完全準同型暗号を用いたゲノム秘匿検索の並列分散処理による高速化

山本百合　（指導教員：小口正人）

1　はじめに

バイオインフォマティクスの研究において，研究機関や各病院が所持するゲノムデータの活用が求められている．そのため，大量のゲノムデータをクラウド事業者などに委託し，利用者が問い合わせを行うことで統計処理が可能なゲノムデータ委託システムが今後広がっていくと考えられる．しかしヒトゲノムは個人の識別子となるため，プライバシ保護の観点から，暗号を適用した秘匿検索手法によるデータ活用が必要となる．

先行研究 [1] では，暗号文同士の加法と乗法が成立する完全準同型暗号を秘匿検索に適用し，復号することなく統計処理などの複雑な演算が可能な手法を目指し，アルゴリズムの高速化を進めている．しかしながら，完全準同型暗号演算の計算量が大きいために，サーバ側での計算負荷が大きくなりやすい．本研究では，先行研究が用いる手法のサーバ側での演算に対してマスタ・ワーカ型の分散処理を適用することで高速化を行い，クラウドコンピューティングへの適用可能な分散処理システムの実装を目指す．

2　完全準同型暗号

完全準同型暗号とは式 (1)，(2) のように暗号文同士の加算と乗算の演算が成立する性質を持ち，暗号化した状態で平文と同様の多項式演算が可能な暗号である．完全準同型暗号は公開鍵暗号方式の機能を持つが，秘密鍵を用いることなく暗号文同士の演算から平文同士の演算を暗号化した値を導くことが可能となる．そのため，ゲノム秘匿検索に完全準同型暗号を適用することで，秘密鍵を渡すことなくサーバがデータ同士の統計処理を行えると期待できる．また完全準同型暗号の概念自体は，1970 年代後半に公開鍵暗号が考案された当初より提唱され，2009 年に Gentry が実現する手法を提案した．提案当時は計算量の大きさから実用性が乏しかったが，その後も様々な研究によって高速化や改良が進められ，簡単な計算であれば十分な性能レベルを示している．しかし，暗号の解読困難性を保つためのノイズの付加と解読時のノイズの除去のための工夫の必要性などから，依然として複雑な計算や大きなデータに対する演算では，暗号文サイズが大きくなる傾向にあり，計算量がかかる難点を持つ．

完全準同型暗号

$$Encrypt(m) \oplus Encrypt(n) = Encrypt(m+n) \quad (1)$$

$$Encrypt(m) \otimes Encrypt(n) = Encrypt(m \times n) \quad (2)$$

3　提案手法

本研究では，以下の完全準同型暗号を用いたゲノム秘匿検索のマスタ・ワーカ型の分散システムを提案する．提案システムの概要を図 1 に示す．

(1) クライアントはクエリの 1 文字を暗号化し，公開鍵と共にマスタへ送信する．

(2) マスタは受け取ったデータを各ワーカに転送する．

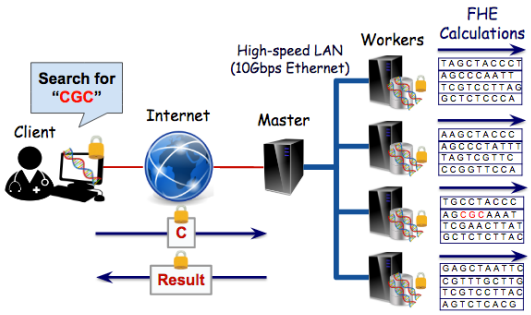

図 1: 提案手法概観

(3) ワーカは完全準同型暗号を用いた演算を行い，結果をマスタへ転送する．

(4) マスタは結果を収集し，クライアントへ結果を送信する．

(5) クライアントは復号を行い，結果を得る．また，その結果を用いてクエリの次の 1 文字を暗号化した後に再びマスタに送信する．

(6) (2)〜(5) をクエリの長さの回数分，繰り返して終了する．

以上のプロトコルを C++で実装した．また，完全準同型暗号計算には HElib[2] を，分散化における各マシンの制御のために MPI(Message Passing Interface) を利用した．

4　分散方法

今回適用するアプリケーションの分散化手法として，個体のデータごとでデータベースを分割するデータの分散，独立性の高い計算部位に適用する分散処理，また独立性の高い手順に適用する分散処理などが考えられる．先行研究の秘匿計算手法は，クエリとデータベースの要素に対して完全準同型暗号演算を行う手法を適用している．そのため，分散化した各ワーカマシンにデータベースを設置することにより，クエリとデータベースの要素間の暗号演算が可能となり，より多くのクエリとデータベース間のマッチング有無を調べることが可能となる．したがって今回の提案システムでは，もっともシンプルなデータベースの分割による分散処理を適用した．

5　実験

5.1　実験環境

実装したプログラムを複数のマシンにおいて実行した．各マシンの環境は，Intel®Xeon ®Processor E5-2643 v3 3.4GHz，6 コア，12 スレッド，メモリ容量 512GB，ストレージは RAID0 の SSD が 480GB，HDD が 2TB であり，同スペックのマシンを 4 台使用する．1 台をマスタの機能を持ったマシンとし，同時にワーカとして 1 スロット分の演算も行う．また他 3 台をワーカとして最大 2 スロット稼働させた．最大で 7 スロット分のワーカを稼働させてワーカ数ごとの実行時間を

比較する実験を行った．実験に使用するゲノムデータは 1 サンプルあたり 10,000 文字のデータを 2,184 サンプル用意した．また，検索クエリは長さ 5 文字のものを用いた．さらに秘匿検索の秘匿性を高めるダミー検索を加えることにより，データベース上の 50 箇所を始点とした文字列検索を行った．

5.2 ワーカ数ごとのマスタとクライアントにおける実行時間の評価

クエリとデータベースとのマッチングの有無の判定を行うプログラムを分散化した環境上で稼働させた．ワーカ数ごとのマスタとクライアントの実行時間のグラフをそれぞれ図 2，図 3 に示す．

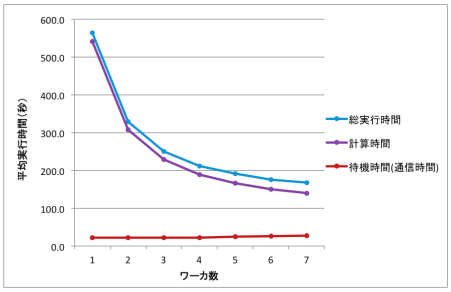

図 2: ワーカ数ごとのマスタにおける実行時間 (秒)

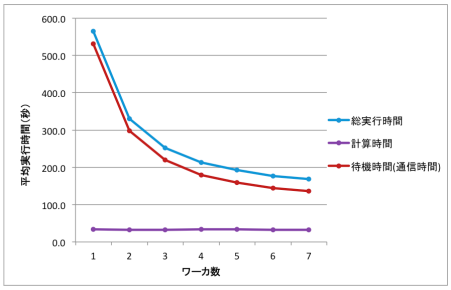

図 3: ワーカ数ごとのクライアントにおける実行時間 (秒)

マスタ側の実行時間において，ワーカ数が増加するにつれて計算時間を減少させることができた．しかし計算手順の中にデータベースの大きさに依存しない計算が含まれることから，ワーカ数が増えるにつれて計算時間における分散化効果は徐々に横ばいになっていることがわかる．またワーカ数の増加に対して，クライアントにおける復号計算時間も含めた通信時間はほとんど変化しないため，クライアントとの相互通信におけるオーバヘッドは小さい．

クライアント側の実行時間は，現在マスタ側の計算が終了するまでクライアントは待機する仕様となっているため，マスタでの計算時間の分が待機時間としてかかるが，クライアント自身の復号計算時間はワーカ数による大きな変化は見られなかった．

5.3 ポジション数ごとの分散効率の評価

秘匿性を高めるダミーポジションの付加によって検索ポジションの総数を変化させた実験を行った．ポジション数とワーカ数における分散処理の高速化率を比較した結果を図 4 に示す．また分散化の評価となる高速化率は，式 (3) より算出した．

高速化率 = 逐次実行時間 (秒)/並列実行時間 (秒)　(3)

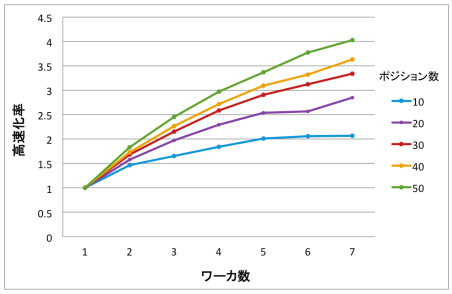

図 4: ポジション数におけるワーカ数ごとの高速化率

ワーカ数とポジション数の増加に伴い高速化率が上昇することが図 4 から読み取れる．先行研究はゲノム秘匿検索の高速化のために様々な手法が用いている．クエリの検索はデータベースを隈なく検索するのではなく，クエリの長さの分のみデータベースと演算することで最長マッチ数を算出する高速化手法が行われている．そのためポジション数が少ない実験においては，前処理などの並列分散化できない演算の影響が大きく，分散効率が抑えられてしまう．しかしポジション数が多い実験においては，データベース上の検索を開始する箇所であるポジションがデータベース上に散らばって演算範囲が広がるため，データベースによる分散化の効果が現れやすくなったと考えられる．

今後は秘匿検索システムの高速化を進めると同時に，今回実装したデータベースの分割による分散処理手法による効果が期待できるゲノム統計手法に対する本手法の適用を考えたい．

6 まとめと今後の課題

完全準同型暗号を用いたクライアント・サーバ型ゲノム秘匿計算のサーバ側の処理に，マスタ・ワーカ型の分散処理システムを適用し実験を行った．今回は分散処理方法としてデータベースの分割を適用した．その結果マスタ側の計算時間が分散台数に応じて減少し，通信時間も含めた待機時間はほぼ変わらない結果となった．また高速化率による評価を行った結果，検索のデータベース上の開始点を示すポジション数が多い方が高速化率が高いことが判明した．これはポジションが増加するほどデータベースの全体と演算を行うことになるため，データベースによる分散の効果が現れやすいからだと考えられる．

今後は実装を改善し，より高速な分散処理が可能な手法や本手法が効果的に作用するゲノム統計処理への適用を提案していきたい．

謝辞

本研究を進めるにあたり，大変有益なアドバイスを頂いた早稲田大学山名研究室の皆様に感謝いたします．

また本研究は，JST CREST の支援を受けております．

参考文献

[1] 石巻優, 清水佳奈, 縫田光司, 山名早人 :「完全準同型暗号を用いた高速なゲノム秘匿検索」, SCIS 2016, 2A2-2, 2016 年 1 月

[2] Shoup V. and Halevi S. : http://shaih.github.io/HElib/index.html, 2017 年 1 月閲覧