Evidian SafeKit

重要なアプリケーションを保護するための高可用性ソフトウェアソリューション

本ホワイト ペーパーでは高可用性ソリューションの SafeKit について説明します。SafeKit は、既存の標準的なサーバーに展開できるソフトウェア クラスタリング ソリューションであり重要なアプリケーションを高可用化します。

高可用性
ビジネスの継続性と障害復旧

コンピュータ システムを利用するすべてのアクティビティは、その大小に関係なく、コンピュータ障害の問題に直面する可能性があります。もしも、残念なことに高可用性ソリューションが実装されていないコンピュータシステム上で障害が発生してしまった場合、それが例え小さな問題であっても、ビジネスの全体的な危機につながる可能性があります。

SafeKit ソフトウェア クラスタリングが選ばれる 10 の理由

1. ソフトウェアのみで実現する高可用性ソリューション

Evidian SafeKit（以下、「SafeKit」という。） は、追加ハードウェアが不要なソフトウェアのみで構成可能な高可用性ソリューションです。このソリューションは、重要なアプリケーションの連続運用を容易にかつ迅速に保護します。

SafeKit は、既存の物理サーバーまたは仮想サーバーで動作し、標準的な OS およびデータベースで使用できます。

従来の高可用性ソリューションは物理サーバーまたは仮想サーバーのフェイルオーバーが対象でしたが、SafeKit は重要なアプリケーションのフェイルオーバーに注目しています。

このアプリケーション中心でハードウェアに依存しないソリューションは、クラウド内の重要なアプリケーションに対しても高可用性を容易に提供できます。

2. あらゆる種類の障害に対処

アプリケーションを利用できなくなる問題には 3 つの種類があります。

- ハードウェアと環境　：　損傷（コンピュータ ルームの損傷を含みます。）
- ソフトウェア　：　ソフトウェアの更新での障害、サービスの過負荷、ソフトウェアのバグ
- 人的ミス　：　管理・監視エラーおよび操作の失敗

SafeKit は、重要なアプリケーションの高可用性の保証にとって極めて重要なこれらの問題に対処します。

3. ソフトウェアクラスタリングの 3 つの最善な使用法

SafeKit は、15 年を超える経験を経て、3 つのケースにおけるクラスタリング ソリューション市場での地位を確立してきました。

1. ソフトウェアベンダーは、EvidianからOEMされたSafeKit を利用して自社のアプリケーション スイートへ容易にソフトウェア高可用性オプションを追加できます。
2. ソリューションベンダーは、高可用性に関する特別な IT スキルがなくても、標準的なハードウェア上に高可用性ソリューションを展開できます。
3. データ センターベンダーは、Windows と Linux プラットフォーム上のソリュー ションとして、ロード バランシング、リアルタイム データ レプリケーション、 フェイルオーバーと共に、高可用性を複数のアプリケーションに提供できます。

4. 市場で唯一の統合化された製品

従来、アプリケーション クラスターを構築するには 以下の3 つの異なる製品が必要でした。

- ロード バランシング ネットワーク ボックス
- データ可用性のために SAN で同期複製されるディスク ベイ
- アプリケーション障害復旧のための高可用性ツールキット

SafeKit は、これら 3 つの機能を一つのソフトウェア製品で提供します。
さらに、実装コストを抑えるため、SafeKit は、既存の物理サーバーまたは仮想サーバーで動作し、標準的なOS およびデータベースで使用できます。Oracle、Microsoft SQL Server、その他のデータベースまたはファイル、そして Windows for PC にも対応します。

5. ソフトウェアクラスターのプラグアンドプレイ展開

フェイルオーバー モジュールをアプリケーションに合わせて構成しテストした後は、特別な IT スキルがなくても展開できます。アプリケーション、SafeKit ソフトウェア、そしてフェイルオーバー モジュールをWindows サーバーまたは Linux サーバーにインストー ルするだけです。

6. アプリケーション統合の豊富な選択肢

SafeKit はさまざまな種類のソフトウェア クラスターに対応しています。アプリケーションに対するクラスター構成の選択肢は豊富で、1 つまたは複数のアプリケーション モジュールで構成されます。SafeKit は、ミラー モジュール (プライマリ/セカンダリ レプリケーションおよびフェイルオーバー)、ファーム モジュール (ネットワーク ロード バランシングおよびフェイルオーバー)、および複数のモジュールの混合構成に対応可能です。モジュールは、サーバーとクラスターの名前、ファーム モジュールのロード バランシング規則、ミラー モジュール用に複製するファイル ディレクトリ、ハードウェアおよびソフトウェアの障害検出機能、障害時に再開するサービスで構成されます。

7. 人的ミスを防ぐ為のユーザーフレンドリな管理コンソール

管理者はSafeKit の Web コンソールで、クラスター内のアプリケーションの状態 (レッド、グリーン、マゼンタ色で状態を識別) を遠隔監視し、ボタン操作 (start、stop) だけで別のサーバーのアプリケーションを再起動できます。

提供されるマニュアルにドキュメントには、高可用性モードでのアプリケーションの適切な機能を検証するためのテスト方法、トラブルシューティングの手順が含まれています。

SafeKit の汎用コマンド ライン インターフェースを使用すると、SafeKit によって保護される重要なアプリケーションの監視を、特定の管理コンソール に容易に組み込むことができます。

8. アプリケーションの為の同期レプリケーション

SafeKit の同期レプリケーション機能は、高可用性機能を強化し、データの損失を防ぎます。このメカニズムでは、トランザクションを処理するアプリケーションによってディスクにコミットされたデータは、セカンダリ マシンに複製されます。

アプリケーション サーバーは、地理的に離れたコンピュータ ルームに置いて拡張 LAN によって接続でき、コンピュータ ルームがそっくり失われるような災害にも耐えることができます。

9. 非同期レプリケーション下でのデータ損失の防止

非同期レプリケーション (他の市販ソリューションによる従来の実装) では、プライマリ サーバーで障害が発生した場合にデータが失われる高いリスクがあります。したがって、重要なアプリケーションの高可用性のためには、SafeKit のような同期レプリケーション ソリューションの方が常に推奨されます。

10. ソフトウェアクラスターを 1 時間で実装

無料で SafeKit をテストできます。わずか 1 時間で、2 台の仮想マシンまたは物理マシンにソフトウェア クラスターを実装できます。

ソフトウェアクラスタリング

SafeKit アプリケーションモジュールで Windows および Linux クラスターを構成

アプリケーション中心の高可用性ソリューション

SafeKitアプリケーション モジュールには、SafeKit ソフトウェア クラスター機能をアプリケーションで使用出来るように高可用性構成とリカバリー手順が定義／提供されています。SafeKitアプリケーション モジュールでは次のことが可能です。

- アプリケーションと共にプラグアンドプレイでソフトウェア クラスターが展開出来ます。
- 用途に応じてソフトウェア クラスターの種類が選択出来ます。
- ソフトウェア クラスター内でアプリケーション リカバリーを構成出来ます。
- テンプレートの提供により新しいアプリケーションの統合に要する時間を短縮出来ます。

ソフトウェア クラスターのプラグアンドプレイ展開

アプリケーション モジュールを構成しテストした後は、特別な IT スキルがなくても展開できます。

1. 2 台の標準的な Windows サーバーまたは Linux サーバーにアプリケーションを インストールします。
2. SafeKit ソフトウェアを両方のサーバーにインストールします。
3. SafeKitアプリケーション モジュールを両方のサーバーにインストールします。
4. サーバーとクラスターの新しい名前を構成します。

ソフトウェア クラスターの種類

SafeKit では、基本的なソフトウェア クラスターとしてミラー クラスターとファーム クラスターの 2 種類が提供されます。同じソフトウェア クラスターに複数のSafeKitアプリケーション モジュールを展開できます。したがって、ファーム/ミラーの混合、アクティブ/アクティブ、N-1 など、高度なクラスタリング アーキテクチャを実装できます。

ソフトウェア クラスター内でのアプリケーション統合

SafeKitアプリケーション モジュールには次のものが含まれます。

1. 次のものを含むメイン構成ファイル（userconfig.xml ）
   - サーバーの名前または物理 IP アドレス
   - クラスターの名前または仮想 IP アドレス
   - リアルタイムで複製するためのファイル ディレクトリ (ミラー モジュールの場合)
   - ネットワーク ロード バランシング条件 (ファーム モジュールの場合)
   - ソフトウェアおよびハードウェア障害検出機能の構成

2. アプリケーションの停止および開始スクリプト

SafeKit でのミラー ソフトウェア クラスター

リアルタイム ファイル レプリケーションおよびアプリケーション フェイルオーバーを備えた高可用性クラスター

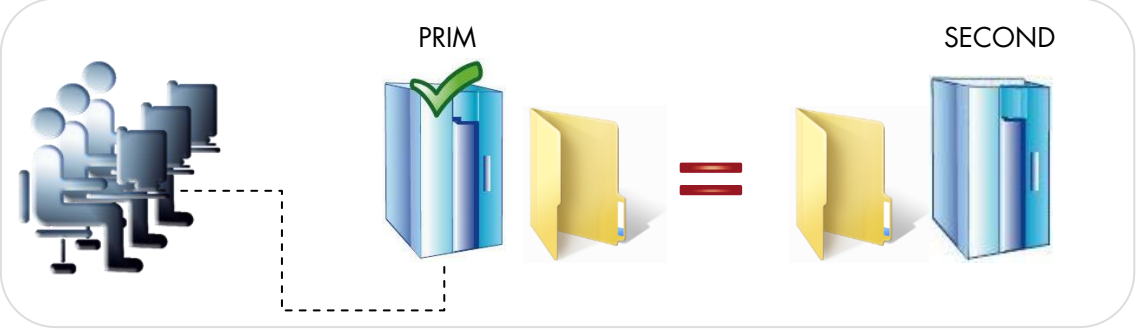
ミラー ソフトウェア クラスターは、プライマリバックアップの高可用性ソリューションです。アプリケーションは、プライマリ サーバー上で実行し、プライマリ サーバーで障害が発生した場合はセカンダリ サーバー上で自動的に再起動されます。

ミラー クラスターの構成では、ファイル レプリケーションを使用しても、使用しなくてもかまいません。ファイル レプリケーション機能を備えたこのアーキテクチャは、重要なデータを含むバックエンド アプリケーションに高可用性を提供して障害から保護するのに、特に適しています。

Microsoft SQL Server.Safe、MySQL.Safe、Oracle.Safe は、「ミラー」タイプのアプリケーション モジュールの例です。汎用モジュールの Mirror.Safe を基にして、アプリケーション用に独自のミラー モジュールを作成できます。

ミラー ソフトウェア クラスターは次のように動作します。

ステップ 1. 通常動作



レプリケーションの場合、SafeKit にはファイル ディレクトリの名前だけが構成されます。2 台のサーバーのディスク編成に対して前提条件はありません。複製対象のディレクトリがシステム ディスクに存在してもかまいません。
サーバー 1 (PRIM) がアプリケーションを実行します。
SafeKit は、アプリケーションによって開かれたファイルを複製します。アプリケーションによってファイルに対して行われた変更だけが、ネットワーク経由でリアルタイムに複製されるので、トラフィックは限られます。

ステップ 2．フェイルオーバー

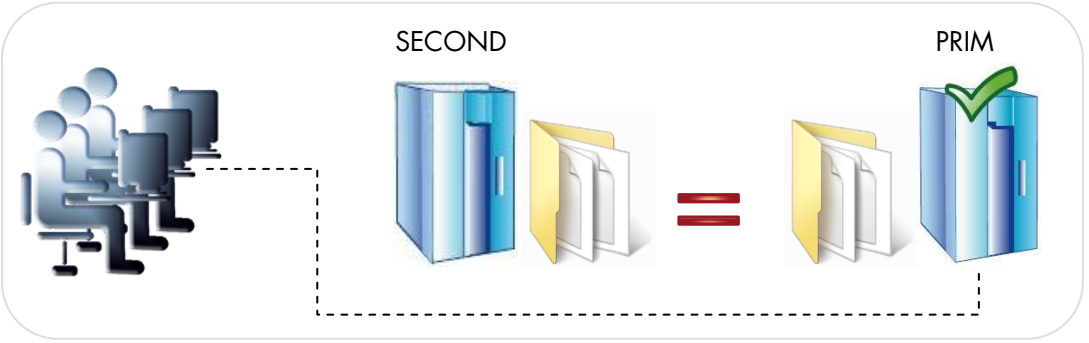
サーバー 1 で障害が発生すると、サーバー 2 が引き継ぎます。SafeKit は、クラスターの仮想 IP アドレスを切り替え、サーバー 2 でアプリケーションを自動的に再起動します。アプリケーションは、SafeKit がサーバー 1 とサーバー 2 の間で行った同期レプリケーションにより、サーバー 2 で最新のファイルを検出します。アプリケーションは、サーバー 1 には複製されなくなったファイルをローカルに変更することによって、サーバー 2 上で実行を続けます。

切り替え時間は、障害検出時間 (デフォルトでは 30 秒に設定されています) にアプリケーション起動時間を加えたものです。ディスク レプリケーション ソリューションとは異なり、ファイル システムの再マウントとリカバリー手順の実行に伴う遅延はありません。

ステップ 3．フェイルバックと再統合

フェイルバックには、サーバー 1 の障害の原因になった問題の修正と、その後のサーバー 1 の再起動が含まれます。SafeKit は、ファイルを自動的に再同期し、サーバー 1 が停止している間にサーバー 2 で変更されたファイルだけを更新します。

この再統合はアプリケーションを中断することなく行われ、アプリケーションはサーバー 2 上で実行し続けることができます。この機能は SafeKit が他のソリューションと大きく異なる点であり、他のソリューションではサーバー 1 を再同期するためにサーバー 2 のアプリケーションを停止する必要があります。



再統合の後、ファイルは再びステップ 1 と同様にミラー モードになります。システムは高可用性モードに戻り、アプリケーションはサーバー 2 で実行し、SafeKit はファイルの更新をバックアップ サーバー 1 に複製します。

管理者は、アプリケーションをサーバー 1 で実行する必要がある場合は、適当なタイミングで手動により、または構成によって自動的に、「swap」コマンドを実行できます。

同期レプリケーションと非同期レプリケーションの比較

SafeKit のミラー ソリューションによって提供される同期レプリケーションと、他のファイル レプリケーション ソリューションによって提供される従来の非同期レプリケーションの間には、大きな違いがあります。

同期レプリケーションでは、プライマリ サーバー上でアプリケーションまたはファイル キャッシュ システムによって複製対象ファイル内で IO が実行されると、SafeKit は、ローカル ディスクおよびセカンダリ サーバーからの IO 確認応答を待った後、アプリケーションまたはファイル システム キャッシュに IO 確認応答を送信します。このメカニズムは、トランザクション アプリケーションのリカバリーに不可欠です。

同期データ レプリケーションを実装するには、サーバー間の LAN の帯域幅が必要であり、地理的に離れた場所にある 2 つのコンピュータ ルームでは拡張 LAN によって提供される場合があります。

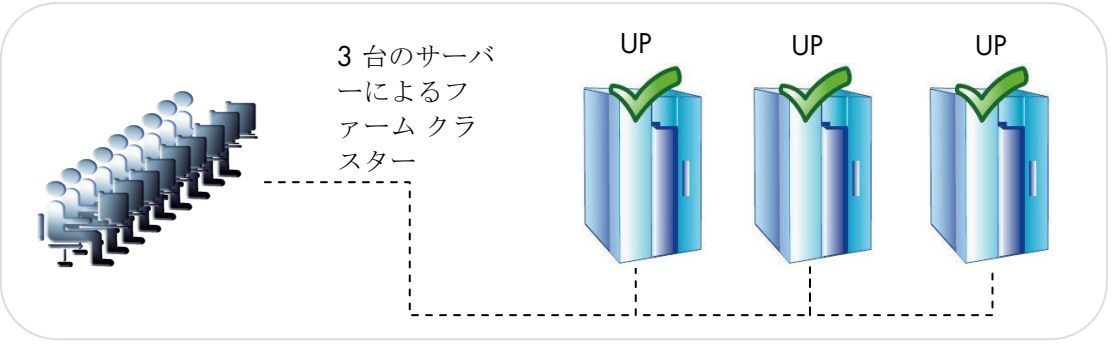
他のソリューションによって実装される非同期レプリケーションの場合は、IO はプライマリ サーバーのキューに入れられますが、プライマリ サーバーはセカンダリ サーバーの IO 確認応答を待ちません。したがって、最初のサーバーで障害が発生した場合、ネットワーク経由で第 2 のサーバーにコピーされる時間がなかったデータはすべて失われます。

特に、トランザクション アプリケーションでは、障害時にはコミットされたデータが失われます。非同期レプリケーションは、低速の WAN を使用してリモートでデータをバックアップするデータ レプリケーションに使用できます。

SafeKit は非同期ソリューションを提供しますが、プライマリ マシンではなくセカンダリ マシンに非同期性を実装します。このソリューションでは、SafeKit は常に、2 台のマシンの確認応答を待ってから、アプリケーションまたはシステム キャッシュに確認応答を送信します。ただし、セカンダリでは、非同期または同期の 2 つのオプションがあります。非同期の場合、セカンダリは IO を受信するとプライマリに確認応答を送信し、その後でディスクに書き込みます。同期の場合は、セカンダリはディスクに IO を書き込んだ後、プライマリに確認応答を送信します。2 台のサーバーで同時に停電が発生し、先のプライマリ サーバーを再起動できず、セカンダリで再起動する必要があるような状況を考慮する場合は、同期モードが必要です。

SafeKit でのファーム ソフトウェアクラスター

ネットワーク ロード バランシングとアプリケーション フェイルオーバーによる拡張性と高可用性



ファーム ソフトウェア クラスターでは、ネットワーク トラフィックの透過的な分配によるネットワーク ロード バランシングと、ソフトウェアおよびハードウェアのフェイルオーバーの両方が提供されます。このアーキテクチャは、増加するシステム負荷に対し容易に高可用性ソリューションを提供します。

同じアプリケーションが各サーバーで実行しており、負荷はファームの異なるサーバー間でネットワーク アクティビティを分散させることによって平均化されます。

ファーム クラスターは、Web サービスのようなフロントエンド アプリケーションに適しています。

Apache_farm.Safe や Microsoft IIS_farm.Safe はファーム アプリケーション モジュールの例です。汎用モジュールの Farm.safe を基にして、アプリケーション用に独自のファーム モジュールを作成できます。

ネットワーク ロード バランシングでの仮想 IP アドレスの原理

仮想 IP アドレスは、ファームの各サーバーにローカルに構成されます。このアドレスに対する入力トラフィックは、各サーバーのカーネル内にあるフィルタによって下位レベルでサーバー間に分割されます。

フィルタの内部でのロード バランシング アルゴリズムは、クライアント パケットのアイデンティティに基づきます (クライアント IP アドレス、クライアント TCP ポート)。クライアント パケット入力のアイデンティティに応じて、サーバー ファーム内の 1 つのフィルタ インスタンスはそのパケットを上位ネットワーク層に転送し、他のサーバーの他のフィルタ インスタンスはそのパケットをドロップします。あるサーバーのフィルタによってパケットが受け付けられると、そのサーバーの CPU とメモリだけが、クライアントの要求に応答するアプリケーションによって使用されます。出力メッセージは、アプリケーション サーバーからクライアントに直接送信されます。

1 台のサーバーで障害が発生すると、SafeKit のメンバシップ プロトコルは、ファームのフィルタを再構成し、残っている使用可能なサーバーでトラフィックを再度平均化します。

ステートフルまたはステートレスな Web サービスのロード バランシング

ステートフルなサーバーでは、セッション ・アフィニティが存在します。同じクライアントが複数の HTTP/TCP セッションで同じサーバーに接続し、サーバー上のコンテキストを取得する必要があります。この場合、SafeKit のロード バランシング規則はクライアント IP アドレスに構成されます。したがって、同じクライアントが複数の TCP セッションで同じサーバーに常に接続されます。また、異なるクライアントはファーム内の異なるサーバーに分散されます。セッション・アフィニティが存在するときは、この構成が使用されます。

ステートレスなサーバーでは、セッション・アフィニティが存在しません。同じクライアントが、複数の HTTP/TCP セッションでファーム内の異なるサーバーに接続できます。セッションが変わるときにコンテキストがサーバー上にローカルに格納されることはありません。この場合、SafeKit のロード バランシング規則は TCP クライアント セッション アイデンティティに構成されます。この構成はサーバー間にセッションを分散させるためには最善ですが、セッション・アフィニティのない TCP サービスが必要です。

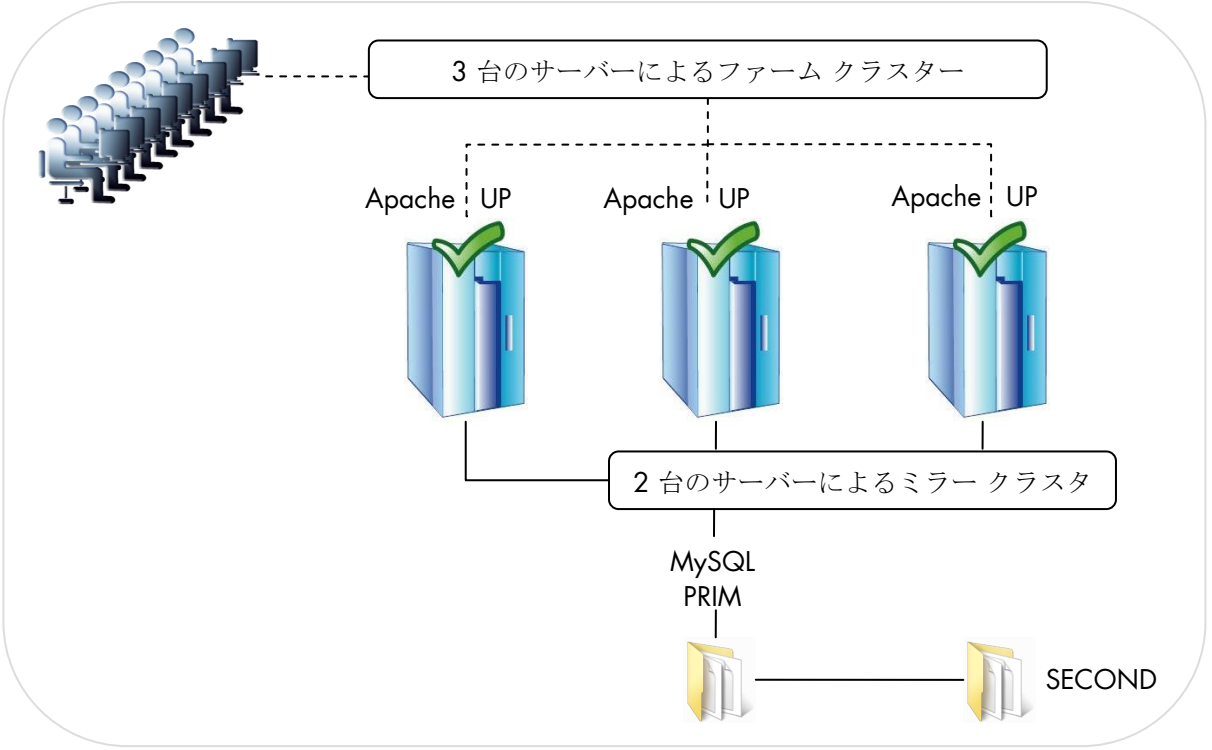
UDP を利用するサービスやファイアウォール等のためには、他のロード バランシング アルゴリズムが提供されます。

SafeKit でのファーム/ミラー混合ソフトウェア クラスター

ネットワーク ロード バランシング、ファイル レプリケーション、アプリケーション フェイルオーバー

ファームと**ミラー**の両方のアプリケーション モジュールを、同じサーバー クラスターに混在させることができます。

このオプションを使用すると、Apache_farm.Safe (ロード バランシングとフェイルオーバーのファーム アーキテクチャ) や MySQL.Safe (レプリケーションとフェイルオーバーのミラー アーキテクチャ) のような複数階層のアプリケーション アーキテクチャを同じアプリケーション サーバーに実装できます。

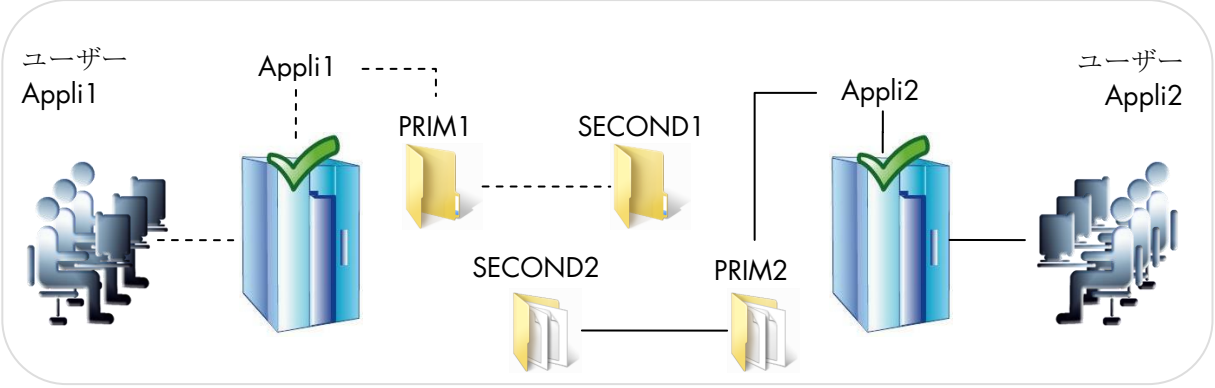


結果として、ロード バランシング、ファイル レプリケーション、フェイルオーバーが、同じサーバー上で一貫して管理されます。市販されている製品でこのような混合クラスターを提供するのは SafeKit だけです。

SafeKit でのアクティブ/アクティブ ソフトウェア クラスター

クロス レプリケーションと相互テイクオーバー

アクティブ/アクティブ クラスターでは、2 台のサーバーと 2 個のミラー アプリケーション モジュールが相互テイクオーバーに存在します (Appli1.Safe と Appli2.Safe)。各アプリケーション サーバーは他のサーバーのバックアップです。



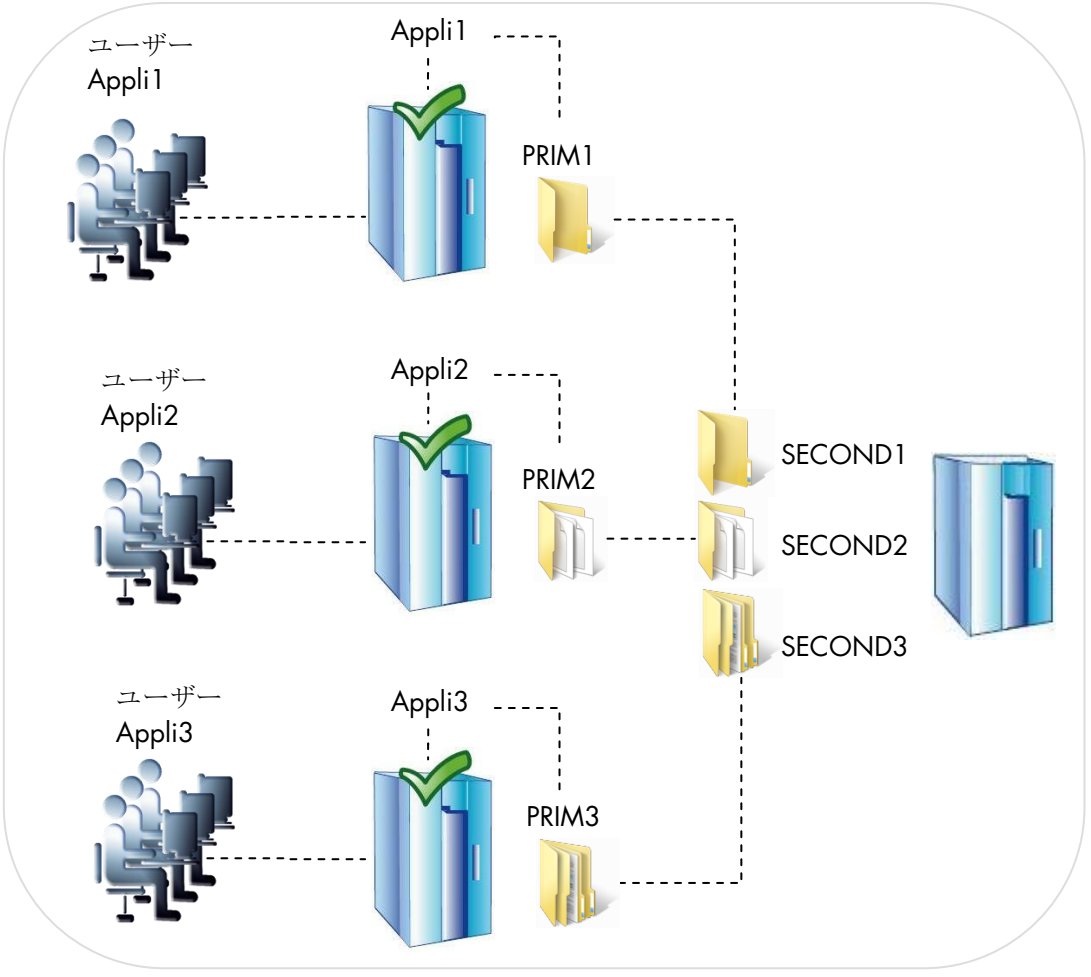
1 台のアプリケーション サーバーで障害が発生した場合、両方のアプリケーションが同じ物理サーバーでアクティブになります。障害が発生したサーバーが再起動した後、アプリケーションは再びデフォルトのプライマリ サーバーで実行するようになります。

相互テイクオーバー クラスターは、ほとんどの時間をプライマリ サーバーの障害を待ってアイドル状態で過ごすバックアップ サーバーに投資する必要がないので、2 つの異なるミラー クラスターより経済的なソリューションです。
※障害が発生している間、残っているサーバーは両方のアプリケーションの作業負荷をまとめて処理できる必要があることに注意してください。

SafeKit でのソフトウェア クラスター N-to-1

N 台のサーバーから 1 台へのレプリケーションとアプリケーション フェイルオーバー

N-to-1 クラスターでは、N 個のミラー アプリケーション モジュールが、N 台のプライマリ サーバーと 1 台のバックアップ サーバーにインストールされます。



N 台のアクティブなサーバーの 1 つで障害が発生した場合、単一のバックアップ サーバーで障害が発生したサーバーのモジュールが再起動されます。問題が解決して障害が発生したサーバーが再起動すると、アプリケーションは元のサーバーに戻されます。

障害が発生した場合、**アクティブ/アクティブ クラスター**とは異なり、バックアップ サーバーは 2 倍の作業負荷を処理する必要はありません。これは、発生する障害が一度に

1 つだけと想定した場合です。このソリューションは同時に複数のプライマリ サーバー障害の発生をサポートできますが、その場合は、単一のバックアップ サーバーで障害が発生したすべてのサーバーの作業負荷をまとめて処理する必要があります。

For more information: www.evidian.com/ja/

atos.net

39 J2 03IU 07