情報処理学会論文誌 Vol.55 No.3 1177–1185 (Mar. 2014)

推薦論文

多数決投票を用いた分散システムの可用性の最大化

松井 佑記[1,a)] 小島 英春[1,b)] 土屋 達弘[1,c)]

受付日 2013年6月2日, 採録日 2013年12月4日

概要：本研究では，ネットワーク上に分散したレプリカからなる多重化データの可用性を最大化する手法について議論する．障害が生じる状況下において，データのレプリカに対する一貫性を提供する機構として，多数決投票システムが知られている．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．そこで，本研究では，可用性を最大化する票割当てを MAX-SMT 問題と呼ばれる組合せ最適化問題として定式化し，高速な MAX-SMT ソルバを用いて，票割当て問題の解を求める手法を提案する．提案手法の有効性を評価するため，トポロジと故障・修復を考慮したネットワークのモデルを構築し，その上で稼動する分散システムを想定して提案手法の評価を行った．その結果，レプリカ数 10 以下のシステムであれば，実用的な時間で提案手法を適用し可用性を最大化する票割当てを求めることができること，および，最適な票割当てを用いることで，障害が生じる状況における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：レプリケーション，多数決投票，可用性，パーティション，SMT

Maximizing the Availability of Distributed Systems That Use Voting

Yuki Matsui[1,a)] Hideharu Kojima[1,b)] Tatsuhiro Tsuchiya[1,c)]

Received: June 2, 2013, Accepted: December 4, 2013

Abstract: We address the issue of maximization of the availability of replicated data that are distributed in a wide area network. We consider a system that uses majority voting, which is a common mechanism for providing consistency of replicated data in the presence of failures. The data availability provided by this mechanism critically depends on the vote assignment to the replicas. In this paper we formulate the problem of finding the optimal vote assignment into a specific form of a combinatorial optimization problem, namely the MAX-SMT problem. This formulation allows us to use a modern, fast MAX-SMT solver to solve the vote assignment problem. To evaluate the effectiveness of this approach, we build a failure-repair model of underlying networks and perform an experiment using that model. The results of the experiment show that if the number of replicas does not exceed 10, then the proposed approach can produce the optimal vote assignment practically, and that data availability can be significantly improved using the optimal vote assignment in the presence of failures.

Keywords: replication, majority voting, availability, partition, SMT

1. はじめに

分散システムにおけるデータのレプリケーションでは，データの一貫性の実現が重要な課題となる．特に，厳密な一貫性が求められるアプリケーションでは，その実現は容易ではない．たと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．

データの厳密な一貫性を保証するためのよく知られた機

[1] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565–0871, Japan
[a)] y-matsui@ist.osaka-u.ac.jp
[b)] hkojima@ist.osaka-u.ac.jp
[c)] t-tutiya@ist.osaka-u.ac.jp

本論文の内容は 2012 年 9 月の平成 24 年度情報処理学会関西支部支部大会にて報告され，支部長により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

構として，多数決投票が存在する [1], [2]．この機構では，レプリカを管理する各ノードに非負整数の票を割り当て，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データにアクセスしてよいものとする．過半数の票を持つノード集合をどのように 2 つ選んでも，それらは共通のノードを有するため，同時に複数のアクセスが起こらないこと，すなわち，相互排除が実現されるので，データの一貫性を保証することができる．多数決投票は 1970 年代の後期に提案されたが [1], [2]，Apache ZooKeeper のような広く知られた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 [3]．

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．よって，票割当てについて多くの研究がなされているが，1.1 節で詳述するように，先行研究では現実的な状況で可用性を最大化する票割当てを求めることはできない．この要因として以下の点があげられる．

1) ネットワーク分割が起こらないことが想定されている．
2) ネットワークモデルが現実的でない．
3) 得られた票割当てが最適であることが保証されていない．

上記の問題を解決するために，本論文では，パーティションが発生する現実的なネットワークに適用可能な，可用性を最大化する票割当て手法の提案を行う．1)，2) への対応は，従来研究でのモデルを包摂する，より一般的なネットワークモデルを仮定することにより解決する．このネットワークモデルに基づき，可用性を最大化する票割当て問題を MAX-SMT 問題として定式化することで，3) の問題を解決する．MAX-SMT 問題とは，よく知られた組合せ最適化問題である MAX-SAT 問題（充足性最大化問題）を一般化した問題である [4]．MAX-SAT 問題とは，連言標準形のブール式と，各節に対する重みが与えられたときに，値が真となる節の重みの和を最大化する問題である．しかしながら，MAX-SAT 問題を用いて一般の整数値などを扱おうとすると，多数のブール変数で表さなくてはならないため，入力となるブール式が複雑になりがちである [5]．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ており，たとえば，整数変数上の線形算術による論理式を用いることができる．票割当て問題を MAX-SMT 問題に定式化することで，近年高速化が著しい MAX-SMT ソルバを利用して，最適な票割当てを求めることが可能となる．

提案手法の評価は実験を通じて行う．実験では，まず，現実的な通信ネットワークの特性を反映したモデルの実例を構築する．具体的には，現実的なネットワークトポロジを考慮したうえで，リンクとルータという構成要素の故障と修復をシミュレーションし，そこで得られた結果からモデルの実例を構築する．このモデルの実例に提案手法を適用することで，可用性を最大化する票割当てを得るために要した時間と，その票割当てによって達成される可用性を評価する．システムの動作中にネットワークやシステムの環境が変化した場合には，その環境に合わせて提案手法を用いて，最適な票割当てを再計算し，変更することで可用性の向上が可能である．しかしながら，システムを停止させず，票割当てを変更する機構については研究されているが [6], [7]，実用化の例はほとんどない．そのため，本研究では，長期間にわたって，ネットワークやシステムの特性が大きく変化しない環境を前提としている．

1.1 関連研究

文献 [8] では可用性を最大化する票割当てを求めるアルゴリズムを提案している．しかし，このアルゴリズムではネットワークのトポロジや通信リンクの故障を考慮しておらず，正常なノードどうしはつねに通信可能であると仮定している．

ネットワークトポロジを考慮したノードへの票割当ては，文献 [8], [9], [10], [11], [12], [13] で議論されている．これらの研究では，ネットワークトポロジは無向グラフとしてモデル化されており，グラフの頂点で表されるどのノードも，レプリカを保持するノードであるとともに，ルータでもあると想定されている．このようなモデルは，各ノードが互い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が，広域ネットワーク上に実現される分散システムのモデルとしては不適切である．また，これらの先行研究の中で，可用性最大の票割当てを示しているものとして文献 [11], [12], [13] があげられるが，ツリーやリングのような特定のネットワークトポロジを扱っており，一般のトポロジについては検討されていない．

文献 [14] では，整数線形計画法を用いてスループットを最大化する票割当てを行っている．この文献による手法は，ネットワークトポロジに依存しない点で本提案手法に類似しているが，目的関数が可用性でないこと，得られる結果の最適性を保証していない点などが，提案手法と異なる．

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 [15], [16], [17] で行われている．文献 [15] では，実システムから得られたデータを基に，文献 [8] のアルゴリズムによる票割当てと，各ノードに同じ票を割り当てる方法とを可用性の点で比較している．文献 [16], [17] では，非障害時における票割当てによる性能の最適化について検討されているが，可用性に関する議論はされていない．

なお，我々の研究の一部の成果は文献 [18], [19] で報告しており，本論文では実験結果を追加するとともに，内容の詳述化を行っている．

1.2 構成

本論文の構成は次のとおりである．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について述べる．実験では，提案手法が求解に必要とする計算時間と，得られた票割当てによる可用性の向上について評価する．最後に 5 章で本論文の結論を示す．

2. モデル

本章では，仮定する分散システムのモデルについて述べる．システムは，ネットワークに接続されている n 台のノードで構成されており，これらのノードの集合を $\Pi = \{1, 2, \ldots, n\}$ と表す．ノードは正常，もしくは故障の 2 つの状態をとる．任意の正常な 2 台のノードは，ネットワーク障害に依存して，他方のノードと連結しているか，連結していないかの 2 つの状態をとる．連結しているとは，接続するネットワークを介して他方のノードと通信が行えることを示す．ここで，互いに連結している正常なノードの極大集合をパーティションと呼ぶ．Π の空集合でない部分集合 g に対して，g がパーティションである確率を $P(g)$ で表す．確率 $P(g)$ は所与であるとする．

このモデルは単純であるが，票割当て問題に関する先行研究で想定されているネットワークモデルを一般化したモデルになっている．たとえば，ネットワーク分割が起こらずノードの故障のみを想定した場合，提案するモデルでは以下のように表現できる．

$$P(g) = \prod_{p \in g} r_p \prod_{p \notin g} (1 - r_p)$$

ただし，r_p はノード p の信頼度（正常である確率）である（他の例については，3.3 節で触れる）．

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．つまり，票割当て v は関数 $v : \Pi \to \mathbb{N}$ である．以降，ノード p に割り当てる票数を $v(p)$ と表す．データの一貫性を保持するために，システムの全票数のうち過半数の票数を連結しているノードから集めることのできたノードのみデータにアクセスできる．過半数の票数を持つパーティションは最大で 1 つしかないため，データの一貫性が保証される．多重化したデータの可用性を，データにアクセス可能な確率と定義する．

票割当て v による可用性は次のとおりである．

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

ここで，2^{Π} は Π の冪集合，$total = \sum_{p \in \Pi} v(p)$，$[P]$ は Iverson の記号で論理式 P が真ならば 1，偽ならば 0 をとる．$A(v)$ が上記の式で与えられる根拠は次のとおりである．$g, g' \in 2^{\Pi}$（$g \neq g'$）を過半数の票を有する（つまり，$\sum_{p \in g} v(p) > total/2$，$\sum_{p \in g'} v(p) > total/2$）任意の 2 つのノード集合とする．$g$ と g' が同時にパーティションとなることは不可能なので，それらがパーティションになるという事象は排反である．したがって，可用性 $A(v)$ は，過半数の票を有するノード集合がパーティションとなる確率の総和に等しい．

3. 票割当て問題の定式化

本章では，票割当て問題の定式化について述べる．まずは，定式化に用いる MAX-SMT 問題について説明し，その後に，可用性を最大化する票割当て問題を，MAX-SMT 問題として定式化する手法を説明する．

3.1 重み付き部分 MAX-SMT 問題

前章でモデル化した分散システムに対する可用性を最大化する票割当て問題を MAX-SMT 問題として定式化する．より正確には，重み付き部分 MAX-SMT 問題を考える．この問題の実例は (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．問題の解は，次の条件を満たす変数への値の割当て a である．

(i) すべ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，変数への値の割当てが a であるときの，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．具体的には，制約に以下のような式を用いる．

(i) 整数定数と変数，加減演算子（$+$，$-$），不等号（$=$，$<$，$\leq$，$>$，$\geq$）から構成される論理式．
(ii) ブール定数と変数から構成されるブール式．
(iii) (i) と (ii) を論理結合したもの．

重み付き部分 MAX-SMT 問題の例

簡単な重み付き部分 MAX-SMT 問題の例を示す．非負整数の変数 v_1，v_2 に対して，ハード制約を，

$$v_1 - v_2 \leq 3$$

とおく．ソフト制約 c_1，c_2，c_3 を，

$$c_1(v_1, v_2) = v_1 + v_2 \leq 10$$
$$c_2(v_1, v_2) = v_1 \geq 7$$
$$c_3(v_1, v_2) = v_2 \leq 4$$

とし，それぞれのソフト制約に対応する重みを，

$$w(c_1) = 1, \quad w(c_2) = 2, \quad w(c_3) = 3$$

と定める．重みの総和は $v_1 = 7$，$v_2 = 4$ のとき，最大値

$$\sum_{c \in \{c_1, c_2, c_3\}} (w(c) * [c(7, 4) = true]) = 5$$

をとる．したがって，$v_1 = 7$，$v_2 = 4$ が解となる．

3.2 問題の定式化

本節では，可用性 $A(v)$ 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．問題の定式化のために，次のような変数を定める．

- $p \in \Pi$ に対する，非負整数変数 v_p．p に割り当てる票数を表す．
- Π の空集合でない部分集合 g に対する，ブール型の変数 x_g．

上記の変数に対して，ハード制約を次のように定める．

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

上式により，変数 x_g は，システム全体の票数がノードの部分集合 g の有する票数の 2 倍より少ないとき真，そうでないとき偽の値をとることが表現されている．すなわち，g が過半数の票数を有することが，x_g が真となる必要十分条件であることを表す．

空集合でない部分集合 g に対して，ソフト制約 c_g を次のように定める．

$$x_g = true$$

ソフト制約 c_g に対する重み $w(c_g)$ を，g がパーティションを構成する確率 $P(g)$ と，整数値に正規化するための定数 $const$ を用いて，次のように定める．

$$w(c_g) = P(g) * const$$

このとき，可用性 $A(v)$ は，

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

と表される．よって，可用性 $A(v)$ の最大化問題は，上記のソフト制約による重みの総和を最大化する問題に帰着される．

3.3 定式化の例

定式化を行う例として，3 台のノードで構成されるシステムを考える[*1]．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とリンクのみ故障するものとする．ノードの信頼度は 0.95，リンクの信頼性は 0.99 であるとする．このネットワークは，以下のようにノードの部分集合 g に対して以下のように $P(g)$ を定めることで，2 章のモデルとして表現できる．

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

ハード制約の集合は，ブール変数 x_g が真となる必要十分条件を定める以下の式から構成される．

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

ソフト制約の集合とそれに関する重みは，$const = 10{,}000$ として正規化を行うと次のようになる．

$$\begin{aligned} &x_{\{1\}} = true && weight : 128 \\ &x_{\{2\}} = true && weight : 128 \\ &x_{\{3\}} = true && weight : 128 \\ &x_{\{1,2\}} = true && weight : 526 \\ &x_{\{1,3\}} = true && weight : 526 \\ &x_{\{2,3\}} = true && weight : 526 \\ &x_{\{1,2,3\}} = true && weight : 8{,}319 \end{aligned}$$

MAX-SMT ソルバを用いると，この問題に対する最適解の 1 つが得られる．たとえば，次のような変数割当てが最適解の 1 つとなる．

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

最適解において充足するソフト制約の重みの総和は 9,897 となる．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

定式化した MAX-SMT 問題に対して，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．

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

$$x_{\Pi} = true$$

[*1] ノード数 3 の場合，単一ノードのみに票を割り当てるか，もしくはすべてのノードに 1 票ずつ割り当てる場合が最適であることが知られている [20]．

すべての $p \in \Pi$ に対して $v_p = 0$ でなければ，このハード制約はつねに満たされる．すべての $p \in \Pi$ に対して $v_p = 0$ とする票割当ては最適解には決してならないので，上式の制約はつねに満たされるため，削除することが可能である．

2 つ目の最適化として，$P(g) = 0$ となるノードの部分集合 g についてのソフト制約とハード制約を取り除く．つまり，g がパーティションを構成する可能性がない場合，g に対応する制約を省略することが可能である．これは，ソフト制約による重みが 0 となるとき，x_g の真偽値が解に影響を与えないためである．

4. 評価

4.1 評価手順

提案手法がどの程度の規模のシステムに対して適用可能か，また，得られた可用性を最大化する票割当てによってどの程度可用性が向上するかを評価するため，実験を行った．**表 1** に実験に用いた計算機の仕様を示す．

実験では，まず，現実的なネットワークトポロジと，そのネットワーク上のどこにレプリカを維持するノードが配置されるかを定める．そのうえで，ネットワークの構成要素の故障と修復をシミュレートすることで，ノードの部分集合 g がパーティションとなる確率 $P(g)$ を求め，2 章で説明したネットワークモデルを構成する．なお，レプリカを保持するシステムのノードと，下部ネットワークのグラフの要素としてのノードを区別する必要がある場合，前者をサーバ，後者をルータと呼ぶ．このようにして得られたネットワークモデルに対し，提案手法を適用して MAX-SMT 問題の実例を生成し，既存の MAX-SMT ソルバである Yices 1.0 [21] を用いて求解する．

この実験の結果をもとに，4.3 節では，提案手法のスケーラビリティと可用性向上に関する効果を評価する．

4.2 シミュレーション

本節では，各パーティションの発生確率を求めるために行ったシミュレーション方法を説明する．**表 2** にシミュレーションの設定を示す．下部ネットワークのトポロジのモデルには Barabási-Albert モデル（BA モデル）を用いた [22]．このモデルを用いて，ルータ数 10 とルータ数 50 のトポロジを生成した．BA トポロジは，スケールフリー性とスモールワールド性の性質を持つ．スケールフリー性とは，多数のリンクに接続される少数のハブノードと少数のリンクに接続される多数のノードを有する性質である．スモールワールド性とは，ノード間の平均最短ホップ数が少ないという性質である．これらの性質はインターネットトポロジの持つ性質であり，そのため，BA トポロジはインターネットトポロジの主要なモデルとなっている．サーバは BA トポロジのノードに接続されるものとした．**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．

表 1 実験に用いた計算機の仕様
Table 1 Specification of the computer used for the experiment.

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

ネットワーク上で各パーティションが発生する確率を取得するために，各構成要素の故障と修復をシミュレーションした．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とを繰り返す．ルータ数の異なる 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 [23] 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．

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．$P(g)$ は，シミュレーション時間のうち，g がパーティションを構成している時間の占める割合で表される．簡単のために，図 1 と同様，サーバ数が 3 とすると，3.3 節で述べたようにパーティションは次の 7 つが発生しうる．

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

たとえば，パーティション $\{1\}$ がシミュレーションの

表 2 シミュレーションの設定
Table 2 Simulation setting.

シミュレーション時間	100,000 単位時間
ネットワークトポロジ	BA トポロジ
サーバ数	2, 3, . . . , 10
ルータ数	10, 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)

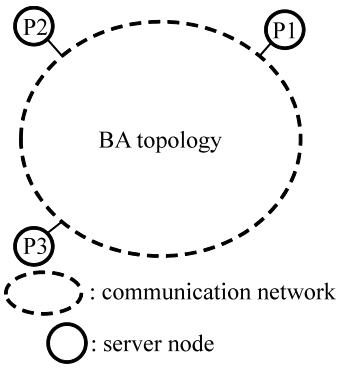

図 1 シミュレーションを行ったネットワークの概略
Fig. 1 A schematic view of the simulated network.

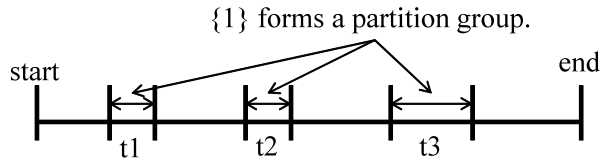

図 2 パーティション {1} が発生した区間
Fig. 2 Periods in which {1} forms a partition in a simulation run.

表 3 可用性を最大化する票割当ての計算に要した時間（ルータ数 10）
Table 3 Calculation time required to obtain the optimal vote assignment (10 router node network).

サーバ数	計算時間の平均 [s]	計算時間の最大値 [s]
3	0.0198	0.211
4	0.0177	0.0310
5	0.0212	0.0418
6	0.0364	0.0547
7	0.128	0.288
8	7.75	42.9
9	561.5	783.1
10	1,394.8	1,545.2
11	3,148.13	3,311.69
12	5,936.58	6,509.63
13	11,223.22	13,053.68
14	20,811.18	25,240.99
15	45,519.48	53,258.3

期間で**図 2** のように生じた場合，パーティション {1} の発生確率 $P(\{1\})$ は，シミュレーション時間を T とすると，$(t1+t2+t3)/T$ で表される．その他の 6 つのパーティションの発生確率も同様に求められる．また，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．

4.3 結果

シミュレーションで得られた各パーティションの発生確率を仮定して提案手法を適用し，最適な票割当てを求めた．求解に必要な時間を評価するため，故障，修復のシミュレーションと提案手法の適用を，ルータ数 10，50 の各トポロジに対して行った．故障，修復のシミュレーションと提案手法の適用は，サーバ数 3〜10 の場合は 100 回，計算時間が長いサーバ数 11〜15 の場合は 3 回行った．このときのルータ数 10 の場合における，MAX-SMT ソルバの実行時間の平均と最大値を**表 3** に示す．表 3 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分かる．

定式化された MAX-SMT 問題に含まれる制約式の個数の平均と最大値を**表 4** に示す．ここで，制約式の個数とは最適化後の制約式の個数である．すなわち，構成される確率が 0 でないパーティションの個数の 2 倍から 1 を減じた値となる．この数は，すべてのサーバの部分集合がパーティションとなりうる場合に最大値 $2*(2^n-2)$ となる．表 4 の結果より，実験で解いた MAX-SMT 問題の大きさは，サーバ数 10 以下のとき，ほぼこの上限値に近い値をとっている．したがって，異なるトポロジや故障確率を有する下部ネットワークを仮定したとしても，サーバ数（レプリカ数）が 10 以下であれば，実用的な時間で最適な票割当てを求めることができることが分かる．ルータが 50 台の結果については，$n \leq 10$ の場合の計算時間は 30 分以内に抑えられており，制約式の数もルータ数 10 の場合とほぼ等しかったため，ここでは省略している．

表 4 制約式の個数（ルータ数 10）
Table 4 Number of constraints (10 router nodes netowrk).

サーバ数	制約式の数の平均	制約式の数の最大値
3	12	12
4	28	28
5	60	60
6	128	128
7	252	252
8	507.7	508
9	1,012.76	1,020
10	1,972.6	2,022
11	3,733.33	3,810
12	6,714.67	6,854
13	11,276	11,646
14	17,805.33	18,866
15	26,460.67	28,840

現実の応用においては，厳密な一貫性を有する多重化データを管理する場合，レプリカを保持するサーバの台数は多くても 10 程度と考えられる．たとえば，マイクロソフトの Niobe large-scale enterprise storage system では，高信頼性と厳密な一貫性が求められるシステムの大域的状態の管理を，10 台以下のマシンから構成される大域的状態管理（GSM）モジュールを用いて行っている [24]．また，Apache ZooKeeper と同様な目的で使用されるよく知られたアルゴリズムとして Paxos があるが [25]，Google 社の Chubby や Spanner といった実システムでの Paxos の実装の規模も，通常数ノード程度である [26], [27]．したがって，本提案手法は現実規模の問題を扱うのに十分なスケーラビリティを有しているといえる．また，多くの Paxos の実装では過半数のノードグループに対しデータアクセスを行うことでデータの一貫性を保っているが，この仕組みは多数決投票により拡張することができる [25]．このように拡張された Paxos の実装に対しては，提案手法を適用することで可用性の最適化が可能となる．

次に，可用性を最大化する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．ルータ数 10 のときの可用性を**図 3**，**図 5**，ルータ数 50 のときの可用性の平均を**図 4**，**図 6** に示す．横軸はデータを多重化したサーバ数，

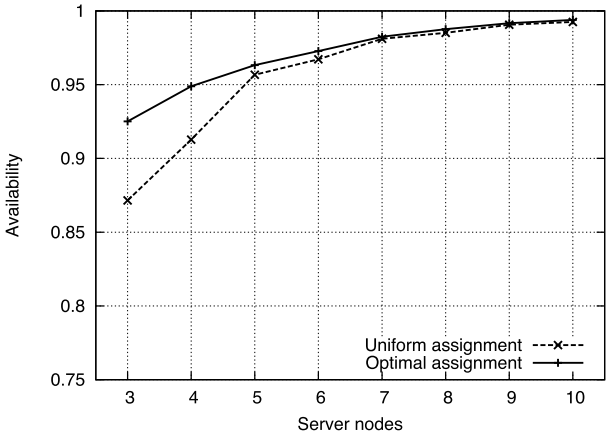

図 3 可用性（ルータ数 10）

Fig. 3 Availability (10 router node network).

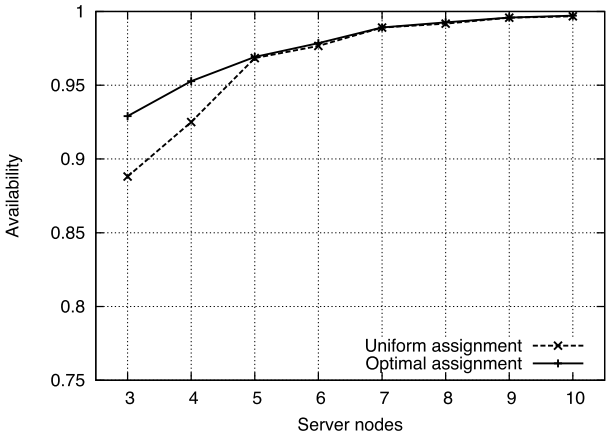

図 4 可用性（ルータ数 50）

Fig. 4 Availability (50 router node network).

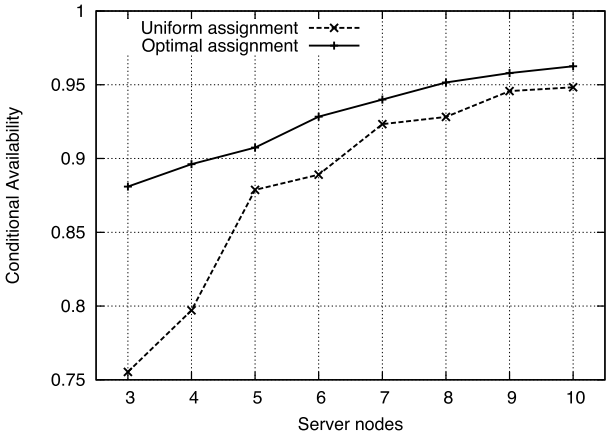

図 5 サーバの故障，もしくは複数のパーティションが発生した場合での条件付き可用性（ルータ数 10）

Fig. 5 Conditional availability given that some server node fails or multiple partitions occur (10 router node network).

縦軸は上にいくほど可用性が高いことを示している．ここでは，均等な票割当て（Uniform assignment）と可用性を最大化する票割当て（Optimal assignment）を比較している．なお，均等な票割当てでは，サーバ数が偶数の場合は

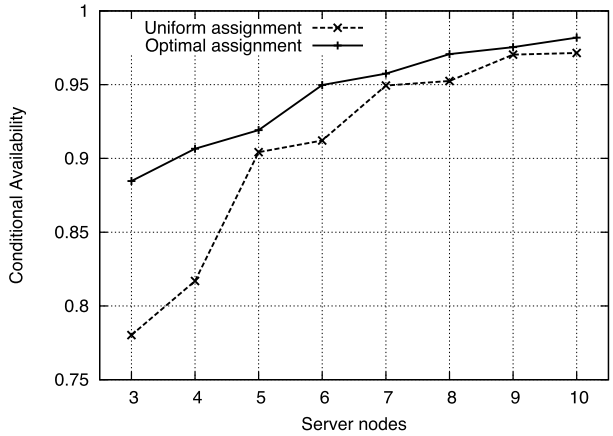

図 6 サーバの故障，もしくは複数のパーティションが発生した場合での条件付き可用性（ルータ数 50）

Fig. 6 Conditional availability given that some server node fails or multiple partitions occur (50 router node network).

ランダムに選択したサーバに 1 票を追加で割り当てるという最適化を適用している．このことで，完全に票が同数である場合に比べて，可用性が必ず向上することが知られている [20]．

図 3，図 4 は，シミュレーション時間全体のうち，システムが稼働している時間の占める割合である可用性 $A(v)$ を表している．一方，図 5，図 6 では，可用性そのものではなく，サーバ単体の故障も含め，サーバ間の連結に何らかの影響を与える故障が発生しているという事象の下での条件付きの可用性を表している．票割当て v に対し，この値は以下の式によって求められる．

$$\frac{A(v) - P(\Pi)}{1 - P(\Pi)}$$

ここで，$P(\Pi)$ は全サーバからなるノード集合がパーティションであること，すなわち，それらすべてが正常であり，かつ，互いに連結している確率である．この条件付きの可用性を考えることで，故障への耐性の度合いをより明確に比較することが可能となる．

図 3，図 4，図 5，図 6 から，サーバ数がいずれの場合にも，提案手法によって可用性が改善されていることが分かる．特に，サーバ数が 5 未満のときに大きな改善が見られる．

5. おわりに

本論文では，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，その解を求めることで最適な票割当てを求めた．実験の結果，システムのノード数が 10 以下であれば提案手法によって実用的な時間で最適な票割当てを求められる

ことが分かった．このノード数は実際の応用において十分な規模といえる．さらに，得られた票割当てによって，均等な票割当てと比べて故障発生時の可用性を改善できること，また，その効果はノード数が少ないときに顕著であることを示した．

今後の課題としては，遅延やスループットなどの可用性以外の評価尺度に関する最適化があげられる．また，ネットワークやシステムの環境の変化に対して，システムの動作中に票割当ての変更を行うことを考える．システムの動作中にネットワークやシステムの環境が変化したとき，票割当ての変更をいつ行うか，また，その票割当てに必要なパーティションの発生確率をどのように取得するかについて検討する必要がある．

謝辞 本研究にあたって，先行研究を行ってくださった橋下洋氏，および有益なコメントをくださった査読者の方々に深く感謝いたします．

参考文献

[1] Thomas, R.H.: A Majority consensus approach to concurrency control for multiple copy databases, *ACM Trans. Database Syst.*, Vol.4, No.2, pp.180–209 (online), DOI: 10.1145/320071.320076 (1979).

[2] Gifford, D.K.: Weighted voting for replicated data, *Proc. 7th ACM Symposium on Operating Systems Principles*, *SOSP '79*, pp.150–162 (online), DOI: 10.1145/800215.806583 (1979).

[3] Hunt, P., Konar, M., Junqueira, F.P. and Reed, B.: ZooKeeper: Wait-free coordination for internet-scale systems, *Proc. 2010 USENIX Annual Technical Conference* (2010).

[4] Nieuwenhuis, R. and Oliveras, A.: On SAT Modulo Theories and Optimization Problems, *9th International Conference on Theory and Applications of Satisfiability Testing* (*SAT2006*), pp.156–169 (2006).

[5] 岩沼宏治，鍋島英知：SMT：個別理論を取り扱う SAT 技術，人工知能学会誌，Vol.25, No.1, pp.86–95 (2010).

[6] Jajodia, S. and Mutchler, D.: Dynamic voting algorithms for maintaining the consistency of a replicated database, *ACM Trans. Database Syst.*, Vol.15, No.2, pp.230–280 (1990).

[7] Bearden, M. and Bianchini, R.P., Jr.: A fault-tolerant algorithm for decentralized on-line quorum adaptation, *Proc. 28th International Symposium on Fault-Tolerant Computing* (*FTCS 98*), pp.262–271 (1998).

[8] Tong, Z. and Kain, R.: Vote assignments in weighted voting mechanisms, *IEEE Trans. Comput.*, Vol.40, No.5, pp.664–667 (online), DOI: 10.1109/12.88491 (1991).

[9] Barbara, D. and Garcia-Molina, H.: The vulnerability of vote assignments, *ACM Trans. Comput. Syst.*, Vol.4, No.3, pp.187–213 (online), DOI: 10.1145/6420.6421 (1986).

[10] Barbara, D. and Garcia-Molina, H.: The Reliability of Voting Mechanisms, *IEEE Trans. Comput.*, Vol.C-36, No.10, pp.1197–1208 (online), DOI: 10.1109/TC.1987.1676860 (1987).

[11] Diks, K., Kranakis, E., Krizanc, D., Mans, B. and Pelc, A.: Optimal coteries and voting schemes, *Information Processing Letters*, Vol.51, No.1, pp.1–6 (online), DOI: 10.1016/0020-0190(94)00064-6 (1994).

[12] Ibaraki, T., Nagamochi, H. and Kameda, T.: Optimal coteries for rings and related networks, *Distributed Computing*, Vol.8, pp.191–201 (1995).

[13] Papadimitriou, C.H. and Sideri, M.: Optimal coteries, *Proc. 10th Annual ACM Symposium on Principles of Distributed Computing*, *PODC '91*, pp.75–80 (online), DOI: 10.1145/112600.112608 (1991).

[14] Venkaiah, D. and Jalote, P.: An integer programming approach for assigning votes in a distributed system, *Proc. 14th Symposium on Reliable Distributed Systems* (*SRDS 1995*), pp.128–134 (online), DOI: 10.1109/RELDIS.1995.526220 (1995).

[15] Amir, Y. and Wool, A.: Evaluating quorum systems over the Internet, *Proc. 26th Int'l Symp. on Fault-Tolerant Computing* (*FTCS-26*), pp.26–35 (online), DOI: 10.1109/FTCS.1996.534591 (1996).

[16] Gupta, A., Maggs, B.M., Oprea, F. and Reiter, M.K.: Quorum placement in networks to minimize access delays, *Proc. 24th ACM Symp. on Principles of Distributed Computing* (*PODC '05*), pp.87–96 (2005).

[17] Oprea, F. and Reiter, M.: Minimizing Response Time for Quorum-System Protocols over Wide-Area Networks, *Proc. 37th Int'l Conf. on Dependable Systems and Network* (*DSN 2007*), pp.409–418 (online), DOI: 10.1109/DSN.2007.66 (2007).

[18] 松井佑記，小島英春，土屋達弘：Voting を用いた分散システムの可用性の最大化，平成 24 年度情報処理学会関西支部支部大会講演論文集 (2012).

[19] Matsui, Y., Kojima, H. and Tsuchiya, T.: Maximizing Availability of Consistent Data in Unreliable Networks, *IEEE 18th International Conference on Parallel and Distributed Systems* (*ICPADS 2012*), pp.117–123 (2012).

[20] Garcia-Molina, H. and Barbara, D.: How to assign votes in a distributed system, *J. ACM*, Vol.32, No.4, pp.841–860 (online), DOI: 10.1145/4221.4223 (1985).

[21] Dutertre, B. and de Moura, L.M.: A Fast Linear-Arithmetic Solver for DPLL(T), *Proc. 18th Conf. on Computer Aided Verification* (*CAV 2006*), LNCS, Vol.4144, pp.81–94 (2006).

[22] Barabási, A.-L. and Albert, R.: Emergence of Scaling in Random Networks, *Science*, Vol.286, pp.509–512 (1999).

[23] Kuusela, P. and Norros, I.: On/off process modeling of IP network failures, *Proc. 40th Int'l Conf. on Dependable Systems and Network* (*DSN 2010*), pp.585–594 (2010).

[24] Maccormick, J., Thekkath, C.A., Jager, M., Roomp, K., Zhou, L. and Peterson, R.: Niobe: A practical replication protocol, *ACM Trans. Storage*, Vol.3, No.4, pp.1:1–1:43 (online), DOI: 10.1145/1326542.1326543 (2008).

[25] Lamport, L.: The part-time parliament, *ACM Trans. Comput. Syst.* (*TOCS*), Vol.16, No.2, pp.133–169 (1998).

[26] Chandra, T., Griesemer, R. and Redstone, J.: Paxos made live-an engineering perspective (2006 invited talk), *Proc. 26th ACM Symposium on Principles of Distributed Computing* (*PODC*) (2007).

[27] Corbett, J.C., Dean, J., Epstein, M., Fikes, A., Frost, C., Furman, J., Ghemawat, S., Gubarev, A., Heiser, C., Hochschild, P., Hsieh, W., Kanthak, S., Kogan, E., Li, H., Lloyd, A., Melnik, S., Mwaura, D., Nagle, D., Quinlan, S., Rao, R., Rolig, L., Saito, Y., Szymaniak, M., Taylor, C., Wang, R. and Woodford, D.: Spanner: Google's globally-distributed database, *Proc. 10th*

USENIX Symposium on Operating Systems Design and Implementation (*OSDI*) (2012).

推薦文

関西支部では，推薦論文の検討対象として支部大会を利用することとした．そこで支部大会で口頭発表された論文（74 件）を対象とし，各セッションの座長，実行委員から広く推薦を集めて候補を 7 件に絞り，各論文に対し事後評価者 2 名の評価を加え，実行委員会による審議を経て 2 件の推薦論文候補を決定した．本論文は，多数決投票を用いた分散システムの可用性に関する研究であり，問題定式化のアプローチおよび評価結果は高く評価できるものであり，推薦論文にふさわしいと判断した．

（関西支部長　黒橋禎夫）

松井 佑記

2012 年 3 月大阪大学基礎工学部情報科学科卒業．2012 年大阪大学大学院情報科学研究科入学．現在，博士前期課程在籍．

小島 英春

2001 年広島市立大学大学院情報科学研究科修士課程修了，2009 年広島市立大学大学院情報科学研究科博士後期課程修了（情報工学）．広島市立大学院情報科学研究科特任助教を経て，2012 年より大阪大学大学院情報科学研究科助教．ネットワークソフトウェア，ネットワークプロトコル，ソフトウェアテストの研究に従事．

土屋 達弘　（正会員）

1995 年 3 月大阪大学大学院基礎工学研究科博士前期課程修了．博士（工学）．同研究科助手を経て，現在大阪大学大学院情報科学研究科教授．ソフトウェアを中心とする情報システムのテスト，検証，高信頼化に関する研究に従事．