

共有ストレージ上のOracle
WebLogic : ベスト・プラクティス

Oracle Maximum Availability Architecture

ホワイト・ペーパー

2013年5月

Maximum Availability Architecture

高可用性に関するオラクルの
ベスト・プラクティス

はじめに	1
共有ストレージのメリット	1
このホワイト・ペーパーの構成	2
概要	3
Oracle Fusion Middlewareのアーティファクト	3
ディレクトリ構造	4
共有ストレージの Protokol およびデバイス	10
Windowsクライアントのサポート	13
段階的な例：Oracle FMWアプリケーション用のトポロジのインストール および構成（Exalogicに適合）	14
トポロジ	14
ハードウェア・プロビジョニング	15
アプリケーションのインストール	16
ドメインの作成	16
ポスト・ドメインの作成	17
HTTP Serverの構成	20
スケールアウト：新しいホストのプロビジョニングと管理対象サーバーの再配置	20
高度なノード・マネージャ構成	23
例：1つの共有ドメイン・ホーム	24
1つのドメイン・ホームのみの作成	24
管理サーバーの可用性	25
管理対象サーバーの移行	25
新しいマシンの追加	25
カスタム・アプリケーションのベスト・プラクティス	25
アプリケーションのデプロイ	25
構成およびステート管理	26
クラスター・ディレクトリ	27
一元化されたログ	28
ライフ・サイクルのベスト・プラクティス	29
複数のボリューム	29
バックアップとリカバリ	30
パッチ適用とアップグレード	30

共有ストレージのトラブルシューティング.....	30
認証および権限に関する問題.....	30
一般的なファイル・ロックに関する問題.....	31
Oracle HTTP Serverのパフォーマンスとロック	33

はじめに

このホワイト・ペーパーでは、ソフトウェアと構成を共有ストレージに保存する Oracle WebLogic Server のエンタープライズ・アプリケーション・デプロイメントのベスト・プラクティスについて説明します。このホワイト・ペーパーで提示されるディレクトリ構造は、共有アクセスに適しており、かつ、ローカル・ストレージとリモート・ストレージのハイブリッド・モデルに対応できる柔軟性を持ちます。また、カスタム・アプリケーションをデプロイするためのベスト・プラクティスについても説明します。このホワイト・ペーパーは、Oracle Fusion Middleware (Oracle FMW) 11g にのみ適用されます。

共有ストレージのメリット

共有ストレージは通常、専用のストレージ・サーバー上に配備され、すべてのクライアント・マシンから等しくアクセスできる外部物理ディスクです。共有ストレージ上のソフトウェアを使用することにより、複数の物理クライアント・マシンへのデプロイメントが容易になります。

この構成には、ソフトウェアと構成の一元管理が提供されるというメリットがあります。新しいホストごとに新しいソフトウェアをインストールして構成する必要がなくなり、多数の物理サーバーにわたるプロビジョニングが簡素化されます。共有される保存場所には、マシンに障害が発生し、フェイルオーバーが実行される場合に、すぐにアクセスできる必要がある共通の構成や共通のバイナリおよびアーティファクトも保存できます。クラスタのすべてのメンバーがアクセスできる共有アーティファクトは、一部のアプリケーションの要件にもなっています。また、共有モデルでは、個々の物理マシンに対する多数の依存関係も排除されるため、これらのマシンを相互に交換する必要のある環境にも適しています。

最後に、共有ストレージにより、ファイル・システム・オブジェクトの古いインカネーションへのアクセスを可能にするスナップショットやレプリケーションなどの組込み機能が提供されることで、バックアップ/リカバリ操作が簡素化されます。

このホワイト・ペーパーの構成

このホワイト・ペーパーでは、すべてのクライアント・マシンが同じバイナリおよび同じクラスタ・ディレクトリにアクセスする共有ストレージ・モデルを提示します。ドメイン・ホームは、ローカル・ディスクまたは共有ディスクのいずれに配置することもできます。

このホワイト・ペーパーの最初の項では、共有ストレージ上に保存される Fusion Middleware インストールに関するディレクトリ構造のガイドラインについて説明します。このディレクトリ構造は、Fusion Middleware アーティファクトの分類（バイナリ、ドメイン・ディレクトリ、共有ディレクトリの3つのタイプに大きく分類される）に基づきます。また、各種の共有ストレージ・オプションおよび推奨構成の概要についても説明します。

このホワイト・ペーパーの2つ目の項では、実施例を提示します。WebLogic Server をインストールして共有ストレージ・ディレクトリを構成する手順と、エンタープライズ・アプリケーションのカスタマイズに固有の考慮事項を順番に説明します。これには、管理性とパフォーマンスを最適化するために異なる種類の共有ストレージまたはローカル・ディスク上で Oracle FMW ディレクトリ構造をパーティショニングする方法も含まれます。さらに、ドメインを追加ノードにスケールアウトする手順も示します。

別の実施例では、クラスタ全体で共有され、すべてのサーバーを実行するために使用されるただ1つのドメイン・ホームを使用してカスタム・アプリケーションをデプロイする方法を示します。この構成のメリットは、より迅速なプロビジョニングが実現されることです。このオプションは、アプリケーションが対応している場合に使用できます。

このホワイト・ペーパーの最後の項では、カスタム・アプリケーションに関する考慮事項とベスト・プラクティス（トポロジのライフ・サイクル、構成の管理、デプロイメントの管理、最後に、共有ストレージに関する問題のトラブルシューティング）について説明します。

概要

このホワイト・ペーパー全体を通じて述べる基本的な物理アーキテクチャは、ネットワーク上の同じストレージ・サーバーにアクセスする1台以上の物理マシンのアーキテクチャです。ホスト・マシンには、LinuxやWindowsなどが使用されます。ストレージ・サーバーは、ストレージの使用を可能にするもう一つのデバイス（NAS、SANなど）です。

これらのホスト・マシンは、それ自体にローカル・ストレージを備えています。外部ストレージ・サーバー上に存在するストレージ・ボリュームをマウントすることもできます。

共有ストレージへのOracle Fusion Middlewareソフトウェアとファイルの配置方法を理解するには、まず、インストールにおける各種のアーティファクトとその特性を理解する必要があります。

Oracle Fusion Middlewareのアーティファクト

Oracle Weblogicのインストールは、異なる種類のアーティファクトによって構成されます。これらのアーティファクトはそれぞれ、異なるライフ・サイクル特性を持ち、それらが共有されるかどうかという要件や、読取り/書込みプロファイルも異なります。

共有の要件

アーティファクトは、共有、非共有（プライベート）、または使用可能のいずれかである必要があります。

共有：このアーティファクトは、共有ストレージ上に存在し、すべてのクライアント・マシンが同時に表示およびアクセスできます。

使用可能：このアーティファクトも共有ストレージ上に存在しますが、一度に1つのクライアント・マシンだけが表示およびアクセスできます。たとえば、このマシンに障害が発生すると、別のクライアント・マシンがアーティファクトにアクセスできるようになります。

非共有：このアーティファクトは、ローカル・ストレージ上に常駐させることができ、別のマシンからアクセスできる必要はありません。

読取り/書込みプロファイル

アーティファクトは、固有の読取り/書込み要件も持ちます。

読取り専用：このアーティファクトは、まれにしか変更されませんが、実行時には読取り専用になります。

読取り/書込み：このアーティファクトは、実行時に読取りと書込みの両方が可能です。

アーティファクトの特性：

上記のことを念頭に置くことで、Oracle Fusion Middlewareの各種アーティファクトとその特性を簡単に説明できます。これらの種類のオブジェクトについては、次の項でそれぞれ詳しく説明します。

	共有	読取り/書込み
バイナリ	共有	読取り専用
管理対象サーバー・ドメイン・ホーム	非共有	読取り/書込み
管理サーバー・ドメイン・ホーム	使用可能	読取り/書込み
クラスタ・ファイル	使用可能	読取り/書込み
ノード・マネージャの構成	非共有	読取り/書込み
アプリケーション固有のファイル	共有	読取り/書込み

ディレクトリ構造

Fusion Middlewareアーティファクトの推奨される共有ディレクトリ構造は、以下の特性を持ちます。

バイナリと構成の分離

最上位の区分は、Fusion Middlewareホーム（FMW_HOME）のバイナリと/adminなどのディレクトリの下にある構成ディレクトリとの区分です。

バイナリには、Oracle WebLogicホーム、Oracle JDK、Coherenceバイナリ、および必要なすべてのFusion Middlewareバイナリが含まれます。バイナリは、優先的または排他的に読み込まれ、まれに書き込まれるオブジェクトです。

構成ディレクトリには、すべてのドメイン・ホーム、アプリケーション・ファイル、およびノード・マネージャ構成が含まれます。これらのディレクトリは、実行時に読取りと書込みの両方が可能です。

共有バイナリ

すべてのマシンが、一連の同じバイナリを使用してプロセスを実行します。バイナリは、一度インストールされると、後のパッチ適用時やアップグレード時には修正されるだけです。

ローカルまたは共有管理対象サーバー・ドメイン・ホーム

管理対象サーバー・ドメイン・ホームは、ローカル・ホスト・マシン上で管理対象サーバーを実行するために使用されます。パフォーマンス上の理由およびその他の理由から、これらのドメイン・ホームは、ホスト・マシンのローカル・ストレージ上に配置することが望ましい場合があります。ただし、これらは、共有ストレージ上の個別の専用ボリューム（マシンごとに1つのボリューム）に配置することもできます。さらに、アプリケーションが許可する場合は、すべてのクライアント・マシンが同じドメイン・ホームを使用して、それらの管理対象サーバーを配備することもできます。

管理サーバーのドメイン・ホーム

管理サーバーを実行するドメイン・ホームは、個別のボリュームに配置され、このボリュームは任意のマシンにマウントできます。このため、管理サーバーをホストするマシンへの障害の発生や、その他の理由で使用できなくなっても、その管理サーバーをすぐに使用可能にすることができます。

共有クラスタ・ファイル

クラスタのすべてのメンバーで使用できる必要がある場合があるファイルおよびディレクトリは、それらの専用ディレクトリに分離されます。これらには、JMSファイルや、クラスタ内の1つのメンバー・マシンにのみ属しながら、フェイルオーバーが実行されると他のマシンで使用できる必要がある場合があるアーティファクトやトランザクション・ログが含まれます。

ローカル・ノード・マネージャ・ディレクトリ

各クライアント・マシンは、それ自体のノード・マネージャとノード・マネージャ構成ディレクトリを持ちます。ノード・マネージャは1つの物理マシンに関連付けられるため、オプションで、これらをローカル・ストレージ上に配置することもできます。

共有アプリケーション固有のファイル

共有される必要があり、かつ同時にアクセス（読取り/書込み）できる必要のあるファイルは、アプリケーションによって異なります。Oracle FMW WebLogicインフラストラクチャにはこの要件を持つファイルは含まれていませんが、一部のOracle FMW製品（Oracle WebCenter Contentなど）およびカスタム・アプリケーションではディスク上に共有ファイルが必要な場合があります。

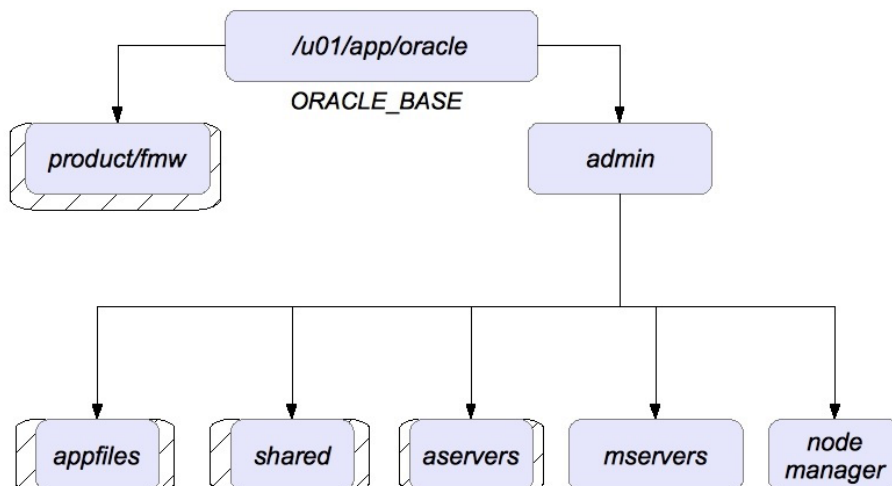


図1：最上位のディレクトリおよびマウント・ポイント

上の図は、最上位のディレクトリ構造を示しています。すべての製品およびアプリケーション・ファイルは、\$ORACLE_BASEとして指定するディレクトリの下に保存されます。

すべてのバイナリは、`$ORACLE_BASE/product/fmw`ディレクトリに保存されます。これは、クラスタ内のすべてのマシンによってマウントされる共有ストレージ上の個別のボリュームです。

図1内の斜線の枠は、マウント・ポイントを表しています。たとえば、次のディレクトリは、ローカル・マウント・ポイントです。

```
/u01/app/oracle/product/fmw
```

上の図では、`msservers`ディレクトリと`nodemanager`ディレクトリがローカル・ストレージ上にあり、ローカル管理対象サーバーと、マシンのノード・マネージャを実行するために使用されます。前述のように、これらのディレクトリは、オプションで共有ストレージ上に配置できますが、その場合でも、複数のノードでマウントする必要はありません。

管理サーバー・ドメイン・ホームは、`aservers`ディレクトリの下にあります。これは、任意のマシンが独立してマウントできる個別のディレクトリです。

`admin`ディレクトリの下にあるディレクトリの詳細を、次の図に示します。

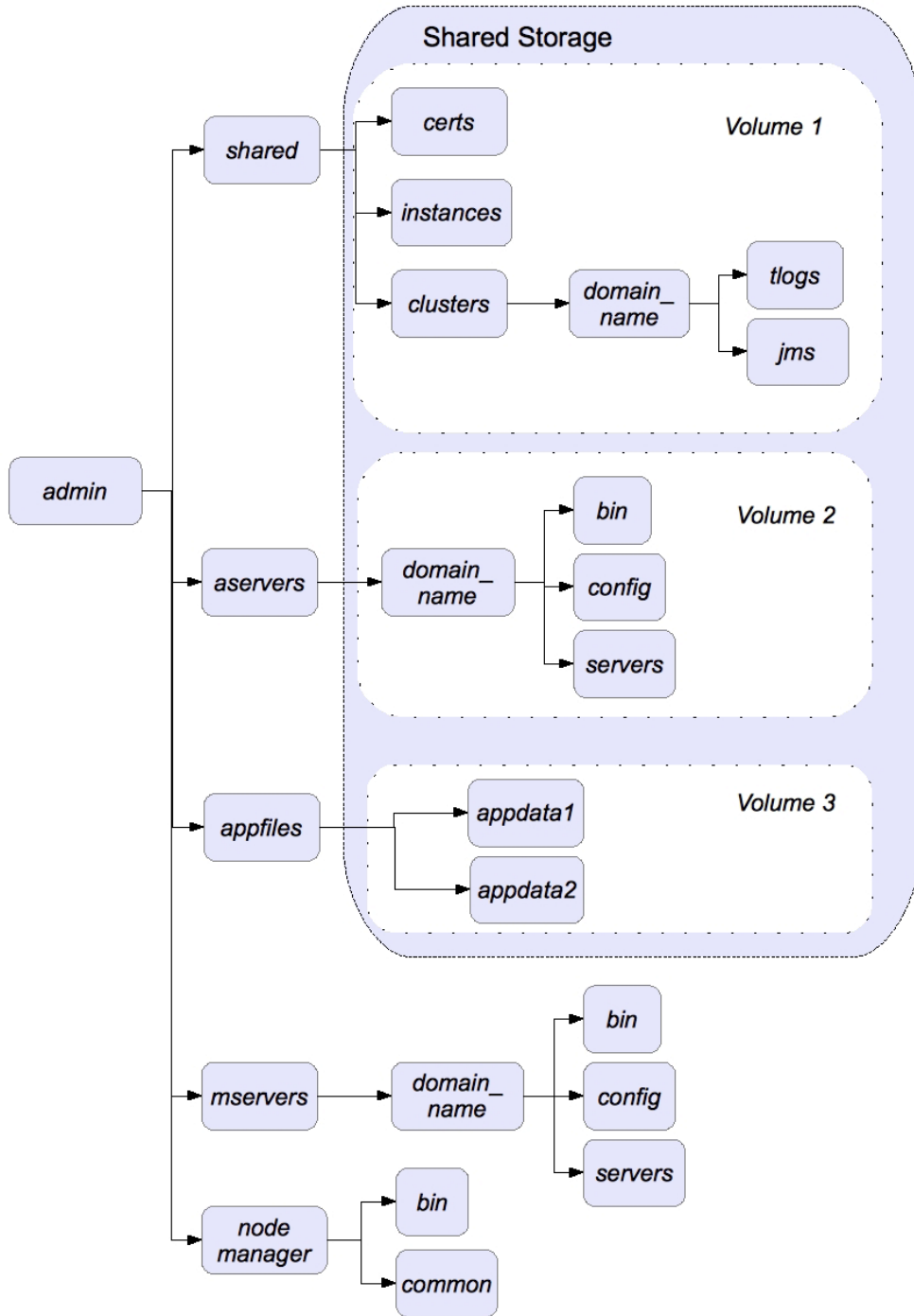


図2 : adminディレクトリの下のディレクトリ

各クライアント・マシンは、共有アーティファクト（永続ストア、トランザクション・ログ、証明書とキーストアなど）にアクセスするために、sharedボリューム（ボリューム1）をマウントします。

aserversボリューム（ボリューム2）は、ドメインの管理サーバーを実行するマシンによってマウントされます。

appfilesボリューム（ボリューム3）は、同時にアクセスされるファイル用のストアを必要とするアプリケーションを実行するすべてのマシンによってマウントされます。

mserversディレクトリと**nodemanager**ディレクトリは、共有ストレージ上にある必要はありません。オプションで、**mservers**ディレクトリと**nodemanager**ディレクトリを共有ストレージ上に配置することもできます。**mservers**ディレクトリは、各マシンのプライベート・ディレクトリとすることも、（このホワイト・ペーパーの例のように）1つ以上のマシンで共有することもできます。

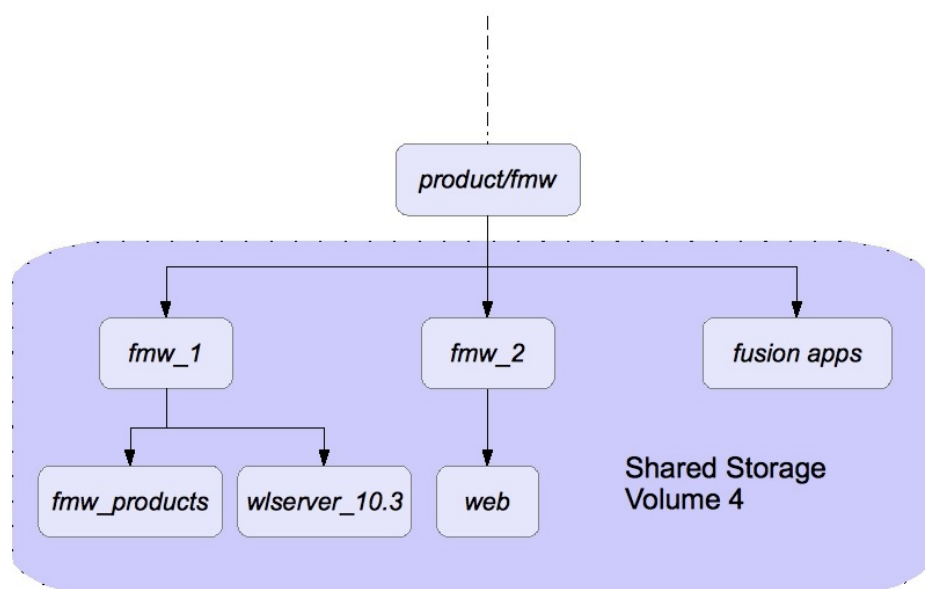


図3：共有ストレージ上のOracle FMWディレクトリ構造

バイナリ・ディレクトリは、すべてのクライアント・マシンによってマウントされ、共有されます。

一部のディレクトリにとって重要なディレクトリを以下に示します。

\$ORACLE_BASE

これは、最上位のOracleディレクトリです（今回の例では、/u01/app/oracle）。

\$ORACLE_BASE/product/fmw

すべてのOracle FMW製品ディレクトリは、このディレクトリ・ツリーの下に存在します。これは、すべてのマシンによってマウントされ、Oracle WebLogic Serverを実行するために使用されます。

\$ORACLE_BASE/product/fmw/fmw_install

これは、特定のOracle FMWバイナリをインストールするためのディレクトリ・ツリーです。実際には、個別の製品またはOracle FMWの各バージョン用に、fmwディレクトリの下に個別のディレクトリが必要な場合があります。

\$ORACLE_BASE/admin

バイナリ以外のすべてのものが、このディレクトリ・ツリーの下に存在します。

\$ORACLE_BASE/admin/mservers

すべての管理対象サーバー・ドメインが、このディレクトリの下に存在します。このディレクトリは、各マシン上にローカルに配置することもできます。

\$ORACLE_BASE/admin/mservers/*domain_name*

ここで、*domain_name*は、'MyDomain'などの実際のドメイン名です。

\$ORACLE_BASE/admin/shared

このディレクトリは、異なる種類の共有アーティファクト用の単一マウント・ポイントおよびコンテナとして使用されます。

\$ORACLE_BASE/admin/shared/instances

Oracleインスタンス用のディレクトリです。Web層に含まれます。

\$ORACLE_BASE/admin/shared/certs

証明書やアイデンティティおよびトラスト・キーストアを格納するために使用されるディレクトリです。すべての証明書は、一意の名前を持ちます。キーストアは、同じドメインおよび複数のドメインにわたるすべてのメンバーによって共有されます。

\$ORACLE_BASE/admin/shared/clusters/*domain_name*

ドメインごとに、すべてのクラスタ情報が、このディレクトリに格納されます。オプションで、このディレクトリの直下をクラスタ名にすることができます。

このディレクトリに含まれるその他のサブディレクトリは、アプリケーションによって異なります。一部のアプリケーションでは、このディレクトリに、JMSおよびトランザクション・ログ用のディレクトリが含まれる場合があります。

\$ORACLE_BASE/admin/nodemanager

ノード・マネージャ・ホームは、ここに移動されます。この構成では、物理マシンごとに1つのノード・マネージャが存在します。ノード・マネージャ・ホームは、
\$ORACLE_BASE/admin/nodemanager/{machine_name}/commonにあります。ノード・マネージャ
起動スクリプトは、\$ORACLE_BASE/admin/nodemanager/{machine_name}/binにあります。

この個別ノード・マネージャ構成を実現する手順は、このホワイト・ペーパーで説明します。

ここでも、machine_nameディレクトリは**オプション**であり、各クライアント・マシンがローカル・ディスク上にnodemanagerディレクトリを持つ場合は不要です。

共有ストレージのプロトコルおよびデバイス

このホワイト・ペーパーで使用する主要な例は、NFS v3またはNFS v4として公開されているリモート・ファイル・システムにアクセスするLinuxクライアントです。ここでは、オプションの範囲と、これらのデバイスを構成およびマウントする方法について簡単に説明します。特定の共有ストレージ・ベンダーおよびマシンについては、ここでは説明しません。

複数のクライアント・マシンによってアクセスできるストレージ用の一部のオプションを、次の図に示します。

	Windowsソリューション	Linuxソリューション
ブロック・ストレージ	ISCSI/NTFS (MSCS)	ISCSI/EXT3
共有ファイル・システム	CIFS、NFS、ACFS、DBFS	NFS、ACFS、DBFS

図4：WindowsおよびLinuxの共有ストレージ・オプション

リモート・ブロック・デバイスは、ローカルSCSIデバイスと同じプロトコルの多くを使用してリモート・ストレージにアクセスすることを可能にするISCSIなどのプロトコルによって使用可能になります。その後、クライアント・マシンに適したNTFS (Windows) またはEXT3 (Linux) などのファイル・システムを、このデバイス上で構成できます。その結果、これらのファイル・システムは物理的に共有可能になりますが、このソリューションには、同じファイル・システムおよびファイルに同時にアクセスする複数のクライアント・マシンを管理するために必要なセマンティックおよびプロトコルがありません。

ここでは、フェイルオーバー・クラスタとも呼ばれるMicrosoft Cluster Services (MSCS) を例に挙げます。MSCSも、共有アーティファクトへの複数の**同時**アクセスを管理するためのソリューションではありません。フェイルオーバー・クラスタは、1つのクライアント・マシンだけがリモートNTFSファイル・システムに対して読取り/書込みを実行し、このマシンに障害が発生した場合にはリモート・ファイル・システムが自動的に別のクライアント・マシンに使用可能になるという状態を確立することによって機能します。

共有ファイル・システムは、ファイル・システムと、複数の同時アクセスを管理するための一連のプロトコルの両方を指します。これらのプロトコルでもっとも一般的なものは、CIFS (Windowsクライアント用) およびNFS (Unixクライアント用) です。これらのソリューションは、バックエンド・ファイル・システム (ZFS、EXT3など) と、アクセスを管理および仲介するサーバー・プロセス (NFSDなど) から成ります。

オラクルのACFSファイル・システム・ソリューションは、これらとは多少異なり、サーバーがアクセスを管理する代わりにOracle Automatic Storage Management (Oracle ASM) およびOracle Clusterwareがアクセスを管理します。オラクルのDBFSファイル・システムは、FUSE (Filesystem in User Space) ファイル・システム・インタフェースをOracleデータベースに提供して、データベース自体が同時実行性を管理することを可能にします。

これらのソリューションのそれぞれについて、この後で詳しく説明します。また、それぞれについて、Oracle Fusion Middleware用の共有ファイル・システム・ソリューションとして使用できるかどうかについても説明します。

NFS v3

NFSバージョン3はおそらく、リモート共有ファイル・システムを必要とするUnixクライアント用のもっとも一般的な規格です。クライアント・マシンとストレージ・サーバーの間の権限は、ユーザーのUIDを使用することによってマッピングされます。NFS v3は、ほとんどのデバイス上で比較的容易に構成およびエクスポートされます。その後、クライアント・マシンは、次のようなコマンドによってリモート・デバイスをローカルにマウントできます。

```
$ mount -t nfs storage_machine:/export/nfsshare
/locmntddir -o
rw,bg,hard,nointr,tcp,vers=3,timeo=300,rsize=32768,wsiz
e=32768
```

このコマンドにより、リモート・ファイル・システムが/locmntddirにマウントされます。上記のコマンドで指定されている値のほとんどはデフォルト値であるため、NFS v3デバイスは次のようなコマンドでもマウントできます。

```
$ mount -t nfs storage_machine:/export/nfsshare
/locmntddir -o nointr,timeo=300
```

rsize、wsize、およびtimeoの適切な値は、クライアント・マシンとストレージ・サーバーの間のネットワークの属性によって異なります。たとえば、タイムアウト値を大きくすることは、タイムアウト・エラーを最小限に抑えるために役立ちます。

属性のnoacとactimeoは、ここでは使用されません。これらは、属性キャッシングを無効にするために使用されます。また、Oracleデータ・ファイルなどのオブジェクトがNFS上に配置される場合に必要になることがあります。ただし、これらの属性は、ほとんどのWebLogicアプリケーションまたは特定のOracle FMWアプリケーション (Oracle Service-Oriented Architecture (Oracle SOA)、WebCenter、Oracle Business Intelligence (Oracle BI) など) については不要です。また、これらのパラメータを有効にすると、パフォーマンスも低下します。

その他のオプションとして、noatimeなどがあります。noatimeは、アクセス時間書込み要件をなくすことにより、読取り集中型アプリケーションのパフォーマンスを少し向上させます。

NFS v4

NFS v4は、NFS v3よりもいくつかの点で優れています。その一つは、UIDに依存せずにユーザー権限をマッピングできるNISなどの認証プロトコルのサポートです。その他に、リースによるファイル・ロックの有効期限もサポートしています。

指定された時間が経過すると、ストレージ・サーバーは、要求のないファイル・ロックを解除します。デフォルトのリース有効期限は、ストレージ・サーバーによって異なります。一般的な値は、45または120秒です。Sun ZFSシステムのデフォルトは、90秒です。フェイルオーバー・シナリオでは、サーバーに障害が発生したときにそのサーバーが所有するファイルを別のサーバーが要求またはアクセスする必要がありますが、これは、リース有効期限の後に可能になります。新しいサーバーがファイルのロックを正常に取得できるように、この値は、フェイルオーバー時間よりも短く設定する必要があります。たとえば、JMSファイルベースの永続性を使用するクラスタでは、この値を、サーバー移行の時間よりも短く設定する必要があります。

この点は、ロックが無期限に保持されるNFS v3と異なります。NFS v3では、中断されたプロセスによって保持されたロックを手動で解除する必要があります。

NFS v4ファイル・システムは、NFS v3と同様に次のようなコマンドでマウントできます。

```
$ mount -t nfs4 storage_machine:/export/nfsv4share /locmntddir
-o nointr,timeo=300
```

CIFS

Windowsストレージでは、一般に、共通インターネット・ファイル・システム (CIFS: Common Internet File System、SMBとも呼ばれる) が使用されます。CIFSは、同時実行性の低いシナリオにおいてパフォーマンスを向上させるコミット時ロックをサポートします。

コミット時ロックでは、クライアント (マシン1) がファイル・ロックを要求し、CIFSサーバーに通知します。クライアントは、それ以降、そのクライアントがロックを保持すると想定できます。クライアントは、NFS v4のようにリースを更新する必要はありません。別のクライアント (マシン2) がロックを要求すると、サーバーは、マシン1に通知してロックの解除を要求します。

フェイルオーバー・シナリオでは、CIFSサーバーは、第2のクライアントがロックを要求するまで、マシン1によるロックの解除を要求しません。マシン1に障害が発生し、マシン1からのレスポンスがなくなると、サーバーは、タイムアウト時間の経過後にプロセスを進め、ロックを解除します。

権限は、Windows Active Directory認証を使用してマッピングできます。CIFS共有は、ほとんどのストレージ・サーバーで、NFS v3またはv4共有としても公開できます。サーバーは、異なるロック・メカニズムを自動的に調整します。

CIFS共有にアクセスするには、

[\\storage_machine\export\cifsshare](#)形式のリモート・ドライブをローカル・ドライブ (E:など) にマッピングする必要があります。

Oracle ACFS

共有ストレージのインストールは、オラクルのASM クラスタ・ファイル・システム (ACFS) でも正常に実行できます。また、これは、パフォーマンスに優れたソリューションであることが確認されています。ただし、ACFSに関するガイドラインおよび使用方法は、このホワイト・ペーパーの対象外です。

DBFS

Database Filesystem (DBFS) は、Oracle データベースに対する FUSE (Filesystem in UserSpace) フロントエンドです。ファイルは、CLOBs として Oracle データベースに格納され、複数のリモート・クライアントによってアクセスされます。ファイルは、データベース・オブジェクトであるため、データベースによって提供されるすべてのトランザクション制御およびセマンティックによって管理されます。DBFS への Oracle FMW ミドルウェアの実装は、このホワイト・ペーパーの対象外です。

次の表に、各種の共有ストレージ・オプションと、このホワイト・ペーパーでのそれらへの対応をまとめます。

共有ストレージ	このホワイト・ペーパーでの対応
NFS V3	推奨
NFS V4	推奨
CIFS	推奨
ACFS	このホワイト・ペーパーの対象外
DBFS	このホワイト・ペーパーの対象外
ISCSI/NTFS	
ISCSI/EXT3	複数の同時アクセス (共有アプリケーション・データ用など) のオプションとして非推奨

Windows クライアントのサポート

以降の項では、Linux マシン用のコマンドを示します。同じ手順を Windows クライアントの構成に使用できます。相当する Windows コマンドが明らかな場合は (ディレクトリ作成やファイル編集のコマンドなど)、その Windows コマンドを示していません。いくつかのコマンドでは、Windows の手順が大きく異なるため、その項の最後に必要な説明を追加しています。

段階的な例：Oracle FMWアプリケーション用のトポロジのインストール および構成（Exalogicに適合）

この例では、1つ以上のボリュームにわたって分散する共有環境にOracle FMW WebLogicをインストールする手順を順番に示します。カスタム・エンタープライズ・アプリケーションをこの環境にデプロイすることが前提となっています。

トポロジ

このトポロジは、以下の要素によって構成されます。

2つ以上のアプリケーション・ノード

ここで述べるように、アプリケーション・ノードをいつでも追加して迅速にプロビジョニングできることが前提となっています。

共有ストレージ

共有ストレージ・デバイスに、すべてのバイナリ、すべてのドメイン・ファイル、すべての構成ファイルが保存されます。

それでも、プロビジョニングでは、いくつかのローカル・ファイル（beahomelist、Oracle Inventory など）が必要です。これらのファイルをローカル・ディスク上に保存する主要な理由は、他のローカル・インストールとの共存を可能にすることです。

それ以外には、このトポロジでローカル・ディスク上に保存されるものはありません。共有ディスクに、すべてのソフトウェアとすべてのアプリケーションが保存されます。これにより、移植性が向上し、メンテナンスが容易になります。ただし、この環境の準備および管理（どのマシンから何を実行するかを含む）とパッチおよびアップグレードによるこの環境のメンテナンスの両方で注意が必要です。

プライベート・ローカル・ドメイン・ホーム

各マシン上の管理対象サーバーは、プライベート・ドメイン・ホームから実行されます。このドメイン・ホームは、共有ストレージ上に存在しますが、1つのノードのみによってマウントされます。

管理サーバーは、共有ストレージ上にそのサーバー専用のドメイン・ホームを保持するため、どのノードにもマウントできます。

1つのノード・マネージャ・ボリューム

単純化するために、すべてのノードのノード・マネージャを共有ストレージの1つのボリューム上に作成します。ノード・マネージャ・ディレクトリは、マシン名によって分割されます。

管理サーバー・フェイルオーバー

管理サーバー・ドメイン・ホームは、共有ストレージ上に存在するため、どのマシンにもマウントできます。管理サーバーはシングルトンであり、競合を避けるために、一度に1つのマシンにのみマウントすることを推奨します。

ハードウェア・プロビジョニング

共有ディスク

Oracle Weblogic環境を実行するすべてのマシンに共有ディスクをマウントします。

/adminおよび/product/fmwまでのローカル・ディレクトリ構造を作成した後、共有ストレージ上に必要なボリュームを作成し、それらをマウントする必要があります。次の表に、2コンピュート・ノード・クラスタ用のストレージ上の異なるボリュームと、それらをマウントする場所を示します。

ボリューム (ローカル・ディレクトリ・パス)	説明	マウントする場所
1. /product/fmw	バイナリ・インストール	ノード1、ノード2
2. /mserver	管理対象サーバー・ドメイン・ホーム	ノード1
3. /mserver	管理対象サーバー・ドメイン・ホーム	ノード2
4. /aserver	管理サーバー・ホーム	ノード1
5. /nodemanager	すべてのノード・マネージャ	ノード1、ノード2
6. /shared	共有アーティファクト	ノード1、ノード2

以下に、Linuxシステムにマウントするためのマウント・コマンドの例を示します。

すべてのノードに同じバイナリ・ディレクトリをマウントする

```
Node1$ mount storage:/export/binaries /u01/app/oracle/product/fmw -t nfs4
```

```
Node2$ mount storage:/export/binaries /u01/app/oracle/product/fmw -t nfs4
```

各ノードに個別の管理対象サーバー・ディレクトリをマウントする

```
Node1$ mount storage:/export/mserver1 /u01/app/oracle/admin/mserver -t nfs4
```

```
Node2$ mount storage:/export/mserver2 /u01/app/oracle/admin/mserver -t nfs4
```

各ノードにノード・マネージャ・ディレクトリをマウントする

```
Node1$ mount storage:/export/nodemgr /u01/app/oracle/admin/nodemanager -t nfs4
```

```
Node2$ mount storage:/export/nodemgr /u01/app/oracle/admin/nodemanager -t nfs4
```

1つのノードに管理ディレクトリをマウントする

```
Node1$ mount storage:/export/aserver /u01/app/oracle/admin/aserver -t nfs4
```

仮想IP

仮想IPの主要な目的は、特定のサーバーのフェイルオーバーを可能にすることです。たとえば、管理サーバーのフローティングIPを構成することにより、管理サーバーを任意のホストで起動できるようになります。管理サーバーのクライアントは、依然として、同じアドレスでサーバーにアクセスできます。

自動サーバー移行に参加するサーバーのフローティングIPも構成します。

アプリケーションのインストール

すべてのソフトウェアを1つのノードからプロビジョニングできます。まず、WebLogic Serverバイナリだけをインストールします。

WebLogic Serverのフル・インストールのインストール

インストール先は、ORACLE_BASE/product/fmw/fmw_installである必要があります。ここで、fmw_installは、この特定のインストール用にユーザーが指定する名前です。これ以降、このインストール・ディレクトリをFMW_HOMEと表記します。

WeblogicホームおよびCoherenceバイナリのデフォルトの保存場所を受け入れます。

他の任意のアプリケーション・バイナリのインストール

Oracle FMWバイナリは、WebLogicと同じディレクトリにインストールできます。

APP_HOMEの場所は、FMW_HOME/App1などです。1つのインストールが複数のマシン上の複数のドメインをサポートできることが前提となっています。

ドメインの作成

ドメインは、\$ORACLE_BASE/admin/aserversに作成する必要があります。アプリケーション・ホームは、\$ORACLE_BASE/admin/aservers/applicationsとして指定する必要があります。

注：applicationsフォルダをaserverボリューム内に保持することは重要です。このボリュームはグループとしてフェイルオーバーするため、applicationsフォルダは同じボリューム内に保持する必要があります。

クラスタに含まれるすべての物理マシン用のマシンを作成します。複数の管理対象サーバーを作成する場合は、それらのリスニング・ポートがすべて異なっていることを確認してください。これにより、マシン間での管理対象サーバーの移動が簡素化されます。

管理対象サーバー・ドメイン・ホームの作成

管理対象サーバー・ドメイン・ホームについては、パック/アンパック (pack/unpack) ユーティリティを使用して、ローカル・マシンと後続のすべてのマシンの管理対象サーバーを作成する必要があります。

1. ノード1でpackを使用してアーカイブを作成します。

```
$ $FMW_HOME/wlserver_10.3/common/bin/pack.sh -  
domain=$ORACLE_BASE/admin/aservers/mydomain_name -  
template=$ORACLE_BASE/admin/shared/stage/mydomain_template.  
jar -template_name="mydomain_template" -managed=true
```

2. ローカル・ノードと後続のすべてのノードでunpackを使用して管理対象サーバー・ドメインを作成します。

```
$ $FMW_HOME/wlserver_10.3/common/bin/unpack.sh -  
template=$ORACLE_BASE/admin/shared/stage/mydomain_template.  
jar -domain=$ORACLE_BASE/admin/mservers/mydomain_name -
```

構成ウィザードを使用してドメインがさらに拡張されている場合は、これらの手順を繰り返す必要があります。必ず、ドメイン内のすべてのマシンについて、この手順を実行してください。

上記の手順2のunpackコマンドにoverwrite_domain=trueというパラメータを追加しなければならない場合があることに注意してください。これは、ドメイン・ホームがその場所にすでに存在する場合に必要なになります。

ポスト・ドメインの作成

JMS永続ストア

管理対象サーバーがJMSメッセージを保留状態にするためにJMS永続ストアを必要とする場合は、このストアを共有保存場所として設定して、リカバリされたサーバーが同じ場所にある永続ファイルにアクセスできるようにすることが可能です。このストアの場所は、ドメイン内のすべてのマシンについて\$ORACLE_BASE/admin/shared/clusters/domain_name/jmsと設定できます。

このディレクトリの場所は、管理コンソールの左側のペインで「Services」→「Persistence Store」をクリックし、個別の永続ストアを選択することによって、個別の管理対象サーバーごとに設定できます。

トランザクション・リカバリ用の永続ストア

一部のアプリケーションについては、各サーバーが、コミットされたトランザクションとコミットされていないトランザクションに関する情報をトランザクション・ログに格納することもできます。これは、リカバリ時に、進行中のトランザクションをクリーンアップするために使用できます。

このログも、共有ストレージの保存場所に格納する必要があります。このログの保存場所は、ドメイン内のすべてのマシンについて \$ORACLE_BASE/admin/shared/clusters/domain_name/tlogs と設定できます。この保存場所を設定するには、個別の各サーバーをクリックし、「Services」タブをクリックして、そのページの Default Store フィールドに場所を入力します。

ノード・マネージャの構成および起動

ノード・マネージャは、異なるマシンから実行し、一連の同じバイナリおよび構成ファイルを共有できます。ただし、以下のような理由から、ノード・マネージャの構成およびログ・ファイルを個別に用意することにも十分な理由があります。

- ノード・マネージャと管理サーバーの間の信頼される通信を設定する場合、各ノード・マネージャは、そのノード・マネージャ用の証明書が必要とします。
- 自動サーバー移行を構成する場合、ノード・マネージャは、異なるローカル・ネットワーク・インタフェースによって構成する必要がある場合があります。
- 各ノード・マネージャは、そのノード・マネージャ用のロック・ファイルとログ・ファイルを作成しますが、どのログ・ファイルがどのノード・マネージャに属しているのかを簡単に確認できる名前付けスキームがありません。

共有ノード・マネージャ構成では、ノード・マネージャの起動時に共有の nodemanager.properties ファイルの値を起動パラメータでオーバーライドすることによって、これらの問題を解決できます。

このプロセスを簡素化して混乱を防ぐために、物理ホストごとに個別の StartNodeManager.sh ファイルを作成し、物理ホストごとに個別のノード・マネージャ・ホームを作成することを推奨します。以下の項では、これを実行する方法を示します。

ノード・マネージャ・ディレクトリの作成

このマシンのノード・マネージャ用の構成ファイルを格納するディレクトリを作成します。以下の例では、node_name は、ノード・マネージャのリスニング・アドレスです。これは、通常、マシンの物理ホスト名です。

/common ディレクトリは、このマシンのノード・マネージャ・ホームです。/bin ディレクトリには、このマシンのノード・マネージャ用の起動スクリプトが格納されます。

```
$ mkdir -p $ORACLE_BASE/admin/nodemanager/node_name
$ mkdir $ORACLE_BASE/admin/nodemanager/node_name/bin
$ mkdir $ORACLE_BASE/admin/nodemanager/node_name/common
```

デフォルト・ファイルのコピー

/common ディレクトリは、新しいノード・マネージャ・ホームであり、ノード・マネージャのすべてのログ・ファイルと構成ファイルが格納されます。デフォルトでは、ノード・マネージャ・ホームは、\$FMW_HOME/wlserver_10.3/common/nodemanager です。この時点では、このディレクトリにあるファイルは、nodemanager.domains のみのはずです。このファイルをコピーします。

```
$ cp $FMW_HOME/wlserver_10.3/common/nodemanager/*  
$ORACLE_BASE/admin/nodemanager/node_name/common
```

また、後で、各マシンのノード・マネージャ用の起動スクリプトもカスタマイズします。そのため、このファイルもコピーします。

```
$ cp $FMW_HOME/wlserver_10.3/server/bin/startNodeManager.sh  
$ORACLE_BASE/admin/nodemanager/node_name/bin
```

Windowsの場合：

```
$FMW_HOME/wlserver_10.3/server/bin/installNodeMgrSvc.cmd  
とuninstallNodeMgrSvc.cmd  
を  
$ORACLE_BASE/admin/nodemanager/node_name/binにコピーします。
```

起動スクリプトの編集

コピーしたstartNodeManager.shファイルを編集します。次の行を探して、

```
NODEMGR_HOME="$ {WL_HOME} /common/nodemanager "
```

を

```
NODEMGR_HOME="$ORACLE_BASE/admin/nodemanager/node_name/common"に修正してくだ  
さい。
```

\$ORACLE_BASEの部分は、完全なディレクトリ名に置き換えてください。

Windowsの場合：

installNodeMgrSvc.cmdを編集して次の行を追加します。

```
set
```

```
NODEMGR_HOME=$ORACLE_BASE/admin/nodemanager/node_name/com
```

```
mon
```

ノード・マネージャの起動

ここで、この少し修正したスクリプトを使用して、このマシンのノード・マネージャを起動します。

```
$ cd $ORACLE_BASE/admin/nodemanager/node_name/bin  
$ ./startNodeManager.sh
```

\$ORACLE_BASE/admin/nodemanager/node_name/common ディレクトリには、この時点で、このノード・マネージャの構成ファイルとログ・ファイルが格納されているはずです。

Windows の場合：

次のファイルを実行します。

```
$ORACLE_BASE/admin/nodemanager/node_name/bin/installNodeMgrSvc.cmd
```

サービス・パネルを使用してサービスを起動します。

HTTP Server の構成

1つ以上の Oracle HTTP Server で構成される Web 層も、共有ストレージにインストールし、異なるマシンで実行できます。バイナリ・インストールは一度だけ実行されます。構成ツールを実行して、このホームから実行される個別の HTTP Server ごとに新しいインスタンスを作成します。

HTTP Server のインストール

HTTP Server のインストールを実行し、バイナリを \$ORACLE_BASE/product/fmw/fmw_name/web にインストールします。

この保存場所は、Web 層インストール用の Oracle ホームです。

インスタンスを作成する場合は、デフォルトのディレクトリ・パスを編集し、admin ディレクトリ・ツリーにインスタンスを作成します。

```
$ORACLE_BASE/admin/shared/instances/instance1
```

作成されるコンポーネント (ohs など) にはすべて、インスタンス番号と同じ接尾辞が付いている必要があります。たとえば、instance1 には ohs1 が、instance2 には ohs2 (以降も同様) が格納されます。インスタンスだけではなく各コンポーネントも *ドメイン内で一意の名前* を持つことが重要です。これは、後で、Enterprise Manager によるこれらのコンポーネントの管理を容易にするためです。

スケールアウト：新しいホストのプロビジョニングと管理対象サーバーの再配置

この項では、共有ストレージ構成を新しいホストに拡張し、管理対象サーバーを手動で再配置する場合の考慮事項を順番に説明します。

これらの手順により、2つ目のホストをプロビジョニングし、将来、任意の追加ホストをプロビジョニングすることができます。

新しいホストへの共有ディレクトリのマウント

共有ボリュームは、クラスタ内のすべてのクライアント・マシンによってまったく同じディレクトリ・パスにマウントされる必要があります。

多数のWebLogicファイルに多数のターゲットのフルパスが組み込まれ、共有ファイル・システムが異なるマウント・ポイントにマウントされると、予期しない動作が発生する場合があります。

beahomelistへのホームの追加

新しいホストで、beahomelistを作成し（まだ存在しない場合）、ミドルウェア・ホーム・ディレクトリを追加します。このファイルは、WebLogicインストールによって、このマシン上の既存のインストーラを検出するために使用されます。

```
$ mkdir ~/bea
$ touch ~/bea/beahomelist
```

新しいノード・マネージャ・ディレクトリの作成

最初のマシンで行ったように、この新しいマシンのノード・マネージャ・ホームを作成する必要があります。

```
$ cd $ORACLE_BASE/admin/nodemanager
$ mkdir node2
$ mkdir ./node2/common
$ mkdir ./node2/bin
```

nodemanager.domainsファイルを最初のマシンのディレクトリからコピーします。

```
$ cp
$ORACLE_BASE/admin/nodemanager/node1/common/nodemanager.domain
s $ORACLE_BASE/admin/nodemanager/node2/common
```

元の起動ファイルをコピーします。

```
$ cp $MW_HOME/wlserver_10.3/server/bin/startNodeManager.sh
$ORACLE_BASE/admin/nodemanager/node2/bin
```

さらに、このファイルを編集して、NODEMGR_HOMEを\$ORACLE_BASE/admin/nodemanager/node2/commonに設定します。

最後に、ノード・マネージャを起動します。

```
$ ./startNodeManager.sh &
```

管理対象サーバーの再配置

すべてのサーバーの仮想IPを構成することは可能ですが、管理対象サーバーのリスニング・アドレスおよび関連付けられたマシンを手動で変更することによって、管理対象サーバーをあるマシンから別のマシンに再配置することもできます。

1. コンソールを使用して、管理対象サーバー2を停止します。
2. 新しいマシン（ノード2）を作成し、そのノード・マネージャのリスニング・アドレスをノード2の物理アドレスに設定します。
3. 管理対象サーバー2を編集します。マシンをノード2に変更し、リスニング・アドレスをノード2に変更してください。
4. すべての変更を保存し、アクティブ化します。
5. 管理対象サーバー2を起動します。

これで、ノード2の新しいノード・マネージャによって管理対象サーバーが制御されるようになります。

HTTP構成ファイルで管理対象サーバー2の新しい場所が指定されていることを確認してください。これは、この管理対象サーバーがWebLogicClusterパラメータのリストに実際に示されている場合にのみ必要です。それ以外の場合は、クラスタが、再配置されたクラスタ・メンバーを動的に検出します。

追加のホストおよびサーバーのプロビジョニング

追加のホストおよび新しい管理対象サーバーをプロビジョニングするには、以下の手順を実行します。

1. Cloneコマンドを使用して（「Environment」→「Servers」を選択し、「Server」、「Clone」の順に選択）、既存の管理対象サーバーから追加のサーバーを作成します。
2. 上記の手順を実行して、新しいノード・マネージャをプロビジョニングし、新しいサーバーのリスニング・アドレスを正しく設定します。

追加のHTTP Serverのプロビジョニング

HTTP ServerなどのWeb層コンポーネントの新しいインスタンスは、\$WEB_HOME/bin/config.shにある構成ツールを使用して構成する必要があります。ここで、WEB_HOMEは通常、次のディレクトリの下にあります。

```
$ORACLE_BASE/product/fmw/web
```

追加のインスタンスは、\$ORACLE_BASE/admin/instancesの下にある既存のWeb層構成ホームにインストールする必要があります。新しいコンポーネントがドメイン内の他のコンポーネントと同じ名前になっていないことを確認してください。

管理サーバーのフェイルオーバー

管理サーバーのフェイルオーバーを利用する場合は、仮想 IP を使用して、管理サーバーを必要とするすべてのアプリケーションに対してその場所を示すアドレスが変わらないようにすることを推奨します。

管理サーバーを再配置するには、管理サーバーを停止し、その仮想IPを再配置してから、再起動します。

1. 最初のマシン上の管理サーバーを停止します。
2. IPを再配置します。Linuxの場合：
 - a. NODE1> /sbin/ifconfig ethX:Y down
 - b. NODE2> /sbin/ifconfig <interface:index>
<IP_Address> netmask <netmask>
3. 2つ目のマシンで管理サーバーを起動します。

高度なノード・マネージャ構成

ノード・マネージャを分離するメリットは、それらの構成を個別に管理できることです。この項では、ノード・マネージャを構成する際のいくつかの警告および注意事項を示します。

ホスト名の検証

ノード・マネージャは、ホスト名を検証するために構成できます。サーバーでホスト名検証が有効になっている場合、交換される証明書に含まれるホスト名が一致している必要があります。これを構成する手順（共有キーストアの生成など）は、他の場所（Fusion Middlewareの管理ガイドなど）に記載されています。

このトポロジには、いくつかの考慮事項があります。

1. 証明書は、ホスト名検証を使用するすべてのホスト名および仮想ホスト名について作成する必要があります。
2. アイデンティティ・キーストアは、環境内のすべてのメンバーによって共有され、ドメインには依存しません。このため、アイデンティティ・キーストアは、次の共有される保存場所に格納する必要があります。
\$ORACLE_BASE/admin/shared/certs
3. トラスト・キーストアも、すべてのメンバーによって共有され、ドメインには依存しません。このため、やはり、\$ORACLE_BASE/admin/shared/certsディレクトリに格納する必要があります。
4. CustomIdentityAliasを各ノード・マネージャに追加する際には注意が必要です。各ノード・マネージャは、その証明書に割り当てられたホスト名にマッピングされる、nodemanager.propertiesファイルで指定された一意のCustomIdentityAliasを持つ必要があります。

サーバー移行

サーバー移行では、管理対象サーバーが、障害発生時に別のホスト上で再起動するように構成されます。この構成の場合、フェイルオーバーの必要なサーバーは、WebLogic Serverの移行によって自動的にフェイルオーバーされる特定のフローティングIPアドレスをリスニングします。

サーバー移行の手順は、このホワイト・ペーパーの対象外です。ここでも、共有トポロジに固有のいくつかの考慮事項のみを示します。

1. サーバー移行に関係するすべてのノード・マネージャについて、ネットワーク・インタフェース・プロパティ（インタフェース、ネットマスクなど）を設定する必要があります。これらは、関係するすべてのノード・マネージャの個別の `nodemanager.properties` ファイルで設定してください。
2. 各マシンの `$PATH` 変数は、その特定のマシン用の `nodemanager.domains` ファイルを指している必要があります。すべてのノード・マネージャが、同じ `wlscontrol` および `wlsifconfig` スクリプトを共有できます。
3. 障害の発生したサーバーは、同じディレクトリを使用して再起動します。クラッシュ解析のために古いログ・ファイルが必要な場合は、これらのファイルが、起動ログが書き込まれるとすぐにアーカイブされることに注意してください。

新しいドメインの追加

Oracle FMW 11g では、新しいドメインを作成するために構成ウィザードを実行すると、ノード・マネージャ・エントリがデフォルトのノード・マネージャ・ホームに追加されるだけです。新しいドメインを作成する場合は、正しいドメイン・エントリを、そのドメインに含まれるすべてのノード・マネージャに手動で追加する必要があります。これには、`nodemanager.domains` ファイルを編集して、エントリを次の形式で追加する必要があります。

```
Domain_name=Domain_home
```

ここで、`Domain_home` は、ドメイン・ディレクトリのフルパスです。

例：1つの共有ドメイン・ホーム

このホワイト・ペーパーでこれまで説明してきたトポロジでは、1つの管理サーバーを作成し、その後、クラスタ内の各マシンについて管理対象サーバーを作成しました。

Oracle WebLogic は、クラスタのすべてのメンバーによって共有され、クラスタ環境全体を実行するために使用される1つのドメイン・ホームを使用できます。このトポロジには、新しいマシンを迅速にプロビジョニングでき、管理対象サーバーをマシン間でより容易に移動できるというメリットがあります。デメリットは、クラスタ全体のシングル・ポイント障害となることです。このドメイン・ホームは頻繁にバックアップする必要があり、それでもなお、クラスタの停止時間の発生を想定する必要があります。

これを実行する手順は、順番に示した例の手順と似ています。相違する点と固有の考慮事項は、このホワイト・ペーパーの対象外です。

1つのドメイン・ホームのみの作成

`aserver` ドメイン・ホームを作成してから管理対象サーバー・ホームを作成する代わりに、構成ウィザードを使用して、共有ストレージ・ボリューム上に1つのドメイン・ホームを作成します。パック/アンパック・ユーティリティは、使用しないでください。このドメイン・ホームは、ストレージ・サーバー・スナップショットを使用するか、手動で、頻繁にバックアップする必要があります。

すべてのマシンによる同じドメイン・ホームのマウント

このドメイン・ホームは、共有ストレージ上のボリュームに配置されます。この同じボリュームが、クラスタ内のすべてのマシンによってマウントされます。

管理サーバーの可用性

この1つのドメイン・ホームは、すでに管理サーバー・ホームです。1つのマシンに障害が発生した場合、管理サーバーの仮想IPを別のマシンに移行し、そのマシンで管理サーバーをすぐに起動できます。

管理対象サーバーの移行

管理対象サーバーは、以下の手順で、異なるマシンに移行できます。

1. サーバーを停止します。
2. 管理サーバー・コントロール・パネルで、関連付けられたマシンとそれらのリスニング・アドレスを変更します。
3. サーバーを起動します。

複数のドメイン・ホームがある場合でも、上記の方法で管理対象サーバーを移行できます。1つのドメイン・ホームのメリットは、管理対象サーバーが、同じサーバー・ディレクトリ（そのログ・ファイルと、サーバー起動スクリプトに加えたすべてのカスタマイズが含まれる）を保持することです。

新しいマシンの追加

1つのドメイン・ホームのモデルでは、新しいマシンのプロビジョニングが簡素化されます。新しいマシンをプロビジョニングする手順は、以下のとおりです。

1. コンソールを使用して、新しいマシンを作成し、その新しいマシン上で実行するすべてのサーバーを事前に作成します。これが既存のクラスタである場合は、Cloneコマンドを使用して、クラスタ内の既存のサーバーをクローンし、リスニング・アドレスとマシンを変更することで、この手順を簡単に実行できます。
2. バイナリとドメイン・ホームを含むすべてのボリュームをマウントします。
3. このマシン用の新しいノード・マネージャをプロビジョニングして起動します。
4. この新しいマシン用のサーバーを起動します。

カスタム・アプリケーションのベスト・プラクティス

任意のカスタム・アプリケーションを、共有ストレージ上の環境にデプロイできます。その際、念頭に置くべきいくつかの考慮事項とベスト・プラクティスがあります。

アプリケーションのデプロイ

アプリケーションは、EARファイルとしてサーバーにデプロイされるか、展開ディレクトリとして読み取られます。このアーカイブは、クラスタのすべてのメンバーが使用できる共有ディレクトリ（`$ORACLE_BASE/admin/shared/deploy`など）に配置できます。

共有サービス環境では、アプリケーションをすべてNOSTAGEとしてデプロイする必要があります。これは、管理サーバーへのデプロイのデフォルトですが、管理対象サーバー・クラスタへのデプロイのデフォルトはSTAGEです。NOSTAGEでは、クラスタのすべてのメンバーがデプロイを開始できる場所にアプリケーション・ファイルが配置されます。

アプリケーションは、展開アーカイブとしてデプロイすることもできます。これらも同様に共有ディレクトリにデプロイし、NOSTAGEとしてクラスタのすべてのメンバーがアクセスできるようにすることが可能です。この場合、ディレクトリの場所は、アーカイブが展開されるデプロイヤに提供されます。各メンバーは、同じアーカイブ・ディレクトリから操作します。異なるメンバーが同じファイルにアクセスすることによって発生する可能性のある同時実行性の問題は、アプリケーション自体で解決する必要があります。

ただし、同時実行性の問題は、この後の数項で説明するいくつかのベスト・プラクティスで最小限に抑えることができます。

構成およびステート管理

アプリケーションを正常に管理するには、インスタンス固有の構成と、アプリケーションの異なるインスタンスの共有構成を分離して、競合を回避する必要があります。

共有構成

共有構成は、クラスタのすべてのメンバーによって読み取られ、更新される必要のある構成です。共有構成は、データベースに含めるか、1つ以上の構成ファイルとして共有ディスク上で管理できます。

共有構成ファイルのもっとも一般的な使用例は、起動するたびにクラスタの各メンバーによって読み取られるパラメータが含まれる初期化ファイルです。共有ファイルは、同時実行性の問題を最小限に抑えるために、大部分が読み取り専用であり、まれに更新されるだけである必要があります。それ以外の場合は、トランザクション・ソース（データベースなど）を共有データ用として考慮するか、ファイルまたはファイルのセクションへの同時アクセスのために必要なロジックをアプリケーションに組み込む必要があります。

ここでは、共有構成ファイルを管理するための3つのオプションと、それぞれのベスト・プラクティスについて説明します。

1. 展開されたデプロイメントに含まれる更新可能な構成ファイル

これは、容易に実装できるオプションです。すべてのサーバーが展開されたデプロイメントを共有し、そのデプロイメント内の構成ファイルの読み取り/書き込みを実行します。このオプションは、一般には推奨されません。

この初期デプロイメント・ファイルは、静的な、読み取り専用のファイルと考える必要があります。どのような形であれデプロイメント・ファイルを更新すると、アプリケーションの長期にわたるメンテナンス（バージョン管理、パッチ適用、バックアップ）が非常に困難なものになります。アプリケーションの再デプロイによって、構成全体が上書きされる可能性もあります。

2. config ディレクトリをターゲットとする構成ファイル

共通構成ディレクトリをアプリケーションによって指定できます。これは、たとえば、クラスタのすべてのメンバーがアクセスできる任意の絶対パスである場合があります。構成ディレクトリは固定され、クラスタのすべてのメンバーがアクセスできます。これは、デプロイメントに依存しないため、アプリケーションの再デプロイの影響を受けません。

3. 別の構成ファイルにリダイレクトされる、ユーザーが編集可能な構成ファイル

これは、もう一つの間接化の方法です。構成ファイルには構成情報が含まれませんが、実際の構成ファイルの場所を示すポインタが含まれます。ユーザーは、このファイルを編集して、実際の構成ファイルを再配置できます。

この方法のメリットは、構成をどこにでも配置できることであり、動的な場合には、ドメイン・ディレクトリの外部に配置することもできます。たとえば、

`$ORACLE_BASE/admin/shared/clusters/domain_name`に配置できます。

個別の構成

クラスタ内のサーバーは、クラスタの他のメンバーによる使用を意図しない個別の構成またはプライベート・データを必要とする場合があります。この構成の保存場所は明確にパーティション化して、複数のクラスタ・メンバーのすべてが同じディレクトリ構造を共有できるようにする必要があります。

たとえば、クラスタのメンバーは、その構成を、

`$DOMAIN_HOME/config/configfile.xml`ではなく、

`$DOMAIN_HOME/config/server_name/configfile.xml`に格納する必要があります。これにより、同じ種類の複数のサーバーを同じドメイン・ホーム内に共存させることが可能になります。

初期構成後に構成情報が変更されることが予期される場合は、管理対象サーバーを再配置しても常に管理対象サーバーからアクセスできる場所に構成を配置する必要があります。このために、たとえば、JDBCデータソースを使用できます。

実行中のサーバーの名前は、Mbeanとしてアプリケーションに提供されます。これは、データベースにおける出力ディレクトリ名または必要なテーブルの作成に使用されます。

クラスタ・ディレクトリ

クラスタ・ディレクトリは、クラスタのすべてのメンバーが共有する必要があるアーティファクトまたはクラスタの他のメンバーが所有およびアクセスできるように提供される必要のあるアーティファクト用に提供されます。これらのディレクトリを永続ストア用に使用することもできます。

JMS永続ストアの設定

管理コンソールで「**Services**」 → 「**Persistence Stores**」を選択すると表示されるパネルで構成できます。

ドメイン内のクラスタによって使用されるすべての永続ストアの場所を、次のように入力します。

`$ORACLE_BASE/admin/shared/clusters/domain_name/jms`

デフォルトの永続ストアの設定

管理コンソールを開き、「Environment」→「Servers」を選択して、サーバー名を選択します。「Services」タブをクリックします。Default Store フィールドに、次のように入力します。

```
$ORACLE_BASE/admin/shared/clusters/domain_name/tlogs
```

これにより、クラスタ内の他のサーバーがトランザクション・リカバリに使用できる共有ディレクトリに、個別のサーバー・ログが配置されます。

動的構成

クラスタのすべてのメンバーが必要とする動的データは、できれば、データベースに含めてください。それが不可能な場合は、動的データをクラスタのサブディレクトリに含めてください。たとえば、Oracle Universal Content Management (Oracle UCM) は、

```
$ORACLE_BASE/admin/shared/clusters/domain_name/ucmvault
```

に含めることができる共有ボールド・ファイルを必要とします。

一元化されたログ

すべてのサーバーおよびアプリケーション・ログを1つのディレクトリに一元化すると便利な場合があります。これは、次のようないくつかの理由によります。

1. 診断する場所の一元化

ログを一元化することにより、すべてのログを1つのディレクトリまたは共通の保存場所で調べることができるため、システムで発生する問題の診断プロセスが簡素化されます。

2. ドメイン・ディレクトリからの動的ファイルの分離

ログ・ファイルは、ドメイン・ディレクトリ内でもっとも動的なファイルです。これらのファイルを分離することにより、これらのアーティファクトに関して異なるバックアップ戦略を使用することが可能になります。

3. 複数のドメイン・ホームのサポート

一元化されたログは、管理対象サーバーを別のドメイン・ディレクトリに再配置しなければならない可能性がある場合に便利です。管理対象サーバーが新しいドメイン・ホームに再配置されても、ログの保存先は変更されません。

管理対象サーバー用のデフォルトのログの保存場所を変更するには、

「Environment」→「Servers」を選択して、管理対象サーバーを選択し、「Logging」タブをクリックします。新しいログの保存場所を入力します。たとえば、`$ORACLE_BASE/admin/shared/logs` とすることができます。

ライフ・サイクルのベスト・プラクティス

これまでの項では、インストール、構成、アプリケーションのデプロイについて説明しました。この項では、共有ストレージ・インストールのライフ・サイクルの管理に関する推奨事項を説明します。

複数のボリューム

すべてのバイナリ、アプリケーション、および構成のソースが単一の場合は、アプリケーション・エラー、ユーザー・エラー、またはその他のハードウェア/ソフトウェア障害によって環境が損なわれる可能性があるという問題が発生します。1つの物理マシンでエラーが発生すると、共有環境の特性により、すべてのマシンで問題が発生する可能性があります。

このリスクを最小限に抑える一つの方法として、ストレージ・サーバー自体の機能を使用できます。最新のストレージ・サーバーは、ボリュームをアーカイブしたり、ボリュームのスナップショットを作成したりして新しいボリュームを作成する機能を備えています。

管理のためには、共有ボリュームごとに異なるポリシーを実装することも有用です。

ボリューム1：バイナリ

ディレクトリ・ツリー：\$ORACLE_BASE/fmw/

このボリュームは、アプリケーション・バイナリで構成されます。これらは、静的であり、大部分が読み取り専用である必要があるため、頻繁なバックアップを必要としません。このボリュームは、少なくとも、新しい製品がFMW_HOMEにインストールされるか、バイナリのパッチ適用または更新が実行されるたびに、バックアップ/レプリケーションされる必要があります。

ボリューム2：共有構成

ディレクトリ・ツリー：\$ORACLE_BASE/admin/shared

このツリーのクラスタ・ディレクトリは、ディスク上のほとんどのアーカイブ・データの保存場所であり、異なるクラスタ・メンバーによって読み取り/書き込みが実行されます。データの重要性は、アプリケーションによって異なります。共有データが一時的なものでしかない場合もありますし、ここにあるデータが正常なリカバリのために不可欠である場合もあります。このディレクトリ・ツリーは、他の2つのボリュームよりも頻繁にバックアップされる必要があります。

ボリューム3：管理サーバー・ドメイン・ホーム

ディレクトリ・ツリー：\$ORACLE_BASE/admin/aservers

これには、どのインストールでも最も重要なドメイン・ホームが含まれます。このディレクトリ・ツリーは、通常の実行時にはあまり変更されませんが、ドメイン構成の変更、新しいドメインまたはアプリケーションの追加、その他の実行時構成の変更といった管理操作時に変更されます。

このディレクトリ・ツリーは、少なくとも、何らかの管理操作を実行した後はバックアップする必要があります。

バックアップとリカバリ

バックアップ戦略では、前述のように、異なるアーティファクトの異なる変動性を考慮に入れる必要があります。ストレージ・サーバーによってアーカイブ・ボリュームのクローンまたはコピーを作成でき、これらをバックアップとして使用できます。

バイナリ・ボリュームと構成ボリュームについては、システムの初期構成の直後にバックアップする必要があります。これらのボリュームのいずれかが壊れた場合は、プライマリ・ボリュームをアンマウントし、バックアップ・ボリュームをマウントできます。

バックアップは、ボリューム間でのビット単位のコピーによって作成しないように注意してください。この方法では、ファイル権限が正しく設定されていないためにエラーが発生する可能性があります。バックアップは、サーバー側のメカニズムを使用して作成してください。

パッチ適用とアップグレード

パッチ適用およびアップグレード時の停止時間を最小限に抑えるために、複数のボリューム・ソリューションを使用して、アウトオブプレース・パッチ適用を実行することもできます。このシナリオでは、以下の手順を実行します。

1. ストレージ・ボリュームをコピーして、新しいボリュームを作成します。
2. コピーしたボリュームにパッチを適用します。
3. サーバーを停止します。ローリング・パッチ適用シナリオの場合は、これが一度に1つのマシンで実行されます。サーバーとノード・マネージャを含むそのマシンでのすべてのプロセスが停止します。
4. 古いボリュームがアンマウントされます。
5. 新しいボリュームがマウントされます。
6. すべてのプロセスが再起動されます。

共有ストレージのトラブルシューティング

ローカル・デバイスや専用デバイスではなく共有デバイス上にOracle WebLogicをデプロイして実行する場合は、注意すべきいくつかの追加の考慮事項があります。これらの一般的な問題のいくつかについて、ここで説明します。

認証および権限に関する問題

共有ストレージを管理するストレージ・サーバーは、独自のユーザーと認証を持つ独立したデバイスです。ローカル・ユーザーおよびグループと権限のデバイスへのマッピングや、マシン間でのマッピング方法は、正しく構成する必要があります。これは、ストレージ・プロトコル (NFS v3、NFS v4、CIFSなど) によっても異なります。

ID管理およびロール・マッピング

アクセスを管理するには、リモート・ファイル・システムのマウントおよび読み取り/書き込みと、権限の正常なマッピングを実行できる必要があります。これらの権限は、通常、ストレージ・サーバー固有であり、このホワイト・ペーパーの対象外です。

IDを管理するには、複数のクライアント・マシンとストレージ・サーバーで、適切な読取り/書き込み権限を異なる種類のユーザーに割り当てることができる必要があります。NFS v3では、クライアント・マシンおよびストレージ・サーバー上のユーザーをUIDだけでマッピングできます。このデフォルトのマッピングでは、このストレージ・サーバーを使用するすべてのユーザーの間でUIDに一貫性があることを確認する必要があります。これは、手動で、またはNISを使用して実行できます。このルール唯一的例外は、特別なuserid 0（ルート）です。ルートは、ストレージ・サーバー上のユーザーに手動でマッピングする必要があります。手動でマッピングしないと、ルートは、通常、'nobody'などのユーザーにマッピングされます。

IDを管理するために推奨される方法は、認証を外部ソース（LDAPサーバー、NISなど）またはActive Directoryに委任することです。ストレージ・サーバーをクライアント・マシンと同じディレクトリ・サービスに登録できる場合は、認証と権限をすべて一元管理できます。この構成を使用する場合は、ローカル・ユーザーがリモート・ストレージ・デバイスにアクセスできない可能性があることに注意してください。

ファイル権限

ロール・マッピングまたはディレクトリ・サービスによってユーザーが正しくマッピングされたら、次はファイル権限が透過的である必要があります。ここでは、次の2つの場合に注意する必要があります。

setuid

Fusion Middlewareのインストールでは、setuidを構成する必要はありません。ただし、前述のように、setuidに関する問題が、通常、ロール・マッピングの構成方法における問題を指し示す点は注目に値します。特に、ストレージ・サーバー上でルートに十分な権限が与えられている（通常、デフォルトでは与えられない）ことを確認してください。

拡張属性

ほとんどのクライアント・マシンおよびストレージ・サーバーは、ACLと拡張ファイル属性をサポートしています。ただし、各デバイスが拡張属性を処理する方法は一致していない可能性があります。このために、'cp -p'コマンドを実行すると、多くの場合、警告やエラーが表示されます。この問題が発生する場合、最善の選択肢は、リモート・ストレージのマウント時にACLを無効にすることです。これは、マウント・コマンドのマウント・オプションのリストにnoaclオプションを追加することによって実行できます。

一般的なファイル・ロックに関する問題

異なるストレージ・プロトコルでは、ロックの処理方法が異なります。

NFS v3

NFS v3では、ロックの所有者が消えた後もロックを維持できます。クラッシュが突然発生し、この状態になると、これらのロックを手動で解除する必要があります。

NFS v4では、ロックの所有者がロックのリースを更新しなかった場合、一定時間が経過すると、ロックが自動的に期限切れになります。

サーバーに問題が発生して再起動した場合、特にサーバーが突然停止した後は、サーバー・ログを調べることによって、ファイル・ロックに関する問題を検出できます。次のようなメッセージが、ログに表示されることがあります。

```
java.io.IOException:Error from fcntl() for file locking,  
Resource  
temporarily unavailable, errno=11
```

これらの問題は、次のいずれかの方法で、影響のあったファイルのロックを解除することによって一時的に解決できます。

1. ファイルをコピーして、ファイルの名前を変更してから、そのファイルを元の場所にコピーしなおします。たとえば、次のようなコマンドを実行します。

```
$ mv locked_file locked_file_tmp  
$ cp locked_file_tmp locked_file
```

2. ストレージ・サーバー上のロックを手動で解除します。この手順は、ストレージ・デバイスおよびプロトコルによって異なります。たとえば、CIFSでは、開いているロックは、通常、ストレージ・サーバー上のインタフェースを使用して検査および管理できます。

問題が解決しない場合、Oracle WebLogicでは、各種のアーティファクトについて、ファイル・ロックを無効にすることができます。ただし、これは、他のオプションがすべて成功しなかった場合以外は、推奨されません。

たとえば、デフォルト・ファイル・ストアのファイル・ロックを解除するには、「**Configuration**」→「**Services**」を選択し、特定のサーバーの「Default Store」セクションにある「Advanced」セクションを選択します。「**Enable File Locking**」チェック・ボックスの選択を解除してください。

同様にして、カスタム・ファイル・ストア、JMSページング・ファイル・ストア、または診断ファイル・ストアのファイル・ロックを無効にすることができます。

NFS v4

NFS v4では、ロックの所有者がロックのリースを更新しなかった場合、一定時間が経過すると、ロックが自動的に期限切れになります。

フェイルオーバーの発生後にロックが期限切れになっていない場合は、リース有効期限を短くする必要があります。このパラメータの構成手順は、使用しているストレージ・サーバーによって異なります。

たとえば、Sun ZFSサーバーでは、サーバーのNFS構成に含まれるパラメータによって設定できます。

具体的には、ZFSコンソールで、「**Configuration**」→「**Services**」→「**NFS**」を選択し、

‘Grace Period’パラメータを指定してNFSv4リース・タイムアウトを制御できます。

フェイルオーバー時間を短くする必要がある場合は、このパラメータの値を小さくすることができます。このパラメータの値を大きくすると、NFS v4の動作が、ロックが無期限に保持されるNFS v3に近くなります。

Oracle HTTP Serverのパフォーマンスとロック

Oracle HTTP Serverが、共有ストレージ上にある場合に、通常のパフォーマンスが発揮されないときは、以下の構成を試みることができます。

OHS DocumentRootを移動する

DocumentRootをローカル・ファイル・システムに移動すると、OHSがネットワーク・ファイル・システム上で実行する書込みが大幅に削減されます。これは、httpd.confファイルで変更できます。

ロックをセマフォに設定する

デフォルトでは、ソケット・シリアライズに使用されるロック・ファイルがファイル・システム上に作成されます。これらは、以下の方法で、セマフォによって置き換えることができます。

httpd.confで、以下の手順を実行します。

- AcceptMutex sysvsem行を追加し、その他のすべてのAcceptMutex行をコメント・アウトします。
- LockFileディレクティブをコメント・アウトします（表示される場合）。

ロック・ファイルをローカル・ファイル・システムに移動する

別の方法として、LockFileディレクティブを使用してロック・ファイルをローカル・ファイル・システムに移動できます。

Oracle HTTP Serverの起動時に'No Locks Available'というメッセージ（または類似のロック・エラー）が表示され、エラーが発生する場合は、上記の2つのロック・ファイル・ソリューションの一方も使用してください。

ORACLE®

共有ストレージ上のOracle WebLogic

2013年5月

著者：Richard Delval

共著者：Pradeep Bhat、Shailesh Dwivedi

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



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2011, 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。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社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。その他の名称はそれぞれの会社の商標です。

0109