



AWS クラウドのストレージオプション

Joseph Baron, Amazon Web Services
Robert Schneider, Think88

2010 年 12 月

序論

Amazon Web Services (AWS) は、柔軟で使いやすく費用効果の高いクラウドコンピューティングプラットフォームで、様々なクラウドベースのデータストレージオプションをサポートします。また、アーキテクトや開発者を対象とした広範なオプションも用意されています。

このホワイトペーパーは、AWS クラウドコンピューティングプラットフォームで利用できる主なデータストレージオプションを理解する際に役立ちます。例えば、各ストレージオプションの概要を示し、理想的な使用シナリオについて説明するほか、弾力性や拡張性など、他の重要なクラウド固有の特徴についても確認します。また、各ストレージオプションの耐久性の特定については、特に注目しています。このホワイトペーパーの最後にはクイックリファレンステーブルがあります。このテーブルでは、ここで紹介したストレージオプションを比較します。「参考文献」セクションで、その他のリソースについても必ずご確認ください。

最後に紹介するのは、[ストレージの使用事例がいくつか示されている別のドキュメント](#)です。このドキュメントでは、複数の AWS クラウドストレージオプションを一緒に使用方法について説明しています。この使用事例を、独自のストレージアーキテクチャを設計するときのガイドとして使用できます。

従来のストレージとクラウドベースのストレージ

従来の構内 IT インフラストラクチャとアプリケーションのアーキテクトには、次に示すように、膨大なデータストレージオプションがあります。

- **メモリー**：ファイルキャッシュ、オブジェクトキャッシュ、インメモリーデータベース、RAM ディスクなど、インメモリーストレージでは非常に迅速にデータにアクセスできます。
- **メッセージキュー**：コンピュータシステム間またはアプリケーションコンポーネント間で非同期的に送信されるデータを対象とした耐久性に優れた一時ストレージ。
- **ストレージエリアネットワーク (SAN)**：専用 SAN 上のブロックデバイス（仮想ディスク LUN）は、通常、ビジネスクリティカルなファイルのデータとデータベースストレージの両方に対して最高レベルのディスクパフォーマンスと耐久性を提供しますが、最もコストのかかるシステムの 1 つです。
- **DAS (Direct Attached Storage)**：各サーバーに存在するローカルハードディスクまたはアレイは、SAN よりも高いパフォーマンスを提供しますが、一時ファイルと永続ファイル、データベースストレージ、およびオペレーティングシステム (OS) ブートストレージの耐久性は SAN よりも低くなっています。
- **NAS (Network Attached Storage)**：NAS ストレージは、ファイルレベルのインターフェイスを、複数のシステムで共有できるストレージに提供します。NAS の速度は、SAN または DAS よりも遅くなる傾向にあります。
- **データベース**：構造化されたデータは、通常、MySQL、PostgreSQL、Oracle、Microsoft SQL Server、DB2 などのリレーショナルデータベースまたは非リレーショナルデータベースリポジトリを使用して維持およびアクセスされます。一般的に、データベースストレージボリュームは SAN または DAS デバイスにあります。

- **オフライン**：バックアップおよびアーカイブの目的で保存されたデータは、通常、テープ、CD、DVD などの非ディスクメディアに配置されます。このメディアは、多くの場合、障害復旧のために安全なリモートの場所に保存されます。

この従来のストレージオプションはそれぞれ、パフォーマンス、耐久性、コストだけでなく、インターフェイスも異なります。アーキテクトはすべての要素を考慮したうえで、対象のタスクに適したストレージソリューションを特定します。特に、IT インフラストラクチャとアプリケーションアーキテクチャのほとんどが、複数のストレージテクノロジーを採用しています。この各テクノロジーが、特定のデータストレージのサブクラスのニーズを満たすために選択され、その組み合わせがデータストレージ層の階層を形成するのです。

このホワイトペーパーで説明するように、AWS には、複数のクラウドベースのストレージオプションが用意され、各オプションで、パフォーマンス、耐久性、コスト、およびインターフェイスが一意に組み合わせられています。また、このオプションは、弾力性、可用性、拡張性などの追加要素によってさらに強化されます。こうした追加要素は、ウェブスケールのクラウドベースのソリューションにとって重要です。従来の構内アプリケーションと同様、複数のクラウドストレージオプションを一緒に使用して、包括的なデータストレージ階層を形成できます。

このホワイトペーパーでは、次の AWS クラウドストレージオプションについて説明します。

- **Amazon EC2 Elastic Block Storage (EBS) ボリューム**
- **Amazon EC2 ローカル インスタンス ストア (一時) ボリューム**
- **Amazon Simple Storage Service (Amazon S3)**
- **Amazon Simple Queue Service (SQS)**
- **Amazon SimpleDB**
- **Amazon EC2 Relational Databases**
- **Amazon Relational Database Service (RDS)**

各 AWS ストレージオプションについて詳しく見ていきましょう。オプションごとに次の情報を示します。

- 名前と説明
- 理想的な使用シナリオ
- パフォーマンス
- 耐久性と可用性
- コスト
- 弾力性と拡張性

- インターフェイス
- アンチパターン。他のストレージオプションの方が適していると思われる状況

AWS ストレージコレクション間の追加の比較分類については、36 ページの「AWS ストレージ クイックリファレンス」をご覧ください。

Amazon Elastic Block Store (EBS) ボリューム

Amazon Elastic Block Store (EBS) ボリュームは、Amazon EC2 インスタンス（仮想マシン）で使用するための耐久性に優れたブロックレベルのストレージを提供します。Amazon EBS ボリュームは永続性のあるネットワーク接続ストレージで、1 つの Amazon EC2 インスタンスの運用状況に左右されません。Amazon EC2 インスタンスに接続された EBS ボリュームは、通常、選択したファイルシステムでフォーマットすることで、物理ハードディスクドライブのように自由に操作できます。EBS ボリュームを使用すると、Amazon EC2 インスタンス（EBS AMI のみ）を起動できます。また、複数の EBS ボリュームを 1 つの Amazon EC2 インスタンスに接続できます。ただし、1 つの EBS ボリュームが接続できるのは、任意の時点における 1 つの Amazon EC2 インスタンスだけです。1 つの EBS ボリュームを、他のユーザーと共有することはできません。ただし、EBS スナップショットを作成している場合は除きます（次の「耐久性と可用性」をご覧ください）。EBS ボリュームのサイズは 1 GB ~1 TB の範囲で、1 GB 単位で割り当てられます。

理想的な使用シナリオ

Amazon EBS は、比較的頻繁に変更され長期永続性が必要なデータを対象としています。この EBS は、Amazon EC2 仮想サーバーに対して持続的な仮想ブロックモードのストレージを提供するので、物理サーバー上のハードドライブのように使用することができます。Amazon EBS は、特に、ファイルシステムの主要ストレージ、データベースとして使用するのに適しています。また、詳細な更新や、フォーマットされていないブロックレベルの未加工ストレージへのアクセスを必要とするアプリケーションにも合っています。

パフォーマンス

一般的に、個別の EBS ボリュームには、外部電源の USB ドライブと同等のパフォーマンス、故障までの平均時間 (MTTF)、および信頼性を期待できます。EBS ボリュームは、ローカル ディスク ドライブのように見えますが、実際は Amazon EC2 インスタンスにネットワーク接続されていることに注意してください。したがって、そのインスタンスが実行する他のネットワーク I/O や共有ネットワークの全負荷が、個別の EBS ボリュームのパフォーマンスに影響を及ぼす可能性があります。

各アプリケーション（および関連するパフォーマンス）は一意ですが、EBS ボリュームでは、従来からある数多くのディスクスループット最適化手法を自由に設計および採用できます。Amazon EC2 と EBS を組み合わせて使用すると、構内サーバーおよびストレージで使用しているものと同じパフォーマンス最適化手法の多くを使用できます。例えば、複数のボリュームを作成し、そのボリュームすべてを 1 つの Amazon EC2 インスタンスに接続できます。複数の EBS ボリュームが接続されている場合は、全アプリケーション I/O 負荷をパーティションで区切ることができます。これを行うには、ログデータとデータベースにボリュームを 1 つずつ、そして、ファイルデータに別のボリュームを 1 つ割り当てます。また、ソフトウェア RAID 0 デバイスドライバを使用して複数の EBS ボリュームにわたるデータをストライプ化し、使用できる IOP、総ボリュームスループット、および総ボリュームサイズを集計することもできます。

耐久性と可用性

1つのハードウェアコンポーネントで障害が発生したときにデータが紛失しないように、各 Amazon EBS ボリュームが同じ利用可能ゾーン内で自動的にレプリケートされます。Amazon EBS は、ボリュームのポイントインタイムスナップショットを作成して、Amazon S3 に保管することができます（以下を参照）。このスナップショットは、新しい Amazon EBS ボリュームを立ち上げるときにも、データを長期間格納するときにも役立ちます。

EBS ボリュームの耐久性は、ボリュームのサイズと最後にスナップショットを作成してから変更されたデータ量の両方に依存します。EBS スナップショットはインクリメンタルなポイントインタイムバックアップで、最後にスナップショットを作成してから変更されたデータブロックのみが含まれます。最新のスナップショットの作成後に変更されたデータが 20 GB 以下の EBS ボリュームについては、年間の障害発生率（AFR）が 0.1 ~ 0.5% です。EBS ボリュームに保存されているデータの耐久性と可用性を最大限に高めるには、EBS ボリュームのスナップショットを頻繁に作成します。たとえ Amazon EBS ボリュームで障害が発生しても、ボリュームのスナップショットはすべて元の状態を保っているため、最新のスナップショットからボリュームを再作成できます。

Amazon EBS ボリュームは、高い可用性を実現できるよう設計されています。ただし、EBS ボリュームは、特定の利用可能ゾーンに作成されるので、利用可能ゾーン自体が使用できない場合は、ボリュームも使用できなくなります。1つの EBS ボリュームは1つの利用可能ゾーンに制限されますが、ボリュームの EBS スナップショットは、リージョン内のすべての利用可能ゾーンにわたって使用できることに注意してください。したがって、EBS スナップショットを使って、1つ以上の EBS ボリュームを任意の利用可能ゾーンに新しく作成できます。EBS スナップショットを他のユーザーアカウントを共有することもできます。これにより、使いやすい「ディスククローン」および「ディスクイメージ」バックアップと共有メカニズムが実現します。

EBS データの耐久性と可用性を最大限に高めるには、EBS ボリュームのスナップショットを頻繁に作成します。

コスト

すべての Amazon Web Services と同様、Amazon Elastic Block Store は従量課金制なので、最低料金も長期契約もありません。Amazon EBS は、プロビジョニングされたストレージについては 1 GB あたりの料金が設定され、1 か月ごとに課金されます。また、リクエストについては 100 万回の I/O リクエストあたりの料金が設定されています。ボリュームストレージの料金は、割り当てた量に対して、解放するまでの間発生します。Amazon EBS スナップショットは、保存されているデータについては、1 GB あたりの料金が設定され、1 か月ごとに課金されます。また、スナップショットの保存および読み込みについては、1,000 PUT リクエストおよび 10,000 GET リクエストあたりの料金が設定されています。EBS スナップショットについては、実際に使用した（消費した）ストレージに対してのみ課金されます。EBS スナップショットはインクリメンタルで、圧縮されています。したがって、スナップショットで使用されるストレージは、通常、EBS ボリュームで使用されているストレージよりも格段に少なくなっています。料金設定の詳細については、<http://aws.amazon.com/ec2/pricing/> をご覧ください。

様々な AWS ストレージ（Amazon EC2 インスタンスと EBS、Amazon S3、Amazon RDS など）の間で転送される情報は、同じ AWS リージョン内にある限りは課金されません。

弾力性と拡張性

AWS を使用すると、Amazon EC2 インスタンスが使用できるストレージ領域を簡単に拡張できます。新しい EBS ボリュームを作成してインスタンスに接続します。その新しく作成したボリュームは、既存のボリュームと一緒に使用できます。新しいボリュームを作成して管理したくない場合は、次の方法で 1 つのボリュームのサイズを拡張できます。

1. アプリケーションまたはファイルシステムを休止します。
2. EBS ボリュームのスナップショットを Amazon S3 に作成します（「Create Snapshot from Volume」を使用）。
3. スナップショットから新しい EBS ボリュームを作成します。ただし、元のボリュームよりも大きなサイズを指定します。
4. 新しく作成した大きなボリュームを Amazon EC2 インスタンスに接続します。
5. 元の EBS ボリュームの接続を解除して、削除します。

インターフェイス

Amazon EC2 インスタンスの EBS ボリュームを作成、削除、記述、接続、および接続解除するコントロールには、SOAP 形式と REST 形式の両方があります。また、EBS から Amazon S3 へのスナップショットと関連する属性の作成、削除、および記述には、API を使用できます。グラフィカルツールを使いたい場合は、AWS Management Console および ElasticFox Firefox 拡張機能により、API のすべての機能が使いやすいブラウザインターフェイスで利用できます。EBS ボリュームの作成方法に関係なく、すべてのストレージがボリュームの作成時に割り当てられます。そのストレージについては、使用していなくても課金されます。

EBS では、ブロックデバイスのインターフェイスが Amazon EC2 インスタンスに提供されます。つまり、Amazon EC2 インスタンスでは、ローカル ディスク ドライブのように EBS ボリュームが示されます。したがって、選択されたオペレーティングシステムのネイティブのファイル システム インターフェイスを使用して、EBS ボリュームでデータの書き込みと読み取りを行うことができます。

Amazon EBS アンチパターン

前に説明したように、EBS は、1 つの Amazon EC2 インスタンスの運用状況とは関係なく維持する必要がある情報に適しています。ただし、ある状況では、他の AWS ストレージオプションの方が適していることがあります。

- **一時ストレージ**：Amazon EC2 インスタンスの終了後、そのデータに今後アクセスできるようにするかどうかを気にする必要がない場合は（一時ファイル、クラッチディスク、バッファなど）、ほとんどの Amazon EC2 インスタンスで自動的に提供されるストレージボリュームを利用することを検討します。この一時ボリュームは、Amazon EC2 の標準コスト以外の追加コストなしで提供されます。詳細については、次のセクションをご覧ください。

- **耐久性に優れたストレージ**：耐久性に優れたストレージが必要な場合は、Amazon S3 を使用します。Amazon S3 標準ストレージは、オブジェクトあたり 99.999999999% の耐久性を実現するために設計されています。一方、最新のスナップショットの作成後に変更されたデータが 20 GB 未満の EBS ボリュームについては、年間 99.5 ~ 99.9% の耐久性を実現できるように設計されています。変更データがさらに多い場合は、そのデータ量の増加に応じて耐久性が低下することが予想されます。
- **静的データまたはウェブコンテンツ**：データの変更がそれほど頻繁に行われえない場合、Amazon S3 では、より費用対効果と拡張性に優れたソリューションを使用してこの固定情報を保存できます。また、EBS から提供されるウェブコンテンツについては、Amazon EC2 でウェブサーバーが実行されている必要があります。一方、Amazon S3 から直接ウェブコンテンツを提供することはできます。
- **キーと値のペアに関する情報**：14 ページの「Amazon SimpleDB」で説明するように、Amazon SimpleDB を使用すると、弾力性および拡張性の高い方法で「スキーマレス」データを保存できます。リレーショナルテクノロジー（関連する EBS ボリュームを含む）を使用して、キーと値の情報を管理しようとしている場合は、代わりに SimpleDB を試してみる価値があります。

Amazon Elastic Compute Cloud (EC2) ローカルインスタンスストア ボリューム

Amazon EC2 のローカルインスタンスストア ボリューム（一時ドライブとも呼ばれます）は、Amazon EC2 インスタンスのブロックレベルの一時ストレージインスタンスを提供します。Amazon マシンイメージ（AMI）から Amazon EC2 インスタンスを作成するとき、ほとんどの場合、そのインスタンスには、事前に接続されたディスクストレージの事前設定されたブロックが付属しています。¹EBS ボリュームとは異なり、インスタンスストア ボリュームのデータは、関連する Amazon EC2 インスタンスの運用中は維持されます。ディスクストレージ容量は、160 GB ~ 1.7 TB の範囲で、これは Amazon EC2 インスタンスタイプに応じて異なります。インスタンスストアのボリュームの数とサイズは、Amazon EC2 インスタンスが大きくなるほど増大します。ただし、この量は十分のように見え、役に立つことが多いものの、このストレージは一時的なもので、スクラッチボリューム、RAM ディスクのように使用するのが最適です。

理想的な使用シナリオ

ローカルインスタンスストア ボリュームは、継続的に変更される情報（バッファ、キャッシュ、スクラッチデータ、その他の一時コンテンツなど）、およびインスタンスのフリートにわたってレプリケートされたデータ（負荷分散されたウェブサーバー プールなど）に適しています。Amazon EC2 インスタンスストレージは、この目的で設計され、仮想マシンブート デバイス（インスタンスストア AMI のみ）と、Amazon EC2 インスタンス専用の 1 つ以上の追加ボリューム（EBS AMI とインスタンスストア AMI の両方）で構成されています。また、ストレージの運用期間中に、1 つの Amazon EC2 インスタンスからのみ使用できます。EBS ボリュームとは異なり、インスタンスストア ボリュームは接続を解除したり、他のインスタンスに接続したりすることはできません。

¹ほとんどの Amazon EC2 インスタンスタイプに、ローカルインスタンスストレージ ボリュームがあります。ただし、type t1.micro などのマイクロインスタンスが提供するものは EBS ストレージだけです。また、一時インスタンスストア ボリュームは、デフォルトでは、ルートデバイス（「EBS からブート」の Amazon EBS を使用するインスタンスでは公開されません。インスタンスストア ボリュームは、ブロック デバイス マッピングを指定することで、インスタンスの起動時に必要に応じて公開できます。詳細については、「Amazon EC2 User Guide」をご覧ください。

パフォーマンス

Amazon EC2 インスタンス仮想マシンとローカルインスタンスストア ボリュームが同じ物理サーバーにあるため、特に順次アクセスの場合は、非常に高速にこのストレージとやり取りできます。1 秒あたりの I/O 操作 (IOPS) を増やすか、ディスクスループットを向上させるために、RAID 0 (デスクストライピング) ソフトウェアを使用して、複数のインスタンスストア ボリュームをまとめることができます。

耐久性と可用性

Amazon EC2 ローカルインスタンスストア ボリュームは、耐久性に優れたディスクストレージとして使用するためのものではありません。ローカルインスタンスストア ボリュームに保存されているデータは、Amazon EC2 インスタンスの運用期間中のみ維持されます。Amazon EC2 ローカルインスタンスストア ボリュームのデータは、インスタンスが規則正しくリブートされる限り維持されますが、Amazon EC2 インスタンスが終了した場合、または障害/再起動が繰り返されている状況では維持されません。

永続ファイル、データベースストレージなど、長期間維持する必要があるデータにはローカルインスタンスストア ボリュームを使用しないようにします。ただし、データを定期的にコピーして EBS または Amazon S3 にバックアップすれば、ローカルインスタンスストア ボリュームは維持されませんが、データを維持することはできます。

コスト

Amazon EC2 インスタンスのコストには、ローカルインスタンスストア (一時) ボリュームすべてが含まれます。ローカルインスタンスストア ボリュームのデータストレージには料金はかかりませんが、Amazon EC2 リージョン外の Amazon EC2 インスタンスストア ボリュームとやり取りされるデータについてはデータ転送料金が発生することに注意してください。また、Amazon S3、EBS ボリューム、EBS スナップショットなど、永続ストレージの使用に対しては追加料金が適用されます。Amazon EC2、EBS、およびデータ転送の価格設定の詳細については、<http://aws.amazon.com/ec2/pricing/> をご覧ください。

弾力性と拡張性

ローカルインスタンスストア ボリュームは、指定された Amazon EC2 インスタンスタイプについてはサイズが固定されており、特定のインスタンスに関連付けられています。したがって、このタイプのストレージはかなり弾力性に欠けています。ただし、Amazon S3、Elastic Block Storage (EBS) など、他の適切なストレージオプションのいずれかを、Amazon EC2 ストレージ戦略に取り入れると、完全なストレージ弾力性を実現できます。

インターフェイス

Amazon EC2 インスタンスは、ローカルディスクドライブのように動作するローカルインスタンスストア ボリュームと連携します。つまり、ローカルインスタンスストレージでホストされているデータを、Windows NTFS、Linux XFS など、ネイティブのファイルシステムメカニズムを使用して操作することができます。状況によっては、ローカルインスタンスストア ボリューム デバイスは、起動時に Amazon EC2 インスタンスに接続されますが、使用前に適切なファイルシステムでフォーマットして、マウントする必要があります。

Amazon EC2 ローカル インスタンス ストア ボリューム アンチパターン

アマゾン EC2 ローカル インスタンス ストア ボリュームは無料の高速「スクラッチボリューム」（Amazon EC2 インスタンスの価格に含まれます）です。これは、簡単に再生成できる一時データの保存に最適です。ただし、ある状況では、他の AWS ストレージオプションの方が適していることがあります。

- **永続ストレージ**：1 つの Amazon EC2 インスタンスの運用期間よりも長く維持する必要があるファイルまたはその他のデータに対して、物理ディスクドライブに似た永続仮想ディスクストレージを必要とする場合は、EBS ボリュームまたは Amazon S3 の方が適しています。
- **データベースストレージ**：ほとんどの場合、データベースには、1 つの Amazon EC2 インスタンスの運用期間よりも長く維持されるストレージが必要なため、EBS ボリュームを選択するのが自然です。
- **共有ストレージ**：インスタンス ストア ボリュームは、1 つの Amazon EC2 インスタンス専用です。他のシステムまたはユーザーと共有することはできません。1 つのインスタンスからの接続を解除し、他のインスタンスに接続できるストレージが必要な場合、または、データを簡単に共有する機能が必要な場合は、Amazon S3 または EBS ボリュームを選択する方が適切です。
- **スナップショット**：ポイントインタイムスナップショットに対して利便性、長期耐久性、可用性を求める場合、またスナップショットを共有できるようにしたい場合は、EBS ボリュームを選択する方が適切です。

Amazon Simple Storage Service (Amazon S3)

Amazon S3 は、ミッションクリティカルな主要データストレージ向けに設計された拡張性、耐久性、可用性に優れた分散オブジェクトストアで、使いやすいウェブ サービス インターフェイスを備えています。従来の構内アプリケーションでは、この種類のデータは、通常、SAN または NAS で維持されています。ただし、Amazon S3 などのクラウドベースのメカニズムの方が、格段に敏捷性、柔軟性に優れ、地理的冗長性に優れています。Amazon S3 には、シンプルなウェブ サービス インターフェイスが用意されています。このインターフェイスを使用すると、いつでも、Amazon EC2 内からでも、ウェブ上のどこからでも、容量に関係なくデータを保存および取得できます。それぞれ 1 バイトから 5 テラバイトまでのデータが含まれるオブジェクトの書き込み、読み取り、および削除が可能です。また、Amazon S3 バケットに保存できるオブジェクトの数に制限はありません。また、Amazon S3 は拡張性に優れているので、多数の個別のクライアントまたはアプリケーションスレッドが同時に読み取りまたは書き込みアクセスできます。

理想的な使用シナリオ

Amazon S3 の使用方法として一般的なものは、静的なウェブコンテンツを保存するというものです。Amazon S3 の各オブジェクトに一意の HTTP URL アドレスがあるため、このコンテンツは Amazon S3 からウェブサーバーに直接配信できます。また、Amazon CloudFront などの Content Delivery Network (CDN) 経由で配信することもできます。Amazon S3 は弾力性に優れているため、帯域幅の需要が「急激に」高まるウェブコンテンツをホストする際に特に役立ちます。また、ストレージプロビジョニングもないので、ビデオおよび写真共有など、データ量が多いユーザー生成コンテンツをホストし、急速に拡大するウェブサイトにも適しています。

Amazon S3 は、金融取引またはクリックストリームデータ、メディアトランスコーディングなど、大規模コンピューティング用のデータストアとしてもよく使用されます。Amazon S3 の水平拡張性により、1 つの接続に制限されずに、複数のコンピューティングノードから同時にデータにアクセスできます。

最後に挙げるのは、Amazon S3 は、ミッションクリティカルなデータ、Amazon EBS ボリューム用の非常に信頼性の高いスナップショットストレージ、バックアップストレージアプリケーション、およびビジネス継続性を目的とした「ホットな」障害復旧ソリューションのオリジナルストアとして使用されることが非常に多いという点です。Amazon S3 では、複数の施設の複数のデバイスにオブジェクトが冗長的に保存されるので、こうしたシナリオに必要な耐久性に優れたストレージインフラストラクチャが実現します。

パフォーマンス

同じリージョンの Amazon Elastic Compute Cloud (Amazon EC2) 内からの Amazon S3 へのアクセスは非常に高速です。複数のスレッド、複数のアプリケーション、または複数のクライアントを同時に使用して Amazon S3 にアクセスすると、総 Amazon S3 集計スループットは、通常、1 つのサーバーが生成または使用できる速度を大幅に上回る数値に合わせて拡大します。

関連データへのアクセスの速度を上げるために、多くの開発者が Amazon S3 と Amazon SimpleDB を組み合わせて利用しています。Amazon S3 には実際の情報が保存され、SimpleDB は、関連するメタデータ（オブジェクト名、サイズ、キーワードなど）のリポジトリとして機能します。SimpleDB では自動的にインデックスが設定されるので、メタデータ検索によるオブジェクト参照の特定を非常に効率的に行えます。この結果を利用して、Amazon S3 からオブジェクト自体を見つけ、取得することができます。

耐久性と可用性

Amazon S3 ストレージでは、選択した地理的リージョン内で複数のデバイスおよび施設両方にわたってデータを自動的にかつ同期的に保存することにより、AWS プラットフォームで優れた耐久性と可用性を実現しています。エラー修正が組み込まれ、単一障害点もありません。Amazon S3 は、2 つの施設で同時にデータが紛失しないように設計されているので、ミッションクリティカルなデータの主要データストレージに最適です。実際、Amazon S3 は、1 年間にわたって、オブジェクトあたり 99.99999999%（「イレブンナイン」）の耐久性と 99.99% の可用性を実現しています。冗長性が組み込まれているほか、アプリケーションの障害や、Amazon S3 バージョン管理の誤使用による意図しない削除から Amazon S3 のデータを保護することもできます。MFA 削除によって Amazon S3 バージョン管理を有効にすることもできます。バケットでこの機能を有効にすると、Amazon S3 オブジェクトを削除する際に 2 つのフォームの認証、つまり有効な AWS アカウント資格情報と、物理トークンデバイスの 6 ケタのコード（1 回だけ使用する時間ベースのパスワード）が必要になります。

トランスコードメディア、画像のサムネールなど、必要に応じて簡単に再生成できるクリティカルでないデータについては、Amazon S3 の Reduced Redundancy Storage (RRS) オプションを使用します。このオプションを使用すると、耐久性のレベルを低くしてストレージコストを抑えることができます。RRS オプションを使用して保存されているオブジェクトは、標準の Amazon S3 ストレージを使用して保存されているオブジェクトよりも冗長性が低くなります。いずれの場合も、データは引き続き複数の場所の複数のデバイスに保存されます。RRS は、年間オブジェクトあたり 99.99% の耐久性を提供するよう設計されています。RRS は、標準の Amazon S3 よりも耐久性に欠けていますが、それでも標準のディスクドライブの耐久性の 400 倍です。

コスト

すべての Amazon Web Services と同様、Amazon S3 は従量課金制なので、最低料金も長期契約もありません。Amazon S3 には、ストレージ（GB あたり 1 か月ごと）、データ転送（送受信）（GB あたり 1 か月ごと）、およびリクエスト（n 千リクエストあたり 1 か月ごと）の 3 つの価格設定要素があります。新規のお客様については、最大 5 GB の Amazon S3 ストレージを無料で利用できます。Amazon S3 の価格設定の詳細については、<http://aws.amazon.com/s3/pricing/> をご覧ください。

弾力性と拡張性

Amazon S3 は、非常に優れた弾力性と拡張性を自動的に提供するように設計されています。多数のファイルをディレクトリに保存するときに問題が発生する標準のファイルシステムとは異なり、Amazon S3 がバケットでサポートするファイルの数に制限はありません。また、ドライブやサーバーにわたるデータをパーティションで区切るまでは保存できる総データ量に制限があるディスクドライブとは異なり、Amazon S3 バケットが保存できるバイト数は無制限です。オブジェクトをいくつでも保存でき、Amazon S3 は、情報の重複コピーの拡張と、同じリージョン内の他の場所にある他のサーバーへの分散を管理します。このすべてにおいて、Amazon の高パフォーマンスインフラストラクチャが使用されます。

インターフェイス

Amazon S3 は、SOAP と REST 形式両方のウェブサービス API を提供します。この API を使用すると、Amazon S3 オブジェクト（ファイル）を、一意の名前が付けられたバケット（トップレベルのフォルダ）に保存できます。各オブジェクトには、バケット内でオブジェクトの識別子として機能する一意のオブジェクトキー（ファイル名）が必要です。Amazon S3 は、従来のファイルシステムではなくウェブベースのオブジェクトストアですが、ファイルシステム階層（folder1/folder2/ファイル）を容易にエミュレートできます。それには、各ファイルの完全パス名に対応するオブジェクトキー名を作成します。

Amazon S3 でアプリケーションを構築する開発者のほとんどが、高度なツールキットを使用します。AWS およびサードパーティは、次に示すよく使用されるソフトウェア開発言語およびプラットフォームを対象とした Amazon S3 インターフェイスツールキットとライブラリを作成しました。

- **Java**。Java 用 AWS SDK (<http://aws.amazon.com/sdkforjava/>) および AWS Toolkit for Eclipse (<http://aws.amazon.com/eclipse/>) を使用。
- **C#**。 .NET 用 AWS SDK (<http://aws.amazon.com/sdkfor.net/>) および Visual Studio テンプレートを使用。
- **PHP**。 PHP5 の基本インストール、または PHP 用 AWS SDK (<http://aws.amazon.com/sdkforphp/>) を使用。
- **Perl**。 Digest::SHA1、Bundle::LWP、および XML::Simple モジュールを使用。すべて Comprehensive Perl Archive Network (<http://www.cpan.org>) からダウンロードできます。

Amazon S3 が主に API でのアクセスを目的として設計されている一方で、AWS Management Console やサードパーティツールにも、Amazon S3 バケットオブジェクトを操作するグラフィカルおよびコマンドラインインターフェイスが用意されています。また、Amazon S3 は、BitTorrent プロトコルもサポートします。これにより、コンシューマが S3 と他のプロバイダから同時に情報を取得できます。

大規模データのアップロードまたはダウンロードするために、AWS Import/Export サービス (<http://aws.amazon.com/importexport/>) を使用することもできます。AWS Import/Export により、転送用のポータブルストレージデバイスを使用して、AWS 内外への大容量データの転送を高速化できます。Amazon は、デバイスをマウントし、指定した Amazon S3 バケットでデータをコピーします。このアプローチは、通常、インターネットを介して膨大なデータを転送するよりも高速で、コストもかかりません。

RRS では、標準の Amazon S3 と同じインターフェイスが使用されていることに注意してください。RRS オプションは、PUT 操作中に Storage Class プロパティを設定するか、AWS Management Console で [Use Reduced Redundancy Storage] 設定を選択することで、Amazon S3 オブジェクトまたはバケットに対して指定されます。

Amazon S3 アンチパターン

Amazon S3 は、比較的静的で、その耐久性、可用性、弾力性のメリットを利用できる、膨大な種類の情報を保存するのに適しています。ただし、Amazon S3 が最適なソリューションではない状況もいくつかあります。

- **ファイルシステム**：Amazon S3 は、フラットな名前空間を使用します。POSIX 対応のスタンドアロンファイルシステムを意図したものではありません。ただし、区切り記号（一般的には「/」または「\」のいずれか）を使用すると、キーを作成し、指定されたバケット内にファイルシステムの階層フォルダ構造をエミュレートできます。
- **クエリ付きの構造化データ**：Amazon S3 は、クエリ機能をサポートしていません。つまり、特定のオブジェクトを取得するには、バケット名とキーが既にわかっている必要があります。したがって、Amazon S3 をそれだけでデータベースとして使用することはできません。使用パターンとしてよくあるのは、オブジェクトを Amazon S3 に配置し、Amazon SimpleDB を使用して、オブジェクトのメタデータを保持するというものです。また、Amazon RDS などの他のデータベーステクノロジーや、Amazon EC2 インスタンスで実行されているデータベースを使用することもできます。そして、そのデータベースを検索してオブジェクトのバケット名とキーを特定し、Amazon S3 バケットからオブジェクト自体を取得します。
- **急速に変化するデータ**：頻繁に更新する必要があるデータは、EBS、データベースなど、読み取り／書き込み遅延が少ないストレージソリューションで提供の方が適しています。

Amazon Simple Queue Service

Amazon Simple Queue Service (Amazon SQS) は、信頼性と可用性に優れたホスト型メッセージキューサービスを、一時ストレージおよび短い (64 KB 以下) テキストベースデータのメッセージに対して提供します。Amazon SQS キューは、処理を待っているメッセージ (通常は、あるアプリケーションで生成され、他のアプリケーションで使用されるのを待っているメッセージ) の一時データリポジトリです。Amazon SQS メッセージは、Amazon EC2 環境内またはインターネットの任意の場所にあるサーバーまたは分散アプリケーションコンポーネントで送受信できます。Amazon SQS がサポートするキューの数に制限はありません。また、メッセージの順序なし at-least-once 配信をサポートします。

Amazon SQS およびその他のメッセージキューサービスは、通常、非同期通信プロトコルですが、この Amazon SQS は、多くの種類のアプリケーションに対して耐久性のある一時データストレージを提供するストアと見することもできます。Amazon SQS を一時ストレージとして使用すると、一時ディスクファイルなど、他のストレージメカニズムの使用を最小限に抑えることができます。

理想的な使用シナリオ

Amazon SQS は、複数のアプリケーションコンポーネントが自身の作業を緩く連携させながら伝達および管理する必要があるシナリオに適しています。これは特に、一部のコンポーネントの動作が他よりも速いか遅い、または連携する複数のコンポーネントが時間または負荷に応じて変化する、プロデューサ/コンシューマシナリオで発生します。Amazon SQS は、「ソフトウェアの接着剤」として機能し、コンポーネントが緊密に関連付けられていたり、同期操作または決まった数のコンポーネントに依存していなくても、確実に通信できるようにします。

従来、Amazon SQS は、複数のステップの処理パイプラインを調整するという目的で使用します。この場合、各メッセージが、処理する必要があるタスクと関連付けられます。Amazon SQS メッセージによって各タスクが記述され、処理するタスクと、Amazon S3 内のタスクデータへのポインタを示します。

例えば、エンコードするイメージファイルが複数あるとします。SNS ワーカーキューで、ファイルごとに SNS メッセージを作成し、コマンド (jpeg-encode) と Amazon S3 におけるファイルの場所を指定します。必要なイメージ処理ソフトウェアを実行している Amazon EC2 インスタンスのプールは、以下を行います。

1. タスクメッセージを非同期的にキューから抽出します
2. 名前付きファイルを取得します
3. 変換処理します
4. イメージを Amazon S3 に書き戻します
5. 「task complete」メッセージを他のキューに書き込みます
6. 元のタスクメッセージを削除します
7. ワーカーキューのその他のメッセージを確認します

Amazon SQS キューを使用すると、複数のワーカーインスタンスを拡大または縮小できます。また、各ワーカーインスタンスの処理能力を、アプリケーションを変更せずに、全作業負荷に合わせて拡大または縮小することもできます。

パフォーマンス

Amazon SQS は、シングルスレッドの送受信速度ではなく、水平拡張性に合わせて最適化された分散キューシステムです。1 つのクライアントが、1 秒あたり約 5~50 メッセージの速度で Amazon SQS メッセージを送受信できます。受信パフォーマンスをさらに高めるには、1 回の呼び出しで複数のメッセージ (最大 10) をリクエストします。キューに入ったメッセージが受信できるようになるには、数秒かかる場合があります。

耐久性と可用性

Amazon SQS のメッセージは意図的に耐久性が高くなっています。ただし、それは一時的なものです。メッセージが失われたり、利用不能になることを防ぐため、すべてのメッセージは複数のサーバーやデータセンターに冗長的に保存されます。メッセージ保存時間は、1 時間～ 14 日の範囲でキューごとに設定できます。メッセージは、明示的に削除されるか、保存期間の期限が切れて自動的に削除されるまで、キューに保持されます。

コスト

すべての Amazon Web Services と同様、Amazon SQS は従量課金制なので、最低料金も長期契約もありません。Amazon SQS を開始し、シンプルなアプリケーションをサポートするために、Amazon SQS には無料のサービスが用意されており、1 か月あたり 100,000 のリクエストに対応できます。無料のサービスを超えた場合、Amazon SQS の価格は、リクエスト数（10,000 リクエストあたりの価格）およびデータ転送量（送受信）（GB あたり 1 か月ごとの価格）に基づいて設定されます。Amazon SQS の価格設定の詳細については、<http://aws.amazon.com/sqs/pricing/> をご覧ください。

弾力性と拡張性

Amazon SQS は、弾力性に優れ、可用性も極めて高くなっています。この Amazon SQS は、無数のコンピュータが無数のメッセージをいつでも読み込んだり書き込んだりできるよう設計されています。任意のユーザーについて、サポートするキューの数と、キューあたりのメッセージの数に制限はありません。

インターフェイス

Amazon SQS には、SOAP および Query (HTTP) インターフェイスを使用してアクセスできます。CreateQueue、SendMessage、ReceiveMessage、ChangeMessageVisibility、DeleteMessage の 5 つの API により、開発者が SQS を開始しやすくなっています。追加 API は高度な機能を提供します。すべての状況で、SOAP およびクエリ API は、Java、C#、Perl、および PHP で使用できます。

Amazon SQS アンチパターン

- **バイナリまたは大きなデータ** : Amazon SQS メッセージはテキストである必要があります。また、長さの最大値は 64 KB です。この長さを超えるデータをキューに保存する必要がある場合、またはバイナリデータの場合は、Amazon S3 または RDS を使用して大きなデータまたはバイナリデータを保存し、データへのポインタを Amazon SQS に保存するのが適切です。
- **長期ストレージ** : メッセージデータを 14 日を超えて保存する必要がある場合は、Amazon S3 またはその他のストレージメカニズムの方が適しています。

Amazon SimpleDB

Amazon SimpleDB では、新しいアプローチで、クラウド内の構造化データを保存および管理できます。これは、従来のリレーショナル データベース サーバーの展開手法とは異なります。SimpleDB は、可用性、拡張性、および柔軟性に優れた非リレーショナル データ ストアで、データベース管理と関連するシステム管理の負担を大幅に軽減します。Amazon SimpleDB により「スキーマレス」データモデルが提供され、様々な数の名前/値ペアで構成されるデータアイテムを保存できます。

理想的な使用シナリオ

このストレージは、高い可用性と耐久性を確保しながら、詳細に設定された構造化データを維持し、クエリ（主に読み取り操作）を実行する必要がある場合、ただし、完全なリレーショナルデータベースの管理オーバーヘッドを発生させたくない、または発生させる必要がない場合に適しています。

組織の多くが、SimpleDB を、大規模なクラウドベース ストレージ アーキテクチャの一部として使用しています。よくある使用パターンは、SimpleDB を使用して、他の AWS サービスに保存されている情報に関するメタデータを追跡するというものです。例えば、何百万マルチメガバイトのイメージを保存するクラウドベースソリューションを構築するとします。この状況では、Amazon Elastic Block Storage (EBS) または Amazon Simple Storage Service (Amazon S3) のいずれかを使用してイメージ自体を保持し、SimpleDB は、各イメージの詳細情報のリポジトリとして機能します。その後、クエリを SimpleDB に発行して特定のイメージを検索し、この情報を使用して、他の Amazon ストレージサービスから結果を取得できます。

パフォーマンス

SimpleDB アプローチでは、Amazon の実績のある高速インフラストラクチャを利用して、可用性と拡張性に優れた高速データベースを作成します。最高のスループットを実現し遅延を最小限に抑えるために、Amazon SimpleDB はデフォルトで最終的に一貫性が保たれる読み取りを提供します。「最後の書き込みの読み取り」を必要とするアプリケーションについては、一貫した読み取りも提供されます。

耐久性と可用性

SimpleDB では、地理的に冗長なレプリケーションによって、非常に高いデータ耐久性が実現しています。すべての SimpleDB データアイテムの複数の複製が、選択した地理的リージョン内の様々な場所に保存されます。

コスト

すべての Amazon Web Services と同様、Amazon SimpleDB は従量課金制なので、前払い金も長期契約もありません。SimpleDB のコストには、PUT 操作および GET 操作に関連付けられている構造化データストレージ (GB あたり 1 か月ごと)、データ転送 (GB あたり 1 か月ごと)、およびマシン時間 (1 か月ごと) が含まれます。最初の 1 か月の構造化データストレージ 1GB、および 1 か月の最初の 25 マシン時間 (約 2M GET または SELECT API リクエスト) については無料です。この無料の価格設定により、アプリケーションの多くで SimpleDB に対する料金が発生しません。

Amazon SimpleDB の価格設定の詳細については、<http://aws.amazon.com/simpledb/pricing/> をご覧ください。

弾力性と拡張性

SimpleDB ドメインを作成するとき、AWS は、ドメインあたり最大 10 GB のストレージを割り当てます。使用するコンピュートおよびストレージリソースに対してのみ料金を支払うので、透過的にゼロまで縮小したり拡大したりできます。ストレージおよびインデックス要件は Amazon SimpleDB によって処理され、自動的にインデックスが付けられたデータは、SimpleDB ドメインを作成したときに選択したリージョン内の複数の場所にわたって冗長的に保存されます。これによりデータベース管理が不要になり、データに対してただ PUT 操作と GET 操作だけを行えばよくなります。スループットを向上させる水平拡張性、または 1 つのドメインの制限

を超えるデータセットサイズを実現するために、複数の SimpleDB ドメインを作成し、複数の操作を並行して実行できます。また、10 GB を超えるデータセットについては、複数の SimpleDB ドメインにわたるデータをパーティションで区切ることを検討します。

インターフェイス

様々な手法を自由に利用して、プログラムによって SimpleDB を操作できます。Amazon にはウェブサービス API が用意されており、SOAP と REST 形式の両方で利用できます。API が提供する操作は現在 9 個あり、ドメイン管理（メタデータの作成、リスト表示、削除、および取得）と属性の操作（取得、書き込み（個別または一括）、および削除）を可能にします。標準の SQL は SimpleDB では使用できませんが、SimpleDB Select 操作を使用して、指定した条件に基づいて一連の属性を取得する SQL に似たクエリを作成することはできます。

ソフトウェア開発キットを使用したい場合は、次に示すよく使用されるソフトウェア開発言語およびプラットフォームを対象としたツールキットとライブラリを利用できます。

- **Java**。Java 用 AWS SDK (<http://aws.amazon.com/sdkforjava/>) および AWS Toolkit for Eclipse (<http://aws.amazon.com/eclipse/>) を使用。
- **C#**。 .NET 用 AWS SDK (<http://aws.amazon.com/sdkfornet/>) および Visual Studio テンプレートを使用。
- **PHP**。 PHP5 の基本インストール、または PHP 用 AWS SDK (<http://aws.amazon.com/sdkforphp/>) を使用。
- **Perl**。 Digest::SHA1、Bundle::LWP、および XML::Simple モジュールを使用。すべて Comprehensive Perl Archive Network (<http://www.cpan.org>) からダウンロードできます。

Amazon SimpleDB アンチパターン

どの使用シナリオが SimpleDB に適していないかを判断するのは簡単です。次の条件のいずれかが当てはまる場合は、他の AWS ストレージオプションのいずれかを使用することを検討してください。

- **事前に記述されたアプリケーションが従来のリレーショナルデータベースに関連付けられている**：既存のアプリケーションを AWS クラウドに移植しようとしている場合、リレーショナルデータベースを引き続き使用する必要があるときは、Amazon RDS（データベースが MySQL の場合）または事前設定された多数の Amazon EC2 データベース AMI のいずれかを選択します。独自の Amazon EC2 インスタンスを作成し、データベースエンジンをそのインスタンスにインストールすることもできます。
- **結合／複合トランザクション**：多くのソリューションが SimpleDB を利用してユーザーをサポートできますが、従来のデータベースプラットフォームが提供する他のリレーショナルインフラストラクチャ、結合トランザクション、複合トランザクションをアプリケーションが必要とする場合があります。この場合は、Amazon RDS または Amazon EC2、およびインストールされているデータベースを試してみるとよいでしょう。
- **BLOB（バイナリ ラージオブジェクト）データ**：バイナリデータ（ビデオ、画像、音楽など）を保存する場合は、Amazon Elastic Block Storage（EBS）または Amazon Simple Storage Service（Amazon S3）について検討することをお勧めします。ただし、このシナリオでも SimpleDB には役割があり、バイナリオブジェクトに関するメタデータ（アイテムの名前、サイズ、作成日、所有者、場所など）を記録する必要があります。

- **定型（数値）データ**：SimpleDB では、すべてのデータがテキスト文字列として保存されます。したがって、定型（または数値）データを操作する必要がある場合は、RDS または Amazon EC2 の従来のデータベースを使用したソリューションを使用することをお勧めします。
- **大量のデータ**：前に説明したように、SimpleDB が提供する価格設定やストレージモデルは多くのアプリケーションにとって非常に魅力的です。ただし、大量のデータセットを管理しなければならない場合は、Amazon S3、EBS などのストレージも検討してください。繰り返しになりますが、この場合も SimpleDB は、大量のデータセットのメタデータを追跡するための効率的なメカニズムとして引き続き利用できます。

Amazon Relational Database Service (Amazon RDS)

Amazon Relational Database Service (RDS) は、MySQL リレーショナルデータベースのすべての機能をマネージド型のクラウドベースサービスとして提供するものです。アプリケーションにリレーショナルストレージが必要だが、データベース管理の時間を短縮したい場合、Amazon RDS を使用すると、手間と総所有コストを削減し、一般的な管理作業の自動化を実現できます。Amazon RDS は自動的にデータベースをバックアップし、管理します。これによって、開発者はアプリケーション開発により多くの時間を割くことが可能となります。これは、Amazon の他のデータベースサービス SimpleDB の機能よりも高度なクエリおよび結合機能を必要とする、従来の構造化データに適しています。

理想的な使用シナリオ

Amazon RDS は、MySQL を情報リポジトリとして使用しているすべてのアプリケーションに適しており、コードを変更せずに、拡張性と費用対効果に優れ、メンテナンスの手間がかからない、クラウドベースのデータベースを利用したい場合にお勧めします。

パフォーマンス

Amazon RDS では、Amazon の実績ある世界レベルのインフラストラクチャで実行されている設定可能なインスタンスと、完全に自動化されたメンテナンスおよびバックアップ操作の組み合わせにより、優れたパフォーマンスが実現しています。使用できるのは、小さなインスタンス（64 ビットプラットフォーム、1.7 GB の RAM と 1 つの Elastic Computing Unit (ECU) を搭載）から、その 4 倍の非常に大きなインスタンス（64-bit platform with 68 GB of RAM and 26 ECUs）までのデータベース構成です。

最適なパフォーマンスを実現するために、設計者と管理者は、特定のコンピューティングニーズに応じて適切なインスタンスプロファイル（RAM およびストレージを含む）を選択する必要があります。調整が必要な場合は、Amazon が提供する API を使用してデータベース設定を調整できます。Amazon CloudWatch は、パフォーマンスの調整や弾力性の強化が必要なタイミングを判断するのに役立つ指標を提供します。需要が現在のインスタンスのリソースを上回る場合は、最終的に、より大きなデータベース構成に移行することを管理者が選択することもあります。

弾力性と拡張性

RDS リソースは、データベースストレージサイズ、データベース インスタンス計算処理能力、および読み取り複製の数など、複数の側面から弾力的に拡張できます。

計算処理の弾力性を高めるには、追加の Amazon RDS インスタンスを設定し、パーティションを利用して、作業負荷を分散します。

RDS データベースを弾力的に拡張するには、コマンドラインツール、API、または AWS Management Console を使用して、追加のストレージをリクエストします。このストレージ追加処理は、ニーズに応じて、すぐに行うことも次のメンテナンス時間に行うことも可能です。

計算リソースは、1 つの API または AWS Management Console コマンドで簡単に拡大または縮小できます。例えば、追加の計算処理機能で、各月の末に請求書を作成しなければならないことがあります。この場合は、4 倍の非常に大きなインスタンスに一時的に拡大し、そこで膨大な作業を計算処理して、月末以外は費用対効果の高い小さな設定に戻します。この作業は簡単に行うことができます。

RDS データベースリソースを拡大または縮小するために（複数の RDS インスタンスを使用した水平拡張）、管理者は 1 つ以上の RDS 読み取り複製を作成できます。読み取り複製は、MySQL の組み込み非同期レプリケーション機能を使用して、RDS リソースを弾力的に拡大し、読み取りが多い作業をサポートできるようにします。

最後に、管理者は追加の Amazon RDS インスタンスを設定し、データベースのパーティションまたはシャーディングを利用して作業負荷を分散し、データベースの弾力性と拡張性を高めます。

耐久性と可用性

耐久性を高めるために、RDS には、複数の利用可能ゾーンにわたってレプリケートされる 2 種類のデータベースバックアップが用意されています。1 つは自動 DB インスタンスバックアップ、もう 1 つはユーザーが開始するデータベーススナップショットです。自動 DB インスタンスバックアップを有効にすると、指定したバックアップ時間に、完全バックアップが毎日自動実行され、DB トランザクションログが取得されます。この自動バックアップは追加料金なしで行われ、最大 8 日間、保存されます。このバックアップを使用すると、保存期間が開始してから、現在の時刻の約 5 分前までの間の任意のポイントに、ポイントインタイム復元することができます。DB スナップショットはユーザーが開始し、いつでも作成できます。また、明示的に削除されるまで保持されます。DB スナップショットを使用すると、データベースを既知の状態に復元できます。

RDS Multi-AZ 展開機能は、主要 RDS DB インスタンスと、他の利用可能ゾーンのスタンバイインスタンス間で同期的にデータをレプリケートすることで、データベースの耐久性と可用性の両方を強化します。万一 DB コンポーネントまたは利用可能ゾーンで障害が発生した場合、RDS はスタンバイに自動的にフェールオーバーし（通常約 3 分かかります）、スタンバイが昇格するとすぐにデータベーストランザクションが再開します。レプリケーションは同期的に行われるので、データが失われることはありません。

RDS Multi-AZ 展開機能が提供する同期レプリケーションは、RDS 読み取り複製が提供する組み込みの非同期レプリケーションを補完するものです。それぞれの機能を単独で使用することも、両方を組み合わせて使用することも可能です。

コスト

すべての Amazon Web Services と同様、Amazon RDS は従量課金制なので、最低料金も長期契約もありません。Amazon RDS の価格構成は、データベースインスタンスのサイズ、展開の種類（Single-AZ/Multi-AZ）、および AWS リージョンに基づいており、DB インスタンス時間（時間ごと）、プロビジョニングされたデータベース

ストレージ（GB あたり 1 か月ごと）、追加のバックアップストレージ（GB あたり 1 か月ごと）、データ転送（送受信）（GB あたり 1 か月ごと）など、複数の要素に基づいて価格が設定されます。RDS の価格設定の詳細については、<http://aws.amazon.com/rds/pricing/> をご覧ください。

インターフェイス

Amazon には、データベースインスタンスを作成、維持、モニタリング、および停止するための一連のわかりやすいコマンドラインスクリプトが用意されており、開始するのに役立ちます。また、Amazon のウェブサービススペースの操作によって、同様の一連の作業を行うこともできます。

インスタンスが作成されたら、リレーショナルデータベースで使用できる任意のツールを使用して、スキーマとデータを自由に設定できます。既存のオンサイトデータの量と場所によっては、mysqldump ユーティリティを使用してローカルデータを抽出し、MySQL 実行可能ファイルに直接パイプして、Amazon RDS に挿入できる場合があります。大きなデータセットの場合は、最初にスキーマを Amazon RDS に作成し、データをローカルのテキストファイルに抽出して、mysqlimport ユーティリティを使用して Amazon RDS にアップロードするとさらに有利な場合があります。Amazon RDS への移行については、今後のホワイトペーパーで詳しく説明します。

スキーマとデータの準備ができたら、標準の MySQL SQL、JDBC とその他の一般的な API、およびリレーショナルデータを操作できるグラフィカルツールを使用して情報を操作します。コードを変更しなくても、つまり、インスタンスの作成時に、データベースサーバーのアドレス（dbserver.yourcompany.com）を、AWS から提供されたパブリック DNS エンドポイント（例：myinstance.c0cafngtpzd2.us-east-1.rds.amazonaws.com）と置き換えるだけで、アプリケーションと RDS の間で情報をやり取りできます。この DNS エンドポイントは、インスタンスの運用期間中は同じ状態に保たれます。エンドポイントの設定だけではなく、MySQL ベースのアプリケーションに関する他のすべてが変更されません。

Amazon RDS アンチパターン

Amazon RDS は、クラウドベースの MySQL データには優れたソリューションですが、適していないシナリオもいくつかあります。

- **インデックスおよびクエリ中心のデータ**：クラウドベースのソリューションは、多くの場合、リレーショナルデータベースの高度な機能（結合トランザクション、複合トランザクションなど）を必要としません。アプリケーションでインデックスを設定したりクエリを実行したりする機会が多い場合は、Amazon SimpleDB の方がニーズに対応していると思われます。
- **大量のバイナリラージオブジェクト（BLOB）**：BLOB 保存用として MySQL（および Amazon RDS）が用意されていますが、アプリケーションがその BLOB（音声ファイル、画像など）を頻繁に使用する場合は、Amazon Simple Storage Service（Amazon S3）の方が適しています。
- **弾力性が自動化されている場合**：前に説明したように、Amazon RDS 環境を設定して弾力性を実現するのは管理者です。弾力性を自動化する必要がある場合（また、データ構造が適切である場合）は、Amazon SimpleDB、Amazon S3 などのストレージを選択します。
- **他のデータベースプラットフォーム**：現時点では、Amazon RDS には MySQL データベースが用意されています。他のデータベースプラットフォーム（IBM DB2、Informix、Microsoft SQL Server、Oracle、PostgreSQL、Sybase など）が必要な場合は、特別な Amazon EC2 リレーショナルデータベース AMI を展開するか、独自のものを作成します。

Amazon EC2/Relational Database AMI のリレーショナルデータベース

Amazon EC2 と EBS ボリュームは、クラウドで独自のリレーショナルデータベースを操作するのに最適なプラットフォームを提供します。主要データベースソリューションの多くが、事前作成されたすぐに使用できる Amazon EC2 AMI (IBM DB2、Informix、Oracle Database、MySQL、Microsoft SQL Server、PostgreSQL、Sybase、EnterpriseDB、および Vertica を含む) として利用できます。

理想的な使用シナリオ

従来のリレーショナルデータベースを必要とするアプリケーションのユーザー、または最大レベルの管理と設定を必要とするユーザーについては、Amazon EC2 および EBS でリレーショナルデータベースを実行するのが理想的です。

パフォーマンス

Amazon EC2 のリレーショナル データベース インスタンスのパフォーマンスは、Amazon EC2 インスタンスタイプ、EBS ボリュームの数と設定、データベースソフトウェアとその設定、アプリケーションの作業負荷など、多くの要因に依存します。一般的に、Amazon EC2 のデータベースには、同様に設定された構内の装置にインストールされている同じデータベースと同程度のパフォーマンスを期待できます。最善の設定を選択する目的で、複数のストレージ設定を使用して、複数のタイプの Amazon EC2 インスタンス上で実際のアプリケーションを評価することをお勧めします。

データベースのパフォーマンスを向上させるには、より大きなサイズの Amazon EC2 インスタンスを選択してメモリーおよびコンピュートリソースを増やします。I/O パフォーマンスを高めるには、EBS ボリュームの数を変更するか、複数の EBS ボリュームにわたってソフトウェア RAID 0 (ディスクストライピング) を使用します。これにより、総 IOPS と帯域幅が向上します。多くの場合、データベースのクラスタリング、レプリケーション、および複数の読み取りスレーブで水平的に拡張して、データベースシステムのパフォーマンス全体を向上させることもできます。一般的に Amazon EC2 環境のデータベースパフォーマンス調整オプションは、物理サーバー環境のオプションと同じです。

耐久性と可用性

Amazon EC2 のリレーショナルデータベースは、EBS ボリュームをデータストアとして使用して、構造化データに対して永続ストレージを提供します。したがって、EBS データの耐久性と可用性に関するすべての注意事項がここにも当てはまります。また、繰り返しになりますが、EBS ボリュームに保存されているリレーショナルデータの耐久性と可用性は、EBS スナップショットを使用するか、サードパーティデータベースのバックアップユーティリティ (Oracle の RMAN など) を使用してデータベースバックアップを Amazon S3 に保存することで、さらに高めることができます。

コスト

Amazon EC2 でのデータベースの実行は従量課金制です。最低料金も長期契約もありません。Amazon EC2 で独自のデータベースを実行するコストは、データベースを実行する際に使用される Amazon EC2 のサイズと数、データベースストレージで使用される EBS ボリュームのサイズ、および Amazon EC2 で送受信される転送データの量に依存します。また、サードパーティ データベース ソフトウェアのライセンスコストに左右されること

も少なくありません。オープンソース データベース パッケージの多くが、コストがかからないライセンスモデルを採用しており、一部の市販のソフトウェアベンダーは Amazon DevPay モデルを採用しています。また、独自のライセンスモデルを持つベンダーも多数あります。適用されるライセンスコストの価格設定モデルについては、データベース ソフトウェア ベンダーまたは Amazon Web Services にお問い合わせください。

Amazon EC2、EBS、およびデータ転送の価格設定の詳細については、<http://aws.amazon.com/ec2/pricing/> をご覧ください。

弾力性と拡張性

多くの場合、Amazon EC2 の従来のリレーショナル データベース ソリューションのユーザーは、基盤となる AWS プラットフォームの弾力性と拡張性を利用できます。例えば、データベースソリューションで Amazon EC2 インスタンスを設定したら、そのインスタンスをカスタム AMI にバンドルできます。これを行うには、インスタンスストア AMI については Bundle コマンドを、EBS AMI については Create Image コマンドを使用します。その後すぐに、データベース構成の複数のインスタンスを新しく作成することができます。

Amazon EC2 のリレーショナルデータベースのアンチパターン

多くのユーザーが、優れたソリューションとして Amazon EC2 で独自のリレーショナルデータベースを実行できます。ただし、他のソリューションが適切なシナリオもいくつかあります。

- **インデックスおよびクエリ中心のデータ**：クラウドベースのソリューションは、多くの場合、リレーショナルデータベースの高度な機能（結合トランザクション、複合トランザクションなど）を必要としません。アプリケーションでインデックスを設定したりクエリを実行したりする機会が多い場合は、Amazon SimpleDB の方がニーズに対応しており、格段に管理が容易であると思われる
- **大量のバイナリ ラージオブジェクト (BLOB)**：多くのリレーショナルデータベースが BLOB（音声ファイル、ビデオ、画像など）をサポートしています。アプリケーションがその BLOB を頻繁に使用する場合は、Amazon S3 の方が適していると思われる。メタデータの管理に、リレーショナルデータベースまたは SimpleDB を使用します。
- **弾力性が自動化されている場合**：前に説明したように、多くの場合、AWS のリレーショナルデータベースのユーザーは、基盤となる AWS プラットフォームの弾力性と拡張性を利用します。ただし、それには、システム管理者または DBA が手動のタスクまたはスクリプト化されたタスクを実行する必要があります。弾力性を完全に自動化する必要がある場合（また、データ構造が適切である場合は）、Amazon SimpleDB、Amazon S3 などのストレージを選択します。

クラウドストレージの使用事例

AWS ストレージオプションの実際の使用方法については、ドキュメント「Storage Options in the AWS Cloud: Use Cases」 (http://media.amazonwebservices.com/AWS_Storage_Use_Cases.pdf) をご覧ください。

参考文献

AWS ストレージサービス

Amazon Elastic Block Store (EBS) – <http://aws.amazon.com/ebs>

Amazon EC2 Instance Store Volumes – <http://docs.amazonwebservices.com/AWSAmazonEC2/latest/UserGuide>
(「Instance Types」、「Instance Storage」、および「Block Device Mapping」を参照)

Amazon Simple Storage Service (Amazon S3) – <http://aws.amazon.com/s3>

Amazon Simple Queue Service (Amazon SQS) – <http://aws.amazon.com/sqs>

Amazon SimpleDB – <http://aws.amazon.com/simpledb>

Running Databases on AWS – http://aws.amazon.com/running_databases

Amazon Relational Database Service (Amazon RDS) – <http://aws.amazon.com/rds>

AWS ストレージの記事

Werner Vogel の記事「Choosing Consistency」 –

http://www.allthingsdistributed.com/2010/02/strong_consistency_simpledb.html

Amazon Web Services ブログの記事「Amazon SimpleDB Consistency Enhancements」 –

<http://aws.typepad.com/aws/2010/02/amazon-simpledb-consistency-enhancements.html>

Amazon Web Services 開発者センターの記事「Amazon SimpleDB Consistency Enhancements」 –

<http://developer.amazonwebservices.com/connect/entry.jspa?externalID=3572&categoryID=152>

Amazon Web Services Simple Monthly Calculator – <http://aws.amazon.com/calculator>

Oracle in the Cloud FAQ – <http://www.oracle.com/technetwork/topics/cloud/faq-098970.html>

Amazon Web Services 開発者センターの記事「Running Databases on AWS」 –

http://aws.amazon.com/running_databases/

Oracle Case Study: Oracle Database in the Cloud –

<http://www.oracle.com/technetwork/database/features/availability/311356-129177.pdf>

その他の AWS 情報

AWS 無料使用範囲 – <http://aws.amazon.com/free>

Public Data Sets on AWS – <http://aws.amazon.com/publicdatasets>

AWS Import/Export – <http://aws.amazon.com/importexport>

Amazon CloudFront – <http://aws.amazon.com/cloudfront>

AWS Case Studies – <http://aws.amazon.com/solutions/case-studies>

付録：AWS ストレージクイックリファレンス

	非構造化データ			構造化データ		
	Amazon EC2 インスタンスストレージ	Amazon EBS ボリューム	Amazon S3	Amazon SimpleDB	他のリレーショナル DB (EC2 および EBS 上)	Amazon RDS
パフォーマンス	高	高	中 (1つのスレッド) ~ 最高 (複数のスレッド)	中~高 (一括 Put/Get)	高	高
耐久性	低	中	高	高	高	中
コスト	EC2 に含まれる	プロビジョニング済み GB あたり月ごと	保存済み GB あたり月ごと	プロビジョニング済み最初の GB は無料、その後 GB あたり月ごと	プロビジョニング済み (EBS と同じ)	プロビジョニング済み GB あたり月ごと (5 GB 以上)
可用性	低	中~高 (EBS スナップショットを使用)	高	高	中~高	高
弾力性/拡張性	いいえ	手動 (ボリューム追加)	自動	自動	手動	手動 (1つのコマンドで DB インスタンスを変更)
サイズ制限	160 GB ~ 1.6 TB (インスタンスが大きくなると、ボリュームのサイズと数が増大)	ボリュームあたり 1 GB ~ 1 TB (大容量に対応するために複数のボリュームまたはストライピングを使用可能)	事実上無制限 (オブジェクトあたり 5 TB、バケットあたり無制限のオブジェクト)	10 GB/ドメイン 100 ドメイン (リクエスト時にさらに多くのドメインを使用可能)	(EBS と同じ)	DB インスタンスあたり 5 GB ~ 1 TB
インスタンスにわたって保持	いいえ	はい	はい	はい	はい	はい
インターフェイス	ブロックデバイス、EC で OS/ファイルシステムを介してアクセス	該当なし、EC2 OS/ファイルシステムを介してアクセス	HTTP、REST または SOAP	REST または SOAP	MySQL または JDBC ライブラリ	MySQL または JDBC ライブラリ
セキュリティ (停止時に暗号化)	暗号化された FS を実行	暗号化された FS を実行	256 ビット AES を使用して暗号化	256 ビット AES を使用して暗号化		
セキュリティ (移行中に暗号化)	該当なし	該当なし	SSL (HTTPS)	SSL (HTTPS)	SSL (HTTPS)	SSL (HTTPS)
RDBMS プラットフォームをサポート	MySQL、SQL Server、Oracle、DB2 など	MySQL、SQL Server、Oracle、DB2 など	該当なし	該当なし	MySQL、SQL Server、Oracle、DB2 など	MySQL 5.1
モデル (リレーショナルまたはその他)	ブロック	ブロック	オブジェクト	柔軟な非リレーショナルスキーマ、エンティティストア	リレーショナル	リレーショナル

	非構造化データ			構造化データ		
	Amazon EC2 インスタンスストレージ	Amazon EBS ボリューム	Amazon S3	Amazon SimpleDB	他のリレーショナル DB (EC2 および EBS 上)	Amazon RDS
自動化の程度	なし	自動ミラーリング	自動レプリケーション	インデックス設定、レプリケーション、プロビジョニング、パッチの適用	DB に依存	自動バックアップ、ソフトウェアパッチの適用
冗長性の程度	冗長性なし	利用可能ゾーン内では冗長	複数のデータセンターにわたり高い冗長性を実現	すべてのユーザーデータの様々な場所に保存された複数のコピーを保持	なし (非同期レプリケーションが使用可能)	1 つの DB インスタンス (1 つの AZ) および Multi-AZ オプションの両方を提供
クロスインスタンスアクセス (共有可能性)	いいえ	いいえ	はい	はい	はい	はい
管理	手動	手動	自動	自動	手動	自動