

高速大規模 I/O が可能なファイルシステムによる ビッグデータ統合利用基盤の性能向上

藤島 永太^{†1} 中島 健司^{†2} 山口 実靖^{†3}

工学院大学大学院 工学研究科 電気・電子工学専攻^{††}

1. はじめに

近年, IT 技術の進歩に伴い, 地球上で生成されるデータ量が指数関数的に増加しており, それらのデータから有用な情報を収集・蓄積・分析し, 活用することに注目が集まっている. ただし, 医療・ライフサイエンスに代表されるセンシティブなデータを扱うセキュアなコンテンツ共有・流通基盤では, 暗号技術の併用が欠かせない. この場合, 大規模なデータを全て暗号化すると, データ量が大幅に増加し, 実用的な処理時間を実現するためには大規模データ処理の I/O 速度の向上が重要となる.

本稿では, 大規模なデータを格納する HDD の特性に着目し, 巨大な I/O に対して有効であると考えられる大きなブロックサイズでファイルの管理を行うファイルシステムを構築し, 性能評価によりその有効性を示す.

2. ゲノム配列の秘匿検索アプリケーション

バイオインフォマティクス分野の研究では, ゲノム配列の検索が頻繁に行われる. しかし, 個人ゲノムの利用はプライバシー保護への対策が必須であり, 暗号化したままでゲノム配列の検索処理を行う必要がある. これを実現するための従来研究では, PBWT 離散データ構造と準同型暗号を組み合わせた手法が提案されている. 本稿では, PBWT 離散データ構造と完全準同型暗号を用いたゲノム秘匿検索手法 [1] に焦点を当て, そのアプリケーションの性能について考察を行う.

当該手法を実装したアプリケーションは, コホート研究で得られるデータの利用など, クエリとデータベースの両方にセンシティブな情報が含まれている場合を想定している. このモデルでは, クライアント(検索者)はサーバ(データ保有者)に対してクエリ内容を秘匿し, サーバはクライアントに対してクエリの検索結果以外の情報を秘匿する. 当アプリケーションの現在の実装では, サ

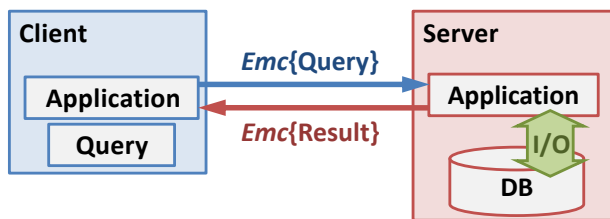


図 1. ゲノム配列検索の概略図

Performance Improvement of Platform for Integrated Big Data Utilization with a Fast Filesystem Specialized for Fast Large Scale I/O Processes

^{†1}. Eita Fujishima.

^{†2}. Kenji Nakashima.

^{†3}. Saneyasu Yamaguchi.

^{††}. Electrical Engineering and Electronics, Kogakuin University Graduate School

ーバは長さ N のゲノム配列が M 本登録されているデータベースを持つ. 性能評価では, 1000 ゲノムプロジェクト [2] における SNP 配列 2,184 サンプルを持つデータベースを使用している.

3. ディスク I/O 解析

アプリケーション実行中のディスク I/O を解析するために, Linux カーネルを改変し, SCSI 層における I/O 要求の内容を調査した. 測定環境は, 物理計算機 1 台でサーバとクライアントを並列に動作させ, データベースの容量は 2 GB とした. 測定に使用した物理計算機の仕様は表 1 の通りであり, データベースを格納する HDD の仕様は表 2 の通りである.

表 1. 測定用物理計算機の仕様

CPU	Intel Core i5-4670 3.40 GHz
OS	CentOS 7 x86_64 minimal
Kernel	Linux-3.10.103 (modified)
SSD	512 GB (ext4)
HDD	1 TB (ext3)
Main Memory	16 GB

表 2. 測定用 HDD の仕様

型番	WD1003FZEX
インタフェース	SATA 3.0 (6.0 Gbps)
容量	1 TB
バッファ容量	64MB
回転数	7,200 rpm

測定結果を図 2 に示す. 横軸はアプリケーション実行中の経過時間 [sec], 縦軸はディスク内のアドレス [GB] を示している. 経過時間 10~22 秒の時間帯で, サーバにおけるデータベースの読込処理が行われている. この

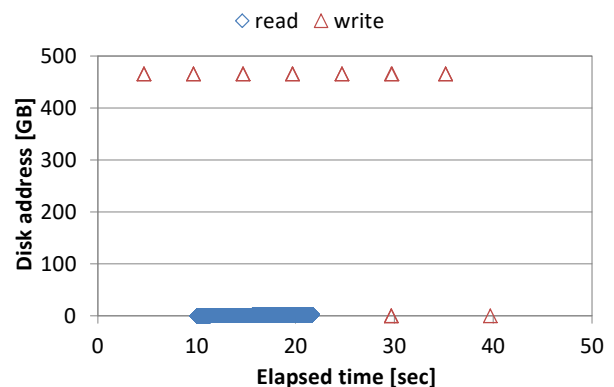


図 2. SCSI 層における I/O 要求

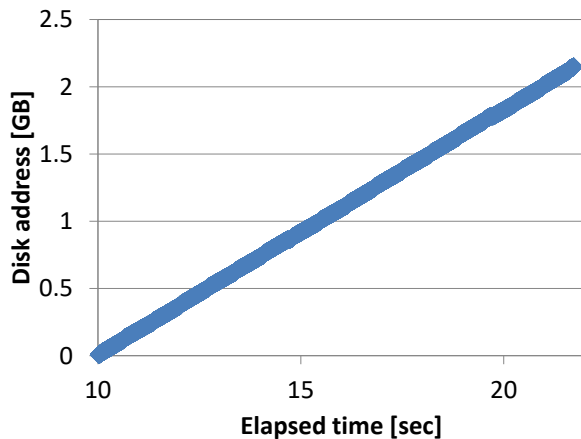


図 3. SCSI 層における I/O 要求 (拡大図)

部分のグラフを拡大したものを図 3 に示す。この結果から、当該アプリケーションの I/O 性能を向上させるためには、巨大なファイルに対する連続的な読み込み性能を向上させる必要があることが分かる。

4. 提案手法

前章のディスク I/O 解析の結果より、アプリケーションの I/O 性能を向上させるためには巨大なファイルに対するシーケンシャル I/O 性能を向上させる必要があることが分かった。そこで、巨大なブロックサイズでファイルの管理を行うファイルシステムを構築し、そのファイルシステム上にデータベースを格納することにより、I/O 性能を向上させる。手法の詳細については、以下の節で述べる。

4.1. 手法の概要

本手法は、我々の過去の研究 [3] で得られた知見に基づいて動作する。具体的には、定記録密度方式の HDD に対してシーケンシャルアクセスを行う場合、ディスク外周側の連続領域に対して行う方が、内周側に対して行うよりも高い I/O 性能が得られるという知見に基づく。また、連続性の高い I/O と連続性の低い I/O では、連続性の高い I/O の方が HDD の持つアクセス性能を活かせるため、巨大なブロックサイズでファイル管理を行う方式とした。

4.2. 手法の実装

本手法の実装は、FUSE (Filesystem in Userspace)を用いて行った。ユーザ空間上での実装であるが、ディスクへの I/O 処理はデバイススペシャルファイルに対して直接行い、各ファイルのディスク内のオフセット位置などのファイル管理情報は配列を用いてメモリ上で管理している。ファイルシステムのブロックサイズは 256 MB に設定しており、ディスク領域はこのブロック単位で分割される。そして、ディスク領域は全てファイルのデータ領域として使用することにより、連続性の高いディスクアクセスを実現している。シーケンシャルアクセス性能の高いディスク外周部を積極的に使用するために、新規ファイルの格納位置検索方法は、ディスクアドレスの低い方から順に空き領域の検索を行う仕様とした。

5. 性能評価

本章では、提案手法の性能評価について述べる。

5.1. 測定方法および環境

ゲノム配列検索アプリケーションのデータベースを格納するファイルシステムを Ext3 または提案手法としたときのデータベース読み込み時間を比較することにより性能評価を行った。アプリケーションはそれぞれ 10 回ずつ実行し、平均読み込み時間を比較対象とした。測定環境は、データベースの格納に用いるファイルシステムを変更した点以外は 3 章と同様である。

5.2. 測定結果

データベースの平均読み込み時間の測定結果を図 4 に示す。ファイルシステムに Ext3 を使用した場合の平均読み込み時間は 11.72 秒、提案手法を使用した場合の平均読み込み時間は 10.16 秒となり、Ext3 と比較して 13.3% の性能向上を確認した。

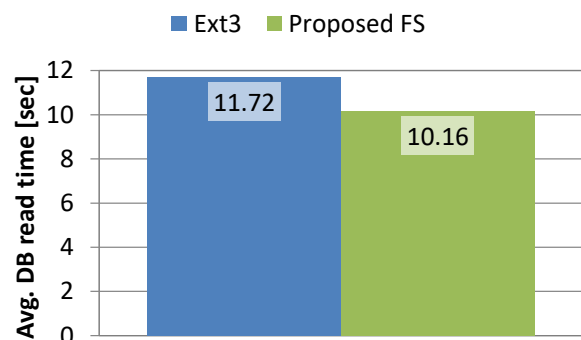


図 4. データベースの平均読み込み時間

6. おわりに

本稿では、セキュアなコンテンツ共有・流通基盤を発展させるために作成されたゲノム配列の秘匿検索アプリケーションの I/O 性能向上について考察し、巨大なブロックサイズでファイル管理を行うファイルシステムを提案し、性能評価によりその有効性を確認した。

今後の課題として、キャッシュに影響を及ぼす巨大な I/O に対して Direct I/O を適用することや、ユーザ空間で実装を行った当手法を、よりオーバーヘッドが小さいと考えられるカーネル空間で実装することなどが考えられる。

謝辞

本研究は、JSPS 科研費 25280022, 26730040, 15H02696 の助成を受けたものである。

本研究は、JST, CREST の支援を受けたものである。

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”, 2016 Symposium on Cryptography and Information Security (SCIS2016), (2016.1).
- [2] The 1000 Genome Project Consortium: “An integrated map of genetic variation from 1,092 human genomes”, Nature, Vol.491, pp.56-65, (2012).
- [3] Eita Fujishima, and Saneyasu Yamaguchi, “Dynamic File Placing Control for Improving the I/O Performance in the Reduce Phase of Hadoop,” 10th International Conference on Ubiquitous Information Management and Communication (ACM IMCOM’16). No.8-2. DOI=http://dx.doi.org/10.1145/2857546.2857595.