Oracle Tuxedo による高可用性の実現

Oracle ホワイト・ペーパー
2008 年10 月

ORACLE

Oracle Tuxedo による高可用性の実現

Oracle Corporation 発行「Achieving High Availability with Oracle Tuxedo」の翻訳版です。

はじめに

オンライン・トランザクション処理（OLTP）アプリケーションは、世界の大半のビジネスに関するコア・プロセスをサポートしています。このようなシステムは大容量でミッション・クリティカルであるため、障害が発生するとビジネス存続への影響は甚大なものとなります。通常、OLTP アプリケーションに伴う要件には、次のようなものがあります。

- 何千人もの同時ユーザーのサポート機能
- 大容量データの処理機能
- 高いトランザクション・スループット
- 予測可能な短い応答時間
- 高いデータ整合性とセキュリティ
- 同時データベース・アクセス
- 高いアプリケーション可用性（たいていは 24 時間 365 日）

たいていの OLTP システムでは、コンポーネントの停止やアプリケーションのアップグレードの必要性が生じた場合でも、24 時間 365 日稼働することが必須要件となっています。停止時間の発生は、金銭的損失、顧客満足度の低下、ビジネス損失につながるため、コンピューティング・システムを業務のバックボーンとして使用している企業にとっては深刻な問題です。

Oracle Tuxedo には、顧客が高可用性アプリケーション・サービスを配置して実行する際に使用できる多数の組込み機能が搭載されています。また、Oracle RAC と、Veritas Cluster Server や HP の MCServiceGuard といったサード・パーティのクラスタリング・ソリューションを組み合わせて使用すれば、配置されたアプリケーションの可用性と堅牢性をよりいっそう向上させることが可能です。

システム可用性

システム可用性とは、システムの使用可能時間をパーセントで表したものです。可用性は、システムの障害率とリカバリ時間という 2 つの要素によって低下します。これらの要素は、次の 2 つの測定単位によって定量化されます。1 つは平均故障間隔（MTBF）で、障害発生までのシステムの平均稼働時間を示す、システム信頼性の測定基準です。もう 1 つは平均修復（またはリカバリ）時間（MTTR）で、障害発生後のシステムの修復にかかる時間、またはシステムが自動的にリカバリされる時間を示します。そうすると、可用性は次のように定義できます。

可用性 = MTBF/（MTBF + MTTR）

したがって、信頼性が向上してリカバリ時間が短縮されると、可用性は向上します。

システム停止の原因

「顧客はミッション・クリティカルなビジネス・システムにほぼ 100%の可用性を要求しています。こうした品質の期待に応えるには、成熟した IT 運用プロセスが必要です。」

Gartner Vice President、Donna Scott

一般的なコンピュータ・システムとアプリケーションのさまざまな停止は、計画停止と計画外停止に分けることができます。計画停止は、ハードウェアやソフトウェアのアップグレード、バックアップ、およびそのほかのメンテナンスでシステムをオフラインにする必要が生じた場合に発生します。最近のシステムは、冗長でホット・プラガブルなハードウェア・コンポーネントやオンラインでの OS アップデート、アプリケーションを実行したままアップグレードできる機能を組み合わせることで、計画停止が完全に不要になるように設計されています。Oracle Tuxedo に対して透過的なアップグレードが、アプリケーションに影響を及ぼすことはありません。透過的でないアップグレードでは、複数のノード間で実行中のサービスをシフトして、停止の発生しないローリング・アップグレードを実施できます。

計画外停止は、運用環境における何らかの障害に起因するものであり、次の種類に分類できます。

- 環境障害（電力、通信、エアコンなど）
- 運用障害（構成時や処理時の人的エラーなど）
- ハードウェア障害（プロセッサ、メモリ、I/O コントローラ、ネットワーク装置などのあらゆるハードウェア・デバイス）
- ソフトウェア障害（オペレーティング・システム、データベース、トランザクション・モニター、アプリケーション）

近年では、ハードウェアとソフトウェアの堅牢性の向上により、次の表に示すように、運用、管理、メンテナンスが停止時間発生のおもな原因となっています。

20%	**ハードウェア**
ディスク、ネットワーク、メモリ、プロセッサ	
30%	**ソフトウェア**
アプリケーション、ミドルウェア、データベース	
50%	**運用、管理、メンテナンス**
• 定期的なメンテナンス • 不注意による誤った構成、誤ったホストの停止、誤ったケーブルの切断など • 意図的なハッキングや DoS 攻撃など	

システム可用性の最大化

高可用性（HA）システムは、こうした障害からのリカバリを自動化することにより、コンポーネント障害の影響からユーザーを保護し、アプリケーション環境のMTTR を短縮することを目的としています。この目的の達成方法を理解するには、まず可用性のさまざまなレベルを定義する必要があります。

*標準可用性*システムとは、障害処理リカバリを実現するハードウェアの冗長性とソフトウェアの拡張機能が備わっていない汎用コンピュータを指します。このシステムでは、通常運用を再開する前に、ユーザー介入により手動で障害を特定して修復し、システムを再起動する必要があります。

ビジネスは 24 時間 365 日オンラインである必要があります。重要なアプリケーションやサーバー、データが使用できなくなると、ビジネス全体が危険にさらされます。収益の損失、顧客満足度の低下、違約金、否定的な報道が、組織の評判に継続的な影響を及ぼします。Oracle Tuxedo は、もっとも一般的な障害発生要因である人的エラーをはじめとした、計画停止時間および計画外停止時間のあらゆる一般的な発生要因からビジネスを保護します。

さらに、オラクルの Maximum Availability Architecture フレームワークにより、オラクルの実績ある高可用性テクノロジーを使用してベスト・プラクティスを実装する明確かつ簡単なガイダンスを提供します。

*高可用性*システムでは、ハードウェアとソフトウェアの冗長性とともに状態監視機能を使用します。これは多くの場合、疎結合のコンピュータ・クラスタを使用しておこない、ノード・レベルで冗長性を提供します。クラスタは、自動障害検出および修正手順を提供するソフトウェア（クラスタ・マネージャ）で管理されます。このクラスタ・システムでは、障害の特定、影響を受けたコンピュータのバイパス手順の実行、およびシステム管理者への通知を手動でおこなう必要はありません。たいていの種類の障害に自動的に対応し、中断を最小限に抑えてサービスへのアクセスをリストアします。クライアント/サーバー・アーキテクチャには、次のような 2 つの異なる高可用性モデルがあります。

- **レプリケートされたサービス・モデル**
  このモデルでは、複数のサーバーで分散アプリケーションと分散データベースを使用します。アプリケーション・サービスを複数のサーバーで使用できるため、特定のサーバーに障害が発生しても、同一サービスを提供する別のサーバーで引き続きリクエストを処理できます。データは一部または全部のサーバーにレプリケートされるか、それ以外の場合でも複数のシステムから（たとえば、Oracle RAC のような並列データベース・システム経由で）参照できます。そのため、サーバー障害の発生時に、データとアプリケーションに別のノードからアクセスできます。
- **フェイルオーバー・モデル**
  このモデルでは、ハードウェアの二重構成を使用して、一方のシステムをデータ・サービスとアプリケーション・サービスに使用するプライマリ・サーバー、もう一方のシステムをプライマリ・システムの状態を監視するバックアップ・サーバーとして使用します。プライマリ・サーバーの障害がバックアップ・ノードで検出されると、バックアップ・ノードによりプライマリ・サーバーのロールと ID が引き継がれます。

*フォルト・トレラント・*システムは、独自仕様で高額の密結合された二重構成のコンポーネントで構成されています。障害処理機能は、オペレーティング・システムの 1 機能として統合されています。このシステムでは、手動で障害が発生したコンポーネントを特定し、システム障害を回避する手順を実行する必要はありません。障害に対して完全に自動で対応し、まったく中断されることのないサービスを提供します。

Oracle Tuxedo プラットフォームおよびインフラストラクチャ

TP モニターは、本番環境で継続的に実行するサーバー・ベースの OLTP アプリケーションに対して、実行環境を提供するものです。Oracle Tuxedo は、オープン・システム向けの主要な TP モニターで、分散トランザクション処理とメッセージング・ミドルウェアを提供します。また、COBOL、C、C++アプリケーションをホストするアプリケーション・サーバーの機能を、同期型、非同期型、対話型をサポートする高速で信頼性の高いメッセージングやイベント・ベースのパブリッシュ・サブスクライブのメッセージング・モデルに向けたサービス・バス機能と統合します。Oracle Tuxedo には、メッセージ・ベースの通信を介して接続する分散アプリケーション・コンポーネントを構築できる高度な API が備わっています。コンポーネントは、Oracle Tuxedo のコア・サービスで実装される管理対象サーバー環境（コンテナ）で実行されます。このサービスでは、次のようなトランザクションおよびアプリケーションの高度な管理機能と、分散システムの総合管理機能が実装されます。

- データ整合性とアトミックな更新を実現する透過的な 2 フェーズ・コミット
- トランザクションとエラーのロギング
- 一元管理または分散トランザクションを管理する管理フレームワーク
- 高速キャッシュ・アクセス方法、自動サーバー作成、バッファ管理、およびルーティング機能によるトランザクション・スループットの最適化

Oracle Tuxedo に搭載されている豊富な組込みの高可用性機能は、次のとおりです。

- 組込みの冗長性、レプリケーション、分散、およびクラスタリング機能
- バディ・システムとハートビート・メカニズムを使用したサーバーおよびクライアント・プロセスの監視と障害検出
- 障害リカバリ機能：再起動、再ルーティング、およびフェイルオーバー
- データ依存ルーティング
- ACID プロパティを使用した分散トランザクション

さらに、Oracle Tuxedo のサービス仮想化機能により、実装、配置、レプリケーション、フェイルオーバー、およびリカバリの透過性がサポートされ、動的ルーティング、ロードバランシング、フェイルオーバー/フェイルバックが実現します。Oracle Tuxedo アプリケーションは、アプリケーション問題の分離やセキュリティ・ゾーンなど、考慮事項での必要に応じて、単一マシン・ドメイン、複数ノードのドメインやクラスタ、複数ドメインに配置できます。

Oracle Tuxedo クラスタリング - MP ドメイン

Oracle Tuxedo には組込みのクラスタリング機能が備わっているため、追加でソフトウェアやハードウェアのクラスタリングをおこなうことなく、複数のノードに Oracle Tuxedo アプリケーションを配置できます。Oracle Tuxedo クラスタ（別名 MP ドメイン）では、マルチプラットフォーム・アーキテクチャでサービスを透過的に共有してロードバランシングできます。この環境ではシングル・ポイント障害は発生せず、動的管理により、マシンやサーバー（プロセス）、サービスを追

加または削除して継続運用を実現できます。また、それらの機能により、エンドユーザーのサービス可用性に影響を与えることなく、Oracle Tuxedo プラットフォームとアプリケーションのローリング・アップグレードを実行できます。

Oracle Tuxedo ドメインを使用して、先ほど説明した 2 つの HA モデルのいずれかを実現できます。

- **フェイルオーバー・モデル**：アクティブ/パッシブ構成を使用

これは一般的に使用されるモデルであり、ミラー化構成が必要なため、ハードウェアおよびソフトウェア要件は 2 倍になりますが、対応できるワークロード総容量は半分になります。データ・レプリケーション（ファイルとデータベース）は、Oracle Data Guard やサード・パーティの同様のソリューションなどで実施する必要があります。引継ぎノードですでにサービスが実行されるアクティブ/アクティブ構成とは異なり、パッシブ・システムでは、起動プロセスを実行して障害が発生しているノードを引き継ぐ必要があるため、定期的にテストを実施しないと信頼性の問題が生じる可能性があります。

- **レプリケートされたサービス・モデル**：アクティブ/アクティブ構成を使用

この構成では、Oracle Tuxedo で 1 台または複数のマシンで構成される複数ドメインをサポートできます。全リソースが使用中となるため、障害が発生しているノードやドメインのワークロードを別のノードやドメインで確実に引き継ぐことができますが、パフォーマンスが低下する可能性があります。また、アクティブ/アクティブ構成には、サービスのレプリケーションやデータ依存ルーティングによりシステムを拡張できるというもう 1 つの利点があります。こうして高可用性とスケーラビリティを組み合わせることができるため、アクティブ/アクティブ構成は大規模な OLTP ワークロードに最適です。

用語

ノード - OS のコピーを実行し、ネットワーク接続とディスク・ストレージをもつ 1 基または複数の CPU と共有メモリ。

クラスタまたはドメイン - ディスク・ストレージとネットワークを共有し、制御情報とハートビートを伝達するプライベート・インターコネクトで接続されている 1 つまたは複数のノード。アプリケーションの視点では、Oracle Tuxedo ドメインは、1 つの Oracle Tuxedo 構成ファイルで単一のユニットとして管理される、一連の Oracle Tuxedo システム、クライアント、およびサーバー・プロセスを指します。Oracle Tuxedo ドメインは、多数のシステム・プロセス、1 つまたは複数のアプリケーション・クライアント・プロセス、アプリケーション・サーバー・プロセス、ネットワーク経由で接続された 1 台または複数のコンピュータ・マシンにより構成されています。Oracle Tuxedo ドメインでは、ATMI サービスと CORBA オブジェクトのいずれかまたは両方が提供されます。

論理ホスト - 物理ホスト（ノード）上に存在するよう定義された仮想ノードで、別の物理ホストへの移行も可能です。1 つの物理ホスト上に複数の論理ホストを定義できます。

論理マシン - 仮想ノードを指す Oracle Tuxedo 用語。Oracle Tuxedo 構成では、仮想ノードは論理マシン ID（LMID）で識別されます。

マスター/スレーブ・ノード - Oracle Tuxedo システムでは、アプリケーションのマ

ミッション・クリティカルなアプリケーションに実証されたスケーラビリティ、信頼性、およびパフォーマンスを提供

80% の市場シェアをもつ Oracle Tuxedo は、ミッション・クリティカルなアプリケーションに向けた、オープン・システムの分散トランザクション処理プラットフォームです。世界の多数の大手企業にバックボーンを提供し、支払いネットワーク、電子送金、ATM、航空、通信システムなどの大規模トランザクション・システムの一部を実行しています。

XTP クラスのアプリケーションを構築中か、既存のメインフレーム・システムの移行を検討中のお客様は、トランザクションを処理および統合する際に、その実績から Oracle Tuxedo を選択しています。C、C++、COBOL アプリケーション向けのクラス最高のアプリケーション・サーバーです。

スターとして構成されるコンピュータを"マスター・ノード"と呼びます。マスター・ノードは、構成のマスター・コピーが格納されている DBBL を実行するコンピュータです。もう 1 つのノードは、"スレーブ・ノード"と呼ばれます。スレーブ・ノードの 1 つを、マスター・ノードを移行して新しいマスター・ノードとして機能させることができる、"バックアップ・ノード"として構成できます。

バックアップ・マスター - マスター・ロールを継承できる Oracle Tuxedo の論理マシン。Oracle Tuxedo のマスター論理マシンに障害が発生した場合に、マスター・ロールを引き継ぎます。

プライマリ/バックアップ・ノード - Oracle Tuxedo の各サーバー・グループに、2 つのノード（プライマリ・ノード 1 つとバックアップ・ノード 1 つ）を構成できます。サーバー・グループはまずプライマリ・ノードで起動し、バックアップ・ノードに手動で移行できます。

対象範囲

このホワイト・ペーパーでは、Oracle Tuxedo の高可用性機能の概要について説明します。ここでは、データセンター内のハードウェアやソフトウェアに障害が発生した場合のシステム可用性に重点を置いています。環境障害や運用障害、リモート・クライアント・マシンの障害は対象としていません。

高可用性構成の Oracle Tuxedo

Oracle Tuxedo は、業界第 1 位の分散トランザクション処理向けプラットフォームです。C、C++、COBOL で記述されたソフトウェアの分散オープン・システムでメインフレーム・クラスのスケーラビリティとパフォーマンスを実現する Oracle Tuxedo は、オープン・システムのプラットフォームとグリッド・インフラストラクチャに新しい超高速トランザクション処理（XTP）アプリケーションを構築し、メインフレーム・アプリケーションを再ホストするのに最適なプラットフォームです。また、費用効率の高い信頼性と最大数十万トランザクション/秒のきわめて高いスケーラビリティを実現し、既存の IT 資産を最新アーキテクチャ（SOA など）の一部として使用することにより耐用年数を延長して、投資の保護を実現します。Oracle Tuxedo は、Oracle Fusion Middleware の戦略的なトランザクション処理製品です。

Oracle Tuxedo は、オープン・システム向けのトランザクション処理アプリケーション・プラットフォームの第 1 位であり、長い間、オープン OLTP テクノロジーやアプリケーションのパフォーマンス、移植性、スケーラビリティ、および可用性の標準であり続けてきました。この Oracle Tuxedo には、ミッション・クリティカルな分散アプリケーションの厳しいニーズを満たす豊富な機能が備わっています。とくに、従来の TP モニターに比べて強力な高可用性機能が搭載されています。この強力な機能の鍵となっているのは、Oracle Tuxedo、ランタイム・モニター、診断および修復機能で使用されるシステム・コンポーネントの主要パラメータを提供する一元管理システム構成です。この構成では、全コンポーネントの関連特性（サーバー・プロセス、サービスのインスタンス化、クライアント・プロセス、プロセッサ・ノード、ネットワーク接続、リソース、とくに再構成オプション）が取得されます。

ここからは、高可用性の実現に寄与するOracle Tuxedoの機能について詳しく説明します。Oracle Tuxedoの詳細情報や特定の機能の詳細は、ここでは説明されていません。http://www.oracle.com/lang/jp/products/middleware/tuxedo/index.htmlを参照し

てください。

ネットワークにアプリケーションを分散する理由

分散アプリケーションには、いくつかの重要な利点があります。初期のビジネス・アプリケーションは、1 台の大型メインフレーム・コンピュータで実行するという目的の下開発されていました。計算はすべて 1 台のマシンで実行されていたため、ひとたび障害が発生するとシステム全体がダウンするおそれがありました。分散アプリケーションが普及するにつれ、こうしたシステム障害が発生するおそれは少なくなっています。

アプリケーションを分散させるもう 1 つの利点は、アプリケーションの一部を論理的にグループ化して、もっとも効果的な場所に論理グループとして配置できる点です。たとえば、サーバーのグループを作成することにより、大規模アプリケーションを管理可能なサイズのビジネス別コンポーネントに分割して、最適な場所に配置できます。

分散アプリケーションでは、次のことが可能です。

- **データ依存のパーティション化の実行**
- **複数のリソース管理**
- **クライアント/サーバー・モデルの拡大**
- **BEA Tuxedo システム・サービスへの透過的なアクセス**
- **複数のサーバー・グループの構築**
- **複数のコンピュータで同時に 1 つのアプリケーションを処理することによる、スループットの向上と応答時間の短縮**
- **リソースのレプリケートによる可用性の向上**

管理インフラストラクチャ

Oracle Tuxedo は、分散アプリケーションを管理していく上での深刻な問題に対して、ソリューションを構築します。Oracle Tuxedo の管理インフラストラクチャには、管理情報ベース（MIB）を中心に構築されたマネージャ/エージェント・モデルが導入されています。

Oracle Tuxedo の MIB を使用すれば、アプリケーション管理者が 1 つの一元管理アプリケーションを構成する際に、アプリケーションを構成するハードウェア、ソフトウェア、ネットワーク・リソースを定義できます。アプリケーション設計者は、サーバーとサービスの実行場所や、プロセッサに障害が発生した場合のそれらの移行先を指定できます。また、アプリケーション・サーバーのスケジュール情報、プロセス・リカバリ基準、タイムアウト時間などのさまざまな特性を割り当てることができます。

MIB 上に実装されている管理インタフェースには、総合的なコマンドライン・ツール、プログラム・インタフェース（スプリクトを含む）、および Oracle Tuxedo を管理対象アプリケーションとして大規模管理環境内に統合する SNMP エージェントが搭載されています。また、GUI ベースの管理アプリケーションで MIB を活用し、Oracle Tuxedo 環境でシングル・ポイントの高度な制御をおこなうことができます。

MIB では、全システム・パラメータのポーリング（および許可部分の変更）を実行できるのに加え、あらゆるシステム・イベントをサポートしています。これらのイベントは、Oracle Tuxedo 環境内で重大な状態変更が発生するたびにポストされます。

また、Oracle Tuxedo には、アプリケーション可用性のランタイム・サポートを提供する、次のような充実した内部メカニズムが備わっています。

- BBL - Bulletin Board Liaison の略で、ノードの全プロセス（アプリケーションおよび管理プロセス）の監視を担当するノード・モニターです。
- DBBL - Distinguished BBL の略で、各ノードの BBL の監視を担当するマスター・モニターです。ネットワーク・アプリケーションでは、バックアップ・ノードを DBBL に指定できます。
- BRIDGE - ネットワーク・アプリケーションのノード内通信を提供するサーバーです。
- TMS - Transaction Management Server の略で、分散トランザクション処理を使用する場合の、特定の DBMS（または MQ キューなどのそのほかのリソース）専用のトランザクション管理サーバーです。

そのほか、各種クライアント・ハンドラ、イベント・ブローカ、キュー・マネージャなどのオプションのシステム・サーバーも同様に使用できます。これらのサーバーは、システム起動時に一元管理構成に従って起動します。

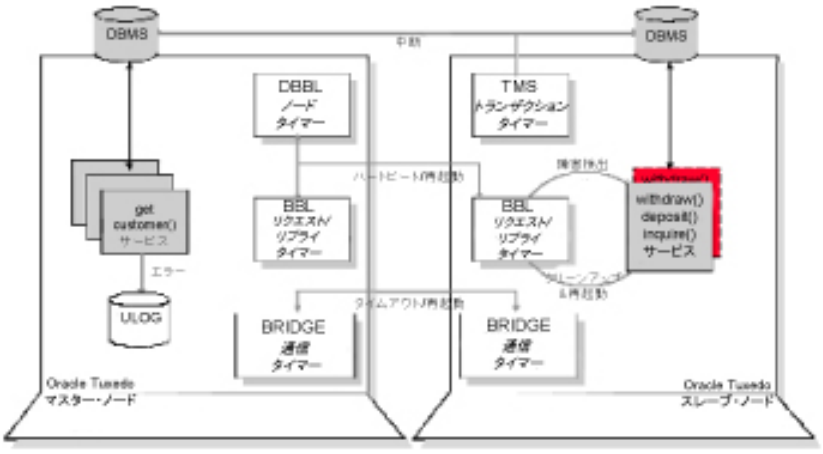

図 1. 2 ノード構成の Oracle Tuxedo クラスタ（キー・サーバーあり）

障害検出および自動リカバリ

ランタイム機能

Oracle Tuxedo のランタイム管理では、ソフトウェア障害を自動検出して修正します。おもな機能について、図 1 で説明しています。ノード、ネットワーク接続、アプリケーション・サーバー、クライアントに加え、Oracle Tuxedo の管理サーバー自体（DBBL、BBL、BRIDGE を含む）もすべて監視されます。また、オペレーターの介入を必要としない障害修正機能が常時実行されます。アプリケーションがサブスクライブするシステム・イベントをポストすることによってエラーを報告し、障害検出範囲を拡大してアプリケーション駆動型のリカバリを実現できます。また、エラーはすべて Oracle Tuxedo エラー・ログ（ULOG）に記録され、アプリケーションからアクセスできます。ログ・ファイルはマシンごとに維持されますが、統合して Oracle Manager Log Central で監視することもできます。

ノード・ステータスの確認

DBBL は、各 BBL からの定期的なハートビートの受信を予期します。ハートビートが発生しない場合は、マスターによって問題のあるノードで BBL の再起動が試行されます。BBL が再起動されない場合、そのノードは"partitioned"（使用不可）とマークされます。通信でタイムアウトが発生した場合や、通信を再構築できない場合は、BRIDGE サーバーでもノードのパーティション化がマークされます（詳しくは、ネットワーク接続の項を参照してください）。障害の永続性が不明の場合は、パーティション化されたノードはサービスから自動的に削除されません。パーティションの発生は、システム・イベントを通じて報告され、エラー・ログに記録されます。問題が一時的な通信停止である場合は、通信が再確立されると接続が自動的にリストアされます。問題が深刻な場合は、管理者が管理コマンドを使用してそのノードをサービスから削除できます。存続ノードは、1 つのアプリケーションとして動作を継続します。

高可用性および障害管理

分散クライアント/サーバー環境では、多数の独立プロセッサとプロセスの連携によりアプリケーションを実行する必要があります。そのため、多くの誤動作が発生する場合があります。障害が発生しても、Oracle Tuxedo は次の方法でアプリケーションの動作を継続できます。

- **何かが故障しても動作を継続できるレプリケートされたサーバー・グループを提供することで、シングル・ポイント障害を回避します。**
- **障害発生後に、実行中のアプリケーションを障害発生前の状態にリストアします。**

E ビジネス・アプリケーションへの常時接続を確保することが、Oracle Tuxedo のおもな機能です。システム・コンポーネントのアプリケーション、トランザクション、ネットワーク、ハードウェアの障害を常時監視します。障害発生時には、Oracle Tuxedo がそのコンポーネントをシステムから切断し、必要なリカバリ手順を管理して、メッセージとトランザクションを存続システムに再ルーティングします。これらはすべてエンドユーザーに透過的に実行され、サービスが中断されることはありません。

サーバー・ステータスの確認

BBL は、ノードで実行中の各アプリケーション・サーバー（プロセス）の可用性を定期的に確認します。障害が検出されると、Oracle Tuxedo が未処理トランザクションを中断しますが、アプリケーション・サーバーが自動的に再起動されるように設定できます。失われるのは処理中のサービス・リクエストのみですが、その場合でもリクエスタに正しく通知されるため、適切な操作（再試行など）を実行できます。待機中のサービス・リクエストは、すべて通常どおり処理されます。さらに、Oracle Tuxedo では、アプリケーション・サーバーが再起動されると同時に、アプリケーションで定義されたプロセスが自動的に起動するように設定できます。アプリケーションで失敗したサービス・リクエストを自動的に再送信する必要がある場合は、永続キューを使用できます。

管理のセルフ・チェック

DBBL と BBL は、ステータスを定期的に相互確認し、必要に応じて相互に再起動します。マスター・ノードの損失時には、一元管理構成での指定に従って DBBL を移行する必要があります。この移行は、手動で実行することも、Oracle Enterprise Manager Grid Control やサード・パーティのクラスタリング・ソフトウェアを使用して自動化することもできます。

クライアント・ステータス

BBL は、クライアント・プロセスのステータスを確認し、異常終了を検出するとクリーンアップを実施します。ネイティブ・クライアント（Oracle Tuxedo サーバー・ノードで直接実行されるクライアント）の場合は、プロセス・ステータスによって判断されます。ネイティブ・クライアントのプロセス障害は、応答キューの BBL で維持されるタイムアウトを通じて検出されます。

リモート・クライアントの場合は、異常終了が非アクティブのタイムアウトに基づいて検出されます（クライアントに"keep-alive"プロトコルを使用して、誤ったタイムアウトの発生を回避できます）。クライアント障害が検出されると、進行中の通信はすべて終了し、クライアント・ステータスがクリーンアップされます。ユーザーは、再度ログオンして作業を再開する必要があります。ただし、Oracle Tuxedo のワークステーション・クライアント（WS）や複数のワークステーション・リスナー（WSL）のサーバー・アドレスを指定できるため、WS クライアントでフェイルオーバーをラウンドロビン方式で実行できます。

ネットワーク接続

BRIDGE プロセスでは、一元管理構成で指定された複数のネットワーク・アドレスの階層を通じて、ノード内通信を維持できます。BRIDGE では優先順位がもっとも高い接続を使用しており、ネットワーク停止時には次に優先順位が高い接続に自動的にフェイルオーバーされ、優先順位がもっとも高い接続が使用可能になると、その接続にフェイルバックされます（また、BRIDGE プロセスでは、最初のネットワーク・アドレスがブロックされると、同じ優先順位の 2 番目のネットワーク・アドレスが使用されるため、高負荷ネットワークのスループットが向上します）。これらのネットワーク・アドレスは、複数のネットワーク・インタフェースでサポートできるため、ホスト・アダプタ・カードやコネクタ、ネットワーク・ケーブルに障害が発生しても、かならず通信に支障をきたすということではあり

ません。

ノード間の通信は、BRIDGE サーバーで維持されるタイムアウトによって監視されます。ある接続で障害が検出されると、通信の再確立が自動的に試行されます。全ネットワーク・アドレス経由の通信が失われて再確立できない場合、通信が存続しているノードが引き続きアプリケーションとして動作を継続します。ここで留意すべき点は、アプリケーションをパーティション化して、各パーティションで動作を継続できるということです。

トランザクション

分散トランザクション処理を有効にすると、各グローバル・トランザクションがタイムアウトによって監視されます。しきい値を超えると、トランザクションが中断されます。トランザクションが開始すると、Oracle Tuxedo がアクセスされる全 DBMS（および JMS や MQ キューといったそのほかの XA 対応リソース）を追跡します。トランザクションのプリコミットが完了する前にタイムアウトが発生した場合や、トランザクションがアプリケーションによって明示的に中断された場合は、トランザクション・スコープ内で実行された更新をロールバックするよう全 DBMS に指示されます。関与するノードに障害が発生した場合でも、ロールバックは実行されます。この場合は、ノードが再びオンラインになると、ロールバックが実行されます（データ整合性の説明をあわせてご参照ください）。

アプリケーションに返されるエラー

アプリケーションで検出されたエラーは、システム・イベントを通じて管理者に報告し、エラー・ログに記録できます。このシステム・イベントは、一貫性のあるエラー報告機能を提供し、これを使用して管理者が手動で監視することもできますが、通常はシステム管理ソフトウェアによって自動的に監視します。

障害時のデータ整合性の維持

Oracle Tuxedo は、X/Open 分散トランザクション処理（DTP）標準を実装します。トランザクション管理サーバー（TMS）は、2 フェーズ・コミット・プロトコルを使用して、グローバル・トランザクションがアトミックであることを確認します。グローバル・トランザクションには、複数の異機種ノードで実行される複数の異機種 DBMS が含まれます。レプリケートした冗長な TMS でも、障害復旧時のトランザクションの中断とリカバリを管理できます。

DBMS との通信はすべて、X/Open の標準 XA インタフェースを使用しておこないます。XA は、グローバル・トランザクションに代わって作業の開始と終了をおこなう機能、グローバル・トランザクションのプリコミット、コミット、中断の機能、およびリカバリ機能を提供します。XA ライブラリの提供は、DBMS ベンダーが担当します。

分散トランザクション管理

Oracle Tuxedo は、1 台または複数のサーバー・マシンの発信ポイント（通常はクライアント上）からアプリケーションの代わりにトランザクションを管理し、発信元のクライアントに返すということに特化しています。トランザクションが終了すると、Oracle Tuxedo は、トランザクションに関連する全システムで一貫性が保たれていることを確認します。Oracle Tuxedo では、トランザクションの実行、システム間でのルーティング、実行のロードバランシング、そして障害発生後の再起動が可能です。

Oracle Tuxedo は、複数のサイトからアクセスされるデータや、さまざまなデータベース製品で管理されるデータの整合性を確認します。トランザクションの参加者を追跡し、2 フェーズ・コミット・プロトコルを監視して、トランザクションのコミットとロールバックが各サイトで正しく処理されていることを確認します。

また、Oracle Tuxedo システムは、サイト障害やネットワーク障害、グローバル・リソースのデッドロックの発生時に、トランザクションのリカバリを調整します。Oracle Tuxedo システムでは、各種リソース・マネージャとの通信に X/Open XA インタフェースを使用します。このインタフェースは、Oracle Tuxedo の開発者が提案し、X/Open の承認を受けたトランザクション・マネージャとリソース・マネージャ間のおける、分散トランザクションを管理するための標準インタフェースです。

Oracle Tuxedo システムには、独自の ATMI トランザクション管理機能（ルーチン、verb）に加え、トランザクション境界の X/Open TX インタフェースが組み込まれています。このインタフェースを使用すれば、アプリケーション記述者がアプリケーション内の操作グループを括弧で囲み（トランザクション境界を定義して）、全操作を実行するか、いずれの操作も実行しないかを設定できます。つまり、1 つのアトミックな作業単位としてトランザクションのコミットやロールバックを実行し、マシンに障害が発生した場合にも、全関連データベースの同期を維持できます。

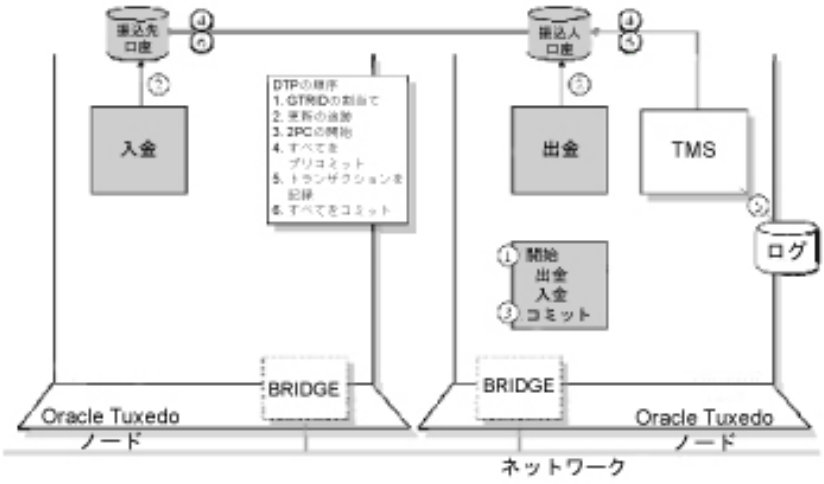

図 2. 分散トランザクション処理のデータ整合性

トランザクション調整サービス

Oracle Tuxedo は、通信プラットフォームとして、トランザクション処理中に更新された DBMS などのリソースを追跡します。トランザクションのコミット時には、2 フェーズ・コミット・プロトコルにより使用リソースが調整されます。トランザクション管理の X/Open DTP "TX" API として、tx_begin、tx_commit、tx_rollback がサポートされています。オプションで、トランザクションを自動的に開始およびコミットするよう設定して、明示的なプログラムを不要にすることもできます。

トランザクションを開始すると、グローバル・トランザクション識別子（GTRID）が生成されます。サービス・リクエストが処理されると、Oracle Tuxedo は関連する DBMS に対し、GTRID に代わって作業を開始および終了するよう指示します。GTRID のローカル・トランザクションへのマッピングは、DBMS が担当します。トランザクションがコミットされると、関連する DBMS のリストが TMS に渡されます。TMS は、XA インタフェースを使用して 2 フェーズ・コミットを実行します。プリコミット・フェーズが完了すると、GTRID ステートがディスクに記録され、中断されたコミット・フェーズがリカバリ可能になります。

トランザクション・リカバリ・サービス

障害からのリカバリは、TMS によっても実行できます。たとえば、ノード障害が発生すると、ノードの動作再開時に未処理トランザクションはすべて正しくコミットされるか、すべて中断されます。トランザクション・ログと XA プリミティブを組み合わせて使用し、DBMS ステートの問合せを実行することにより、必要な基本サポートが提供されます。

透過性

トランザクションの調整とリカバリは、アプリケーションに完全に透過的です。トランザクション・マネージャは、DBMS ベンダーの提供する X/Open DTP XA インタフェースを使用して、トランザクションの追跡、調整、およびリカバリをおこないます。

障害発生時の運用維持

Oracle Tuxedo には、障害発生時に運用を維持する強力なツール・セットが備わっています。また、Oracle Tuxedo アプリケーションでは、レプリケーション機能を活用して、最小限のアプリケーション・コードでコンポーネント障害時の回避策を提供できます。

単一データベース構成のレプリケートされたサービス

Oracle Tuxedo は、任意のサービス・レプリケーションを提供します。ノード障害に対する簡単で効果的な防御方法は、重要なサービスを 2 つ以上のノードにレプリケートすることです。この構成では、データベースへの同時アクセスが必要です。これは通常、Oracle OPS などのパラレル DBMS 製品を使用して実現します。この方法は、とくにデータ依存ルーティング（DDR）を使用する場合に便利です。通常運用では、Oracle Tuxedo は複数のノードにサービス・リクエストを分散します。1 つのノードに障害が発生すると、管理コマンドを使用して動作が停止されます。パーティション化されたノードがランタイム MIB から"消去"されると、存続ノードが負荷を引き継ぎます。

同等の選択肢としては、通常運用時に各ノードで異なるサービス・セットを提供し、障害発生時に Oracle Tuxedo の移行機能を使用する方法もあります。つまり、1 つのノードで障害が発生すると、そのサービス・セット（1 つまたは複数の管理グループにカプセル化が必要）が使用可能な別のノードに移行されます。ただし、問題なのは、アプリケーション・サーバー・ノードでの障害発生時に、DTP を使用する場合です。グループの移行時に、Oracle Tuxedo では一元管理構成に記録された文字列 OPEN を使用して、データベースをオープンします。DBMS 製品によっては、障害が発生したデータベースをオープンしようとすると、フォールバックされて代替データベースがオープンする場合があります。その場合はまったく問題ありません。そうでない場合は、管理コマンドで構成を変更する必要があります。こうした変更も自動化することが可能です。

障害が発生したノードで実行中のクライアントは、再接続が必要です。障害が発生したノードでそのクライアントがネイティブ・プロセスとして実行されている場合は、使用可能な別のノードにユーザーがログオンする必要があります。障害が発生したノードのハンドラでホストされているリモート・クライアントの場合は、"障害時再接続"ロジックの実装が必要です。ただし、別のノードに接続されているリモート・クライアントは、影響を受けません。

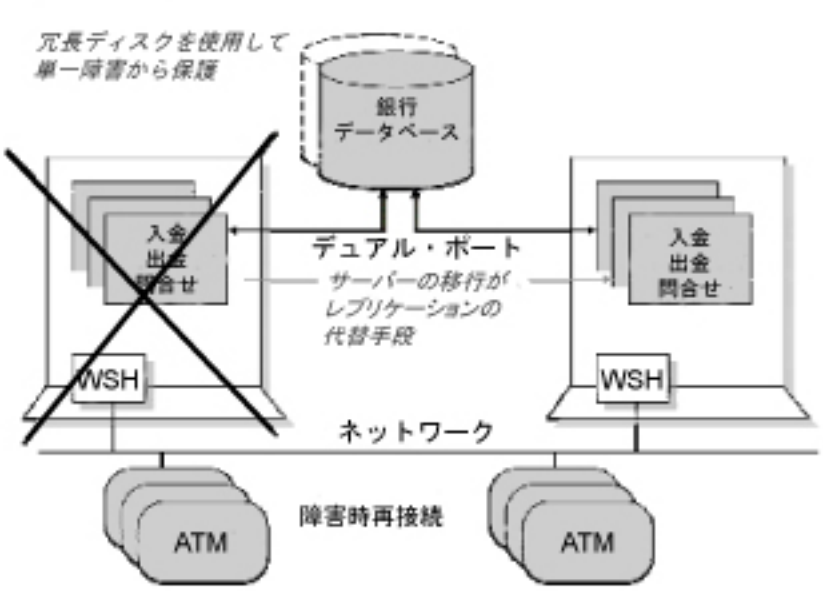

図 3. 単一データベースによるレプリケートされたサーバー

顧客の成功事例

Pacific Gas and Electric Company（PG&E）は、カリフォルニア北部および中部の 7 万平方マイルにわたって、1,500 万人の人々に電力を供給しています。2005 年、PG&E は、顧客に料金と使用量の詳細情報を提供し、消費電力の理解、管理、削減を促進する SmartMeter? プログラムを展開しました。そのため、計測機能を 1 カ月に 1 回から 1 時間に 1 回とし、720 倍に拡張する必要がありました。

同社は、旧来のメインフレームによる顧客ケアおよび請求アプリケーションを、Oracle Tuxedo および Oracle WebLogic クラスタに移行して、メインフレームで実行している旧式の IBM DB2 データベースを、より小規模でエネルギー効率の高いシステムで実行される高いスケーラビリティを備えたクラスタに置き換えました。このクラスタは 8 つの Oracle Database RAC ノードで構成されます。その結果、年間コストが 500 万ドル節減されました。PG&E は、レプリケートされたデータ・サーバー配置アーキテクチャによるレプリケートされたサービスを使用して、メインフレーム・クラスのサービス品質を実現しました。

レプリケートされたデータ・サーバー/Oracle RAC によるレプリケートされたサービス

最近の DBMS には、データベースをレプリケートできる機能が備わっています。方法は異なりますが、最終的にはデータベースの 1 つのコピーの更新が、別のデータベースにレプリケートされます。アプリケーション・プログラマーは、Oracle Tuxedo と DBMS のレプリケーション機能を活用して、単一障害から保護できます。その方法は、先ほど説明したように、2 つ以上のノードにサービスをレプリケートするだけです。

アプリケーションは、1 台のデータ・サーバーの損失を検出して、必要に応じて（環境変数などを使用して）代替データ・サーバーをオープンする手配をし、検出されたサーバーを中断します。Oracle Tuxedo がアプリケーション・サーバーを再起動する際に、使用可能なデータ・サーバーに再接続し、サービス・リクエストの処理を継続します。この方法は、DDR 使用時にも有効です。

Oracle Real Application Clusters（Oracle RAC）は、同じ Oracle データベースにアクセスする、レプリケートされた Oracle データベース・サービスを使用するマシンのクラスタリングをサポートします。Oracle RAC には、複数の Oracle サーバー・マシンに物理的に配置されたインスタンスから同じ Oracle データベースに同時アクセスできる機能や、失敗したデータベース・インスタンスを別の場所にフェイルオーバーできる機能が備わっています。

Oracle RAC は、オラクルのグリッド・コンピューティング製品の主力製品で、データベース・サーバーのクラスタを使用して、データベースの比類なきスケーラビリティと可用性を実現します。さらに、Oracle Database には自動ストレージ管理（ASM）機能が備わっているため、クラスタ化および非クラスタ化データベース環境で、ストレージの共有プールを使用できます。ASM は、ストレージの動的プロビジョニングに加え、簡素化および自動化されたストレージ管理を提供します。

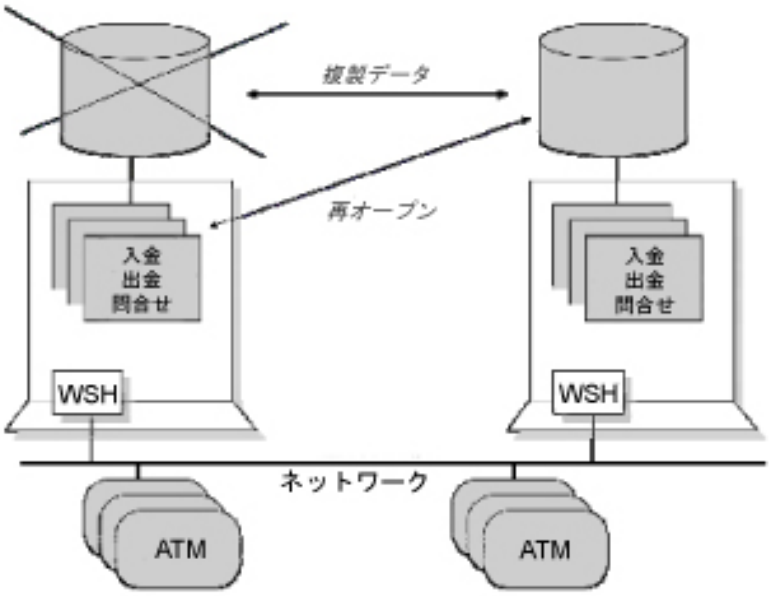

図 4. レプリケート・データベースを使用したレプリケートされたサーバー

障害に対する保護

クラスタ構成でさえ、自然災害や火災、洪水などによるデータセンターの完全な障害を切り抜けることはできません。こうした場合には、遠隔地へのフェイルオーバーが必要です。エンタープライズ・グリッドを複数の場所を包含する設計とし、ワークロードをそれらの場所に動的にシフトして最高の信頼性を実現できます。

Oracle Tuxedo ドメイン・コンポーネントは、地理的な場所や論理的に分散されたビジネス・アプリケーション間の相互運用性を実現するインフラストラクチャを提供します。Oracle Tuxedo のドメイン・レベルのフェイルオーバー・メカニズムにより、プライマリ・リモート・ドメインで障害が検出されると、代替リモート・ドメインにリクエストが転送されます。また、ドメインがリストアされると、プライマリ・リモート・ドメインへのフェイルバックが実行されます。

Oracle Active Data Guard は、スタンバイおよび障害時リカバリ用に、本番 Oracle Database の最新のレプリカを作成する機能を提供します。さらに、このレプリカは、問合せやレポート作成、バックアップなどの、リソースを大量に消費する読取り専用操作でも使用できます。

こうした配置アーキテクチャの例には、オラクルの Maximum Availability Architecture（MAA）があります。このアーキテクチャでは、ハードウェア・コンポーネントやソフトウェア・コンポーネントを含む全テクノロジー・スタック・レイヤーの計画停止時間および計画外停止時間を最小限に抑えるか、または排除することによって、優れた可用性を実現します。障害イベントがデータ破損を引き起こすハードウェア障害によるものであろうと、または地理的に広範囲に影響を与える壊滅的な自然災害によるものであろうと、その範囲に関係なく、データ保護と高可用性を実現します。

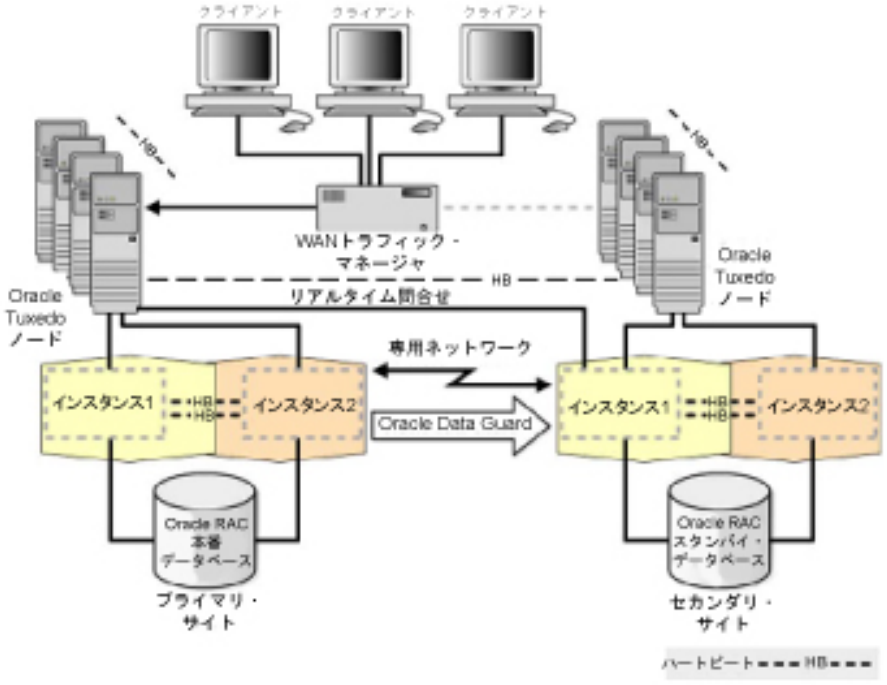

図 5. オラクルの Maximum Availability Architecture

Oracle Corporation 発行「Achieving High Availability with Oracle Tuxedo」の翻訳版です。

また、MAA は、Oracle HA テクノロジーをフル活用した高可用性アーキテクチャの実装に伴う憶測や不確定要素を取り除きます。MAA ベスト・プラクティスは、一連のテクニカル・ホワイト・ペーパーとドキュメントで確認でき、最適な高可用性アーキテクチャの設計、実装、管理を支援します。

レプリケートされたアプリケーション

Oracle Tuxedo は、アプリケーションとデータ・サーバーをレプリケートするメカニズムの提供により、単一ノード障害から保護します。別々の DBMS にアクセスする別々のノードで、プライマリおよびスタンバイの同一のアプリケーション・サービスを維持する方法です。プライマリに障害が発生すると、スタンバイが動作を継続します（その逆も同様です）。

ただし、DBMS の同期を取る必要があります。レプリケートされたデータ・サーバーのシナリオとは異なり、データベース・レプリケーション・メカニズムは使用されません。代わりに、両方のノードで同じ処理を実行する同一リクエストのストリームが 2 つ、アプリケーションで作成されます（最終的には、両方の DBMS に同一の変更が適用されます）。これは、分散トランザクション処理機能と高信頼性キュー機能を使用して実行できます。

DTP を使用して、アプリケーションで単一トランザクションの傘下にあるプライマリ・サービスとバックアップ・サービスの両方が非同期で起動します。非同期の起動により、並列処理がおこなわれて高スループットが実現します。管理者は、単純なネーミング規則を使用して、バックアップ・サービス名を作成できます。また、見せかけの書込みでアプリケーション・プログラマーから並列起動を隠すことができます。

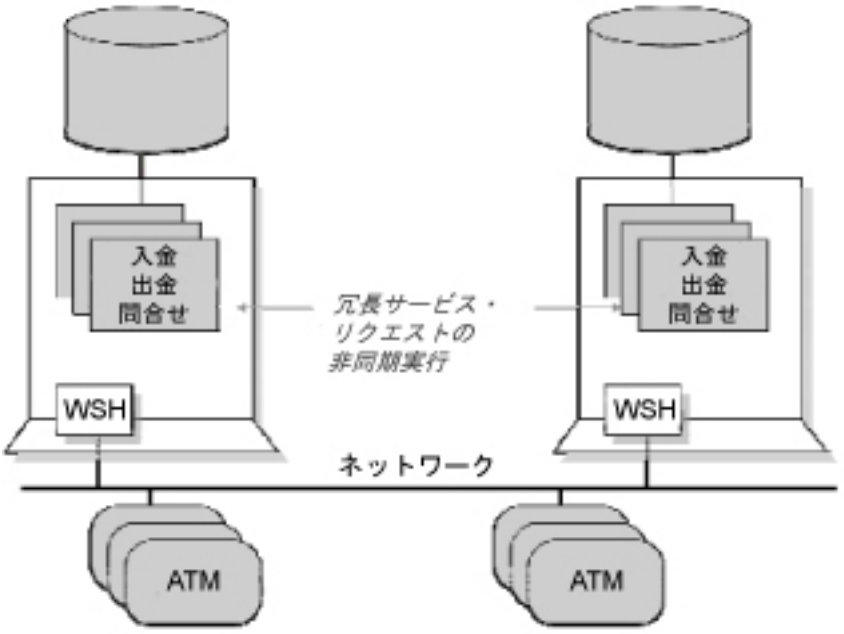

図 6. レプリケートされたアプリケーション

障害発生時の措置は、アプリケーションによって異なります。シンプルな対策としては、万一障害が発生しても使用可能な存続システムで動作が継続されるように、プライマリ・システムとバックアップ・システムのステータスをポストするという方法があります。ここでの真の課題は、修復完了後の DBMS ステートの再同期です。通常、これには休止時間が必要です。再同期方法の 1 つとしては、Oracle Tuxedo の System Q にある高信頼性キュー機能を使用する方法があります。これは、障害が発生すると、2 つのパラレルなオンライン・サービス・リクエストの代わりに、アプリケーションが信頼性の高いキューにサービス・リクエストの 1 つ（障害が発生したノードが送信先のもの）をエンキューするという方法です（これはオンライン・リクエストとパラレル実行することも可能です）。障害が発生したノードが修復されてオンラインになると、エンキューされたサービス・リクエストがすべて適用されます。

リストア処理

リストア処理が自動的に実行されるのは、一部の場合に限ります。通常は、メンテナンス操作を反転させる管理コマンドが必要です。

自動リストア

自動リカバリの項で説明したように、アプリケーション・サーバーに障害が発生した場合のリカバリ、管理プロセス、クライアント・プロセス、ネットワークの切断、またはトランザクションのタイムアウトは、自動的に実行されます。

ノードの再起動

停止されたノードのリストアは、特定のノードに boot コマンド（または activate コマンド）を適用することによっておこないます。ノードがオンラインに復帰すると、不完全なトランザクションが自動的にコミットまたはロールバックされます。

移行したグループのリストア

移行したサービスのリストアは、ノードのリストア完了後に、元のノードにサービスを移行し直すことによっておこないます。

クライアント・プロセス

クライアント・プロセスでは、リストアされたノードへのログオンが必要です。

結論

Oracle Tuxedo は、トランザクション処理や分散された COBOL、C、C++アプリケーションに対応したオラクルの主要製品です。アプリケーションに信頼性、可用性、スケーラビリティ、動的なシステム再構成を提供し、コンピュータやデータベース、ネットワークの異種混交性を感じさせないようにします。さらに、Oracle Tuxedo には優れたフェイルオーバー機能が備わっており、トランザクション処理アプリケーション・コンポーネントを残存ノードに自動フェイルオーバーし、ユーザーへの影響を最小限に抑えることにより、強力な高可用性ソリューションを提供します。

オラクルは他社に先駆けてサーバーのフェイルオーバー方法を多数開発しており、いくつかのサーバー・タイプ用の自動フェイルオーバー機能を IT 部門に提供しています。たとえば、Oracle Database RAC、Oracle WebLogic Server、Oracle Tuxedo、Oracle Coherence クラスタは、クラスタ内の複数のサーバーの障害に耐えつつ動作を継続できます。IT 部門は、障害が発生したサーバーをサービスから除外して、修復または置換えをおこなったあと、サーバー・グリッドに再び追加するだけで済みます。ロードバランシング、ワークロード管理、オーバーロード保護、自動移行、サービスや全サーバーのフェイルオーバーによって、アプリケーションの常時稼働が実現します。

ビジネスの可用性要件に最適なアーキテクチャを選択して導入することは、非常に困難な場合があります。こうしたアーキテクチャは、適切な冗長性を備え、いかなる種類の停止に対しても適切な保護を提供し、一貫して高いパフォーマンスと堅牢なセキュリティを実現する一方で、簡単に配置、管理、拡張できるものでなくてはなりません。また、こうしたアーキテクチャは、十分に理解されたビジネス要件によって決定する必要があります。

そのようなアーキテクチャを構築、実装、維持するには、IT システムとビジネス・プロセスの技術面と運用面の両方に関連した高可用性のベスト・プラクティスが必要です。このベスト・プラクティスを使用すれば、高可用性アーキテクチャの設計に伴う複雑さを緩和し、最小限のシステム・リソースを使用して可用性を最大限に高め、高可用性システムの実装コストと保守コストを削減し、高可用性アーキテクチャをほかのビジネス領域に簡単に複製できます。

高可用性分析フレームワーク、ビジネス推進要因、システム機能を包含した明確な高可用性のベスト・プラクティスは、運用の弾力性を向上し、ビジネスの俊敏性を高めます。

Oracle Tuxedo による高可用性の実現
2008 年 10 月
著者：Deepak Goel、Mark Rakhmilevich

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

海外からのお問い合わせ窓口：
電話：+1.650.506.7000
ファクシミリ：+1.650.506.7200
www.oracle.com

Copyright © 2008,Oracle and/or its affiliates.All rights reserved.
本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。Oracle は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。そのほかの名称はそれぞれの会社の商標です。1008