完全準同型暗号アプリケーション実行時の課題に対する高性能 SSD 適用手法の評価

廣江 彩乃　（指導教員：小口 正人）

1　はじめに

近年ゲノムデータなどの秘匿情報を活用する取り組みが増えている．これらのデータの処理を外部のサーバに委託する場合，セキュリティの観点から，完全準同型暗号を用いるなどして暗号化したまま処理することができることが望ましい．しかしこの場合，暗号化処理の計算量が多くなるため，実用上に向かない計算時間がかかってしまう．暗号化の際の計算量に加えて，ゲノムデータなどの大きなデータを扱う処理でメインメモリが不足してストレージへのアクセスが発生することも，実行時間が長くなる要因である．ストレージへのアクセス速度はメインメモリへのアクセスに比べて格段に遅いためである．また，完全準同型暗号方式を用いると暗号化データが元の数万倍のデータ量になるため，大概メインメモリが不足するが，メモリが高価であることを考慮すると，実行時間の課題に加えてコストの面にも課題がある．

そこで，実行効率とコストの課題を解決するため，近年高速化に向けて研究開発が進む，高性能で比較的安価な SSD の有効利用を検討する．本研究では，完全準同型暗号を用いたゲノムの秘匿検索アプリケーション [1] を用いて，暗号化アプリケーションの実行時負荷を計測・検証する．

2　先行研究

石巻ら (2016) の先行研究によるゲノム秘匿検索アプリケーション [1] を，本研究のメモリ性能評価に用いる．図 1 にアプリケーションの流れを示す．これはサーバとクライアントが 1:1 で問い合わせを行うものである．また，ゲノムデータは A,C,G,T の四文字から成る配列であるため，文字列検索と見なせる．サーバはクライアントから，検索したい文字列を暗号化処理したものと，その文字列を検索したい配列上の検索開始地点を受け取り，秘匿検索を行ってマッチしたか否かの結果を返す．クライアントから送られる暗号化された文字列をクエリ，文字列検索開始地点をポジションと呼ぶ．

3　実験

実験を大きく分けて 2 つ行った．1 つ目は，ゲノム秘匿検索アプリケーション [1] の調査である．2 つ目は，本アプリケーションを用いた，SSD の性能比較実験である．

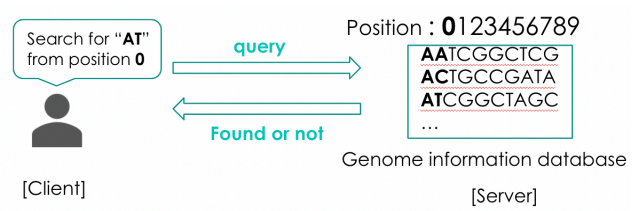

図 1: アプリケーションの流れ

表 1: サーバ性能

CPU	Intel®Xeon®Processor 6 Cores × 2 Sockets
DRAM	DDR4 512GB 2133MT/s
HDD	HGST SATA 2TB

3.1　実験 1

3.1.1　実験概要

サーバクライアント型暗号化アプリケーションの課題は，秘匿検索を行うサーバ側の負荷が大きいことである．そこで perf コマンドを用いて，本アプリケーション実行時のサーバ側のカーネル処理の内訳を調べた．用いたサーバのスペックは表 1 に示した．

3.1.2　実験結果

処理全体の 8 割以上が推論を行うための NTL という数学ライブラリによるものであった．このことから，本アプリケーションは演算処理が最も大きな処理であることが分かった．また，1 クロックあたりに実行される命令数を IPC という値を計測した．その結果，IPC は 2.79 と高い値となった．CPU 効率が高いという結果からも，演算処理が本アプリケーションのボトルネックであることが分かった．

3.2　実験 2

3.2.1　実験概要

実験 2 ではゲノム秘匿検索アプリケーションを用いて，コンピュータリソースへの負荷やメモリの使用量を計測する．そして，実行環境の差による分析を行う．

プログラム実行に必要なメインメモリの量が容量を超える際には，ストレージに領域を確保する必要がある．この場合において，メインメモリに比べると圧倒的に遅いストレージへのアクセスが，大きなデータを扱う暗号化アプリケーションの実行時間にどの程度影響するか調べたい．ストレージの性能を見る上で，まず swap 処理に着目する．実験の都合上小さいテストデータを用いる本研究では，Docker コンテナを用いて，使用可能メモリが不足する状況を作る．そして，コンテナが使用可能なメインメモリを制限して，メインメモリの外の swap 領域へのアクセスを発生させる．さらに，その swap 先デバイスに高性能 SSD を指定して，アクセス速度の差を検証していく．

用いたサーバのスペックは表 2 に示した．このサーバ上に，十分なメインメモリを割り当てた Docker コンテナと，swap 処理を発生させるためにメインメモリを 1GB のみ搭載した Docker コンテナを構築した．

表 2: サーバ性能

CPU	Intel®Xeon®Silver 4313 16 Cores
DRAM	DDR4-2667 16GB*12=192GB 3200MHz
不揮発メモリ	Intel Optane 200 Series DDR-T 256GB
PCIe Gen	4.0
OS	Ubuntu 18.04.6 LTS
Docker version	20.10.7
Docker Container OS	CentOS Linux release 8.4.2105
GNU libc version	2.28

表 3: 使用した 4 種類の SSD

Intel Optane SSD 905P	Samsung 980 PRO	Kioxia EXCERIA PLUS SSD	INTEL SC2KB 019T8 (SATA)

3.2.2 高性能記憶装置の比較

swap デバイスに不揮発メモリや高性能な SSD を用いた場合の性能評価を行っていく．比較対象とする高性能 SSD は 4 種類である．それらの製品の名称をまとめたのが表 3 である．不揮発メモリと表 3 の 4 種類の SSD を swap 先として指定して実行する他に，実行に十分なメモリを Docker コンテナに割り当てることで swap 処理を発生させない条件を加えて，以下の 6 種類の条件で計測していく．

(1) DRAM 上で処理が完結する場合
(2) swap 処理により，不揮発メモリへのアクセスが発生する場合
(3) swap 処理により，Intel Optane SSD へのアクセスが発生する場合
(4) swap 処理により，Kioxia SSD へのアクセスが発生する場合
(5) swap 処理により，Samsung SSD へのアクセスが発生する場合
(6) swap 処理により，Intel SATA 接続 SSD へのアクセスが発生する場合

3.2.3 実験結果及び考察

条件 (1)-(3) と条件 (3),(4) の実行時間をそれぞれ図 2，図 3 に示す．まず条件 (1)-(3) を比較した結果から，実行パラメータであるクエリ長が長くなっても，不揮発メモリを用いた条件 (2) と SSD を用いた条件 (3) がほぼ同じ実行時間であることがわかる．不揮発メモリはメインメモリと同じスロットに挿すため，SSD とは一般的に桁違いの読み書き速度を持つが，今回用いたアプリケーション実行に際しては差が見られなかった．

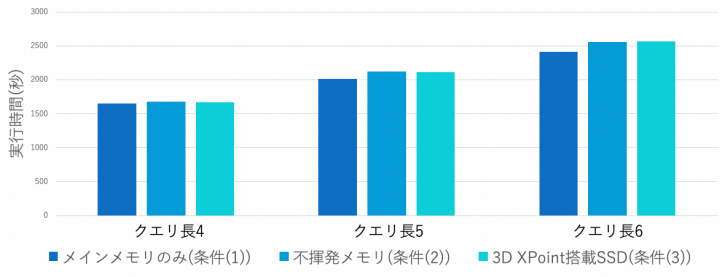

図 2: 実行時間比較

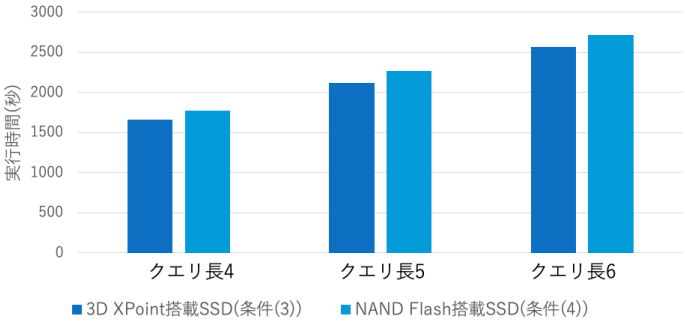

図 3: 実行時間比較

この両者は記憶素子には同じデバイスを用いており，インタフェースが異なるが，今回はインタフェースの性能差が実行結果に影響を与えず，記憶素子のアクセス速度で実行結果が決まったと考えられる．よって，用いる際には費用対効果の有無を検証する必要があることがわかる．次に，異なる種類のメモリを搭載している SSD を用いた条件 (3),(4) の結果について考察する．ここで用いた SSD には，条件 3 のものは 3D XPoint メモリが，条件 4 のものは NAND Flash メモリが搭載されている．前者の方がレイテンシが短いという特徴がある．パラメータのクエリ長が長くなるにつれて実行時間の差の開き方も変わっていることから，処理内容が増えてメインメモリが多く使われるようになるほど，swap 処理が増えて swap 領域の SSD が使われ，SSD のレイテンシ性能の差が表れることがわかる．

4 まとめと今後の課題

本研究では，演算処理が主な完全準同型暗号アプリケーションに対して様々な性質を持つ記憶装置を用いて実験を行った．実行時間短縮に寄与する性質がわかったことから，この結果を用いて費用対効果のある装置の活用方法を模索することが今後の課題である．

謝辞

本研究の一部はお茶の水女子大学とキオクシア株式会社との奨励研究契約に基づくものである．また，本研究にご協力頂いたキオクシアの圓戸辰郎氏に深謝する．

参考文献

[1] Y. Ishimaki et al.,“Privacy-preserving string search for genome sequences with FHE bootstrapping optimization.”2016 IEEE International Conference on Big Data (Big Data). IEEE. 2016, pp. 3989–3991.