

誤り発生確率が変化するボランティアコンピューティング における信頼度に基づく多数決法の評価と改善

渡邊 寛^{1,a)} 福土 将² 船曳 信生¹ 中西 透¹

概要：本稿では、誤った計算結果の発生確率（妨害率）が時間的に変化するボランティアコンピューティング（VC）環境を対象として、信頼度に基づく多数決法の計算信頼性保証能力の評価と改善を行う。同手法は、各参加者が正しい結果を返す確率（信頼度）を計算し、多数決で誤りが採択されてしまう確率を常に指定値以下に抑えることで信頼性保証を実現する。しかし、同手法では、妨害率は時間的に一定な定数値として仮定されているため、実環境のように、妨害率が変動する環境に適用した場合の有効性は十分に検証されていない。そこで本稿では、計算の信頼性を最も大きく低下させると思われる、妨害率がステップ関数に従って変化するモデルを用いて、同手法の信頼性保証能力を評価する。さらに、本モデルを用いた場合の計算信頼性の数学的な解析を行い、パラメータ制限を加える改善法を提案する。この改善により、妨害率が時間的にどのように変化する場合でも信頼性保証が実現できることを示す。

キーワード：妨害者対策，計算信頼性，並列分散処理，数学モデル化，デスクトップグリッド

Evaluation and Improvement of Credibility-based Voting for Volunteer Computing Systems with Dynamic Error Rate

KAN WATANABE^{1,a)} MASARU FUKUSHI² NOBUO FUNABIKI¹ TORU NAKANISHI¹

Abstract: To ensure the reliability of computations in Volunteer Computing (VC) systems, *credibility-based voting method* is proposed. This method has the capability of guaranteeing the reliability condition that the error rate is less than the specified acceptable value. The capability has been demonstrated under the model that the probability of returning incorrect results s is a constant. However, in actual VC systems, the probability should be changed with time for several reasons, e.g. hardware/software failures or age-related degradations. In this paper, we evaluate the capability of credibility-based voting through simulations with step-functions type model of s . We also propose an improvement of credibility-based voting by a simple parameter tuning to ensure the reliability condition for any model of s .

Keywords: Sabotage-tolerance, Reliability, Parallel Computing, Mathematical Modeling, Desktop Grids

1. まえがき

ボランティアコンピューティング（VC）は、インターネットに接続されたパーソナルコンピュータ（PC）やスマートフォン・家庭用ゲーム機などを持つ一般の参加者から、遊休計算資源（アイドル CPU サイクル・空きストレージな

ど）を無償で提供してもらうことで、安価で高性能な計算プラットフォームを実現するものである。近年の個人用・家庭用電子機器の高性能化に伴い、有り余る遊休計算資源を有効活用する方法として、VC は、PC を対象とする SETI@home[1] などの古典的なものから、GPU[2] や PlayStation3[3]、Android ベースの携帯やタブレット [4] を対象とする異機種混在の大規模で複雑なプラットフォームへと発展している。

このような大規模な VC における問題点として、参加者

¹ 岡山大学 大学院自然科学研究科

² 山口大学 大学院理工学研究科

^{a)} can@okayama-u.ac.jp

から提供された計算資源（ワーカ）が、与えられた計算問題（ジョブ）に対して誤った計算結果を返すことがある、という問題が指摘されている [5], [6], [7]。計算誤りが発生する原因としては、オーバークロックやハードウェア障害に加え、参加の褒章を当てるとする悪意など様々なものが考えられる。発生した誤りは、その原因に寄らず VC システムの計算信頼性を低下させるため、誤りを発生させるワーカを妨害者と呼ぶ。VC において妨害者の参入は不可避であり、またその数も参加者数に比例して増加するため、管理者側で計算信頼性を向上させる工夫を行う必要がある。

そこで、同じジョブを複数のワーカに計算させて多数決を行うことで信頼性を向上させる、信頼度に基づく多数決法 [6] が提案されている。この手法では、各ワーカに対して、抜取検査と呼ばれる正答が既知のジョブを与え、その結果に基づいて、そのワーカが正しい計算結果を返す確率（信頼度）を求める。信頼度を多数決における重みとすることで、信頼性の高いワーカの計算結果が採択されやすくなり、多数決結果に含まれる誤りの割合（誤り率 ϵ ）を効率的に低減することができる。また、信頼度は未知の誤り発生確率 s に対する下限値として計算されるため、この手法は、任意の s に対して誤り率 ϵ を常に所望値（許容誤り率： ϵ_{acc} ）以下とする信頼性保証を実現できる、という特長がある（ $\forall s \epsilon(s) \leq \epsilon_{acc}$ ）。

この手法の問題点は、誤り発生確率 s が時間的に一定と仮定して信頼度を計算している点にある。例えば、機器の経年劣化による誤り発生の場合、バスタブモデル等に代表されるように s が時間的に変動するのが自然である。このため、従来の信頼度計算式を実際の VC でそのまま適用した場合、正しい信頼度が計算できず、「常に誤り率 $\epsilon \leq$ 許容誤り率 ϵ_{acc} 」という信頼性条件を満足できなくなる可能性がある。

そこで本稿では、まず、誤り発生確率 s が時間的に変化するモデルにおいて、信頼度に基づく多数決法の信頼性保証能力を評価する。誤り率 ϵ の期待値を最大化するモデルとして、 s をステップ関数状に変化させるモデルが適していることを示し、同モデルを構成するパラメータ s_1, T_s, s_2 を様々に変化させてシミュレーションを行う。このモデルでは、悪意ある参加者群が、各自の信頼度が閾値以下では小さな s を用いることで抜取検査を通過し、信頼度が閾値を超えた段階でより大きな s を用いることで攻撃を開始するといった、誤り率をより大きくする挙動を想定している。

次に、信頼度に基づく多数決法の拡張として、VC 管理者が任意に設定可能なパラメータについて条件を課す改善法を提案する。提案手法により、どのような s_1, T_s, s_2 を用いたステップ関数モデルに対しても、信頼性条件 $\epsilon \leq \epsilon_{acc}$ を満足できることを示す。さらに、 s がステップ関数以外の、一般的な変化をする場合についての検討を行う。誤り率 ϵ の期待値は、 s がステップ関数状に変化する場合に最

大化されると予想される（厳密な証明は今後の課題である）ため、この予想の証明により、ステップ関数を含む任意の s の関数について、信頼性保証が実現できることになる。

本稿の構成は以下の通りである。まず、2 章で想定する VC のモデルを述べ、3 章で信頼度に基づく多数決法について述べる。次に、4 章で ϵ の期待値を最大化する s の関数モデルとパラメータの設定方法を示す。5 章では、ステップ関数状に変化する s を用いたシミュレーションによる評価を行い、最後に 6 章で結論と今後の課題を述べる。

2. VC モデル

2.1 計算モデル

VC の計算モデルとして、図 1 に示す、マスタワーカモデル [8] を仮定する。本モデルの詳細は以下の通りである。

- 計算プロジェクトは、 N 個の独立したジョブから成る。プロジェクトは、全てのジョブが完了するか、時刻が期限 T_D に達したら終了となる。
- マスタ 1 台と、ボランティアにより提供されるワーカ W 台で、計算プロジェクトを実行する。
- 各ワーカは、アイドル状態になるとマスタにジョブを 1 つ要求し、ジョブを実行後、その計算結果（リザルト）を返却する。

2.2 妨害者モデル

VC における参加者は、PC の故障や故意の攻撃等を原因として、誤ったりザルトをマスタに返却することがある。このような誤ったりザルトを返す参加者は妨害者と呼ばれ、常に正しいリザルトを返すような非妨害者と区別して扱われる。

妨害者の存在は、次のようにモデル化される [6]。まず、妨害者の割合を f とし、全ワーカ W 台のうちの $[f \times W]$ 台を妨害者とする（ここで、 $[\]$ はフロア関数）。各妨害者は、誤ったりザルトを確率 s （誤り発生確率・妨害率）で返却する。 f と s はマスタにとって未知の値である。

妨害者の存在を前提とした VC では、マスタによりリザルトの検査等が行われる場合がある。このため各ジョブは、ワーカからリザルトが回収され、マスタによる多数決等の検査を経て最終的なリザルトが確定した時点で、ジョ

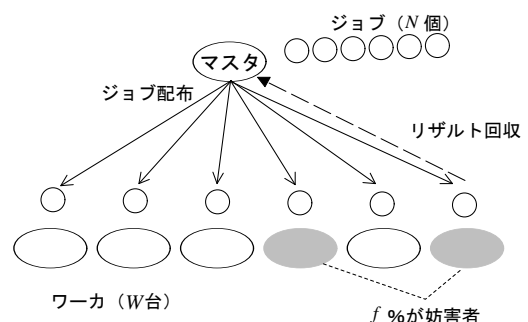


図 1 VC の計算モデル

ブの完了となる．このようなジョブを完了ジョブと呼ぶ．全てのジョブが期限 T_D までに完了ジョブとなった場合、計算開始から完了までに要した時間を計算時間 T とする ($T \leq T_D$)．また、完了したジョブのうち、誤ったりザルトにより完了したジョブを誤りジョブと呼び、誤りジョブが全体に占める割合を誤り率 ϵ と定義する．

2.3 信頼性指標

本稿では、VC の計算信頼性の指標として、誤り率 ϵ を用いる．1つのジョブに対してリザルトを1つ生成するだけでは、リザルトの誤りが直接ジョブの誤りとなり、誤り率 ϵ が著しく増大してしまう．例えば、各ワーカの性能やジョブの計算量が均一の場合、誤り率は $\epsilon = \frac{N \times f \times s}{N} = f \times s$ となる．よって、1つのジョブに対して複数のリザルトを集めたり、妨害者を検出し排除するなどして誤り率 ϵ を小さくする、信頼性向上手法が不可欠となる．

基本的な信頼性向上手法として、VC ミドルウェア BOINC[8] で用いられている単純多数決法がある．これは、同じジョブを複数のワーカに計算させて冗長にリザルトを集め、単純な多数決を行う手法である．この手法では、経験則等に基づいて手動で冗長度を決定するため、信頼性向上の度合いが予測できない．このため、誤り率 ϵ が許容値 ϵ_{acc} を超えてしまい、計算結果が信用できないといった問題が生じる．誤り率の許容値の例としては、例えば3次元映像のレンダリングでは誤り率 1% [6] などがある．

3. 信頼度に基づく多数決法

3.1 概要

信頼度に基づく多数決法 [6] は、効率的な信頼性向上を図るために、信頼度と呼ばれるパラメータを導入し、信頼度を重みとした重み付き多数決を行う手法である．信頼度は、ワーカやリザルトなどの VC 環境内の各要素に対して与えられる、その要素の正しさを表す条件付確率である．信頼度を重みとすることで、信頼性の高いワーカのリザルトが多数決で採択されやすくなるため、少ないリザルト数で誤り率 ϵ を効率的に低減することができる．

また、ワーカの信頼度は、各ワーカに対して一定確率 (検査率 q) で抜取検査を行った結果に基づいて、そのワーカが正しいリザルトを返す確率の下限値として求める．ここで、確率の下限として信頼度を計算する理由は、実際に誤りである確率が未知パラメータである s や f に依存するためである．同様に、ジョブの信頼度も、そのジョブにおける多数派のリザルトが正しい確率の下限として与えられる．信頼度が閾値 $\theta = 1 - \epsilon_{acc}$ 以上のジョブのみを受け入れることで、任意の許容誤り率 ϵ_{acc} について、 s や f によらず常に $\epsilon \leq \epsilon_{acc}$ とすることができる．

3.2 使用するパラメータと仮定

信頼度に基づく多数決法 [6] では、信頼度を計算する際に、以下の仮定をおいている．

- 全ワーカの性能は均一で、1つのジョブを計算するのに1単位時間 (ターン) を要する．
- 実際の妨害者割合 f は未知数であるため、その上限として f_{max} を仮定する ($f \leq f_{max}$) ．
- 実際の妨害率 s は未知数であるが、 s は時間的に一定値であり、全妨害者で共通の値であると仮定する．
- 抜取検査により妨害者として検出された時点で、そのワーカが以前に生成したりザルトは全て無効化される．抜取検査により妨害者として検出された後のワーカの振る舞いとして、そのワーカがマスタのブラックリストに登録され、以後の計算に参加できなくなる場合 (ブラックリスト有効の場合) と、検出された時点で新しいワーカとして ID 等を再度取得し、計算に参加し続ける場合 (ブラックリスト無効の場合) がある．本稿では、ブラックリスト有効の場合について考える．

3.3 信頼度の計算式

3.2 節に示した仮定の下で、ワーカ w の信頼度 $C_W(w)_{bl}$ は、検査通過回数 k を用いて、式 (1) で求められる [6] ．

$$C_W(w)_{bl} = \begin{cases} 1 - f_{max}, & \text{if } k = 0, \\ 1 - \frac{f_{max}}{(1 - f_{max}) \times k e}, & \text{otherwise.} \end{cases} \quad (1)$$

ただし、 e は自然対数の底である．

リザルト r の信頼度 $C_R(r)$ は、そのリザルトを生成したワーカ w の信頼度 $C_W(w)_{bl}$ と等しいものとする．

$$C_R(r) = C_W(w)_{bl} \quad (2)$$

ジョブ j に対するリザルト群は、値の等しいもの同士で、リザルトグループと呼ばれるグループに分けられる．ジョブ j のリザルトが g 個のグループ $G_1, G_2, \dots, G_a, \dots, G_g$ に分けられたとすると、 G_a の信頼度 $C_G(G_a)$ は、式 (3)-(5) で計算される． $C_G(G_a)$ は、 G_a に含まれるリザルトのみが正しく、他の全てのリザルトグループが誤りである確率となる．

$$C_G(G_a) = \frac{P_T(G_a) \prod_{i \neq a} P_F(G_i)}{\prod_{i=1}^g P_F(G_i) + \sum_{n=1}^g P_T(G_n) \prod_{i \neq n} P_F(G_i)} \quad (3)$$

$$P_T(G_a) = \prod_{r \in G_a} C_R(r) \quad (4)$$

$$P_F(G_a) = \prod_{r \in G_a} (1 - C_R(r)) \quad (5)$$

ジョブ j の信頼度 $C_J(j)$ は、 G_1, \dots, G_g の中で最大の信頼度を持つグループ G_x の信頼度として定義される．

$$C_J(j) = C_G(G_x) = \max_{1 \leq a \leq g} C_G(G_a) \quad (6)$$

ジョブの信頼度が閾値 $\theta = 1 - \epsilon_{acc}$ を超えた場合、マスタは G_x のリザルトをジョブ j のリザルトとして受け入れ、ジョブ j を完了とする。

4. 妨害率が時間的に変化する VC モデル

4.1 モデルの拡張と考察の手順

3.2 節で述べたように、信頼度に基づく多数決法は、信頼度の計算過程において妨害率 s が一定値であることを仮定している。しかし実際の VC では、経年劣化等により計算誤りの頻度が時間的に増加するなど、 s の値が変化することは自然に起き得る。本節では、このように妨害率 s が時間的に変化する時、どのような変化をした場合に誤り率が最大化されるかを明らかにするため、以下の順番で従来の VC モデルを拡張し考察を行う。

4.2 節では、 s が時間的に一定となる従来の VC モデルから、 s が計算中に 1 度だけ変化するようなステップ関数型のモデルに拡張する。信頼性条件 $\epsilon \leq \epsilon_{acc}$ を保証するためには、誤り率 ϵ を最大化するような状況を明らかにする必要があるため、まず、このステップ関数型モデルにおいて、単純に誤りリザルトの“数”を最大化するような s の変化の仕方を求める。次に 4.3 節では、信頼度の大きさで誤りリザルトが受け入れられるかどうか異なる点に着目し、受け入れられる誤りリザルトの数、すなわち誤り率を最大化するような s の変化の仕方を求める。次に 4.4 節では、誤り率の上限を計算し、上限を ϵ_{acc} 以下にするようなパラメータ設定法の提案により、信頼性条件が保証可能なことを示す。最後に、4.5 節にて、 s がステップ関数型以外の変化をする場合について考察する。

4.2 ステップ関数モデルの解析

まず、妨害率 s のモデルとして、図 2 に示すような、計算開始から完了までの間に、一度だけ s を変化させるステップ関数型モデルを考える。このモデルは以下の 3 つのパラメータで記述される。

- 時刻 T_s
 計算開始時点から、時刻 T_s までの間を前期、それ以降を後期とする。ただし $T_s \leq T \leq T_D$ である。
- 前期妨害率 s_1
 T_s までの間、各ジョブに誤りを返す確率。
- 後期妨害率 s_2
 T_s 以降、各ジョブに誤りを返す確率。

各期間（前期・後期）において、この妨害者が抜取検査を通過し続け生き残る確率と、その間に確率 s で生成する誤りリザルト数の期待値は以下のようにまとめられる。

- T_s まで生き残る確率 $P_1 = (1 - qs_1)^{T_s}$
- T_s 以降 T 又は T_D まで生き残る確率 P_2
 $P_2 = (1 - qs_2)^{\min(T, T_D) - T_s}$
- T_s までに生成する誤りリザルト数 $Rn_1 = s_1 T_s$

- T_s 以降に生成する誤りリザルト数 Rn_2

$$Rn_2 = s_2(\min(T, T_D) - T_s)$$

次に、ステップ関数型モデルにおいて、計算終了時 (T 又は T_D) において有効な誤りリザルト数の期待値 E_n と、 E_n を最大化するような T_s, s_1, s_2 の組み合わせを求める。

任意の T_s, s_1, s_2 の組み合わせに対し、 $E_n(s_1, s_2, T_s)$ は、次のように表される。

$$\begin{aligned} E_n(s_1, s_2, T_s) &= P_1 \times P_2 \times (Rn_1 + Rn_2) \\ &= (1 - qs_1)^{T_s} (1 - qs_2)^{\min(T, T_D) - T_s} \\ &\quad \times (s_1 T_s + s_2(\min(T, T_D) - T_s)) \quad (7) \end{aligned}$$

ここで、 s_1 と s_2 は独立しているため、それぞれの変数について偏微分を行い、極値を求める。

$$\frac{\partial}{\partial s_1} E_n(s_1 = \hat{s}_1, s_2, T_s) = 0 \quad (8)$$

$$\hat{s}_1 = \min\left(1, \frac{1 - qs_2(\min(T, T_D) - T_s)}{q(1 + T_s)}\right) \quad (9)$$

$$\frac{\partial}{\partial s_2} E_n(s_1, s_2 = \hat{s}_2, T_s) = 0 \quad (10)$$

$$\hat{s}_2 = \min\left(1, \frac{1 - qs_1 T_s}{q(1 + \min(T, T_D) - T_s)}\right) \quad (11)$$

式 (9) と式 (11) を連立方程式として解くと、式 (12) を得る。

$$\hat{s}_1 = \hat{s}_2 = \min\left(1, \frac{1}{q(1 + \min(T, T_D))}\right) \quad (12)$$

式 (12) は、誤りリザルト数の期待値 E_n が、 $s_1 = s_2$ の時に極値（最大値）を取り、途中で s を変化させる ($s_1 \neq s_2$ となる) ことは、 E_n の減少につながることを意味している。誤りリザルトの数を最大化する場合は、計算開始から終了まで常に式 (12) で得られる s を用いればよいことが示された。

4.3 誤り率の最大化

信頼度に基づく多数決法を用いた場合、生成された誤りリザルトは、マスタにより採択されて初めて、誤り率を増加させる。よって、誤り率を最大化するためには、単純に誤りリザルト数の期待値 E_n を最大化するのではなく、受け入れられる誤りリザルト数の期待値 $E'_n (\leq E_n)$ を最大化する必要がある。

ここで、信頼度に基づく多数決法の性質として、ある妨

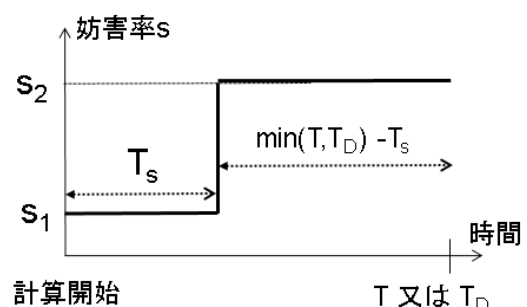


図 2 ステップ関数型の妨害率モデル

害者の信頼度が閾値 $1 - \epsilon_{acc}$ を超えた場合、リザルト1つでジョブを完了させることができるので、誤りリザルトが確実に受け入れられることに注意する。また、信頼度が閾値に達していない場合は、完了に他のリザルトを必要とするが、各妨害者は個別に誤りを生成しているため、誤りリザルト同士の値が偶然一致して採択される確率は十分小さく無視できる。

このような性質を考慮すると、受け入れられる誤りリザルト数の期待値 E'_n とその上限は、次のように計算できる。まず、信頼度が閾値以下の時（前期）には、誤りリザルトを生成しても受け入れられないので $s_1 = 0$ とし、閾値を越えた後（後期）に $s = s_2$ を用いる。すると、後期に生成された誤りリザルトは全て受け入れられるので、 E'_n は以下の式で表される。

$$E'_n(s_1 = 0, s_2, T_s) = P_1 \times P_2 \times (Rn_1 + Rn_2) \quad (13)$$

$$= (1 - qs_2)^{\min(T, T_D) - T_s} \times s_2(\min(T, T_D) - T_s)$$

E'_n を最大化するため、 s_2 について偏微分する。

$$\frac{\partial}{\partial s_2} E'_n(s_1 = 0, s_2 = \hat{s}_2, T_s) = 0 \quad (14)$$

$$\hat{s}_2 = \min\left(1, \frac{1}{q(1 + \min(T, T_D) - T_s)}\right) \quad (15)$$

式(13)に式(15)を代入すると、 E'_n の最大値を得る。

$$E'_n(s_1 = 0, s_2 = \hat{s}_2, T_s) = \quad (16)$$

$$\max\left(\frac{1}{q} \times \left(\frac{\min(T, T_D) - T_s}{\min(T, T_D) - T_s + 1}\right)^{\min(T, T_D) - T_s + 1}, (1 - q)^{\min(T, T_D) - T_s} (\min(T, T_D) - T_s)\right)$$

ここで、第1項は $s_2 \neq 0$ 、第2項は $s_2 = 1$ の場合である。

4.4 信頼度に基づく多数決法の改善

式(16)で表される E'_n の最大値に対し、さらに以下の不等式を適用する。式(16)において、 $\min(T, T_D) - T_s = x (x \geq 0)$ とすると、0以上の任意の整数 x と $0 \leq q \leq 1$ に対して

$$\left(\frac{x}{x+1}\right)^{x+1} < 1 \quad (17)$$

$$(1 - q)^x \times x \leq \frac{1}{q} \quad (18)$$

が成立するため、 E'_n の上界を得ることができる。

$$E'_n(s_1 = 0, s_2 = \hat{s}_2, T_s) \leq \frac{1}{q} \quad (19)$$

式(19)は、ある妨害者がどのような s_2 と T_s を用いても、受け入れられる誤りリザルト数は $1/q$ 以下であることを意味している。各妨害者の信頼度が閾値を越えるまでの時間 T_s は、それぞれの妨害者が抜取検査を受けるタイミングによって異なるが、どのような場合でも、妨害者1台あたりの上限は $1/q$ となる。

そこで、式(19)を利用して、各妨害者の s_2 と T_s がどの

ような値であっても信頼性条件を保証できるよう、信頼度に基づく多数決法を以下のように拡張する。ここで、各ワーカの性能が異なる場合は信頼度計算式も異なるため [10]、簡単に、全ワーカの性能が均一である場合を考える。

まず、妨害者の最大台数は $[W \times f_{max}]$ なので、受け入れられる誤りの総数 E_{all} の期待値上限は以下で与えられる。

$$E_{all} \leq \frac{1}{q} \times [W \times f_{max}] \quad (20)$$

ここで、各ジョブが平均 n 個のリザルトで完了したと仮定する。冗長度 n は、実行時間 T の期待値とワーカ数 W 、ジョブ数 N から算出することができる [9]。十分な数のジョブを用意した場合、各ジョブを1リザルトで終わらせることができれば $n = 1$ となる。

生成されるリザルトの総数は $W \times \min(T, T_D)$ なので、式(20)と n を用いて、 ϵ の上限は式(21)で与えられる。

$$\epsilon = \frac{E_{all}}{W \times \min(T, T_D)} \leq \frac{n \times f_{max}}{q \times \min(T, T_D)} \quad (21)$$

よって、以下の条件を満たす場合、 $\epsilon \leq \epsilon_{acc}$ となる。

$$q \times \min(T, T_D) \geq \frac{n \times f_{max}}{\epsilon_{acc}} \quad (22)$$

式(22)は、この条件式を満たす q と T_D を用いれば、各妨害者がどのような s_2 と T_s を用いても、信頼性条件の保証が可能であることを意味している。

4.5 ステップ関数以外のモデルの考察

信頼度に基づく多数決法では、妨害者の信頼度が閾値を超えた場合には、その妨害者が生成した誤りリザルトは全て受け入れられる。ここで、この妨害者は T_s 以降でも自由に s の値を変えることができるが、全ての誤りリザルトが受け入れられる以上、誤り率の最大化は、誤りリザルトの数の最大化に等しい。その数が最大化されるのは、4.2節で述べたように、 s の値をある一定値とした場合である。つまり、妨害者が妨害率 s を T_s 以降に変化させる場合は、それが増減のどちらの変化であっても、誤り率は減少する。また、妨害者は何度でも s を変化させることができるが、変化の度に誤り率は減少することになる。

以上の考察により、誤り率が最大化されるのは、妨害者が T_s 以降、1度も妨害率 s を変化させず、4.3節で示したステップ関数型のモデルに従う場合であると考えられる。このため、ステップ関数型の場合の誤り率を指定値 ϵ_{acc} 以下に抑えることができれば、 s がどのように変化する場合でも、信頼性条件 $\epsilon \leq \epsilon_{acc}$ が保証される。ただし、この考察は、計算開始から T_s （信頼度が閾値を越えるまで）の期間、 $s_1 = 0$ を保つことが誤り率の期待値を最大化する、という4.3節の仮説に基づいている。よって、 s が任意の変化をする場合に対する厳密な証明のためには、この仮説が正しいものであるか詳細に検討する必要がある。

5. シミュレーションによる評価

5.1 本シミュレーションの狙い

本稿では、ワーカの妨害率 s が、4.3 節に示したステップ関数型モデルに従って時間的に変化する場合の、信頼度に基づく多数決法の信頼性保証能力を評価する。

まず、5.2.1 節では、妨害率がステップ関数状に変化する場合の誤り率を評価するため、様々な s_2 と T_s を用い、以下の仮定の下でシミュレーションを行った。

- 妨害者にとって自身の検査通過回数 k は不明であるが、その期待値はターン毎に q ずつ増加する。よって、 k の期待値と式 (1) から自身の信頼度を計算し、これが閾値を超えていない場合は $s = 0$ 、閾値を超えた場合は十分に信頼度が上昇したと判断して、 s_2 を用いた誤りの生成を開始する。

次に、5.2.2 節では、4.4 節で提案した、信頼度に基づく多数決法のパラメータ制限による改善法の効果を調べるため、妨害者が誤り率を最大化するような、以下の仮定に基づいたシミュレーションを行った。

- 妨害者にとって q と自身の信頼度が常に既知であるとし、信頼度が閾値を超えていない場合は $s = 0$ を、超えている場合には s_2 を用いる。

表 1 にシミュレーションに用いたパラメータの値を示す。また、ジョブの実行順序が誤り率や計算時間に与える影響を小さくするため、ジョブスケジューリング法としてラウンドロビン法を用いている [6]。

5.2 シミュレーション結果

5.2.1 期待値の T_s に達した時点で s を変化させる場合

図 3 に、横軸に後期妨害率 s_2 を取った場合の誤り率とスループット (期限 T_D までに完了したジョブ数) を示す。図中には、各 q の値に応じて設定された T_s の値を併記している。 $\epsilon_{acc} = 0.05$, $f_{max} = 0.35$ の場合、式 (1) から $k = 4$ 回の抜取検査に通過すればワーカの信頼度が閾値 $1 - \epsilon_{acc}$ を上回るため、それぞれ $q \times T_s = 4$ となるように設定している。ただし、 $T_s \geq \min(T, T_D)$ となるような場合は、 $s = s_1 = 0$ の状態が計算終了まで続き、単純に誤り率 ϵ が 0 となってしまう。そのため、この場合は $T_s = 0$ として、最初から誤りリザルトを確率 s_2 で返すようにし、誤り率を最大化できるようにした。

表 1 シミュレーションに用いたパラメータ

ジョブ数 N	10000
ワーカ数 W	100
チェック率 q	0.1 - 0.5
許容誤り率 ϵ_{acc}	0.01, 0.05
妨害者割合 f	0.35
妨害者割合の上限 f_{max}	0.35
後期妨害率 s_2	0 ~ 1

図 3(a) から、どの q の場合も、 s_2 がある程度まで大きくなるにつれて、誤り率 ϵ も増大することが分かる。これは、 T_s までに十分な信頼度を稼いでおけば、 T_s 以降に確率 s_2 で生成された誤りリザルトが、マスタに受け入れられていることを意味する。しかし、 s_2 が一定値以上に大きい場合、例えば $q = 0.5$ の場合には $s_2 \geq 0.5$ になると、抜取検査により妨害者が検出される確率が大きくなるため、誤りリザルト数の期待値が減少し、 ϵ が減少していく。

また、図 3(b) に、 T_D までに完了したジョブ数 (スループット) を示す。後期妨害率 s_2 が大きくなるほど、妨害者が検出されやすくなるため、ワーカ全体で生成するリザルト数が減少しスループットも減少する傾向が分かる。また、 $q = 0.5$ など、検査率 q が大きすぎる場合には、検査のために本来のジョブが実行できない機会が増えスループットが減少してしまう。この場合、誤り率が常に ϵ_{acc} 以下であり、最も高いスループットが得られている $q = 0.3$ が、性能と信頼性のバランスが取れた最適値といえる。

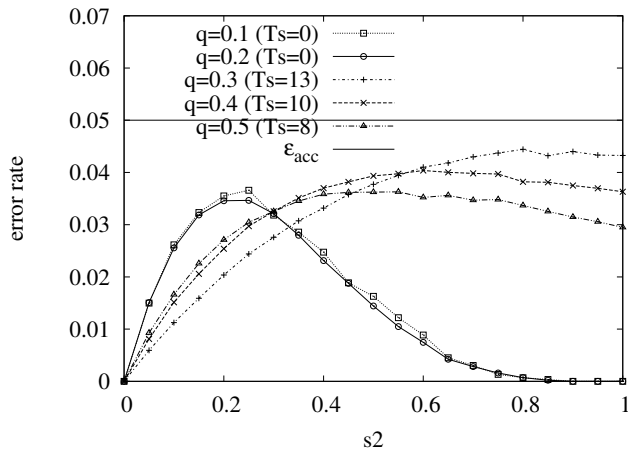
5.2.2 信頼度が閾値に達した時点で s を変化させる場合

図 4 に、横軸に後期妨害率 s_2 を取った場合の誤り率とスループットを示す。各妨害者は、妨害率として、自身の信頼度が閾値を上回るまでは $s_1 = 0$ 、信頼度が閾値に達した時点以降は s_2 を用いる。この仮定は、本来は自身の信頼度を知ることの出来ない妨害者が、誤り率が最大となってしまうような挙動を偶然起こしてしまった場合を想定している。また、各妨害者の信頼度が閾値を越えるまでの期間の長さはそれぞれ異なるが、全妨害者がそれぞれで誤り率を最大化するような挙動をする、最悪の場合を想定している。

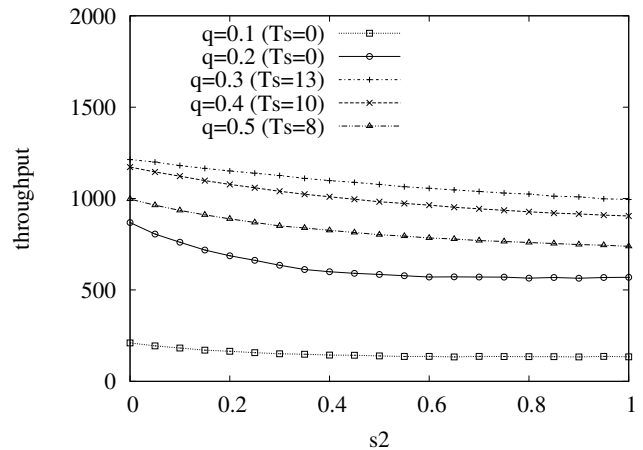
図 4(a) に示すように、このような場合の妨害者に対しては、誤り率が閾値 ϵ_{acc} を超えてしまい、信頼性保証が出来ない場合がある。すなわち、信頼度計算の前提条件が満足されない (s が時間的に変化する) 場合には、従来の信頼度に基づく多数決法では対応することができない。また、この結果は、妨害者が悪意を持たず故障等によって偶然 s が変化した場合でも、そのタイミングによっては誤り率が閾値 ϵ_{acc} を超えてしまう可能性があることを示唆している。

これに対して、本稿で提案した改善法である、式 (22) に基づくパラメータ制限を用いると、このような最悪の妨害者挙動に対しても信頼性保証が実現できる。例えば図 4(a) においては、十分な数のジョブが用意されているので冗長度 $n = 1$ として、条件式 $q \times \min(T, T_D) \geq \frac{n \times f_{max}}{\epsilon_{acc}} = 7$ を満たす $q = 0.4$ と $q = 0.5$ の場合には、どのような s_2 に対しても常に信頼性条件 $\epsilon \leq \epsilon_{acc}$ が満足されている。信頼性条件が満足されないのは、提案式の条件を満たさない $q = 0.1, 0.2, 0.3$ の場合のみである。

また、信頼性条件を満たすだけであれば $q = 0.4$ と $q = 0.5$ のどちらでも良いが、図 4(b) を見ると、 $q = 0.4$ の方がスループットが高いことが分かる。一般に、ある一定値以上



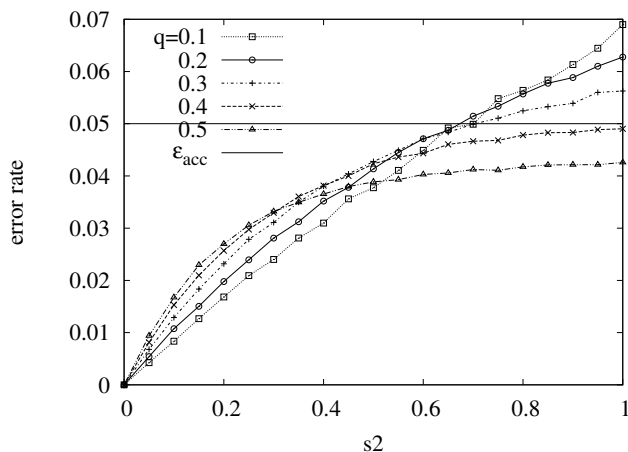
(a) 誤り率 ϵ



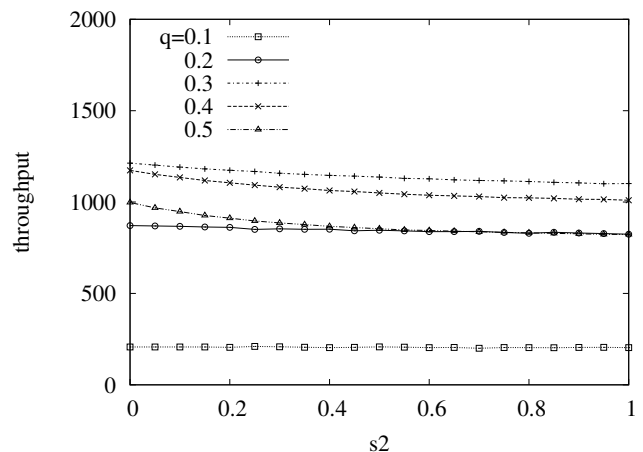
(b) スループット P

図 3 期待値の T_s に達した時点で s を変化させる場合の誤り率とスループット

($\epsilon_{acc} = 0.05, s_1 = 0.0, f = f_{max} = 0.35, \epsilon_{acc} = 0.05, T_D = 20$)



(a) 誤り率 ϵ



(b) スループット P

図 4 信頼度が閾値に達した時点で s を変化させる場合の誤り率とスループット

($\epsilon_{acc} = 0.05, s_1 = 0.0, f = f_{max} = 0.35, \epsilon_{acc} = 0.05, T_D = 20$)

に q が大きい場合は、検査問題ばかりになり性能が低下してしまう傾向があるが、本稿で提案した式 (22) を用いることで、信頼性保証を実現しつつ、出来るだけスループットが高くなるような最小の q を算出することができる。

また、提案式を用いる場合、右辺の条件式の分母に ϵ_{acc} があるため、小さな ϵ_{acc} に対して信頼性保証を実現するためには、 ϵ_{acc} に反比例して q が $\min(T, T_D)$ を大きくする必要がある。ここで、検査問題の数を十分に用意できないような場合や、 q を大きくしすぎて性能が低下してしまうような場合には、 T_D を大きくして対応することが考えられる。

図 5 に、 $\epsilon_{acc} = 0.01$ とし、 $W = 100$ 台のワーカで $N = 10000$ のジョブ全てを終わらせるのに十分な長さ

として、 $T_D = 300$ を設定した場合の誤り率とスループットを示す。この場合、提案式は $q \times \min(T, T_D) \geq \frac{n \times f_{max}}{\epsilon_{acc}} \cong 35 \times \frac{T \times W}{N}$ となり、この条件を満足する $q = 0.3, 0.4, 0.5$ の場合には、信頼性保証が出来ていることが分かる。一方、 $q = 0.1$ や 0.2 の場合には提案式の条件を満足しておらず、図からも、 $q = 0.2$ において誤り率が許容値 ϵ_{acc} を超えてしまうことが確認できる。また、 $q = 0.1$ の場合は検査頻度が低すぎるため、計算時間の間に妨害者が十分な信頼度を稼ぐことができず、結果として、誤り率がほぼ 0 となる代わりに計算時間が長くなってしまふ。

図 5(b) から、提案法を用いた場合の性能として、 $q = 0.3$ の時に計算時間が 180 程度となっている。単純な多数決法では、各ジョブに対して最低 2 個ずつリザルトを生成する

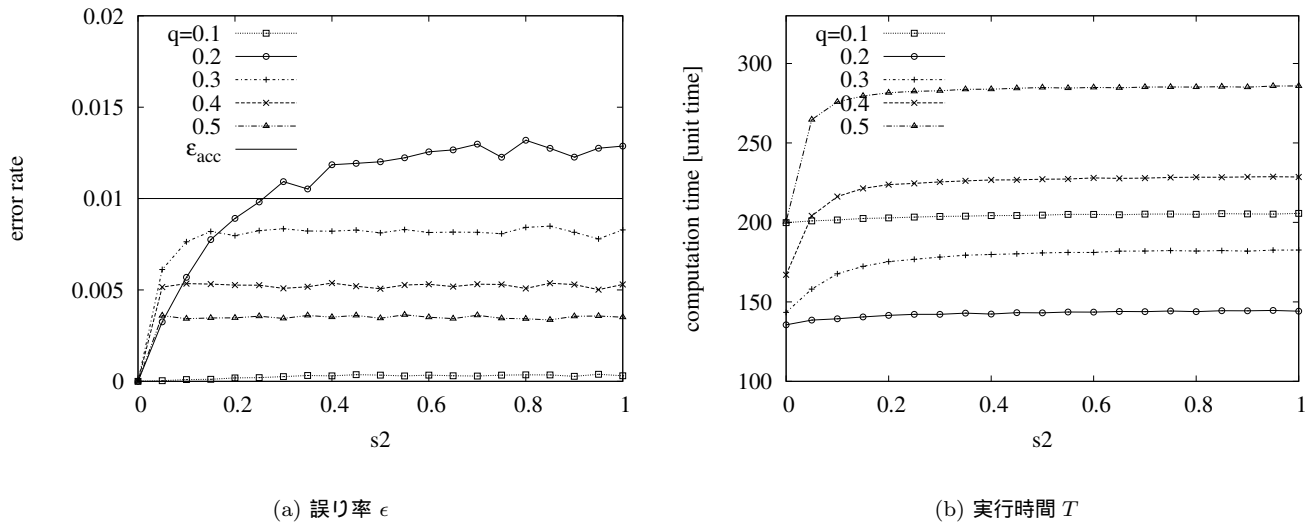


図 5 信頼度が閾値に達した時点で s を変化させる場合の誤り率とスループット
($\epsilon_{acc} = 0.01, s_1 = 0.0, f = f_{max} = 0.35, \epsilon_{acc} = 0.01, T_D = 300$)

ため、 $T = N/W \times 2 = 200$ 以上の時間がかかる。この結果は、信頼度に基づく多数決法で信頼性保証を実現しつつ、提案式を用いて適切な q と T_D を設定することで、高い性能と信頼性を両立した例といえる。

6. まとめと今後の課題

本稿では、VC システムにおける誤り発生確率（妨害率 s ）が時間的に変化する場合を想定して、信頼度に基づく多数決法のシミュレーションによる評価と、同手法の改善を行った。まず、誤り率 ϵ の期待値を最大化するモデルとして、 s_1, T_s, s_2 の 3 パラメータから成るステップ関数モデルを示し、このモデルでシミュレーションを行った結果、従来の信頼度計算式では信頼性保証が実現できないことが分かった。次に、信頼度に基づく多数決法の改善として、式 (22) に示す条件を課すことで信頼性保証が実現されることを解析的に示し、またシミュレーションによりこれを確認した。

今後の課題として、ワーカの妨害率が $s_1 = 0$ を用いたステップ関数状に変化する場合に、誤り率 ϵ の期待値が最大化されることを厳密に証明することが挙げられる。この証明により、妨害率の変動がステップ関数を含む任意の関数に従う場合において、信頼性保証が実現できることが示される。また、各ワーカの性能が異なる場合を想定した信頼度計算式 [10] と本稿の提案を組み合わせることで、ワーカの性能や妨害率によらず常に信頼性保証を実現する、より実用的な信頼性保証手法を提案していく予定である。

参考文献

[1] SETI@home <http://setiathome.ssl.berkeley.edu/>
[2] B. Knispel, et.al, “Pulsar Discovery by Global Volunteer

Computing”, Science, Vol.329, no.5994, 2010.
[3] Folding@home <http://folding.stanford.edu/>
[4] UC BERKELEY News Center (July 22, 2013) <http://newscenter.berkeley.edu/2013/07/22/new-app-puts-idle-smartphones-to-work-for-science/>
[5] D. Kondo, F. Araujo, P. Malecot, P. Domingues, L. M. Silva, G. Fedak, and F. Cappello, “Characterizing Error Rates in Internet Desktop Grids”, 13th European Conf. Parallel and Distributed Comput., pp. 361–371, 2007.
[6] L. F. G. Sarmenta, “Sabotage-Tolerance Mechanisms for Volunteer Computing Systems”, Future Generation Computer Systems, Vol. 18, Issue 4, pp.561-572, 2002.
[7] F. Araujo, J. Farinha, P. Domingues, G.C. Silaghi, and D. Kondo, “A Maximum Independent Set Approach for Collusion Detection in Voting Pools”, J. Parallel and Distributed Comput., Vol. 71 (10), pp. 1356 – 1366, 2011.
[8] BOINC <http://boinc.berkeley.edu/>
[9] K. Watanabe, M. Fukushi and S. Horiguchi, “Optimal Spot-checking for Computation Time Minimization in Volunteer Computing”, Journal of Grid Computing, Vol. 7, Issue 4, pp.575 – 600, 2009.
[10] 渡邊寛, 舩曳信生, 中西透, 福士将, “ボランティアコンピュティングにおけるワーカの性能差を考慮した信頼度計算式の拡張”, 電子情報通信学会論文誌 D, Vol.96-D, No.6, pp.1413-1424, 2013.