6-2　セキュリティインシデント解析の高速化を支援する分散ストレージ技術と並列 I/O 技術の動向

6-2 A Survey of Distributed Storage and Parallel I/O Technique for Security Incident Analysis

神坂紀久子
KAMISAKA Kikuko

要旨

近年、インターネットを介した不正アクセスやウィルスによる攻撃が問題となっており、サイバー攻撃を監視し、情報を収集して分析することでインシデントに対応することが多くなってきた。しかし、ネットワークが大規模化、高速化したことにより、収集、解析データ量が増大し、動的拡張性を持つ分散ストレージ技術や高速な並列 I/O 技術が必要となってきている。

そこで本稿では、インシデント解析を高速に実行するために必要な、分散ストレージ技術、並列 I/O 技術の動向について調査した。

Recently, unauthorized access and virus attack via open Internet are a big issue in many fields. With the spread of high-speed and large-scale network, a large amount of data is required for analysis of security incident information. Therefore, it is necessary to access and process these data with high-speed.

In this paper, we survey recent distributed storage and parallel I/O techniques for the high-speed incident analysis, and consider possible applications of these techniques in a research area of security.

[キーワード]
並列I/O，分散ストレージ，クラスタシステム，グリッドコンピューティング
Parallel I/O, Distributed storage, Cluster system, Grid computing

1　まえがき

近年、企業や組織のネットワークやインターネットサービスプロバイダにおいて、不正アクセスやコンピュータウィルスによるサイバー攻撃が問題になっている。インターネットを利用したこのような攻撃から防御する対策として、ファイアウォールや IDS（Intrusion Detection System）により、それらの侵入や攻撃を監視してインシデントを分析する方法がある。年々、不正アクセスやウィルス攻撃は高度化しており、トラフィック、ウィルス、ログ情報等を収集してリアルタイムに分析するためには、データ処理の高速化や効率化が重要となってきている。

しかし、一方で、ブロードバンドネットワークの普及、Gigabit/10 Gigabit の高速ネットワーク等の普及によって、企業や組織間で使用するネットワークインフラはますます複雑化、大規模化してきている。インターネット、イントラネットを流れるデータ量も急増しており、不正アクセスやウィルス攻撃の解析のために収集したデータ量も今後更に増大していくと考えられる。そのような大規模なデータを処理するためには、可用性や拡張性を考慮した高性能な演算能力と大規模な記憶領域を持つ計算機資源が必要である。

そこで本稿では、セキュリティインシデントの解析のために、大規模データを効率的かつ高速に処理するためのデータ基盤技術として、分散スト

レージや並列 I/O 技術に着目して調査を行った。

2 ストレージアーキテクチャ

本節では、トラフィックやログ情報などを格納する仕組みとして、まず現在普及しているストレージアーキテクチャについて述べる。

従前から、サーバに接続されるストレージの基本的な形態として、RAID システム、またはサーバに SCSI インタフェースなどで直接接続した DAS（Direct Attached Storage）が一般的に使用されてきた。しかし、DAS では、サーバごとに異なるストレージデバイスを使用する必要があり、管理が複雑化し、ストレージ容量を有効に活用できないという問題があった。その後、アーキテクチャ規模が年々拡大していることを背景に、効率的なデータ管理を実現できるストレージネットワーキング技術が発展し、ネットワークを介することで DAS の問題点を解消した SAN（Storage Area Network）や NAS（Network Attached Storage）が普及している。SAN は、サーバと複数のストレージデバイスを接続した高速な専用ネットワークのことであり、分散しているストレージを統合して、ディスク資源の有効利用が可能となる。高速かつ高価なファイバチャネルベースの FC プロトコルや安価な IP ネットワークベースの iSCSI プロトコルを使用し、ネットワークを介したストレージアクセスを行う。しかし、SAN はストレージデバイスに対してブロックレベルのアクセスを行うため、ファイル共有はサポートしておらず、ファイルを扱うためには上位層に何らかのファイルシステムが必要となる。一方、NAS は、ネットワーク上で計算機間のファイル共有をすることを目的としたファイルレベルのデータストレージである。ファイル転送プロトコルとして、NFS（Network File System）や CIFS（Common Internet File System）を使用しているため、異なるプラットフォームにおけるファイル共有が可能である。しかし、ブロックベースの SAN と比較すると、データへのアクセス性能は劣る。

近年のストレージネットワーキング分野における研究は、ディザスタリカバリなどの可用性を重視したバックアップ、リモートミラーリング、高速リストアや、管理を効率化するストレージ共有、ストレージの仮想化などに注目が置かれている。また、最近では、ストレージアクセス性能という観点から、メタデータサーバを設置し、高速なデータ管理を可能にしたオブジェクトベースのストレージアーキテクチャも増加傾向にある。また、データインテンシブな処理やアプリケーションが重要になってきていることから、拡張性や I/O 性能の高い分散グリッドベースまたはクラスタベースのストレージアーキテクチャも注目を浴びている。

大量のデータ処理が求められる HPC 分野では、高速にデータアクセスを行う技術に関する研究が、多くなされている。次節では、主に HPC 分野において使用されているファイルシステムや並列 I/O 技術について述べる。

3 HPC 分野における並列ファイルシステム技術

HPC 分野、特に科学技術計算を扱う分野においては、並列 I/O を提供する階層構造は、I/O ライブラリ、I/O ミドルウェア、並列ファイルシステムのようになっている[1]。図 1 は、科学技術計算における I/O の階層構造である。

I/O ライブラリレベルでは、巨大データの格納や並列計算機を想定したハイレベルの入出力ライブラリを提供する。HDF 5（Hierarchical Data Format Version 5）は、NCSA で開発された科学データを保存するためのファイル形式のライブラリである。netCDF（Network Common Data Formant）は、多次元データ形式をサポートするライブラリであり、天文・気象系の計算アプリケーションで多く使用されている。Parallel netCDF は、MPI-IO を使用することにより並列入出力をサポートしている。

I/O ミドルウェアレベルでは、複数のプロセスの協調動作によって I/O を処理する。代表的な例である MPI-IO は、分散メモリ型並列計算機におけるメッセージパッシングインタフェースである MPI（Message Passing Interface I/O）の I/O 性能を高速化したものである。

その下位層には、ファイルレベル高速な並列アクセスを可能にする並列ファイルシステムである。

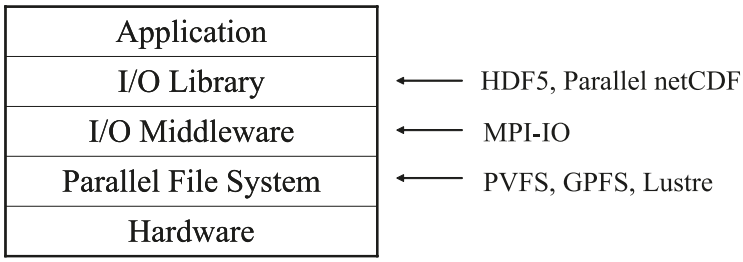


図1　科学技術計算におけるI/Oの階層構造

3.1　クラスタファイルシステム：GPFS、Lustre、PVFS

クラスタシステムには、主にHAクラスタ、負荷分散クラスタ、HPCクラスタなどの種類がある。HAクラスタは、複数のサーバを並列に配置することで、可用性を向上させるシステムである。システムに故障が発生した場合に冗長化することでシステム停止時間を最小限に抑え、共有ディスクの使用やリモートバックアップ、ミラーリングなどによって高可用性を確保する。負荷分散クラスタは、サーバへのアクセス集中を抑えるために、負荷を分散させるためクラスタ構成にするものである。HPCクラスタは、科学技術計算等を実行することを目的として、計算ノードを複数配置し、処理やデータを並行処理させることにより、大規模な高速演算を可能にしたものである。

本節では、効率的なデータ管理を調査するため、既存のHPCクラスタにおけるファイルシステムについて述べる。

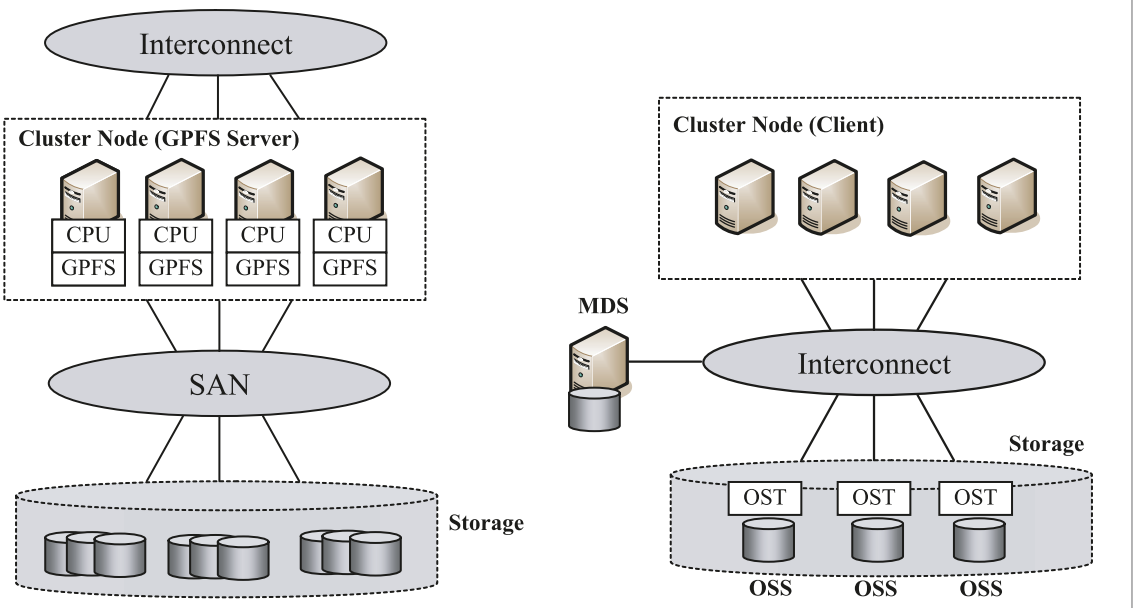
HPCクラスタで使用される並列ファイルシステムには、GPFS（General Parallel File System）[2]、Lustre File System[3][4]、PVFS2[5]などがある。

GPFSはIBMが開発したPOSIX準拠の共有ディスクによるクラスタファイルシステムである。図2(a)は、GPFSを使用したクラスタ構成例を示しいている。AIXまたはLinuxクラスタで使用でき、ストレージアーキテクチャとしてSANのような共有ブロックデバイスに適用可能である。GPFSは分散共有ファイルシステムであり、複数のサーバディスクに単一のファイルをストライピングして分散配置することによって高速なI/Oスループットを実現しており、NFSよりも高速にアクセスできる。また、単一ファイルへの同時アクセスも最適化されたMPI-IOの使用により可能となる。さらに、一つのGPFSサーバがダウンしても、他のGPFSサーバに処理を引き継ぐことができるため、故障時も容易に復旧ができる。

一方、Lustre File Systemは、Cluster File Systems社によって開発されたLinuxまたはSolarisで使用可能なOpen Sourceの分散共有ファイルシステムである。図2(b)は、Lustre File Systemを使用したクラスタ構成例を示している。Lustreは、様々なネットワークを使用可能であり、メタデータを管理するMDS（MetaData Sserver）、ファイルデータを格納するストレージ管理サーバであるOST（Object Storage Target）、クライアント（ノード）によって構成される。メタデータとI/Oデータを分離し、ファイルデータは各OSTにストライピングされることによって、格納される。複数クライアントからの同時アクセスが可能であり、複数のOSTにまたがった大規模ファイルもMPI-IOを利用して並列I/Oが可能である。しかし、MDSが故障するとOSTにアクセスできなくなるため、MDSを複数台用意して、冗長的な構成をとらなければならない。数百規模のクライアントからI/Oノードにアクセスが集中した場合、障害が発生することもあり、安定性より性能を重視した並列ファイルシステムといえる。

文献[6]において、Yuらは、ファイルのストライピング処理が性能を低下させるとして、スプリットライティングと階層的なストライピングによって、性能向上を達成している。

PVFS 2（Parallel Virtual File System 2）は、クレムソン大学を中心に開発された、オープンソースのクラスタ向けファイルベースのストレージモデ



(a) GPFS　(b) Lustre

図2 GPKS と Lustre を使用した構成例

ルである。PVFS では、Linux のローカルディスクに対し、ファイルデータをストライピングすることによって格納している。ノード間の通信は TCP/IP ベースで実行されるため、大量データを使用する場合にはネットワークがボトルネックとなりやすいだけでなく、複数のアプリケーションを使用した同時アクセスの場合は、スケーラブルな性能を得るのは難しい。

3.2 グリッドファイルシステム：Gfarm

HPC 分野における並列処理アーキテクチャの一つとしてグリッドコンピューティング技術がある。

グリッドコンピューティングとは、広域ネットワーク上にある複数のコンピュータを接続することで、高性能な計算を実行するものである。グリッドで使用される並列ファイルシステムとして、産業総合研究所で開発された Gfarm（Grid Datafarm）[7][8] は、データインテンシブなアプリケーションを地域的に分散配置された計算資源とストレージ資源を利用して実行することを目的としたものである。図 3 は Gfarm のアーキテクチャである。小規模から大規模な PC クラスタに適用できるだけでなく、広域分散環境における大規模なデータ解析を高速に処理できるように設計されている。Gfarm のアーキテクチャは、libgfarm ライブラリを持つクライアント、メタデータをキャッシュするメタデータキャッシュサーバ、メタデータを管理するメタデータサーバ、計算ノードと I/O ノードが統合されたファイルシステムノードで構成されている。計算ノードと I/O ノードが分離されておらず、計算ノードはローカル I/O を利用してストライピングすることによって、データインテンシブアプリケーションを実行した際に高性能な並列 I/O を可能にする。また、広域ネットワーク上でファイルの複製を持つことで、負荷分散や耐故障性を可能にし、クライアントはどこにファイルが存在するかを気にせずに利用できる。文献[7] では、ローカルな環境で NFS と同程度の性能が得られたことが実験により明らかになっている。

4 分散ファイルシステム技術

Web アプリケーション向けの分散ファイルシステムとして、現在注目を浴びている技術に Google が開発した GFS（Google File System）[9]、Bigtable[10]、MapReduce[11] がある。GFS は、Linux ベースの分散ファイルシステムであり、Bigtable は分散ストレージシステム、MapReduce

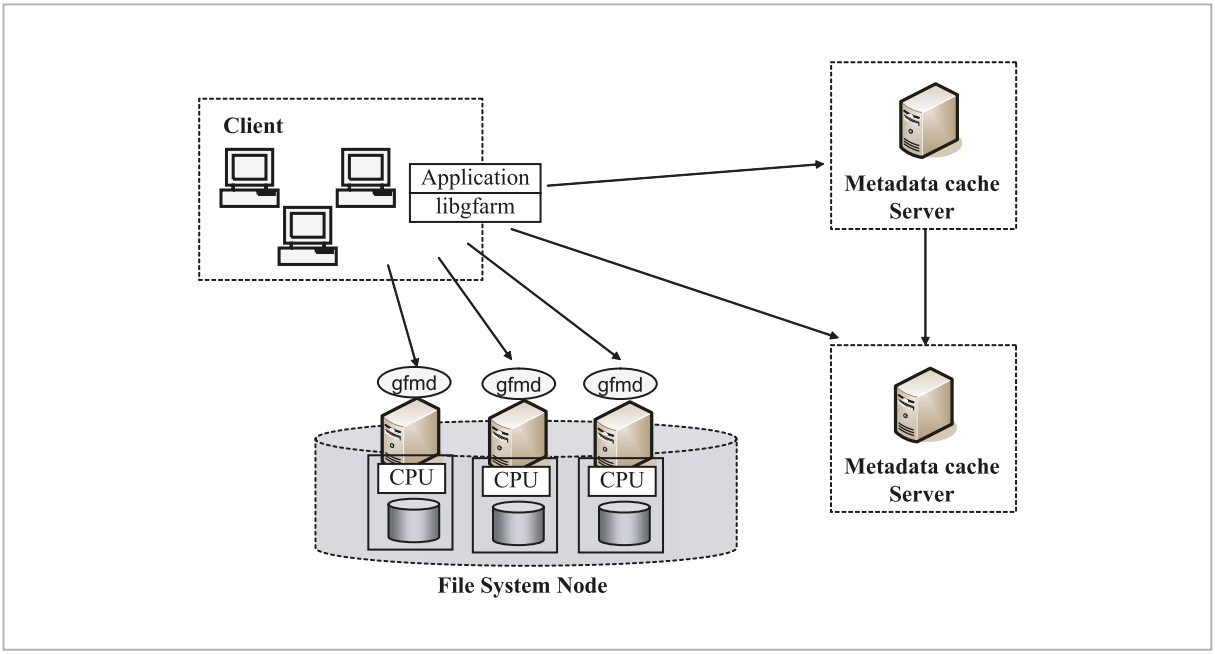


図3 Gfarm

は分散データ処理を行うプログラミングモデルのことである。GFS は、チャンクサーバの位置やファイル名、ロックの管理などを行うマスター、データを格納する複数のチャンクサーバ、クライアントの 3 種類で構成されている。高性能なマシンを使用するのではなく、安価なハードウェアを大量に使用してデータを分散配置することで、障害発生に対応した冗長性の高いデータ管理を実現している。POSIX との互換性をなくすことで性能を向上させているが、小さいファイルの読み書きやランダムアクセスによる性能は高くない。大規模クラスタにおいて簡単に並列データ処理が可能な MapReduce[10] は、マルチコア環境における使用が注目されている。MapReduce は、大規模なデータ処理を Map タスクと Reduce タスクに分解し、処理を分散させて独立に動作させ、それらを組み合わせることで並列処理プログラムを簡単に記述できるようにしたものである。Map タスクでは情報の分解と抽出、Reduce タスクでは情報の集約や計算を行い、結果を出力する。マルチコア環境やウェブ検索、バッチ処理などにおいて高速処理を可能にしている。文献[12]では、SMP クラスタ向けの MapReduce ライブラリの実装について報告されている。それぞれ、GPFS、MapReduce、BigTable のオープンソース版が HDFS（Hadoop Distributed File System）、Hadoop MapReduce、hBase となっている。

5 むすび

近年のネットワーク規模の拡大により、サイバー攻撃を監視し、インシデントを解析するために、大規模データを効率的に保存し、高速にアクセスすることが必要となってきた。

本稿では、セキュリティインシデントの高速な解析には、冗長性や拡張性を考慮した大規模なデータ基盤が必要不可欠であると考え、HPC 分野の観点から、ストレージアーキテクチャ、分散ストレージ技術と並列 I/O 技術について調査した。しかし、それらのストレージアーキテクチャやファイルシステムは、異なる分野又はアーキテクチャレベルで発展してきたため、組合せによって実際に性能がどの程度異なるのかについては、まだ明らかではない。データの種類、データ量、利用目的などによって、様々な形態が考えられ、セキュリティインシデント解析のために必要なシステム構成を十分に考えた上で最適なシステムを構築する必要がある。今後の予定として、既存の分散ストレージなどの技術や並列 I/O 技術をセキュリティ分野に適用する上で何が足りないのかを調査し、補足するための新たな提案を考える。

参考文献

1 Robert Ross, Rajeev Thakur, and Alok Choudhary, "Achievements and Challenges for I/O in Computational Science, " Journal of Physics: Conference Series (SciDAC 2005), pp.501-509, 2005.

2 Frank Schmuck and Roger Haskin, "GPFS: A Shared-Disk File System for Large Computing Clusters", In Proc. the Conference on File and Storage Technologies (FAST'02), 28-30, pp.231-244, Jan. 2002.

3 "Lustre File System", http://wiki.lustre.org/index.php?title=Main_Page.

4 BRAAM, P. J., AND SCHWAN, P. Lustre: The Intergalactic File System. In Proc. the Ottawa Linux Symposium (2002), pp.50-54.

5 "PVFS2", http://www. pvfs.org/

6 Weikuan Yu, Jeffrey Vetter, R. Shane Canon, and Song Jiang, "Exploiting Lustre File Joining for Effective Collective I/O", In Proc. The 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07), May 2007.

7 "Gfarm", http://datafarm.apgrid.org/index.ja.html

8 Yusuke Tanimura, Yoshio Tanaka, Satoshi Sekiguchi, and Osamu Tatebe, "Performance Evaluation of Gfarm Version 1.4 as a Cluster Filesystem", In Proc. the 3rd International Workshop on Grid Computing and Applications (GCA 2007), pp.38-52, 2007.

9 Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google system", In Proc. of the 19th ACM SOSP (Dec.2003), pp.29-43.

10 Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. In 7th USENIX Symposium on OSDI, pp.205-218, Nov. 2006.

11 Jeffrey Dean and San jay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", In Proc. OSDI'04: Sixth Symposium on Operating System Design and Implementation, pp.137-150.

12 Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, and Christos Kozyrakis, "Evaluating MapReduce for Multi-core and Multiprocessor Systems", In Proc. IEEE 13th International Symposium on High Performance Computer Architecture (HPCA 07), pp.13-24.

神坂紀久子（かみさかきくこ）
情報通信セキュリティ研究センター
レーサブルネットワークグループ専攻
研究員　博士（理学）
ネットワークセキュリティ