背景およびビジネスの課題

■ ファイルサーバの重要性増加

近年、企業内で取り扱われるデータ量は爆発的に増えています。従来の基幹システム等で扱われる構造化データはもちろんの事、それ以外にも、メールやマルチメディアデータ、「ビッグデータ」というキーワードで扱われるログデータ、ソーシャルアプリケーションや、各種プローブ、センサーからの分析目的で収集される情報などの構造化されていないデータの扱いが劇的に増加しています。

またビジネスの規模、拠点の拡大や多様化に伴い、様々な利用者が、様々な場所からそれらの膨大なデータにアクセスするための需要が高まっています。

ビジネスに関わる全ての人が、あらゆる場所から、同一の情報にアクセスできる事が、ビジネスの効率性向上や戦略的な視点でも極めて重要になってきています。

そのような背景の中で、ファイルサーバは、企業活動を行う上で利用される基本的な情報共有基盤として、その利用シーン、重要性およびデータ量が増加の一途をたどっています。

このように、ファイルサーバの重要性が増すにつれて、以下のような課題が指摘されるようになってきています。

■ データ量の増加に伴う運用管理負荷の増大

データが劇的に増加する事で、システム管理者は定期的にストレージ容量の追加を行う業務に追われています。RAID 等、従来のストレージアーキテクチャは、容量の追加に伴い再構成を行う必要があり、その都度、データ全体のバックアップと復元という作業が必要になります。またその間のサービス停止が発生します。

データのバックアップ、復元も大きな課題です。大容量のバックアップおよび障害時の復元は処理時間が長時間化し、利用者から見たサービスレベルの低下やデータ復元にかかるリスクが増大します。

■ ネットワーク遅延による生産性の低下

データにアクセスするためのネットワークに十分な性能が無い事で、ビジネスの生産性を阻害する可能性があります。仮に単一のファイルアクセスに、数 10 秒の時間がかかるとすれば、知的ワークの思考は妨げられ、想像的なアイデアには余計な割り込みが入り、顧客は苛立ちを感じるでしょう。

これらの問題を避けるためには、非常に大きなコストをかけて、潤沢な帯域を持つネットワークインフラを構築するか、データそのものを利用者の近くに配置する工夫が必要となります。

■ 情報の分断による一元的整合性の欠如

地理的な制限、あるいはネットワーク遅延による影響を排除するために、データを利用者の近くに配置する戦略が多く採られています。

しかしながら、その分断されたデータの一貫性を保持することは難しく、ネットワーク速度を向上させるためのコストと情報の同期鮮度は常にトレードオフの関係にあります。

サーバの分断は、不必要なデータの複製を生み、一貫性の無いデータが企業内に散在する状況をもたらします。

■ 情報の重要性、機密性に見合うガバナンスの欠如

情報共有基盤としてのファイルサーバが有効に活用されるにつれ、取り扱われる情報の重要性が高まってきています。重要な情報を扱う基盤として、管理者による適切な運用が求められるのは当然の流れです。

IT 管理者は、情報の損失リスクに備えバックアップを取得し、不適切な利用を防ぐためにアクセス権を設定、メンテナンスし、利用履歴を監査する事が求められます。

これらの管理タスクは、特に情報が分断されて配置されている場合に、その作業負荷やコストを理由に、十分に行われていない現状があります。

■ オブジェクトストレージ技術によるソリューション

これらの課題の中で、データ容量の増大については、近年オブジェクトストレージと呼ばれる技術とそれを利用したクラウドサービスの登場により解決されようとしています。

オブジェクトストレージは、データをブロックイメージで保存するのではなく、オブジェクト（ファイル）単位で格納します。

オブジェクトストレージ実装の多くは、シンプルな x86 サーバで稼働し複数のサーバでクラスタを構成して稼働します。

格納されたデータは、複数のクラスタノードに複製されることで、安全性が担保されます。

データ容量の追加は、クラスタを構成するサーバノードを追加する事で行います。自律的に格納領域の拡張と、格納データの分散が行われます。

これらの機能によりオブジェクトストレージは、動的な容量拡張、単一障害点の排除、多重複製によりバックアップを不要にするなどの、従来型ストレージアーキテクチャである RAID では実現が困難であった数々の課題に対する解決策を提供しています。

そして、それらオブジェクトストレージの特徴を生かしながら、通常のファイルシステムとしての利用を可能にするストレージゲートウェイ実装と共に、ファイルサーバ領域の容量における課題は解決されつつあります。

従来手法によるアプローチ、 および残された課題

■ 大規模な集中管理方式

大規模な投資が可能な一部の企業は、巨大で高性能なストレージサーバを中央集中で配置し、高速なプライベートネットワークを事業拠点間に張り巡らせる方法を採用しています。

専任のシステム管理者が 24 時間体制で利用を監視し、ストレージサーバの高度な機能を用いて高速にバックアップを取得します。

遠隔地には、災害対策用のデータセンタを配備し、Gbps 規模の専用回線でデータの複製を行っています。

これは一見完璧なソリューションです。情報共有基盤に年間数億円の投資が可能な企業には最適なソリューションかもしれません。

しかしながら、多くの企業がそこまでのコストをかけて情報基盤を整備することは難しい現実があります。

多くの企業では、何かを妥協してもでも投資に見合う効果が得られるソリューションを探しています。

■ 小規模ファイルサーバによる分散配置

別の企業ではストレージサーバを分散配置するというアプローチを採用しています。低価格なファイルサーバやネットワークアクセスストレージを事業拠点毎に配置し利用する方法です。

この方法では、利用者の近くで高速な LAN で接続された環境にサーバが存在するため、快適な性能で、低コストでの利用が可能です。実際、ここ 10 年ほどの間に企業内で低価格 NAS は爆発的に普及し、企業内で扱うデータ量の多くを格納するまでに至っています。

しかしながら、分散配置された情報をガバナンスを効かせて管理することは大きな負担と困難を伴います。事実、事業拠点に配置された低価格 NAS の多くは、適切な運用管理者が配置されず、バックアップが取得されていない、アクセス管理も十分ではないという状況で利用されています。

これは、企業の戦略的な情報資産を扱う基盤としては、看過できない問題です。

■ WAN 高速化装置によるアクセス速度向上

ネットワーク通信の高速化と、エッジ（遠隔地）でのデータキャッシュを中心としたアプローチです。インターネットでのコンテンツ配信に用いられる CMS のアプローチに似ています。

基本的には、一定のライフサイクルでキャッシュを管理し、期限が切れた場合に中央集中のマスターサーバからデータをキャッシュし直すという方式です。

キャッシュのリフレッシュに、マスタからの PUSH や、クライアントからのイベントが利用できるものもあります。

一般的に、これらの企業内導入ソリューションは、ハードウェアアプライアンスの形態をとる事が多く、その多くは非常にコストが高いソリューションです。

■ DFS による分散ファイルサーバ

ネームサーバによる分散処理であり、データの一貫性管理と遠隔拠点でのアクセス性能の向上を実現できるものではありません。

ただし、大規模なファイルサーバを小規模なサーバリソースを組み合わせることで利用できる考え方は、中央集中型のトポロジを採用する中で、コストというデメリットの一部を解決できる可能性があります。

■ クラスタファイルシステム

複数のサーバのデータを同期することで、同じデータを複数の拠点で利用できるようにするアプローチです。

拠点間のロックの管理やデータの即時レプリケーションにかかる負荷、性能低下などの理由により、ファイルサーバの用途としてはほとんど採用されていません。

■ 残された課題

いくつかのアプローチを見るにつけ、情報基盤のガバナンスとパフォーマンス、そしてコストは、容易には全てを満たす事ができない課題、ジレンマとして、今でもシステム管理者を悩ませています。

低コストで情報基盤のガバナンスを維持するためには、一貫性を持つ全てのデータに管理者が容易にアクセスできる環境が必要です。

また、遠隔地からでも良いデータアクセスパフォーマンスを得るためには、ネットワーク帯域を増強するか、利用者の近くにデータを配置する必要があります。低コストでという前提が付けば、現在の高速なネットワークのコストを考えた時、利用者の近くにデータを配置するというアプローチがリーズナブルであるという判断になります。

クラウドのアーキテクチャにヒントあり

世の中はクラウドブームです。一部にはパスワードと化していると思われる向きもありますが、クラウドの根底にある考え方は、従来のプログラミングモデル、情報処理モデルを大きく変革する可能性を秘めています。

先述のファイルサーバの課題も、クラウドの考え方のなかに解決のヒントがあるかもしれません。

■ CAP 定理による既存アプローチの分析

クラウドシステムのアーキテクチャを語る上で、整合性 (Consistency)、可用性 (Availability)、分断耐性 (Partition) は、同時に2つまでしか満たすことが出来ないという CAP 定理という主張があります。

CAP 定理は証明された定理ではありませんが、先述のファイルサーバのジレンマに実に良く当てはまります。

これらに照らし合わせて、従来の課題解決アプローチを見てみましょう。

従来のファイルサーバ等で用いられるファイルシステムは、分断耐性を犠牲にして整合性と可用性を満たす事を目指して構築されていると考えられます。というよりはそもそも分断して利用する事を想定していません。

中央集中で巨大なファイルサーバを構築する手法はもちろんの事、低価格 NAS を拠点に分散配置する方法も、データの連携が無いのであれば、ここで述べる分断耐性を持っている事にはあたらないでしょう。

DFS も単にネームサーバがデータが存在する個別のサーバに処理をリダイレクトしているのみであり、個々のサーバで見た場合には分断耐性が犠牲にされていると見ることができます。

従来のクラスタファイルシステムは、ノード間のロックとデータレプリケーションによって分断耐性が実現されています。これは一見可用性がある仕組みであると思われがちですが、問題があった場合にはノードを切り離して動作する。その意味では特定ノードの可用性を犠牲にして一貫性を保つ仕組みという事ができます。

■ BASE トランザクションの適用

クラウドシステムのアーキテクチャやプログラミングモデルに、従来の ACID と呼ばれるトランザクションの考え方に変わり BASE トランザクションという考え方が重要視されてきています。

この考え方は、基本的には、一時的な整合性 (Eventually Consistent という) に妥協点を見出して、分断耐性と可用性を担保するというアプローチに基づいています。

ファイルサーバにもこのアプローチが適用できないでしょうか。一時的な整合性に妥協する事で、分断耐性と可用性を持ったファイルサーバを構築する。

先に述べた通り、低コストでアクセス速度を得るためには、ファイルサーバを分断して利用者の近くに配置する必要があります。そのためには分断耐性は犠牲にできません。

拠点配置されたファイルサーバは、分断されたひとつのノードですが、拠点で見た場合は単一のサービスとなりますので、可用性を犠牲にする事も出来ません。

分断耐性と可用性を犠牲にしない事を前提とした場合、おのずと整合性に妥協点を見出すしかありません。

ルーズリー (緩やかな) クラスタ ファイルシステムというコンセプト

ファイルサーバという情報基盤において、分断耐性と可用性を犠牲に出来ないとしたら、妥協できる一時的な整合性の欠如とはどのようなものでしょうか。

東京と大阪でそれぞれファイルサーバを配置し、東京でのデータの更新を即時に大阪で参照しなければならぬケースはどの程度あるのでしょうか。

そもそものファイルサーバの利用者は人間であり、主に非同期での情報伝達手段として用いるものと考えられます。視点を変えれば、人間が同時に情報を共有する事を目的とするのであれば、リモート会議やシステムの同期通信によって行う方が適しています。

であるとすれば、ファイルの遠隔地への反映は一定の遅延があっても許容できるという見方も可能です。WAN 高速化装置に代表されるキャッシュを用いたシステムや、ファイルシステムの非同期レプリケーション機能は基本的にはこの考え方に基づいています。

しかしながら、情報同期の遅延はもうひとつの厄介な問題を引き起こします。更新の競合です。

先の例で言えば、東京でのあるファイルに対するデータ更新が大阪に反映する前に、大阪で同じファイルに別の更新を行った場合どうなるのでしょうか。

多くのケースでは、先に行った更新が、後に行った更新によって上書きされ失われるという問題を引き起こすでしょう。これはもはや一時的な整合性の欠如ではなく、情報の喪失にあたります。

従来の仕組みではこれを防ぐ目的で、遠隔地でファイルの排他制御を行う、あるいはデータの更新差分を抽出して反映するなど、極めて複雑な仕組みを導入しているケースが多く、結果的にソリューションが高コストなものになっているのです。

もうひとつの発想の転換

(データの上書きをしない)

折角「クラウド」という新しい要素技術を取り入れる発想の転換を行うのであれば、もうひとつ斬新な発想の転換を試みるのも面白いかもしれません。

更新の競合による情報の喪失は同じファイルを複数のユーザが同時に上書きする事で発生します。逆の視点で考えれば、データを上書きしなければ更新の競合による情報の喪失は発生しないという事になります。

これが、もう一つの発想の転換「PUT ONLY」です。

分散配置されたファイルサーバにおいて、複数のノードからほぼ同時に発生した特定ファイルに対する更新が、全て同一ファイルの別の版として保存されるとすればどうでしょうか。

先に述べた更新の競合による情報の喪失という課題は防げる事になります。

保存された複数の版から、どの版が利用者にとって必要なものか、最終的な利用者である人間が判断できる事になります。

もちろん、何のガイドもなしに多くの更新版が存在する事は、不要な複製を多く管理する事に繋がります。複数の版から必要なものを選択する手間もばかになりません。

ここでファイルサーバに求められる事は、シンプルなアルゴリズムで複数の版の中から最も有用であろう候補を提案する事です。それは完全である事は重要ではありません。候補が利用者にとって求める版ではないと認識した場合に、複数の版から求めるものが得られる事が重要です。

これまでに述べた、分断耐性と可用性を備え、ファイルサーバとして新しく適用すべき2つの特徴、「BASE トランザクション」と「PUT ONLY アプローチ」を備えたファイルシステムを、「ルーズリー (緩やかな) クラスタファイルシステム」というコンセプト名で呼ぶ事にします。

■ ルーズリークラスタファイルシステム実装としての CSCFS3

FOBAS CSC Ver.3.0 の中核にある、CSCFS3 は、これまでに述べたルーズリークラスタファイルシステムというコンセプトを具体的に実装したものです。

従来の FOBAS CSC は、オブジェクトストレージ、あるいはそれらの技術を応用したクラウドストレージに、ファイルシステムインタフェースを提供するストレージゲートウェイ製品として利用されてきました。

ユーザが格納したデータは、特定のブロックサイズに分割され、暗号化され、圧縮され、重複排除の後に、バックグラウンドのクラウドストレージあるいはオブジェクトストレージに格納されます。

CSCFS3 でも、これらの基本機能に変わりはありません。

バックグラウンドに格納されたデータとユーザから見たファイルシステムを紐付けるメタデータは、基本的にはサーバローカルで管理されますが、CSCFS3 では、バックグラウンドストレージを通して、複数のクラスタノードに非同期でレプリケーションされます。

このメタデータのレプリケーションにより、複数のノードで同じファイルを扱う事ができるようになります。

データの利用者は、キャッシュにデータの実体がある場合はそのまま利用しますが、キャッシュにあるデータの版と、レプリケーションされたメタデータの版が異なる場合はバックグラウンドストレージからデータ実体を取り出して、ファイルの実体を再現します。

ここまでの機能で、一時的な整合性の欠如 (メタデータレプリケーションの遅延) はあれども、分断耐性と可用性という、低コストでアクセス性能が良い分散配置型ファイルサーバとしての必要条件は満たせる事になります。

あとは、更新競合による情報の喪失をどのように防ぐかです。

CSCFS3 では、クラスタファイルシステムのコンセプトでも述べた通り、PUT ONLY 戦略を採用しています。つまり、同一ファイルに対する別ノードからの更新は、上書き更新をせず、全て別の版として独立して保存します。

ただ、ファイルシステムインタフェースとしての特性上、単一のファイルは単一のものとしてしか扱えません。CSCFS3 では、保存された複数の版から、ファイルシステムインタフェースから優先的に取り出す版を、以下の2つのシンプルなアルゴリズムで決定する事を選択できます。

1) タイムスタンプ

更新日時が最も新しいものを優先的に扱うアルゴリズムです。

2) サイトプライオリティ

クラスタを構成するサーバノードはそれぞれ固有の ID を持つ事ができます。その ID が小さいものを優先するというアルゴリズムです。もちろん、全ての版にこのアルゴリズムを採用してしまうと、サイトプライオリティが高いサイトの更新がいつまでも優先版として扱われてしまう事になります。このアルゴリズムが利用されるのは、レプリケーションウィンドと呼ばれるメタデータ同期処理の遅延時間 (基本的には 10 分間) に、複数のノードからの同時更新が発生した場合のみです。特定ノードからの更新をさせたくないのであれば、それはむしろアクセス権の設定等で解決すべき問題です。

優先的に扱われなかった版は、FOBAS CSC Web コントロールパネルから、保存された全ての版を参照し、更新ノード、更新日時、更新ユーザなどのメタ情報を参考に、いつでも必要な版を取り出す事ができます。

ソフトウェアアプライアンスとしての FOBAS CSC Ver.3

FOBAS CSC Ver.3 は、CSCFS3 に利用インタフェースとしての、CIFS(Samba)、NFS、FTP、WebDAV を構成して、使いやすい形でパッケージした、仮想アプライアンス製品です。

ユーザは、プログラムが利用するライブラリの依存性や、前提となる OS、ミドルウェアバージョンなどに煩わされる事なく、FOBAS によって事前検証された組合せを仮想マシンとして即座に利用する事ができます。

■ 高いスケーラビリティ

FOBAS CSC Ver.3 のスケーラビリティの考え方は非常にシンプルです。基本要件を満たす仮想マシンに、テンプレートから展開してサーバの利用を開始します。性能の不足を感じた場合は、同様のスペックを持つ仮想マシンで稼働する FOBAS CSC Ver.3 サーバをもうひとつ起動し、先のサーバのクラスタメンバとして参加をさせるだけです。

自律的にメタデータの同期を行い、同じ論理ファイルシステムを構成する別サーバとして稼働を始めます。

FOBAS CSC Ver.3 では、最大 128 ノードまでのクラスタノードをサポートします。

■ 高い可用性

FOBAS CSC Ver.3 のルーズリークスタファイルシステムは、可用性の向上目的でもソリューションとなります。クラスタを構成する各ノードは、最大でも 10 分の遅延で情報が同期される論理的に同一のファイルシステムです。

2 台を並列で配置し、IP ベースで利用ノードを Stick したロードバランシングを行う事で、単一ノードの障害の影響を受けない高い可用性を持つファイルサーバシステムを構築可能です。

それとは別に、データの欠損が許されない場合には、従来の FOBAS CSC で利用されてきた Shared Nothing Cluster の構成を取る事も可能です。

この場合、構成された Shared Nothing Cluster は、ルーズリークスタファイルシステムの論理的な 1 ノードとして扱われます。

■ バックアップ

FOBAS CSC Ver.3 では、クラスタノードへのレプリケーションの前提として、定期的なマスターノードのメタデータバックアップ（スナップショット）と、各クラスタノードのメタデータ更新差分のバックアップを行っています。

運用者は、明示的なバックアップの必要が無く、いつでも取得されたスナップショットとメタデータ更新差分情報により、障害発生時の最大 10 分前の状態に戻す事が可能です。

■ 災害対策

先述のバックアップからのリストアは、拠点に依存しません。データの格納先であるバックグラウンドストレージにアクセスが可能な場所であればどこでもサーバの復元が可能です。

この事は、災害対策等を意識した大がかりな設備投資をせずとも、災害時にはどこでもデータが復元出来る事を意味します。

■ アクセス権、属性の設定

分散配置されたファイルサーバでは、その管理タスクの負荷が問題でした。FOBAS CSC Ver.3 では、クラスタメンバの 1 ノードに対して、アクセス権の設定や、ファイルサーバ属性の設定を行う事で、全てのノードに設定を反映する事が可能です。

世界各地に分散した、最大 128 台のサーバを、管理拠点の 1 台のサーバから設定、管理を行う事が可能です。

■ アクセスログ

FOBAS CSC Ver.3 では、クラスタを構成する全てのノードで行われた操作がメタデータの更新差分としてバックグラウンドストレージに集約管理されます。

この事は、クラスタを構成する全てのノードで行われた、ファイルサーバに対する操作履歴が管理されている事と相違ありません。システム管理者は、これらの更新差分の保存期間の範囲で、任意の操作履歴を抽出し管理に役立てる事が可能です。

想定されるユースケース

■ 拠点配置が可能な分散ファイルシステム

ルーズリークスタファイルシステムの特徴を生かし、中央で管理するファイルサーバのレプリカを、分散拠点に配置する事が可能です。

わずか 10 分程度の同期遅延はありますが、中央で管理するファイルサーバと同じ情報を各拠点ローカルで利用する事が可能です。

ローカル拠点内では、通常のファイルシステムとして、排他制御を含め高速なアクセス速度で利用可能です。

■ 大規模ファイルサーバの負荷分散ソリューション

大企業におけるファイルサーバは、時として数万人のユーザが利用するケースがあり、従来は専用の特種な大規模サーバでしか構築できないものでした。

ルーズリークスタファイルシステムでは、クラスタメンバを構成する全てのサーバで同一のデータを利用できますので、例えば利用者 1,000 人毎にサーバを分割して 30 台のクラスタを構成する事で、3 万ユーザをサポートするシステムを構築する事ができます。

管理者はクラスタを構成するサーバの 1 台に設定を行う事で、全てのクラスタメンバに設定が反映し、全てのサーバに対するアクティビティを 1 台のサーバから取得する事ができます。

特定のサーバの障害時には、一時的に他のサーバを利用させながら、新しいサーバノードの追加を行う事で、自律的にデータの復元を行います。

■ 大規模オブジェクトストレージ利用のゲートウェイ

ビッグデータというキーワードで語られる、分析目的の大規模データは、時としてペタバイトオーダーに達する事もあります。

それらのデータの格納先として、オブジェクトストレージは大変有力なソリューションとして注目されています。

オブジェクトストレージをマウント可能なファイルシステムとして変換するストレージゲートウェイ製品は存在しましたが、オブジェクトストレージの真価であるペタバイト級のデータを扱う事ができる真のスケラビリティを持ったストレージゲートウェイ製品は有りませんでした。

FOBAS CSC Ver.3 は、ルーズリーククラスタファイルシステムの採用により、単一のファイルシステムに対して、最大 128 ノードでの並列書き込みが可能になります。

一例として、1Gbps のネットワークインタフェースを持つストレージゲートウェイがオブジェクトストレージに書き込みができるスループットを 20MB/sec と仮定した場合、1PB のデータを単一のファイルシステムに格納するには、従来は、1 年半もの時間がかかる計算になります。

128 ノードのルーズリーククラスタファイルシステムでは、この時間を理論上 5 日間に短縮する事が可能です。

注意) このホワイトペーパーは、2014 年 9 月 1 日時点の製品計画をベースに記述されています。記載された全ての機能が、製品リリース時点で提供される事を保証するものではありません。

FOBAS®

FOBAS コンサルティング株式会社
<http://www.fobas.jp/>