

Solaris 11で深化するクラウド

～プライベートクラウドの構築とウェブの邂逅～

ジャストプレイヤー株式会社
代表取締役社長
瀧 康史

今日の議題 (アジェンダ)

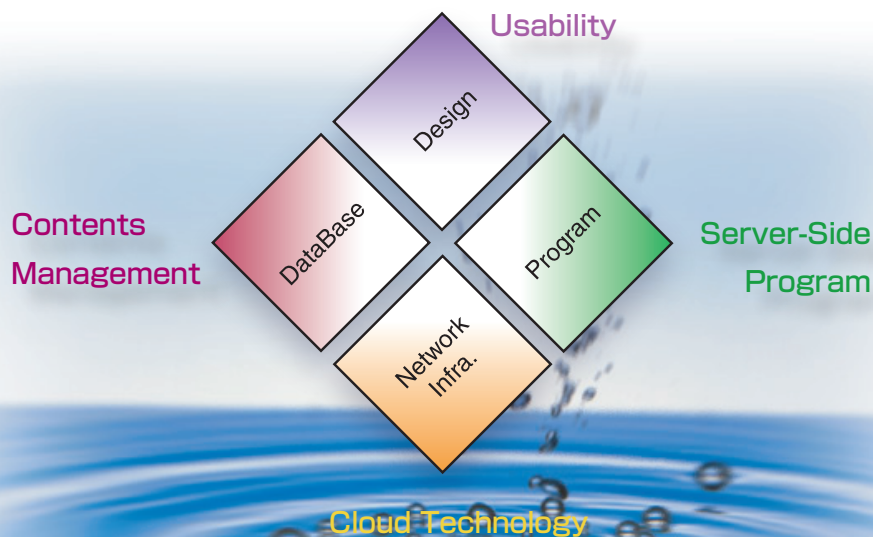
- WIKIPLUS 2 ～カンタンに編集できるウェブ～
- クラウドサービスを作る
- ハイブリッドクラウドのススメ
- Solaris 11がWIKIPLUS 2に与えたその他のベネフィット
- まとめ

ジャストプレイヤー株式会社について



JUSTPLAYER → すぐに遊べるソフトウェアの開発

When You Want Is When You Play → 欲しい時が、遊ぶ時



事業内容

- WEB(ホームページ)制作事業
 - WEB企画、デザイン制作
 - WEBシステム開発
- クラウドソフトウェア制作事業
- クラウドインフラ賃貸事業、ホスティング賃貸事業

その他

- 第二種電気通信事業者届出 届出番号 C-18-1421
- 静岡県中小企業新事業活動促進法に基づく、経営革新計画の承認

自己紹介：瀧 康史

- ジャストプレイヤー株式会社 代表取締役社長
- 出身地
 - 静岡市 (旧清水市)
- 業務履歴
 - ライター業 (テクニカルライター)
 - S社' X紙' (S社のパソコン向け雑誌)
 - ゲームディレクター、アシスタントプロデューサー、プランナー
 - S社にてS社用ゲーム機のゲームソフト開発
 - 2001年、有限会社ジャストプレイヤーを起業。
 - 2006年、ジャストプレイヤー株式会社に改組。
- その他
 - 日本OpenSolarisUsersGroup Leader
 - 静岡WEBデベロッパーフォーラム



やすし虫

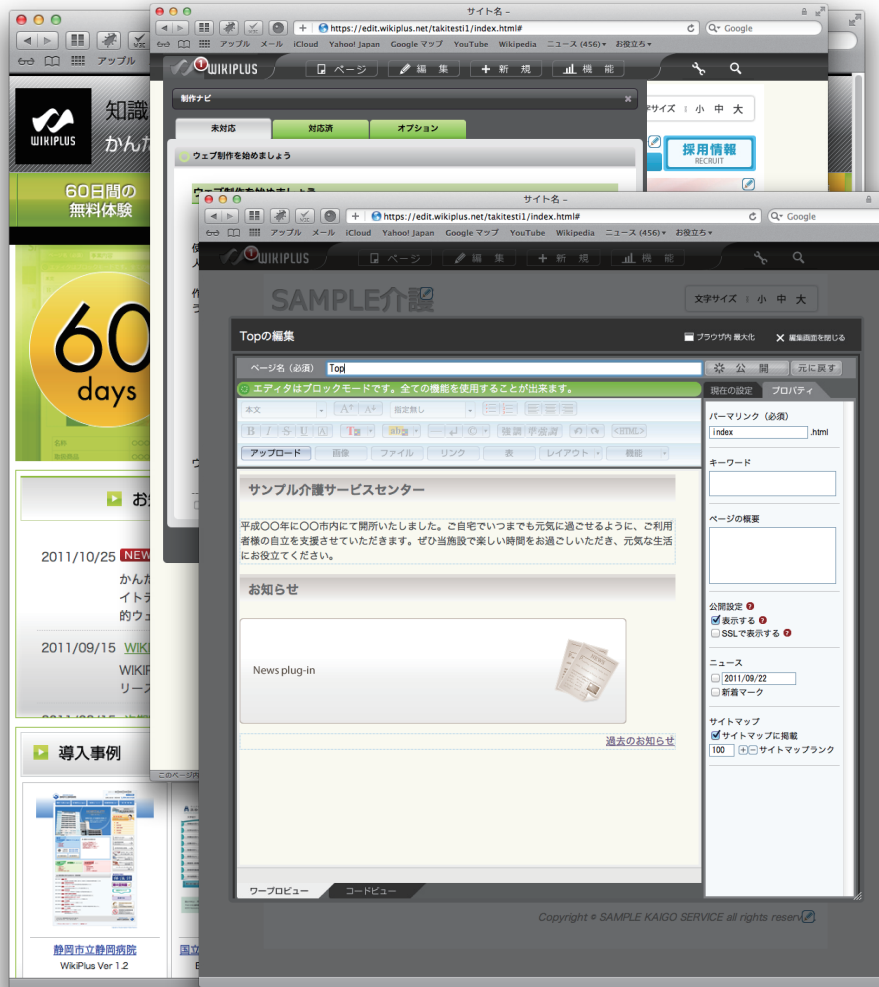
WIKIPLUS 2

～カンタンに編集できるウェブ～



ウェブサービスが Solaris 11 に出会い
出会ったことによってできたこと

カンタン編集 WIKIPLUS Ver 2

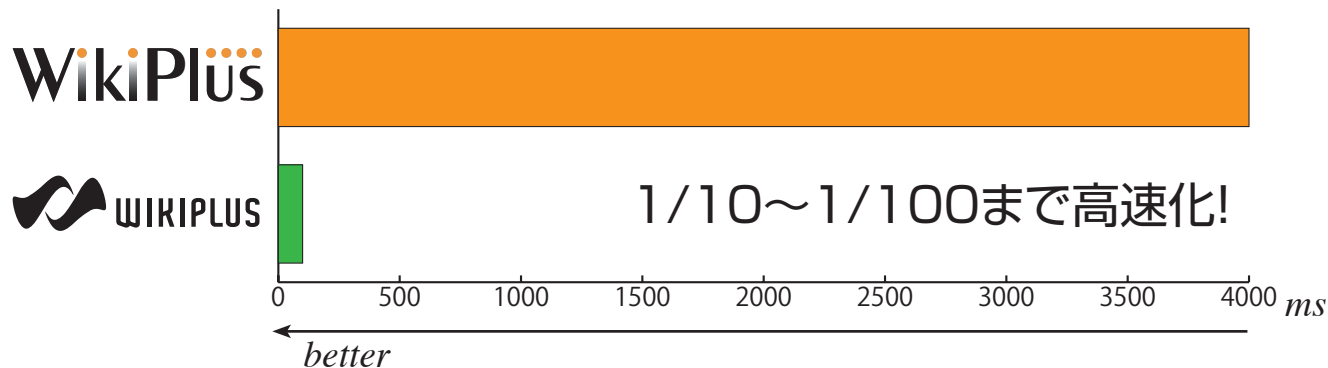


- カンタンに編集できるウェブサイト、「そのもの」になるために作成されたSaaS。ホームページそのものがインフラという意味では、ある種のクラウド。
- Solaris 11で作られたクラウドシステムで動作。
- WikiPlus Ver 1.0を2005年9月にリリースしてから7年目のリリース。旧バージョンが持っていた大半のボトルネックを、Solaris 11の機能を使って改善。大きくリファクタリングした。

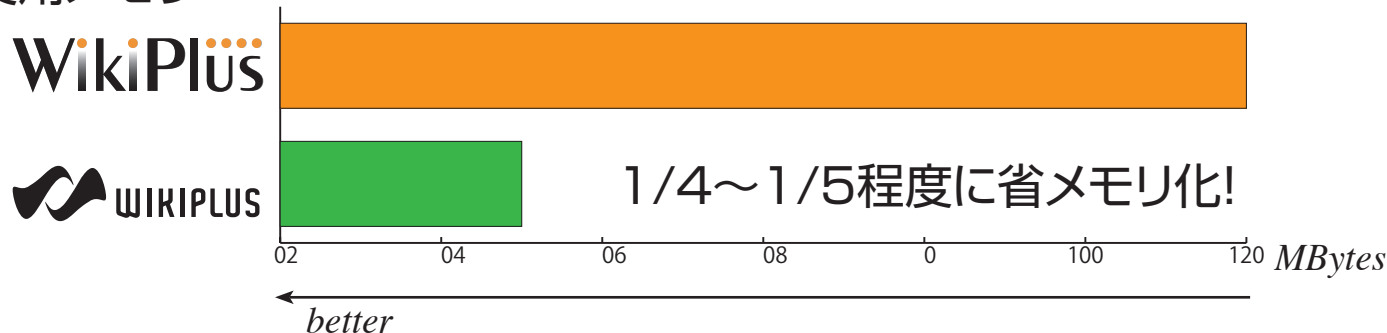
Ver1系～Ver2系での改善

※ これらの最適化はD-Traceなどを利用して解析した。

■ 応答速度



■ 使用メモリ



その他、物理サーバ1台あたりのリソース効率も、比較にならないほど効率化!

旧 WikiPlus の問題その1

- サービスプランとハードウェアが密着
 - BASICプランサーバ、Proプランサーバ、Standardサーバ、Advancedサーバ、特定顧客用サーバ、旧バージョンサーバ……。それぞれが1Uサーバ……。
- OS固有の仕様に密着
 - Redhat Enterprise Linux V4のパッケージに依存
- ユーザニーズに合わせてプランが作れない。
 - 設定値や、リソースの差異がハードウェアに密着したため、新たなプランを作る度にサーバが必要になるか、ニーズにキャッチアップしたプランが作れないなどの問題が生じた。

→ 安価に、リソース共有率が高いクラウドスタックが必要

その他のサービス事情……

WIKIPLUSに限らず、当社で行っていたその他のサービスの問題も……

- 他のInternet向けサービスサーバの数の増加問題
- Legacy Linuxで作ったサービスのアップデート問題
- 個別システム案件毎の問題
 - ウェブホスティング収容を超えると、突然初期費用見積が上がる。
 - サーバ準備費など
 - サーバの準備、撤去が案件のライフサイクルで発生する。
 - 手間がかかる

→やはり、これらを安価にサービスできるクラウド環境が必要。

クラウドサービスを作る



クラウドサービスをどう作るのか？

クラウドとはなにか？

様々な人の定義がありますが、共通しているのは、

「インフラを借りること」

にあるのではないかと、思います。では、「インフラとはなにか？」というと、

「僕は、どういう仕組みなのか知らないけれど、
ちゃんと動いてないと困るもの」

ではないでしょうか？なにをインフラとみるかは「その人次第」です。

「ちゃんと動かないと困るインフラを提供する仕組み」を作り上げる話です。

クラウドスタック選び

※プロジェクト開始時は仮想化インフラ選びと言われていた。

「ハードウェア資源と、サーバを分離する」

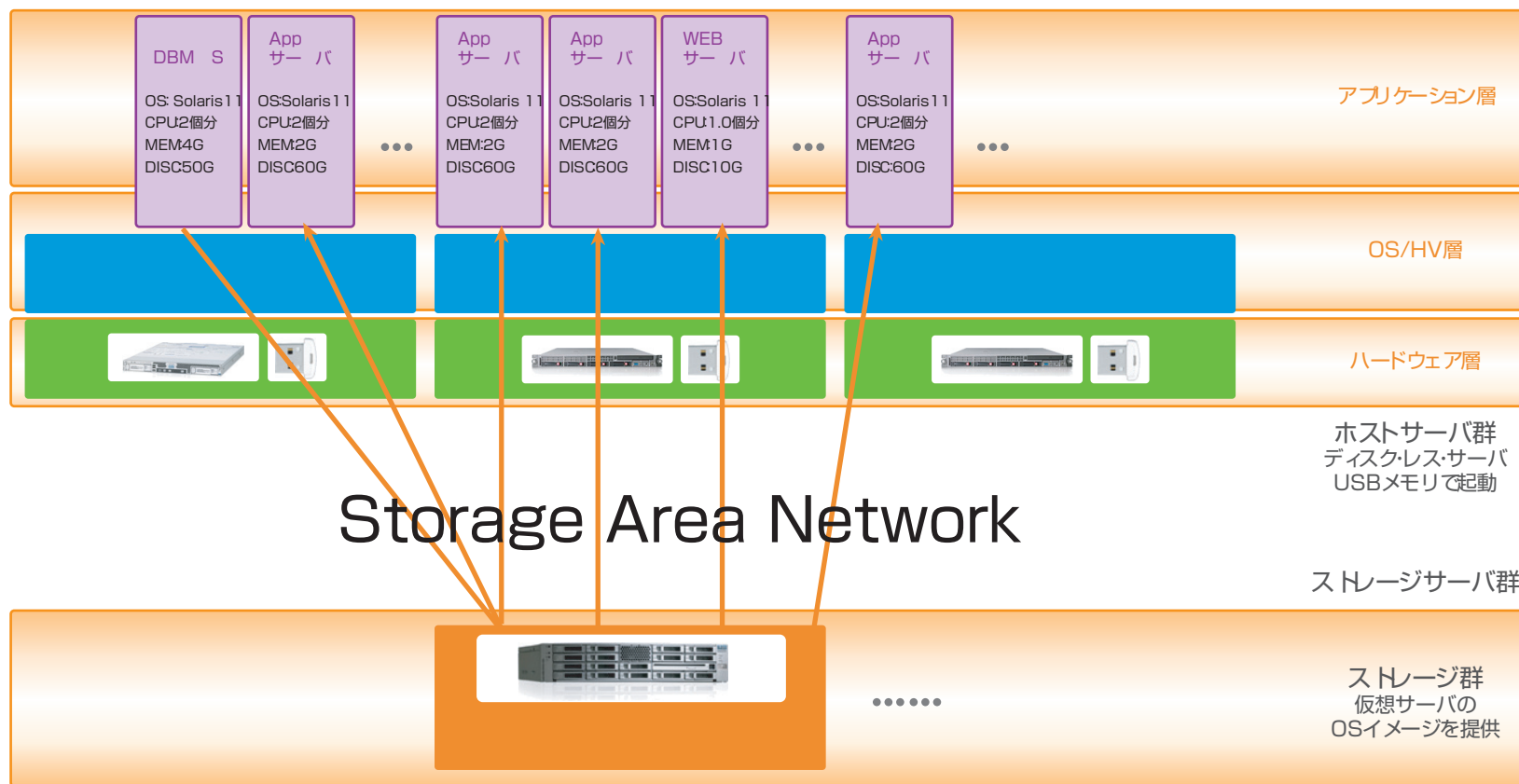


1台の物理サーバの上に、複数の仮想サーバを動かすのではなく、
複数台の物理サーバの上に、たくさん仮想サーバを動かしたい!!!

ストレージの仮想化が必須

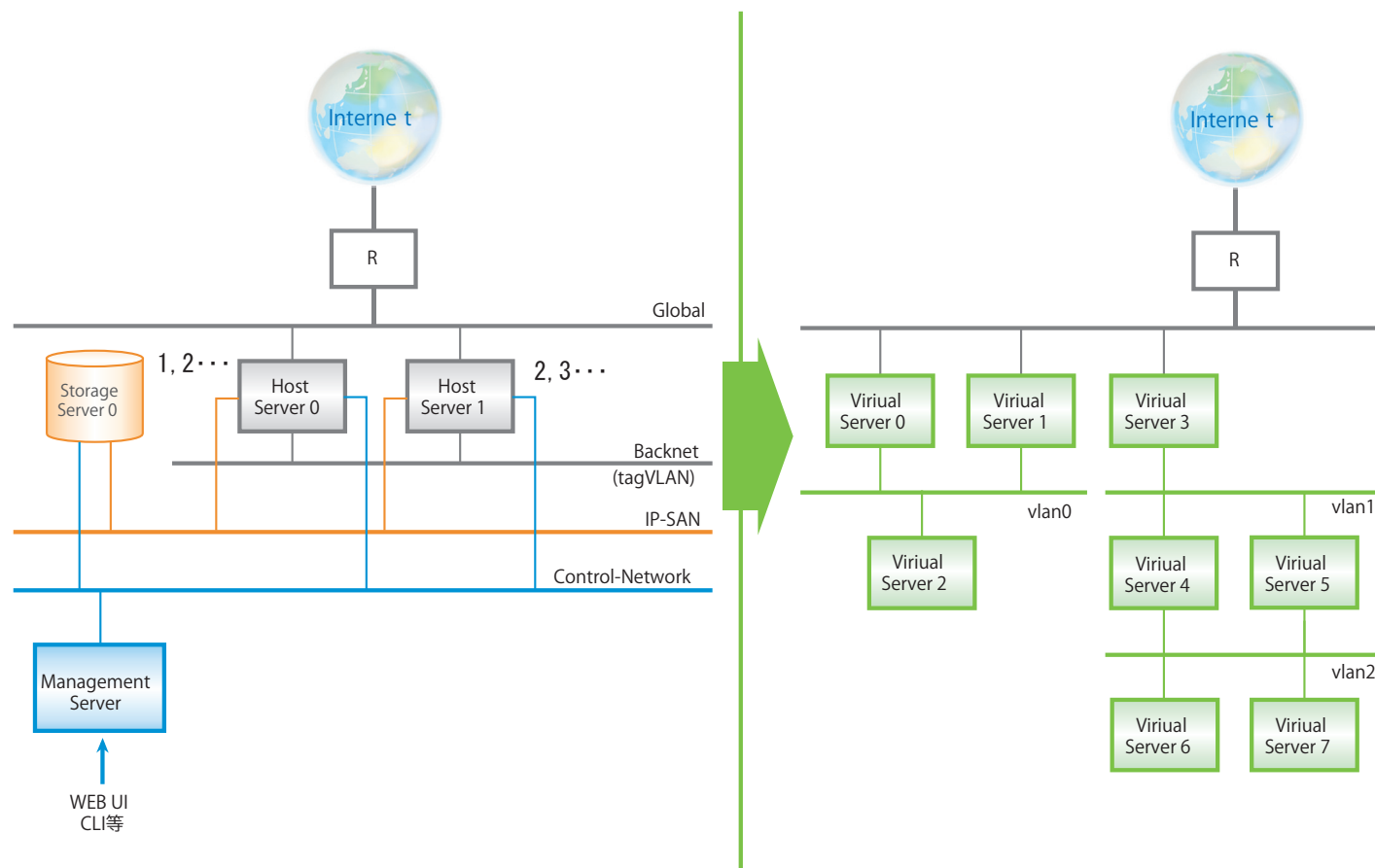
仮想サーバを複数の物理機にいつでも移動できるように設置。

物理機には一切のデータを起さたくない。



ネットワークの仮想化も重要

ハードウェアの再結線なしで、論理サーバセットのプライベートネットワークを自由に増やしたり隔離したりできることが必要



サーバの仮想化

ハイパーバイザ型

- Oracle VM Server
- VMware ESX Server
- Citrix Xen Server
- Redhat KVM
- Microsoft Hyper V

コンテナ型

- Solaris 11 Zones
- Linux lxc/cgroups, OpenVZ/Virtuozzo
- FreeBSD jail

ハイパバイザ型のメリット

- 1つの物理機で、異なる種類のOSを動かすことができる
 - パラバーチャリゼーションドライバの出来次第で速度が大きく変わる。
 - 「ハイパバイザがサポートするOS」と、「OSが動作保証するハイパバイザ」の双方がきちんと整合していないと、肝心なところでサポートが受けられない可能性もあるので注意が必要。
- ライブマイグレーション
 - 仮想サーバ単位で、収容ホスト（物理機）間を稼働させたままを移動。
 - 現実問題として、「ハードが壊れ始めたら、その上のVMはもう信じられない」ので、壊れそうなら移動するなんてことを検出するのは実際は難しい。
 - ハイパバイザのメンテナンス時に逃がすというメリットの方が大きい。

ハイパバイザ型のデメリット

最大のデメリット

- 遅くなるプログラムは、本当に遅くなる!(しかも原因究明が難しい)

■ あるプログラム(Java Servlet + 某RDB)の例では……

ベアメタル

PVM

VMware ESX

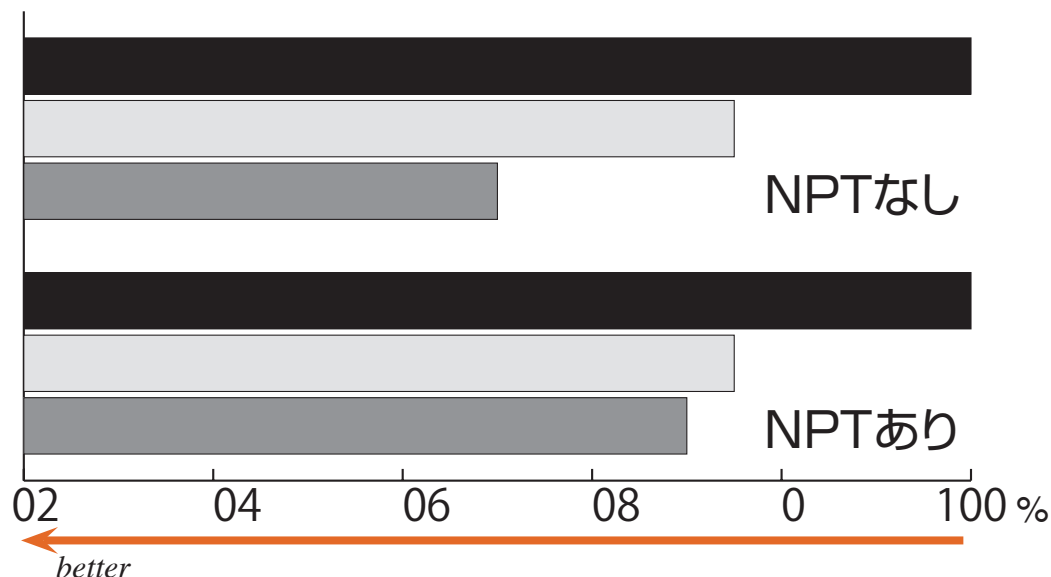
NPTなし

ベアメタル

PVM

VMware ESX

NPTあり



あるウェブサイトはVMware ESXで仮想化するだけで、同時リクエスト数が半分ぐらいいまで落ち込むことがあった。→ESXの問題ではなく、他でも大差ない。

コンテナ型仮想化のメリット

ハイパバイザ型に比べて、次のようなメリットがあります。

- リソースを使わない
 - カーネルは共通。プロセスが本当に利用するメモリ、CPUだけでよい。
- パフォーマンスボトルネックがほとんど無い!
 - グローバルゾーンから見たら1つのプロセスに過ぎない。
- ブートが速い
 - カーネルが起動しない分、ブート、リブートがとても速い



資源共有率が高いので、
1つのサーバにたくさん収容できる

コンテナ型のデメリット

ハイパバイザ型に比べて、次のようなデメリットがあります。

- コンテナごとの隔離性が低い
- 異なるOSの動作は不可能
- 比較的、互いの負荷影響を受けやすい
- 一部に動かないアプリケーションがある
- ライブマイグレーションなどの機能がない

カーネルが共通で資源共有率が高いことによる裏返し

Solaris 11 Zones の特徴

- OS標準の機能である
 - パッチなどによる提供ではないため、ノングローバルゾーン (仮想サーバ) とグローバルゾーン (収容機) のアプリケーションレベルの互換性が高い
 - その他の仮想化機能 (ネットワーク、ストレージ、パッケージシステム) との親和性
- リソース管理 (CPU制限、メモリ制限、カーネルリソース制限) が強力
 - 1つのコンテナの負荷の影響を、他のコンテナに波及させにくい。
- ネットワークスタックの委譲と仮想化
 - NAT-BOX、ファイルサーバ、L/B等々を作ることができる
- ファイルシステムの委譲と仮想化
 - ZFSによるファイルシステム単位の委譲。
- 旧バージョンのOSを混在可能
 - Solaris10 Zones
- マイグレーション (別のマシンへの移設) が可能

サーバとは何か？≠サーバ物理機

サーバとはなにか？ もう一度考え直してみた。

下記のような「サービスをする単位」なのではないか？

- WEBサーバ
- アプリケーション
- メールサーバ
- データベースサーバ
- ファイルサーバ
- ストレージサーバ
- ネームサーバ
- DHCPサーバ
- NAT
- ロードバランサー、ルータ、等々…もうかけない…

「サーバ」を実現する技術空間

「サーバ」を実現する技術空間ユニット、セットはなんだろう？

1. プロセスやユーザ単位
2. プロセスのグループ(SolarisではProjectやTask)
3. ネットワーク
4. ファイルシステム。OSのユーザ・ランド。
5. OSのカーネルのサービスモジュール。
6. ハードウェアとOSのドライバ群

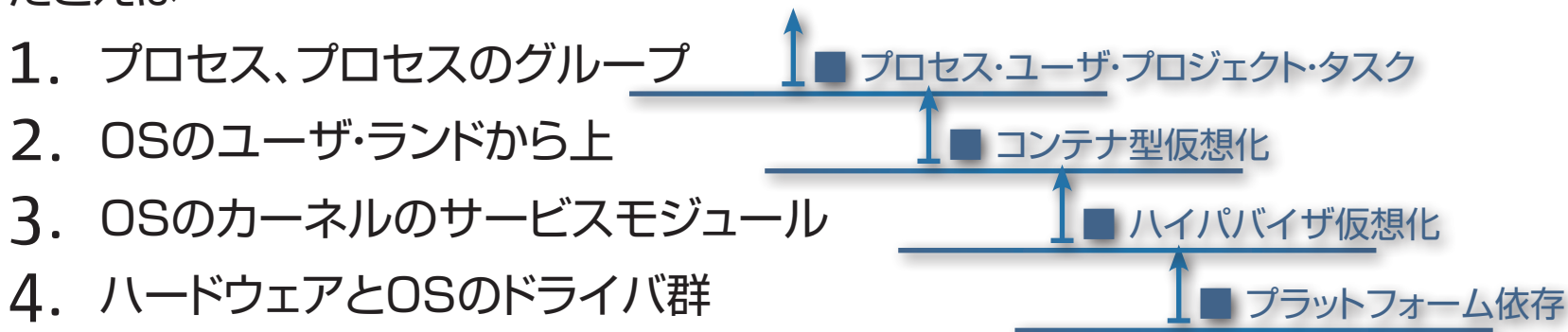
サーバ(サービスのエンジン)とは、あるサービスを実現するための単位。

これらのサーバ群に最適なリソースを与え、制御し、システムを正常に稼働・運用させることが、できれば技術的には何だって良い!

どの層で隔離・仮想化するのか？

どういう仮想化が必要なのかは、自分が必要な「サーバ」がどういう単位(隔離レベル)のものなのか?を考えることで決まります。

たとえば…



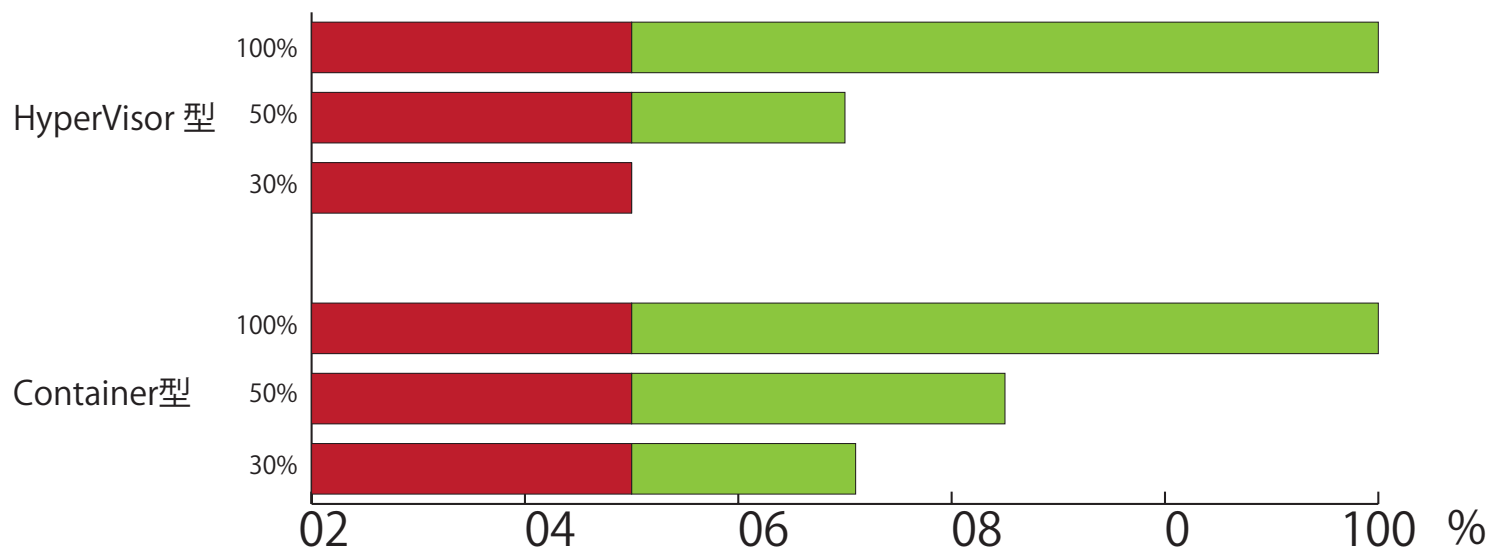
下に行くほど、一般的に仮想化コストが上がります。

何を「サーバ」としてとらえるかによって、
選ぶべき技術は異なることに注意が必要

仮想化アーキテクチャの違いによる臨界点の動作

仮想化アーキテクチャの違いで、リソース制限時の挙動が大きく異なる。


■ イメージ図(赤:SYSCALL、緑:ユーザ)



1つの仮想化ソフトウェアだけでクラウドを構築するのは、

利便性は高いが、効率は結構、悪い。

ハイブリッド・クラウドの ススメ



クラウドを実現するために、
Solaris 11 が支援する機能

ハイブリッド・クラウドのススメ

ハイブリッドに2つの仮想化アーキテクチャを、インフラに同居化させる。

x64のクラウド環境例:

- VMware ESX
 - 動作保証のOS、被動作保証ハイパバイザのカバレッジが広い。
- Solaris 11 Zones
 - リソース利用率が高い。

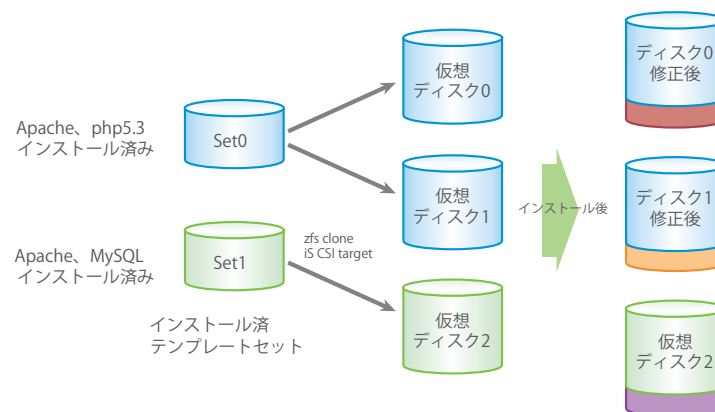
Solaris 11のZoneで動かせるものはなるべく動かし、OS依存のアプリはVMwareにそのOSを入れていく利用スタイルとなる。

リソース利用効率が上がり、省力化に繋がる。

ストレージの仮想化 (ZFS)

● ZFSによる管理

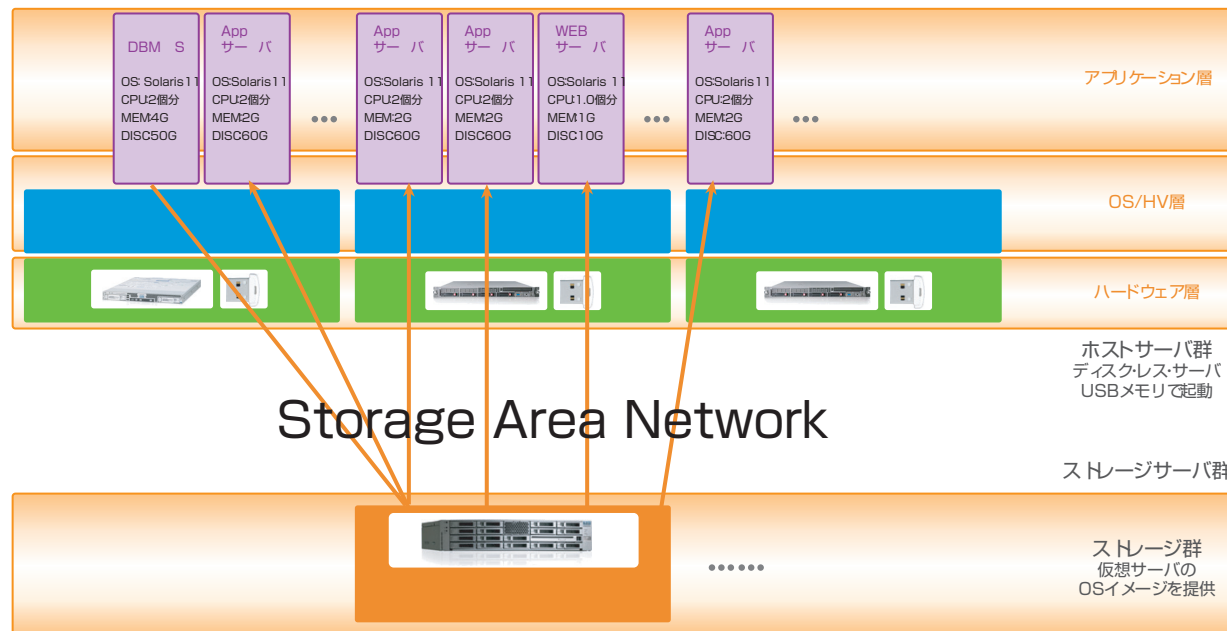
- ZFSはツリー構造でデータセット (ファイルシステム) を管理している。
- 仮想サーバ毎に、ZFSを委譲して、利用することが可能。
- 仮想サーバ毎の、利用量のCAPの制御ができる。
- スナップショット、ロールバック、クローンなどができ、仮想化と相性が良い。
- スナップショット毎に、バックアップを簡単に取ることができる。
- ただし、仮想サーバ毎のストレージのアクセス頻度管理は難しい (知る方法はあるが、制御は難しい)



非常に高価なアプライアンス機器と同じ機能を、PCサーバでも実現できる

SAN 機能

- SANを利用した管理
 - iSCSIやFCなどのTARGET機能 (COMSTAR)。ZFSと連携して、高価なDAS並の機能を標準で利用することができる。
 - SANを利用し、収容サーバをディスクレスにすると、さらに仮想化の隔離性が増す。
 - フェイルセーフのための多重化も簡単にできる



Networkの仮想化

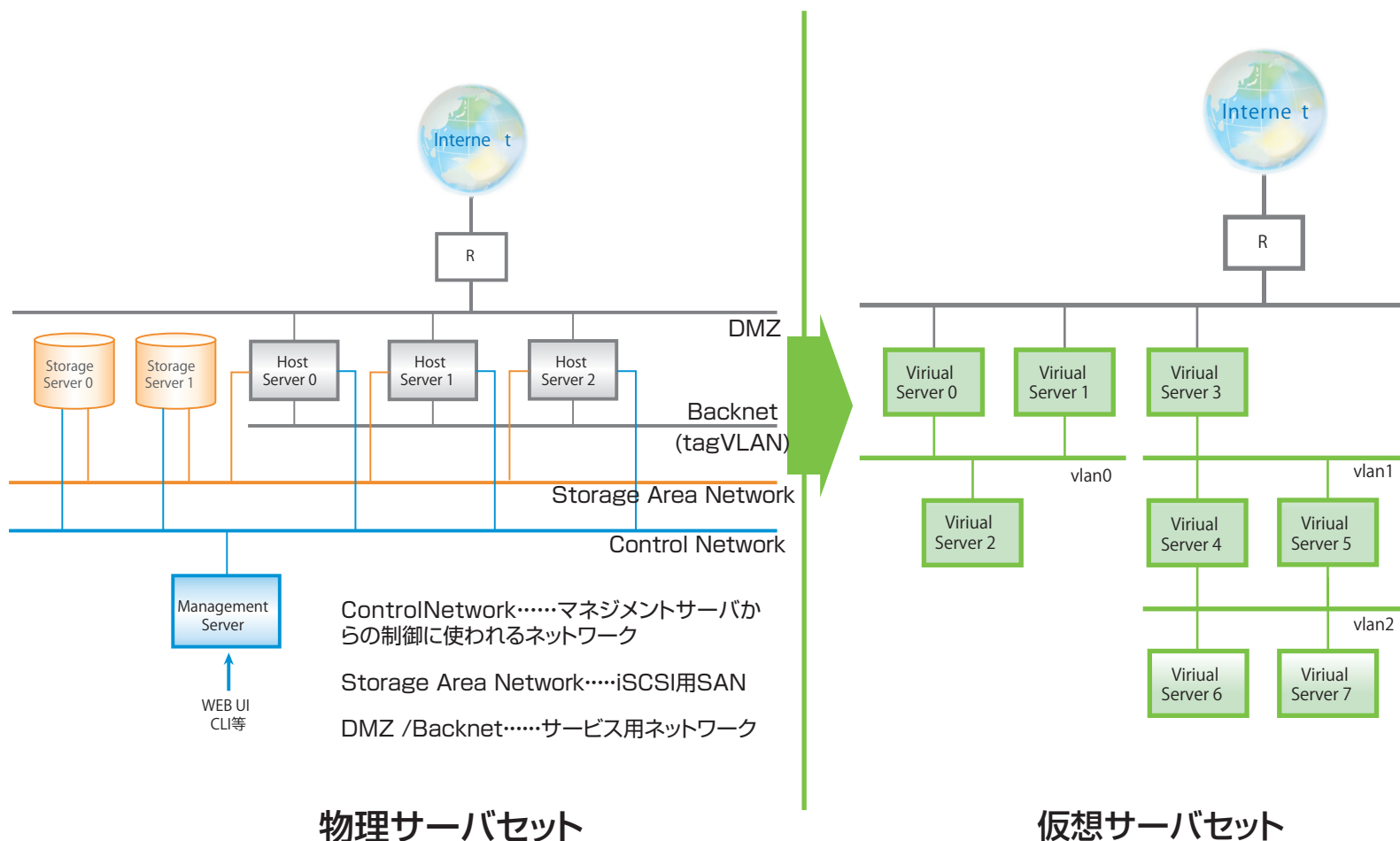
- dladm(CROSSBOW,CLEARVIEW)を使った仮想ネットワークの管理
 - 仮想サーバ毎に、仮想NICを作成して委譲。
 - 仮想NICの帯域制御。
 - spoofingプロテクト。
 - CPU負荷の制御。
 - IPの固定制御。
 - 仮想Switch (Etherstub) の作成
 - 仮想Bridgeの作成。
 - VLAN、フィルタ、ルーティング等
 - 仮想サーバへのMACアドレスの貼り付け。
 - 帯域の測定、監視など。

仮想ネットワークとスイッチのtagVLANを組み合わせることで、
全体でネットワークを仮想化することができる。

また、固有の機能を使わないので別のシステムとの連携も。

物理 Network と仮想 Network

これらを、物理結線のし直しや、スイッチの再起動なしに行える



Solaris11 のクラウド環境


メリット

- 実際、安い。
 - これらの基本的な機能が、Solaris11の標準機能で行うことができる。
- Interoperabilityが高い。
 - iSCSIや、tagVLAN、LinkAggregationなど、基本的には標準的な技術なので、別のOSとの組み合わせが行いやすい。

ベネフィット

- WIKIPLUSのサービスレベルの向上、ユーザの収容率、同時受付数の向上。
- その他のサービスの収容が可能。
- Solaris11のインフラを案件単位で他社に貸し出し可能。
- VPSサービスも視野に。

Solaris 11が WIKIPLUS 2に与えた その他のベネフィット

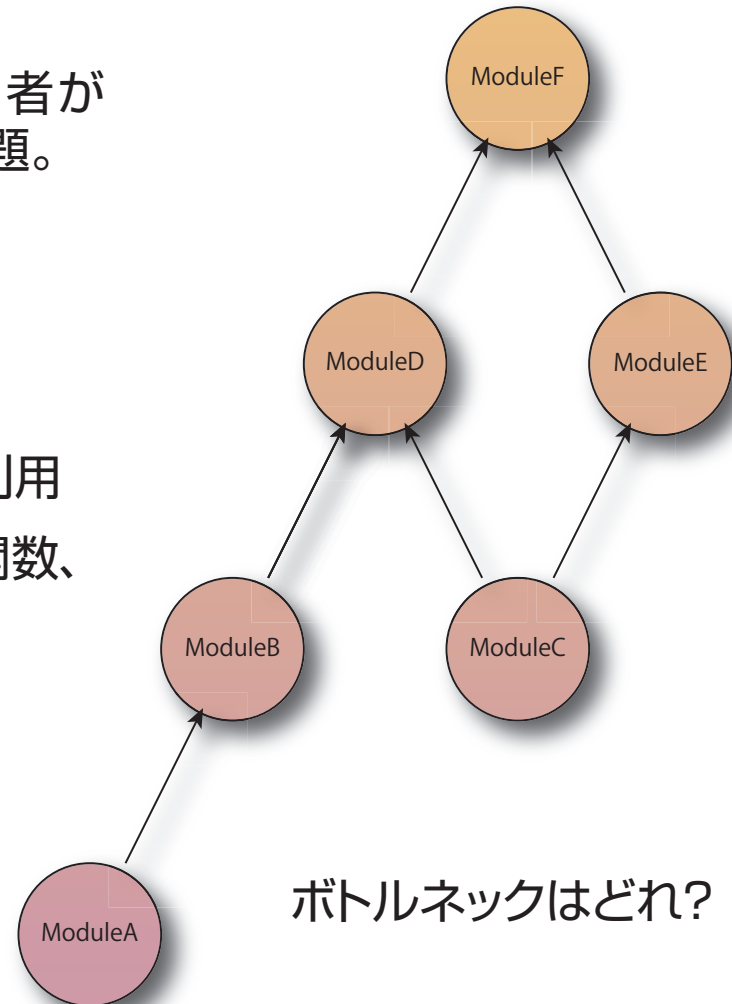


Solaris だからこそ
できたこと

D-Trace による最適化

- 接ぎ木されたプログラム
 - バージョンアップを続け、さらに担当者が変わり、接ぎ木されたプログラムの問題。
 - 通常、動いてるモジュールは触らない。
 - カプセル化されると更に分からない。
- D-Trace Toolkit(/opt/DTT/php)を利用
 - サービスサーバで、よく呼ばれるphp関数、時間がかかるphp関数などを発見
 - 不可解な呼び順で呼ばれる関数など

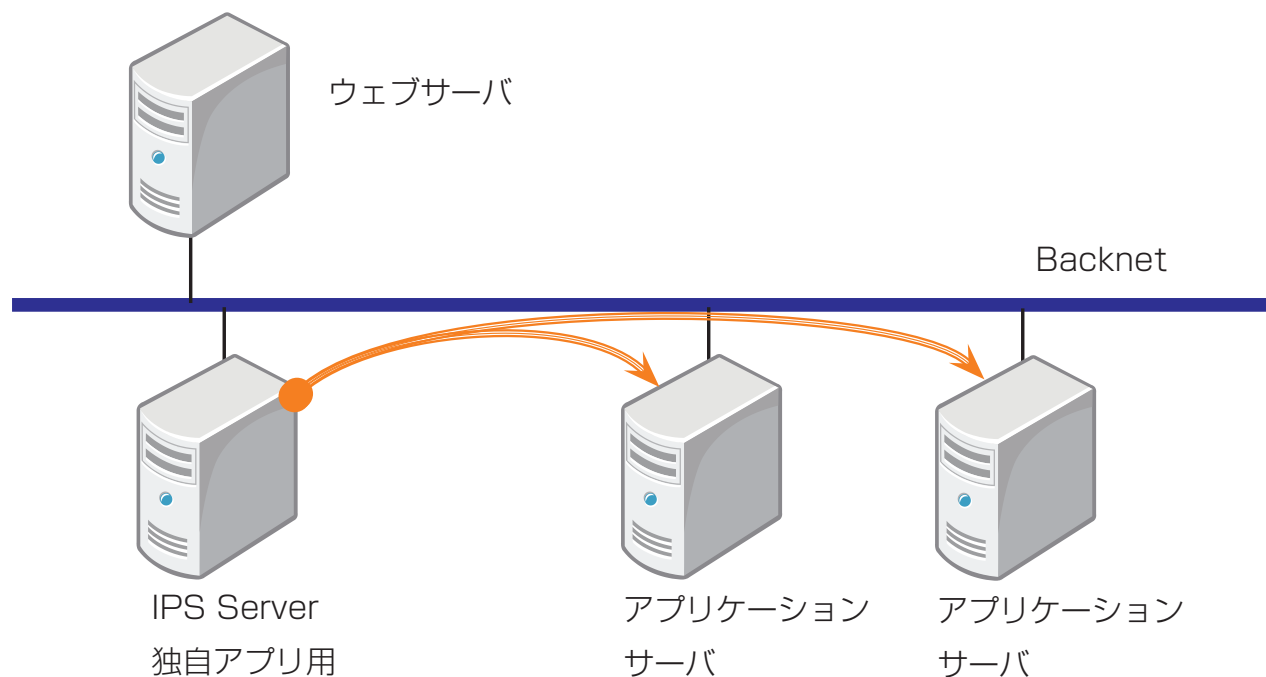
リファクタリングすべき順序が明確化!



ボトルネックはどれ?

パッケージシステムによる同期

- Private Network内のパッケージ配布サーバから、ソフトウェアのアップデートができる。
- これによって、Solaris 11のパッケージ同期や、ミドルウェアのインストール状態の調整、ソフトウェアのリビジョンの調整ができるようになった



まとめ～収容率～

クラウド時代は、リソース利用率=コストと顕著になる。

そのため、速度、収容率はより顕著。

- クラウドの為のシステムはSolaris 11の機能を組み合わせることによって、効果的に作ることができる。
- これらの機能のinteroperabilityは高いので、物理機群や、別の仮想化システムとも比較的親和性を持って融合することができる。
- Solaris Zonesの仮想化コストは低く、資源共有率は高い。そのため、収容率を上げることができるため、サービスのコストダウンに繋がる。また、速度による問題が起きにくいので、負荷が高いシステムにも有用。
- これらの仕組みはプライベートクラウドにも流用が容易。
- Solarisは様々なシステムのボトルネック解析が容易。

ご静聴ありがとうございました。

ジャストプレイヤー株式会社

代表取締役社長

瀧 康史

<http://www.justplayer.co.jp/>

TEL: 054-203-5535

Mail: [info @ justplayer.com](mailto:info@justplayer.com)