L-024

# セキュアプロセッシングにおける先行処理による処理時間改善に対する定量的評価
# Evaluation of advance processing in terms of processing time on Secure Processing

廣瀬吉隆 † 稲元 勉 ‡ 樋上喜信 ‡ 小林真也 ‡
Yoshitaka Hirose Tsutomu Inamoto Yoshinobu Higami Shin-ya Kobayashi

## 1. まえがき

近年のインターネットの普及により，グリッドコンピューティングは身近になりつつある．グリッドコンピューティングは，信頼できない主体が管理する計算機を用いるエクスターナル GRID と，そうでないインターナル GRID とに大別できる．エクスターナル GRID は，処理実行用の計算機（“処理ノード”）を潜在的に多数有する．一方で，処理ノードに処理プログラムや処理データが漏洩し不適切に利用されたり，処理ノードが意図的に不正な処理結果を返す恐れがある．著者らは，エクスターナル GRID の欠点を補うための方法に “セキュアプロセッシング” と名付けて取り組んでいる．

著者らは，セキュアプロセッシングの一環として，処理結果の信頼性を向上するための “多重処理”[1]，処理内容を隠蔽するための “プログラム分割”[2] という手法を提案している．多重処理は，生起が稀な出力データを信頼できないと判定し除外するために，同じ処理内容を複数の処理ノードへ依頼して得られた複数の出力データにわたる多数決をとる手法である．プログラム分割では，エクスターナル GRID で処理される “全体プログラム” は，順に処理される “プログラム断片” へと分割される．プログラム断片の処理順を “段” と表したとき，$k+1$ 段のプログラム断片の処理を開始するには，$k$ 段のプログラム断片の処理完了を待たなければならない．この待ち時間を短縮するために，筆者らは，$k$ 段の処理が完了する前に $k'$ $(> k)$ 段の処理を開始する “先行処理” という手法を提案している．この手法については 2 節に後述する．本稿は，先行処理を用いた場合の処理完了時間を定式化し，その時間の先行処理による改善度合いを定量的に評価することを目的とする．

## 2. 先行処理

プログラム分割を用いた場合，$k$ 段目のプログラム断片の処理結果を得てから，これを入力データとして $k+1$ 段目のプログラム断片の処理を開始することが基本である．プログラム分割にくわえ多重処理を用いると，各段において，多数決の条件を満足する処理結果が集まってから次段の処理が開始される．このような状況下で全体プログラムの処理完了時間を短縮するには，ある時点で得られている出力データを処理結果と仮定し，後続するプログラム断片の処理を開始することが考えられる．この手法を “先行処理” と呼ぶ．この手法を用いることで，グリッドに含まれるであろう高速な処理ノードによる出力データが得られれば後続段の処理を開始できるため，処理完了時間の短縮が可能である．

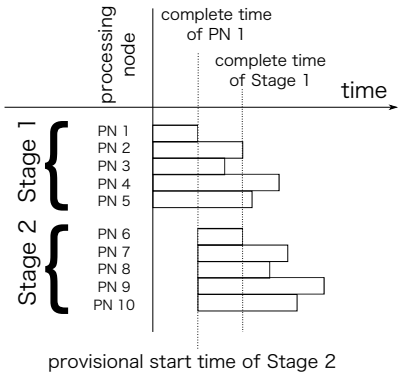


図 1: 暫定型先行処理の処理過程の例

先行処理は，暫定型と網羅型に大別できる．暫定型では，ある個数（“暫定用閾値”）だけ集まった同一の出力データを “暫定結果” と呼び，暫定結果を処理結果とみなして後続段の処理が開始される．$k$ 段目で同じ出力データがある個数（“確定用閾値”）だけ集まって $k$ 段目の処理結果が確定し，処理結果と暫定結果が異なることが判明すると，それまで進めていた $k+1$ 段目以降の処理は破棄し，$k+1$ 段目の処理結果を入力データとして $k+2$ 段目の処理を開始する．この一連の処理を “ロールバック” と呼ぶ．網羅型は，ある段で新規な出力データが得られるたびに，その出力データを処理結果とみなして後続段の処理を開始する手法である．

多重度 5・確定用閾値 3 であり，ロールバックの生じない暫定型先行処理の処理過程の例を図 1 に示す．この例では，先行処理を行わない場合，3 番目に処理が完了する処理ノード 2 の出力データが得られるまで 1 段目の処理結果は確定しない．暫定用閾値 1 の暫定型先行処理を行った場合，先頭から 1 番目に完了する処理ノード 1 の出力データが得られると，これを 1 段目の処理結果であるとみなして，2 段目の処理を開始する．

## 3. 処理完了時間の定式化

### 3.1. 前提

管理ノードは 1 つ，処理ノードは十分多いとする．また，ノード間の通信時間は考慮せず，管理ノードの処理性能は十分に高く，処理ノードの処理性能は形状尺度 5，尺度母数 2/5 のガンマ分布に従うとする．くわえて，全体プログラムのサイズは 50 であり，プログラム断片のサイズは均一であり，ある処理ノードが正しい出力データを返す確率である “ノード真正処理率” は既知とする．

†愛媛大学工学部, Faculty of Engineering, Ehime University
（※ 2015 年 4 月より いよぎんコンピュータサービス）
‡愛媛大学大学院理工学研究科, Graduate School of Science and Engineering, Ehime University

### 3.2. 記号の定義

ノード真正処理率を $p$，プログラム分割数を $D$，多重度を $M$，確定用閾値を $\theta := \lfloor M/2 \rfloor + 1$，暫定用閾値を $\tilde{\theta}$，多重処理の際 $m$ 個目の出力データが得られるまでの時間の期待値を $G(m)$ とする．$G(m)$ の値はガンマ分布にしたがう擬似乱数系列を用いて数値的に算出する．

多重処理において $a$ 回目の処理ノードが返した出力データが，正しい／不正な $b$ 個目のものである確率 $f(a,b)$, $\bar{f}(a,b)$ は，それぞれつぎのように定まる．

$$f(a,b) := {}_{a-1}\mathrm{C}_{b-1} p^{b-1} (1-p)^{a-b} \cdot p,$$
$$\bar{f}(a,b) := {}_{a-1}\mathrm{C}_{b-1} (1-p)^{b-1} p^{a-b} \cdot (1-p).$$

$\tilde{M} := 2\tilde{\theta} - 1$ としたとき，1 回の多重処理で処理結果が確定するまでの時間の期待値 $V$，暫定型先行処理で次段の処理を開始するまでの時間の期待値 $\tilde{V}$ はつぎのようになる．

$$V := G(\theta - 1) + \sum_{i=\theta}^{M} \left( f(i,\theta) + \bar{f}(i,\theta) \right) G(i),$$
$$\tilde{V} := G\left(\tilde{\theta} - 1\right) + \sum_{i=\tilde{\theta}}^{\tilde{M}} \left( f\left(i,\tilde{\theta}\right) + \bar{f}\left(i,\tilde{\theta}\right) \right) G(i).$$

暫定結果と処理結果が一致せずロールバックが生じる確率 $R$ は，(1) 式に示す関数 $h(q)$ を用いてつぎのように表せる．

$$R := h(p) + h(1-p).$$
$$h(q) := \sum_{i=\tilde{\theta}}^{\tilde{M}} {}_{\tilde{M}}\mathrm{C}_i q^i (1-q)^{\tilde{M}-i} \cdot \sum_{j=\theta-\tilde{M}+i}^{M-\tilde{M}} {}_{M-\tilde{M}}\mathrm{C}_j (1-q)^j q^{M-\tilde{M}-j}. \quad (1)$$

### 3.3. 処理完了時間の期待値の定式化

先行処理を用いない場合の全体プログラムの処理完了時間の期待値は，段ごとに $V$ かかるため，$D \cdot V$ となる．

暫定型先行処理を用いる場合の処理完了時間は，ロールバックが生じた場合に $V$，そうでない場合に $\tilde{V}$ の計算時間がかかるとして，つぎのように定まる．

$$\sum_{i=0}^{D-1} {}_{D-1}\mathrm{C}_i R^i (1-R)^{D-1-i} \left( iV + (D-1-i)\tilde{V} \right) + V.$$

多重処理において，$i$ 個目の出力データが，新出かつ正しい確率を $g(i)$，新出かつ不正な確率を $\bar{g}(i)$ とすれば，それらはつぎのように定まる．

$$g(i) := (1-p)^{i-1} p, \quad \bar{g}(i) := p^{i-1}(1-p).$$

これらを用い，網羅型先行処理を用いる場合の全体プログラムの処理完了時刻はつぎのようになる．

$$(D-1)\left( G(1) + \sum_{i=2}^{\theta-1} \left( g(i) + \bar{g}(i) \right) G(i) \right) + V.$$

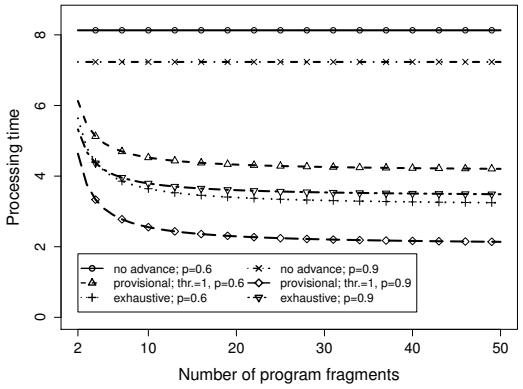


図 2: 全体プログラムの処理完了時間の期待値

### 4. 評価結果

ノード真正処理率 $p$ が 0.6, 0.9 である場合の，先行処理を用いない手法，暫定用閾値 1 での暫定型先行処理，および網羅型先行処理それぞれの全体プログラムの処理が完了するまでの時間の期待値を図 2 に示す．図 2 より，先行処理により処理時間を短縮できること，およびノード真正処理率が低い場合は網羅型のほうが，高い場合は暫定型のほうが，処理完了時間が短いことを確認できる．

### 5. あとがき

本稿では，エクスターナル GRID の信頼性を向上するための手法であるプログラム分割・多重処理を併用した場合，全体の処理完了時間が長くなるという問題への対策である先行処理を取り上げた．先行処理を暫定型と網羅型に大別し，それぞれの場合について全体プログラムの処理完了時間を定式化し，いくつかの状況下で具体値を解析的に算出した．この結果，処理ノードが正しい出力データを返す確率が高い場合は暫定型が，そうでない場合は網羅型が，より有効であることが示された．

今後の課題として，定式化の妥当性の検証やノード間の通信時間を考慮することなどがあげられる．

### 謝辞

本研究の一部は JSPS 科研費（基盤研究 (C)）26330105 の助成を受けたものである．

### 参考文献

[1] Sugimoto, K., Hirata, K., Higami, Y. and Kobayashi, S.: Multiplexing Scheme with Distributed Processing in External Grids, *Polish Journal of Environmental Studies (Selected Paper of ACS 2009)*, Vol. 18, No. 4A, pp. 50–53 (2009).

[2] Himeda, K., Hirata, K., Higami, Y. and Kobayashi, S.: Consideration of Characteristics of Programs for Concealing Purpose of Processing in Distributed Computing Systems, *Polish Journal of Environmental Studies (Selected Paper of ACS 2008)*, Vol. 17, No. 4C, pp. 226–229 (2008).