1ZA-08

エクスターナルグリッドにおける処理ノード再利用時の機密性向上手法の提案

大西 伊吹 † 遠藤 慶一 † 小林 真也 †
愛媛大学大学院理工学研究科 †

1. はじめに

エクスターナルグリッドは，インターネット上に存在する不特定多数の計算機でグリッドを構成し，分散処理を行う技術である．しかし，不特定多数の計算機を用いるため，処理内容流出の危険性がある．処理内容を不正に解析しようとする処理ノードを悪人と呼ぶ．これまで，機密性，高速性，信頼性の向上のため，プログラム分割・多重化・先行処理などが提案されてきた [1]．特に，複数の計算機で同一処理を行い，最速の結果を用いて処理時間を短縮しつつ，多数決で信頼性を確保する先行処理では，機密性が低下することがわかっている．機密性改善のため，過去に利用した計算機の処理性能を記録し，性能の近い計算機同士を利用する手法がある．性能の近い計算機の利用は，先行処理開始から結果確定までの時間を短縮し，計算機の利用数を減らすことでこの危険性を低減する．一方で，同一計算機が同一プログラム内で何度も利用され，プログラム断片を数多く取得される危険性がある．本稿では，特定の悪人が処理内容を解析しづらくすることを目的とした手法を提案し，それによって機密性が向上することを定量的に示す．

2. 提案手法

本稿では，特定の悪人が複数の断片を取得した際でも，依存関係のある断片同士が連続せず，解析がしづらくなる手法を提案する．この手法は，特定の悪人が依存関係のある断片を連続で取得しづらくするため，記録している処理ノードを複数グループに分割する．そして，依存関係のある断片同士を可能な限り別のグループの処理ノードで処理することで，プログラム断片を特定の処理ノードが連続して取得することを防ぐ．また，非常に性能の高い処理ノードは数が少なく，先行処理開始から処理結果の確定までの時間が長くなる要因となるため，その利用は避ける．

2.1. 提案手法のアルゴリズム

提案手法のアルゴリズムを以下に示す．

1. これまでに利用してきた処理ノードのうち，処理性能の近いものが多いと考えられる，処理性能上位20%から40%のものを高性能な処理ノードとする．また，上位50%以下のものを低性能な処理ノードとする．
2. 高性能な処理ノードと低性能な処理ノードをそれぞれ n 個のグループに分ける．グループ分けの際は各グループの処理ノードの数ができる限り一定になるように行う．このときの高性能なグループを $G_{h1}, G_{h2} \cdots G_{hn}$，低性能なグループを $G_{l1}, G_{l2} \cdots G_{ln}$ とする．
3. あるプログラム断片の処理を，多重度の数だけの処理ノードに依頼する．依頼する処理ノードは以下の方法で決定する．
   - プログラム断片の全ての直前断片で利用されていないグループ G_{hx},G_{lx} が存在する場合
     (a) G_{hx} から多重度の4割の処理ノードを選ぶ．G_{hx} からは，利用された日時が最も古いものを利用する．
     (b) G_{lx} から多重度の1割の処理ノードを選ぶ．G_{lx} からは，利用された日時が最も古いものを利用する．
     (c) 残り5割の処理ノードをインターネット上から選ぶ．
   - プログラム断片の全ての直前断片で利用されていないグループが存在しない場合
     (a) インターネット上から多重度の数だけの処理ノードを選ぶ．

表 1: 対象のプログラムにおける各断片の直前断片数の分布

直前断片数	0	1	2	3	4	5
断片数	258	1244	1362	1458	682	668

3. シミュレーションによる検証方法

2節で提案したアルゴリズムの効果を定量的に評価するため，シミュレーションを行う際の前提条件や評価の指標について説明する．

3.1. シミュレーションの前提条件

シミュレーションでは，C言語で記述された，高速フーリエ変換 (fft) を行うプログラム [2][3] を用いる．シミュレーションにあたって，このプログラムを1行ごとのプログラム断片に分割し，依存関係を調査した．それぞれの断片の直前断片数は表1のように分布していた．なお，直前断片を6つ以上持つプログラム断片は存在しなかった．

また，処理ノードは，形状尺度 $k = 5$，尺度母数 $\Theta = 2/5$，期待値2に基づくガンマ分布に沿って存在するものとする．

3.2. シミュレーションの評価指標

提案手法の機密性を定量的に評価するための評価指標を以下の3つに定める．これら3つの評価指標をグループ分けのない場合及びグループ数2〜6の場合で比較する．

Proposal of a method to improve confidentiality when reusing processing nodes in an external grid
I. Oonishi, K. Endo, S. Kobayashi, Graduate School of Science and Engineering, Ehime University

Copyright ©2022 Information Processing Society of Japan.
All Rights Reserved.

表 2: グループ数を変化させた際の被取得数の変化

グループ数	なし	2	3	4	5	6
被取得断片数	337	336	337	336	332	335

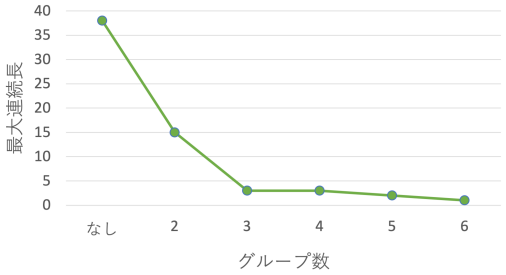

図 1: グループ数を変化させた際の最大連続長の変化

- **プログラム断片の被取得数**
  プログラム断片のうち，実際に悪人に取得されたプログラム断片の数．被取得数が大きいほど，機密性は低いと言える．
- **最大連続長**
  悪人に取得されたプログラム断片のうち，連続して依存するプログラム断片が取得された最大の数．この数が大きいほど，悪人は処理内容の解析が容易になる．
- **連続断片の数**
  悪人に取得されたプログラム断片のうち，1,2,3 及び 4 以上連続して依存するプログラム断片が取得された数．長く連続しているものが多いほど，悪人は処理内容の解析が容易になる．

4. シミュレーションの結果と考察

本節では，提案手法を用いない場合，提案手法を用いてグループ数を 2〜6 つに変化させた場合での各種シミュレーションの結果と考察を示す．なお，表や図中の「なし」の表記は，提案手法を利用しない場合を表す．

4.1. 被取得数の変化

プログラム断片の被取得数を各条件で比較した．結果を表 2 に示す．グループ数はいずれの場合でも大きく変化することはなかった．このことからプログラム断片の被取得数は，提案手法の利用の有無によって変化しないことがわかる．

4.2. 最大連続長の変化

取得されたプログラム断片の最大連続長を各条件で比較した．結果を図 1 に示す．最大連続長は提案手法を用いない場合が最も大きく，グループ数を多くするごとに小さくなった．このことから，最大連続長は，提案手法の利用によって短くなり，更にグループ数を増やすごとに短くなる事がわかる．

4.3. 連続断片数の変化

取得されたプログラム断片のうち，各連続数ごとの連続断片の数を各条件で比較した．結果を図 2, 図 3 に示す．連続数 4 以上の断片の数は，提案手法を用いない場合が最も多く，グループ数が多くなるごとに数を減らす結果となった．連続数 2,3 の断片数は，提案手法を用いない場合よりも，グループ数 2 のときの方が数が多い結果となった．連続数 1 の断片に関しては，提案手法を用いてるときが最も少なく，グループ数が多くなるごとに数を増やす結果となった．このことから，4.1 節より取得されたプログラム断片数は一定である一方で，連続数が少ない断片が多くなるため，取得されるプログラム断片をより細かく分割されたといえる．

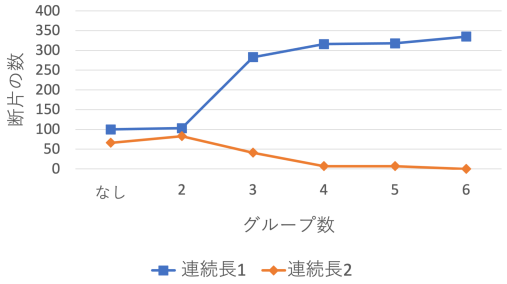

図 2: グループ数を変化させた際の連続数 1,2 断片数の変化

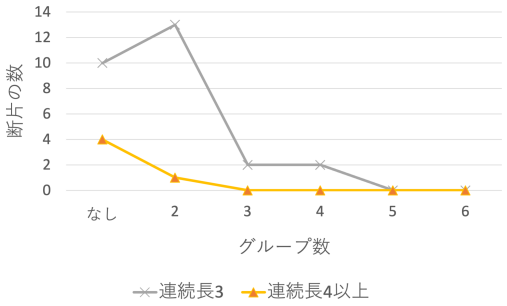

図 3: グループ数を変化させた際の連続数 3 及び 4 以上の断片数の変化

5. まとめ

本稿では， エクスターナルグリッドにおいて，過去に利用した処理ノードを再利用する方法において，特定の悪人が多くのプログラム断片を取得してしまう問題を解決するための手法を提案し，定量的な評価を行った．提案手法を用いることで，悪人が取得するプログラム断片の依存関係を切ることができ，機密性の向上につながる．今後の課題として，複数の悪人が共謀した際の機密性を改善することが挙げられる．

参考文献

[1] 大西伊吹 遠藤慶一 小林真也 (2021) “エクスターナルグリッドにおいて処理ノードの処理性能が機密性に及ぼす影響の評価” 情報処理学会第 83 回全国大会講演論文集

[2] 奥村晴彦 (2018) “[改訂新版]C 言語による標準アルゴリズム事典” 技術評論社

[3] 奥村晴彦 “[改訂新版]C 言語による標準アルゴリズム事典サポートページ” https://github.com/okumuralab/algo-c/blob/master/src/fft.c(2021-12-29)

Copyright ©2022 Information Processing Society of Japan.
All Rights Reserved.