

提供 IBM

IBM 限定版

ファイル&オブジェクト ストレージ

FOR
DUMMIES[®]

本書で学べること:

- ファイルストレージとオブジェクトストレージ
- 新しいIBM Spectrum Scale™
- IBM Elastic Storage™ Server
- IBM Spectrum Scale オブジェクトストレージの10の利点



Neal Ekker 著

ファイル&オブジェクト ストレージ

FOR
DUMMIES[®]

IBM 限定版

Neal Ekker 著

WILEY

ファイル&オブジェクトストレージ For Dummies® IBM 限定版

発行者

John Wiley & Sons, Inc.

111 River St.

Hoboken, NJ 07030 - 5774

www.wiley.com

Copyright © 2016 by John Wiley & Sons, Inc.

本出版物のいかなる部分も、1976年アメリカ合衆国著作権法の107条又は108条において許可されている場合を除き、発行者の書面による事前の許諾なしに、電子的、機械的、写真複写、録音、スキャン、その他いかなる形態又はいかなる手段によっても、複製、情報検索システムへの格納、伝送することを禁じます。発行者への許可申請は以下までお問合せください。Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030、電話 (201) 748-6011、Fax (201) 748-6008、オンラインによる受付 <http://www.wiley.com/go/permissions>。

商標：Wiley、For Dummies、Dummies Manのロゴ、The Dummies Way、Dummies.com、Making Everything Easier、その他関連トレードドレスは、アメリカ合衆国およびその他の国における、John Wiley & Sons, Inc.及びその関連会社、又はそのいずれか一方の商標又は登録商標であり、書面による許諾なしに使用することを禁じます。IBM及びIBMのロゴは世界の多くの国々で登録されたInternational Business Machines Corporationの米国およびその他の国における登録商標です。その他すべての商標はそれぞれの所有者に帰属します。John Wiley & Sons, Inc.は本書に記載されているいずれの製品又はベンダーとも何ら関連はありません。

法的責任の制限及び免責条項：本書の発行者及び著者は、本書の内容の正確性及び完全性に関していかなる表明も保証も行わないものとし、特定目的に対する適合性に関していかなる保証も行いません。販売や販促資料は、保証や保証の拡大を付与するものではありません。本書に含まれるアドバイス及び戦略は、必ずしもすべての状況に適合するものではありません。本書は、本書の発行者が法務、会計、又はその他専門分野の業務に従事していないという了解の下で販売されるものです。専門的な支援が必要な場合は、しかるべき専門家のサービスを求めてください。本書の発行者と著者はいずれも、本書により生じる損害に一切責任を負うものではありません。本書における企業組織又はウェブサイトの引用や詳細情報源としての参照は、著者や発行者が、かかる企業組織又はウェブサイトの提供する情報や推奨する内容を是認することを意味するものではありません。さらに、本書に記載されたインターネットのウェブサイトは、本書が執筆されてから読者が実際に読むまでの間に、変更又は削除されている可能性があることにご留意ください。

社内のその他の製品及びサービスに関する一般情報、又は企業や組織でFor Dummiesカスタムブックを制作する方法については、当社の事業開発部 (米国877-409-4177、info@dummies.biz) にご連絡いただくか、ウェブサイト (www.wiley.com/go/custompub) をご覧ください。製品やサービスにおけるFor Dummiesブランドのライセンス許諾については、メール (BrandedRights&Licenses@Wiley.com) でお問い合わせください。

ISBN: 978-1-119-28884-8 (pbk); ISBN: 978-1-119-28885-5 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

発行者謝辞

本書の出版にあたり、以下の方々をはじめとする多くの方々にご協力いただきました。

プロジェクトエディター：

Carrie A. Johnson

編集マネージャー：Rev Mengle

発注エディター：Steve Hayes

事業開発担当：Christiane Cormier

特別協力：Derek Gascon、Dean

Hildebrand、Chandra

Mukhyala、Douglas O'Flaherty

目次

はじめに	1
本書について	2
著者の勝手な想定	2
本書で使用されるアイコン	2
第1章ストレージ入門	3
データアクセスとデータ管理の課題	4
ストレージの3つの重要機能	6
ストレージタイプの定義	7
ブロックストレージ	7
ネットワークアタッチドストレージ (NAS)	9
オブジェクトストレージ	10
分散ファイルシステム	12
第2章ファイルストレージとオブジェクトストレージ	15
ファイルストレージとオブジェクトストレージの ユースケース	16
Software Defined Storage	20
Software Defined Storageの定義	20
Software Defined Storageの主な利点	21
柔軟性と俊敏性の向上	21
インテリジェントな資源活用と管理の自動化	22
コスト効率	22
無制限で伸縮自在なデータ拡張	22
ファイル&オブジェクト統合ストレージシステム	23
第3章IBM Spectrum Scaleとは	25
IBM Spectrum Scaleとは	25
階層アーキテクチャ	26
データ関連のボトルネックを解消	28
データ管理を簡易化	30
クラスタの基本的な環境設定	31
複数のSpectrum Scaleクラスタにおけるデータの共有	33
データライフサイクル全体をより低いコストで管理	37
Spectrum Scaleとオブジェクトストレージ	38
Spectrum ScaleとOpenStack	39
Spectrum Scaleオブジェクトストレージ	40

第4章 IBM Elastic Storage Server について学ぶ 45

エンドツーエンドのESSソリューションを提供	46
並列処理能力を耐障害性に活用	50
高度なデータ保護	50
ディスク処理能力の有効活用	51
すべての機能を集約	52

第5章 IBM Spectrum Scale オブジェクトストレージの10の利点 53

ファイルストレージとオブジェクトストレージの統合	54
高度なデータ保護	54
アプリケーション性能の向上	55
Spectrum Scaleの多彩な機能	56
実績のある製品の信頼性と回復力	58
データセキュリティの向上	59
高度なストレージの効率性	60
限界のない拡張性	61
グローバルな共有とコラボレーション	62
Hadoopのような新しい処理をシームレスに追加	63
クラウドストレージの組み込み	65

はじめに

データ、トランザクション、デジタルデバイスの急激かつ加速度的な増加は、今日の情報技術（IT）基盤と運用に多大な負担がかかっています。同時に、ストレージにかかる費用は増加の一途をたどり、ユーザーからの要求やコスト削減のプレッシャーも増していきばかりです。データベースが最も処理しやすい構造化データの増加もさることながら、その大部分が商業活動、研究調査活動、個人の活動により生成される非構造化データ（テキスト、画像、さらにはセンサー情報など）の増大はまさに息を飲むほどで、明らかに大きな課題となってきました。そして年を追うごとに、ありとあらゆる企業や組織が、こうした非構造化データの格納・管理にファイルシステムを使用するようになりました。近年の非構造化データの爆発的な増加と、それを効率的に管理・検索するニーズの加速度的な高まりを受け、新しいストレージへの大きな動きが生まれました。これが「オブジェクトストレージ」です。

このような急激なデータの増加、特に非構造化データの増大を背景として、現在ニーズが高まっているのが、ファイルとオブジェクトに対応した高性能なストレージソリューションです。『ファイル&オブジェクトストレージ For Dummies IBM 限定版』では、IBM Spectrum Scaleという強力なツールに基づいた様々なファイル&オブジェクトストレージソリューションを紹介していきます。

Spectrum Scaleは多数の実績を誇る高性能な製品で、世界中の何千もの企業、研究機関、学術機関、政府機関において、ミッションクリティカルなデータの格納と管理に使用されています。本書では、この優れたストレージプラットフォームを紹介し、その利点を詳しく見ていきます。また、OpenStack SwiftやElastic Storage Serverといった技術についても触れ、どのように導入すれば、企業においてITコストの削減、意思決定プロセスの改善、最も価値の高い資産である「情報」から導き出される洞察と価値の創造を実現できるかを検討します。

本書について

『ファイル&オブジェクトストレージ For Dummies IBM 限定版』では、データの格納・管理における課題を考察し、高性能でコスト効率の良いファイル&オブジェクトストレージを実現するための、Spectrum Scaleを使用した革新的なソリューションについて説明します。

著者の勝手な想定

本書はストレージ管理者やITマネージャーなど、主に技術系の方や組織における意思決定権限を持つ方を対象としています。そのため、読者が、企業向けデータストレージ技術に関して基本的な理解があること、またストレージ管理の課題を解決する意欲があることを想定しています。

本書で使用されるアイコン

本書全体において、重要な情報が一目で分かるよう、特別なアイコンを使用しています。ウインクしている顔文字や可愛い絵文字ではありませんが、これらのアイコンが出てきたら気に留めてください。使用されているアイコンは以下の通りです。



このアイコンは、(大切な記念日や誕生日などと一緒)脳の「持久記憶」に留めておく価値のある大切な情報が記載されている箇所に表示されています。



技術情報と言っても、ヒトゲノムの配列やIBM Watsonの青写真が記載されているわけではありません。このアイコンが表示されている箇所は専門用語が多用され、各種技術が説明されています。



本書をお読みいただきありがとうございます。楽しんでいただけると良いのですが、このアイコンは、役に立つ提案や、有益な情報が記載されている箇所に表示されています。



自己責任でお読みください…。なんて、冗談です。「注意」といってもそこまで危険な内容ではありません。このアイコンは、大きな代償を払うことになりかねない潜在的なミスを回避できるよう実際のアドバイスが記載されている箇所に表示されています。

第1章

ストレージ入門

本章の内容

- ▶ データのアクセスと管理の課題
- ▶ ストレージの3つの重要機能
- ▶ ストレージタイプの定義
- ▶ 分散ファイルシステム

企業ユーザーやデータ利用者は、膨大な量のドキュメントやリッチメディア（画像、音声、動画など）を量産しています。さらに、増加する一方のアプリケーションがこの状況に火を注ぎ、非構造化コンテンツの氾濫を招いています。個人や企業がインテリジェンス、分析、情報共有においてコラボレーションする現代においては、こうした情報の管理は、その量および複雑さにおいて、あらゆる種類・規模の組織にとって大きな課題となっています。

ファイル&オブジェクトストレージという特定のトピックに入る前に、本章ではデジタル・ストレージ技術とストレージ業界の概要について説明します。最初に、ストレージ業界の課題について要約し、情報技術（IT）基盤において、ストレージがどのような役割を果たしているのを見ていきます。その際、一般的に使用されている各種ストレージと、アプリケーションの利用方法に応じたそれぞれの利点についても解説します。次に、デジタルデータを格納するハードウェアの要点と、データの整理における一般的なソフトウェア・アプローチについて触れ、最後にスケーラブルなファイルシステムを紹介します。

データのアクセスと管理の課題

構造化・非構造化どちらのデジタル・データもあらゆるところに存在し、目を見張るペースで増え続けています。毎日、世界中で15ペタバイトもの新しい情報が生成され、デジタルデータの総量は約2年ごとに2倍に増えているほどです。一方で、ストレージに割り当てられる予算は、毎年1~5%しか増加していません。そのためデータ量の増加とストレージ予算のギャップは開いていくばかりです（図1-1参照）。データの爆発的な増加は、その性質、使用頻度の増加と相まって、あらゆる企業やIT部門に途方もないデータストレージの課題を突き付けています。簡単に言ってしまうと、需要に対応するためには、今よりも安価なストレージが必要なのです。



構造化データとは、例えばデータベースなどで、整理されているデータのことです。非構造化データは、定義されたモデルやフレームワークを持たないデータのことです。例えばマルチメディアファイルなどがこれに該当します。

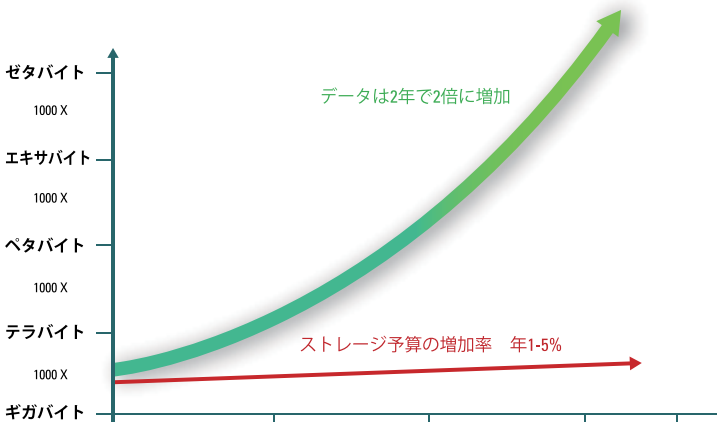


図1-1：ストレージ要件が資本的支出と経常的支出を圧迫

今日のデータの増加は、ある程度まではハードディスク・ドライブを大規模化したり、ネットワーク・コンポーネントを高速度化したりすることで対応することができるとは思います。しかし、こうした技術が進歩するにつれ、データの有効活用が難しくなるという問題があります。より大規模なハードディスク・ドラ

ライブを使用すれば、より多くのデータを格納することは可能になりますが、ドライブを使用する側のハードウェア機器の性能が追いつかない場合が多々あるのです。そうなるとデータは宝の持ち腐れです。安全に格納されてはいるものの、ビジネスの成長を促進したり、意思決定を迅速化したりするのに十分に活用されることなく終わってしまいます。



非構造化データの増加とファイルベースの情報の急増に対処するために、今日多くのITマネージャーが、NAS (Network Attached Storage) 装置を次々と導入しています。NAS装置は、購入時点では比較的成本効率が良いものの、台数を増やしてファイル・ストレージを拡張する際には、データの管理（移行、バックアップ、アーカイブなど）のコストが跳ね上がります。NAS装置を追加するアプローチでは、短期的なストレージ容量の問題は解消できても、アプリケーションの入出力 (I/O) 性能は必ずしも向上するとは限りませんし、管理コストの削減やアプリケーション・ワークフローの改善は見込めません。

今日、多くのデータセンターが、「NAS装置の増殖」の犠牲になっています。IT部門は、手に負えなくなったストレージの需要に何とか応えようと、多数のNAS機器を導入するというむなししい努力を続けています。しかし、実はNAS装置は、容量に対するコストという点で見ると、かなり高くつくだけでなく、新しいデータ・タイプやアプリケーション処理の多くに対して適応性があまり高くありません。さらに、かつてない規模のストレージ需要にただやみくもに応えようとするだけの、こうした間に合わせの処置や暫定的な対応策は、以下のようなデータのアクセス・管理に関する別の課題を引き起こしてしまいがちです。

- ファイルベースのデータを管理するための運用コストが増加する
- 遠隔地におけるデータ・アクセスが制限される
- 継続的なデータの可用性と安全性を維持するのが非常に困難になる
- バックアップやアーカイブ化の作業がデータの増加に追いつかない
- データが増えるにつれアプリケーションの性能要件に適應できなくなり性能の低下が生じる

そして、NAS装置の増殖により生じるこうした問題のいずれかが、ビジネスに影響を及ぼすことも考えられます。各NAS装置は、それ自身がサイロ化されたデータであり、アプリケーションや従業員の情報へのアクセスを制限してしまいます。グローバルチーム、ビッグデータ分析、物流、カスタマー・サービス、コラ

ポレションなどはすべて、ビジネス情報を共有することで成り立っていますから、これが大きな問題であるのは明らかです。



効率的なコンピューティング基盤を実現するには、効果的なデータのアクセスと管理が不可欠です。効率的な基盤に求められるのは、「コンピューティング」「ネットワーク」「ストレージ」という3つの要素の適切なバランスです。通常、企業のIT部門にとって最も難しい課題となるのが、ネットワークとデータ・ストレージです。

ストレージの3つの重要機能

最も基本的なレベルで、企業のストレージは、4つの重要な機能を果たしています。

- ✓ データを格納する（処理中のデータと最終データ）
- ✓ 格納されているデータを管理する
- ✓ データが消失しないよう保護する
- ✓ （コンピューターが意図した作業をできるように）コンピューター・プロセッサにデータを提供する

このうち、格納機能、管理機能、保護機能の必要性については、これまで常にもストレージ管理者により認識されてきましたし、従来のストレージベンダーも十分なソリューションを提供しています。

ただ、現在ストレージ管理者がより重要視するようになってきたのが、ストレージからプロセッサへのデータの転送です。なぜならこれが性能のボトルネックとなっているからです。そして今のところほとんどのベンダーがこの問題に適切に対処できていません。考えてみてください。過去10年間で、

- ✓ CPUの性能は8~10倍向上しました
- ✓ DRAMの性能は7~9倍向上しました
- ✓ ネットワークの性能は100倍近く向上しました
- ✓ バス (bus) のスピードは20倍速くなりました
- ✓ ハードディスク・ドライブ (HDD) の性能はわずか1.2倍しか向上していません

ストレージのボトルネックにより生じる問題の1つが、アプリケーションの応答の遅延です。これにより生産性に悪影響が出て、データセンター内の他の高価な基盤のキャパシティが無駄になってしまいます。

ストレージタイプの定義

すべてのストレージが同じように作られているわけではありません。ストレージ・システムは一般的に、ブロック、ファイル、オブジェクトに分けることができます。そして接続方法にはDAS (Direct Attached Storage)、NAS、SAN (Storage Area Network) があります。



スケールアップとは、単一のアレイまたはシステムに容量を追加することを意味します。例えばストレージ・システムまたはアレイ内の2テラバイト (TB) のドライブを6TBのドライブに交換することなどがスケールアップに該当します。スケールアップしても、ストレージの性能は向上しません。一方のスケールアウトとは、ストレージコンポーネント/ノードあるいはアレイ全体を追加することを意味します。スケールアウトに使用される各ユニットには、コンピューティング機能とネットワーク機能が含まれていますから、スケールアウトすることで性能も向上します。ただ、スケールアウトの場合、より多くの連携調整や管理作業が必要になります。IBMのストレージソフトウェア「Spectrum Scale」のような洗練されたスケールアウト・ソリューションを使えば、多くの処理においてほぼ直線的な性能の向上を実現することが可能です。

ブロックストレージ

ブロックベースのストレージでは、データを、固定サイズまたは固定長のビットまたはバイトのシーケンス (ブロック) としてハードドライブに格納します。そしてブロックベースのストレージ・システムでは、コンピューター (サーバー) のオペレーティングシステム (OS) がハードドライブにアクセスされます。ブロックストレージには以下のような複数のインターフェースでアクセスすることが可能です。

- ✓ ファイバーチャネル (別名FCP：ファイバーチャネルプロトコル)
- ✓ SCSI (スモールコンピューターシステムインターフェース) およびiSCSI (インターネットプロトコルSCSI)
- ✓ SAS (シリアルアタッチドSCSI)
- ✓ ATA (アドバンスドテクノロジーアタッチメント) およびSATA (シリアルATA)



ファイバーチャネル、iSCSIインターフェース、SASは、一般的にコンピューターの外に設置されるストレージに使用されます。これらのストレージはほとんどの場合ネットワークで接続

されます。SASとATAは通常DASで使用されます。ブロックベースのストレージ・システムは、DASまたはネットワークベースのいずれでも実装することが可能です。

DAS (ダイレクトアタッチドストレージ)

DASは、コンピューターに接続される、最もシンプルかつ安価なブロック・ストレージです。その名称が示唆する通り、DASはバスインターフェースでコンピューターまたはサーバーに直接接続されます(図1-2参照)。

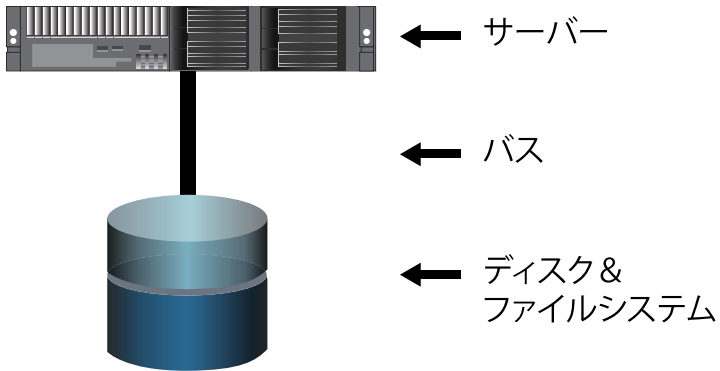


図1-2 : DASではバスインターフェースによりコンピューターまたはサーバーに直接ハードドライブを接続

DASでは、サーバーで利用可能な物理スロット数と同じ数のドライブしか導入できないため、容量が限られます。

SAN (ストレージエリアネットワーク)

SANでは、独立したストレージ・システムを単数または複数のサーバーにストレージ専用ネットワークで接続します(図1-3)。

SANは複数のサーバーで使用することが可能です。各サーバーには、1つ以上のストレージアレイに対して、1回線以上のストレージ専用高速回線が設置されます。SANでは複数のコンピューターが同じストレージ・コントローラーへアクセスすることが可能です。これにより、企業のIT基盤を維持するのに必要な高い柔軟性が実現されます。また大規模な組織においては、システム管理者がコンピューターを管理し、ストレージ管理者がSANを管理するといった作業の分担が可能になります。複数のコンピューターが同じデータへのアクセスを共有できることは、多くのアプリケーションやビジネスプロセスにとって重要です。

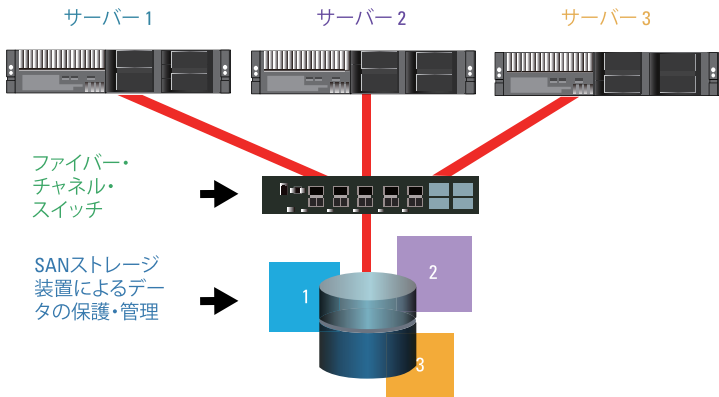


図1-3：SANではストレージ専用ネットワークでサーバーにストレージアレイを接続



LUN（論理ユニット番号）は、ホストサーバーがアクセスできるように、ストレージ・システム内の一連のディスクに割り当てられる識別子で、ストレージコントローラー内で定義、分割されます。これにより、コンピューターはLUNを使ってデータを格納することができるようになります。例えば、LUN内にファイルを格納するための場所として、ファイルシステムを作成するといったことが可能です。ボリュームとは、ボリューム管理ソフトウェア内で作成されたLUNの一部のことです。

SANは一般的にミッションクリティカルな環境あるいは頻繁なトランザクション/IOPS（毎秒のI/O数）が発生する環境で使用されます。例としては、オンライントランザクション処理（OLTP）データベース、エンタープライズリソースプランニング（ERP）、仮想システムなどが挙げられます。

SANは最大限の信頼性と性能を実現するために設計されているため、他のストレージ・システムと比較すると高価です。さらに、SANの管理には専任の管理者と十分な管理体制が必要になるでしょう。特に今日のデータ要件に見合う規模にSANを拡張したいのであればなおさらです。

ネットワークアタッチドストレージ (NAS)

NAS機器のようなファイルベースのストレージ・システムは、通常、非構造化データを、ファイルとして、ディレクトリ構造

でハードドライブに格納します。NAS装置にはそれぞれプロセッサとOSが搭載されています。また、SANのような専用ネットワークではなく、ローカルエリアネットワーク（LAN）上で標準的なプロトコルを使用してアクセスします。以下は一般的なNASのプロトコルです。

- ✓ **SMB（サーバーメッセージブロック）またはCIFS（共通インターネットファイルシステム）**：SMB（またはCIFS）は一般的にWindowsベースのネットワークで使用します
- ✓ **NFS（ネットワークファイルシステム）**：NFSは一般的にUnixまたはLinuxベースのネットワークで使用します

NAS機器は比較的導入しやすく、クライアントのアクセスも共通プロトコルを使用するため簡単です。コンピューターとNAS機器はすべてLANまたは共有インターネット（TCP/IP）ネットワーク上で接続されます。NAS機器に格納されたデータは、OSに関わらず実質的にどのコンピューターからもアクセスすることが可能です。



今日のデータセンターにおいてNAS機器はかなり一般的です。ただNAS機器にはいくつかの大きな弱点があります。NASは通常DASやSANに比べて遅く、すべてのデータがNAS自身のプロセッサを経由する必要があるため、ストレージ性能のボトルネックとなる可能性があります。またNAS機器は拡張性が限られています。1台のNAS機器がいっぱいになると、次の機器を追加し、それがいっぱいになったら、また次を追加し…ということを繰り返さなければなりません。これにより、独立した「ストレージの島」が多数できてしまい、管理が非常に非効率的になります（図1-4参照）。また、NASでは通常カスタムハードウェアと専有技術が使用されるため、コストもかかります。さらに、他の製品やシステムとの統合が簡単ではありません。

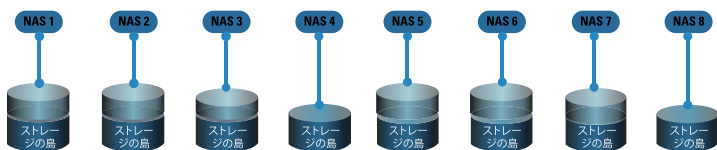


図1-4：NAS装置はデータセンター内で「ストレージの島」になりやすい

オブジェクトストレージ

オブジェクトベースのストレージ・システムでは、ファイルベースのストレージ・システムで見られる階層型ディレクトリベースのファイルシステムではなく、単一のフラットなアドレス空間に、オブジェクトと呼ばれる論理単位を使ってデータを格納します（図1-5参照）。

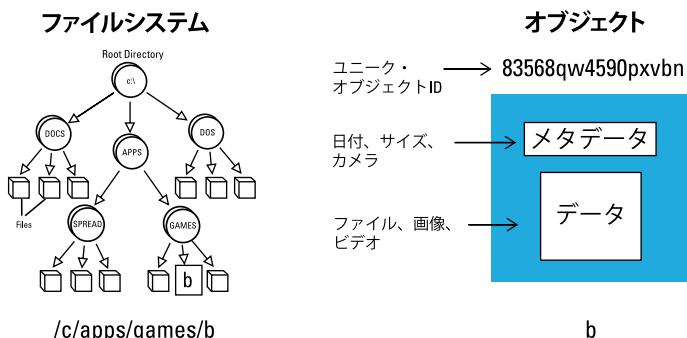


図1-5：ファイルシステムとオブジェクトベースストレージ・システムの比較

オブジェクトには実際のデータ（例：画像や動画）、メタデータ（例：日付、サイズ、カメラの種類）、一意の名前が格納されています。名前はオブジェクトの作成時に割り当てられ、必要に応じてオブジェクトの参照や検索に使用されます。オブジェクトベースのストレージ・システムに格納されているデータには、通常インターネット・プロトコル（HTTP）を使用してウェブブラウザ経由でアクセスするか、リプレイゼンテーション・ステート・トランスファー（REST）のようなアプリケーション・インターフェース（API）経由で直接アクセスします。オブジェクトベース・ストレージ・システムのフラットなアドレス空間は、シンプルで扱いやすく、高い拡張性があります。



オブジェクトベース・ストレージは、一般的にIBM SoftLayerやAmazon S3、Googleといったプロバイダーのクラウド・ストレージ・サービスに使用されています。

最も一般的な3つのプロトコルは以下の通りです。

- ✓ **アカウント**：アカウントは最上層に位置します。アカウントには1つ以上のコンテナが格納され、コンテナにオブジェクトが格納されます。
- ✓ **OpenStack Swift**：OpenStack Cloudプロジェクトの一部であるSwiftは、企業やさまざまな公的機関、技術ベンダーが広く支援するオープンソース・プロジェクトです。Swiftでは、オブジェクトはコンテナに格納されます。
- ✓ **S3**：Amazon Web Servicesで使用されているオブジェクトの仕様です。S3はいち早く完全に定義されたオブジェクト・プロトコルの1つであり、AWS以外で稼働している多くのアプリケーションでも使用されています。S3では、オブジェクトはバケットに格納されます。

OpenStack SwiftとS3のAPIは類似しており、多くのオブジェクトストアが両方に対応しています。



オブジェクト・ストレージは高い拡張性を誇るだけでなく、ユーザーにとってデータを見つけやすいという利点があります。ファイルシステムのような階層がないため、アプリケーションは、オブジェクトの名前さえ分かれば、データを見つけてアクセスすることが可能です。ただし、ファイルシステムがないことにより、ユーザーは階層をナビゲートしてオブジェクトを探すということはできません。ストレージ容量の追加に関しては、同一アレイ内に追加することもできますし、インターネットのような大規模な分散ネットワーク上に追加することも可能です。

分散ファイルシステム

従来のNASシステムの限界に対処するため、分散ファイルシステム（クラスタ・ファイルシステムと呼ばれることも多い）を導入する企業が増えてきました。分散ファイルシステムでは、複数のストレージ装置またはノードからデータが供給されます。CIFSおよびNFSでは、データは通常、ストレージが直接接続またはSAN経由で接続されている単一のノードまたはNAS機器から供給されます。しかし分散ファイルシステムでは、ネットワーク上で同時に多数のコンピューターから複数のストレージ装置にアクセスすることが可能です。分散ファイルシステムでは、データを複数のストレージ装置に分散することで、高いI/O性能を実現しています。またストレージ装置はすべて同じグローバルな名前スペースを共有しているため、拡張性も高いのも特徴です。そのため迅速かつ大幅に拡張する必要のあるストレージ環境に特に適しています。

分散ファイルシステムは、CIFSやNFSといったネットワークプロトコルに類似していますが、相違点が2つあります。

- ✓ 分散ファイルシステムは、アプリケーションのホストコンピューターから直接アクセス可能なため、通常のネットワーク・プロトコルと比較して、性能面でいくつかの利点があります。
- ✓ クラスタ化されたファイルサーバーは互いに密接に接続され、より高度な通信を行っています。そのため例えばアプリケーションが複数のノードから1つのファイルに対して読み取り・書き込みを実行するといったことが可能です。

多くの分散ファイルシステムは、同じ機能をTCP/IPまたはInfiniBand上で拡張します。これは、データへのアクセスにネットワークを使用するNFSやCIFSのアプローチと類似していますが、分散ファイルシステムの場合、データの転送に使用されるネットワークプロトコルは、分散ファイルシステム・ソリューション自体に統合されています。この緊密な統合により、分散ファイルシステムは高い性能とネットワーク上の高度なアクセスパターンを実現しています。また、これらのプロトコルはリモート・ダイレクト・メモリ・アクセス（RDMA）などの技術を活用することで、さらに高速なデータ処理が可能です。

分散ファイルシステムは、非常に緊密な統合により、高速に高度な機能を提供することが可能ですが、ワークステーションのアクセスには適していません。例えば、音楽にアクセスするためにタブレット上で分散ファイルシステムを実行するというようなことはありません。分散ファイルシステムは、例えば多数のユーザーが音楽をダウンロードするような大規模なエンタープライズIT基盤におけるファイルへのデータアクセスを強化するために設計されています。

分散ファイルシステムは、複数のストレージ・ノードで並行して読み取りや書き込みを実行可能で、非常に高度な性能、拡張性、データ保護を実現しています。分散ファイルシステムについては、第2章で詳しく見ていきます。

14 ファイル&オブジェクトストレージ For Dummies IBM 限定版 _____

第2章

ファイルストレージと オブジェクトストレージ

本章の内容

- ▶ ストレージのタイプが複数存在する理由
- ▶ ファイルとオブジェクトの利用方法
- ▶ Software Defined Storage (SDS)

情報システムの処理機能とネットワーク機能が進化するにつれ、ストレージも多様なアプリケーションに対応できるように進化してきました。すべてのコンピューティングがメインフレームのような単一のシステムで行われていた頃は、ストレージは環境に対して最適化されたローカルの専有接続で配置することができていました。しかし、さまざまな分散コンピューティングシステムが広く普及するようになると、それらに対応するために、装置、フォルダ、サブフォルダ、名前という共通の原則に基づいた共有ファイルシステムが開発されました。この考え方は、基盤となる技術にこそ変化があったものの、何十年にも渡り業界で採用されてきています。

ファイルシステムがこのように進化し続けたのは、混沌に秩序をもたらしたためと言えるでしょう。構造を持たない情報に、人間が理解し簡単にナビゲートすることのできる構造を与えたからです。しかし、ファイルシステムにも限界があります。

データセットと呼ばれる情報の集まりが、1つのストレージ装置で扱える大きさを超えるようになると、密接に連携するマルチストレージ装置という概念が生まれてきました。第1章では、そのうちの分散（クラスタ）ファイルシステムについて触れました。分散ファイルシステムは、データの物理的な場所とアプリケーションを切り離すことで、多数のストレージ装置を使用することを可能にしました。しかもアプリケーションから

見た場合には、システムは単一のストレージとして認識されず。この設計またはアーキテクチャの利点は、ストレージ容量や性能の追加・拡張が非常に容易だということです。

複数のサーバーがクラスタ環境で連携して機能できるようになると、複数のネットワークリンクでストレージへ並行アクセスできることが重要になってきます。そこで開発されたのが、分散ファイルシステムです。しかしこうした開発が、ファイルアクセスの基本的な設計を変えることはありませんでした。クラスタ化であれ並行アクセスであれ、アプリケーションやエンドユーザーにとっては、情報は、装置>フォルダ>サブフォルダ>ファイルというおなじみの階層を持った単一の装置として認識されていました。

しかし、連携して動作するコンピューターの数が増えてくると、新しいデータストレージのタイプが登場します。オブジェクトです。オブジェクトストアでは、おなじみの階層構成を一掃し、各データユニットに一意の名前を割り当てるとする方法を使っています。

ファイルストレージとオブジェクトストレージの利用方法

以降、本書で焦点となるストレージ技術に入る前に、一步下がって、基本的な質問を考えてみましょう。なぜファイルストレージやオブジェクトストレージが使用されているのでしょうか？ 何のために使用されているのでしょうか？ そして、誰が使用しているのでしょうか？

構造化データは、その生来の性質上、内在する秩序、つまり構造をもたらしめました。その構造は通常、階層的な性質を持っており、データベースと呼ばれるアプリケーションで管理、操作できる形になっています。これまで何十年にも渡り、データベースは、ビジネス、政府、科学分野を牽引する大部分のアプリケーションに構造化データを供給してきました。

しかしその一方で、内在する階層構造や秩序を持たない情報も常に存在していました。本書を構成している内容などはその好例です。こうした構造を持たないタイプの情報を、より効率良く有効に格納、発見、管理、操作、検索、分析するために開発されたのがファイルシステムです。ファイルシステムは、基本的な非構造化データに、ディレクトリ、フォルダ、サブフォル

だ、ファイルといった構造の層を与えました。これにより、ユーザーやアプリケーションはより効率よく、非構造化データを格納、カタログ化したり、特定のデータを必要に応じて発見、取得、使用したり、すべての変更履歴を明確に追跡したりすることができるようになったのです。今ではほとんどすべてのビジネスにおいて、作成された基本的な文書、プレゼンテーション、スプレッドシート、画像は、ファイルシステムに格納され、ローカルエリアネットワーク（LAN）経由でアクセスできるようになっています。

ファイルストレージは、人間の働き方に合った性質を持っていると言えるでしょう。正確性を保護するため、データの編集中にはそのデータをロックし、他の人が編集できないようにすることも可能です。さらに、ファイルは名前を変更せずに大きくも小さくもすることができるため、非構造化データへのアクセス、使用、更新にあたり非常に便利です。以上のような特性により、ファイルは、データベースやアプリケーションによるデータの操作、対比に適したソリューションなのです。ファイルシステムの一般的な利用方法としては以下のようなものが挙げられます。

- **コラボレーション**：ファイルシステムの技術により、複数のユーザーが、バージョン管理の問題を引き起こすことなく、同時に同じファイルにアクセスすることが可能です。
- **構造化データ用アプリケーション**：ファイルシステムにより、構造化データ用に設計されたアプリケーションで非構造化データを使用することが可能です。
- **急速に大きくなるファイル**：ファイルシステムは、装置のログ、メディアの記録、科学情報など、急速に変化、増大するファイルを簡単に対応できます。
- **ビッグデータ分析**：大規模な情報の蓄積は、複数に分割し、それぞれネットワーク上で並行処理、アクセスできるようにすることで、効率的に処理することができます。ファイルシステムはそのような環境に最適なソリューションです。

非構造化データは、構造化データの増加速度を上回る速度で増え続けているため、今日、ファイルシステムはかつてないほど重要となり普及しています。例えば、多くの科学研究が大規模なファイルシステムに依存していますし、ヒトや動物のゲノムシーケンス解析にはファイルシステムが使われています。もちろん、世界中の企業や政府機関でも未曾有の量の文書や情報が生み出されています。さらに、新しいアプリケーションについても考えてみてください。ソーシャルメディア、モバイルアプ

り、そしておそらく非構造化データの最大の貯蔵庫であるインターネットもあります。今日および将来的なファイルシステムの利用方法を考えた場合、その一覧はここには収めきれないほど長いものになるでしょう。

しかし、情報の未来へ進む道のりにおいて、もう1つ考えなければいけない要素が出てきました。速度です。増殖し続けるデータの量と種類に、この速度という要素が加わると、IT基盤における新たな課題が浮上してきます。インターネットでテレビ番組をストリーミングすることが、技術的にどのくらい大変かを考えてみてください。ましてや、スイスにあるCERN（欧州原子核研究機構）の大型ハドロン衝突型加速器から得られる結果を世界中の科学者たちが利用できるようにすることは、どれほど難しいことでしょうか。実際、あなたが祖父母の写真をお気に入りのソーシャルアプリのページに投稿して、世界中の誰もがダウンロードできるようにすることだけでも、技術的には非常に高度なのです。特にその写真が絶対に失われないようにし、東京でもスペインのトレドでも同じくらい速くダウンロードできるようにしなければなおさらです。

こうした種類の困難を克服すべく開発されたのがオブジェクトストレージです。オブジェクトとはデータの単位で、オブジェクトにはデータ自体の情報が含まれます。このような情報は、従来であればデータベースやファイルシステムといった他のデータストレージ技術から供給されるものです。しかしオブジェクトは、自らこの情報を供給しているため、こうしたストレージ技術からある程度独立して存在することができます。そのためITソリューションプロバイダーにとって、オブジェクトストレージは、どのように実装し活用するかという点において高い柔軟性があるのです。次に、オブジェクトストレージは必ずしも階層的な構造で整理されているわけではありません。むしろオブジェクトでは、「フラットな」（階層的でない）アドレス空間に格納された何らかの形式のシンプルな名前またはキーが使用されます。

オブジェクトストレージのこうした特性は、ストリーミング動画やクラウドコンピューティングの時代である現代において大きなメリットがあります。オブジェクトストレージにより、管理ソフトやシステムの複雑性と間接費が軽減され、特にインターネットベースのアプリケーションにおいてデータを非常に柔軟に活用することが可能になります。

つまり、「スタートレック」や「アイ・ラブ・ルーシー」、「ボナンザ」、あるいは「Xファイル」のエピソードを検索、ダウンロードし、ノートパソコンで簡単に素早く再生できると

ということです。これは、その基盤にある技術的な困難を考えると驚くべきことです。あるいは、かかりつけの医師がPET（陽電子放射断層撮影）スキャンの結果について、同時に同じ画像を見ながら、インドにいる専門家と話し合うことができるということです。そして、あなたの祖父母の写真の複数の複製が、世界の異なるロケーションにあるストレージシステムに存在し、1つのデータセンターで災害が起こっても、その写真を見たいと思った人がどこからでも好きな時にインターネット上で見ることができるということです。

オブジェクトストレージは、複数のコンピューターが異なるロケーションから利用することができます。クラスタ化されたアプリケーション（例：ビッグデータ分析）や、同じデータを必要とする異なるアプリケーション（例：同じ写真を異なるウェブサイトで使用）などがそうした環境に該当します。オブジェクトストレージのインターネットプロトコル（RESTful）およびアプリケーションインタフェース（API）は、異なるシステム間で負荷を分散することが可能です。このように同時に複数の場所にデータを存在させることができるというのは、特定の物理的な場所でシステムに予期せぬ障害が起きた場合などに、ストレージをダウンさせたり、遅延させたりすることなくサービスを継続するのに特に役立ちます。

オブジェクトストアでは、1人のユーザーが情報にアクセスして更新している間、バージョン管理の問題を回避するためにファイルをロックする必要がありません。オブジェクトは一旦作成されると、大きくしたり小さくしたりすることはできません。大きさを問わず、オブジェクトが書かれると、ストレージシステムにより一意の名前が割り当てられます。データへのアップデートを処理するには、新しいオブジェクトが書き込まれます。オブジェクトストレージの一般的な利用方法としては以下のようなものが挙げられます。

- コンピューティング機能が追加できるよう、ゆるく組み合わせられたストレージ・インタフェースを活用するウェブサービス
- 異なるサービスのために同じデータを参照するアプリケーション
- 仮想マシンのイメージのように、書き込みは頻繁ではないものの、多種類のデータを頻繁に読み取る必要があるもの
- 頻繁にアクセスしないデータのスケールアウト（プロトコルの間接費を伴うことなくスケールアウトできるオブジェクトストアの拡張性を活用）

Software Defined Storage

ファイルシステムやオブジェクトストレージはどちらも非常に高機能なソフトウェアであり、標準的なハードウェアを使用したストレージソリューションの設計/構築に適しています。こうしたアプローチは、*Software Defined Storage*と呼ばれています。

Software Defined Storage の定義

簡単に言うと、*Software Defined Storage*とは、標準的なハードウェアを使用し、重要なストレージ機能/管理機能はインテリジェントなソフトウェアに実行させるデータストレージのことです。ストレージの仮想化として考えると理解しやすいかも知れませんが、ハードウェアは、ストレージシステムの実際の機能と「対になっていない」、つまり物理的に独立しており、ソフトウェアにより制御されます。これは本質的には、アプリケーションサーバーにおける仮想化と同じです。

優れたSoftware Defined Storageは、基盤となるストレージ基盤全体を組織化することで、ポリシーに基づき自動化され、アプリケーションに対応したストレージサービスを提供します。

Software Defined Storageのその他の特徴としては以下があります。

- ✔ 情報ライフサイクルマネジメント (ILM) やキャパシティプロビジョニングなど、ストレージ管理機能における、自動化されたポリシーに基づく管理
- ✔ ストレージの仮想化 (アプリケーションがデータの物理的なロケーションを認識しない)
- ✔ ストレージ基盤と、その基盤上のデータを別々に管理するための独立したコントロール層とデータ層

これらの特徴は、ストレージ機能の実行において、ハードウェアベースのカスタム・コントローラに大きく依存した従来のストレージシステムとは対照的です。

Software Defined Storage の主な利点



今日、企業は、Software Defined Storageの多くの利点について認識し始めています。利点としては以下が挙げられます。

- ✓ 柔軟性の向上
- ✓ 自動化されたストレージ管理
- ✓ コスト効率
- ✓ 制限のない拡張性
- ✓ 使用するハードウェアが選べる自由
- ✓ システムの追加（スケールアウト）によるキャパシティの増加

柔軟性と俊敏性の向上

SANやNAS（第1章参照）など、企業における従来のストレージ・プラットフォームは、通常専有システムを基本としており、高い総所有コスト（TCO）がかかります。SANソリューションは、通常高価で複雑なSANスイッチ、ストレージアレイ、その他専有コンポーネントが必要です。

NASは、装置自体は比較的手ごろなコストであるものの、拡張性が限られています。1つのNAS装置で容量が足りなくなれば、新しくNAS装置を追加する必要があります。ただし、これは本当の意味でのスケールアウトとは言えません。なぜなら、個々のNAS装置は、別々に管理される独立したストレージとして扱われるからです。

Software Defined Storageソリューションでは、専有ではない標準的なハードウェアを使用できるようにすることで柔軟性を高めており、多くの場合、既存のストレージ基盤を企業のストレージ・ソリューションの一部として活用することが可能です。さらに、企業は、ソリューションのキャパシティを増やし性能を向上させるために、必要に応じて異種機混在のハードウェア・コンポーネントを追加して大規模な拡張を実現することが可能です。Software Defined Storageの拡張性は、急速に増加する大量の非構造化データを持つファイルシステムやオブジェクトストレージ環境で特に有用です。

インテリジェントな資源活用と管理の自動化

Software Defined Storageソリューションでは、自動化された、ポリシーに基づく管理により、コストや運用を効率化しています。例えば、Software Defined Storageは、ディスクキャッシング、スナップショット、レプリケーション、クラスタリングといった重要なストレージ機能を管理します。簡潔に言うと、こうしたSoftware Defined Storageの機能により、適切なデータを適切な場所に、適切なタイミング、適切な性能、適切なコストで、自動的に配置することが可能になるのです。

コスト効率

Software Defined Storageでは、高価な専有ハードウェアを使用する代わりに標準的なハードウェアを使用することで、企業向けのストレージ・ソリューションの取得コストおよびTCO（総所有コスト）を大幅に削減しています。Software Defined Storageソリューションのソフトウェアは、標準機能で、ストレージ基盤とストレージシステム内のデータの両方を管理します。

多くの場合、組織は以前に投資した既存のストレージ、ネットワーク、サーバー基盤を活用して、非常にコスト効率に優れたSoftware Defined Storageによるファイル/オブジェクトストレージを実現することが可能です。



2012年7月のガートナー社の報告によると、従来のマルチティアストレージシステムにおける1ギガバイト当たりの取得コストは、0.90米ドル/GB～5.00米ドル/GB（約102～570円/GB）でした。一方、Software Defined Storageソリューションの2012年の取得コストの平均は0.40米ドル/GB（約46円/GB）で、以降下がり続けています。

無制限で伸縮自在なデータ拡張

SANやNASといった従来のストレージ装置と異なり、Software Defined Storageは、ストレージを単一の企業向けのストレージ・システムとして管理しつつ、比較的手頃な価格の標準的なハードウェアでスケールアウトすることが可能です。ファイルやオブジェクトのストレージ基盤をスケールアウトすることで、性能や信頼性も向上します。例えば、IBMのSoftware Defined

Storageソリューションである「IBM Spectrum Storageファミリー」において、ファイルシステム管理機能を担うIBM Spectrum Scaleは、ハードウェアが追加された場合に、従来のNASと比較して桁違いのI/O性能を実現しています（図2-1参照）。IBM Spectrum Scaleについては、第3章で紹介し詳しく説明します。

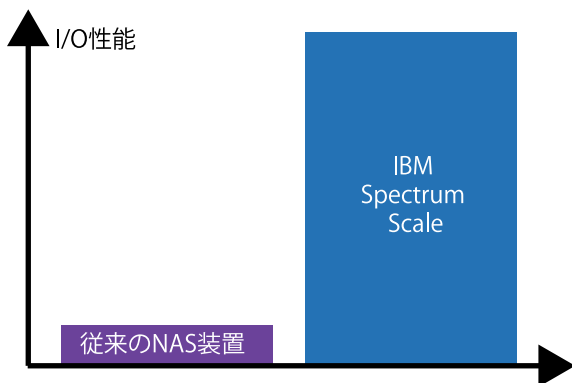


図2-1： IBM Spectrum Scaleは優れたI/O性能を実現

Software Defined Storageは、大規模な、事実上制限のない拡張性を提供しています。例えば、IBM Spectrum Scaleは次に挙げる特徴を備えています。

- ✓ ファイルシステムサイズ：最大100万ヨタバイト
- ✓ ファイル数：ファイルシステム当たり 2^{63} （約900京）
- ✓ IPv6
- ✓ ノード数：クラスタ当たり1～16,384



1ヨタバイトは、1兆テラバイトです。

ファイル&オブジェクト統合 ストレージシステム

Software Defined Storageは、ファイルストレージとオブジェクトストレージにおいて、非常に重要な役割を果たしています。非構造化データの爆発的な増加を背景に、ファイルベースのデー

タの急速かつ継続的な増加と、オブジェクトストレージの普及が起これ、企業はそれらに対応するために、当然の成り行きとしてシステムを「追加」しています。しかし、もともと整然としてそれなりに居心地の良かった家に、ただ部屋や翼棟を追加しただけでは、混沌とした雑な作りの屋敷になってしまいます。同様にストレージシステムをただ追加するだけでは、多くの意味で複雑性を招き、コストを倍増させてしまいます。

データの量や種類、速度が増加しても、IT予算は同じように増えてはくれません。そのため企業やIT部門は、ファイルストレージやオブジェクトストレージを含む情報システムをできる限り効率化し、コストを節約することを余儀なくされています。

こうした状況に対処するために設計されたのが、Software Defined Storageソリューションです。論理的あるいは「仮想的」に、多数の多様な、多くの場合サイロ化しているストレージ・システムを、ファイルストレージとオブジェクトストレージの両方に対応した単一のソリューションに統合することにより、Software Defined Storageは簡便性と効率性を劇的に向上させ、大幅なコスト削減を実現しました。アプリケーションがSoftware Defined Storageの鏡を覗いたとき、そこには混沌としたストレージの屋敷ではなく、整然と片付いた扱いやすいファミリーホームが映っているはずです。さて、ファイル&オブジェクトストレージの話が面白くなるのは、いよいよここからです。

第3章

IBM Spectrum Scaleとは

本章の内容

- ▶ IBM Spectrum Scale
- ▶ Spectrum Scaleのファイル&オブジェクト統一アーキテクチャ
- ▶ Spectrum ScaleとOpenStackを組み合わせる利点
- ▶ オブジェクトストレージとしてSpectrum Scaleを使用する利点

IBMは、Software Defined Storageにおいて市場をリードしています。Software Defined Storage製品シリーズのSpectrum Storageファミリーに含まれるIBM Spectrum Scaleは、成熟し高度に洗練された、それでいてシンプルかつパワフルなソリューションで、構造化および非構造化データから最大限の価値を引き出します。本章では、Spectrum Scaleが、いかに多くの面で企業のITコストを削減し、ビジネスの俊敏性を高め、優位な競争力をもたらすかを見ていきます。Spectrum Scaleではファイルストレージとオブジェクトストレージの両方に、統一インターフェースからアクセスすることが可能です。Spectrum Scaleのレイヤーアーキテクチャにより、ユーザーは、特定のアプリケーションに必要な最先端のファイルストレージおよびオブジェクトストレージの環境/プロトコルを導入することが可能です。

IBM Spectrum Scaleとは

IBM Spectrum Scaleは、分散ファイルシステムとして開発され、今日までに大きな進化を遂げてきました。現在のSpectrum Scaleは、ストレージの仮想化、統合された高い可用性、ストレージの自動階層化管理、高性能な環境設定など、膨大なデータを

効果的に管理するためのデータ管理機能をすべて実装しています。Spectrum Scaleは、様々なアクセス・プロトコルを取り入れることで、多種多様なアプリケーションに対応できるように設計されており、大規模かつ要求の厳しい環境において非常に効果的であることが証明されています。

他のストレージ・ソリューションでは、ファイルやオブジェクトを処理するために別々のシステムを追加で実装する必要がありますが、Spectrum Scaleでは、すべてのデータを、そのアクセス方法に関わらず、Spectrum Scaleファイルシステム内に格納することが可能です。ファイルシステムに格納されたデータは、ブロック、ファイル、オブジェクトのいずれかに関わらず、基本的に同じ方法でアクセス、管理することができます。さらに、スナップショット、情報ライフサイクル管理 (ILM) タスク、ストレージ階層化、非同期・同期データのレプリケーション、独自のデータ保護戦略など、多彩なデータストレージのサービスや機能を、必要に応じてすべてのデータに適用することが可能です。

他のSoftware Defined Storage製品、特にオブジェクトストレージ市場の他の製品とは異なり、IBM Spectrum ScaleはOpenStack Swift、Amazon S3、CIFS、NFSv3、NFS v4、POSIXといったほぼすべての標準ストレージ・プロトコルを利用できます。ブロック、ファイル、オブジェクトデータに対して、高性能かつスケラブルなネイティブアクセスを提供します。他のソリューションの場合は、データの格納方法にアクセスプロトコルが紐づけされているため、データを取得するには、ゲートウェイを使用して異なるデータアクセス・プロトコルを解釈しなければなりません。



Spectrum Scaleは、単なるオブジェクトストアでもなければ、純粋なNASボックスでもありません。非常に安定性が高く、極めて優れた拡張性を有する、I/O性能に最適化された分散ファイルシステムです。様々なプロトコルが組み込まれており、高い対応性も実現しています。Spectrum Scaleは他のシステムとは一線を画す、非常にパワフルで柔軟性の高いシステムです。

階層アーキテクチャ

Spectrum ScaleのSoftware Defined Storageは、複数のデータプロトコルを展開できるストレージサービスの中核と考えることができます。ストレージ・サービスには、テープからディスク、フラッシュまで、あらゆるタイプのストレージを使用することが可能です。Software Defined StorageであるSpectrum Scaleは、図3-1で示されている通り、サーバーのローカルディスクを含む事実上あらゆるストレージプールを統合することが可能です。

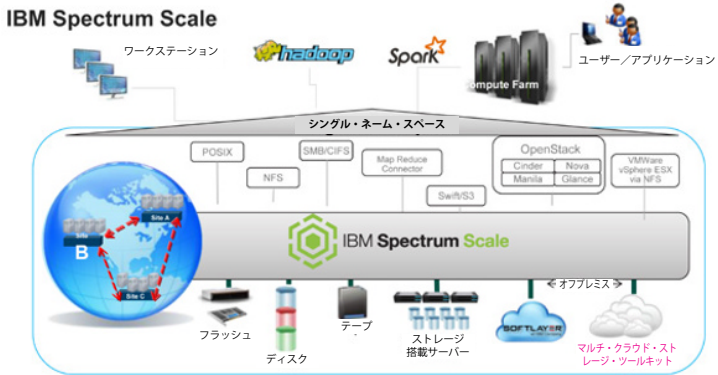


図3-1：IBM Spectrum Scaleは共通ストレージ層を提供



ストレージへのアクセスは、データサービスにより可能になります。データサービスには、業界標準高速ファイルアクセスのPOSIX、NFS、CIFS/SMB、HDFS、さらにOpenStack SwiftやS3インターフェースのオブジェクトストアがあります。Spectrum Scaleでは、ストレージ管理者は、こうしたデータアクセス・プロトコルに左右されることなく、レプリケーション、階層化、圧縮など高機能なストレージサービスを利用することが可能です。POSIX（ポータブル・オペレーティングシステム・インターフェース）とは、UNIXを始めとする異なるオペレーティングシステム間の適合性を維持するためにIEEE（米国電気電子工学会）が策定したファイル志向の規格で、大部分のオペレーティングシステムでデータのアクセスに使用されている標準ファイル・インターフェースです。

Spectrum Scaleでは、複数のファイルサーバーやオブジェクトサーバーを使用することで、性能とスループットを向上することが可能だけでなく、レジリエンシー（回復力）を確保するフェイルオーバー機能も提供されています。サーバー群は、共通のストレージ基盤、ネットワーク、様々な接続タイプを通じて、共通のファイルデータに同時にアクセスすることが可能です。サーバーには、AIX、Linux、Windows Serverを自由に組み合わせて使用することができます。Spectrum Scaleは、ストレージ管理、ILMツール、一元化された運用管理を提供しています。加えて、グローバルなネームスペースや関連ストレージプールをすべて一元管理しており、リモートのSpectrum Scaleクラスタからファイルシステムに共有アクセスすることも可能です。

Spectrum Scaleクラスタは、例えば1台のサーバーの場合もあれば、データベース・アプリケーションに対応した高い可用性のプ

ラットフォームを提供する2台のサーバーの場合もあるでしょうし、あるいは気象パターンのモデリング用アプリケーションに使用されるような何千台ものサーバーの場合も考えられます。Spectrum Scaleは、高性能が求められる処理に対応できるように設計されており、これまで多岐に渡るアプリケーションにおいて非常に効果的であることが実証されています。今日、Spectrum Scaleは、ビッグデータ分析や遺伝子シーケンス解析、デジタルメディア、拡張性に優れたファイルサービスなどのアプリケーションに対応するためのクラスタ環境に採用されています。そしてこうしたアプリケーションは、金融、小売、デジタルメディア、バイオテクノロジー、政府など多岐に渡る分野に及びます。

Spectrum Scaleは、高度なアプリケーション機能に対応した、独自の広範なインターフェースとコマンドを提供しています。これらのインターフェースを使用することで、アプリケーションは、ファイルのストレージプールの配置決定、ファイルクローンの作成、クォータの管理といったオペレーションを実行することができます。

データのボトルネックを解消

過去10年に渡り、プロセッサ、メモリ、ネットワーク、バスの性能はすべて急激に進化してきました。しかしディスク性能はわずか1.2倍しか向上していません。この性能差により、データ量の多いアプリケーションの速度が落ち、スケジュールが遅れ、高価な基盤が無駄になっています。Spectrum Scaleは、データへの並行アクセスを通じて、処理結果が出るまでの時間を短縮し、基盤の活用を最大化します（図3-2）。

Spectrum Scaleは、以下により高いI/O性能を実現しています。

- 複数のサーバーに接続された複数のディスクにデータを分散配置
- クライアント側で効率的なキャッシングを提供
- 高速なメタデータ (inode) スキャンを実行
- I/O要件に合致できるようにファイルシステムで様々なブロックサイズに対応
- I/Oの運用性を高める高度なアルゴリズムを使用
- 複数のアプリケーション・サーバーによるファイルへの同時アクセスを可能にしつつ、データの一貫性を維持するため非常に洗練されたトークン管理システムに基づいたロック処理を採用

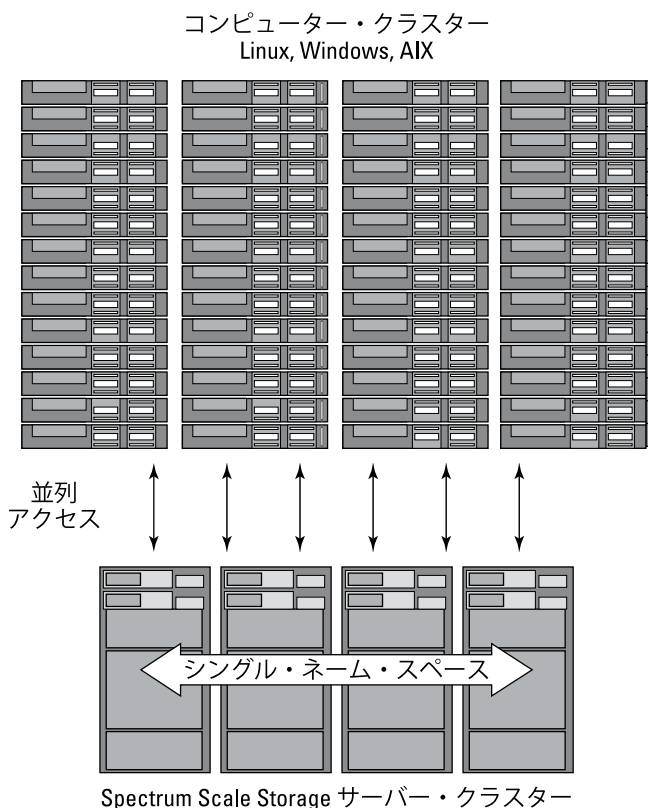


図3-2：IBM Spectrum Scaleはストレージボトルネックを解消

多くのサーバーが同じファイルセットを同時に使用する必要がある場合、ファイルシステムは、すべてのファイルを保護し、あるサーバーが他のサーバーが知らないところで、データを変更できるような状況を防がなければなりません。何千ものサーバーに常にファイルの最新ステータスを共有しつつ、高い性能を維持するのは簡単なことではありませんが、Spectrum Scaleではこれが主要なスケールリング機能の1つとして提供されています。

Spectrum Scaleは、トークンプロセスを使って1つのファイルのオーナー（所有者）を常時1人に制限することでファイルデータの一貫性を保ち、ファイルの整合性を保護しています。そしてこのトークンプロセスには高い拡張性があります。なぜならクラスター内のどのサーバーもファイルを保護する機能を果たすことができるからです。トークンとファイルの一貫性のある管理は、「トークンの配布」と「ファイルのメタデータを最新に保つ」という2つの要素で成り立っています。

最初に、使用されていない全てのファイルに対してトークンを有するサーバーをトークンマネージャーと呼びます。トークンマネージャーには1台あるいは複数のサーバーを割り当てることが可能です。複数のトークンマネージャーは、負荷を分担したり、いずれかのトークンマネージャーがダウンすると他のトークンマネージャーが作業を引き継いだりして、互いに補完し合います。ファイルが開かれると、トークンマネージャーはそのファイルのトークンを、ファイルを開いているサーバーに渡します。すると、ファイルを開いているサーバー（＝トークンを渡されたサーバー）が、そのファイルのメタデータへの変更すべてに責任を負うこととなります。あるサーバーですでに開かれているファイルを、別のサーバーも開きたい場合、トークンマネージャーは、後者のリクエストを、すでにファイルを開いている前者のサーバーにリダイレクトし、両サーバー間で詳細を決定できるようにします。このようにメタデータのメンテナンスをSpectrum Scaleクラスタ全体で分担する方法により、効果的に拡張することが可能なのです。他のファイルストレージ技術の多くは、単一のメタデータ・サーバーや一元化されたデータベースに依存しています。そのような単一/一元化のアプローチでは、すぐにデータの格納量に限界が来てしまいます。Spectrum Scaleは、クラスタ全体に作業負荷を分散することにより、この限界を解消しています。

データ管理の簡易化

何十億ものファイルを作成するのはそれほど難しいことはありません。しかし何十億あるいはそれ以上のファイルを含むストレージ・ソリューションを維持するには、業界でも優れたツールが必要になります。Spectrum Scaleはペタバイト級のデータや何十億ものファイルを管理するのに必要となるツールを兼ね備えています。グローバルなネームスペースは、管理・操作しやすく、必要に応じてスケールアウト資源を追加するだけで簡単に拡張することが可能で、「ファイルの増殖」やそれに付随する問題を排除します。

Spectrum Scaleは簡単に使用でき、ファイルシステムの標準的な管理方法と整合性のある管理モデルを提供しています。これらの機能は、クラスタ管理や、ユーザークォータおよびスナップショットなど、他の標準的なファイルシステム管理機能に対応しています。

Spectrum Scaleでは、単一のコマンドでその機能をSpectrum Scaleクラスタ全体で実行することが可能です。また大部分のコマンドは、クラスタ内のどのサーバーからでも発行することができます。オプションとして、専用の管理サーバーを設けて、すべてのクラスタ管理タスクを実行させることもできますし、

クラスタ全体で1回のログインセッション中のみ管理者コマンドの実行を許可するように設定することも可能です。これにより、サーバーからサーバーへの管理者アクセスの範囲を減らし、高度なセキュリティを実現しています。

クォータはユーザー、グループ、あるいはディレクトリ単位で設定することが可能で、クラスタ全体におけるファイルシステムの使用状況を監視、制御することができます。クォータを使用した場合、ユーザー、グループ、ディレクトリ、inode、データブロック使用状況を含む使用状況レポートを簡単に確認することができます。さらにスナップショットを使えば、人的ミスからデータを保護することが可能です。スナップショットは、ファイルシステム全体あるいはサブディレクトリ（ファイルセットと呼ばれ、オブジェクトコンテナなど単一のデータセットが含まれる場合もある）で作成することが可能で、任意の時点でのファイルシステムのコンテンツを保存するために使用します。スナップショットには、前回のスナップショット作成時から変更があったファイルシステム内のデータコピーのみが含まれ、そのデータは以前のファイルと同じプールに格納されます。こうすることで、使用スペースを最小限に留めることができます。また同じプールを使用することで、スナップショット用に追加スペースを確保しておく必要がなく、ストレージの管理が簡易化されます。スナップショットではオンラインのバックアップ機能が提供されているため、皆さん（およびエンドユーザー）は、誤ってファイルを削除してしまった場合に簡単に復元したり、ファイルを以前のバージョンと比較したりすることができます。

大部分のアプリケーション・クラスタでは、データをクラスタから出し入れする手段が必要になりますが、Spectrum Scaleは標準的なサーバーから直接実行されているため、Spectrum Scaleファイルシステムでは様々なツールを利用してファイルデータを出し入れすることができます。エンドユーザーが、より簡単にSpectrum Scaleファイルシステムにアクセスできるよう、複数のプロトコルを通じて、クラスタ外のクライアントにファイルシステムをエクスポートすることも可能です。これには、複数のサーバーから同じデータをエクスポートすることも含まれます。

クラスタの基本的な環境設定

クラスタ環境設定のオプションに関して言えば、Spectrum Scaleはマルチ機能ツールで、ほとんどすべてのサーバーまたはクラウドに導入することが可能です。同一のSpectrum Scaleソフトウェアが、1つのクラスタ内の全サーバーに導入されます。サーバーの機能やクラスタ内での役割は、利用できるハードウェアとそこで何をやる必要があるかに基づいて決められます。ユ

ーザーがどのファイルシステム機能を必要としているかが、クラスタの環境設定に影響することはありません。

ネットワーク共有ディスク環境設定

Spectrum Scaleのクラスタでは、LAN接続アプリケーション・サーバー上で実行されているアプリケーションに高速なデータアクセスを提供するために、ネットワーク共有ディスク (NSD) プロトコルを用いたクラスタ環境設定を使用する場合があります。データはNSDサーバーからクライアントサーバーに送られます。この環境設定では、ストレージはNSDサーバーを介して、DASやSANなど複数の方法で提供することが可能です。ダイレクトアタッチやSAN等のオプションがそれにあたります。各NSDサーバーは、ストレージ全体あるいは一部に接続されます。

Spectrum Scaleは、制御情報とデータをアプリケーション・クライアントに転送するのにネットワークを使用します。このネットワークは、Spectrum Scale専用である必要はありませんが、Spectrum Scaleおよびネットワークを共有している他のアプリケーションのニーズに合った十分な帯域幅が必要です。

NSDサーバー環境設定では、サーバー・ポピュレーション (サーバー読み込み) 全体のサブセットをNSDサーバーとして定義します。NSDサーバーは、IPベースのネットワークを通じたデータブロックの抽出を担当します。I/Oがリモートであるという事実は、アプリケーションからは認識されません。図3-3は、アプリケーション・サーバーが、高速インターネット接続またはIPベースネットワーク (イーサネット等) を介して、NSDサーバーセットに接続されている環境設定例を示したものです。この例では、NSDサーバーへは共有ストレージネットワーク経由でデータが送られ、アプリケーション・クライアントへはLAN経由でデータおよび制御情報が送られています。

何台のサーバーをNSDサーバーとするかは、個々のケースにおける性能要件とストレージサブシステムの能力によって変わってきます。大量のデータ転送を必要とするクラスタには、高帯域幅のLAN接続を使ってください。高速通信を可能にするために、Spectrum Scaleでは制御情報およびデータの通信に、ユーザーの選んだ任意のハードウェア (1Gbitおよび10Gbitのイーサネット、リンクアグリゲーション、IPoIB (Internet Protocol over InfiniBand) を使用したTCP/IP、およびInfiniBand経由のリモートダイレクトメモリアccess (RDMA) に対応しています。



InfiniBandは、一般的にハイパフォーマンスコンピューティング (HPC) の実装に使用されているスイッチ型ファブリック接続です。通信速度は最大100Gbit/秒、低レイテンシで、I/Oの多い作業負荷に適しています。

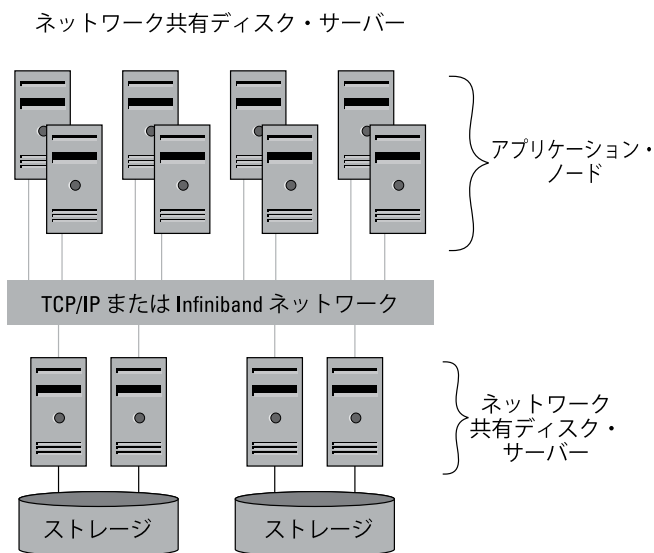


図3-3：ネットワーク I/O



Spectrum Scaleでは、クラスタ内の通信と公開ネットワークに別々のIPインターフェースを指定できる機能があります。これにより、通信トラフィックの分離を明確に定義することが可能です。NSDサーバーアーキテクチャは、NSDサーバーとアプリケーション・クライアント間に十分なネットワーク帯域幅を持つクラスタ、例えば、金融詐欺検知、サプライチェーン管理、データマイニングなどの統計アプリケーションに適しています。

複数のSpectrum Scaleクラスタにおけるデータの共有

ファイルシステムのデータは、Spectrum Scaleマルチクラスタ、あるいはアクティブファイル管理のいずれかの方法により、複数のSpectrum Scaleクラスタ間で共有することができます。

Spectrum Scaleマルチクラスタ

Spectrum Scaleマルチクラスタでは、Spectrum Scale NSDプロトコルを使用して、クラスタ間でデータを共有することができます。この機能を使って、あるクラスタの単一あるいは複数のファイルシステムに別のクラスタがアクセスすることを許可し

たり、アクセスを許可された他のSpectrum Scaleクラスタのファイルシステムをマウントしたりします。マルチクラスタ環境は、管理者が、別のSpectrum Scaleクラスタの特定のファイルシステムにアクセスすることを許可します。これにより、NFSやCIFSといったファイル共有技術よりも高い性能でクラスタ間のデータ共有が可能になります。



Spectrum Scaleマルチクラスタは、デスクトップ・アクセスや信頼できないネットワークリンクでのアクセスに最適化されたファイル共有技術にとって代わることを意図したものではありません。マルチクラスタは、単一または複数の物理的な場所に設置された複数のクラスタ間でのデータ共有に有用です。

図3-4では、クラスタAがストレージを所有しファイルシステムを管理しています。クラスタAは自身が管理しているファイルシステムへのアクセスを、リモートクラスタ（クラスタB）に付与することができます。この例では、クラスタBにはストレージがありませんが、これは要件ではありません。一般的には、クラスタが自身のストレージを持たない場合、そのクラスタは管理を容易にするためにサーバーがグループ化されたものです。リモートクラスタがデータにアクセスする必要がある場合、そのデータの所有者であるクラスタに通信し必要なセキュリティチェックをパスして、該当のファイルシステムをマウントします。図3-4では、クラスタBがNSDプロトコル経由でデータにアクセスしています。

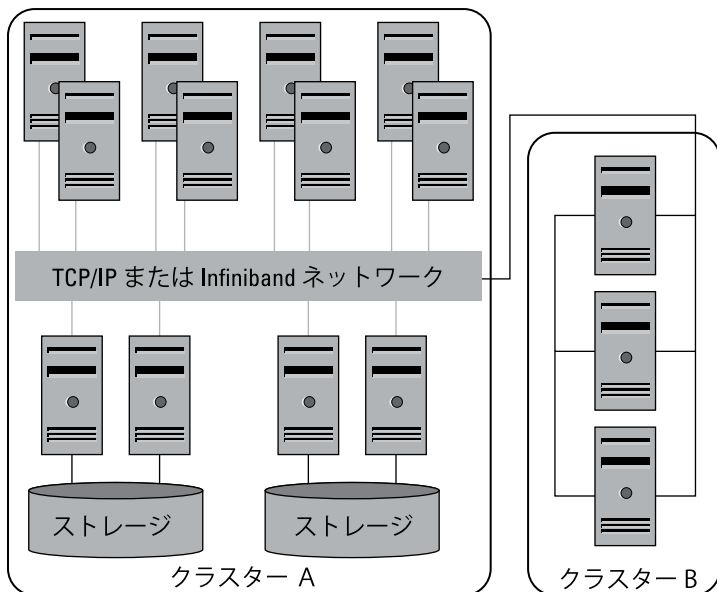


図3-4：マルチクラスタ環境設定

マルチクラスタ環境は、異なる組織に属するクラスタ間でデータを共有してコラボレーティブなコンピューティングを実行したり、管理目的でクライアントをグループ化したり、複数の場所でグローバルなネームスペースの実装を行ったりするのに非常に適しています。マルチクラスタ環境設定を使えば、データセンター内、敷地全体、あるいは信頼できるWAN上の、複数のSpectrum Scaleクラスタに接続することが可能になります。メモ：マルチクラスタ環境では、ファイルシステムにアクセスするすべてのクラスタ間において、データの強固な一貫性が保たれます。つまり、すべての更新内容は即座に全クラスタに同期されます。これは単一のデータセット上でグループがコラボレーションするのに理想的な機能です。

グローバルコラボレーションを可能に

では、ネットワークリンクが非常に長い、あるいは信頼できない場合はどうすればよいのでしょうか？ Spectrum Scaleでは、アクティブファイル管理（AFM）分散ディスクキャッシング技術により、世界中のどこからでも、低レイテンシでデータにアクセスすることが可能です。AFMは、Spectrum Scaleのグローバルなネームスペースを地理的な距離を超えて拡張し、自動的に管理することで、場所に左右されない高速な読み取り・書き込み性能を実現しています。1つの場所でデータが書き込み・変更されると、他のすべての場所では最小限の遅延で同じデータを受け取ります。従来状況を一変するこうした機能により、プロジェクトのスケジュールを加速し、世界中に散らばったチームの生産性を高めることができます。

一見ただけでは、AFMも他のキャッシュと同じように思えるかも知れません。しかし、この基本的な挙動で何ができるのかをよく考えてみると、その選択肢は多種多様であることに気付くはずで、AFMでできることの一部を理解するために、AFMがどのように使用可能かを見ていきましょう。

まず、キャッシュのリレーションシップは、ファイルセットの作成時に定義されます。各Spectrum Scaleファイルシステムは最大1,000個の独立したキャッシュリレーションシップを持つことができます。各キャッシュ・ファイルセットは、共通のストレージセットを共有しつつ、それぞれ固有のモードと環境設定パラメータを持つことができます。1つのキャッシュ・ファイルシステムのサイズは、ファイルシステム全体とすることもできますし、クォータで制限することも可能です。読み取り専用のキャッシュには、変更不可の（つまり読み取りだけができる）データが含まれます。データは、Spectrum Scaleがファイルのリストをプリフェッチ（またはプリキャッシュ）するコマンドを実行するかファイルを開くと、オンデマンドでキャッ

シュファイルセットにコピーされます。必要な作業はすべてキャッシュが行うため、1つのターゲットは、帯域幅が許す限りいくつでも読み取り専用キャッシュ・リレーションシップを持つことができます。ターゲットはそこにキャッシュがあることを認識しておらず、キャッシュのみが自身のターゲットを認識しています。また同一のファイルシステム内にある2つのファイルセットが、同じターゲットからデータをキャッシュすることが可能ですが、この場合お互いのことは認識していません。

このようにキャッシュとターゲットが切り離されていることで、このモデルは高い拡張性を実現しています。各キャッシュは1つのリレーションシップのみを追跡すればよいので、最大1,000個のキャッシュを有することが可能なのです。そして、いよいよ面白くなるのはここからです。なんと多対一のキャッシュリレーションシップだけでなく、キャッシュのカスケードも可能なのです。つまり、ターゲットは、同時にキャッシュとターゲットの二面性を持つことができます。例えば、ロンドンにあるデータが、ニューヨークでキャッシュされたとしましょう。そして東京のオフィスが同じデータのコピーを必要としているとします。この場合東京は、ニューヨークまたはロンドンのどちらか接続状態が良いほうをターゲットとして選ぶことができます。

読み取り専用モードでは、データの一貫性を保つ方法はシンプルで、データを作成・変更できるのはターゲットにおいてのみとなります。しかし、他の機能を提供している別のキャッシュモードも存在します。例えば、独立ライターキャッシュの場合、データはキャッシュおよびターゲットで、作成・変更することが可能です。ファイルがキャッシュで作成または変更された場合、書き込み作業はローカルで完了し、変更内容のみが自動的に非同期でターゲットにコピーされます。複数の独立ライターが同じターゲットのデータを更新することも可能です。ファイルがターゲットで作成または変更された場合、キャッシュは自動的に変更を検知し自ら更新します。

このような基本的なキャッシュ技術を使用することで、すべてのサイトで常に同じデータを共有することが可能です。図3-5は、AFMを使用して、すべてのデータを複数のサイトで常に同じ状態で見られるようにした例です。

この例では、それぞれのサイトがデータの1/3のターゲットとなっています。他の2つのサイトは、ターゲットに対するキャッシュリレーションシップを保持します。これにより、どのサイトにログインしても全く同じファイルシステム構成を見ることができ、すべてのデータにアクセスすることができます。ただログインしているサイトにまだデータがコピーされていない場合は、ファイルを読み取るのに多少時間がかかることはあるかも知れません。

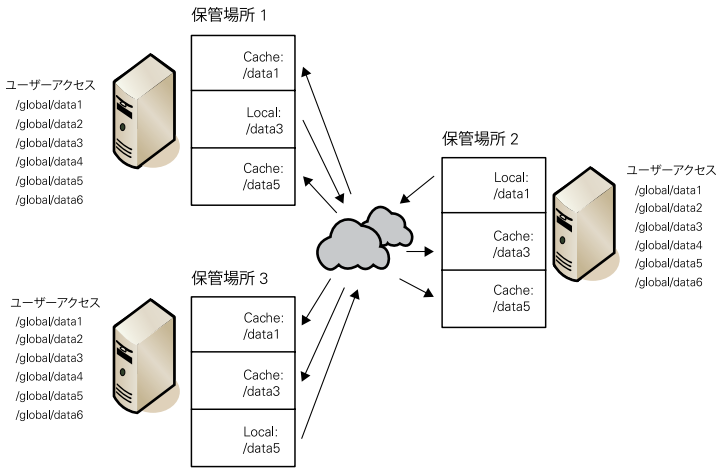


図3-5：AFMを使用したグローバルなネームスペース

データライフサイクル全体をより低いコストで管理

Spectrum Scaleは、テープを含む階層型ストレージを利用することで、情報ライフサイクル管理を強化し、データ管理コストを大幅に削減しています。強力なポリシーに基づく自動化とストレージの階層化管理により、性能、場所、コスト特性に応じてストレージ資源をグループ化して、最適化された階層型ストレージプールを作成することが可能です。テープに移行されたデータもファイルシステム内で見ることができ、エンドユーザーが直接アクセスすることが可能です。データ移行は透過的で、ディレクトリ構成におけるデータの場所を変更することなく1つのストレージプールから別のストレージプールにデータを移行します。

例えば、高性能なプールが80%埋まったらデータを移行する、といったきい値ルールを作成することができます。これにより、そのプールがボトルネックとなるリスクを軽減することができます。また、Spectrum Scaleでは、各ファイルの活動レベル（「アクセス頻度」）を監視しています。そのため、あるファイルの「アクセス頻度が上がったら」、より性能の高いストレージ（フラッシュなど）に自動的にデータを移行し、「アクセス頻度が下がったら」、容量重視のハードディスクドライブに戻す、といったポリシーを作成することも可能です。Spectrum Scaleの情報ライフサイクル管理機能とその利点には以下が含まれます。

- ✓ ポリシーに基づく自動化された階層型ストレージ
- ✓ ストレージコストをデータの価値に整合させる柔軟性
- ✓ 高性能なフラッシュドライブ、高速なディスクドライブ、大容量のディスクドライブなど、ストレージの階層を作成する機能
- ✓ 以下の機能を持つSpectrum ControlやSpectrum Archiveなど他のIBM Spectrum Storageファミリーとの完全な統合
 - Spectrum Scaleがすべてのメタデータ処理を実行し、データをSpectrum Archiveに引渡すと、Spectrum Archiveがテープに格納する
 - データは外部のストレージプールからオンデマンドで取得される（例：アプリケーションがファイルを開く）
 - ポリシーによるストレージプール間のデータの移行は、ディレクトリ構成におけるデータの場所を変更することなく行われる
 - しきい値を設定することで、例えば高性能プールが80%埋まったらファイルを移行するといったことが可能である



階層型データストレージで、データをテープに移行することにより、1テラバイト当たりのコストをディスクの約1/5にまで削減できます。

Spectrum Scaleとオブジェクトストレージ

大規模なファイルシステム管理が必要なのであれば、IBM Spectrum Scaleが最適です。Spectrum Scaleは過去10数年に渡り、世界中の何千という企業、研究機関、学術機関、政府機関で導入され、世界最大級のファイルシステムの課題に効果的に対処してきました。

オブジェクトストレージにより、今や安全でユビキタスなデータアクセスが可能となり、Spectrum Scaleも、オブジェクトストレージならではの課題に取り組むため、他のソフトウェア定義ツールを組み合わせた統合ソリューションの中で使用することが可能になっています。

Spectrum ScaleとOpenStack

さてここで、ファイルシステム管理への対応から、オブジェクトストレージおよびクラウドストレージならではの課題解決へと目を向けてみましょう。この課題に対処するには、OpenStackを話に加えなければなりません。OpenStackの主要なプロジェクトの1つがSwiftです。Swiftはオープンソースのストレージ・ソフトウェア・プラットフォームで、パブリック・クラウドおよびプライベート・クラウドで広く導入されています。Spectrum ScaleとOpenStack Swiftの主要要素を組み合わせることで、極めて重要なデータを効率的に格納、配布、保持する企業向けオブジェクトストレージ・ソリューションを実現することが可能です。

OpenStack技術は、データセンター全体におけるコンピュート、ストレージ、ネットワークの資源プールを制御する、相互に関連した一連のプロジェクトで構成されています。OpenStackは複数のコンポーネントで構成されるモジュラー型のアーキテクチャです。

- ✓ **OpenStackのコンピューティング (Nova)**：クラウド・コンピューティングのファブリックをコントロール
- ✓ **ブロックストレージ (Cinder)**：ブロッグレベルの永続的なストレージデバイスを提供
- ✓ **オブジェクトストレージ (Swift)**：スケーラブルな冗長ストレージシステム
- ✓ **イメージサービス (Glance)**：ディスクおよびサーバーのイメージ/バックアップの検知、登録、配信サービスを提供
- ✓ **ファイル共有サービス (Manila)**：ファイル共有（例：NFSおよびCIFS）管理をコアサービスとしてOpenStackに提供
- ✓ **認証サービス (Keystone)**：すべてのOpenStackプロジェクトにアイデンティティ、トークン、カタログ、ポリシーサービスを提供

OpenStackでは、データセンター全体におけるプロセッシング、ストレージ、ネットワークの資源を制御することが可能です。OpenStackでも、オープンソースのブロックストレージとオブジェクトストレージが提供されていますが、多くのOpenStack開発者は、クラウドスケールなアプリケーションに対応するためには、より強固なストレージが必要だと考えています。また、多くのOpenStack開発者が、OpenStackの処理性能と堅牢性における制限を克服してアーキテクチャを設計することは可能だと感

じているものの、ストレージに関してはレジリエンシー（回復力）と信頼性という点でハードルがかなり高いのが現状です。

IBM Spectrum Scaleでは、OpenStackの仮想マシン（VM）イメージ、ブロックデバイス、オブジェクト、ファイルといった機能が統合され、OpenStackのすべてのプロジェクトに対応しています。管理者とユーザーの認証（Keystone）、ファイル共有のプロビジョニング（Manila）、ボリュームとイメージのプロビジョニングと格納（CinderとGlance）、そしてコンポーネント間で効率的かつ迅速にデータを共有するためのファイルクローンの利用といった機能は、クラウドスケールなOpenStackアプリケーション開発者にとって非常に大きなメリットがあります。

Spectrum Scaleの堅牢性と機能を、OpenStack Swiftのオブジェクト拡張と組み合わせることで、高性能かつ高度なストレージ効率を誇る企業向けオブジェクトストアが実現されました。業界をリードするイレイジャーコーディング（消失訂正符号）アルゴリズムと圧縮、テープの組み込み、広域レプリケーション、透過階層化、暗号化、エンドツーエンドのチェックサム、スナップショットなど、今日の大部分のオブジェクトベースストレージでは実現できないソリューションを提供しています。Spectrum Scaleに組み合わせられたOpenStackは、堅牢かつ効率的なSpectrum Scaleデータプロテクションでデータを保護しつつ、同一データにオブジェクトとファイルの両方でアクセスできる統合ストレージ・ソリューションで、抜群の効率性を実現しています。またSpectrum Scaleでは、完全にレプリケーションに依存しているオブジェクトストレージシステムに比べてストレージ全体（raw storage）の量を削減することが可能です。

Spectrum Scaleオブジェクトストレージ

Spectrum Scaleは、今日最も広く使用されているオープンソース・オブジェクト・ストアであるOpenStack Swiftを組み込むことで、堅牢なオブジェクトストレージ・ソリューションを提供しています。その結果実現されたのが、Spectrum Scaleのエンタープライズ機能と共に、SwiftとAmazon S3という最も普及している2つのAPIに対応したマルチテナントなストレージシステム「Spectrum Scaleオブジェクトストレージ」です。Spectrum Scaleオブジェクトストレージは、複数のサイトに展開することが可能で、1つあるいは複数のサイトに障害が発生しても中断することなくすべてのデータの可用性を担保できる堅牢なアクティブ-アクティブ構成のクラウドストレージ・ソリューションを構築します。例えば、

3つのサイトがあった場合、Spectrum Scaleはそのうち2つのサイトにオブジェクトを格納して、1つのサイトがダウンしてもデータの可用性が担保されるようにしています。

OpenStack Swiftは堅牢なオブジェクトレイヤーを提供しています。また活発な開発者コミュニティでは、バージョンング、レート制限、失効といった革新的な新機能が次々と開発されています。さらに、標準APIおよびSDKを採用しているため、Spectrum Scaleのユーザーは、自身の環境と要件に合わせてソリューションをカスタマイズすることが可能です。ソリューションを構築するのに、Spectrum ScaleとSwiftのどちらのコードも変更する必要がなく、Swiftパッケージとの長期に渡る適合性が確保されています。また、Spectrum Scaleは、CLIや管理・監視CUIを含む堅牢なデプロイメント・管理ツールを提供しているため、OpenStack Swiftのデプロイ・管理についても心配する必要がありません。

図3-6は、Spectrum Scaleと、オブジェクトストレージを構築するSwiftを含む各種OpenStackコンポーネントを組み合わせた基本的なアーキテクチャ例です。一番上に表示されているのは、Spectrum Scaleのデータにアクセスする可能性がある様々な種類のアプリケーションです。ユーザーアプリケーション、モバイルアプリ、有名な分析ソフトウェア基盤「Hadoop」などがこれに当たります。

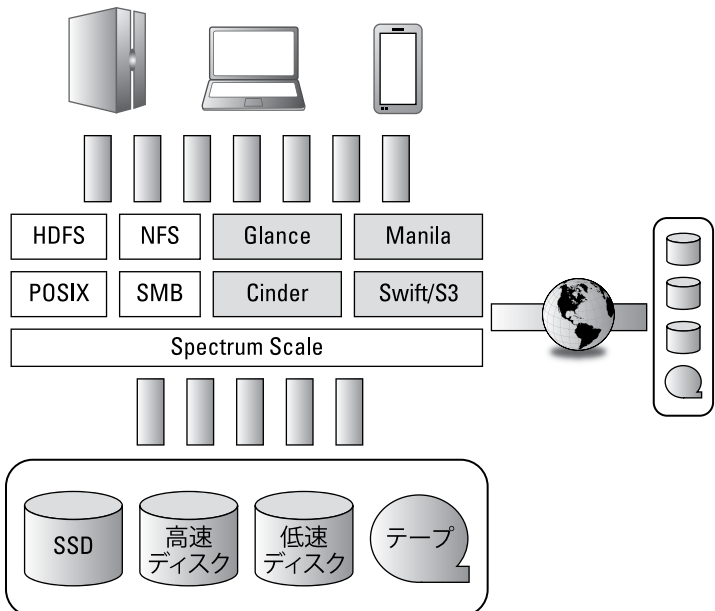


図3-6： Spectrum ScaleとOpenStackを組み合わせたアーキテクチャ例

Spectrum Scaleとオブジェクトストレージを組み合わせたアーキテクチャはシンプルです。ユーザー/クライアントアプリケーションは、高速なイーサネットネットワークを介してSpectrum Scaleにアクセスします。Spectrum Scaleの環境設定の大部分は、OpenStackとの組み合わせにおける使用からは独立しています。そのため、すでにデプロイしているSpectrum Scaleがある場合、Swift等のOpenStackコンポーネントは、新しいSpectrum Scaleの独立ファイルセットにデータを格納した状態で、既存のSpectrum Scaleの上に実装することが可能です。

Spectrum Scaleは多種多様なアプリケーションに向けたデータを格納できるため、オブジェクトストアのデータは独立ファイルセットと呼ばれる別個のSpectrum Scale管理エンティティに格納するのが賢明です。そうすることにより、スナップショットやバックアップといったSpectrum Scale情報ファイルサイクル管理において、オブジェクトストアデータを一意に特定して管理することが可能になります。

物理的なストレージアーキテクチャもおおむね独立していますが、効率性、耐障害性、性能には影響します。ストレージには大量のストレージコストがかかりますから、ストレージハードウェアの選択が、システムの費用対効果を左右します。アーキテクチャ図の中で、物理ストレージの例として示されている通り、SSD（フラッシュドライブ）、高速ディスク（SAS）、低速ディスク（SATA）、テープのすべてをデプロイすることが可能で、多岐に渡る作業負荷要件に対応することが可能です。

Spectrum Scaleオブジェクトストレージのアーキテクチャ例における要点は以下の通りです。

- データの保護はSpectrum Scaleが行い、Swiftは各オブジェクトの単一インスタンスをファイルシステムに書き込む
- Spectrum Scaleの追加ノードにより、オブジェクトアクセスの帯域幅とノード障害時の耐障害性が追加される。
- 各Spectrum Scaleサーバーは、すべてのオブジェクトに独立してアクセスすることができる
- 追加の容量は簡単にデプロイ可能なため、容量を確保するためのオーバープロビジョニングによるコスト増加を回避できる。
- すべて（あるいは少なくともほとんど）の既存のSpectrum Scaleエンタープライズ機能が、オブジェクトストレージ・ソリューションで使用可能である。

単一ストレージシステムでファイルとオブジェクトに対応

ファイルストレージとオブジェクトストレージは、アプリケーションの要件によってそれぞれ利点があります。ファイルが必要としているアプリケーションが、オブジェクトの意味/仕組みに悩まされたり、またその逆があったりしてはならないことを覚えておきましょう。



Spectrum Scaleはネイティブなファイルおよびオブジェクトのアクセスに対応しており、コストをかけてアプリケーションを書き直す必要がありません。ネイティブ・プロトコルへの対応により、プロトコル間でコピー・解釈を行うゲートウェイに伴う複雑性やオーバーヘッドも排除することができます。Spectrum Scaleは、ファイルとオブジェクトのすべての利点を、単一のシステムで実現しており、プロトコルの階層化に悩む必要もありません。

今日のアプリケーションの大部分は、ファイルを前提として作られていますから、単一のストレージシステムでファイルとオブジェクトの両方にアクセスする必要がある利用方法は多々あります。典型的な利用方法としては以下が挙げられます。

- ✓ ファイルとして作成されたデータを、オブジェクトを使用してグローバルで共有する（画像、動画、記録、バックアップなどのコンテンツレポジトリは、この方法で構築されることが多い）
- ✓ 多くのソースからオブジェクトとしてデータを収集し、SAS、Hadoop、Spark、その他の分析フレームワークを使用して分析する（高い性能を実現するには、ファイルインターフェースでネイティブに分析を実行することが不可欠）

要点

Spectrum Scaleオブジェクトストレージは以下のような環境に理想的です。

- ✓ すでにSpectrum Scaleを使用しており、既存のデータと共にオブジェクトストレージへの対応策を探しているサイト
- ✓ OpenStackプライベートクラウドの構築
- ✓ SwiftやS3など、RESTインターフェースを使用してスケールアウトする新しい分析作業とウェブサービス
- ✓ 従来のストレージシステムに比べ、コストが低く拡張が容易な大容量ストレージの導入

- ✔ 同一あるいは別々のデータセットにおいて、ファイルとオブジェクトの両方が要件となるサイト
- ✔ 1つあるいは複数のサイトで、企業用途に対応した、コスト効率が良く、高性能なオブジェクトソリューションを探している組織
- ✔ この利用方法では、Spectrum Scaleは、最初はオブジェクトストアにのみデータ管理を提供するが、ユーザーには、要件の変化に伴い別のタイプのアプリケーションへ拡張するオプションも提供される

Spectrum ScaleとOpenStack Swiftを組み合わせたオブジェクトストレージが、他のオブジェクトストレージと異なる点は、以下のような機能を要求する作業に対して業界をリードするソリューションを提供できる点です。

- ✔ データの損失を防ぐためのスナップショットやバックアップへの対応
- ✔ 高いスループットでのオブジェクトへのアクセス
- ✔ 大規模な容量、高密度なストレージ
- ✔ 今後何年にもわたりIT基盤に求められるデータの拡張性
- ✔ IDの一元管理が可能な、ファイルとオブジェクトの統合ストレージ

Spectrum Scaleオブジェクトストレージは、オブジェクトストレージを、「企業内の困難なハードル」から、「革新を実現し、優位な競争力を獲得する機会」へと変えました。IBM Spectrum Scaleをすでに導入されている場合は、OpenStack Swiftを追加してこの最先端のソリューションを構築するのは朝飯前です。ほぼゼロから構築する場合も、IBMがこれまでOpenStackのイニシアチブとコミュニティに提供してきたコミットメントとサポートが、この相互運用性に実績のあるフル機能のオブジェクトストレージ導入にあたり、皆さんの組織の力になることは間違いありません。

第4章

IBM Elastic Storage Server について学ぶ

本章の内容

- ▶ IBM Spectrum ScaleのElastic Storage Server (ESS) ソリューション
- ▶ ESSの主要な耐障害機能

IBM Spectrum Scaleは、ストレージ容量を確保するために、クラウドまたはサーバーおよびストレージハードウェアに導入されるソフトウェアソリューションです。Spectrum Scaleは任意の標準サーバーおよびストレージハードウェアで展開することが可能ですが、多くの場合、ハードウェアとSpectrum Scaleを組み合わせた統合ソリューションが選ばれています。統合ソリューションは簡単に展開でき、確実な性能が得られる上、自社で構築するリスクを負う必要がありません。

本章ではIBM Elastic Storage Server (ESS) を紹介します。ESSはIBM Spectrum Scaleに最適なビルディングブロックです。ESSでは、IBM Power Systemsサーバーの性能とSpectrum Scaleソフトウェアを組み合わせ、高性能で拡張性に優れたビルディングブロックアプローチを提供することで、現代におけるファイルストレージとオブジェクトストレージのニーズに応えています。また、組織の現在のニーズに合った環境設定から始め、後からESSを追加することで容量と帯域幅を拡張し、将来的なニーズに応えることも可能です。

ディスクドライブの性能は長い間大きく変わっていません。しかし過去10年でディスクドライブの容量はディスク当たり最大100GBから6,000GB超へと増加しました。実に60倍にも増えているのです。実際、8TBのディスクはすでに実用化されていますし、10TBのドライブも実用化間近です。参考までに、例えば1つのディスクドライブを1秒当たり60MBの速度で読み取れる

と考えてください。2003年には、1つのディスクドライブをバックアップするのにかかる時間は28分でした。しかしディスクドライブの処理速度は2003年からそれほど変わっていないため、今日の大容量ディスクドライブをバックアップしようと思ったらほぼ1日かかります。この性能の問題は、RAID再構築など、ディスクの保守作業にも影響しています。8TBのディスクがダウンしたら、データを読み書きして、新しいディスクに交換するまでに一体どのくらいかかるのでしょうか？ IBM Elastic Storage Server (ESS) は、主にこうした性能の問題を解消することが原動力となり開発されました。

エンドツーエンドの ESSソリューションを提供

ESSでのIBM Spectrum Scaleの実装では、1つ以上のElastic Storage Serverを導入し、各サーバーがSpectrum Scaleソフトウェアを実行して、ストレージ管理とストレージ容量を共有します。このようなビルディングブロック（またはグリッド）アーキテクチャのアプローチを採用することにより、Spectrum Scale全体の実装に多大なレジリエンシー（回復力）が付加されるだけでなく、ESSビルディングブロックの追加による容量と性能の直線的なスケーリングなどの多くの利点が得られます。

個々のElastic Storage Serverは、IBM Power Systemsサーバーおよびストレージメディア（フラッシュまたはディスクドライブ）で構成されており、すべてのソフトウェアとネットワークコンポーネントが含まれた統合ソリューションになっています。IBMは各ESSの信頼性、相互運用性、性能をテスト、最適化し、企業が簡単にシステムを導入し、ビジネス目標の達成に貢献できるようにしています。

Spectrum Scale ESSソリューションには以下が含まれています。

- ストレージにおける、スケーラブルなビルディングブロックアプローチ
- フラッシュおよびディスクドライブの多様なストレージ容量に対応するための複数の環境設定
- GL6モデルで最大348ドライブ
- FDR InfiniBand、10Gbitおよび40Gbitイーサネットインターコネクト（LANケーブル相互接続）、またはその両方

- ✔ イレイジャーコーディング（消失訂正符号）を使用した Spectrum Scale RAID技術による、持続可能かつ予測可能な性能と迅速な再構築
- ✔ ハードウェアコントローラー不要、ディスクのブリーディングおよび管理はSpectrum Scale RAIDソフトウェアが担当



Spectrum Scaleの実装にESSアプローチを採用することで、多くの利点があります。

- ✔ **データの冗長性**：Spectrum Scale RAIDは、非常に信頼性の高い2耐障害性（2台の障害まで回復可能）および3耐障害性（3台の障害まで回復可能）のリード・ソロモンベースのパリティコード（誤り検出符号）と、3-wayおよび4-wayのレプリケーションを提供しています。
- ✔ **大容量キャッシュ**：ESSでは、Power Systemsサーバーの大容量メモリキャッシュを、内部および外部のフラッシュデバイスと組み合わせて使用することで、ニアラインSASドライブの非効率性と長いレイテンシを解消しつつ、その高密度性を活用しています。
- ✔ **グラフィカルユーザーインターフェース（GUI）**：ESSでは、直観的なGUIを使って、ローカルまたはリモートでシステムを管理、監視することができます。
- ✔ **高度なストリーミング性能**：25GB/秒超の性能を持続して提供することができます。
- ✔ **拡張性**：インストール済みの環境にサーバー環境設定が追加されると、容量、帯域幅、性能、シングルネームスペースのすべてが拡張します。つまり、初期のインストールは小規模で始めることができ、データのニーズが大きくなると共に拡張することが可能だということです。

ESSはSpectrum Scaleに最適のストレージビルディングブロックです。

- ✔ 各ESSモデルは、バックエンドストレージと組み合わせるのに最適な処理能力に調整されています。
- ✔ ESSでは、データ移動における高い処理能力とI/O帯域幅を実現するために設計されたIBM Power Systemsサーバーを活用しています。
- ✔ Spectrum Scale RAIDソフトウェアが標準装備されています。



ESSでIBM Power Systemサーバーを使用することにはいくつかの大きな利点があります。

- 12コアプロセッサあたり96ハードウェアスレッド、1コアプロセッサあたり8ハードウェアスレッド
- CPUの（ピーク）メモリ帯域幅は最大410GB/秒
- 一般的なx86システムと比較して、I/O帯域幅は最大5倍、キャッシュ3倍、メモリ帯域幅は実に4~6倍
- Javaにおけるコア当たりの性能はIntelと比較して最大2倍
- 圧縮のオフロードでさらにシンプルに

図4-1は、ESSで使用されるハードウェアおよび異なる出荷形態を示したものです。

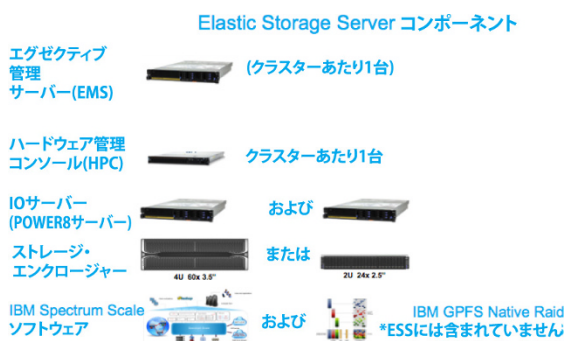


図4-1 : IBM Elastic Storage Server

ESSには、高い可用性を確保するために常に2台のサーバーと、2U（24×2.5インチディスク）または4U（60×3.5インチディスク）構成のディスクエンクロージャー（JBOD：Just a Bunch Of Disks、ただのディスクの束、の意味）が含まれます。ESSには、2U JBODまたは4U JBODを使用したGSまたはGLという2つのモデルがあります。各モデルで、JBOD数またはタイプ、ディスクドライブの容量を選択することが可能です。



RAID（Redundant Array of Independent Disks：複数の個別ディスクから成る冗長アレイの意、元々はRedundant Array of Inexpensive Disks：複数の安価なディスクから成る冗長アレイの意だった）は、データを複数のドライブに分散するデータ格納技術です（RAID levelsと呼ばれる）。データを分散する方法は、必要とされる性能と保護レベルに応じて複数あります。

ESSはストレージ装置に勝るのか？

いいえ、勝りません。まず、物理プロセッサは高速です。とても適いません。しかし、問題を解決するのに全く新しいアプローチを取ることはできません。ESSはSpectrum Scale RAIDという新しいアプローチを取り入れ、大規模ディスクにおけるRAID構築に長時間かかる問題を解消しました。

例えば、従来のRAID 6アレイでは、複数のディスクがグループ化され、データはそれらのすべてのディスクに分散して書き込まれます。その時データ共にパリティビットという情報が一緒に書き込まれます。パリティビットはドライブが故障した場合にデータを「再構築」するために使用される情報です。RAID 6アレイで、1台のドライブが故障した場合、他のドライブのパリティビットが読み込まれ、故障したドライブのデータはそれを基に再構築され、新しいディスクに書き込まれます。故障したディスクのすべてのコンテンツが新しいディスクに書き込まれたら、アレイが完全に復旧し、別の故障が起きても保護される状態に戻ります。

このディスクの再構築には長い時間がかかり、その間データの保護レベルは下がります。しかしSpectrum Scale RAIDでは、イレイジャーコーディング（消失訂正符号）技術を採用しており、Spectrum Scale RAIDグループ内の複数のディスクが動的に使用され、データが入ってくると、パリティビットはすべてのディスクにストライプ方式で分散されます。ファ

イルデータの各ブロックは、イレイジャーコーディングを使用して、ストライプと呼ばれる単位に分けられます。そのため、例えば8D+2Pの場合、1つのブロックは8つのデータストライプと2つのパリティストライプで構成されます。このストライプ数はディスク数とは関係ありません。例えば58台のドライブがあるとして、8D+2Pを使用している場合、データの最初のブロックが10台のドライブに、2つ目のブロックが次の10台に…という具合に分散されます。

では、こうすることで、なぜディスク再構築の時間が短縮されるのでしょうか？ Spectrum Scale RAIDでは、RAID 6のように故障したディスク全体を再構築する必要はなく、故障したドライブに格納されていた全データが、他のドライブでコピーまたは計算（パリティを計算）されるようにするだけで良いのです。つまり、ESSは物理プロセッサに真っ向から挑むのではなく、作業を分散して対処しているのです。この方法だと、4TBドライブが故障した場合、データ全体を新しい1台のドライブに書き込むかわりに、残りの57台のドライブでただ一部のデータを入れ替えるだけですみます。

RAID 6アレイで、4TBドライブを再構築する場合、17～24時間もの間、残りのストレージの性能に影響が出る可能性があります。しかしイレイジャーコーディングを使用したSpectrum Scale RAIDでなら、復旧作業は15分かかりません。

並列処理能力を耐障害性に活用

復旧のスピードおよびその他の高度なデータ保護機能により、ESSは、ドライブの数が増えた場合でも、高い信頼性を確保することができます。このストレージ機能の中核となっているのが、Spectrum Scale RAID技術と呼ばれるESSの主要機能です。Spectrum Scale RAID技術はドライブの数には依存しません。ESSの標準的な環境設定はあるものの、そのデータ保護/RAID技術は、最低11ドライブから、最大数百ドライブまで適用可能です。この特徴により、新しい様々なドライブパッケージやドライブタイプで、ESSを活用することができるのです。

高度なデータ保護

高度なデータ保護は、データの整合性とディスクの健全性という2つの要素で成り立っています。

データの整合性

これまで長い間、Spectrum Scaleは「先週ディスクに書き込んだ内容が今週正しく読み取れるように担保する」といったことを、RAIDコントローラーにまかせてきました。自らRAWデバイスに至るまでのデータを管理することには、より大きな責任と素晴らしい利点が伴います。その責任の1つが、データがディスクに書き込まれてから変更がないように担保することです。ESSでは、データチェックサムを使用してソフトウェアでこれを実現しています。チェックサムとは、生データのデータ列から算出する値のことです。この値は、隔離されたエリアに格納され、データが読み込まれる際に、そのデータが書き込まれた時点から変更されていないことを確認するために使用されます。チェックサム情報が使えるようになったことで、Spectrum Scaleは、ディスクからSpectrum Scaleアプリケーションクライアントまですべてのデータの整合性を担保し、ストレージ、システム、ネットワークのエラーからデータを保護することができますようになりました。

ハードディスクでは、読み込みエラーをレポートしなかったり、実際にはデータが書き込めていないのに書き込めたとしたりといったことがまれに発生します。こうしたエラーはサイレント障害、ファントム書き込み、ドロップド書き込み、オフトラック書き込みなどと呼ばれます。Spectrum Scale RAIDでは、データのサイレント障害を検知するために計算・付加されたエンドツーエンドのチェックサムを実装しています。

チェックサムまたはバージョン番号が読み取りにおいて無効だった場合、Spectrum Scale RAIDはパリティまたはレプリケーションを使用してそのデータを再構築し、再構築したデータと新しく生成したチェックサムをクライアントに返します。このようにして、ディスクの読み取りおよび書き込みにおけるサイレント障害が検知され、修正されています。

ディスクの健全性

Spectrum Scale RAIDが担うようになったもう1つの責任が、ディスクの健全性を保つことです。全てのストレージコンポーネントは、Spectrum Scaleの「ディスクホスピタル」機能により定期的に監視、保全されています。メディアのエラーやバスの問題があったり、単にディスクの挙動がおかしかったりした場合、Spectrum Scaleはこれらのイベントに対して、リクエストの再ルーティングやディスクの電源切断・再投入といった一連のアクションで対応します。

ディスク処理能力の有効活用

従来のRAIDに比べ、Spectrum Scale RAIDは洗練されたデータ配置の仕組みを実装しており、ユーザーのデータと冗長情報をESSストレージアレイのすべてのディスクに均等に分散しています。

Spectrum Scale RAIDでは、稼働するのを待っているアイドル状態のディスクというのは存在しません。予備のディスクも含め、利用可能なすべてのディスクにデータを分散することで、「ホットスペア」を「ホットスペアスペース」に変えることができます。Spectrum Scale RAIDでは、1つのドライブを予備として割り当てるのではなく、ディスクグループのすべてのドライブで、障害イベントのためのスペースを確保しています。

Spectrum Scale RAIDは、ディスク障害からの復旧時間を大幅に短縮し、クライアントアプリケーションにおける再構築コストを削減しています。ディスクが故障した場合、データはアレイ内のすべての稼働ディスクを使用して再構築されるため、従来のRAID 6と比較して、大幅に向上した処理能力を提供することが可能になっています。

ディスクホスピタル機能は、Spectrum Scale RAIDの主要機能であり、ストレージサブシステムのエラーや障害を非同期に診断します。Spectrum Scale RAIDでは、個々のディスクI/O処理が10秒を経過するとタイムアウトするようになっており、障害のあるディスクがクライアントI/O処理へ影響を与えないようにし

ています。障害の疑いがあるディスクは即座にディスクホスピタルに「入院」となり、エラーがディスク自体によるものなのか、パスに原因があるのかが診断されます。ホスピタルがエラーを診断している間、Spectrum Scale RAIDは冗長コードを使用して、消失あるいは削除されたデータストライプを再構築するため、I/O処理に疑わしいディスクを使用しないで済むのです。

すべての機能を集約

Elastic Storage Serverは、Spectrum Scaleの「適応性」という特長を踏襲したストレージ機器です。Spectrum Scale RAIDソフトウェアはすべて、既存のSpectrum Scale機能の追加機能として存在します。つまり、Spectrum Scale RAIDの新しい機能だけでなく、クラスタ内で任意のストレージを組み合わせる機能を始めとした、すべてのSpectrum Scaleの機能を引き続き使用できるということです。既存のクラスタ、さらには既存のファイルシステムにおいて、ESSでネットワーク共有ディスクプロトコルを使用することも可能です。

Spectrum Scale RAIDは、すでにこの分野で実績を上げており、高いI/O作業負荷が要求されるファイルストレージ/オブジェクトストレージで成功を収めています。ESSはSpectrum Scaleソリューションのアーキテクチャ全体において、礎となる重要な構成要素です。

第5章

IBM Spectrum Scale オブジェクトストレージの 10の利点

本章の内容

- ▶ IBM Spectrum Scaleオブジェクトストレージの利点
- ▶ IBM Spectrum Scaleでストレージの課題を解消する方法

ファイルおよびオブジェクトストレージの課題は多種多様です。規模、性能、リテンション、アクセス、可用性、容量に関連してそれぞれ個々の要件が存在します。本章では、こうしたファイルおよびオブジェクトストレージの課題を解消する助けとなるSpectrum Scaleオブジェクトストレージの10の利点（いえ、実はおまけで11あります！）を説明していきます。

ファイルストレージとオブジェクトストレージの統合

Spectrum Scaleでは、単一のファイルシステムでほぼあらゆるプロトコルを使用することが可能で、アプリケーションは、ファイルベースの情報、オブジェクト、さらにはブロックベースのデータまで、多彩な種類のデータを格納することができます。複数のプロトコルがネイティブで対応されているため、各アプリケーションに最適なプロトコルを選択することができます。



オブジェクトストレージとファイルストレージをSpectrum Scaleで統合することにより、ストレージ基盤全体においてコストと効率性で大きな効果があります。単一のストレージ・アー

キテクチャを実装するだけで、すべてのストレージ要件に対応できるためです。この統合ソリューションにより、企業全体の情報を、シングル・ネームスペースに格納することができ、それらの情報は、設定したポリシーに応じて、任意の地理的ロケーションに複製されます。またデータベースからリッチメディアのファイル提供まで、多彩なアプリケーションに対応できる上、速度向上とコスト削減を実現します。



ファイルストレージとオブジェクトストレージを1つのシステムに統合する利点についてもう少し深く考えてみると、すべてのデータタイプを共通管理ドメインに統合することで、異なるタイプの情報にアクセスできることに気付くはずですが、例えば、Spectrum Scaleファイルインターフェースでデータを作成し、そのデータを、オブジェクトインターフェースを使ってグローバルで共有するといったことが可能です。また逆の場合では、データ分析を大幅に強化することができます。データをオブジェクトとして読み込み、ファイルインターフェースを使用してHadoopなどの分析エンジンで利用できるようにします。

高度なデータ保護

データ保護サービスにSpectrum Scaleを使用することにより、以下のような形で、オブジェクトストレージシステムの効率性および性能を上げることができます。

- ✓ Spectrum Scale RAIDにより、ストレージの効率性は33%から最大80%に増加
 - 「遅くて分厚い」ディスクドライブや、さらにはテープといった安価なストレージが活用可能
 - 従来のRAIDスキームと比較して再構築時間を格段に短縮
- ✓ ファイルとオブジェクトの両方で、ポリシーに基づくスナップショットが利用可能
- ✓ ディスク障害復旧時にデータがストレージ・ネットワーク上に溢れ出すことがないため、ネットワーク・トラフィックを削減。復旧は透過的に実施され、アプリケーションへの影響は最小限
- ✓ Spectrum Scaleのストレージサーバーへの書き込みは、各オブジェクトの単一コピーのみのため、アプリケーションはストレージネットワークの帯域幅をフルに活用可能

- ✓ Spectrum Scaleのデータ分散配置により、オブジェクトの最大サイズは5TBまで増加。大規模なオブジェクトによりストレージ容量の不均衡やサーバーのホットスポットが生じることはなく、利用可能なネットワーク帯域幅を効率的に使用可能
- ✓ 単一クラスタ内でデータを複製するため、別のレプリケーション・ネットワークは不要
- ✓ Spectrum Scaleにおけるノード障害では、ノード間またはストレージ媒体間でのデータ移動が不要

アプリケーション性能の向上

データインテンシブ・アプリケーションとは、作業を完了するために大量のデータを読み書きする必要があるアプリケーションのことです。データインテンシブ・アプリケーションの速度を向上させるのは簡単です。より高速なストレージ基盤を使用すれば良いのです。しかし実際には、言うほど簡単ではありません。多くのストレージ・ソリューションはそれほど効率的にスケールできないためです。ストレージ装置が拡張できない場合に、I/O性能を向上させるためには、多数の小さなストレージプールや装置を使用し、アプリケーションを調整して、それらを同時に使用するようになるしかありません。

プロジェクトのデータが複数のプールや装置に分散されていない場合、アプリケーションが進化するにつれ、データのホットスポットが発生する場合があります。データのホットスポットは、アプリケーションやクライアントにより、単一の装置あるいは単一のストレージプールまたは資源内に格納されているデータセットに対して高いI/Oが必要とされる場合に発生します。Spectrum Scaleは、利用可能なすべてのストレージ資源にデータを分散することで、ホットスポットの発生を防ぎます。また、データを、フラッシュなどのより高速なストレージに移動することも可能です。さらに高度なデータキャッシング機能でより高速なアクセスも実現します。

Spectrum Scaleは、多数のプロセッサおよびサーバーが、共通のファイルデータセットに同時にアクセスすることで発生するI/O負荷に対応できるように設計されています。Spectrum Scaleは、多数のディスクや、フラッシュアレイ内のモジュールといったストレージ資源に、データをブロックとして細分化して格納します。使用されているファイルセットにおいてデータを細分化することで、Spectrum Scaleで構成されたストレージは十分に性能を発揮することが可能になります。

多くのストレージ・ソリューションでは、全体のスループットを向上するために、異なるストレージプールまたはストレージ装置をまたいでデータを配置する必要があります。しかし、このアプローチでは、タスクの管理が非常に複雑になり、ストレージ資源を常にフル活用するのが難しくなります。Spectrum Scaleはその逆で、すべてのストレージ資源の性能を十分に活用することが可能です。Spectrum Scaleは利用可能なストレージすべてにデータを分散するため、稼働待ちのアイドル状態となる資源が存在しません。つまり資源にかけた投資が無駄になることがありません。

例えば、エンターテインメント業界では、有名人のニュースが流れると、瞬時にして大きな関心が集まり反響が出ます。ファンは競ってその人物の写真や映画をダウンロードしようとするでしょう。すると、その有名人に関連したデータセットが「ホット」になり、性能が落ちてしまう可能性があります。一部のストレージソリューションでは、ファイル・レプリケーションやDRAMキャッシングといった手法を用いて、こうした急激な負荷の上昇に対応しています。

ただレプリケーション・ソリューションにおける課題は、複製されたファイル用の余分なストレージ容量と、複製を作成するための追加の帯域幅が必要になる点です。またストリーミング・メディアの場合、各ファイルに対する負荷は時間と共に変化します。そのため継続的にレプリケーションを実行しなければなりません。DRAMキャッシングは、何千というメディアストリームが同時発生する巨大なサーバーの場合のみコスト効率が良いと言えるでしょう。

Spectrum Scaleでは、クラスタ内のすべてのサーバーが、全システムデータおよびメタデータに均等に並行してアクセスすることができます。これにより、アプリケーションのI/O処理が高速化されるため、メタデータへの参照が頻発するアプリケーションの性能が向上します。Spectrum Scaleでは、クラスタ内の任意のサーバーが、任意のストレージで読み取り・書き込みすることが可能なため、同時にI/Oを実行するアプリケーションも、非常に高いデータアクセス率を得ることができます。

Spectrum Scaleの多彩な機能

Spectrum Scaleでは、オブジェクトストレージにおいて、以下のような堅牢なストレージ管理機能を多数利用することができます。

- ✓ シンプルなGUI
- ✓ グローバルなネームスペース

- ✔ 暗号化
- ✔ バックアップ
- ✔ 災害復旧
- ✔ ダイナミックな階層化
- ✔ テープの統合
- ✔ リモート・キャッシング

従来のシステムに比べて、Spectrum Scaleのデータ管理では、より広範で強力な機能を提供しています。これまでのストレージ・システムは、ファイルデータに関して、I/Oサイズ、ストレージタイプ、レイテンシ、ロケーションを含む4~5個の属性しか「認識」していませんでした。しかしSpectrum Scaleはそれよりもはるかにインテリジェントで、ファイルの属性を50個近く認識することができます。こうした属性には以下が含まれます。

- ✔ ファイル名とファイルタイプ
- ✔ レイテンシとロケーション
- ✔ ファイル作成時刻
- ✔ ファイルへの最終アクセス時刻と最終更新時刻
- ✔ アクセス許可
- ✔ ファイルの「アクセス頻度」
- ✔ ファイルの所有者であるグループおよびユーザー
- ✔ ファイルのスペース効率
- ✔ メタデータの最終更新時刻

これらの属性が認識されていることにより、データをきめ細やかかつ効率的に制御して管理することが可能になります。これらの属性の1つあるいは複数を用いて、ストレージシステムのポリシーを設定し、特定のファイルやデータセットに適用することができます。例えば、ポリシーを作成して、ある営業所のストレージ性能を向上させたり、特に慎重に扱う必要がある情報へのアクセスをさらに制限してセキュリティを強化したりすることができます。オブジェクトストレージ環境では、オブジェクトのメタデータがストレージ管理における主要な要素となるため、使用や性能に関連した属性を使用することで、環境を強化することができるのです。

実績のある製品の信頼性と回復力

Spectrum Scaleは実績のあるストレージ・プラットフォームで、何千という顧客の環境で導入されており、何年も前から実用されています。

停止時間は、計画したものであれ、突然のものであれ、予算的な意味でも、ITスタッフの時間外作業という意味でもコストがかかります。アプリケーションが常にデータにアクセスできるようにするためには、堅牢で可用性の高いストレージ・ソリューションが必要です。アプリケーションは、ハードウェアに障害があった場合も、またストレージ容量の拡張やハードウェア・ソフトウェアの更新などによるシステム変更の間も、常にデータにアクセスし続けられる必要があります。

Spectrum Scaleソフトウェアは耐障害性があり、サーバーやさらにはストレージ・システムが故障した場合でも、機能し続けます。他のノードはクラスタとファイルシステムコンポーネントのヘルス状態を継続的に監視します。障害が検知されると、適切な復旧アクションが自動的に実行されます。高度なログ取得と復旧機能により、アプリケーション・サーバーによるロックやクラスタサービスの実行が失敗した場合にもメタデータの一貫性は保たれます。

さらなる保護機能として、Spectrum Scaleでは、メタデータやファイルデータの複製を同時に2個または3個作成するよう設定することが可能です。データが複製されたら、どの複製も、アクティブ-アクティブモードでアクセスし読み取ることが可能です。またSpectrum Scaleでは、管理者がどの複製を読み取るかを決定することができるため、サイトをまたいで複製している場合でも、読み取りアクセスをローカルに維持することができます。これは、読み取り負荷が高い場合に効果があり、WANのトラフィックを軽減することにつながります。両方の複製が同じデータセンターに格納されている場合、読み取りは両方から行われるため、読み取りの処理速度が2倍となる点も注目に値します。



Spectrum Scaleは、アクティブ-アクティブ構成でのファイルベースのリプレーションのように、データを管理する新しい方法を導入しています。そのため、異種混合のハードウェアでデータを安全に保管する柔軟性と、同時に使用する機能を提供しています。

データセキュリティの向上

データは保護されなければなりません。そしてSpectrum Scaleはデータを安全に保つためのツールを提供しています。米国においては、HIPAA（米国における医療保険の相互運用性と説明責任に関する法令）やサーベンス・オクスリー法といった政府の規制により、データアクセスの制御、暗号化、情報の適切な削除に関する要件が定められています。

セキュリティはデータへのアクセスを制御するところから始まります。Spectrum Scaleは、オブジェクトストレージへのアクセスに、安全かつ暗号化された、HTTPSを使用したRESTful APIを提供しています。データにアクセスするためのユーザー認証は、ファイルプロトコルまたはオブジェクトプロトコルを使用して行われ、LDAP（Lightweight Directory Access Protocol）またはアクティブディレクトリ（Active Directory）で一元管理することが可能です。そのため、IT部門が既存のLDAP/AD環境にSpectrum Scaleを導入しやすくなっています。

ストレージシステム内で、現在アプリケーションやクライアントによってアクセスされていないデータは、保存状態のデータと呼ばれます。Spectrum Scaleはこのデータを暗号化することにより、他のアプリケーションや管理者による、認可されていない読み取りや書き込みからデータを保護しています。Spectrum Scaleはセキュリティ侵害や未認可アクセスからもデータを保護しており、万が一データが紛失、窃盗、不適切に破棄された場合にも流出するのを防ぎます。

Spectrum Scaleは、米国連邦政府国立標準技術研究所（NIST SP 800-131A）、HIPAA、欧州連合、サーベンス・オクスリー法および連邦情報処理標準（FIPS 140-2）で推奨されているベストプラクティスに準拠し、ディスク、フラッシュ、またはテープにデータを格納する前に、暗号化を行っています。ファイルはディスクに格納される前に暗号化され、暗号化のマスターキーがファイルシステムのディスクに書き込まれることはありません。

また、多くの国において、政府の規制により、復元できない方法によるデータの削除や破壊が要件とされています。単にストレージを上書き可能にするのではなく、データを完全な状態で保持しているSpectrum Scaleは、データの削除も安全かつ完全に行うことが可能です。「デジタルシュレッダー」やマルチパスの上書きは必要ありません。安全な削除は暗号技術によって行われます。ファイルシステムの大規模なサブセットの安全な削除は、Spectrum Scaleが暗号アルゴリズムを使用して、速度、安全性、信頼性を確保して実行されます。

高度なストレージの効率性

Spectrum Scale RAIDには効率性において多くの利点があります。ストレージ処理を利用可能な資源全体に分散することで、ディスクの再構築時間を劇的に短縮し、性能を大幅に向上しています。すべてのストレージ資源を十分に活用し、ディスクがアイドル状態にならないようにしているのです。

Spectrum Scaleの「インテリジェンス」はストレージの効率性にも発揮されています。Spectrum Scaleは、データおよびファイルにおいて50もの異なる属性を収集し、データをより細かく効率的に管理することを可能にしています。データの圧縮が、その好例と言えるでしょう。Spectrum Scaleではポリシー機能と圧縮機能を活用して資源を効率的に使用することができます。Spectrum Scaleにより収集されたデータ属性を使用すれば、圧縮したいデータボリュームにだけポリシーを設定することができます。つまり圧縮で最も効果のあるデータだけを圧縮し、あまり効果のないデータは圧縮しないということが可能になります。

Spectrum Scaleにより収集されたすべてのデータ属性により、非常に効果的で効率的な情報ライフサイクル管理 (ILM) の性能が実現されます。これにより、データをストレージの適した階層 (テープや、さらにはクラウドも含む) に格納することで、コストを大幅に削減することができます。

推定では、すべてのファイルデータの最大90%が、最初に作成されて以降二度と必要とされないとされています。高級ワインと違い、多くのデータは時間と共にその価値が下がっていきます。長い期間に渡り蓄積したペタバイト級のデータと何十億というファイルがある状態で、各アプリケーショングループにそれぞれのデータを削除させるのは合理的ではありません。さらに、政府の規制により、特定のデータは長年に渡り保管する必要があります。この問題を解決するには自動化が必須ですが、Spectrum Scaleでは、ストレージ効率を劇的に向上するために設計されたILMツールを提供しています。またSpectrum Scaleは、Spectrum ProtectやSpectrum ArchiveといったIBMソフトウェア定義ポートフォリオの他の製品と密接に統合することが可能です。

Spectrum Scaleは、ポリシーベースの自動化フレームワークとILM機能セットにより、ストレージ運用管理をシンプルにします。これらのツールにより、Spectrum Scaleはユーザーデータを、論理的なディレクトリ構成における配置とは関係なく、物

理的にどの場所に格納するかを自動的に決定します。定義したストレージプール、ファイルセット、ポリシーにより、ストレージ資源のコストとデータの価値を整合させることが可能となり、それによりデータをどこに格納すべきかを決定します。

ストレージプールは、性能、場所、信頼性といった特性をグループ化することで、階層化することができます。例えば、あるプールはフラッシュアレイで、別のプールはより経済的なディスクストレージにするといったことが考えられます。

ファイルまたはオブジェクトが作成されると、Spectrum Scaleはそれをどこに格納するべきか知る必要がありますが、これは配置に関するポリシーを使って行われます。そのポリシーは、ファイルまたはオブジェクトが作成された際に、Spectrum Scaleが認識できるあらゆる情報（例：名前やファイルを作成したユーザー）に基づいて作成することが可能です。

一旦ファイルが作成されると、Spectrum Scaleはそのファイルについて多くのことを把握していることとなります。ファイル作成時に利用可能な属性に加え、ファイルのサイズ、最終アクセス時刻、変更の有無といった補足情報も抽出します。既存のファイルで施行されているポリシーはファイル管理ポリシーと呼ばれ、ファイルの移動、複製、削除を可能にします。ファイル管理ポリシーは、ディレクトリ構成におけるファイルロケーションを変更することなく、あるプールから別のプールへデータを移動するのに使用することができます。ただファイル管理ポリシーが最もよく活用されているのは、実はデータの移動ではなくレポーティングにおいてです。ポリシーのシンタックスは非常に強力なため、例えば、「スペースを最も消費しているファイルタイプに関するカスタムレポートを作成する」といったことが可能になります。



ペタバイト級の格納データを扱う場合、ILMツールが有用であるためには、高度な機能、自動化、非常に大規模なデータセットで運用できる性能が必要になります。Spectrum Scale ILMツールセットは、何ペタバイトものストレージ容量と何十億ものオブジェクトを管理する大規模な環境に対応できるよう設計されています。

限界のないスケーラビリティ

Spectrum Scaleは、小規模から開始し、組織の需要に応じて、ほぼ限界なくストレージ容量を拡張することが可能です。独自のデータオーシャンを構築できると言っても良いかも知れませんが、拡張は様々な面で可能です。ストレージ容量のシームレスな追加、ストレージ性能の向上、別のストレージ・プロトコル

への対応、これらがすべてアプリケーションの稼働を中断することなく実現できるのです。

こうした利点の大部分は、Spectrum Scaleのモジュラー型アーキテクチャにより可能になっています。Elastic Storage Server (ESS) はビルディングブロックとして機能します。CPUとストレージ資源はSpectrum Scaleの実装全体で共有されます。新しいノードが追加されると、ストレージ容量、性能、システム全体のレジリエンシーが直線的に増加します。OpenStack Swiftなどの他のソリューションコンポーネントを使ってファイルやオブジェクトを再配分する必要なしにストレージ容量を増やすことが可能なため、拡張はシームレスで透過的です。Spectrum Scaleを使えば、ストレージ容量、性能、オブジェクト数と、複数の次元において拡張することが可能です。拡張にはほとんど限界がなく、ファイルシステム当たりの最大ファイルシステム数は263 (約900京)、クラスタ当たりの最大ノード数は16,384です。

グローバルな共有とコラボレーション

Spectrum Scaleでは1つのクラスタで、同一ディレクトリ内の2つのファイルに対して、同一タイプのストレージあるいは異なるタイプのストレージを使用して、様々な処理に対応することが可能です。通常データセットは複数のサイトで生成されます。例えば、油層管理を最適化するための代替生産戦略を精査している場合を例にとってみましょう。この場合、詳細な地質学的記述を使用した多数のシミュレーションが要求されます。油層の特性をより正確に予測して新しいシミュレーションを実行するために、油層シミュレーションからの出力結果は通常、地震データセットと組み合わせられて使用されます。しかし多くの場合、地震データセットは別のロケーションにある別のシステムに格納されています。

また別の例として、自動車産業や製造業では、世界中の各地に散らばっている複数チームから新製品のデザインやイノベーションが生まれてくる場合が多々あります。地理的に異なる場所を拠点としている複数のチームのCAD (コンピューター支援設計) ダイアグラムや各種設計データをコラボレーションし、デザインの様々な面に取り入れる必要があるかも知れません。こうした状況では、データは多くの場合、小さなストレージの「島」に、異なる技術を使用して断片的に格納されていて、その拠点独自の使用ポリシーに従ってそれぞれのチームが管理しています。こうした一貫性のないアプローチにより、拠点をまたぐプロジェクトが妨

げられ、各チームは増大するコラボレーションのニーズに応えるためにIT環境を拡大するのに苦心することになります。

エンタープライズ全体で導入できるSpectrum Scale環境なら、データストレージを統一プラットフォームに一本化し、コンピューティング・プラットフォームをまたぐ巨大でグローバルなネームスペースや共通ファイルシステムといった機能により、コスト効率の良い、効率的なコラボレーションを実現することが可能です。データを他のロケーションから転送する必要がないため、ユーザーはどのノードからもシームレスにデータにアクセスすることが可能なので、コラボレーションプロセスが合理化されます。また、重複ファイルを格納する追加のディスクスペースを購入する必要がないため、コスト効率とエネルギー効率の向上も達成できます。さらに、データは高度に並列な状態で利用可能なため、大量のデータへのアクセスが大幅に高速化されます。

Hadoopのような新しい処理をシームレスに追加

ビックデータと呼ばれることもある大量のデータの分析に、MapReduceというアプローチを使用するのが業界のトレンドになっています。MapReduceを使用することで、非常に安価なハードウェアで高いI/Oスループットを達成できるというコスト面での効果が期待できます。これは、各サーバーの内蔵I/Oを活用して、サーバー群全体に大きな問題を解決させるというアプローチです。このタイプの処理を実行するには、それに適したハードウェア・アーキテクチャに対応し、MapReduceアプリケーションが正しいデータを見つけるための適切なインターフェースを提供できるストレージ・ソフトウェアが必要になります。Hadoop分散ファイルシステム（HDFS）およびIBM Spectrum Scaleはどちらも、何千と言うサーバーで構成された大規模な標準ハードウェア上でデータのMapReduceの処理に対応できるストレージ・プラットフォームを提供するために設計されています。



MapReduceは、大規模なデータセットをクラスタ上で分散コンピューティングにより並列処理するためのプログラミングおよびデータ編成モデルです。

HDFSとSpectrum Scaleはどちらも、MapReduceの処理を行うのに必要なストレージツールを提供していますが、両者には明確な違いがあります。HDFSはMapReduceのための基本的なストレージ・ソリューションに過ぎませんが、Spectrum

ScaleはMapReduceに対応した企業向けストレージ・ソリューションです。HDFSには以下のような制限があります。

- ✓ 一元化されたマスター/スレーブ型アーキテクチャ
- ✓ ファイルのロック機能なし
- ✓ ファイルデータは一定サイズのブロックに分割され、クラスタサーバーに分散される
- ✓ write-once-read-many (WORM：単一書き込み複数読み取り)によりデータの一貫性が単純化されており、ユーザーがデータを使ってできることが限られる

Spectrum Scaleには以下のような機能があります。

- ✓ *高速な共有ディスクのクラスタアーキテクチャによりデータアクセスを高速化
- ✓ メタデータの分散、スペースの割り当て、ロック管理によりデータ読み込みを向上
- ✓ 複数のサーバーおよびストレージ資源に分割されたファイルデータブロックにより最高の性能と信頼性を実現
- ✓ ファイルの任意のセクションを開き (open)、読み取り (read)、追加 (append) する機能

Spectrum Scaleには、MapReduce処理に対応した、FPO (File Placement Optimizer：ファイル配置最適化)と呼ばれる機能セットが用意されています。これは、各サーバーが独立して機能しローカルストレージを使用する分散コンピューティングのアーキテクチャです。コンピューティング・タスクは、これらの独立したシステムに分割され、あるサーバーが他のサーバーの処理を待たなければならないという状況は発生しません。Spectrum Scale FPOは、高度なクラスタ技術、動的なファイルシステム管理、高度なデータ・レプリケーション技術により高い可用性を提供しています。

さらに、Spectrum Scaleは、スナップショット、バックアップ、アーカイブ化、階層型ストレージ、データキャッシング、WANデータレプリケーション、管理ポリシーといったあらゆるエンタープライズストレージ機能に対応しています。Spectrum Scaleは、Hadoop MapReduce処理を実行して他の非構造ファイルデータにアクセスする広範なアプリケーションで使用することが可能です。



研究所でのベンチマーク・テストでは、Spectrum Scale FPOベースシステムが直線的に拡張することが示されており、40台のサーバーを有するファイルシステムでは12GB/秒のスループット、400台のサーバーを有するファイルシステムでは120GB/秒のスループットを見込まれます。



HDFSのような、よりシンプルなファイルシステムと比較した場合、Spectrum Scaleにはファイルとオブジェクトの両方の処理を扱えるという利点があります。生データをオブジェクトとして読み込むことは、一般的になりつつありますが、MapReduceベースの分析には、ファイルシステム・アプローチが使用されています。Spectrum Scaleを使用すれば、このギャップは簡単に解消することができます。オブジェクトデータを読み込み、Spectrum Scaleのファイル・インターフェースを使用して、MapReduce分析で利用できるようにすればよいのです。

この機能により、オブジェクトとしての読み込みと、分析アプリケーションへの対応に、別々のストレージ・システムを用意する必要がなくなります。さらにデータの重複と余計な維持費を削減しつつ、リアルタイムの分析およびデータ一貫性の維持が可能となります。つまり、システム間でデータを移動することなく、MapReduceを直接実行できるということです。

クラウドストレージの組み込み

クラウドストレージは、スケーラブルな仮想化基盤をサービスとして提供してくれるため、エンドユーザーは、複雑で細かい資源管理に煩わされる必要がありません。IDC (International Data Corporation) によると、世界の情報量は、今後10年間で44倍になると言われており、その大部分はクラウド・コンピューティングの台頭によるものだそうです。データの量が増え続けるにつれ、より多くの企業が、コストを最低限に抑えつつストレージ容量を拡張する方法として、クラウドストレージの活用を検討するようになっていきます。

IBM Spectrum Scaleは、その非常に高い拡張性、信頼性、コスト効率により、クラウドストレージへの対応に最適です。Spectrum Scaleベースのソリューションは、何百GB/秒というスループットを継続的に実現しつつ何千ものサーバーに拡張することが可能です。

Spectrum ScaleはOpenStackプロトコルのすべてに対応しているだけでなく、POSIXファイルAPIに依存する従来のアプリケーションにも対応しており、さらに豊富な管理ツールも用意しています。POSIXベースの分散ファイルシステムで構築されたクラウドストレージは、アプリケーションを調整することなく使用できるスケーラブルな基盤を提供することで、アプリケーションへの対応を容易にしています。

クラウドストレージは異なるクラスのアプリケーションで共有されるため、オブジェクトプロトコルに加え、標準的なファイルセマンティクスが重要となります。Spectrum ScaleならMapReduceを含む新しい処理に対応しているので、こうした処理を別のソリューションで行う必要がありません。クラウド環境にある大量のデータへのアクセスと、様々な処理が求められる環境において、Spectrum Scaleは賢い選択です。

ストレージの新しいパラダイム 「オブジェクトストレージ」

データ、トランザクション、デジタルデバイスの急激な増加により、今日のIT基盤と運用には多大な負担がかかっています。同時に、ストレージにかかる費用は増加の一途をたどり、ユーザーからの要求やコスト削減のプレッシャーも増していきばかりです。この急激なデータの増加、特に非構造化データの増加を背景に、ファイルやオブジェクトに対応した高性能なストレージソリューションへのニーズが高まってきました。『ファイル&オブジェクトストレージ For Dummies IBM 限定版』では、IBM Spectrum Scaleという強力なツールに基づいたファイル&オブジェクトストレージソリューションの数々をご紹介します。

- **スマートにスタート** — 容量、性能、俊敏性の強化
- **Spectrum Scaleの利点** — 全く新しい統合ストレージ
- **テクノロジーの考察** — OpenStack® SwiftとElastic Storage Server
- **情報から導き出される洞察と価値の促進** — 最も価値のある資産「情報」の活用



本書の内容：

- 最新のストレージの基礎知識
- IBM Elastic Storage Serverの詳細
- IBM Spectrum Scale オブジェクトストレージの10の利点

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley's ebook EULA.