



Oracle ホワイト・ペーパー  
2013年3月

## Oracle Solaris : エンタープライズJavaに 適した最善のプラットフォーム

## 免責事項

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないで下さい。オラクルの製品に関して記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

はじめに .....	1
適切な基盤の使用 .....	2
第一級のパフォーマンスおよびスケーラビリティの活用 .....	3
IBM pSeriesの1/3のコストで3.4倍のパフォーマンスを実現 .....	4
HP ProLiantサーバーおよびMicrosoft Windows 2008 R2と比較して、 チップあたり最大4倍の速度.....	4
最適化によるJavaアプリケーションのパフォーマンスおよびスケーラビリティの向上 .....	5
Java仮想マシンに対する最適化の継続.....	7
エンタープライズJavaアプリケーションに対するミッション・クリティカルな可用性の提供	8
Oracle Solarisの予測的自己修復機能 .....	8
Oracle Solaris Image Packaging System (IPS) による予測可能性の向上 .....	9
Oracle Solaris Cluster .....	10
Oracle Solaris ZFSによるデータ整合性およびデータ可用性の向上 .....	10
セキュリティの強化 .....	11
Javaサービス管理の簡素化.....	13
Oracle JRockit Mission Controlで利用できるOracle Solaris DTraceプロンプト .....	13
クラウド・コンテキスト内でのJavaサービスの管理 .....	14
結論.....	15
追加情報 .....	15

## はじめに

Sun Microsystems (現在のオラクル) によって開発されたJavaは、1990年代半ばから、アプリケーションの開発および保守に付随するコストと複雑さの軽減に貢献してきました。現在、Javaはインターネット上に行き渡っているだけでなく、私たちの日常生活を支える数多くのアプリケーションやデバイスの隠れた原動力となっています。Javaは、パフォーマンス、スケーラビリティ、管理しやすさが不可欠な要件となるミッション・クリティカルなアプリケーションでよく使用されます。Oracle E-Business Suite、PeopleSoft、JD Edwards EnterpriseOneなどのOracle ApplicationsやOracle Fusion Middlewareはすべて、全体的または部分的にJavaテクノロジーを使用して開発されています。

しかし、Javaアプリケーションの実行は、Java仮想マシン (JVM) をサポートするオペレーティング・システムの機能や特性に大きく依存します。Oracle Solarisは現在の市場で入手可能なもっとも信頼性が高く、機能豊富なエンタープライズ・オペレーティング・システムであり、開発チームとJavaコミュニティの緊密な協力を通じて発展をJavaとともに遂げてきました。Oracle Solarisには、Javaアプリケーション向けの最適なオペレーティング・システムとなるように特別に設計された機能が多数搭載されています。現在、もっとも要求の厳しいミッション・クリティカル・アプリケーションは、Oracle Fusion Middleware、Oracle Database、Oracle Solarisオペレーティング・システムの組合せを使用して実装されています。

Oracle Solaris 11は、エンタープライズJavaのセキュリティ、信頼性、診断機能、開発者の生産性、効率、物理的なアプリケーション・パフォーマンスを新しい次元へと高め続けています。一例を挙げると、本書で詳述する最近のJavaベンチマークでは、Oracle Solaris 11およびOracle SPARCサーバーがJavaに関するパフォーマンスの世界記録を塗り替えています。

このホワイト・ペーパーでは、Oracle SolarisをJavaアプリケーションを配置するための最善のプラットフォームにしている主要機能について説明します。本書は、Oracle SolarisをエンタープライズJavaに適した最善のプラットフォームにするテクノロジーを理解しようとするシステム管理者を対象としています。

## 適切な基盤の使用

この項では、Oracleスタック上でエンタープライズJavaアプリケーションを実行するために適切な基盤を提供するOracle Solarisテクノロジーについて説明します。Oracleスタックは、互いに連携するよう設計およびテストされたハードウェアおよびソフトウェア・テクノロジーで構成されています。

Oracle Solarisは、エンタープライズ・アプリケーションおよびクラウド構築向けに設計されています。次の項で説明するパフォーマンスの最適化に加えて、Oracle Solarisには仮想化、セキュリティ、高可用性機能などの重要なテクノロジーが搭載されています。これらのテクノロジーはJavaアプリケーションのサービス・レベルを高めるとともに、プロビジョニングおよび管理作業を簡素化します。Oracle Solarisは業界初の完全仮想化OSであり、完全なサービスの仮想化に加えてストレージとネットワークの仮想化を実現します。

次に、より効率的なJava実装を可能にする重要テクノロジーの例を挙げます。

- **Oracle Solaris 11のネットワーク仮想化**は、Oracle Solaris 11の機能であるOracle Solaris Zonesによって実現されます。これにより、管理者は完全に仮想化されたネットワークを構築して、ネットワーク・リソースや仮想化ネットワーク・インタフェースをより効率的に共有することができます。管理者はネットワークの仮想化を利用することで、仮想ネットワーク・インタフェース間で帯域幅の割当てを管理できるため、仮想化ネットワークを使用した場合でもOracle Solaris Zones上で稼働するJavaサービスを品質保証契約 (SLA) にマッピングできます。さらに、この管理機能では、物理ネットワーク・インタフェースにいたるまでの特定の専用ネットワーク・スタックに対してJavaサービスを割り当てることができます。
- Oracle Solaris 11には、アプリケーション実行を継続する**高可用性機能**が組み込まれています。Oracle Solarisの予測的自己修復機能は、多くのハードウェア障害やアプリケーション障害を自動的に検出し、リカバリします。これにより、ソフトウェアおよび主要なハードウェア・コンポーネントに障害が発生した場合やソフトウェアの設定に誤りがあった場合でも、ビジネス・クリティカルなアプリケーションや必須のシステム・サービスを中断せずに継続できます。また、Oracle Solaris Clusterを使用したインテリジェント・ゾーンの移行およびクラスタ化により、ハードウェア障害の発生時にもアプリケーション実行を継続するための追加ツールが提供されます。
- Oracleサーバーを仮想化するための基盤となる**Oracle Solaris Zones**は、単一サーバーへの複数サービスの統合を低いオーバーヘッドで実行するための手段を提供します。Oracle Solaris ZonesはOracle Solaris 11向けの分離環境を提供することで、別々のOracle Solarisインスタンスのように見える、セキュアかつ柔軟でスケーラブルな軽量仮想化オペレーティング・システム・サービスを提供します。Oracle Solaris Zonesはゾーン固有の可観測性を提供し、1ノードあたり数百ゾーンまで拡張できます。
- **Oracle Solaris DTrace**は、システム動作の監視、デバッグ、チューニングに使用できる包括的な監視機能です。DTraceが提供する高度な可視性を利用することで、管理者はユーザーのサービス・レベルに影響が及ぶ前に問題を見つけて修正することができます。

- **Oracle Solaris ZFS**は、卓越したデータ整合性とスケーラビリティを備え、管理オーバーヘッドをほぼゼロに抑えた統合ファイル・システムおよびボリューム・マネージャを提供します。Oracle Solaris ZFSは、ハイブリッド・ストレージ・プール・アプローチを使用して従来のディスク・ストレージとソリッド・ステート・ディスク・ストレージの利点を活用することで、極めて効率的な読み取りおよび書き込みキャッシュを提供し、データのアクセスおよび読み取りを高速化します。リモート・ストレージにボリュームを作成し、iSCSIおよびFCを介してブロック・デバイスとして共有することができます。その結果、クラウド内の共有ストレージ上でOracle Solaris Zonesをホストできるため、仮想化サーバーに加えてストレージ環境の仮想化が提供されます。
- **ソフトウェアのより迅速なインストールおよび配置のためのツール** Oracle Solaris 11の新機能によって、アプリケーションとサービスのインストールや導入が簡素化および迅速化されます。Oracle Solaris自動インストーラは、迅速な配置を容易にし、促進します。Oracle Solaris Image Packaging Systemはソフトウェアのライフ・サイクル管理を簡素化することで、停止時間を最小化します。Oracle Solaris ZFSでは、スナップショットによるロールバックによって組み込みの安全性が提供されると同時に、データ削減によって必要なストレージ容量も削減されます。さらに、Oracle Solaris ZonesおよびOracle VMテンプレートによって、新規および既存のサービスを極めて迅速に展開できるようになります。

JavaアプリケーションとOracle Solarisの組合せにより、管理者と開発者は、追加設定なしの優れたパフォーマンスと高品質の実装に加えて、コード・デバッグとパフォーマンス・チューニングの作業を簡素化するように設計された革新的ツールによって彼らの作業を実行できます。

## 第一級のパフォーマンスおよびスケーラビリティの活用

Oracle Solaris 11は25年以上にわたる継続的な開発と、総合的な配置および管理を簡素化しながらエンタープライズ・アプリケーションで高いサービス・レベルを達成するための飽くなく探求の成果です。Oracle Solarisは何万ものCPUスレッドと数百テラバイトのメモリ容量に対応しており、将来のSPARCおよびx86サーバーに対して、事実上無制限の仮想化、卓越した信頼性、可用性、および保守性を提供します。また、Oracle Solarisのマルチスレッド機能はOracle SPARC T4サーバーおよびOracle SPARC T5サーバーのパフォーマンス機能を利用しています。最近のベンチマークでは、Oracle SolarisをJavaアプリケーションとともに使用することで得られるパフォーマンス上の利点を実証されています。

## IBM pSeriesの1/3のコストで3.4倍のパフォーマンスを実現

世界新記録となった最近のSPECjEnterpriseベンチマーク結果では、Oracle SolarisおよびSPARC T5サーバーが、IBM pSeriesシステムと比べて3.4倍のパフォーマンス優位性を実証しました。SPARC T5-8サーバーは、SPECjEnterprise2010のEjOPSで16,646.34という結果を出しました。これは8チップのIBM Power 780と比べて3.4倍のパフォーマンスです。また、SPARC T5-8のアプリケーション・ハードウェア・リストと比べた場合、IBM Power 780の8チップ構成は3.7倍のコストを要します。<sup>1</sup>詳しくは、[https://blogs.oracle.com/BestPerf/entry/20130326\\_sparc\\_t5\\_8\\_specjenterprise2010](https://blogs.oracle.com/BestPerf/entry/20130326_sparc_t5_8_specjenterprise2010)を参照してください。

## HP ProLiantサーバーおよびMicrosoft Windows 2008 R2と比較して、チップあたり最大4倍の速度

最近のSPECjbb2013-Multi-JVMベンチマークで、Oracle Solaris 11およびOracle Java Development Kit (JDK) がOracle SPARC T5-2サーバーとともに使用され、2ソケット・サーバーの世界新記録を達成しました。SPECjbb2013-MultiJVMのmax-jOPSで75,658を達成し、SPECjbb2013-MultiJVM critical-jOPSで23,334を達成したこの記録は、Microsoft Windows 2008 R2 <sup>2</sup>を実行する4ソケットのHP DL560p Gen8サーバーと2ソケットのHP ML350p Gen8サーバーの両方を上回りました。

これらの結果は、Oracle構成が次の速度を実現することを実証しました。

- 4チップのHP ProLiant DL560p Gen8によるSPECjbb2013-MultiJVMのmax-jOPS結果と比較して、チップあたり3.2倍の速度
- HP ProLiant DL 560p Gen8によるSPECjbb2013-MultiJVMのcritical-jOPS結果と比較して、チップあたり4倍の速度
- HP ProLiant ML350p Gen 8によるSPECjbb2013-MultiJVMのmax-jOPSと、SPECjbb2013-MultiJVMのcritical-jOPSの両方の結果と比較して、1.9倍の速度

<sup>1</sup> SPECおよびベンチマーク名であるSPECjEnterpriseは、Standard Performance Evaluation Corporation (SPEC) の登録商標です。SPARC T5-8 (SPARC T5-8サーバー・ベース・パッケージ、8xSPARC T5 16コア・プロセッサ、128x16GB-1,066 DIMMS、2x600GB 10K RPM 2.5インチSAS-2 HDD、4x電力ケーブル) のカタログ価格：268,742ドル。IBM Power 780 (IBM Power 780:9179 Model MHB、4x3.8GHz 16コア、64x1つのプロセッサのアクティブ化、4xCECエンクロージャ (IBM Bezel付き)、I/Oバックプレーンおよびシステム・ミッドプレーン、16x0/32GB DDR3メモリ (4x8GB) DIMMS-1,066MHz Power7 CoD Memory、12x1GB DDR3 Power7 Memoryのアクティブ化、5x100GB DDR3 Power7メモリのアクティブ化、1xディスク/メディア・バックプレーン2x146.8GB SAS 15K RPM 2.5インチHDD (AIX/Linuxのみ)、4x1,725wのAC電源) のカタログ価格：992,023ドル。出典：TPC.orgおよびIBM.com、2013年3月25日取得。詳しくは、<http://www.spec.org>を参照してください。

<sup>2</sup> SPECおよびベンチマーク名であるSPECjbbは、Standard Performance Evaluation Corporation (SPEC) の登録商標です。2013年3月25日時点の結果。詳しくは、<http://www.spec.org>を参照してください。Oracle Solaris 11.1およびJDK 7 update 17を使用したSPARC T5-2サーバー：SPECjbb2013-MultiJVM max-jOPS：75,658、SPECjbb2013-MultiJVM critical-jOPS：23,334。Windows 2008 R2およびJDK 7 update 15を使用したHP ProLiant DL560p Gen8：SPECjbb2013-MultiJVM max-jOPS：66,007、SPECjbb2013-MultiJVM critical-jOPS：16,577。Windows 2008 R2およびJDK 7 update 15を使用したHP ProLiant ML350p Gen8：SPECjbb2013-MultiJVM max-jOPS：40,047、SPECjbb2013-MultiJVM critical-jOPS：12,308。

詳しくは、[https://blogs.oracle.com/BestPerf/entry/20130326\\_sparc\\_t5\\_specjbb2013](https://blogs.oracle.com/BestPerf/entry/20130326_sparc_t5_specjbb2013)を参照してください。

## 最適化によるJavaアプリケーションのパフォーマンスおよびスケーラビリティの向上

管理者は次のOracle Solaris機能を利用してJavaアプリケーションのパフォーマンスを上げることで、自身の効率を高めるとともに、アプリケーションのサービス・プロバイダのサービス・レベルを向上することができます。

- **動的スレッディング**: JavaアプリケーションがSPARC T4またはSPARC T5プラットフォームで稼働している場合、Oracle Solarisはサーバーの動的スレッディング機能を利用して、もっとも必要性の高いスレッドに追加のCPUリソースを自動的に割り当てることができます。これにより、全体的なアプリケーション・パフォーマンスが向上します。Oracle Solarisのスケジューラは、できる限り多くのコアやソケットに対してスレッドを自動的に分散することで、システム利用率が100%未満である場合のスレッドあたりのパフォーマンスを最大化します。また、Oracle Solarisでは、管理者や開発者が特定のスレッドを"クリティカル"として定義できます。このように定義されたスレッドがSPARC T4またはSPARC T5プラットフォームで実行されると、CPUコア全体が割り当てられます。専用のCPUコアによって中断せずに実行できるようになるため、これはシングルスレッド・アプリケーションのパフォーマンスを向上するのに適した方法です。
- **ラージ・ページのサポート**: エンタープライズJavaアプリケーションは、通常、大容量のヒープを必要とし、このヒープを物理メモリのラージ・ページに格納することでメリットを得ます。スモール・ページのメモリがアクセスされると、対応する物理メモリ・ページを探すために、変換索引バッファ (TLB) 内ですべての仮想メモリ・アドレスを参照する必要が生じるため、追加の処理が必要になります。TLBはページ変換キャッシュであり、最近使用された仮想アドレスから物理アドレスへの変換を保持します。ラージ・ページを使用した場合、TLBは、キャッシュ内でより少ないエントリを持つより大きな物理アドレス空間を効果的に参照することができます。TLBキャッシュ内でアドレスを見つけることができない場合、プロセッサは階層ページ表から読取りを実行するか、または複数のメモリ・アクセスを実行しなければならないため、パフォーマンスへの影響が発生し、コストが高くなる場合があります。ページ・サイズが大きくなるとTLBに対する負担が少なくなり、より効率的にプロセッサ・メモリを使用できるため、アプリケーションがメモリ参照を待機する時間が短くなります。Oracle Solarisではラージ・ページがデフォルトで有効化されており、デフォルトのページ・サイズは4MBです。このページ・サイズ・パラメータはコマンドライン・オプションの`XXLargePageSizeInBytes=n3`を使用して簡単に変更できます。

<sup>3</sup> Javaアプリケーション・ランチャーの標準オプションおよび非標準オプションについて、詳しくは<http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/java.html>を参照してください。

- **不均一メモリ・アクセス (NUMA) アーキテクチャおよびNUMA I/O** : Java HotSpot VMでは、NUMA対応アロケータがOracle Solarisを実行するNUMAシステムを利用して、自動的にメモリ配置を最適化します。これにより、Oracle SPARC T4およびSPARC T5サーバーなどのNUMAマシンで、最大15%のパフォーマンス向上が実現されます。JVMでNUMA対応が有効化されている場合、新世代領域に含まれるEdenが複数のリージョンに分割され、JavaプロセスではNUMAノードあたり1つのリージョンが使用できます。Edenはすべての新しい割当てが実行される場所であり、NUMAに対応している場合、JVMにより、スレッドによって割り当てられた各オブジェクトがJavaスレッドが現在実行されているローカルNUMAノードに配置されるようになります。これにより、新しいオブジェクトの割当てとその後のアクセスに対する待機時間が短縮されます。さらに、新世代、旧世代、およびPermanent世代のSurvivor領域とのやり取りに対して、ページ・インターリーピングが有効化されます。これにより、これらの領域に対する平均のアクセス待機時間がすべてのスレッドで均一になります。NUMA対応はデフォルトでは有効化されていませんが、コマンドライン・オプション-XX:+UseNUMA\*を使用して有効化できます。さらに、I/Oデバイスがシステムに接続されている場合、Oracle Solarisには、どのNUMAノードにデバイスが関連付けられているかが保持されるため、このデバイスにDMAメモリが割り当てられた場合、デバイスが含まれるノードと同じNUMAノードにこのメモリが配置されます。これにより、DMAが常にローカルNUMAノード上に存在するため、DMAを利用するデバイスI/O処理がNUMAノード間を横断する必要がなくなります。その結果、I/O待機時間が最小化され、I/OデバイスのI/Oスループットが向上します。
- **Oracle Solaris 11.1の仮想メモリ・サブシステム** : Oracle Solaris 11.1の新しい仮想メモリ・サブシステムは、基盤プラットフォームのシステム・メモリ・サイズに合わせて自動的に拡張されます。この新機能はメモリ・リソースの割当てを最適化することで、ラージ・ページ・サイズに対応したマシン上でのメモリ割当てをより効率的で予測可能なものにします。組込みのメモリ予測機能は大容量のメモリ・ページの使用を監視し、アプリケーションの使用パターンに基づいて、収集中に作成されるメモリ・ページのサイズを調整します。このパフォーマンスの最適化により、ページの割当て時に多数のより小さなページを集めてラージ・ページを作成する必要がなくなるため、大容量メモリ・ページの割当て速度が向上します。多くの場合、このラージ・ページはラージ・ページ・プール内であらかじめ提供されています。また、メモリ・ページ・サイズと、アプリケーションが必要とするメモリ・ページのサイズが厳密に一致するため、同じ機能がより小さなメモリ・システムでも役立ちます。結果的に、大容量のメモリ・ページに対応したプラットフォームへ移行すると、エンタープライズ・アプリケーションおよびデータベースが透過的に高速化されます。
- **高速暗号化** : Oracle Solaris Cryptographic Framework機能のライブラリは、一連の暗号化サービスとAPIサポートを提供します。これらを利用することで、カーネル・レベルとユーザー・レベルのアプリケーション・コンシューマは、アプリケーションに新規コードを追加することなく、透過的に暗号化処理をハードウェアに委任できます。JavaアプリケーションがJava暗号化ライブラリ・コールを利用すると、これらのコールが透過的にOracle Solaris Cryptographic Frameworkライブラリに変換され、この機能に対応したSPARCまたはx86プラットフォーム上のハードウェア暗号化リソースが使用されます。これにより、Javaアプリケーションを変更したり、各種のパケット処理およびセキュリティ・アルゴリズムを実装するためにCPUリソースを浪費したりすることなく、ワイヤースピードの暗号化を実現できます。SPARCおよびx86プラットフォーム上のOracle Solarisによる暗号化パフォーマンスの高速化について、詳しくは次の関連ブログ・サイトを参照してください。

<sup>4</sup>詳しくは、<http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/java.html>を参照してください。

- [https://blogs.oracle.com/BestPerf/entry/20130326\\_sparc\\_t5\\_2\\_zfscrypto](https://blogs.oracle.com/BestPerf/entry/20130326_sparc_t5_2_zfscrypto)
  - [https://blogs.oracle.com/DanX/entry/sparc\\_t4\\_openssl\\_engine](https://blogs.oracle.com/DanX/entry/sparc_t4_openssl_engine)
  - [https://blogs.oracle.com/DanX/entry/intel\\_aes\\_ni\\_optimization\\_on](https://blogs.oracle.com/DanX/entry/intel_aes_ni_optimization_on)
- **ネットワーク・パフォーマンス**：より多くのJavaアプリケーションおよびサービスがオンライン化されるに従って、ネットワーク・パフォーマンスが果たす役割が重要になっています。Oracle Solaris内の高度なTCP/IPスタックは、ネットワーク・パケットの処理に必要な命令数を減らすことでオーバーヘッドを軽減します。インターネット・プロトコル・スタックを最適化し、新しいハードウェア・テクノロジーをサポートするドライバの開発を容易にすることで、ネットワークベースのアプリケーションをOracle Solarisに移行する際、50%もの高速化が実現する場合があります。高度なソフトウェア・スタックは、ネットワーク・パケット処理時のCPUオーバーヘッドを軽減するだけでなく、スケーラビリティ、割込み処理、Secure Sockets Layer (SSL) トラフィックを改善します。サポート可能なネットワーク接続数が増えるとともに、CPUおよびインタフェース・カードの数に合わせてスループットを直線的に拡張できるため、最新の10GbEアダプタを使用してワイヤースピードでスループットを集約できるようになります。Oracle SolarisのTCPレイヤーおよびIPレイヤーは部分的に結合されているため、Webアプリケーションのネットワーク・パフォーマンスが大幅に向上します。TCP/IPスタックでのスレディングとコードのキューイングが効率化されており、両レイヤーを介して単一パケットが処理されるため、プロセッサの局所性が高まり、プロセッサ・キャッシュ・パフォーマンスが向上し、コンテキスト切替えのオーバーヘッドが軽減されます。その結果、エンタープライズ環境に必要なネットワーク・スループット・レベルが実現します。

### Java仮想マシンに対する最適化の継続

オラクルは現在、2種類のJava仮想マシン (JVM) を提供しており、それぞれが異なる種類のアプリケーション要件に対応しています。Java環境を最適化するための継続的な取組みの一環として、オラクルはこれら2つのJVMのコードベースを統合しようとしています。統一された新しいJVMは、両方のJVMから最善の機能を集約したものになる予定です。

現在提供されている2つのJVMの主要機能は次のとおりです。

- **Java HotSpot VM**は、オラクルが作成したシステム向けとして主要なJava実装であり、ジャストインタイム・コンパイルや適応型の最適化などのパフォーマンス向上テクノロジーを提供します。Java HotSpot VMという名前は、Javaプログラムのパフォーマンスを継続的に分析し、頻繁に実行されるか繰り返し実行されるコード・セグメントを探す能力に由来しています。これらのホット・スポットはのちに最適化の対象となり、パフォーマンスの重要性が低いコードへのオーバーヘッドを最小限に抑えたまま、パフォーマンスの高い実行を実現します。

- **Oracle JRockit**は、業界有数のリアルタイム・インフラストラクチャ機能と他に例を見ないJVM診断を備えたJavaランタイム・ソリューションのファミリーです。Oracle Solaris 11.1では、Oracle JRockitを含むリアルタイム・アプリケーションに対するサポートが拡張されています。アプリケーションのタイマーとタイムアウトのデフォルト分析が、以前のリリースの10ミリ秒から短縮され、Oracle Solaris 11.1では1ミリ秒になっています。リアルタイムJavaアプリケーションに対して、数ミリ秒の確定的な応答時間を提供するOracle JRockit Real Timeは、もっとも要求の厳しい金融アプリケーションや電気通信アプリケーションの待機時間要件にも適しています。Oracle JRockitに付属しているOracle JRockit Mission Controlは、JVMの強力な診断および管理ツールであり、開発環境と本番環境の両方での使用にふさわしい、高度でありながら目立たないアプリケーション監視を提供します。Oracle JRockit Mission Controlは実行システムへの影響をできる限り最小限に抑えて、Javaランタイム環境のプロファイリングを行います。いったんJVMからこのツールが切断されると、Javaアプリケーションは最大速度で実行できます。

Oracle JRockit Flight Recorder、Oracle JRockit Mission Control、待機時間の短いリアルタイム・ガベージ・コレクションなどのOracle JRockit JVMの主要機能はOracle Java HotSpot VMに集約されています。この集約作業は、現在のOracle JDK 7リリースですでに提供されている初期機能により反復的に実施されており、さらに多くの機能がJDK 7 Updateと次のJDK 8リリースの両方に搭載される予定です。

この間、組織はそれぞれの要件に応じて、Oracle JRockit JVMまたはOracle Java HotSpot VM のいずれかの使用を継続できます。以前からOracle JRockit Mission Controlを使用しているか、または、リアルタイムJavaアプリケーションで短い待機時間を必要とする組織には、Oracle JRockitの使用が適しています。Java HotSpot VMに新機能を追加するための開発と拡張は引き続き行われ、Oracle JRockit JVMに対する完全サポートも継続されます。

## エンタープライズJavaアプリケーションに対するミッション・クリティカルな可用性の提供

Javaアプリケーションは多くの場合、ミッション・クリティカルであり、日常業務の中心的役割を果たします。Oracle Solaris上にJavaアプリケーションを配置することで、組織は最高水準のエンタープライズ・アプリケーション・サービスを提供し続けることができます。

### Oracle Solarisの予測的自己修復機能

Oracle Solarisの予測的自己修復機能は、ハードウェア障害とソフトウェア障害の両方を事前予防的に診断、分離、リカバリすることで、システムおよびアプリケーション・サービスの可用性を最大化することを目的としています。Oracle Solarisの予測的自己修復機能は一連の解読困難なエラー・メッセージを提供する代わりに、CPU、メモリ・リージョン、またはI/Oデバイスを動的にオフラインにすることで、コンポーネントがシステム障害を引き起こす前に、自動的に適切な対応を開始します。アプリケーション障害が発生した場合も、Oracle Solarisのサービス管理機能によって自動的にアプリケーションが再起動されます。

Oracle Solarisの予測的自己修復ソフトウェアのもう1つの不可欠要素であるOracle Solaris Fault Manager機能は、障害のあるコンポーネントを分離して無効化することで、中断のないサービスの提供に貢献します。その上、この処理は管理者が問題の存在に気付く前に行われます。また、リモート・サービス・エージェントによって、障害の根本原因を診断するために不可欠な情報が取得されます。

### Oracle Solaris Image Packaging System (IPS) による予測可能性の向上

エンタープライズJavaアプリケーション向けの環境やリソースは、ユーザーのニーズに対応してアップグレードしたり、変更したりする必要があります。従来のパッチ適用は、依存関係を把握し、適切なパッチを決定して適用するための手動パッチ分析を必要とする、複雑な手動プロセスでした。正しいパッチが適用されていなかったり、パッチの依存関係が適切に理解されていなかったりすると、管理者が誤って問題を引き起こす場合もありました。このため、新しいOS環境を本番環境として使用する前にテストが必要であり、全体的なアップグレード・プロセスは非常に時間のかかるものでした。

Oracle Solaris Image Packaging System (図1) は新しいパッケージ管理アプローチを定義することで、より賢明で安全なOSソフトウェア更新を実行します。パッチをインストールするための管理作業が劇的に削減されるだけでなく、システム更新に必要な停止時間が短縮され、パッチ更新プロセスでエラーが生じるリスクが事実上解消されます。

UnixやLinuxなどのその他多くのパッケージ・モジュールとは異なり、IPSではパッチ適用が不要です。ソフトウェア・リポジトリを使用することで、IPSはシステムやアプリケーション・ソフトウェアの管理者による更新方法を劇的に変革します。

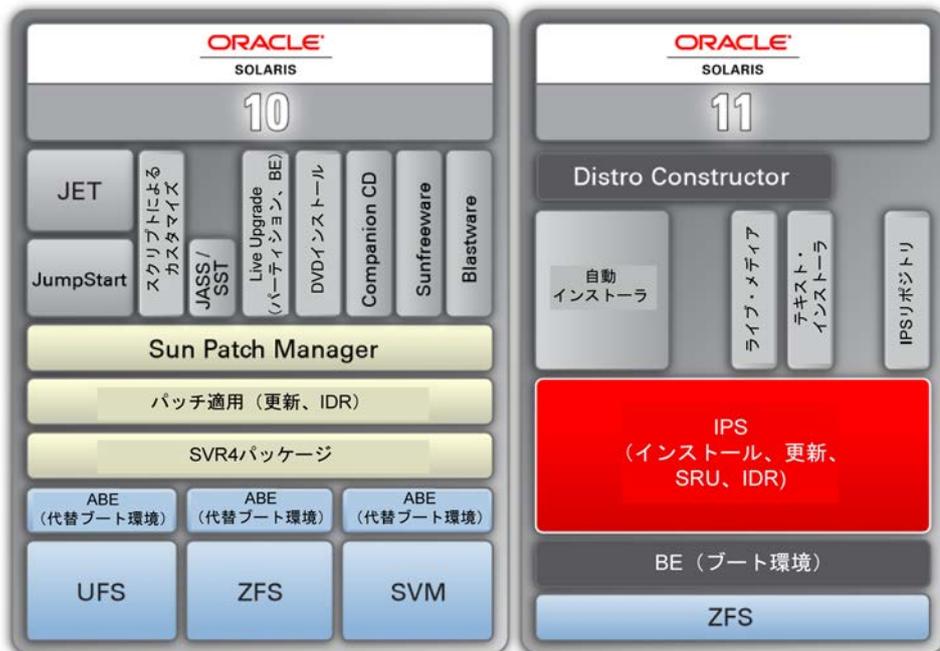


図1 : Oracle Solaris Image Packaging Systemはシステム更新を簡素化し、予測可能性を向上します。

Oracle Solaris Image Packaging Systemは、依存性チェックと自動の依存性解決により、予測可能性を高めて、システム更新が正しく実行されるようにします。更新対象のOSパッケージはすべて、ダウンロードおよびインストールされる前に完全な統合テストを受けているため、これにより、人為的なエラーが非互換性や問題を引き起こすリスクが軽減されます。

さらに、新しいブート環境の作成時に古いブート環境（現在のシステム状態）が自動的に保存されるため、管理者は素早く簡単に以前の状態に戻すことができます。Oracle Solaris ZFSのスナップショットを使用して現在のブート環境のクローンが作成され、パッケージの変更がこのクローンに対して適用されます。新たに作成されたこのクローン環境は、デフォルトでリブートと同時にアクティブ化されます。問題が発生した場合、管理者は以前のブート環境イメージに容易にロールバックできるため、システムを素早く以前の状態で使用できるようにします。この方法によって、Oracle Solaris 11ではOSのアップグレードやソフトウェアの変更に対して管理面での安全策が提供されるため、システムとアプリケーションの可用性が向上します。

## Oracle Solaris Cluster

ほとんどのエンタープライズJavaアプリケーションは非常に重要性が高いため、サーバー障害が発生した場合でも処理を継続する必要があります。Oracle Solaris Clusterは、可用性の高いクラスタ環境でこれらのアプリケーションを実行する機能を提供します。この環境にはOracle Solarisの障害検出テクノロジーとOracle Real Application Clusters (Oracle RAC) が統合されており、Javaアプリケーションの総合的な可用性が最大化されます。

Oracle Solaris Clusterはプライマリ・サーバーの障害発生後1秒未満で、セカンダリ・ローカル・サーバーへのフェイルオーバーを実行します。この迅速な対応のおかげで、基盤システムの障害がJavaアプリケーション・ユーザーに気付かれることは事実上ありません。Oracle Solarisに組み込まれた包括的な障害検出テクノロジーは、Oracle Solaris Clusterに含まれるポリシーベースでアプリケーションに固有の高度なリカバリ・テンプレートを使用することで、特定の種類の障害状態に応じて適切な対応を決定します。さらに、Oracle Enterprise Manager Ops Centerによって、クラスタ・トポロジ検出、プロビジョニング、クラスタ全体のパッチ適用、構成の更新が自動化されるため、管理者が可用性の高いクラスタ配置を容易に管理できるようになります。

## Oracle Solaris ZFSによるデータ整合性およびデータ可用性の向上

現在のJavaアプリケーションの多くは、有益な知的財産である膨大なデータ・リポジトリを作成または利用しているため、ITインフラストラクチャが信頼できるデータ・アクセスと完璧なデータ整合性を提供することが重要になっています。

よく使用されるのは、コンポーネントまたはインターコネクトの冗長性やRAIDなどの基本的データ保護手法ですが、これらは完全な保護を提供するものではありません。Oracle Solaris ZFSには、データの可用性と整合性を向上するために役立つ機能がいくつか搭載されています。Oracle Solaris ZFSはCopy-On-Write方式（データ・ポインタが変更されて書き込みがコミットされる前に、メディア上の新規ブロックにデータが書き込まれる）にエンド・ツー・エンドのチェックサム（後述）を組み合わせることで、ファイル・システムの一貫性を内部的に維持します。ファイル・システムの一貫性が常に確保されるため、システムがクリーンではない状況でシャットダウンされた場合にも、fsck(1)のような時間のかかるリカバリ処理は必要ありません。このため、ユーザーとアプリケーションのサービス・レベルが向上します。

Oracle Solaris ZFSはデータの正確性を維持するため、絶えずデータを読み取ってチェックします。ミラー化されたプール内でエラーを検出すると、このテクノロジーは破損したデータを自動的に修復します。可用性を実現するためのこの絶え間ない監視によって、以前は検出不可能であった発見されにくいデータ破損を含めた、コストと時間のかかるデータ損失が予防されます。これらの修正は、破損データの再構築に役立つパリティ、ストライプ化、不可分操作を利用するRAID-Zの実装によって促進されます。

図2に示すとおり、従来型のファイル・システムはブロック・ベースのチェックサムを使用しているのに対し、Oracle Solaris ZFSはデータ・ツリー全体でチェックサム処理を実行します。図2の左側に示した従来型のアプローチではチェックサムがデータ・ブロックに保存されるため、自己矛盾のないデータ・ブロックはすべてチェックサムを通過します。このアプローチで検証できるのはメディア・データの整合性のみであり、誤余分な書込みを検出することさえできません。Oracle Solaris ZFSはデータ・ブロック・ツリー全体を検証するため、I/Oパス全体の検証が可能になり、メディア（ビット崩壊）をはるかに上回る検証が実行されるため、ファントム書込み、ドライバのバグ、不注意による上書きなどの問題を捕捉できます。Oracle Solaris ZFSではより多くの破損ケースが検出されるだけでなく、これらは自動的に修正されます。



図2 : Oracle Solaris ZFSはチェックサムを使用してデータ・ブロック・ツリー全体を検証します。

## セキュリティの強化

適切に設計されたITインフラストラクチャには、脅威を防止し、検出し、対応するためのセキュリティが最初から組み込まれています。オラクルは、デフォルトでセキュアに構成されたオペレーティング・システムの設計全体にセキュリティを組み込んでいるため、管理者はセキュリティ機能の構成に時間を費やす必要はありません。また、Oracleのソフトウェア保証プロセスには、品質保証プロセスの一環としてセキュリティ・チェックポイントが含まれています。さらに、Oracle Solarisはきめ細かいセキュリティ機能を提供しており、ユーザーとアプリケーションの分離、データの保護、セキュリティ・イベントの監視フレームワークへの統合を実現します。

このようなセキュリティ機能の組合せにより、ビジネス上の利益と知的財産に対する他に例を見ない保護が提供され、幅広いリスクへの対応が可能になります。

Oracle Solarisは、すべてのオペレーティング・システムでもっとも高度なセキュリティ機能のいくつかを提供しており、これらには、Oracle Solaris上でJavaを実行する際のセキュリティを向上する次の機能が含まれています。

- **Java Cryptography Extension** : Java Cryptography Extensionを利用してOracle Solaris上で実行するJavaアプリケーションでは、自動的にOracle Solarisの暗号化フレームワークが活用されます。このフレームワークは、基盤となるサーバー・ハードウェアの暗号化アクセラレーション機能を利用します。Oracle Solaris暗号化フレームワークによって、Javaアプリケーションは、最新式の高度な暗号化アルゴリズムとハードウェア・アクセラレーションを透過的に利用できるようになります。
- **アプリケーションとユーザーの分離** : Oracle Solarisはデータやアプリケーションに対する未承認アクセスを阻止することで、攻撃に対するシステム・セキュリティを強化します。このソフトウェアは、アプリケーションやユーザーのアクセスを制限し、それぞれの役割を果たすために必要なシステム・リソースへのアクセスのみを個別に許可します。この機能は不適切なシステム・アクセスを排除することで、作成された不適当なアプリケーションによる攻撃の可能性を劇的に軽減します。ハッカーがアプリケーションへのアクセスを獲得した場合でも、残りのシステムに影響を及ぼすことはできないため、悪意のあるコードを注入したり、データを損傷したりする可能性が限定されます。
- **アプリケーションのセキュリティ・ポリシー** : 上述したものと同一ロールベースのアクセス制御フレームワークを使用して、管理者はアプリケーション動作を厳密に制御することができます。たとえば、Webサーバーに対して、ポート70のみにバインドするセキュリティ・ポリシーを割り当てると、ジョブを実行できるようにしながら、それ以外の処理を防止することができます。これはOracle Solaris 11.1の新機能であり、セキュリティ・コードを変更することなくアプリケーション動作の変更を可能にします。
- **保管中データおよび移動中データの暗号化** : Oracle Solarisはディスク上のデータと転送中のデータの両方を保護することで、ディスクからクライアントまでにわたってデータの機密保護を実現します。ZFSはブロック・レベルで暗号化を実施することで、未承認アクセスや物理ストレージの盗難に対する追加の保護レイヤーを提供します。暗号化プロパティは、Oracle Solaris ZFSファイル・システムの作成時にアクティブ化され、ファイル・システムのマウント時のパスフレーズや数字キーによるセキュリティ・チェックを含みます。ZFSの暗号化では、Oracle Key Managerを使用して何千ものデータセットに対するエンタープライズ管理を実施できます。移動中のデータは、同じハードウェア・アシストによる暗号化を使用して、各種ユーティリティ (SSH、SCP、Krb)、暗号化API、およびクライアント/サーバー通信 (SSL、IPsec、Kerberosなど) にJava Cryptography Extensionを使用したアプリケーションを介して保護されます。

## Javaサービス管理の簡素化

Javaサービスのサービス・レベルを管理し、基盤インフラストラクチャを保守する作業は、必ずしも困難であるとは限りません。作業を効率的に実行するのに適したツールを使用することで、大きな違いが生まれます。次に、Javaサービスの管理作業を簡素化するOracleテクノロジーを挙げます。

### Oracle JRockit Mission Controlで利用できるOracle Solaris DTraceプローブ

Oracle JRockit Mission Controlは詳細な可視性を提供しているため、管理者はJVMの内部状態およびアクティビティに加えて実行中のJavaアプリケーションを監視できます。この可視性は、Oracle Solaris DTrace内の次の計測プローブを使用して提供されます。

- VMライフ・サイクル・プローブ：VMの初期化および停止用
- スレッド・ライフ・サイクル・プローブ：スレッドの開始および停止イベント用
- クラス・ローディング・プローブ：クラスのロードおよびアンロード・アクティビティ用
- ガベージ・コレクション・プローブ：システム全体のガベージおよびメモリ・プールの収集用
- メソッド・コンパイル・プローブ：コンパイル中のメソッドと使用されたコンパイラの通知用
- 監視プローブ：待機および通知イベント、競合監視エントリおよび終了イベント用
- アプリケーション・プローブ：スレッド実行、メソッド・エントリ/メソッド・リターン、オブジェクト割当てに対するきめ細かい検査用

管理者は、稼働中の本番システムでこれらのプローブを安全に使用して、ユーザー・プログラムとオペレーティング・システム自身の両方の動作を調査できます。

Oracle JRockit Mission Control (図3) でDTraceを使用すると、ユーザーがJavaアプリケーションを監視し、ユーザー・プログラムとOSの動作を調査し、パフォーマンスのボトルネックを発見して、実行時の問題をトラブルシューティングすることができます。

Oracle Java SE Advancedの顧客は、Oracle JRockit Mission Control向けのDTraceプラグインを使用できます。



図3 : Oracle JRockit Mission ControlはDTraceのプロープを利用して、本番で稼働中のJavaアプリケーションに対する詳細な可視性を提供します。

## クラウド・コンテキスト内でのJavaサービスの管理

Oracle Enterprise Manager Cloud Controlは、集中監視、管理作業、ライフ・サイクル管理の機能を、Oracleおよびサード・パーティのテクノロジーを実行するシステムを含む完全なITインフラストラクチャに提供します。Oracle Enterprise Manager Cloud Controlを利用すると、Oracle ITスタック全体を1つのコンソールから監視および管理できます。さらに、Oracle Enterprise Manager Cloud Controlはビジネス主導型のIT管理とビジネス中心のトップダウン・アプリケーション管理をサポートすることで、ビジネス・サービス、ユーザー・エクスペリエンス、インフラストラクチャの管理を可能にします。このコンテキスト内でJavaアプリケーションを管理することで、管理効率が上がります。ビジネス上の優先順位に対するリソースのマッピングも容易になります。

## 結論

Oracle Solarisおよびハードウェアとソフトウェアが統合されたOracleスタックは、エンタープライズJavaアプリケーション向けに最適化されたパフォーマンスと配置効率を実現します。互いに連携するよう特別に設計され、広範囲にわたってテストされたOracle SolarisおよびJavaコンポーネントは、ミッション・クリティカルなJavaアプリケーションに最適な環境を提供します。本書で言及した業界ベンチマークは、Oracle Solarisが第一級のJavaパフォーマンスを提供することを実証しています。暗号化処理、NUMA対応、ラージ・ページのサポートなどのOracle Solarisの最適化により、Javaアプリケーションからサーバー・ハードウェアまでがシームレスに統合されることで、Oracle Solaris上で稼働するJavaアプリケーションのコードをまったく変更することなく最適なパフォーマンスを実現できます。

## 追加情報

本書で言及した製品およびテクノロジーについて、詳しくは次のWebサイトを参照してください。

表1 : 追加情報の参考資料

トピック	Webサイト
Oracle Solaris	<a href="http://www.oracle.com/ip/products/servers-storage/solaris/solaris11/overview/index.html">http://www.oracle.com/ip/products/servers-storage/solaris/solaris11/overview/index.html</a>
Oracle Technology Network (OTN) 上の Oracle Solaris技術コンテンツ	<a href="http://www.oracle.com/technetwork/ip/server-storage/solaris11/overview/index.html">http://www.oracle.com/technetwork/ip/server-storage/solaris11/overview/index.html</a>
Java	<a href="http://www.java.com">www.java.com</a> <a href="http://www.jcp.org">www.jcp.org</a>
Oracle JRockit	<a href="http://www.oracle.com/technetwork/ip/middleware/jrockit/overview/index.html">http://www.oracle.com/technetwork/ip/middleware/jrockit/overview/index.html</a>
Oracle JDK—Java Platform, Standard Edition (Java SE)	<a href="http://www.oracle.com/javase">www.oracle.com/javase</a>
Oracle JDK—Oracle Java SE Advanced	<a href="http://www.oracle.com/ip/technologies/java/standard-edition/advanced-suite/overview/index.html">http://www.oracle.com/ip/technologies/java/standard-edition/advanced-suite/overview/index.html</a>
オラクルのサービス	<a href="http://www.oracle.com/ip/products/index.html">http://www.oracle.com/ip/products/index.html</a>
Oracle Support	<a href="http://www.oracle.com/ip/support/index.html">http://www.oracle.com/ip/support/index.html</a>
オラクルのSunサーバー	<a href="http://www.oracle.com/ip/products/servers/overview/index.html">http://www.oracle.com/ip/products/servers/overview/index.html</a>
Oracle Performance and Best Practices Blog	<a href="https://blogs.oracle.com/BestPerf">https://blogs.oracle.com/BestPerf</a>



Oracle Solaris : エンタープライズJavaに適した最善のプラットフォーム  
2013年3月

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

#### お問い合わせ窓口

#### Oracle Direct

**TEL** 0120-155-096  
**URL** [oracle.com/jp/direct](http://oracle.com/jp/direct)



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

Copyright © 2013, 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およびJavaはOracle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。

著者および共著者 : Staffan Friberg、Steve Sistare、Chris Baker、Art Beckman、Parnian Taidi、Glynn Foster、Joost Pronk

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXはX/Open Company, Ltd.によってライセンス提供された登録商標です。0611

**Hardware and Software, Engineered to Work Together**