
3大仮想化製品 徹底比較 ～ メーカーが言えない、本当のことを教えます ～

2010年2月18日

インフラソリューション推進部
照井 一由

伊藤忠テクノソリューションズ株式会社

2009年4月1日現在

会社名

伊藤忠テクノソリューションズ株式会社 (略称 CTC)

英文社名

ITOCHU Techno-Solutions Corporation

本社所在地

〒100-6080 東京都千代田区霞が関3-2-5 霞が関ビル
TEL: 03-6203-5000(代)
URL: <http://www.ctc-g.co.jp/>

代表者

代表取締役社長 奥田 陽一

創立

1972年(昭和47年)4月1日

資本金

21,763百万円

社員数

7,095名

事業内容

コンピュータ・ネットワークシステムの販売・保守、ソフトウェア受託開発、情報処理サービス、科学・工学系情報サービス、サポート、その他



常に時代に先駆け海外最新技術を先進的に取り込み、日本のオープンシステム市場をリードしてきました。



1985 1990 1995 2000 2005 2009

ホストの時代

オープンシステムの時代

ネットワーク(Internet)の時代

仮想化の時代

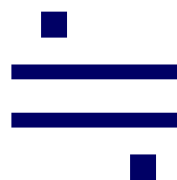
仮想化市場動向

CPU性能の大幅な向上

2004年



ほぼ、同じ性能



2009年



UNIX ハイエンドサーバ : 32CPU
CPU性能指標 : 253
価格 : 数億円

x86 Server : 2CPU
CPU性能指標 : 237
価格 : 数十万円

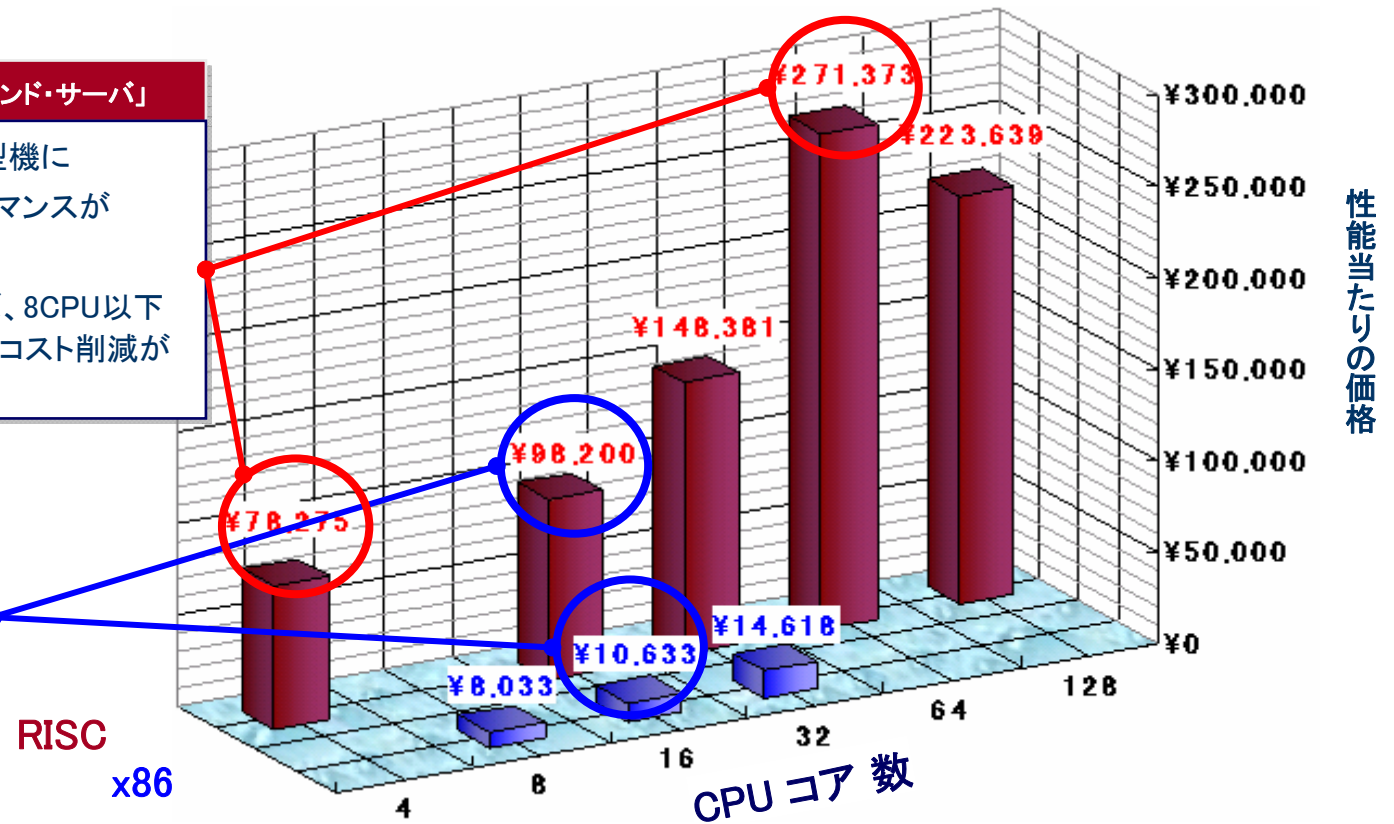
高い x86 Server のコストパフォーマンス

α. 「ハイエンド・サーバ」から「ローエンド・サーバ」

- RISC機は16CPU以上の大型機になると、著しくコストパフォーマンスが低下する。
- 要求性能上問題がなければ、8CPU以下のサーバを選択することで、コスト削減が可能

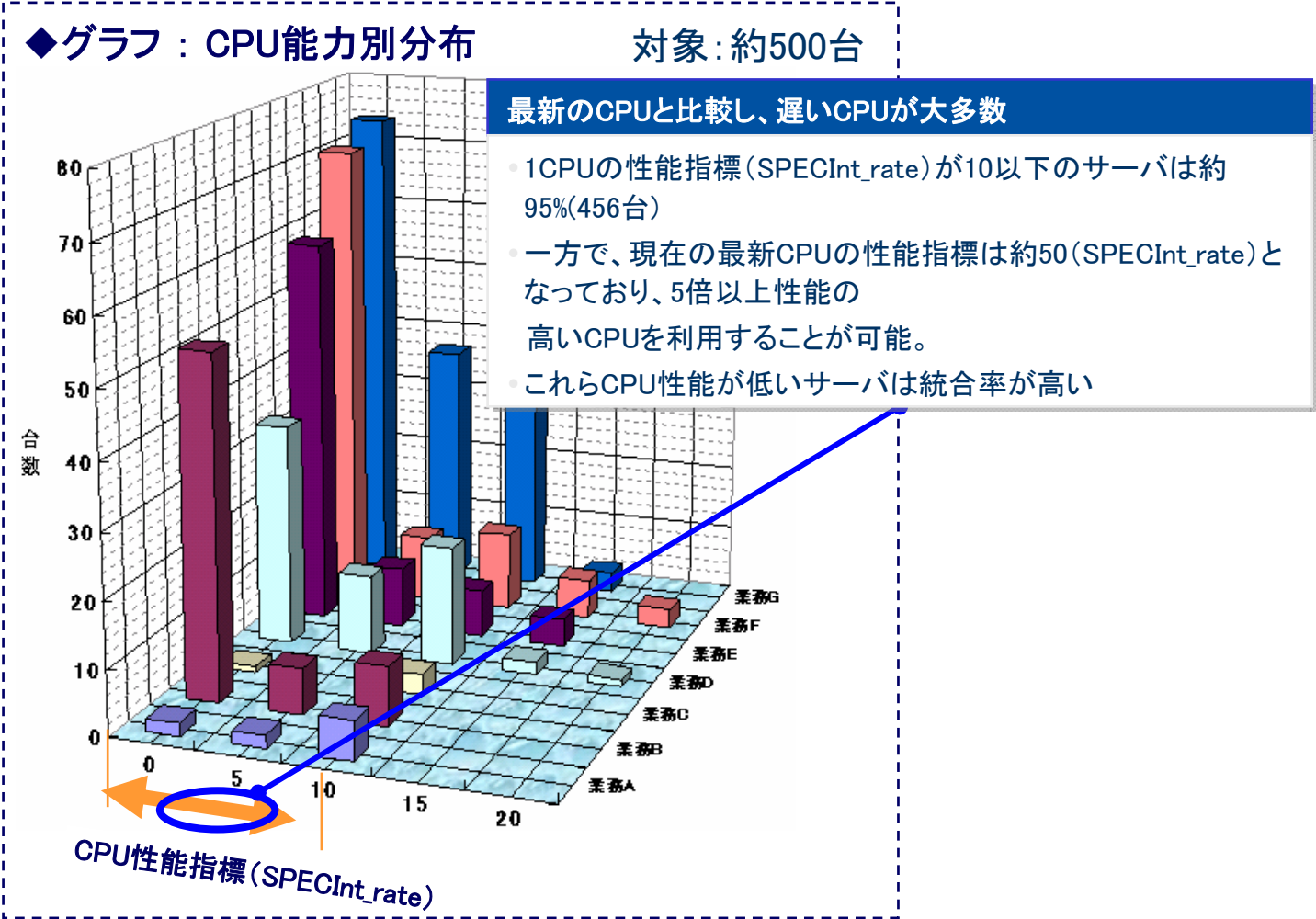
β. 「UNIX/RISCサーバ」から「x86サーバ」

- 同じ処理を実行しようとした場合、RISCの方が、より多くのコストが必要



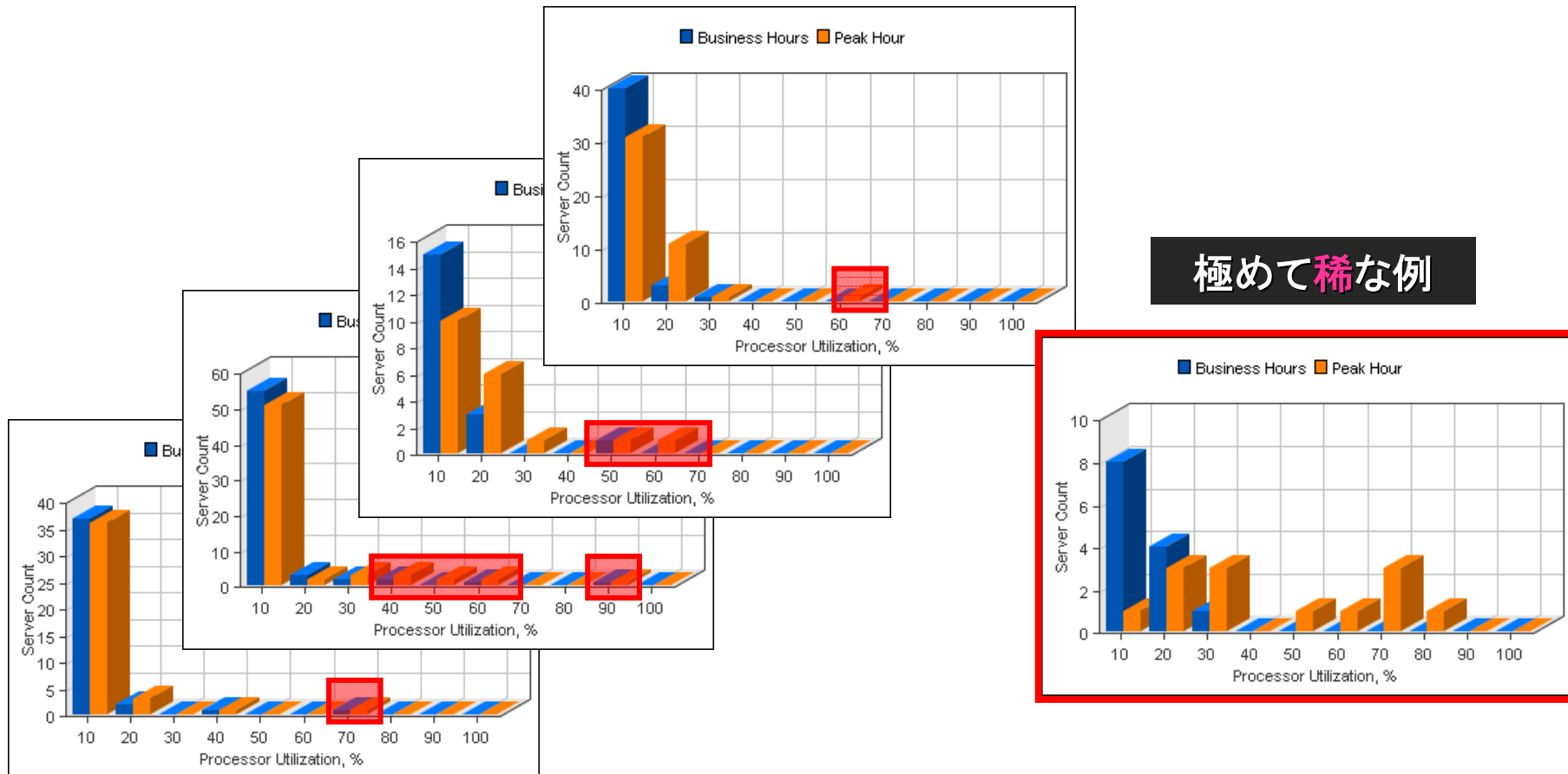
企業内の多くの既存サーバがロースペック

実際にアセスメントしてみると



CPU使用率は驚くほど低い（使用率10%以下）

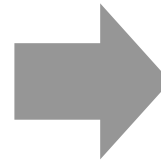
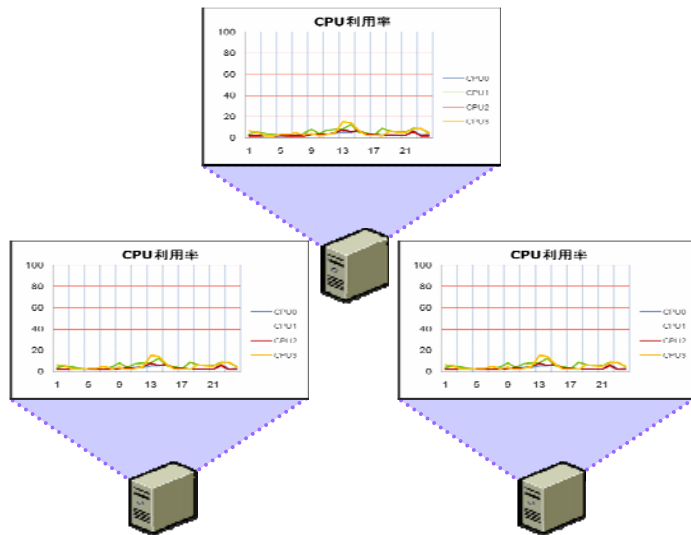
実際にアセスメントしてみると



ロースペックかつ、CPU利用率の低いサーバを、最新のサーバに仮想化技術で統合

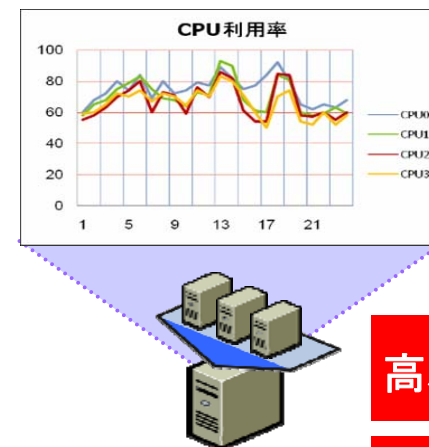
現状：CPU能力が余剰

平均10%未満の使用率



処方箋：アプリケーションの枠を越えたリソース共有

平均60 - 80%の使用率



高パフォーマンス

低価格

アセスメントの肝：サービスレベル定義による、投資コストの全体最適化

サービスレベルの分類を定義

サービスレベルの判定基準を定義

| サービスレベル | コスト比 | サービスレベル判定基準 | | | | サービスレベル判定基準(参考) | | | | システム構成 | | | | |
|---------|------|-----------------------------|----------------------------------|--|-------------------------|--------------------------------|---------------------------|---|--|----------------------------------|--------------------|------------|-------------|-------------|
| | | 計画メンテナンス実施基準 | 障害箇所の許容性 | システム拡張基準 | 障害時サービス停止時間の基準 | システムの利用者 | データの再現性 | 収益への影響 | 社会的影響 | システムの例 | サーバー冗長化構成 | ストレージ冗長化構成 | 担保可能な計画停止時間 | 担保可能な障害停止時間 |
| SA | 100 | 計画メンテナンス作業におけるサービス停止が許容されない | 一般的に考えられる全ての障害箇所に対して自動復旧できる対策を行う | 将来に対して性能拡張性の確保が必要であり、スケールアウトの分散構成が可能である。 | 数十秒以内に、サービスが復旧しなければならない | 顧客の大多数 | 損失したデータの再現が不可能、もしくは極めて困難 | 収益基盤の根幹に多大な影響を及ぼす | ・公共性が極めて高く、社会的信頼が失墜する ・最悪の場合責任追及が発生する可能性がある | 電話通信網システム グローバル生産管理システム | 多重化 | | 2時間以内 | 数十秒 |
| AA | 50 | 計画メンテナンス時は、サービス停止を行うことができる | | | | 顧客の大多数、または一部 ・企業的全組織ならびに従業員 | 損失したデータの再現が困難かつ、再現性が求められる | ・収益基盤の根幹、または一部に多大な影響を及ぼす ・企業内の基幹業務であり、間接的に収益への影響が大きい | ・公共性が高く、社会的信頼が失墜する ・一部の顧客から責任の追及が発生する可能性がある ・社内的な責任問題に発展する | ATM B2C取引システム B2B基幹系取引システム | アクティブ-アクティブ | | 4時間以内 | |
| A | 50 | | | | | | 損失したデータの再現が困難かつ、再現性が求められる | ・顧客からの信頼が大きく失墜する ・社内的な責任問題に発展する | | B2B取引システム 社内メールシステム | ホットスタンバイ | | 8時間以内 | 数分 |
| B | 30 | 計画メンテナンス時は、サービス停止を行うことができる | ハードウェアに対する障害に対してのみ自動復旧できる対策を行う | ハードウェアに対する障害に対してのみ自動復旧できる対策を行う | | ・特定の顧客 ・企業的全組織、または一部 | 損失したデータの再現が容易かつ、再現性が求められる | ・特定顧客からの収益に軽微な影響を及ぼす ・企業内の一部業務であり、収益への影響が小さい | ・特定顧客からの信頼が失墜する ・社内からのクレームが発生する | 社内月例会計システム | コールドスタンバイ(仮想化HA機能) | | 8時間以内 | 数分 |
| C | 20 | | | | | 企業における一部の組織 | 損失したデータの再現性を必要としない | 企業内の一部業務であり、収益への影響が小さい | 社内からのクレームが発生する | 部門Webサイト 部門基幹システム | コールドスタンバイ(手動) | | 1日以内 | 数時間 |
| D | 10 | | | | | 企業における一部の組織、または一部の従業員 | | 直接・間接問わず収益への影響が発生しない | 影響がない | 部門非基幹システム | シングル | | 数日以内 | 数日 |

SAMPLE

サービスレベルとコストの関係を明確化

サーバー・ストレージを組合わせた標準アーキテクチャ・モデルを定義

SOIDEAL-インフラ統合シミュレーション 最終報告書サマリ

1. 目的

1. インフラ老廃化対策
FY10には、過去に構築した多くのIT機器が保守切れを迎え、その対策が急務。
2. より低コストなインフラ構築手法の選定
一方、老朽化対策には多くのコストが必要。現在の経営環境を鑑みると、最小限の投資で老朽化対策を実施できる手法を検討する必要がある

従来通りの方法で老朽化対策を実施した場合に発生するITコスト

| | |
|------------|--------|
| ITインフラ総コスト | 426百万円 |
| ハードウェア費 | 143百万円 |
| ソフトウェア費 | 8百万円 |
| インフラ保守費 | 25百万円 |
| 場所・建物費 | 133百万円 |
| 電力費 | 115百万円 |

2. 課題分析

- 現行サーバ台数：447台
- 低スペックサーバ多数：平均SPECint41（最新機種は100/CPU程度）
- CPU利用率が低い：平均CPU利用率14%（ピーク時平均38%）

3. 方策

1. 仮想化技術を用いた統合
仮想化技術とは、複数のOSを1台のサーバ上で同時に稼働させる事が出来る技術。大量のIT機器を少数の機器に統合する。

2. IT機器の性能向上効果享受
現在、定価50万円で購入可能なサーバの性能は、「CPU:201 (SPECint4), メモリ: 28GB, HDD: 144GB」である。最新機種への統合で集約率を高める。

3. アーキテクチャの標準化
構築した統合ITインフラを今後も陳腐化させない為、アーキテクチャを標準化し、今後の利用基準を作成する。

4. 対象範囲

447台中343台(76%)を統合対象とする
※ 検討した結果、技術的制約により一部を除外

| システム名 | 統合対象 | 全体 |
|----------|------|-----|
| System-A | 227 | 320 |
| System-B | 42 | 50 |
| System-C | 32 | 32 |
| System-D | 19 | 22 |
| System-E | 13 | 13 |
| System-F | 10 | 10 |

5. アーキテクチャ見直し

1. プラットフォーム変更によるコスト削減
SPARC Solaris → x86 Solaris (対象65台)
2. 構成の見直しによる性能の強化
非冗長構成 → 冗長構成

6. モデル構成の標準化

以下の2つの内容を実施
1. 要件毎にサービスレベルを標準化
2. サービスレベルに応じた標準化設計ガイドラインを定義

| システムが採用するOS | システムが採用するOS | | |
|-------------|---------------|-------------|---------|
| | Linux | x86 Solaris | Windows |
| 障害&計画停止不可 | Active-Active | | |
| 障害停止不可 | Active-Active | 32台 | 71台 |
| 30秒以内に復旧 | Hot Standby | | 175台※ |
| 15分以内に復旧 | 仮想化HA | | 17台 |
| 30分以内に復旧 | Cold Standby | | 48台 |
| 単一 | Single | | |

7. 統合効果試算

5年間累計での統合効果は以下の通り

- TCO削減効果：287百万円
- ラック数削減効果：▲297ユニット
- CO2削減効果：▲4,635Mwh

| | ハードウェア費 | ソフトウェア費 | インフラ保守費 | 場所・建物費 | 電力費 | TCO | コスト削減効果 |
|-----------|---------|---------|---------|--------|--------|--------|---------|
| 通常リブレース | 143百万円 | 8百万円 | 25百万円 | 133百万円 | 115百万円 | 426百万円 | |
| Vmware統合 | 50百万円 | 6百万円 | 53百万円 | 20百万円 | 18百万円 | 148百万円 | ▲287百万円 |
| Hyper-V統合 | 60百万円 | 9百万円 | 49百万円 | 25百万円 | 20百万円 | 163百万円 | ▲262百万円 |

環境負荷

| | CO2排出量 | CO2削減効果 | 電力消費量 | CO2削減効果 | サーバ台数 | 台数削減効果 | ラック数 | ラック削減効果 |
|-----------|---------|----------|----------|-----------|-------|--------|------|---------|
| 通常リブレース | 1,854千t | | 5,470Mwh | | 343台 | | 343U | |
| Vmware統合 | 283千t | ▲1,571千t | 835Mwh | ▲4,635Mwh | 24台 | ▲319台 | 46台 | ▲297台 |
| Hyper-V統合 | 326千t | ▲1,524千t | 973Mwh | ▲4,498Mwh | 27台 | ▲316台 | 58台 | ▲285台 |

8. 実行計画

システムの更改時期を元に、リブレース年度を決定。
1. 2009年9月より設計作業を開始し2010年8月の統合環境運用開始を予定
2. 2012年度末に統合が完了

10

仮想化・統合化のアセスメント例

| 顧客 | | 顧客規模(連結) | | 統合前 | | 統合後 | | 統合効果(%) | |
|----|---------------|-------------|---------|------|--------------|-----|-------|---------|------|
| No | 顧客名 | 売上高 (億円) | 社員数 | 台数 | TCO (百万円) | 台数 | TCO | 台数 | TCO |
| 1 | 関東自動車工業株式会社 様 | 7,000 | 7,000名 | 約100 | - | 約20 | - | -80% | -40% |
| 2 | 某 住宅建設業者 様 | 6,000 | 13,000名 | 89 | - | 23 | - | -74% | - |
| 3 | 某 ITサービス業者 様 | 3,000 | 2,000名 | 37 | - | 12 | - | -68% | - |
| 4 | 某 公共事業者 様 | 20,000 | 6,000名 | 453 | 2,659 | 75 | 1,093 | -83% | -59% |
| 5 | 某 公共事業者 様 | 15,000 | 13,000名 | 769 | 17 | 139 | 13 | -82% | -24% |
| 6 | 某 運輸業 様 | 4,000 | 2,000名 | 172 | - | 22 | - | -87% | - |
| 7 | 某 製造業者 様 | 7,000 | 11,000名 | 133 | - | 17 | - | -87% | - |
| 8 | 某 官公庁 様 | - | - | 113 | - | 20 | - | -82% | - |
| 9 | 某 オンライン販売業者 様 | 100 | 300名 | 45 | - | 19 | - | -58% | - |
| 10 | 某 商社 様 | 124,000 | 49,000名 | 39 | - | 15 | - | -62% | - |
| 11 | 某 流通業 様 | | 8,000名 | 18 | - | 3 | - | -83% | - |

【余談】 本当に仮想化して問題ないのか？

1. ミドルウェア・アプリケーションの向き/不向き
2. ミドルウェア・サポートの懸念

1. ミドルウェア・アプリケーションの向き/不向き

A. そもそも動作しない問題

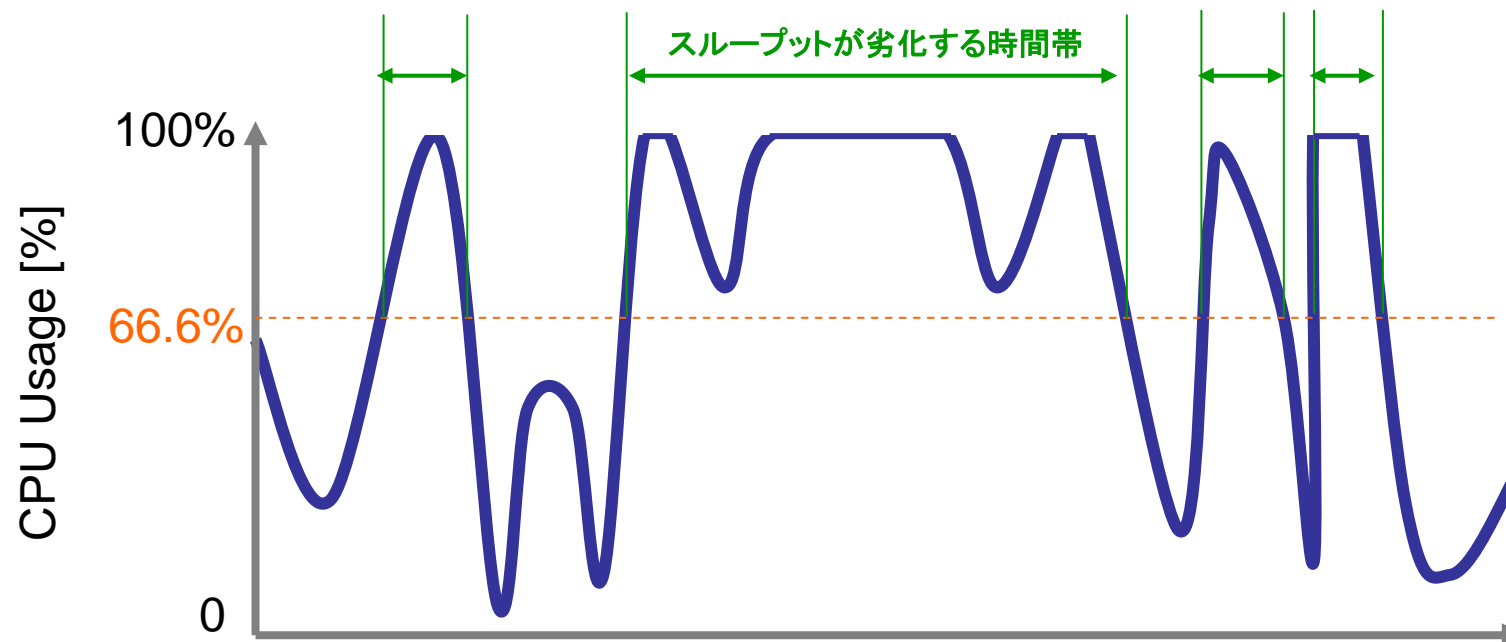
- ゲストOSのバージョンが対応していない
eg.) Windows NTであれば、4.0 Service Pack 6a 以降
- 特殊なデバイスを利用している

B. パフォーマンスが劣化（すると一般的に言われている）

- データベース I/O
- コンパイル
- 大量の描画処理

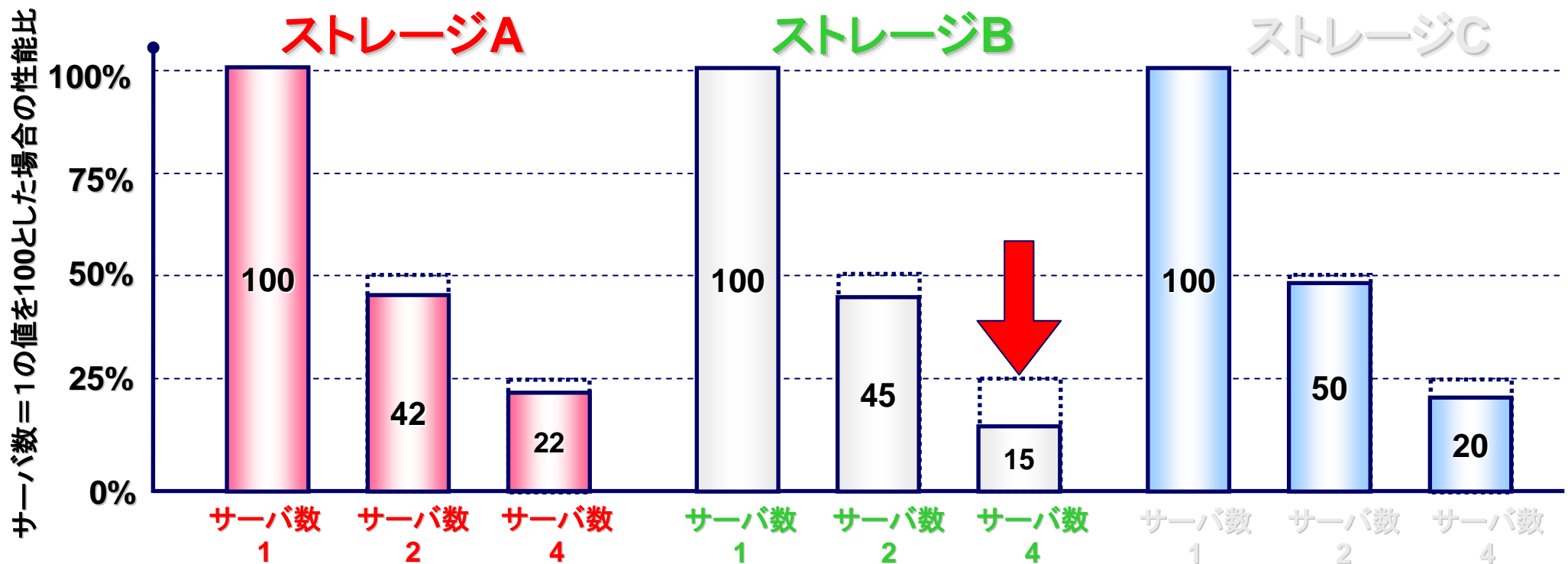
- ✓ 例えば、仮想化のオーバーヘッドによるCPUのオーバーヘッドが150%だとすると、アプリケーションのスループットが影響を受けるCPUの利用率の閾値は
= $100\% \div 150\% = 66.66\%$
- ✓ アプリケーション負荷により、CPU使用率が66.66%以上になる場合には、物理環境に比べスループットは劣化する。
- ✓ 反対に66.66%以下のCPU利用率であれば、CPUに空き(44.44%以上)があるため、スループットは劣化しない

CPUの利用率と、スループットに対するオーバーヘッドの関係 (イメージ図)



- ✓ ストレージの種類によっては、接続するサーバ数を増やすと、極端に能力が低下するものもある。
- ✓ このような問題が顕在化しないよう、機種選定や、設計を考える必要がある。

弊社独自検証結果

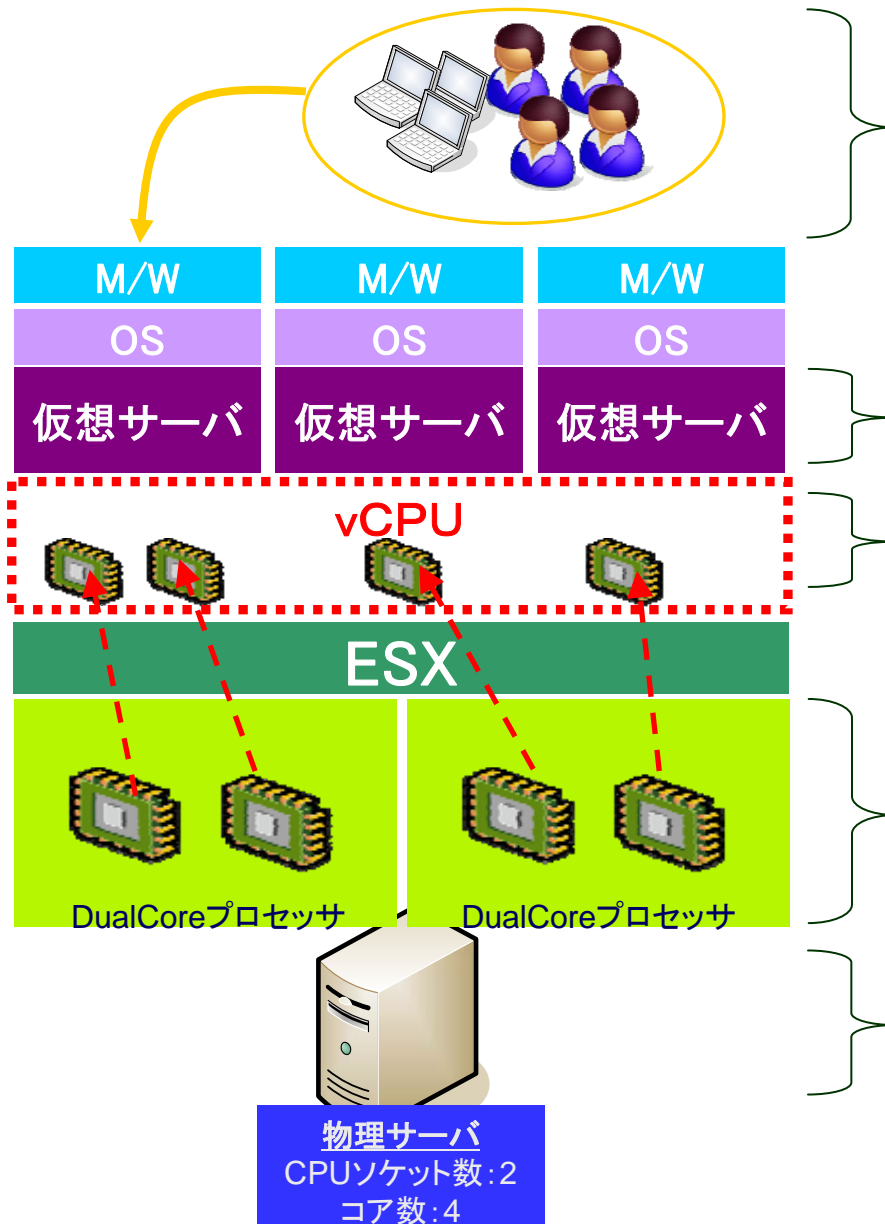


2. ミドルウェア・サポートの懸念

- 調査方法は？
- 調査結果は？
- 製品ごとの差はあるのか？

仮想化環境における ミドルウェアライセンス体系

仮想化ライセンス課金体系 パターン



ユーザ・クライアント課金
 当該ミドルウェアを利用する「ユーザ数」や「デバイス数」に応じた課金体系

仮想サーバ課金
 当該ミドルウェアがインストールされる「仮想サーバ数」に応じた課金体系

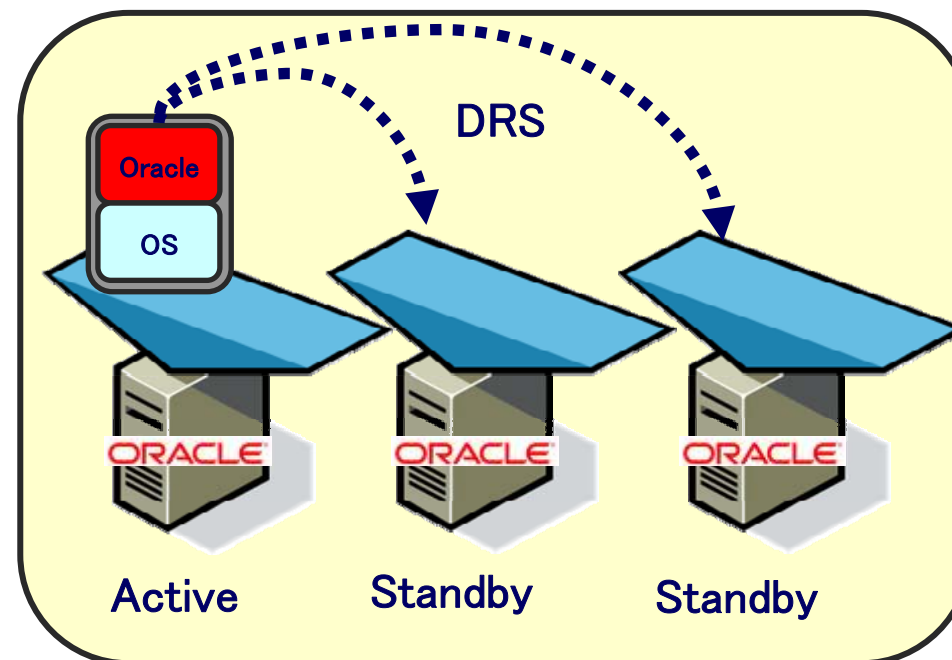
vCPU課金
 当該ミドルウェアがインストールされる仮想サーバに割り当てられる「vCPU数」に応じた課金体系

物理CPU課金
 当該ミドルウェアがインストールされる仮想サーバが稼動するホストサーバに実装された物理CPU数に応じた課金体系

物理サーバ課金
 当該ミドルウェアがインストールされる仮想サーバが稼動するホストサーバの数

物理サーバ
 CPUソケット数: 2
 コア数: 4

Active、Standbyは関係なく、全ての物理マシン分のライセンスとなる。
DRS機能は、障害が起きた物理マシン上の仮想マシンをリソースが余っている物理マシンへ自動で移動させるため、Oracleが導入された仮想マシンがどの物理マシンへ移動するか確定されないためである。

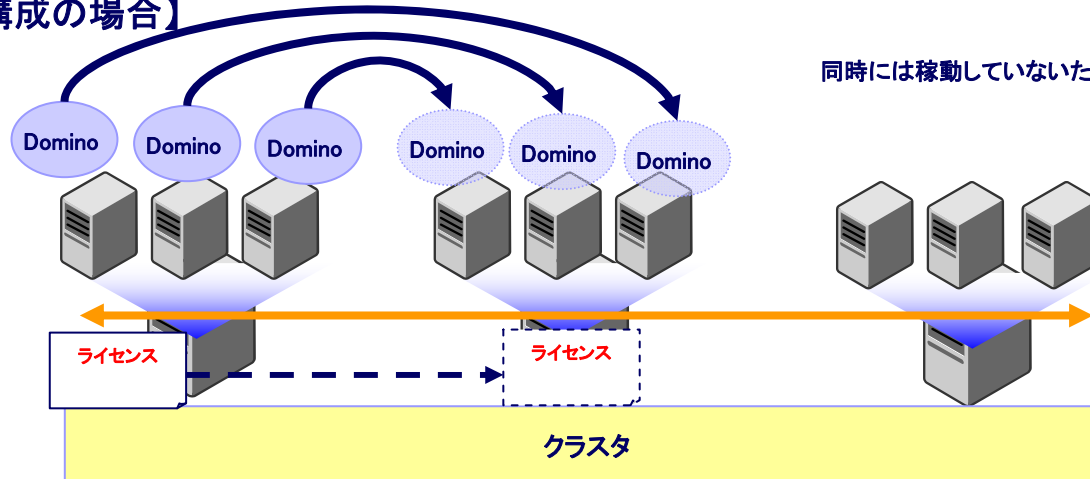


図のように物理マシンが3台で構成している場合
物理マシン 3台分のライセンスが必要

【例】VMware環境を考慮したミドルウェアライセンスの考え方(Domino、WAS)

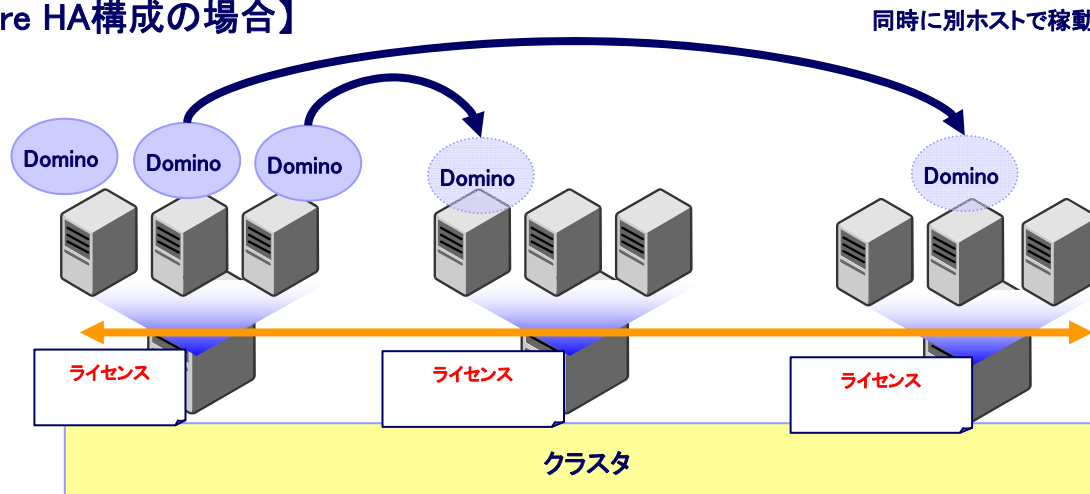
Domino、WASの場合のライセンスの考え方を下記の通りである。

【DRS構成の場合】



※DRSの場合、ルール設定により複数の仮想マシンを同一のホストに移動するように設定可能。

【VMware HA構成の場合】



※VMware HAの場合、ホスト指定は可能だが、何らかの理由(通信不能、リソース不足)により指定ホストに切り替われない場合は同一クラスタ内の別ホストで起動する。

関東自動車工業様 ～ 統合結果 ～

- ✓ サーバ仮想化技術は普及期へ
- ✓ サーバ仮想化導入の課題は徐々にクリアされつつある
- ✓ 現時点ではVMwareがデファクトだが、用途によっては、Hyper-VやXenの利用も検討

CTC

▼ *Challenging Tomorrow's Changes*