

# 信頼度評価に基づく多数決による 高信頼ボランティアコンピューティングシステムの実装

## Implementation of Reliable Volunteer Computing Systems with Credibility-based Voting

小澤 知宏<sup>†</sup>  
Tomohiro Ozawa

渡邊 寛<sup>†</sup>  
Kan Watanabe

福士 将<sup>†</sup>  
Masaru Fukushi

堀口 進<sup>†</sup>  
Susumu Horiguchi

### 1 はじめに

ボランティアコンピューティング (VC) は、多数のボランティア参加者 (ワーカ) により提供される遊休計算資源を利用して、大規模並列計算環境を安価に構築する手法である。VC では、ワーカの中に誤ったリザルトを返す妨害者が混じる可能性があるため、冗長計算等による高信頼化が必要不可欠である。現在多くの VC システムで採用されている  $m$ -first 多数決には、過剰な冗長計算により VC システム全体の性能を著しく悪化させてしまうという問題があり、より効率的な高信頼化手法の開発と実装評価が求められている。本稿では、高信頼化手法として高い効率を示されている、信頼度評価に基づく多数決 [1] に着目し、実際の VC システムにおける有用性を評価するために、同手法を実装した VC システムを構築して性能評価を行った。

### 2 VC システムと高信頼化手法の実装

#### 2.1 VC システム

本稿では、実際の VC で広く用いられているマスターワーカモデルのシステムを、複数の計算機ノードとソケット通信ライブラリを用いて図1のように構築した。マスターワーカモデルは、システム全体を管理するノード (マスター) 1 台と、実際の計算を行う複数のノード (ワーカ) から成り、各ノードはそれぞれ次のような役割を持つ。

- マスタは、計算問題 (ジョブ) をワーカに配り、計算結果 (リザルト) を集める。また、集めたリザルトに対して高信頼化手法を適用する。
- 各ワーカは、任意のタイミングでマスタからジョブを受け取り、そのリザルトをマスタに返却する。

図1に、構築した VC システムにおける、マスタと1つのワーカ間のジョブの配布、リザルトの送信、高信頼化の実行の流れを示す。まず、ワーカは、任意のタイミングでマスタに対してジョブの送信要求を送る。次に、

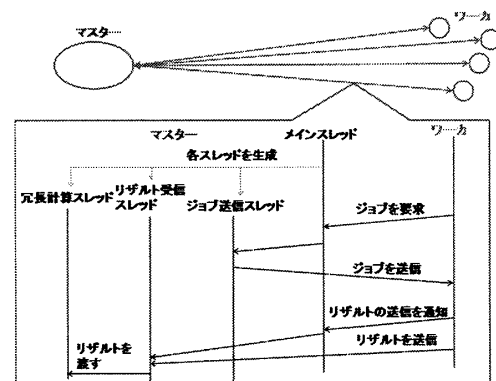


図1: 構築した VC システム

送信要求を受け取ったマスタは、未終了のジョブの中から1つを選び、ワーカに対してジョブを送信する。ジョブを受け取ったワーカは、そのジョブを実行し、リザルトを生成した後、マスタに対してリザルトの送信を通知する。通知を受け取ったマスタは、リザルトを受信後、冗長計算などの高信頼化手法を適用し、そのリザルトを採用するかどうかなどを決定する。

以上の動作のうち、マスタ側で行うジョブ配布、リザルト受信、高信頼化手法の実行は、処理を高速化するためそれぞれ別々のスレッドで独立に行っている。スレッド間のデータ受け渡しには、次の3種類の待ち行列 (キュー) を用いている。

- ジョブ送信キュー: ワーカからのジョブ送信要求と対応するソケット番号を格納する。
- リザルト受信キュー: ワーカからのリザルト送信通知と対応するソケット番号を格納する。
- 高信頼化キュー: ワーカから受信したリザルトを格納する。

#### 2.2 高信頼化手法

本稿では、高信頼化手法として  $2m-1$  多数決、 $m$ -first 多数決、信頼度評価に基づく多数決の3つを実装した。 $2m-1$  多数決は各ジョブに対して  $2m-1$  個のリザル

<sup>†</sup>東北大学 大学院情報科学研究科, Graduate School of Information Sciences, Tohoku University

トを集め多数決を行うことで最終的なリザルトを決定する。  $m$ -first 多数決は、現在多くの VC システムで採用されており、各ジョブに対して少なくとも冗長度 ( $m$ ) 個以上のリザルトを集め多数決を行う。これに対して信頼度評価に基づく多数決 [1] では、あるワーカの返したりザルトが正しい確率 (信頼度) を評価し、信頼度に基づいて必要な冗長度 (通常  $m$  以下) を動的に決定することで、より効率的な高信頼化が可能である。しかし、実際の VC システムで信頼度評価に基づく多数決を行う場合には、ジョブを管理するマスタにおいて信頼度計算等の比較的複雑な計算処理が必要になるため、マスタの性能やシステムの規模によっては、単純な  $m$ -first 多数決を用いた場合よりもシステム全体の性能が低くなってしまいう可能性がある。

### 3 性能評価

信頼度計算等のオーバーヘッドを含めた実際の高信頼化手法の性能を評価するため、マスタ (Core2Duo 3GHz の PC) と 4 台の Play Station 3 (PS3) よりなる VC システムを構築した。各 PS3 上では、1つのスレッドを 1 ワーカとし、250 スレッドを並列実行することで、仮想的に合計 1000 台のワーカを動作させる。ただし、1 台の PS3 つまり仮想的に 250 台のワーカは、間違ったりザルトを返すワーカ (妨害者) であり、各高信頼化手法のパラメータは、それぞれ同程度の信頼性を実現するように設定する。また、各ワーカに与えられるジョブの中身は "sleep" であり、ジョブを受け取ったワーカは指定された時間の後にマスタにリザルトを返却する。

#### 3.1 システムの実行時間の評価

図 2 に、ジョブ総数を 3000 とした場合の、1 ジョブの計算時間に対するシステム全体の実行時間 (すべてのジョブが終了するまでの時間) を示す。図 2 より、1 ジョブの計算時間によらず、信頼度に基づく多数決を用いた場合は他の 2 つの場合よりも高速であることが分かる。また、信頼度に基づく多数決を用いた場合は、1 ジョブの計算時間が増加してもシステム全体の実行時間増加が比較的ゆるやかであり、1 ジョブの計算時間が大きいほど、 $m$ -first 多数決を用いた場合と比較して実行時間が小さくなると思われる。一般的な VC における 1 ジョブの計算時間はおよそ数時間程度なので、信頼度評価に基づく多数決は、VC における高信頼化手法として極めて有効であるといえる。

#### 3.2 高信頼化手法に要する時間の評価

図 3 に、すべてのジョブが終了するまでの間に、高信頼化手法に要した時間の合計値を示す (ジョブ総数 100)。ただし、横軸は 1 つのジョブに対して集めるリザルト数 (冗長度)、縦軸は高信頼化手法に要した時間 (右軸は

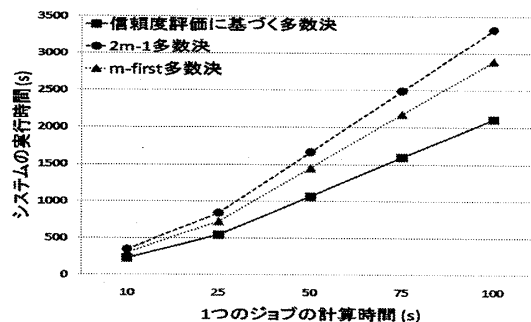


図 2: システム全体の実行時間

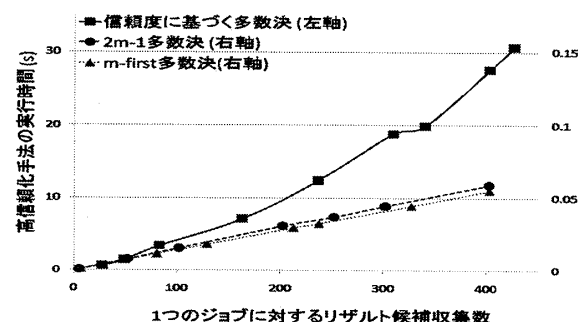


図 3: 高信頼化手法に要する時間

$2m - 1$  多数決と  $m$ -first 多数決、左軸は信頼度評価に基づく多数決の場合) である。図 3 より、冗長度が比較的小さい場合、冗長度にほぼ比例して各手法の実行時間が增大していることが分かる。また、信頼度評価に基づく多数決に要する時間は、例えば冗長度 5 の場合 (一般的な VC システムでは通常 3 から 5)、 $m$ -first 多数決の 0.001 秒程度に対して、0.1 秒程度の計算時間を要している。これらの実行時間は、一般的な VC における 1 ジョブの計算時間に比較して十分小さいため、高信頼化手法によるオーバーヘッドはほぼ無視できるといえる。ただし冗長度が 50 を超える場合、信頼度評価に基づく多数決は計算時間が指数的に増大して傾向にあるため、高い冗長度を設定する高信頼性を要する計算ではオーバーヘッドが非常に大きくなり、システム全体の計算性能を低下させてしまう可能性がある。

### 4 まとめ

本稿では、高信頼化手法を実装したシステムを構築し性能評価を行うことで、実際の VC システムにおける各手法の有用性を評価した。実験により、1 ジョブの計算時間が比較的大きい場合は、 $m$ -first 多数決と比較して信頼度評価に基づく多数決が有効となることが分かった。

### 5 参考文献

- [1] Luis F. G. Sarmenta, "Sabotage-Tolerance Mechanisms for Volunteer Computing Systems", Future Generation Computer Systems, Vol. 18, Issue 4, pp.561-572, 2002.