 > v_1 + v_2 + v_3 \\ x_{\{2\}} = true &\Leftrightarrow 2v_2 > v_1 + v_2 + v_3 \\ x_{\{3\}} = true &\Leftrightarrow 2v_3 > v_1 + v_2 + v_3 \\ x_{\{1,2\}} = true &\Leftrightarrow 2 * (v_1 + v_2) > v_1 + v_2 + v_3 \\ x_{\{1,3\}} = true &\Leftrightarrow 2 * (v_1 + v_3) > v_1 + v_2 + v_3 \\ x_{\{2,3\}} = true &\Leftrightarrow 2 * (v_2 + v_3) > v_1 + v_2 + v_3 \\ x_{\{1,2,3\}} = true &\Leftrightarrow 2 * (v_1 + v_2 + v_3) > v_1 + v_2 + v_3 \end{aligned}$$

ソフト制約の集合とそれに関する重みは,  $const = 10000$  の正規化を行うと次のようになる.

$$\begin{aligned} x_{\{1\}} = true & \quad weight : 128 \\ x_{\{2\}} = true & \quad weight : 128 \\ x_{\{3\}} = true & \quad weight : 128 \\ x_{\{1,2\}} = true & \quad weight : 526 \\ x_{\{1,3\}} = true & \quad weight : 526 \\ x_{\{2,3\}} = true & \quad weight : 526 \\ x_{\{1,2,3\}} = true & \quad weight : 8319 \end{aligned}$$

MAX-SMT ソルバを用いると, この問題に対する最適解の一つが得られる. 例えば, 次のような変数割当てが最適解の一つとなる.

$$\begin{aligned} v_1 = 1, v_2 = 1, v_3 = 1, \\ x_{\{1\}} = false, x_{\{2\}} = false, x_{\{3\}} = false, \\ x_{\{1,2\}} = true, x_{\{1,3\}} = true, x_{\{2,3\}} = true, \\ x_{\{1,2,3\}} = true \end{aligned}$$

この最適解によって, 制約に関する重みの総和は 9897 が得られる. これは可用性の最大値が 0.9897 であることを表す.

### 3.4 最適化

定式化に関する問題として, 次のような簡単な最適化を行う. まず, 次のようなソフト制約とハード制約を取り除くことが可能である.

$$\begin{aligned} x_\Pi = true \Leftrightarrow 2 \sum_{p \in \Pi} v_p > \sum_{p \in \Pi} v_p \\ x_\Pi = true \end{aligned}$$

全ての  $p \in \Pi$  に対して  $v_p = 0$  でなければ, ハード制約は常に満たされる. 全ての  $p \in \Pi$  に対して  $v_p = 0$  となる場合は最適解とならない.

2 つ目の最適化として,  $P(g) = 0$  となる  $g$  についてのソフト制約とハード制約を取り除くことが可能である. つまり,  $g$  となるパーティショングループが構成されなければ, それに対応するソフト制約とハード制約を省略することが出来る. これは,  $x_g = true$  となるような  $g$  はソフト制約による重みが 0 となるため解に影響を与えないことによる. 実際に, ネットワークの性質によって, あるパーティショングループは全く発生しないことが多い.

## 4 評価

提案手法の効果を示すために, シミュレーションによる検証を行う. 表 1 に検証を行ったマシンの性能を示す.

表 1 シミュレーションを行ったマシン

OS	Windows 7 Enterprise
CPU	Intel Core i7-2600 3.40GHz
Memory	12GB
MAX-SMT solver	Yices 1.0

実験では、現実的なネットワークモデルを用いて、ネットワーク上に配置されたノードに対して最適な票割当てを、提案手法によって行う。本節以降では、多重化したデータを持つノードをサーバ、それ以外のネットワーク上に配置されているノードをルータと区別して呼ぶ。

提案手法では、各パーティショングループの発生確率が必要である。そこで 2 章で説明した確率に基づく故障モデルを用いてシミュレーションを行うことで発生確率を計算する。シミュレーションについては、4.1 節に記述する。

確率に基づく故障モデルを用いて、最適な票割当て問題を MAX-SMT 問題として定式化する。既存の MAX-SMT ソルバである Yices 1.0 [14] を用いて MAX-SMT 問題の解を求める。提案手法のスケラビリティを示すために、4.2 節では MAX-SMT ソルバを用いて MAX-SMT 問題の解を求めるのに要した時間を示す。また、均等な票割当てと最適な票割当てから得られるデータの可用性についての比較を行う。

#### 4.1 シミュレーション

表 2 にシミュレーションの設定を示す。ネットワークポロジのモデルには Barabási-Albert モデル (BA モデル) を用いた [15]。このモデルを用いて、ルータ数 10 とルータ数 50 のネットワークモデルを生成した。

ネットワーク上で各パーティショングループが発生する確率を取得する。そのために、故障と回復の状態を取り得る故障モデルを用いたシミュレーションを行った。このシミュレーションでは、サーバ、ルータ、ルータ間リンク、サーバ・ルータ間リンクがそれぞれ故障と回復状態を取る。図 1 に、サーバ数が  $3(n=3)$  とした場合のシステムの概略を示す。

2 つのネットワークモデルに対して、それぞれ 100,000 単位時間のシミュレーションを行った。文献 [16] をもとに、ノードやリンクが故障するまでの時間は指数分布、故障から回復するまでの時間はパレート分布に従うとした。しかしながら、信頼性の高いネットワーク上でのデータ可用性を検証するために、故障率は意図的にオーダー単位で増加させた。

ここで、シミュレーションによって  $P(g)$  を得る手順について簡単に説明する。簡単のために、図 1 に示すようにサーバ数は 3 とする。サーバ数 3 の場合、3.3 節で述べたようにパーティショングループは次の 7 つが発生し得る。

$$\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$$

ノードやリンクの故障がネットワークパーティションを引き起こし得る。このときのパーティショングループの発生時間を記録する。

表 2 シミュレーションの設定

シミュレーション時間	100,000 単位時間
ネットワークポロジ	BA トポロジ
サーバ数	2, 3, ..., 10
ルータ数	10 or 50
故障時間	指数分布 $\lambda = 0.0318443$
回復時間	パレート分布 Pareto(60, 2.3)

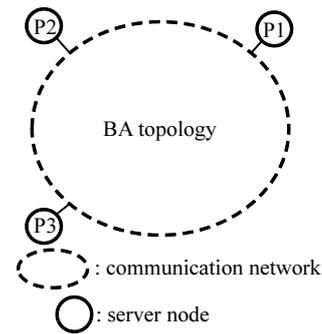


図 1 シミュレーションを行ったシステムの概略

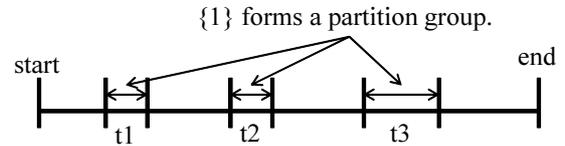


図 2 パーティショングループ {1} が発生した区間

シミュレーションを実行したときにパーティショングループ  $\{1\}$  が発生する区間を図 2 に示す。図 2 は、シミュレーション中にパーティショングループ  $\{1\}$  が 3 区間発生したことを示している。

その他の 6 つのパーティショングループの発生した時間も同様に求められる。各パーティショングループ  $g$  の発生確率  $P(g)$  は、シミュレーション時間のうち  $g$  が発生した時間の占める割合で表される。ソフト制約  $c = (x_g = true)$  に対する重み  $w(c)$  は、 $P(g)$  を非負整数に正規化することで得られる。

#### 4.2 結果

シミュレーションから得られた結果を用いて、最適な票割当てを得る提案手法を適用する。ここではその検証の結果を示す。

まず、問題の解を Yices によって求めた際の計算時間を表 3 に示す。表 3 はネットワーク上のルータ数が 10 の場合に得られた計算時間を表す。この結果からサーバ数が 10 の場合であっても 30 分以内に計算時間が抑えられていることがわかる。

表 3 最適な票割当てに要した計算時間 (ルータ数 10 のネットワーク)

サーバ数	計算時間 [s]
2, 3, ..., 7	< 1
8	9
9	663
10	1330

ルータ数が 50 の場合の結果はルータ数 10 の場合同様 30 分以内に抑えられている。現実には、一貫したデータを保持するためのレプリカ数は 10 以下の数が用いられる。例えば、マイクロソフトの Niobe large-scale enterprise strage system では、10 台以下のマシンから構成される大域的状态管理 (GSM) モジュールを用い、重要なシステムの大域的状态を管理している。そのため、提案手法は現実的な規模まで拡張できると言える。

次に、最適な票割当てによってどの程度可用性が改善されるのかを示す。図 3, 図 4 にそれぞれルータ数を 10, 50 とした場合の可用性を示す。横軸はデータを多重化したサーバ数、縦軸は可用性を示す。均等な票割当て (Existing method) と最適な票割当て (Proposed method) を比較している。均等な票割当てでは、サーバ数が偶数の場合にはランダムに選択したサーバに追加の票を与える [13]。文献 [13] の理論的な結果によると、追加の票を与えることで常に可用性が向上されることが示されている。

図 3, 図 4 における可用性はサーバ数の接続性に影響を与える故障が発生したときの条件付きの可用性を表し、次の式で正規化している。

$$\frac{availability - P(\Pi)}{1 - P(\Pi)}$$

ここで、 $P(\Pi)$  は全サーバが正常に接続されている確率を表す。この条件付きの可用性は故障耐性の度合いを可視化している。

これらの図に見られるように、提案手法による可用性は改善されている。特に、サーバ数が 5 未満のときに大きな改善が見られる。既存手法では、サーバ数が増加した場合にも可用性の減少が見られる。一方で提案手法では、サーバ数が増加するにつれて可用性が増加することがわかる。

## 5 おわりに

本研究では、多数決投票を用いたネットワークに分散するデータのレプリカの可用性を最大化する手法を提案した。可用性の最大値は票の割当てを最適化することで得られる。提案手法では票割当て問題を MAX-SMT 問題に対応付ける。定式化には確率に基づく故障モデルを用いることで、ネットワーク故障による影響を取り入れた。提案手法の効果を示すために、ネットワークモデルを用いて検証を行った。シミュレーションでは、確率に基づく故障モデルをネットワークに取り入れ、最適な票割当てを求めた。検証の結果、提案手法ではシステムの

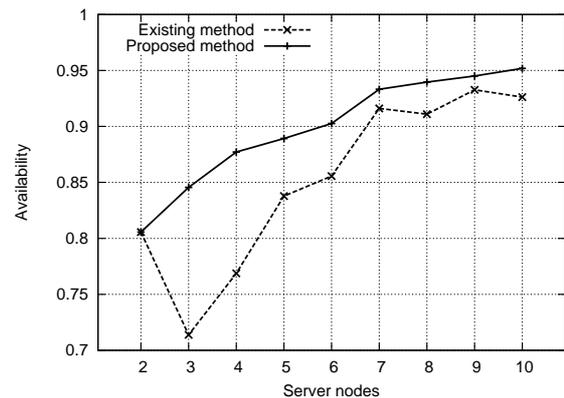


図 3 サーバの故障、もしくは複数のパーティショングループが発生したとする条件付き可用性 (ルータ数 10 のネットワーク)

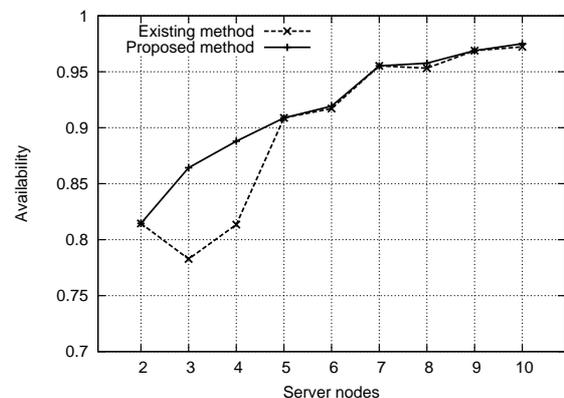


図 4 サーバの故障、もしくは複数のパーティショングループが発生したとする条件付き可用性 (ルータ数 50 のネットワーク)

ノード数は 10 ノードまでの規模に適用することができることがわかった。このノード数は実際の利用に十分適用可能な規模であり、最適な票割当てによってシステムのデータ可用性が改善できることを示した。

## 謝辞

本研究にあたって、先行研究を行って下さった橋下洋氏に深く感謝致します。

## 参考文献

- [1] R. H. Thomas, "A majority consensus approach to concurrency control for multiple copy databases," *ACM Trans. Database Syst.*, vol. 4, no. 2, pp. 180–209, Jun. 1979.
- [2] D. K. Gifford, "Weighted voting for replicated data," in *Proceedings of the seventh ACM symposium on Operat-*

- ing systems principles*, ser. SOSP '79, 1979, pp. 150–162.
- [3] M. Spasojevic and P. Berman, “Voting as the optimal static pessimistic scheme for managing replicated data,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 5, no. 1, pp. 64–73, jan 1994.
- [4] D. Barbara and H. Garcia-Molina, “The vulnerability of vote assignments,” *ACM Trans. Comput. Syst.*, vol. 4, no. 3, pp. 187–213, Aug. 1986.
- [5] —, “The reliability of voting mechanisms,” *IEEE Trans. on Computers*, vol. C-36, no. 10, pp. 1197–1208, oct. 1987.
- [6] K. Diks, E. Kranakis, D. Krizanc, B. Mans, and A. Pelc, “Optimal coterie and voting schemes,” *Information Processing Letters*, vol. 51, no. 1, pp. 1–6, 1994.
- [7] T. Ibaraki, H. Nagamochi, and T. Kameda, “Optimal coterie for rings and related networks,” *Distributed Computing*, vol. 8, pp. 191–201, 1995.
- [8] C. H. Papadimitriou and M. Sideri, “Optimal coterie,” in *Proceedings of the tenth annual ACM symposium on Principles of distributed computing*, ser. PODC '91, 1991, pp. 75–80.
- [9] Z. Tong and R. Kain, “Vote assignments in weighted voting mechanisms,” *IEEE Trans. on Computers*, vol. 40, no. 5, pp. 664–667, May 1991.
- [10] Y. Amir and A. Wool, “Evaluating quorum systems over the internet,” in *Proc. 26th Int'l Symp. on Fault-Tolerant Computing (FTCS-26)*, Jun. 1996, pp. 26–35.
- [11] A. Gupta, B. M. Maggs, F. Oprea, and M. K. Reiter, “Quorum placement in networks to minimize access delays,” in *Proc. 24th ACM Symp. on Principles of Distributed Computing (PODC'05)*, 2005, pp. 87–96.
- [12] F. Oprea and M. Reiter, “Minimizing response time for quorum-system protocols over wide-area networks,” in *Proc. 37th Int'l Conf. on Dependable Systems and Network (DSN 2007)*, june 2007, pp. 409–418.
- [13] H. Garcia-Molina and D. Barbara, “How to assign votes in a distributed system,” *J. ACM*, vol. 32, no. 4, pp. 841–860, Oct. 1985.
- [14] B. Dutertre and L. M. de Moura, “A fast linear-arithmetic solver for DPLL(T),” in *Proc. of 18th Conf. on Computer Aided Verification (CAV 2006)*, ser. LNCS, vol. 4144, Aug. 2006, pp. 81–94.
- [15] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, pp. 509–512, October 1999.
- [16] P. Kuusela and I. Norros, “On/off process modeling of IP network failures,” in *Proc. 40th Int'l Conf. on Dependable Systems and Network (DSN 2010)*, 2010, pp. 585–594.