

Solaris™ コンテナ - サーバの仮想化が進むことによりサービスの管理がいかに簡易化されるか

ITエグゼクティブ/管理者向け技術要約
2002年2月



目次

概要	1
対象読者	2
管理性 - IT ニーズ階層のトップ層	3
サーバの仮想化	5
Solaris コンテナ : サーバの仮想化における次のステップ	6
Solaris コンテナの発展	7
例 : Solaris コンテナの動的な特徴	8
例 : クラスタ化された環境における Solaris コンテナの使用	9
Solaris コンテナ用 Sun™ 製品の動向	12
Solaris コンテナとサービスレベル管理の将来	14
管理性 - Solaris コンテナの価値	16
まとめ	18

第 1 章

概要

大規模分散コンピューティングを 10 年ほど経験すると、企業は自社の IT インフラにスケーラビリティと可用性をいかに組み込めばよいかを理解するようになります。今日では何千ものノードにソフトウェア・コンポーネントをインストールした巨大なサーバ・ネットワークを管理するためのコストが上昇の一途をたどっており、企業の関心はパフォーマンスと可用性を拡大することから IT インフラのコストを抑え、エンドユーザ・サービス・レベルの管理性を向上させることに移行しています。この目標はサーバの仮想化を基盤とした新手法を通して達成できると Sun は考えます。

エンドユーザ向けのアプリケーション・サービスは、複数のサーバに渡って分散されたコンポーネントから成り立っていることが少なくありません。コストを抑えるために企業はアプリケーション・サーバの数を減らすことを切望していますが、アプリケーションの分離性を維持することにも気を向ける必要があります。ハードウェアとソフトウェアの発展は、サーバの仮想化という概念をもたらしました。これは、ネットワーク上のサーバを独立した実行環境として柔軟に区別化し、同一サーバ内での分離を実現するものです。サーバを仮想化すれば、データ・センターは個々のシステムで一杯になった部屋としてではなく、連係したコンピューティング・リソースから成る機構としてビジュアル化し管理できます。

サーバの仮想化における Sun の次の新手法は、Solaris™ コンテナという概念です。Solaris コンテナは、ソフトウェアによって定義された柔軟な境界を利用してソフトウェア・アプリケーションまたはサービスを分離します。この分離により、Solaris™ オペレーティング環境の同じインスタンス内で複数のアプリケーションが稼動している場合でも、各アプリケーションを個別に管理できるようになります。Solaris コンテナは、Solaris オペレーティング環境の単一のインスタンス内に実行環境を作成し、以下の状況を実現します。

- *リソースの徹底的な包囲と制御*: 予測性の高いサービス・レベルを達成する
- *障害分離*: 障害の拡散や予定外のダウンタイムを最小限に抑える
- *セキュリティ分離*: 無認可のアクセスや意図的でない侵入などを防ぐ

この環境には、細部に渡る広範なリソース追跡によりキャパシティ計画モデルや高度なクライアント課金モデルをサポートできるように、リソース利用課金を簡易化する効果もあります。

Sun は、Solaris コンテナを Solaris オペレーティング環境における基礎的な管理オブジェクトとして広く普及させ、Sun の製品ライン全般に渡って利用してもらうことを計画しています。この共通の手法により、リソース制限、障害の拡散、セキュリティなどに捕らわれずにサービス・プロビジョニングを簡易化し、少数のサーバにアプリケーションを容易に統合できるようになります。この遍在的な管理モデルではエンドツーエンドのサービスを統合された単一のオブジェクトとして管理できるため、サービスレベルの管理も簡単になります。このサービス管理環境は柔軟で動的なものです。わかりやすく、手間もさほどかかりません。

Solaris コンテナの代表的な利点を示します。

- サーバ統合とエンドツーエンドのサービスレベル管理を通して管理コストを抑えることができる
- コンテナ間でリソースが動的に再配分されるためリソースの利用率が高まる
- 複数のアプリケーションに渡る障害拡散やセキュリティ侵害が最小限に抑えられ、サービスの可用性が向上する
- ソフトウェアをベースとしたこれらのコンテナは動的に再構成できるため、柔軟性が向上する

この直感的な管理モデルを利用することで、企業はサービス・レベルを犠牲にすることなく総所有コストを大幅に減らすことができます。

対象読者

この文書は、分散型のコンピューティング環境におけるリソース管理のビジョンと手法について説明しています。対象読者としては、大規模な分散コンピューティング・ソリューションの実装に取り組んでいる組織の IT 管理者、設計者、経営陣などを想定しています。

第 2 章

管理性 - IT ニーズ階層のトップ層

何年もの間に IT システムが発展してきたように、エンドユーザや IT スタッフの期待も同様に高まってきました。技術が進歩するごとに期待も新たに生まれ、いったん達成された躍進はすぐに有難味のない標準機能と見なされます。この概念は、最新の技術革新が新開発の基盤として作用するピラミッド・モデルとして示すことができます (図 1 を参照)。

初期の情報システムは、ビジネス問題を解決するだけで成功と見なされました。プログラマ・チームによる管理 / 維持が必要かどうかは問題ではなく、ビジネス上の利点がコスト的に見合えば適切な投資だと判断されたのです。図 1 では、これを IT システムのニーズ階層における最下部の層として示しています。

IT が成熟するに連れ、システムに直接アクセスすることを認められるユーザが増え、管理者は日常業務をこなすためにコンピュータ処理に深く依存するようになります。この時点では、可用性とスケーラビリティ (ピラミッドの中間に位置する 2 つの層) が IT 管理者にとって最大の懸念事項となります。システムに可用性とスケーラビリティをもたせることで得られる正味の利益は、ダウンタイムまたは貧弱なパフォーマンスによる生産性の低下を計算することで評価されました。現在では、可用性とスケーラビリティの実装方法は十分に理解されつつあります。多層的に構成されたハードウェア・サーバ全般に渡って主要なソフトウェア・プロセスを複製することで、可用性とスケーラビリティは段階的に拡張できるのです。

現在の目標は、十分なパフォーマンスを発揮する常時利用型の IT サービスが当然だと考えるユーザを抱える中で、もっとも効果的な方法で可用性とスケーラビリティをさらに高めることです。しかし、回復力とスループットを高めるために IT アーキテクチャ全体に渡って複製されるコンポーネントが増えると、管理が困難でコストも高いシステムから成る複雑なネットワークが不規則に広がってしまいます。Sun Service Point Architecture がこれらのニーズにどのように対応できるかという詳細は、ホワイトペーパー『*Sun Service Point Architecture—Delivering Services on Demand or the Networked World*』¹ を参照してください。

1. <http://www.sun.com/software/whitepapers/wp-spa/wp-spa.pdf>



図1 IT インフラ・ニーズの階層

リソースの利用率を高め、システムの管理コストを減らせば、コストは抑えられます。最近のキャパシティ・プランニングは、時折発生するピーク負荷に対応するために余分なキャパシティを見込んでいることが少なくありません。たとえば、株式仲買業者は急激な需要の伸びが発生する場合にできるだけ多くの収益を得られるように、かつて起きた量をはるかに上回る取引量をこなすようなシステムを設計しがちです。つまり、通常取引レベルで必要となるのは利用できる全キャパシティのごく一部です。

取引システムのリソースが利用されていない時にほかのアプリケーションがそれらを借用できるようにすれば、はるかにコスト効率の良い実装が実現されます。取引量が増える時にはリソースが動的に再配分され、重要度の低いアプリケーションでは遅延が発生する可能性がありますが、主要な取引システムは十分に機能を発揮できます。このような形でリソースを共有すると管理対象となるシステムのトータル数が減り、リソース利用率の向上、総資本コストの減少、システム管理費用の減少につながります。IT オペレーションの効率を高める苦闘の結果、IT 管理者のフォーカスは管理性 (ピラミッドの最上位層) に変わりました。

サーバ統合の効果を十分に引き出すためには、各アプリケーションを独立させたまま管理できなければなりません。これには、同じサーバ上の複数のアプリケーションに渡ってリソース利用の制御、障害分離、セキュリティの管理を行う能力が求められます。つまり、サーバ内で仮想的なサーバ境界を設ける必要があります。

第3章

サーバの仮想化

個々のシステム間の物理的な境界は、問題とされなくなってきました。クライアントに配布されるサービスは、複数のサーバに渡って分散されるかあるいは単一のサーバ上の1つのパーティション内に置かれるアプリケーションを基盤とするものが増えつつあります。重要視されているのは、アプリケーション間に存在するこれらの論理的な境界です。現在では、アプリケーションで利用できるリソースは、システム・リソース (CPU、メモリー、I/O など) をアプリケーションに割り当てる管理ソフトウェアによって定義できます。実際、サーバはネットワーク化されたリソース機構における仮想領域になりつつあります。各仮想領域は、リソースについてセキュリティ境界、障害分離、および制御を論理的に定めた独立したコンピューティング環境です。この新たな柔軟性により、アプリケーションの配備方法や IT オペレーションの管理方法は変化し続けるでしょう。

何年も前に私たちは、利用のピーク時と閑散時が刻々と変わるアプリケーション・サービスを運用している場合にはそれらをひとまとめに統合し、リソースを共有した方が効率が良いということを実験から学びました。このような統合を行えば、使用頻度の低い専用ハードウェアによって各アプリケーション・サービスをサポートするのではなく、リソースを瞬時に再配分できます。この方法では、必要な時に必要な場所へリソースを移動できます。処置が適切であれば、資本設備、メンテナンス、システム管理などのコストは減少し、コンピューティング・インフラ全体の費用削減につながります。

サーバ・リソースをパーティションに分割してアプリケーションを個々に独立させたのもメインフレームが最初です。これにより顧客は、本稼動環境を保護したまま、本稼動システムに影響を与えない独立したワークスペース内でシステム・アップデートや改訂アプリケーションのテストを実施できるようになりました。Sun は、Dynamic System Domains 技術を採用した際 (1996 年) にこの概念をさらに発展させました。

Sun の Dynamic System Domain を使用した場合、システムまたはアプリケーションのエラーがほかのドメインで稼動しているアプリケーションに影響を与えないように、各ドメインはそれぞれ Solaris オペレーティング環境のコピーを実行し、障害を分離させます。これらのドメイン技術とメインフレームにおける初期の論理パーティション技術の主な違いは、ドメインは動的に分

割できるという点です。1つのドメインでリソース不足が発生すると、システムはほかのドメインからリソースを自動的に借用できます。この機能によってデータ・センターはほかのドメインの障害を切り離してセキュリティを維持し、柔軟性を飛躍的に高めることができます。

管理 / 監視が必須の個別の OS インスタンスを必要としない分離機能を取得できれば、管理効率はさらに高まります。

Solaris コンテナ: サーバの仮想化における次のステップ

現在 Sun は、Solaris コンテナを使用してサブ CPU レベルまでドメインを分割できるようにすることでサーバの仮想化というテーマをさらに発展させています。Solaris コンテナは、IT インフラの管理にまったく新しい手法をもたらす基盤としての役割を果たします。コンピューティング・リソースはアプリケーション・サービスのプロビジョニングのために個々の実行環境として柔軟に分割が可能であり、データ・センターはこれらのリソースが相互に結びついた機構として機能できます。アプリケーション環境はわずかな費用でほかのサーバ・パーティションに移すことができるため、アプリケーション・サービスの柔軟性は飛躍的に高まります。アプリケーション・サービスのプロビジョニング時には必ず独立した実行環境が用意されるため、アプリケーションを少数のサーバに整理統合することも容易です。

また、パーティショニング用の共通モデルを利用すれば、エンドツーエンド・アプリケーション・サービスのサービスレベル管理が簡易化されます。Sun は、エンドツーエンドのサービスの管理と監視が行え、かつアプリケーション・コンポーネントとエンドツーエンド・サービスの詳しいリソース利用課金が可能な高レベル・サービス管理アプリケーションの開発を予定しています。この新しい管理パラダイムは、自動化による管理コストの削減という効率の良い管理を実現して総所有コストを大幅に減らすと同時に、エンドユーザ・サービス・レベルの制御に相当な効果を発揮すると見込まれます。

Solaris コンテナはソフトウェア・コンポーネント・セットのための包括的な実行環境であり、ソフトウェア・コンポーネントからプラットフォーム・リソースに渡る仮想マッピングを提供します。Solaris コンテナによりアプリケーション・コンポーネントは、それらが単一の Solaris オペレーティング環境インスタンスを共有している場合でも独立した状態を維持できます。

Solaris コンテナは、メモリーや CPU タイムといったリソースの消費に境界を確立するとともに障害およびセキュリティの分離を可能にします。処理需要に変化がある場合 (世界的な事件が突然発生し、ニュースを中心とする Web サイトにアクセスが殺到する場合など) には、1つ以上のコンテナ境界を拡張してリソース消費の瞬時的な上昇に対応できます。管理者がリソース更新を行うか、一定の条件に達したところで自動更新が起きるように事前設定されたポリシーによって更新するかにかかわらず、この更新時には障害とセキュリティの境界が調整されます。

第 4 章

Solaris コンテナの発展

Solaris コンテナの機能は、リソース、障害、セキュリティ管理などのポリシーを設定することで実現されます。Solaris 9 以降のオペレーティング環境では、リソース境界の構築に欠かせない多数の機能が Sun のリソース管理アプリケーションに含まれる予定です。たとえば、リソース管理ツールを使用すると、単一の Solaris オペレーティング環境インスタンス内で CPU、物理メモリ、入出力帯域幅などのリソースを割り当てることができます。Solaris 9 オペレーティング環境ではこのような方法でリソース指向のコンテナが利用できるようになり、顧客はコンテナの持つ利点の多くを享受できるでしょう。この間も Sun はさらに新たな機能の開発を進めます。

Solaris コンテナが発展するに連れ、コンテナが Dynamic System Domain に類似したより精細な保護レベルを達成できるように、障害分離とセキュリティ分離が追加されます。ユーザレベルのプロセスで障害が発生すると、コンテナ境界によってほかのコンテナへの障害拡散が防止されます。Solaris コンテナには、システム管理レベルでの分離機能も予定されています。root ユーザはコンテナごとに 1 人置かれ、管理者が境界を越えることがないように役割ベースのアクセス制御によって制御できるようになります。たとえば、コンテナ A のシステム管理者はコンテナ B で何が起きているか確認できません。

高レベル・サービス管理アプリケーションが、エンドツーエンドのアプリケーション・サービスを管理 / 監視できるユーザ・インタフェースを提供します。アプリケーション・サービスがネットワーク上で複数の Solaris コンテナと複数の物理システムに渡って分散される場合でも、管理者は高レベル・サービス管理アプリケーションによって単一の構成としてサービスを確認できます。最終的には、低レベル・リソース管理アプリケーションを通して行われるコンテナ境界の確立と管理に伴う細かな作業は管理者にその存在を意識させないものとなり、高レベル・サービス管理アプリケーションが単一の管理インタフェースとなるでしょう。

例: Solaris コンテナの動的な特徴

Solaris コンテナ境界の概念は、簡略化された例で考えるとよく理解できます。図 1 では、Solaris コンテナが 3 次元の箱として描かれています。この図の 3 つの軸は、コンテナで利用できるプロセッサ・リソース、I/O (帯域幅) リソース、およびメモリー・リソースを示します。実際には、Solaris コンテナには種々の境界を多数設定でき、多次元となります。

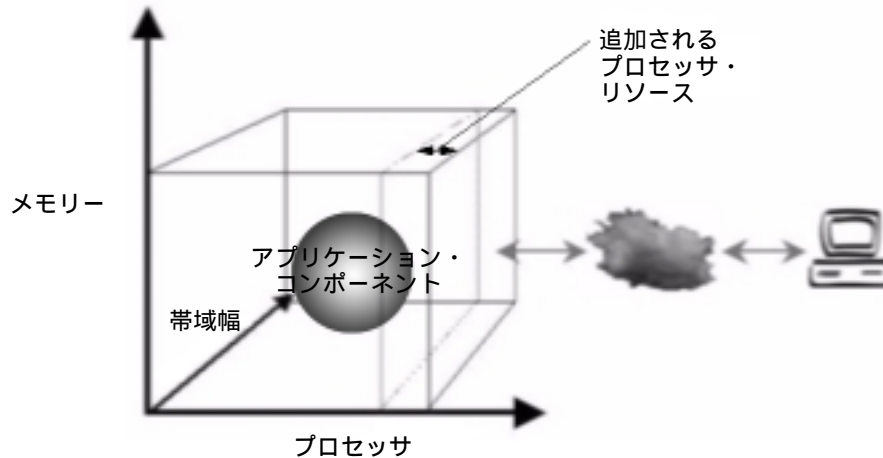


図 1 Solaris コンテナの拡張

この例では、コンテナのプロセッサ方向は、一定数のプロセッサ、または全プロセッサ・リソースのうち共有されている一定数のプロセッサを示します。使用可能なプロセッサ・リソースの拡張は、この立方体の右方向へのボリューム追加で示されています。追加されたこのコンピューティング・リソースを有効にするには、Solaris プロジェクトまたは Solaris 9 リソース管理ソフトウェアを使用して設定されているプロセッサ利用の限度を変更します。

図 1 に示されている Solaris コンテナのメモリー軸は一定量の物理メモリーであり、これも Solaris 9 リソース管理ソフトウェアで設定済みです。帯域幅軸は、ネットワーク帯域幅の一定の割合、または使用可能な帯域幅キャパシティの一部を示します。

重要なのは、コンテナの 1 つの次元を操作するとほかの次元におけるアプリケーション・サービスのリソース需要に影響を与える可能性があるということです。たとえば、あるコンポーネントがその次元のプロセッサ・シェアをすべて使用する場合には、そのコンポーネントは CPU バウンドのように見えるかもしれません。しかし、この状況はボックスの物理的なメモリー容量を増やしてメモリー・ページングによる CPU 消費を防止することで解決できるでしょう。このことはリソース管理ソフトウェアで認識され、その動的なリソース割り当て方策において考慮されるはずで

Solaris コンテナには、リソース制御、障害分離、セキュリティ以外の要件も設定できます。たとえば、可用性要件、フェイルオーバー・セマンティクス、ユーザ・プロセスの最大数などをコンテナに設定し、ワークロードおよびそのサービス可用性を自動的に制御できます。コンテナ全体に

フェイルオーバー・セマンティクスを設定すると、関連性のある複数のプロセスを同時にフェイルオーバーできるという大きなメリットが得られます。次の例は、クラスタ・フェイルオーバー環境で Solaris コンテナがいかにリソース管理を簡易化するかを示しています。

例: クラスタ化された環境における Solaris コンテナの使用

Solaris コンテナは、SunPlex™ システムで提供されるようなクラスタ環境で重要な役割を果たすでしょう。クラスタには、リソースを統合して高度なリソース・ビューを得られるというメリットがあります。クラスタは可用性の高い最適化されたリソースを持つ単一のシステムであり、アプリケーション・サービス向けのフェイルオーバー処理に対応できます。SunPlex システムは、高可用サービスをホスティングするコンテナを管理するための高可用プラットフォームとして利用できます。

図 2 は、データベース・サービス、アプリケーション・サーバ・サービス、および Web サーバ・サービスを含む 3 層のアプリケーション例です。これらのサービスは、クラスタ化された環境内の独立した 4 つのドメインに渡って階層化された 7 つの Solaris コンテナに対応付けられています。これらのドメインは SunPlex システム内の 2 台または 3 台の物理サーバに渡って分散している可能性があります。それらの物理レイアウトは適度の可用性と冗長性を確立することにしか関係がありません。

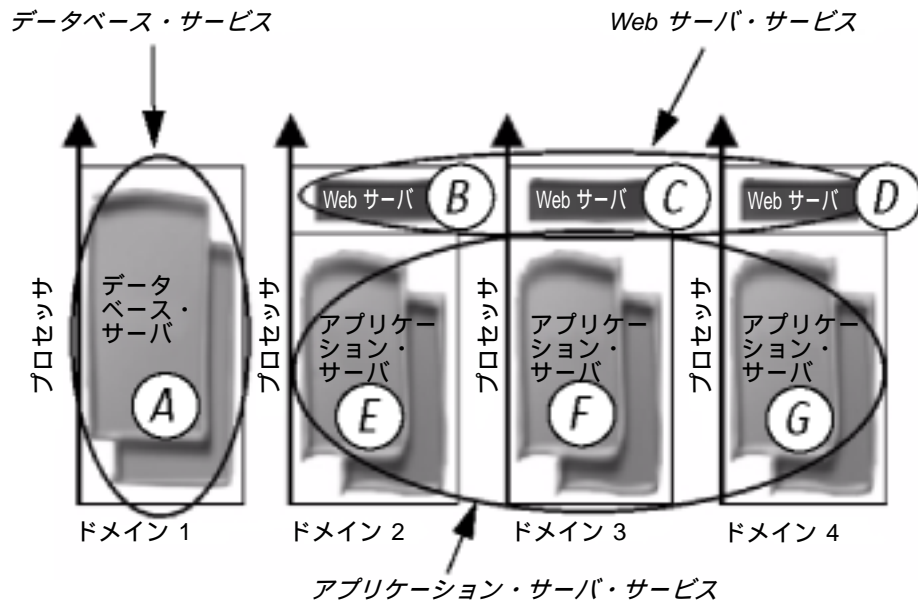


図 2 クラスタ化された環境内での Solaris コンテナの使用

データベース・サービスは、Solaris コンテナ A (丸で囲まれた A として描かれている) として示された単一の Solaris コンテナ内で稼働している主データベース・インスタンスから構成されています。Solaris コンテナ A は、オペレーティング環境がデータベース・サーバ用として最適化された単一のドメインに対応付けられています。このため、このドメインのリソースはすべて単一の Solaris コンテナに割り当てられます。このデータベース・サーバは複数のドメインに分散していないため、Solaris コンテナ A にリソースを追加するにはこのコンテナが稼働しているドメインにキャパシティを追加するしかありません。データベース・パフォーマンスが低下する場合には、構成に応じて ADR (Automatic Dynamic Reconfiguration) を通してほかのドメインからキャパシティを借りることができます。

Webサーバ・サービスとアプリケーション・サーバ・サービスは、それぞれのソフトウェア・サーバの複数のインスタンスから構成されています。これらのソフトウェア・コンポーネントは、この例では 3 つのハードウェア・ドメインに渡って分散した個々の Solaris コンテナに対応付けられています。これらのコンテナは、ほかのコンテナから動的にリソースを借りるか、あるいはコンテナに割り当てる物理リソース (CPU、メモリーなど) を追加することによってサイズを変更できます。また、ほかの既存コンテナのリソースを一部割いて新しいコンテナに供給することもできます。

この環境が柔軟なものであれば、フェイルオーバーを処理する方法は多数あります。図 3 に例を示します。

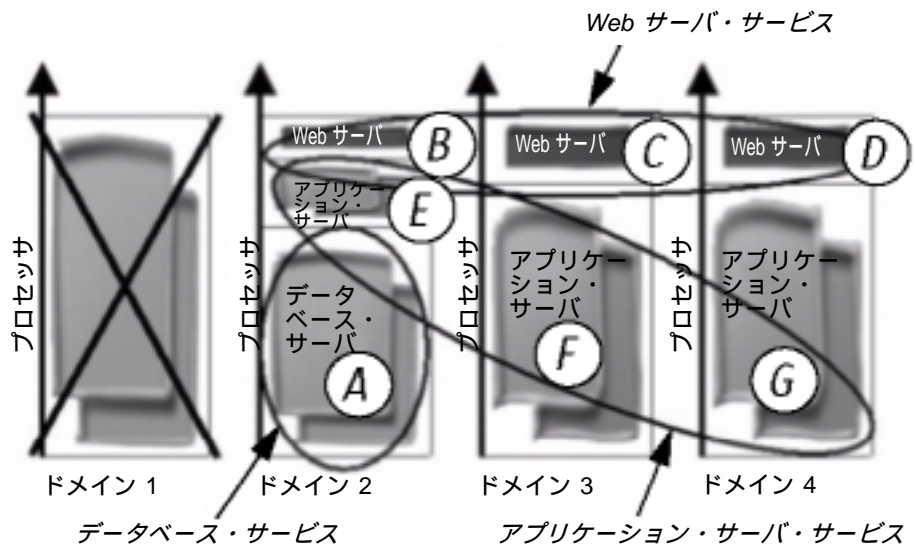


図 3 データベース・サーバのフェイルオーバー後のクラスタ例

データベース・ドメインで障害が発生すると、たとえば SunPlex システムはデータベース・コンテナ (コンテナ A) をドメイン 1 からドメイン 2 へフェイルオーバーできます。フェイルオーバーしたデータベース・サーバ・インスタンス用として領域を設けるため、ドメイン 2 内の残る 2 つのコンテナ (B と E) のサイズが変更されます。サービス管理アプリケーションは、あらかじめ設定されたサービスレベル・ポリシーに応じてボトルネックを最小限に抑え全体的なスループットを最大限に高めるように、各コンテナの監視と調整も継続する可能性があります。このような方法をとることで、物理的なホスティング場所にかかわらず障害とセキュリティの境界はコンテナごとに保持されます。

Solaris コンテナ用 Sun™ 製品の動向

Sun が Solaris コンテナ手法を選択したのは、この手法が管理パラダイムを変え、ネットワーク上に相互に連結したコンピュータ・リソース機構を確立するためです。サービスレベル管理にとって統合された包括的な方法である Solaris コンテナを利用すれば、管理が効率化されるとともに、アプリケーション・サービス・プロビジョニングの柔軟性が飛躍的に高まると期待されます。

単一の OS インスタンス内で複数の実行環境を分離する Solaris コンテナは、不要なパフォーマンス / 管理オーバーヘッドを生じさせることなくサーバの仮想化という概念をサブ CPU レベルまで広げます。ベンダーの中には、仮想マシンの上部でオペレーティング・システムの複製を個別に稼働させることによって包囲するという方法をとっているところもあります。これは、包囲されたアプリケーションから実行される各命令がオペレーティング・システムのほか仮想マシンにも渡される必要があることを意味します。この結果、不要なシステム・オーバーヘッドが起き、余分なオペレーティング・システムの管理が必要なために管理が複雑化します。

これにひきかえ、Solaris コンテナは単一の Solaris オペレーティング環境内で複数動作し、仮想マシンのパフォーマンス・オーバーヘッドや管理環境の複雑化を起こすことなく分離を達成します。

初の Solaris コンテナは、Solaris 9 オペレーティング環境の計画リリースに含まれます。本書の Solaris コンテナ例で述べたように、Solaris 9 ソフトウェアではシステム・リソースを細部に渡って制御する機能が用意されるでしょう。続いて Sun は、ソフトウェア製品ラインおよびシステム管理製品ライン全体を通して Solaris コンテナの概念を統合し、Solaris コンテナ用の障害分離とセキュリティ分離を提供するように Solaris オペレーティング環境を向上させる予定です。

Sun™ Management Center ソフトウェアについては、多くのコンテナ管理作業を自動化して管理者を細かな管理作業から解放する高レベル・サービス管理機能を提供するように引き続き改良を行います。今後のバージョンでは、サービス管理コンソール内からリソース管理ポリシーを設定できる可能性があります (サービス管理コンソールでは、Solaris コンテナを定義し、エンドユーザ・サービスを管理する一環として特定のコンテナにアプリケーション・サービスを割り当てることもできる)。Sun は、管理者が詳細をよく把握できるように、Sun Management Center ソフトウェアによってほかの Sun 管理ツールを監視し、不可欠な Solaris コンテナ境界を確立するという展望を描いています。

Sun は、オープン標準を促進するアプリケーション・プログラミング・インタフェース (API) を引き続き提供しながら、この新しい管理パラダイムをサポートするあらゆる製品を改良していきます。強力なサードパーティ管理ソリューションから成る Sun の豊富な品揃えは、オープン標準のサポートを通してさらに拡大するでしょう。

Solaris コンテナとサービスレベル管理の将来

アプリケーションのリソース消費を制限する境界を設けるという概念は新しいものではありません。実際、既存のリソース管理ツールでもこの種の境界を設定できます。目新しいのは、コンテナによって同じ OS インスタンス内に独立した複数の実行環境を提供し、動的で予測可能な、安全性の高い、管理性に優れた IT インフラを確立できるという発想です。

コンテナ内のソフトウェア・コンポーネントにはハードウェア・プラットフォームを意識させないように、管理者はすべての Sun システムに渡って一様なコンテナ環境を提供できます。また、ソフトウェア・コンポーネントの配備先となる環境に、コンテナ用のリソースが適宜用意されているかぎり、ワークステーション上にも大規模サーバにおける場合と同様にソフトウェア・コンポーネントを簡単に配備できます。ネットワーク化された環境という状況におけるこのハードウェア独立性は新たな柔軟性をもたらし、アプリケーションのプロビジョニング、移行、統一はこれまで以上に簡単に進むようになるでしょう。システム管理者は、個々のプロセスのセキュリティ、障害、リソース配分などを気にすることなくサービス・レベルを管理できるはずです。

Solaris コンテナで計画されている今後の機能には、次に示すように大きな可能性があります。

- 管理 / 課金機能により、複数の Solaris コンテナに分散しているサービス全体を追跡できるようになるでしょう。システムはサービスのパフォーマンスを目標値と照らし、必要があればリソースを動的に割り当て直してサービスレベルを維持できます。システムにさらにリソースを追加する必要がある場合には、システム・オペレータに通知されます。
- システム管理者は、Java™ 2 Platform, Enterprise Edition (J2EE™) 環境用のサービス・コンテナの構築と同時に Solaris コンテナが 1 つ自動作成されるように設定できます。これによりシステム管理が効率的になるだけでなく、個々のソフトウェア・コンポーネントおよびエンドユーザー・サービスの詳しい利用情報を入手して課金や入金相殺を簡易化できます。

- アプリケーションは、ストレージ・デバイスから完全に切り離すことができるようになります。これを実現するのは、仮想ネームスペースや仮想ファイルシステムが提供するアブストラクション層です。今日の SunPlex 環境における場合と同様に、フレームワーク内のどこかにアプリケーションをインストールすればただちに、フレームワーク内に存在するほかの任意のノード上でそのアプリケーションを実行できます。
- エンドツーエンド・サービスを対象とした細かなリソース課金では、ソフトウェア・ライセンス供与のためのユーティリティ・モデルを実現できます。たとえば、エンドツーエンド・サービスでどれだけ CPU タイムが使用されるかに応じてソフトウェアの価格を設定できます。これは、そのサービスが複数の Solaris コンテナに渡っていても問題ありません。対象となるシステムのサイズにもとづいてソフトウェアのライセンスを供与するというやり方は筋が通らなくなるでしょう。Solaris コンテナが使用されている間はソフトウェアがシステム全体にアクセスできないように設定することもできます。
- プロビジョニング・ツール (Solaris Web Start Flash ソフトウェアなど) はシステム・イメージ全体をコピーして複製する必要がなくなり、個々の Solaris コンテナを複製できるようになります。

第7章

管理性 – Solaris コンテナの価値

Solaris コンテナは、IT システムの総所有コストに多大な影響を与えます。Solaris コンテナを利用することで管理者は、リソース、障害、セキュリティなどを気にせずに大規模システムを統合し、簡単に管理できます。これは、システム管理効率やリソース利用率の飛躍的な向上につながります。Solaris コンテナには、管理モデルをサービス中心型に変える効果もあります。基礎的なリソース / システム管理作業は、これまで以上に標準化された自動処理として行えるようになります。

Solaris コンテナの主な利点を示します。

- **管理コストの削減** - 少数のサーバにアプリケーションが確実に統合されるため、管理の手間が省け、管理コストが減少します。また、Sun 製品ライン全体における Solaris コンテナ・モデルへの意図的な標準化によってサービス・プロビジョニングが簡易化され、トレーニングや再構成にかかる費用も抑えられます。
- **リソース利用率の向上や運用コストの減少** - リソースの動的な再割り当てによって使用されていないリソースを必要に応じほかのコンテナに移すことができるため、リソースの利用率が高まります。また、Solaris コンテナの障害とセキュリティが分離されるため、動作状態の悪いアプリケーションのために利用されていない専用のシステムを当てる必要がなくなり、それらのアプリケーションをほかのアプリケーションと統合できるようになります。
- **サービスの可用性の向上** - 障害分離とセキュリティ分離により、エラーの拡散や、パフォーマンスまたは可用性に影響を与える可能性のある各種の侵入 (意図的侵入または意図的ではない侵入) からアプリケーションが保護されます。この環境は一貫性に優れ、いたってシンプルであるため、可用性に影響を与えかねない人的ミスの発生も少なくなります。
- **柔軟性の向上** - Solaris コンテナにより企業は、リソース制約、障害、セキュリティ問題などによってアプリケーションが影響を受けることをさほど心配せずに確実にサーバ統合を実施できます。Solaris コンテナは再構成も容易です。Solaris コンテナを利用すれば、ハードウェア・セット間でアプリケーションを移す作業もいたって簡単に行えます。

Solaris コンテナの柔軟性を示す例を挙げると、Sun Fire™ 3800 上の 4 CPU コンテナは大規模な 106 プロセッサ Sun Fire 15K サーバ上の 4 CPU Solaris コンテナとまったく同じであるかのように見えます。管理者は Solaris コンテナの全プロパティがアプリケーションと共に移動することを認識し、システム間でアプリケーションを必要に応じて自由に移動させることができます。

第 8 章

まとめ

ビジネスを繁栄させるには、エンドユーザを満足させる高いサービス・レベルを維持するだけでなく IT 経費も抑える必要があります。一見対立しているように思われるこれらの目標は、予算を監視しているだけに留まらない的確なアプローチを必要とします。管理性を飛躍的に高める Solaris コンテナを利用することで、企業は作業効率やリソース利用率の向上を通してコストを削減できます。この新しい手法を採用すれば、高いサービス・レベルを求めて止まないエンド・ユーザ需要を継続的に満たしながらコスト節約を実現できるでしょう。

© 2002 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303 U.S.A

All Rights Reserved.

本書および本書に記載された製品（関連する文書を含み、以下同様とします）は著作権により保護されており、その使用、複製、再頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社または同社に対する実施許諾者の書面による事前の許可なく、本書および本書に記載された製品のいかなる部分も、いかなる方法によっても複製することが禁じられます。

フォントを含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。本書に記載された製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

Sun, Sun Microsystems, Java, J3EE, Solaris, Sun Fire、および SunPlex は、米国およびその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems, Inc. の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標の付いた製品は、米国 Sun Microsystems, Inc. が開発したアーキテクチャに基づくものです。

UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行なわれないものとします。

本書および本書に記載された製品が、外国為替および外国貿易管理法（外為法）に定められる戦略物資等（貨物または役務）に該当する場合、それを輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による同意を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。



**Solaris™ コンテナ - サーバの仮想化が進むことにより
サービスの管理がいかに簡易化されるか
IT エグゼクティブ / 管理者向け技術要約
2002年2月**

初 版 2002年2月

監 修 製品事業統括本部
ソフトウェア事業部

発 行 サン・マイクロシステムズ株式会社
〒158-8633 東京都世田谷区用賀4丁目10番1号SBSタワー
電話 (03) 5717-5000
