悪人集団による盗視に対抗する保護処理を用いたエクスターナルグリッドの性能評価

山口 晃右[1]　稲元 勉[1]　樋上 喜信[1]　小林 真也[1]

概要：ネットワーク上に存在する計算機に処理を分散配置し，安価に高性能な計算資源を獲得するための技術として，グリッドコンピューティングが知られている．グリッドコンピューティングの中でも，インターネット上に存在する計算機を利用するものをエクスターナルグリッドと呼ぶ．インターネット上の計算機を利用するエクスターナルグリッドは，実質無制限の計算資源を獲得できるが，インターネット上には悪意を持った計算機の管理者が多数存在すると考えられる．また，ネットワーク上の悪人が共謀し，情報共有を行うことで，悪人集団による不正行為によるリスクが増加すると考えられる．本稿では，このような問題に対して，悪人集団の目的に応じた振る舞いを考慮したエクスターナルグリッドの性能評価を行うと同時に，悪人集団による不正行為に対する対策の効果の評価を行う．

1. はじめに

グリッドコンピューティングは，ネットワーク上の計算機を計算資源として利用することでハイパフォーマンスコンピューティングを実現するための，分散処理技術の一つとして知られている．グリッドコンピューティングの中でも，インターネット上の多数の管理者によって管理されている計算機を利用するものを，エクスターナルグリッドと呼ぶ．インターネットが広く普及した現代において，インターネットに接続された計算機は無数に存在する．エクスターナルグリッドは，無数に存在する計算機を処理ノードとして利用するので，実質的に無制限に計算資源を獲得できる．安価に高性能な計算資源を獲得できるエクスターナルグリッドは，大規模シミュレーションや膨大なタスクの処理への利用が期待される．

しかし，インターネット上の計算機の管理者の中には，悪意を持った人間が，一定数紛れ込んでいると考えられる．このことから，エクスターナルグリッドに処理を依頼した際に，処理内容の盗視や処理結果の真正性が保証されていない．また，悪意を持った計算機の管理者が，共謀し集団を形成することで，悪人集団による盗視の危険性が高まると考えられる．悪人集団による盗視のリスクを軽減するために，文献 [3] では，エクスターナルグリッドに依頼された処理の一部に保護処理を施すことで対策を図った．

文献 [3] において，保護処理を施す手法を評価するに当たって，悪人集団の振る舞いは 1 種類だけ設定した．しかし，悪人集団の振る舞いは，悪人集団の目論見によって異なると考えられ，評価する際に設定する悪人の振る舞いが 1 種類と言うのは，評価として不十分であるといえる．

本稿の目的はエクスターナルグリッドの安全性の向上をめざし，エクスターナルグリッドの安全を定量的に裏付けすることである．到達目標として，悪人集団の目的を考慮し，その目的に沿った悪人ノードの振る舞いを複数設定したエクスターナルグリッドにおける，エクスターナルグリッド自身の性能評価や保護処理を施す手法の性能評価を行う．以上の定量的な評価を行うことで，悪人ノードの振る舞いの違いによって，エクスターナルグリッドに性能にどの程度の影響が現れるのかや，保護処理を施す手法がエクスターナルグリッドの安全性の向上に繋がることを示す．

本稿の第 2 章では，エクスターナルグリッドの問題と，各問題に対して考案されてきた諸技術について述べる．第 3 章では，悪人集団の目的について述べ，エクスターナルグリッドや保護処理を施す手法を評価する際に設定する，悪人の振る舞いについて述べる．第 4 章では，評価を行う際の評価基準について述べ，シミュレーションの条件について述べる．続いて第 5 章では，異なる悪人の振る舞いを設定したシミュレーションの結果を用いて，エクスターナルグリッドの性能や保護処理を施す手法の性能の評価を行う．

2. エクスターナルグリッドと先行研究

2.1 エクスターナルグリッドの現状

エクスターナルグリッドは，インターネット上の計算機

[1] 愛媛大学大学院理工学研究科

を計算資源として利用することで，高い処理性能を実現する．このとき，エクスターナルグリッドの一部として，エクスターナルグリッドに投入されたジョブに参加する計算機を “処理ノード” と呼ぶ．エクスターナルグリッドは，大規模演算やシミュレーションなどの方面での利用が期待されている．現在，エクスターナルグリッドを利用しているサービスとして，SETI@Home[1] が挙げられる．

しかし，今日に至るまでエクスターナルグリッドを利用した営利目的のサービスの成功は確認されていない．原因は，エクスターナルグリッドに処理を依頼した際に，処理内容の盗視や処理結果の真正性が保証されていないためである．インターネット上の処理ノードは多数の管理者によって管理されている．その中には，一定数の悪意を持った管理者が存在すると考えられる．この悪意を持った計算機の管理者を “悪人” と呼び，悪人によって管理されている処理ノードを “悪人ノード” と呼ぶ．この悪人が管理する悪人ノードによる不正行為に対して，セキュアプロセッシングが研究されている．

2.2 セキュアプロセッシング

セキュアプロセッシングは，悪人ノードによる不正行為のリスクを軽減するための技術の総称である．悪人ノードが行う不正行為には，“不正な解析” と “改竄” の 2 種類がある．

不正な解析

悪人が，エクスターナルグリッドに依頼された処理の内容に含まれる情報を盗視するための不正行為である．

改竄

悪人が，エクスターナルグリッドに依頼された処理の結果を誤ったものにするために行う不正行為である．

セキュアプロセッシングでは，上記の不正な解析と改竄に対する対策として，それぞれ “プログラム分割” と “処理の多重化” がある．

プログラム分割

プログラム分割は，エクスターナルグリッドに依頼されたプログラムを分割する．分割された小さなプログラムを “プログラム断片” と呼ぶ．元のプログラムを分割した数を “分割数” と呼ぶ．各処理ノードは，このプログラム断片を実行する．悪人ノードにプログラム断片が渡ったとしても，元のプログラムを分割したものであるので，悪人ノードが取得できる情報量は限られたものになる．このように，悪人ノードによる不正な解析に対して効果がある．

処理の多重化

1 つのプログラム断片を 1 台の処理ノードに実行させるのみでは，断片を実行したノードが改竄を行う悪人ノードである場合，以降の処理が誤った実行結果を元にしたものとなってしまう．

処理の多重化では，1 つのプログラム断片を複数台の処理ノードに実行させる．このときの処理ノードの台数を “多重度” と呼ぶ．そして，多重度台の処理ノードが返した実行結果で投票を行い，過半数を獲得した実行結果を用いて，以降の処理を進める．悪人の存在確率が，0.5 未満であれば，多重化を行うことで，正しい結果を得る確率が向上する．

投票を行う多重度台の処理ノードのまとまりを “ブロック” と呼ぶ．投票の際に，過半数に達する実行結果が一つも無い場合には “票割れ” を起こす．この処理の多重化によって，1 つのプログラム断片の実行結果の信頼性を高め，改竄のリスクを軽減する．

2.3 先行処理

処理の多重化が利用されているエクスターナルグリッドでは，あるブロックの実行結果の確定は，過半数の同一結果が集まった時点である．しかし，実行結果の確定以前に集まってきた結果には，結果的に過半数を確保する処理結果と同一であるものが存在し，その存在確率は，悪人ではないノードの存在確率と同一である．そこで，ブロック内の最早の実行結果を利用して，投票が完了する前に次のブロックに処理を進めることで，エクスターナルグリッド全体の高速化を図る．この方法を，“先行処理” と呼ぶ．ただし，先行処理に利用している最早の実行結果が，ブロック内の投票の結果，過半数を獲得できなかった場合，それまで先行して処理していた内容を取り消し，再処理を行う．再処理が発生した場合，先行処理を用いない，つまり，過半数の獲得後に，次のブロックを開始する方法と性能差が出てこない．

先行処理による再処理を防止する方法として，“網羅法” が提案されている．網羅法は，ブロック内の最早の実行結果のみを先行処理に利用するのではなく，ブロック内の処理ノードの出す実行結果で異なるものが現れた場合，各実行結果ごとに先行処理を行う．網羅法を用いることで，先行処理で発生した再処理を防ぐことができ，安定してエクスターナルグリッドの高速化を行うことができる．反面，ブロック内の処理ノードの返す実行結果を網羅しようとするので，エクスターナルグリッド全体で利用する処理ノードの総数が著しく増加するという欠点がある．[2]

2.4 プログラム断片間の依存関係と悪人の共謀による危険性

2.2 節で述べたプログラム分割によって生成される各プログラム断片の間には，依存関係が存在する場合がある．依存関係で繋がったプログラム断片を “連続したプログラム断片” と呼び，連続したプログラム断片に含まれるプログラム断片の数を “連続長” と呼ぶ．悪人が不正な解析を

行う場合，プログラム断片間の依存関係を手掛かりとすると考えられている．つまり，悪人が取得する連続長が，悪人の不正な解析の難易度に大きく影響するといえる．

更に，エクスターナルグリッド上の各悪人ノードの管理者である悪人が集団を形成する場合もある．悪人が集団を形成した場合，集団全体で取得できるプログラム断片数が増加するので，より大きな連続長を取得することが可能となる．よって，悪人集団による不正な解析の脅威がより大きくなると考えられる．

以上のような，悪人集団による不正な解析のリスクを軽減するために，我々は，エクスターナルグリッドに依頼された処理の一部に対して保護処理を施す手法を提案し，その効果を定量的に評価した．[3]

2.5 処理の一部に保護処理を施す手法

処理の一部に対して保護処理を施す手法とは，プログラム断片の一部を，悪人ノードではないと保証された “信頼できる処理ノード” に実行を依頼することである．信頼できる処理ノードにプログラム断片の実行を依頼することを “保護処理” と呼ぶ．また，保護処理を受けるプログラム断片を “被保護断片” と呼ぶ．連続したプログラム断片を等分する位置にあるプログラム断片を被保護断片とすることで，悪人集団が取得できる最大連続長を制限することができる．2.4 節で述べたように，悪人が不正な解析を行う際には，より長く連続したプログラム断片を取得することが必要である．よって，処理の一部に保護処理を施す手法を用いることは，悪人集団による不正な解析のリスクを軽減することに繋がる．

文献 [3] では，処理の一部に保護処理を施す手法の定量的な評価をシミュレーションを用いて評価した．このとき，エクスターナルグリッド上の悪人ノードの振る舞いについて，1 種類のみを設定した．具体的には，エクスターナルグリッド上に存在する悪人ノードは互いに異なった実行結果を返し，かつ，各悪人ノードが返す実行結果が誤ったものである，という条件である．しかし，実際の悪人集団は，集団内で共通の目的に応じて，各悪人ノードの振る舞いを変化させると考えられる．つまり，エクスターナルグリッドの性能評価や手法の評価を行うに当たって，悪人の振る舞いが 1 種類しか設定されていないというのは，評価として不十分であったといえる．

3. 悪人集団の目的と悪人ノードの振る舞い

3.1 悪人集団の目的

エクスターナルグリッドに処理を依頼する利用者を “クライアント” と呼ぶ．クライアントは，経済活動や研究活動のためにエクスターナルグリッドを利用すると考えられる．悪人が，クライアントに対して何らかの不正行為を働く理由には，以下のようなものが挙げられる．

クライアントの活動に関する情報の収集

例えば，悪人はあるクライアントの新製品に関する情報を取得したいと考えているとする．そして，クライアントがエクスターナルグリッドを利用するということを悪人が知っているなら，悪人はエクスターナルグリッドを介して，そのクライアントの新製品の情報を盗視しようとするはずである．つまり，悪人が不正行為を働く理由の一つには，ターゲットであるクライアントの活動に繋がる情報を得る，というものが挙げられる．この目的を達成するための不正行為に，不正な解析がある．

クライアントの活動に対する妨害

悪人があるクライアントのプロジェクトに対して，何らかの実害を与えたいと考えたとき，悪人は妨害工作を行うと考えられる．このような妨害行為を行う理由は，悪人があるクライアントのプロジェクトを中止・混乱させたい，というものが挙げられる．この目的を達成させる不正行為として，改竄が挙げられる．

以上のように，悪人の目的は大きく二つに分類できる．

3.2 脅威となる悪人ノードの振る舞い

悪人集団の目的は，クライアントの活動に関する情報の収集，クライアントの活動の中止・混乱のいずれか 1 つ，ないしは，その両方を組み合わせたものであると考えられる．そして，悪人ノードの振る舞いは，上記の目的の組み合わせによって変化する．例えば，クライアントの活動に対しての情報収集と活動の中止・混乱の両方が目的であるなら，不正な解析と改竄の 2 つの不正行為を行うことになる．

本稿で評価する際，悪人集団はクライアントの活動に関する情報収集のみを目的としているとする．この場合，悪人集団が目論む行動は二つ考えられる．

一つ目は，エクスターナルグリッドの管理者に悪人ノードと気づかれずに情報収集を行い，今後もエクスターナルグリッドに参加しつづける，というものである．具体的な悪人集団の振る舞いとして，同一の悪人集団に含まれる悪人達は全員正しい実行結果を返す．こうすることで，悪人集団は，その存在を知られることなく，不正行為を行うことが可能である．悪人ノードが悪人ノードであることを，知覚されなければ，他のクライアントがエクスターナルグリッドを利用した際にも，再びエクスターナルグリッド上で情報収集ができる．

二つ目は，悪人集団がより多くの悪人ノードをエクスターナルグリッド上に呼び込み，悪人集団が取得できる情報量を多くする，というものである．この目論見を実現するため，全ての悪人ノードが返す実行結果は誤ったものであり，かつ，各悪人ノード毎に異なった結果を返す，というような振る舞いを取ると考えられる．この振る舞いは，

エクスターナルグリッドの管理者に悪人ノードであると簡単に気づかれてしまう．一方で，票割れや，先行処理における再実行を引き起こし，より多くの悪人ノードが，処理に関わる事になる．その結果，悪人ノードの集団として，より多くの情報の獲得に繋がる．

4. 評価方法

4.1 評価基準

網羅法を利用したエクスターナルグリッドに期待する性能に対して，悪人集団がどの程度脅威になるか，という視点で評価基準を設ける．具体的には，以下の 3 点に注目する．

エクスターナルグリッドの利用処理ノード数

エクスターナルグリッドに依頼された処理が完了するまでに使用した処理ノードの台数を求めることで評価する．エクスターナルグリッドで使用される処理ノード数は，安全性の観点から可能な限り少ない方が良い．何故なら，悪人ノードの存在は，悪人の存在確率によって定まるため，エクスターナルグリッドを構成するノード数の増加が，悪人ノードの増加に繋がるためである．

悪人集団による情報取得に対する耐性

悪人集団が取得する連続長と全プログラム断片中，悪人集団が取得できたプログラム断片の割合．悪人集団が取得する連続長が長いほど，悪人集団は，断片間の依存関係を解析しやすくなる．また，取得されたプログラム断片の割合は，悪人集団が取得した情報量に相当する．

エクスターナルグリッドの処理性能

悪人集団の不正行為に対する対策は，処理性能の低下をもたらす．この点を，エクスターナルグリッドに依頼された処理が完了するまでにかかった時間を求めることで評価する．当然，処理が完了するまでにかかる時間が短い方が望ましい．

4.2 シミュレーション条件

4.1 節で述べた値を，連続したプログラム断片のみからなる，網羅法を利用したエクスターナルグリッドのシミュレーションにより求める．

モデルを特徴づけるパラメータ

シミュレーションモデルを特徴づけるパラメータは，プログラム分割数，多重度，被保護断片数，悪人の存在確率，悪人ノードの振る舞いの 5 つである．

利用可能な処理ノード数

エクスターナルグリッドを構成する利用可能なノード数は無数に存在する．

悪人ノード

エクスターナルグリッドの管理者は，自身に含まれる悪人ノードをそれ以外の処理ノードと識別することはできない．また，悪人ノードは悪人の存在確率に応じて，エクスターナルグリッド上に存在する．

各処理ノードの性能

各処理ノードの処理性能は，各処理ノードが単位時間あたりに処理できるプログラムサイズを単位として表される．処理ノードの性能分布は，形状尺度 $k = 5$，尺度分母 $\Theta = 2/5$，期待値 2 となるガンマ分布に従う．

プログラムサイズ

エクスターナルグリッドに依頼されるプログラムサイズは 100 である．

悪人ノードの振る舞い

3.2 節で述べた，2 種類の振る舞いを設定できる．1 つは，全ての悪人ノードが正しい実行結果を返す，というものである．もう 1 つは，全ての悪人ノードは誤った実行結果を返し，かつ，各悪人ノード毎に異なった実行結果を返す，というものである．

試行回数

同一条件下での試行を 1000 回行う．

5. 評価結果と考察

第 4 章で述べた評価基準に基づいて，シミュレーションによって値を求め評価を行った．以降の図中では，悪人の振る舞い A と B は，それぞれシミュレートした際に設定した悪人ノードの振る舞いを表している．悪人の振る舞い A は，全ての悪人ノードは誤った実行結果を返し，かつ，各悪人ノード毎に異なった実行結果を返す，という振る舞いである．悪人の振る舞い B は，全ての悪人ノードが正しい実行結果を返す，という振る舞いである．

5.1 利用処理ノード数

シミュレーションを用いて，エクスターナルグリッドに依頼された処理が完了するまでに利用された処理ノードの総数の平均を求めた．シミュレーションによって得られた結果が，図 1，図 2 である．ただし，プログラム分割数 100，多重度は 10 とした．

図 1 は，悪人ノードが全員正しく，同じ結果を返した場合の，エクスターナルグリッドの処理ノードの総利用数である．同様に，図 2 は，悪人ノードが全て互いに異なりかつ，誤った結果を返した場合である．

図 1 からわかるように，悪人ノードが全て正しい結果を返す場合，被保護断片数によって異なるものの，悪人の存在確率に関わらず，処理ノードの総数は一定である．図 2 のように，悪人ノードが全て互いに異なる結果を返した場合，悪人の存在確率が増加するに従い，処理ノードの総数が徐々に増加していることが確認できる．

一方で，図 1 と図 2 の何れの場合でも，被保護断片数が 5 の場合の方が，被保護断片数が 0 の場合と比べて，処理

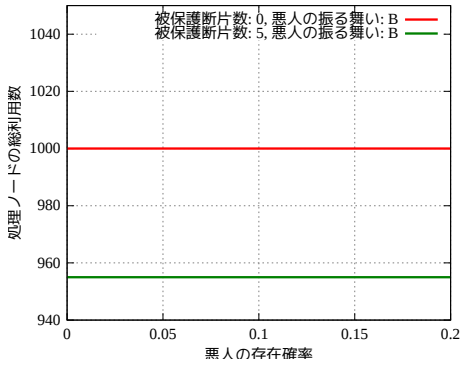

図 1 悪人の存在確率と処理ノードの総利用数
(悪人ノードが全て同一結果を返す場合)

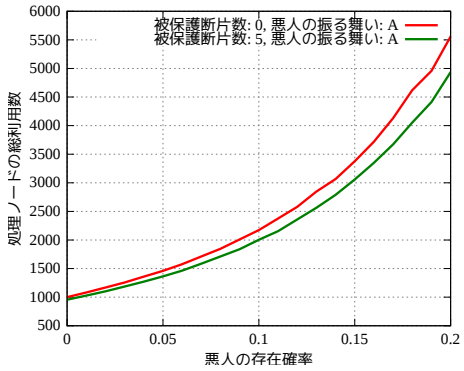

図 2 悪人の存在確率と処理ノードの総利用数
(悪人ノードが全て異なる結果を返す場合)

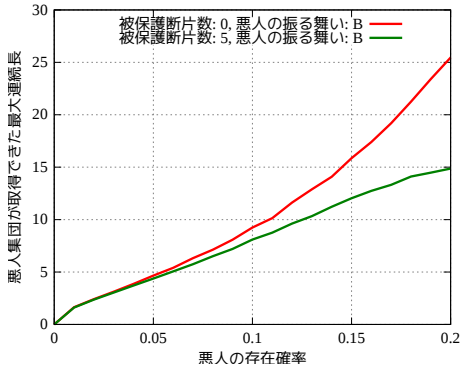

図 3 悪人の存在確率と悪人集団が得た連続長の最大値
(悪人ノードが全て同一結果を返す場合)

ノードの利用数が低減できていると分かる．処理の一部に保護処理を施した場合，悪人の存在確率や悪人ノードの振る舞いによらず，処理ノードの総利用数の低減が行えることが確認できた．

5.2 悪人集団による情報取得に対する耐性の評価結果

5.2.1 悪人集団が取得できた最大連続長

悪人集団が取得する最大連続長の平均を求めた結果を図 3，図 4，図 5 に示す．このとき，プログラム分割数は 100，多重度は 10 とした．

図 3 は，悪人ノードが全員正しく，同じ結果を返した場合の，悪人集団が実際に取得できた最大連続長の平均である．同様に，図 4 は，悪人ノードが全て互いに異なりかつ，誤った結果を返した場合である．そして，図 5 は図 3,4 の二つのグラフを合わせたものである．

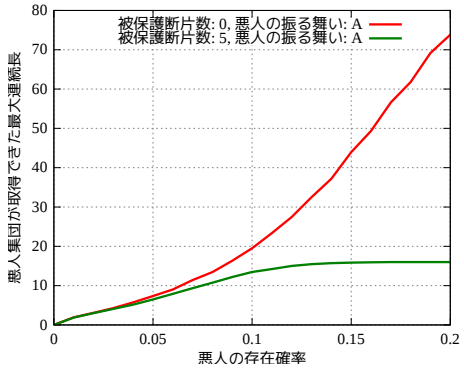

図 4 悪人の存在確率と悪人集団が得た連続長の最大値
(悪人ノードが全て異なる結果を返す場合)

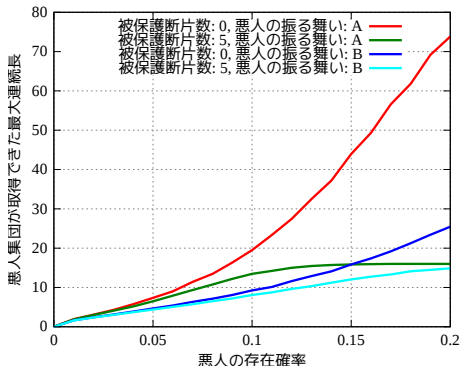

図 5 悪人の存在確率と悪人集団が得た連続長の最大値

図 3，4 より，処理の一部に保護処理を行うことで，悪人の存在確率が増加しても，悪人集団が取得できる最大連続長の増加を抑制，制限できていることが確認できる．

図 5 から，悪人の振る舞いの違いによって，悪人集団が取得できる最大連続長に違いが現れていることがわかる．この原因は，エクスターナルグリッドを構成しているノード中に存在している悪人ノードの台数が異なるためである．悪人集団が全員正しい実行結果を返すという振る舞いをした場合，網羅法を用いていても，依頼された処理を完了するまでに利用する処理ノード数は一定である．一方で，悪人集団が誤った結果を出し，互いに異なる実行結果を返す場合には，網羅法を用いているエクスターナルグリッドを構成するノードの総数は著しく増加する．ノード数の増加は，エクスターナルグリッドを構成するノード群により多くの悪人ノードを取り込むことになり，悪人集団が取得できるプログラム断片の最大連続長が大きくなる．そのため，悪人ノードが全員正しい実行結果を返す振る舞いをした場合には，悪人集団はエクスターナルグリッドに依頼される 1 回の処理中で取得できる最大連続長を犠牲にすることになる．

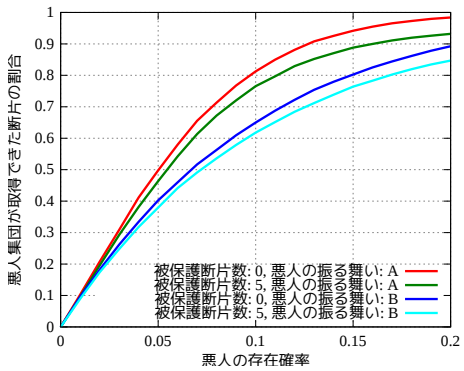

図 6　悪人の存在確率と悪人集団が取得した断片の割合

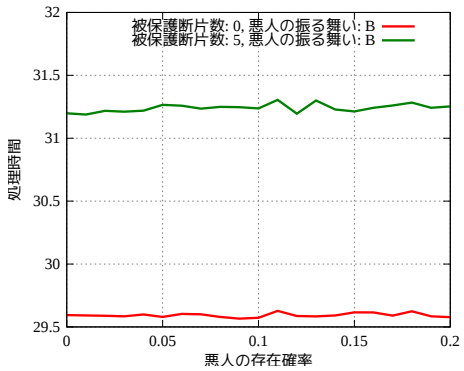

図 7　悪人の存在確率と処理時間
(悪人ノードが全て同一結果を返す場合)

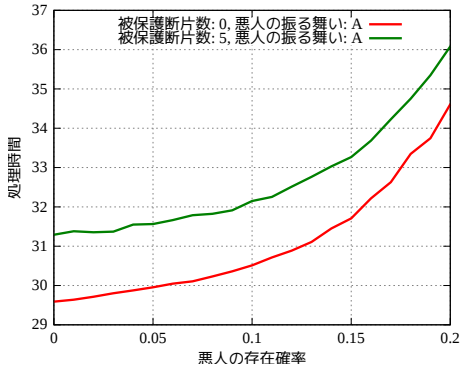

図 8　悪人の存在確率と処理時間
(悪人ノードが全て異なる結果を返す場合)

5.2.2　悪人集団に取得されたプログラム断片の割合

連続したプログラム断片の総数に対して，悪人集団が取得できたプログラム断片割合を求めた．連続したプログラム断片の総数に対して，悪人集団が取得できたプログラム断片の割合を求めることで，悪人集団が取得できた情報量が全体のどの程度かがわかる．

図 6 は，悪人ノードが全員正しく同じ結果を返した場合と，悪人ノードが全員誤った結果を返し，互いに異なる結果を返した場合の，それぞれの条件下で悪人集団が実際に取得できたプログラム断片の割合を表している．

図 6 において，同じ振る舞い同士のグラフを比較したとき，保護処理を施している場合，保護処理を施していない場合に比べて，悪人集団に取得されたプログラム断片の割合が少なくなっていることがわかる．また，同じ被保護断片数同士のグラフを比較したとき，悪人ノードが全て正しい結果を返すといった振る舞いをとった場合の方が，取得されたプログラム断片の割合が小さいことがわかる．例えば，悪人の存在確率が 0.1 である場合，被保護断片数が 0 であるもの同士を比較した際，悪人ノードが全て正しい結果を返すといった振る舞いを取った場合の方が，約 10%程悪人集団に取得されたプログラム断片の割合が小さい．

5.3　処理性能の評価

エクスターナルグリッドに依頼される処理を完了するまでの時間を処理時間と呼び，処理時間をエクスターナルグリッドの処理性能とする．処理時間を求めた結果を，図 7 と図 8 に示す．プログラム分割数は 100，多重度は 10 とした．

図 7 は，悪人ノードが全員正しく，同じ結果を返した場合の，エクスターナルグリッドの処理時間である．同様に，図 8 は，悪人ノードが全て互いに異なりかつ，誤った結果を返した場合である．

5.3.1　悪人の存在確率と処理性能

図 7 からわかるように，悪人ノードが全て同一結果を返した場合，保護処理の有無に関わらず，悪人の存在確率が増加しても，処理時間は概ね一定である．このような結果が出るのは，悪人ノードが全て正しく同一の結果を返す場合には，票割れが起きないためである．

また，図 8 からわかるように，悪人ノードが全て異なる結果を返す場合，悪人の存在確率が増加するに従い，処理時間が増加する傾向にあることが確認できる．例えば，被保護断片数が 0 の時の処理時間が約 29.59 であるのに対し，存在確率が 0.2 の時には，約 34.61 と，処理時間が約 17%程増加している．

処理時間が増加する原因は，各ブロックの投票時における票割れの回数の増加が考えられる．網羅法を用いたエクスターナルグリッドでは，ブロック内の実行結果で投票を行う際に，過半数に達する実行結果が一つも無い場合には，票割れを引き起こす．票割れが起きた際には，それまで先行していた処理が取り消されて再処理が行われる．各悪人ノードは，全員誤った実行結果を返し，ブロック内の悪人ノードは互いに異なった実行結果を返す場合，悪人の存在確率の増加に従って票割れの回数が多くなり，処理の完了時刻が遅くなる．

5.3.2　被保護断片の有無と処理時間

悪人ノードが全て同一の結果を返す場合（図 7），悪人ノードが全て異なる結果を返す場合（図 8）のいずれにおいても，被保護断片を設けた方が，処理時間が増加している．これは，被保護断片が入る事により，先行処理におけ

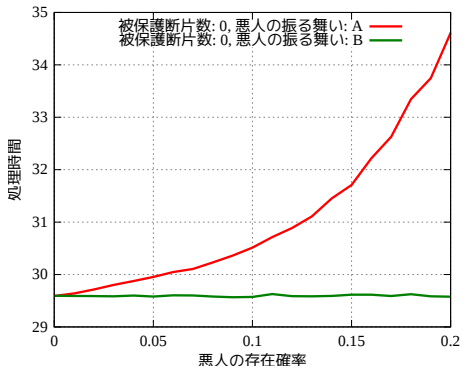

図 9　悪人の振る舞いが異なる場合の悪人の存在確率と処理時間 (被保護断片数 0 個の場合)

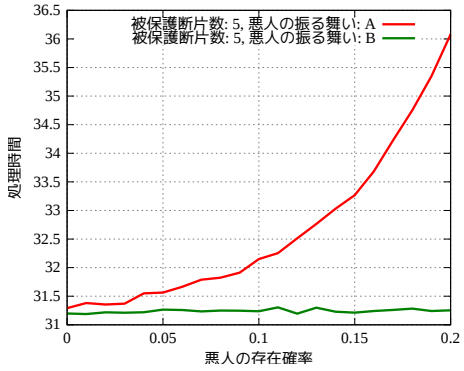

図 10　悪人の振る舞いが異なる場合の悪人の存在確率と処理時間 (被保護断片数 5 個の場合)

る処理性能向上の原因である，処理性能の高いノードを投機的に確保をする効果を活かせないからである．

5.3.3　悪人集団の目論見と処理時間

悪人集団が全て同一の結果を返す場合と，全て異なる結果を返す場合を比較する．図 9 と図 10 は，それぞれ，被保護断片が 0 の場合と 5 の場合の，悪人の異なる 2 つの目論見における，処理時間をプロットしたグラフである．

何れのグラフからも分かる様に，悪人が全て同一の結果を返す場合は，存在確率にかかわらず，処理時間はほぼ一定である．一方，悪人が全て異なる結果を返す場合は，悪人の存在確率が増加すると，処理時間が増加する．これは，5.3.2 節でも述べたように，悪人が全て異なる結果を返すことは票割れを誘因し，その結果として，再処理の発生を増やすことが原因である．

また，図 9 と図 10 のからわかるように，悪人の存在確率が 0 の場合，悪人の振る舞いによる処理時間の差は，殆ど見られない．悪人の存在確率が 0 ということは，悪人ノードが全て異なる結果を返す場合でも，正しくない実行結果を返す悪人ノードが存在しないということなので，網羅法による処理の分岐や，票割れによる再処理が発生しない．同様に，悪人ノードが全て正しい実行結果を返すという振る舞いを取る場合においても，全ての処理ノードが常に正しい実行結果を返すので，処理の分岐や票割れによる再処理が発生しない．このような理由から，悪人の存在確率が 0 の場合，悪人の振る舞いが異なることによる処理時間の差が発生しないと考えられる．

6.　結論

本稿では，エクスターナルグリッドを標的とする悪人集団が，どのような目的を持ってエクスターナルグリッドに対して振る舞うかを議論した．また，2 種類の悪人ノードの振る舞いを条件として設定したエクスターナルグリッドのシミュレーションを行い，網羅法を用いたエクスターナルグリッドの性能と，処理の一部に保護処理を施す手法の性能評価を行った．

悪人ノードが全て正しい結果を返す場合，悪人ノードが全て異なる結果を返す場合と比較して，悪人集団が取得できる連続長やプログラム断片数が少ないということが示された．また，悪人が全て正しい結果を返す場合，網羅法による処理の分岐や票割れによる再処理が発生しないので，悪人の存在確率が変化してもエクスターナルグリッドの処理性能は殆ど変化しなかった．以上より，悪人ノードが全て正しい結果を返すという悪人ノードの振る舞いは，悪人集団が 1 回のエクスターナルグリッドへの参加での情報収集の効率を下げ，また，クライアントの活動に対して妨害することもできない．

一方，悪人ノードが全て異なる結果を返す場合，悪人ノードが全て正しい結果を返す場合と比べ，悪人集団が取得できる情報量が著しく多いことがわかった．例えば，プログラム分割数が 100，多重度が 10，悪人の存在確率が 0.2，被保護断片数が 0 である場合，この悪人ノードの振る舞いと他方の悪人ノードの振る舞いを比較したとき，悪人集団が取得できた最大連続長に 50 程の差が出ている．また，エクスターナルグリッド上の処理ノードの総数や処理性能は，悪人の存在確率の増加に伴い悪化するという結果が出た．悪人ノードが全て異なる結果を返す振る舞いを取る場合，悪人にとっては，悪人集団が 1 回のエクスターナルグリッドへの参加での情報収集の効率が高くなる．同時に，エクスターナルグリッドの最終的な処理時間を遅延させるといった効果があることもわかった．

処理の一部に保護処理を施す手法は，以上の 2 種類の悪人の振る舞いそれぞれの条件下にあるエクスターナルグリッドにおいて，悪人集団が取得できる情報量の制限・抑制に一定の効果があることが示せた．例えば，プログラム分割数が 100，多重度が 10，悪人の存在確率が 0.2，悪人ノードが全て正しい結果を返すといった振る舞いを条件とするエクスターナルグリッドにおいて，被保護断片数が 0 の場合と被保護断片数が 5 の場合を比較すると，悪人集団が取得できた最大連続長は，前者に比べて後者は 2/3 に抑えられていることがわかる．しかし，悪人ノードの振る舞いによらず，処理の一部に保護処理を施す手法を取った場

合の方が，保護処理を施さない場合と比較して，処理時間が増加することも確認できた．

以上より，今回設定した 2 種類の悪人集団の振る舞いを条件とするエクスターナルグリッド上では，処理の一部に保護処理を施すことで，処理時間の増加が見られる一方で，悪人集団による不正な解析や悪人集団による情報収集のリスクを軽減することを示せた．

謝辞

本研究は JSPS 科研費 26234567 の助成を受けたものです．

参考文献

[1] SETI@Home ⟨http://setiathome.ssl.berkeley.edu/⟩（参照 2016-04-28）.

[2] 広瀬 吉隆，稲元 勉，樋上 喜信，小林 真也: “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”，第 14 回情報科学技術フォーラム (FIT2015) 講演論文集，Vol．4，pp．241-242，2015.

[3] 山口 晃右，稲元 勉，樋上 喜信，小林 真也: “エクスターナルグリッドに対する依存関係を利用した不正解析のリスクを軽減する手法”，情報処理学会第 78 回全国大会論文集，5660，2016.