仮想化環境の設計手法
～ 基礎編 ～

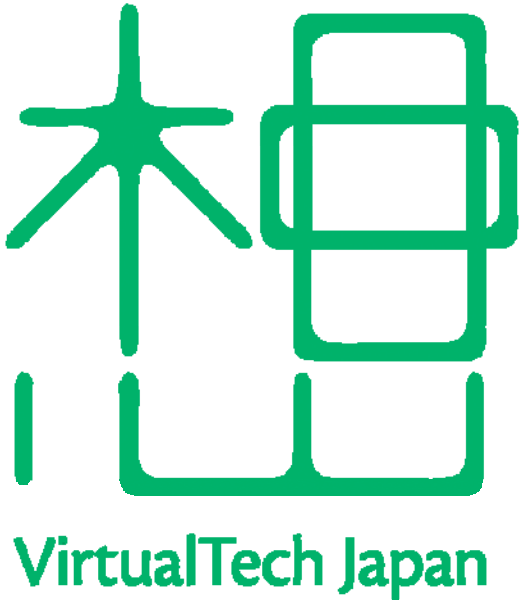
日本仮想化技術株式会社
代表取締役社長兼CEO 宮原 徹
miyahara@VirtualTech.jp

VirtualTech Japan

本日のアジェンダ

- 仮想化環境の設計手法
- ハードウェアの選定
- 省電力サーバーの検討
- ストレージの選定
- ベンチマークによる性能検証

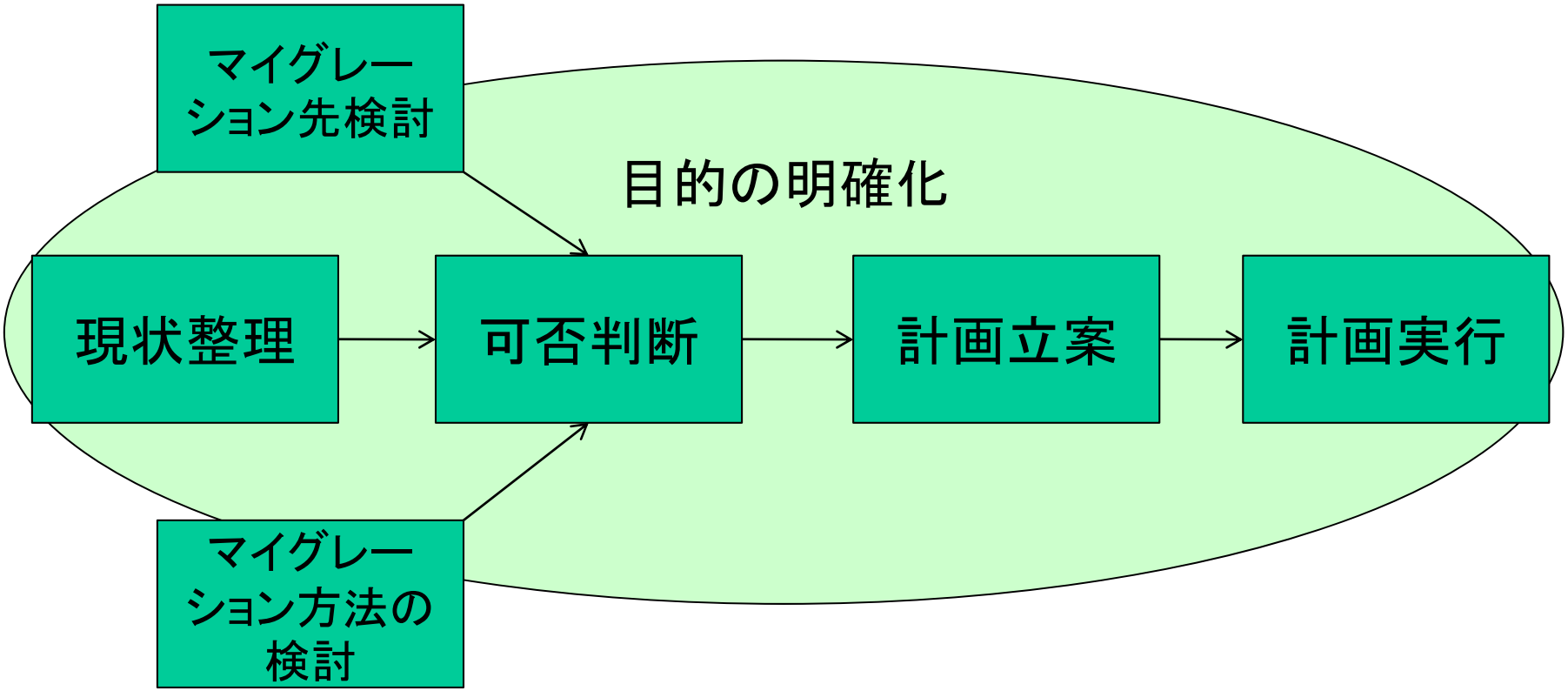
仮想化環境の設計手法



仮想化環境設計のポイント

- 目的の明確化
  - 既存環境移行 ← 現状こちらの方が多い
  - 新規システム構築
- 何を優先するのか
  - コスト：導入コスト、ランニングコスト
  - 機能：運用管理、可用性
  - 性能：実行速度、処理容量

仮想化マイグレーションのプロセス



VirtualTech Japan

「見える化」された目標の例

システム寿命 2年延長

ハードウェア台数を2分の1に削減

ラック本数 3分の2に削減

システムダウンタイム 5分以内

システム復旧時間 1時間以内

VircualTech Japan

設計フェーズ

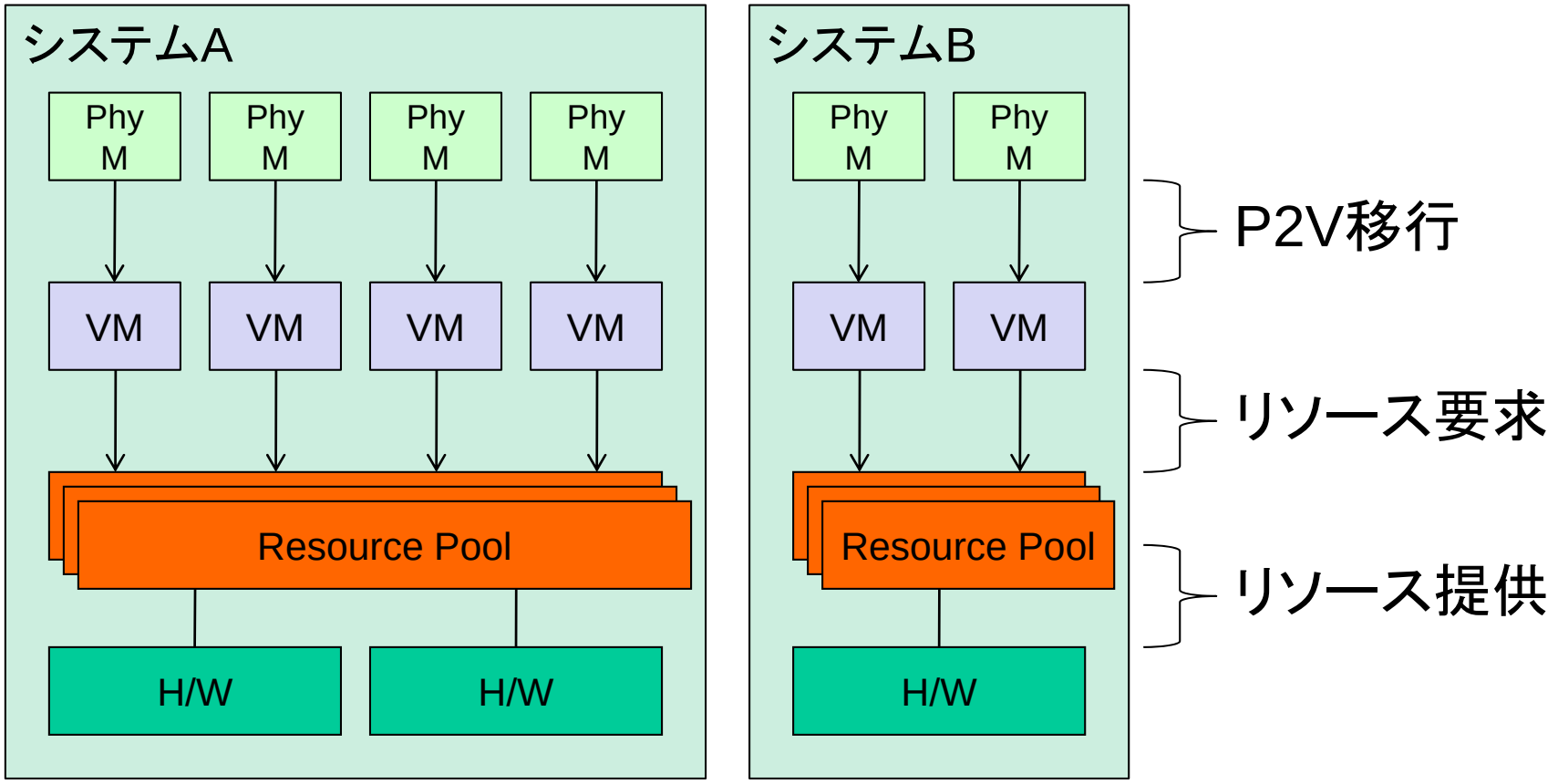
論理設計
- 要求仕様を論理システムにマッピング
- システムを機能単位で捉える

仮想化設計
- 論理システムを仮想化にマッピング
- システムを仮想マシン単位で捉える

物理設計
- 仮想システムを物理H/Wにマッピング
- システムのサイジングや冗長化など

Point!! ギリギリまで物理的なことは考えない

仮想化→物理マッピング



複数H/WでResource Poolを
構成することで冗長性を確保

要求リソースが限定的で
冗長性が不要な場合には
単独H/WでRPを構成

VirtualTech Japan

仮想化物理マッピングの注意点(1)

- リソースプールはN台のH/Wで構成される
  - 最低3台で構成するのが冗長化の観点から望ましい
  - 冗長性を排除するのであれば1台での構成も可能
  - 同一H/W上に複数のリソースプールを作成することも可能
- 最終的な物理設計を行うまで仮想マシンと物理マシンの紐付けは行わない
- 仮想化環境にとって「位置の透過性」が重要
  - 仮想マシンはリソースプール内の「どこか」で動いていると考える
  - どのH/Wで動いているかは考慮しない
  - 障害発生時はリソースプール内で相互にカバーする

仮想化物理マッピングの注意点(2)

- 仮想マシングループにリソースプールを紐づける
  - リソースプール＝仮想マシングループ
  - 複数マシンで構成したクラスタ内にリソースプールを配置することで自然と冗長化される
  - リソースプール単位で仮想マシン群を管理
    - 本番用リソースプール：最優先・リソース潤沢
    - 開発用リソースプール：後回し・リソース限定

設計手順詳細(1)

1. 移行対象サーバーのリストアップ
   - マシンスペックおよび使用率
   - 重要度および負荷率をABCランク分け
   - 移行不要サーバーは除外
2. 特記事項の確認
   - 冗長構成
   - スタンバイ構成 など
3. マシングループ毎に仕分け
   - システムを構成するWeb＋DB＋その他をグループ化

サーバーリストの例

機器リスト.xls

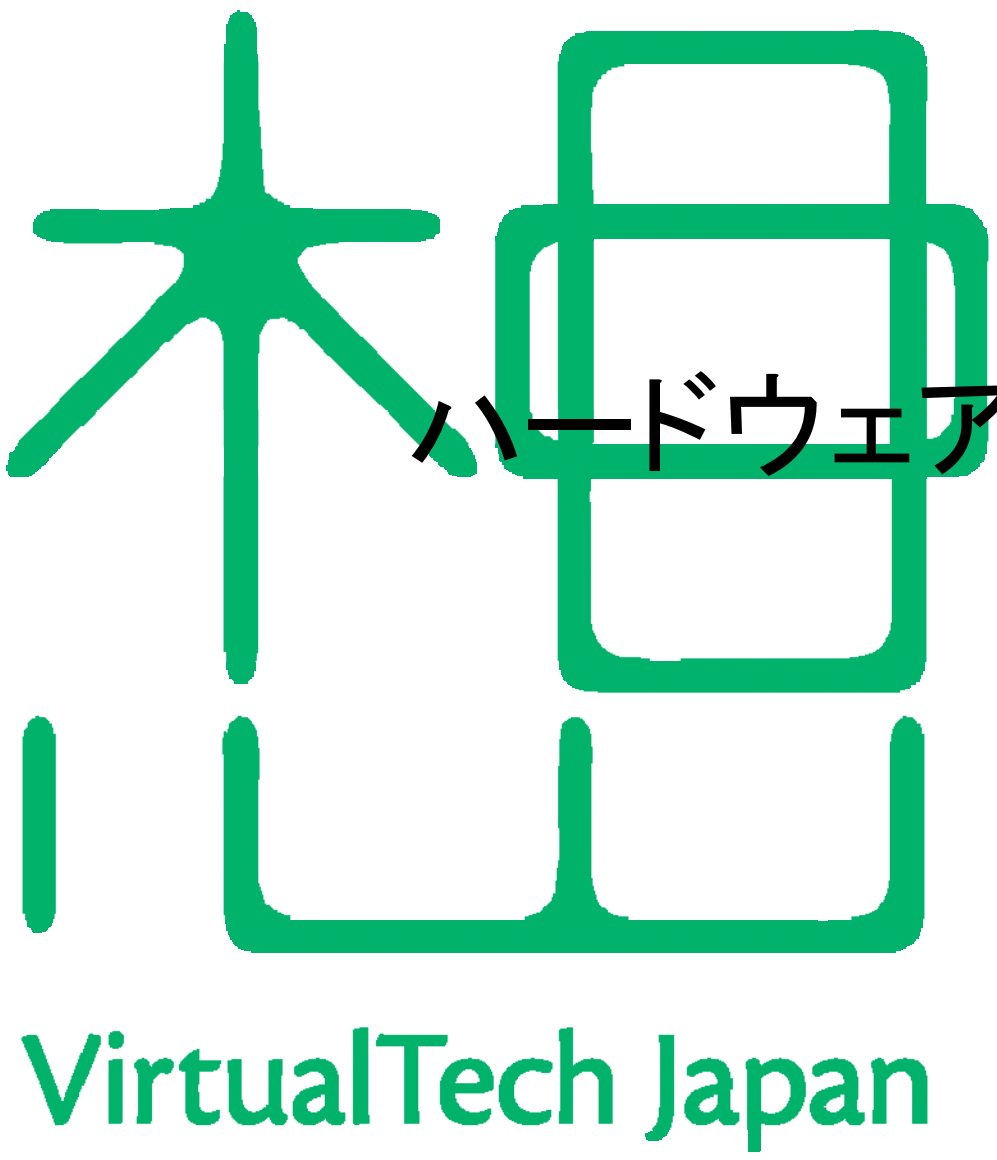
	A	機種名	CPU名	CPU数	コア数/CPU	SPEC CPU2000	CPU性能 (CPU数xコア数xSPEC CPU)	CPU 平均使用率(%)	CPU 最高使用率(%)	CPU性能xCPU平均	CPU性能xCPU最高
1											
2											
3	例	HP DL360G4	Intel Xeon 3.0GHz	2	1	1125	2250	20%	90%	450	2025
4	1	xxx	Xeon 3GHz	2	1	1440	2880	5%	10%	144	288
5	2	xxx	Xeon 3.6GHz	2	1	1715	3430	10%	20%	343	686
6	3	xxx	Xeon 2.8GHz	2	1	879	1758	5%	10%	87.9	175.8
7	4										
8	5										
9	6										
10	7										
11	8										
12	9										
13	10										
14	11										
15	12										
16	13										
17	14										
18	15										
19	16										
20	17										
21	18										
22	19										
23	20										
24							CPU性能			CPU性能(平均)	CPU性能(最大)
25							8068			574.9	1149.8

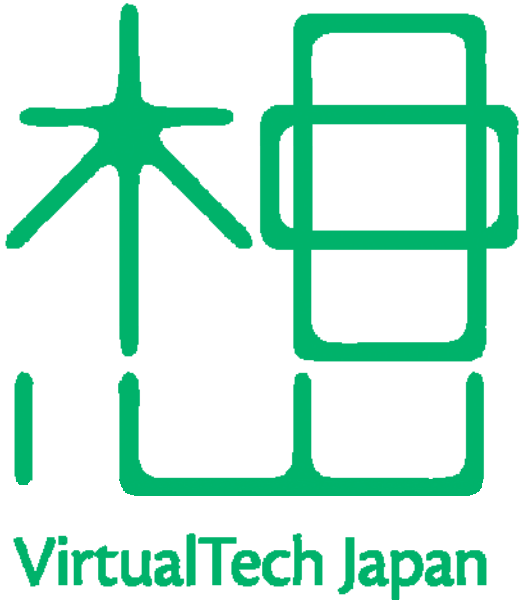
VirtualTech Japan

設計手順詳細(2)

4. グループ毎の要求リソース量を算出
   - CPUクロック数、メモリ量、ディスク容量、ディスクI/O、ネットワークI/Oなどの積算値
5. ターゲットH/Wによるリソースプールと比較
   - リソースプールは最低3台の仮想マシンホストで構成(冗長構成を考慮した場合)
   - CPUクロック数、メモリ量の積算値×60%程度
   - 不足の場合はリソース追加を検討
   - ストレージはパスの帯域幅やディスク本数などで性能値が左右されるので別途検討が必要

ハードウェアの選定





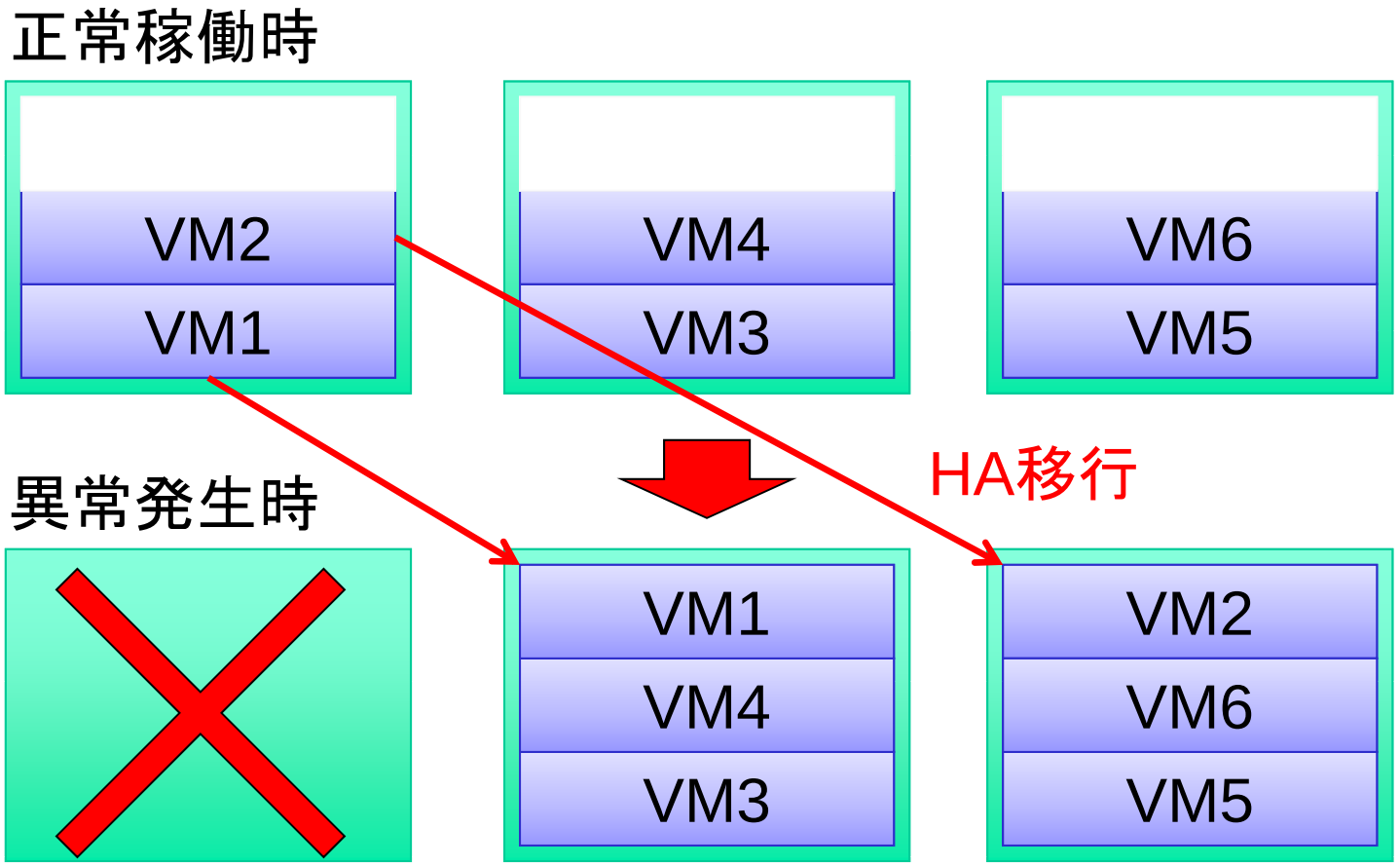
仮想化環境の性能要因

- ハードウェア
  - CPU:コア数、クロック速度、仮想化支援技術
  - メモリ:容量、速度、VMへの割当量
  - I/O:帯域、デバイス速度、仮想化支援技術
- ソフトウェア
  - VMM種別:ホストOS型、ハイパーバイザー型
  - 内部構造:ドライバモデル、スケジューラー
  - その他:OS、アプリケーション、データ量

ハードウェアの選定ポイント

- CPUは仮想化しても性能が下がりにくい
  - マルチコアは4コアが標準になりつつある
- メモリ量が仮想マシン収容力を大幅に左右
  - 32GB〜64GBが当たり前に
  - RVI、TLBなど仮想化支援技術も標準に
- 高速なI/Oの装備が必須
  - FCまたは10G Ethernet
  - Intel VT-d/AMD IOMMUのサポート

60%ルール



VirtualTech Japan

CPU使用率の計算方法

- CPUクロック数の世代間性能差に留意
  - CPUの性能指標であるクロック数は製品の世代によって性能が異なる
  - 実質的なクロック対性能は大きく変わっていないので、クロック数比で計算してもよい？

CPU	コア数	MHz	実質MHz	SPECint2000	MHz/SPECint2000
Xeon 3.0GHz(推定)	1	3000	3000	1,429	2.10
Xeon 3.4GHz	1	3400	3400	1,617	2.10
Xeon 3.0GHz	1	3600	3600	1,718	2.10
Xeon 5110(1.6GHz)	2	1600	3200	1712	1.87
Xeon5160(3GHz)	2	3000	6000	3,025	1.98

CPU使用率の計算例

- CPU使用率30%の物理マシンを、ほぼ同性能・同クロックの仮想マシンホストに移行
  - CPU使用率60%まで可能なので、2VMまで収容可能

収容可能VM目安＝CPUコア数×2

むしろメモリ容量の制約の方が大きい

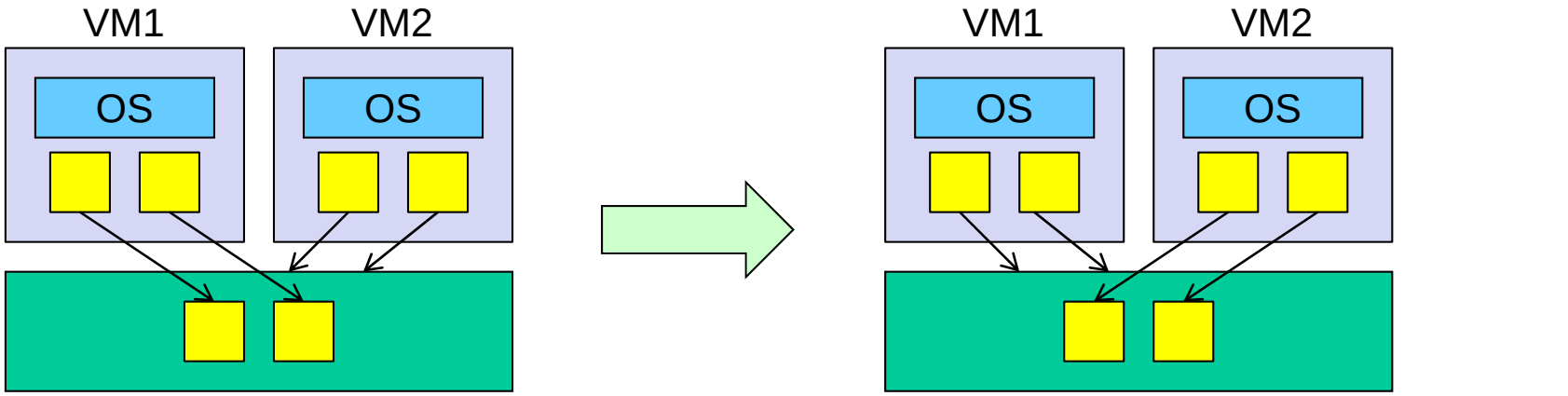
ブレード or ラックマウント？

- 仮想化環境を組むなら最低3台構成
  - RAID 5の考え方と同じ（最低HDD3台）
  - 60%ルールを基準にCPU、メモリなどを選定
- 4台以上ならブレード？
  - 以後の増設計画にも左右される
- ブレードの弱点も克服されつつある
  - HP BL495cではメモリスロット16本、最大128GBメモリ搭載可能、10Gb Ethernet標準

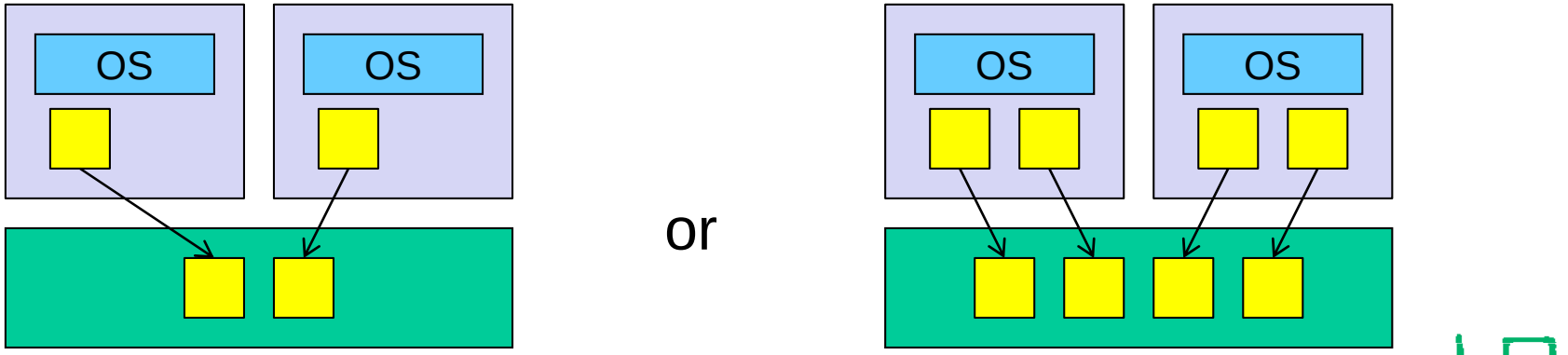
その他サーバ選定のポイント

- 高クロックよりもコア数
  - CPUロックが減少し、仮想マシンの実行並列度が高まる → 全体の性能が向上
  - 低消費電力型を選択できる → ランニングコストの削減
- スタンバイサーバは不要
  - 物理的な区切りで考えず、全体のリソース容量で考えること
  - 60%ルールの徹底

CPUの仮想化



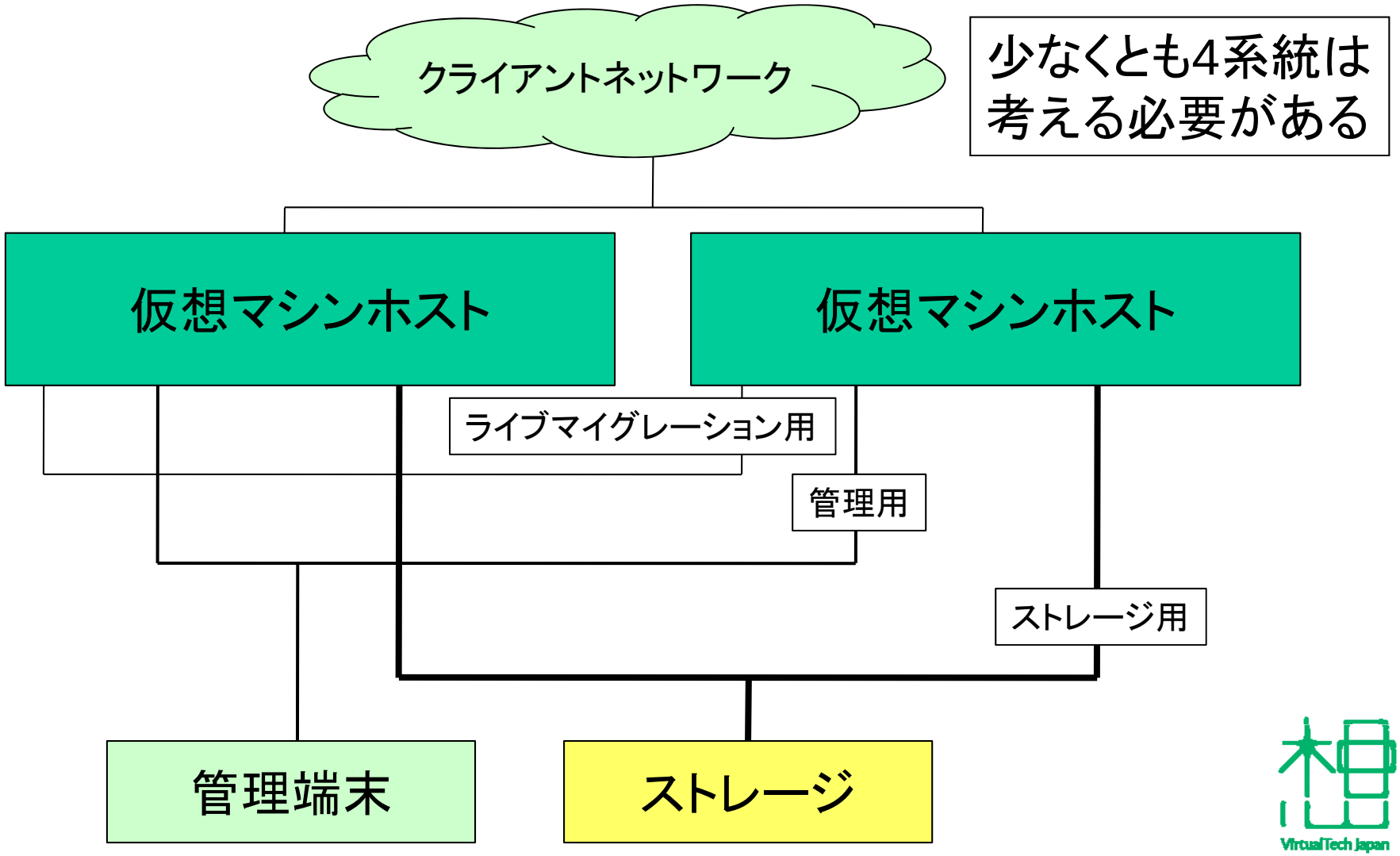
VM1がCPUリソースを専有　　VM切替でVM2がCPUリソース確保



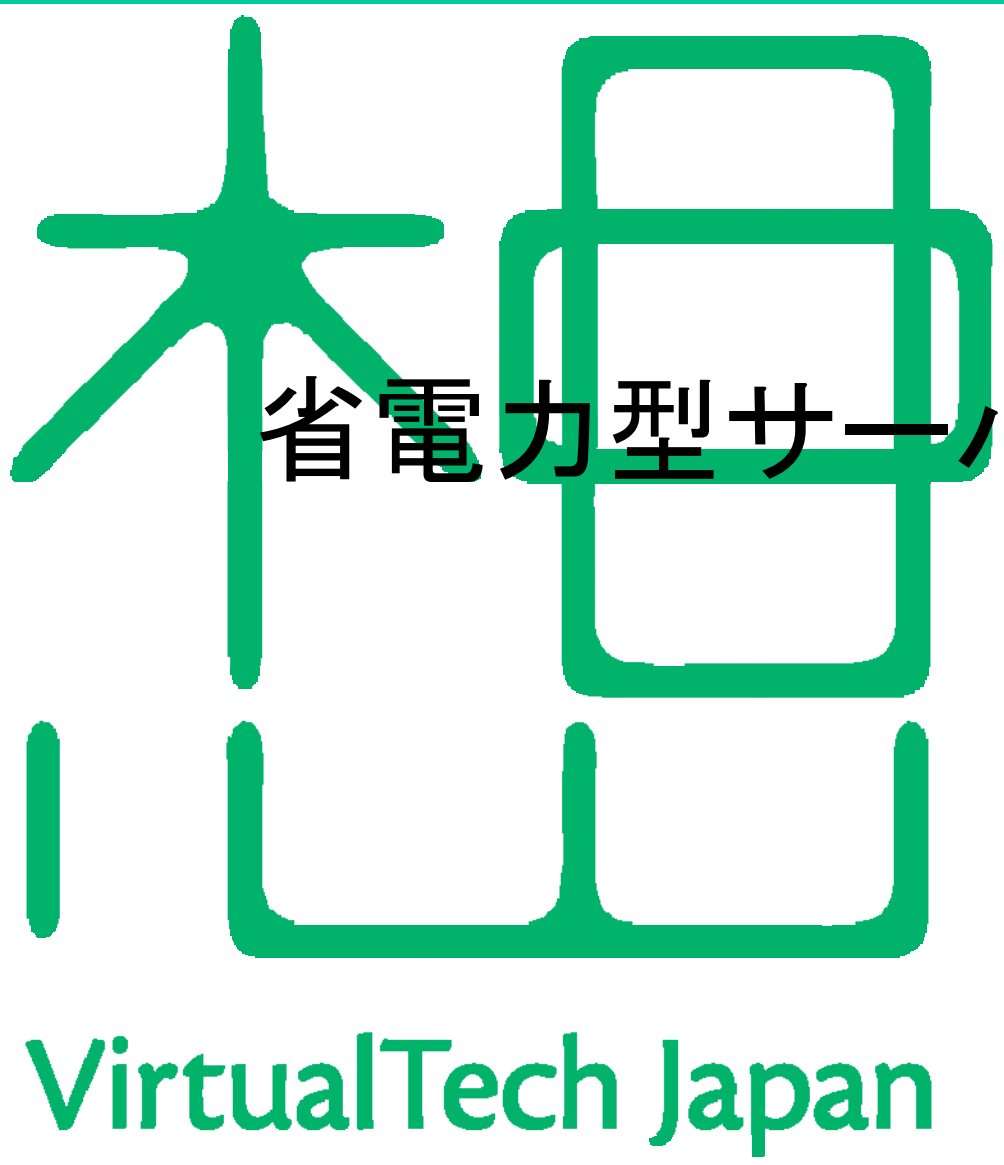
仮想CPU割当を減らす　　物理CPU数を増やす

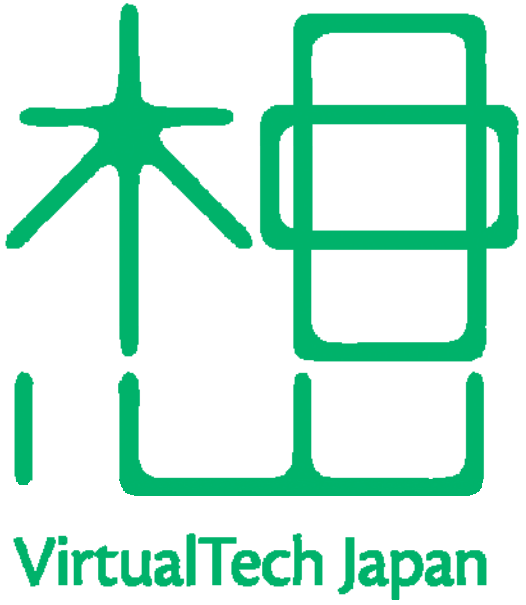
VirtualTech Japan

ネットワーク構成例


クライアントネットワーク
少なくとも4系統は
考える必要がある
仮想マシンホスト
仮想マシンホスト
ライブマイグレーション用
管理用
ストレージ用
管理端末
ストレージ

省電力型サーバーの検討





省電力型サーバー

- CPUが省電力型
  - クロック数はやや抑えめ
- メモリも省電力
  - FB-DIMMは消費電力が高かった
  - 今後は消費電力が低いDDR2/DDR3に移行
- ハードディスクを搭載しない
  - USBメモリやSANからのブート
  - 低消費電力なSSDの利用
- 発熱が少ないので、冷却も楽になる

参考

消費電力計算結果比較

	BL460c	BL460c G5	BL460c G6
CPU	L5410 2.33GHz	L5430 2.66GHz	L5520 2.26GHz
メモリ	32GB	32GB	**48GB**/32GB
アイドル時消費電力(W)	3540	2992	2911/2464
比率	100%	85%(-15%)	**82%**(-18%)/70%(-30%)
100%稼働時消費電力(W)	5262	4694	4441/3998
比率	100%	89%(-11%)	**84%**(-16%)/76%(-24%)
消費電力/クロック	2.3	1.8	2.0/1.8

Intel Xeon 5500番台で消費電力を16%～18%削減しつつ、メモリを1.5倍搭載可能

VirtualTech Japan

参考

実機計測

- BL460c：E5405(2.0GHz) 2P8C
  - 低消費電力型ではないローエンドモデル
- BL460c G6：L5520(2.26GHz) 2P8C
  - メモリ12GB/SAS 36GB 15krpm SFFx2/FC

	BL460c	BL460c G6	比率
最大消費電力	224W	135W	60.3%
最小消費電力	162W	79W	48.8%

低消費電力型への変更が非常に有効

VirtualTech Japan

参考

既存環境移行を計算

- 既存環境を以下のように仮定
  - Xeon 3GHz x 2コア 利用率20%程度
    - 3000MHz x 2 x 0.2=1200MHz
  - メモリ2GB搭載

	5500番台	既存環境	新規/既存
CPU	18080MHz	1200MHz	16.1台
メモリ48GB	49152MB	2048MB	24.0台
メモリ24GB	24576MB	2048MB	12.0台

10VM～20VM程度搭載可能と考えられる

参考

IntelとAMDを比較する

- 以下の仕様でWord Pressの動作性能比較
  - Intel Xeon X5570(2.93GHz x 4コア) x 2プロセッサ(合計8コア)
  - AMD Opteron 2435(2.6GHz x 6コア) x 2プロセッサ(合計12コア)
  - 24GBメモリ＋FCストレージ
  - VM：1CPU 1GB・CentOS 5.3
- 消費電力も比較

参考

性能比較

- 両CPUがほぼ同性能（差が4%程度）
- CPUコア数＝仮想CPU合計数が最も良い結果となっている
- Opteronの8VMは見かけは少ないが、全体のCPU使用率は68%程度で余力あり（4コア余り）

	Opteron 2435 (2.6GHz)		Xeon X5570 (2.93GHz HT On)		Xeon X5570 (2.93GHz HT Off)	
8VM	1553.1	68.6%	2175.3	96.0%	2180.3*	96.2%
12VM	2265.4*	100%	2207.5	97.4%	2171.8	95.9%
16VM	2249.1	99.3%	2248.7*	99.3%	2166.8	95.6%

参考

消費電力比較

- ベンチマーク時の最大消費電力で比較
- OpteronはXeonに対して16%から18%消費電力が低い
- Hyper Threadingが性能向上に対する消費電力の効率があまり良くない(3.8%:9.4%)

	Opteron 2435 (2.6GHz)		Xeon X5570 (2.93GHz HT On)		Xeon X5570 (2.93GHz HT Off)	
8VM	200W	87.3%	259W	113.1%	253W*	110.5%
12VM	229W*	100%	272W	118.8%	258W	112.7%
16VM	229W	100%	279W*	121.8%	255W	111.4%

提案例

消費電力削減効果

- 消費電力を3分の1に削減できます
- 仮想化移行を行うことで年間約127万円、5年間で約635万円の電気代削減効果が見込まれます
  - 1kWh=24.13円で計算しています

	消費電力	年間電気代
既存環境	9000W	¥1,902,409
移行環境	3000W	¥634,136
削減効果	33%に削減	¥1,268,273/年を削減

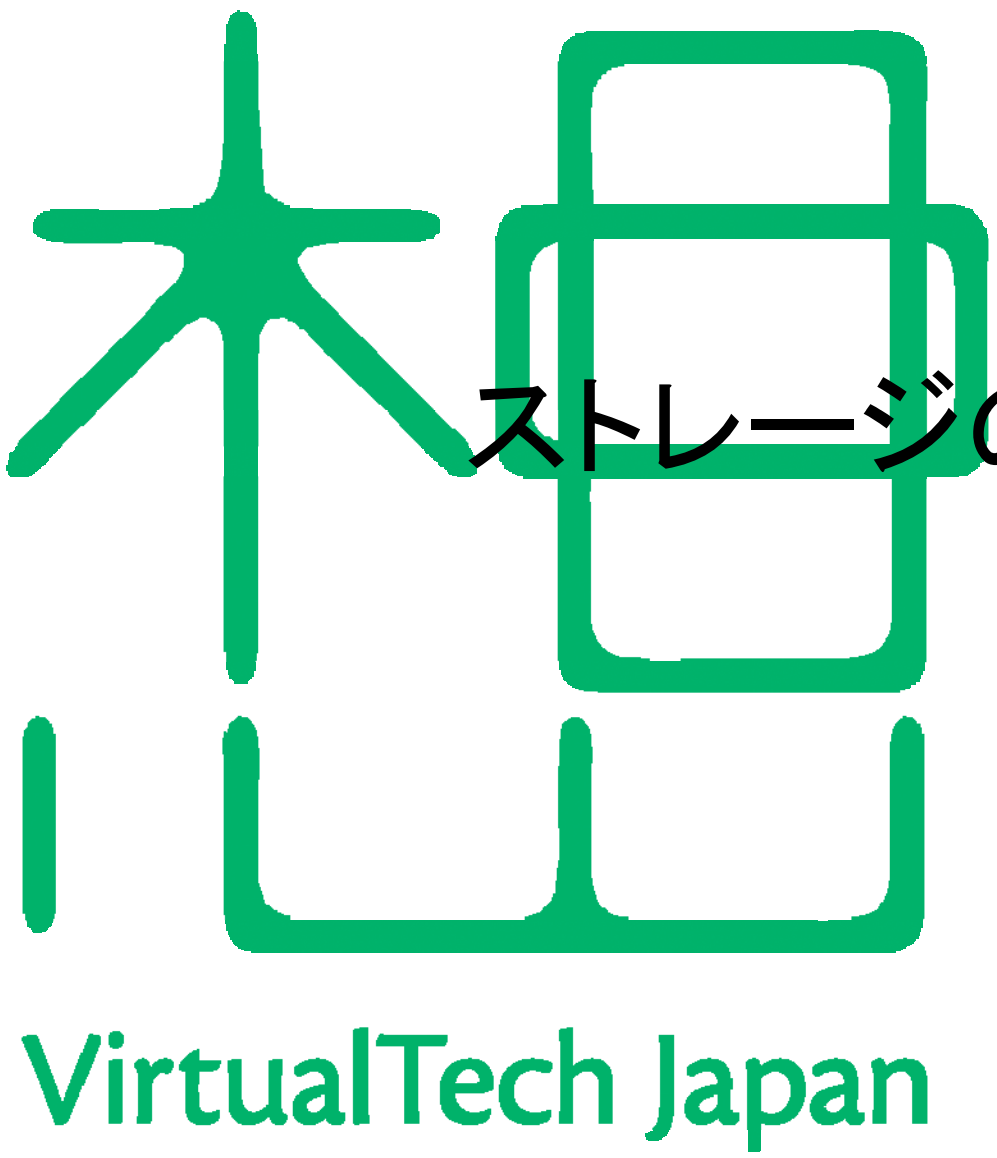
※消費電力は概算値です

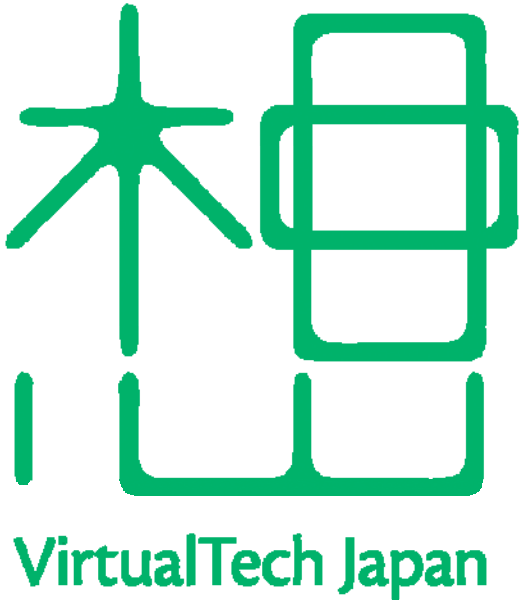
VircualTech Japan

仮想化のためのハード選び

- とりあえず低消費電力型
- クロック数よりもコア数
- メモリ多め
- 消費電力を事前に計算しておくこと
- 消費電力はサーバー増強時のことも考慮しておくこと

ストレージの選定



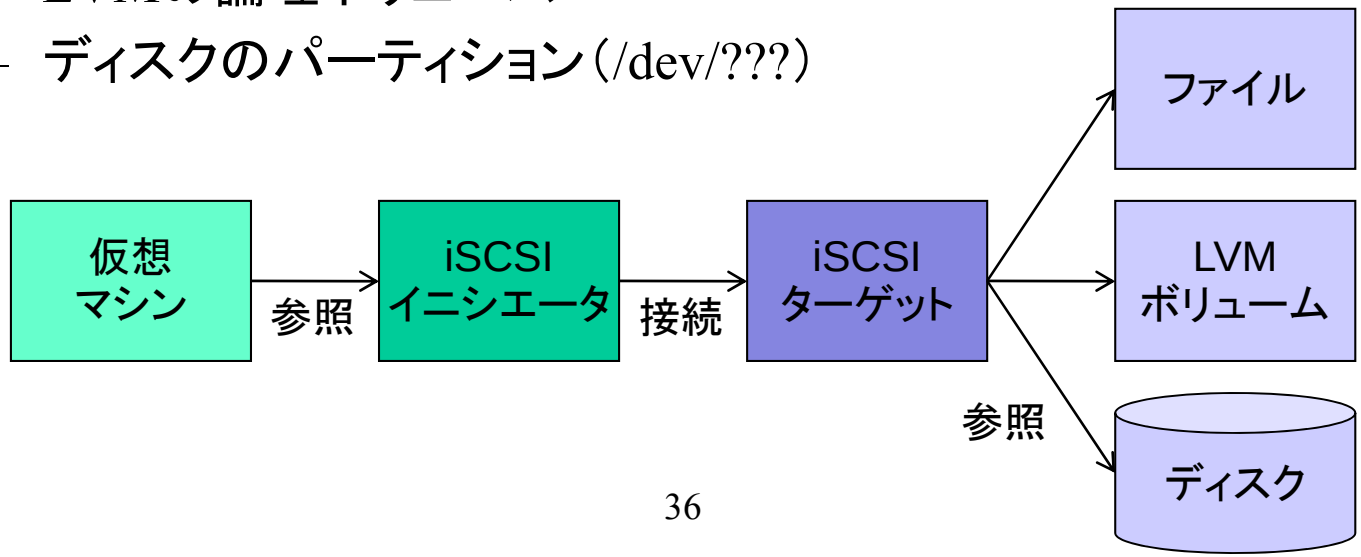


ストレージの選定

- 高速なストレージの選定
  - どの程度の速度が必要か
  - I/Oの性質は？（I/Oブロックサイズの大小）
  - HDDの台数
- 接続方法は？
  - FC接続：高速だが気軽ではない
  - iSCSI接続：小規模向けに人気急上昇
  - NFS接続：扱いやすい、意外と高速

iSCSIとは

- SCSIコマンドをTCP/IPでやり取りする仕組み
- iSCSIイニシエーターからiSCSIターゲットに接続
  - SCSIディスク(/dev/sd?)として参照可能
- iSCSIターゲットが参照可能なもの
  - ファイル
  - LVMの論理ボリューム
  - ディスクのパーティション（/dev/???）



VirtualTech Japan

ストレージ比較表

	扱いやすさ	性能	コスト	規模
DAS	簡単	普通	安い	小規模
NAS	普通	やや速い	安い	小～大規模
FC SAN	難しい	速い	高い	中～大規模
iSCSI	やや難しい	やや速い	やや高い	中～大規模

大まかな比較（構成によって例外あり）

- 性能：FC SAN＞NAS≧iSCSI≧DAS
- 扱いやすさ：DAS＞NAS＞iSCSI＞FC SAN

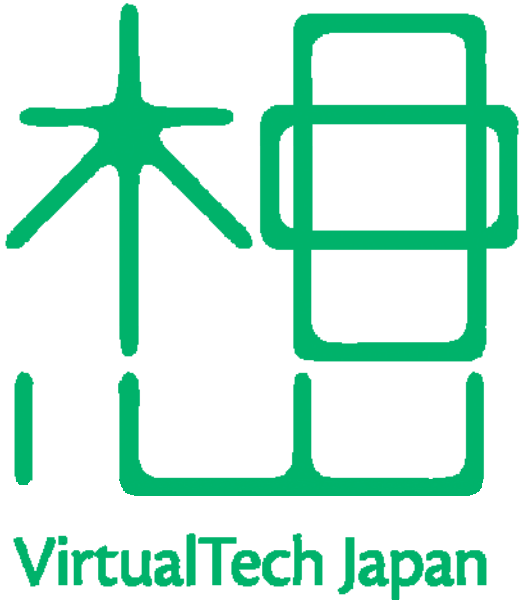
ストレージの追加機能

- 無停止での容量追加
  - ゲストOS側での対応も必要
  - ダイレクト接続＋動的ボリューム管理で対応できるが、仮想マシンの可搬性は低下
- スナップショット
  - 仮想マシンのバックアップに有効だが、費用対効果の見極めが肝心
- レプリケーション
  - DRサイト構築に有用

クラウド的ストレージは？

- クラウド＝分散と捉えると、そこまで必要とするシステムが企業系は少ない
- 低コストな自家製汎用ノードによる分散ストレージよりも、ベンダー保守を選択
  - 今後はアリかもしれない？
- key-value型やMapReduceなどよりも、RDBMSなど標準的なDBを選択
  - 特化型の仕組みなので、別セグメントの話？

ベンチマークによる性能検証



ベンチマークとは

- 処理速度の測定作業
  - あるハードウェア、ソフトウェアの処理速度を知りたい、比較したい
  - ある処理に必要となるハードウェア、ソフトウェアを知りたい
- 測定対象
  - CPU、ストレージ、ネットワーク、メモリ
  - ソフトウェアの処理速度

仮想化で使えるベンチマーク

- VMmark
  - VMwareの開発したベンチマーク
  - DBやWebなど、複数のVMを1セット（タイル）にして、ハードウェアのキャパシティを測定
- SPEC*
  - 業界標準のベンチマークを多数提供
- Iometer
  - ディスクI/Oを測定するベンチマーク
- ab（Apache Bench）
  - Apacheの性能測定用
- 他にも多数

仮想化ベンチマークの注意点

- ベンチマーク負荷作業は外部から
  - 仮想マシン内では時間測定が不正確
  - 内部で実行する場合でも、時間は外部から実行時に取得する
- キャッシュバッファの影響を考慮
  - レイヤーが1段増えることで構造が変わり、キャッシュが増える場合もある＝見かけ上の性能が物理サーバーを超える場合がある
  - 扱うデータ量や、VM数が増えればキャッシュが効かなくなり、性能低下を起こす場合もある

参考:Xen＋NFSを使用したDBベンチマーク結果

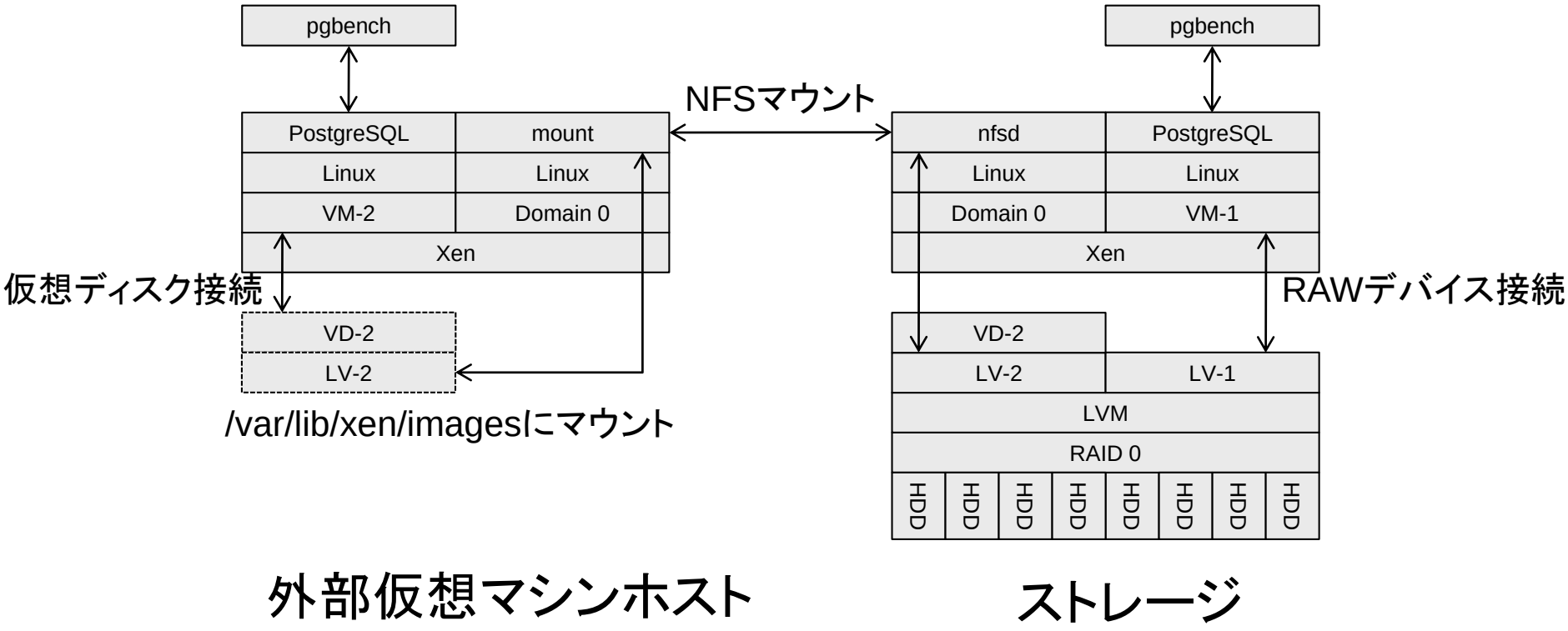
検証環境

- ML350 G5 x 2台
  - Xeon 5150(2.66GHz/Dual Core) x 1プロセッサ
  - 16GBメモリ
  - RAID 0(SAS HDDx8/128MBキャッシュ)
  - ギガビットイーサネット
- 仮想マシン
  - 1仮想CPU
  - 2GBメモリ
- PostgreSQL 8.3.4
  - pgbench(TPC-B相当のDBベンチマーク)

ストレージ環境

- NFSサーバーにてRAID 0ディスクを構築
- LinuxのLVMにて、LVを2つ作成
  - LVは10GBサイズ
  - LV-1はVM-1にRAWデバイス接続
  - LV-2はext3で初期化後、NFSでエクスポート
    - オプション：rw,sync,no_root_squash,no_wdelay
  - VM-2でLV-2領域をマウント
    - オプション：無し、またはrsize=8192,wsize=8192
    - 仮想ディスクファイル（8GB）を作成

ベンチマーク環境模式図



VirtualTech Japan

ベンチマーク作業

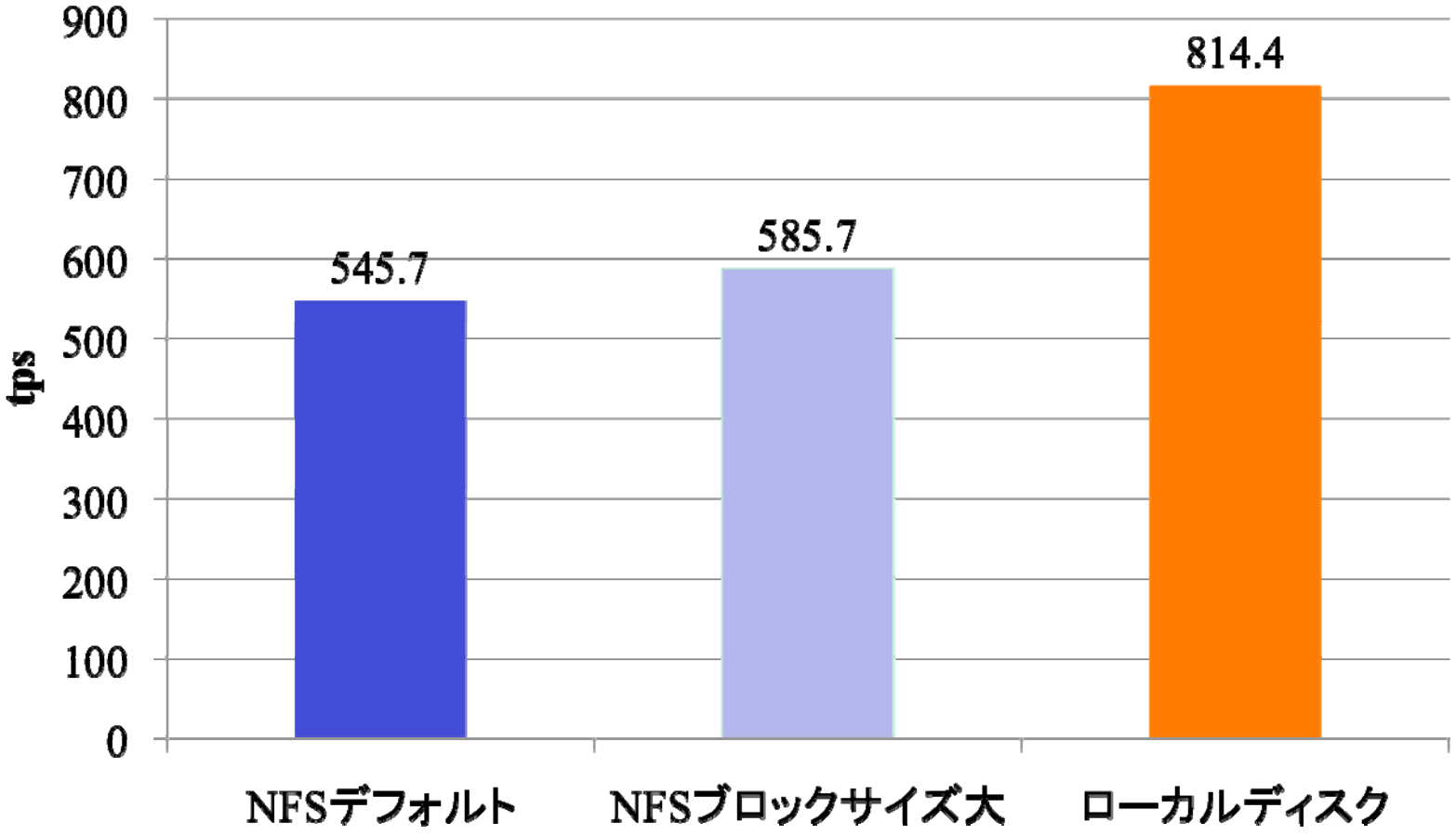
- pgbenchを使用
  - TPC-B相当のトランザクション
    - SELECT,UPDATE,INSERT
  - 検索（SELECT）のみも実行可能
- 今回の実行条件
  - スケール：20（2,000,000行/約300MB）
  - クライアント数：20(スケールと合わせる)
  - トランザクション数：1000
  - 最初の5回の結果は破棄し、6回〜10回のうちの上下1件ずつを取り除いた3回の平均値を取得

pgbenchのトランザクションSQL

以下のSQLを1つのトランザクションとして実行

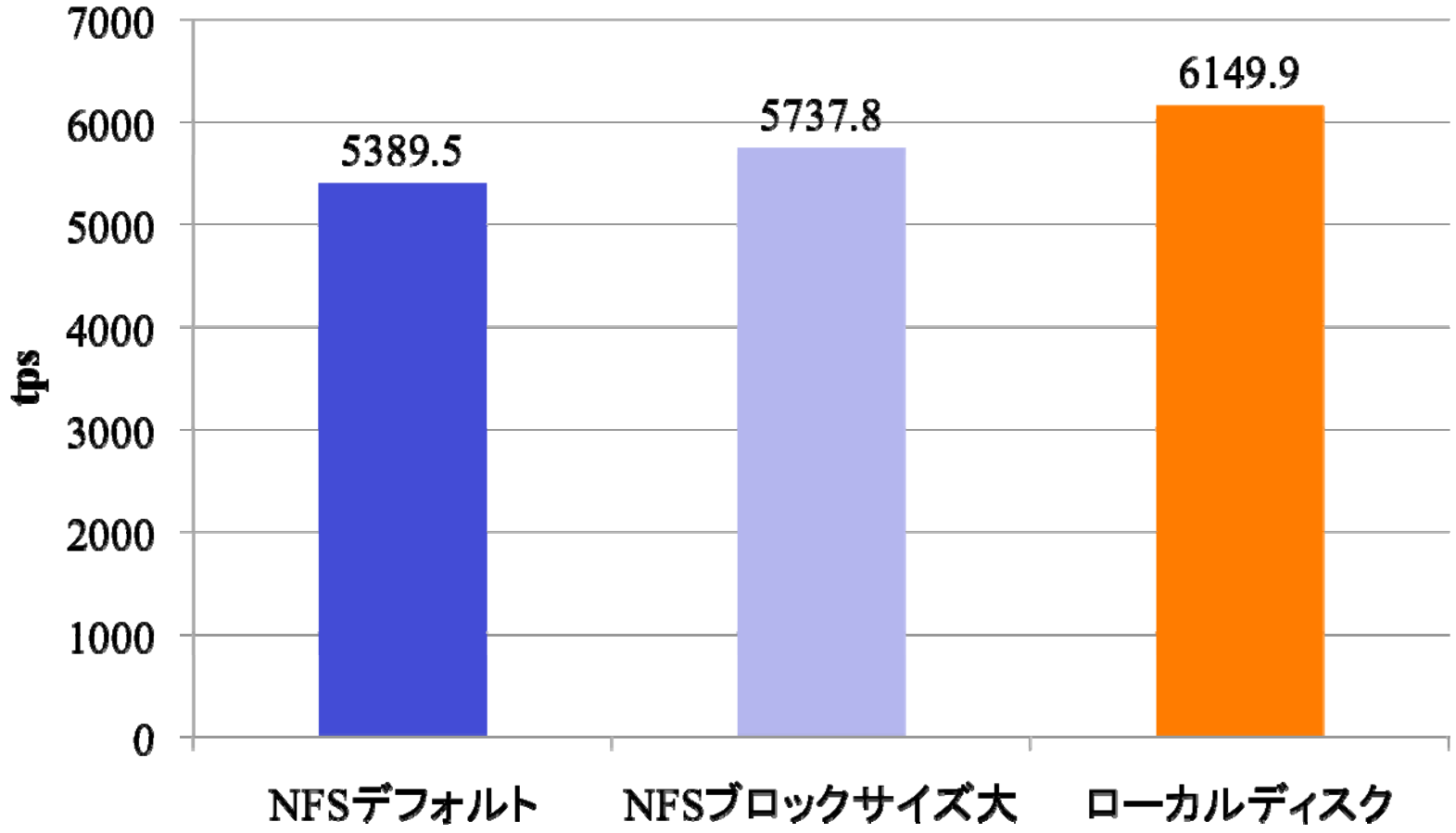
1. BEGIN;
2. UPDATE accounts SET abalance = abalance + :delta WHERE aid = :aid;
3. SELECT abalance FROM accounts WHERE aid = :aid;
4. UPDATE tellers SET tbalance = tbalance + :delta WHERE tid = :tid;
5. UPDATE branches SET bbalance = bbalance + :delta WHERE bid = :bid;
6. INSERT INTO history (tid, bid, aid, delta, mtime) VALUES (:tid, :bid, :aid, :delta, CURRENT_TIMESTAMP);
7. END;

検索更新処理結果



VirtualTech Japan

検索のみ処理結果



結果の考察

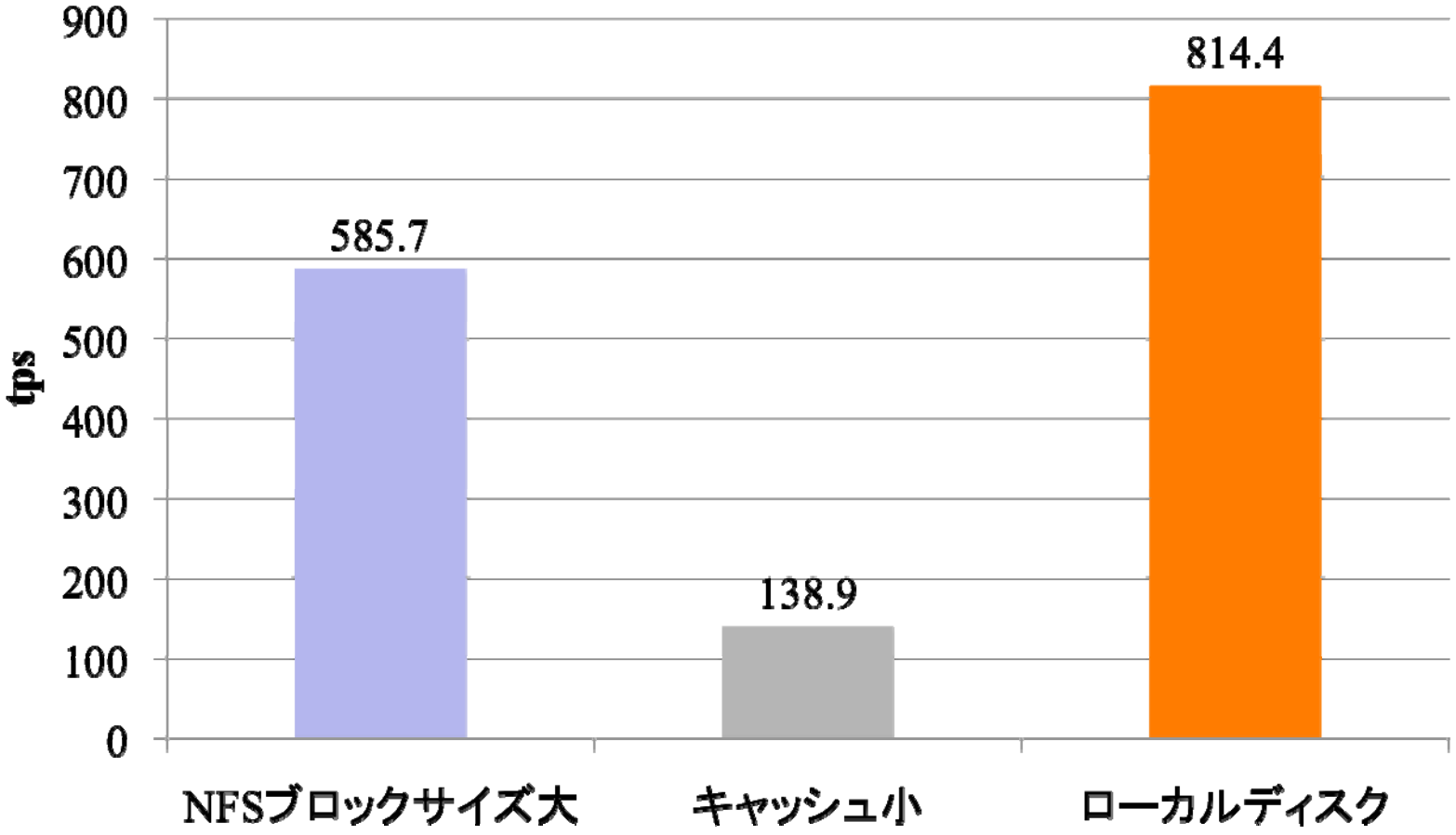
- NFSを使用した場合、検索更新処理ではローカルディスクの65%程度の性能
- NFSブロックサイズを8KBにすると、ローカルの70%まで性能向上（5%アップ）
- 検索のみ処理の場合、顕著な性能差が認められないため、UPDATEおよびINSERT処理のオーバーヘッドによる性能劣化と考えられる

NFSサーバーキャッシュの影響

- NFSサーバーの利用するキャッシュの影響を測定
- NFSサーバーの搭載するメモリを16GBから512MBに変更し、同一内容を測定
  - システム起動時でキャッシュバッファが130MB程度
  - VM-2の仮想ディスクサイズは8GB、DBサイズが300MBのため、キャッシュにすべてが乗り切らなくなる

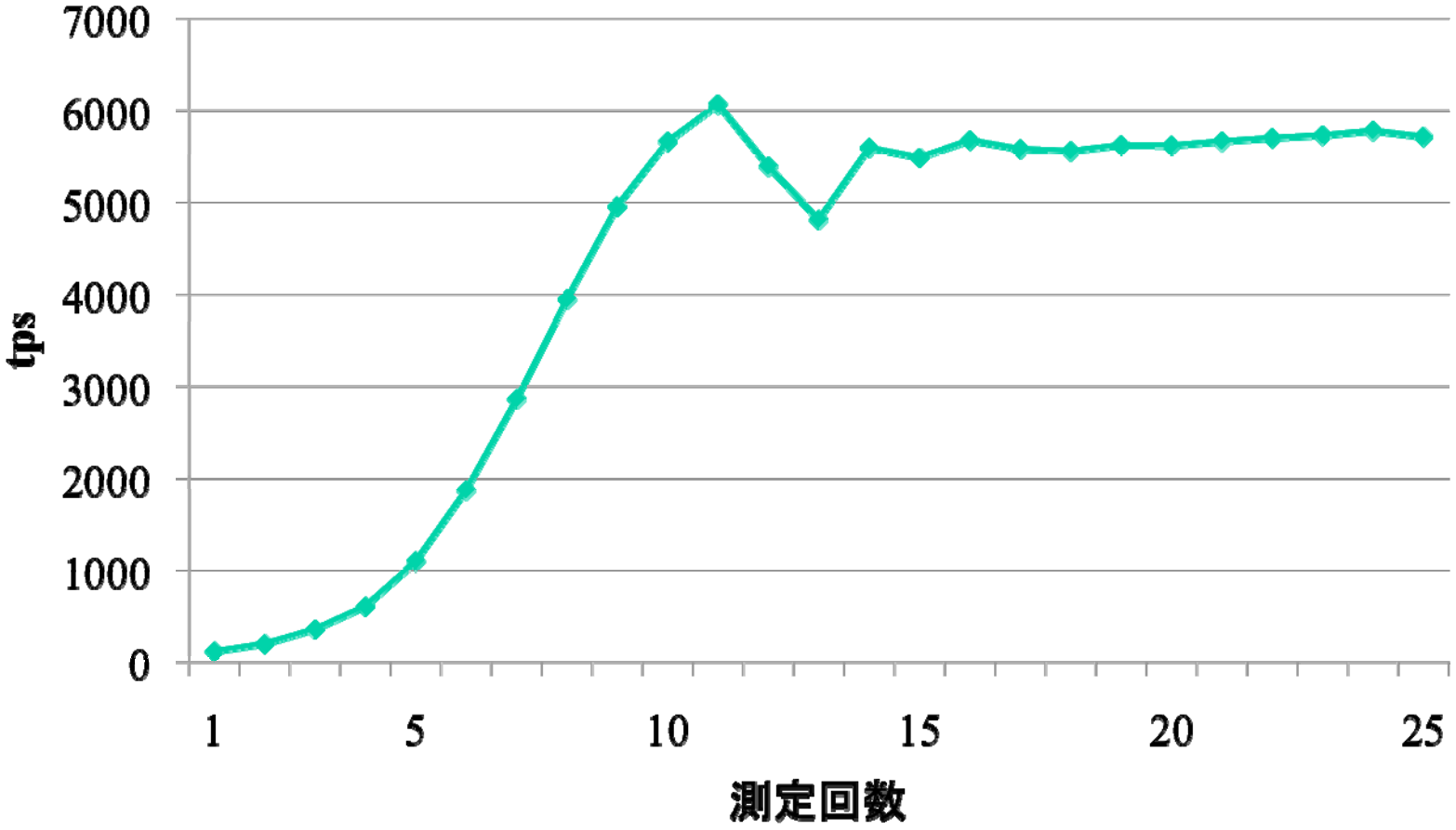
- NFSサーバーキャッシュが不足している場合、検索更新処理ではローカルディスクの17%まで性能劣化
- 検索処理はキャッシュ容量が少なくてもあまり影響を受けないが、性能ピークへの到達に時間がかかる
  - PostgreSQLのローカルバッファやインデックスが影響？

検索更新でのキャッシュの影響



VirtualTech Japan

検索のみ処理の結果推移



考察

- 検索主体のDBでは、NFSをストレージとして使用しても性能劣化は少ない
  - NFSストレージのキャッシュが少ない場合でも、ローカルバッファやインデックスの効果が認められる
- 検索更新のDBでは、NFSをストレージとして使用すると性能劣化が認められる
  - 今回の条件では30%程度の劣化
  - NFSサーバーのキャッシュ容量が不足すると性能劣化の度合いが激しくなる
  - 仮想マシン数が増えると、NFSサーバーのキャッシュ容量が不足する可能性があるので注意

参考

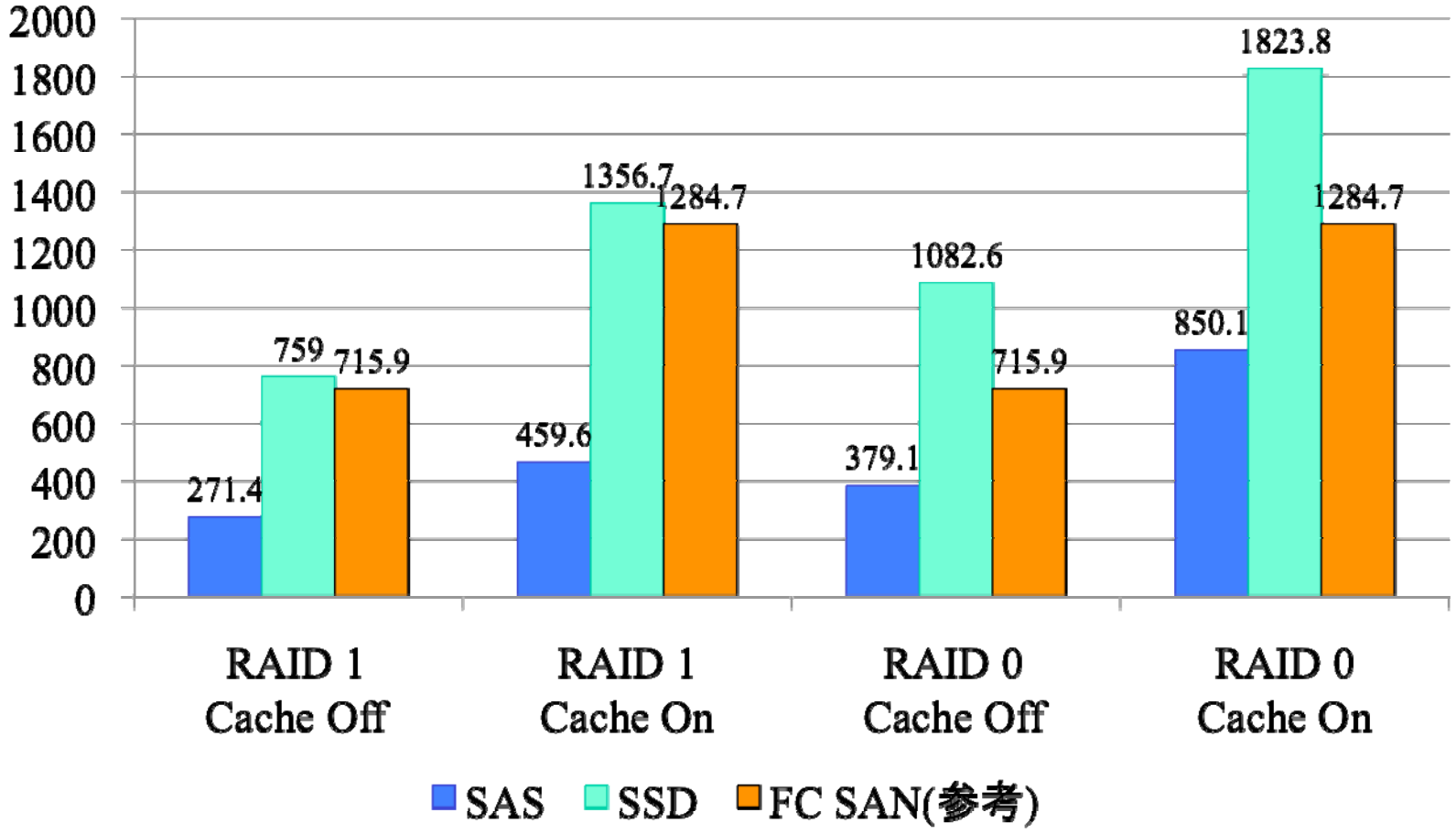
SSDは速いか？

A. SSDは良さそうだ
- 高速なランダムアクセス
- 低消費電力
- 低発熱

B. SSD導入は時期尚早？
- まだ実績が少ない
- まだまだドライブ単価が高い
- 書き換え回数上限への懸念

参考

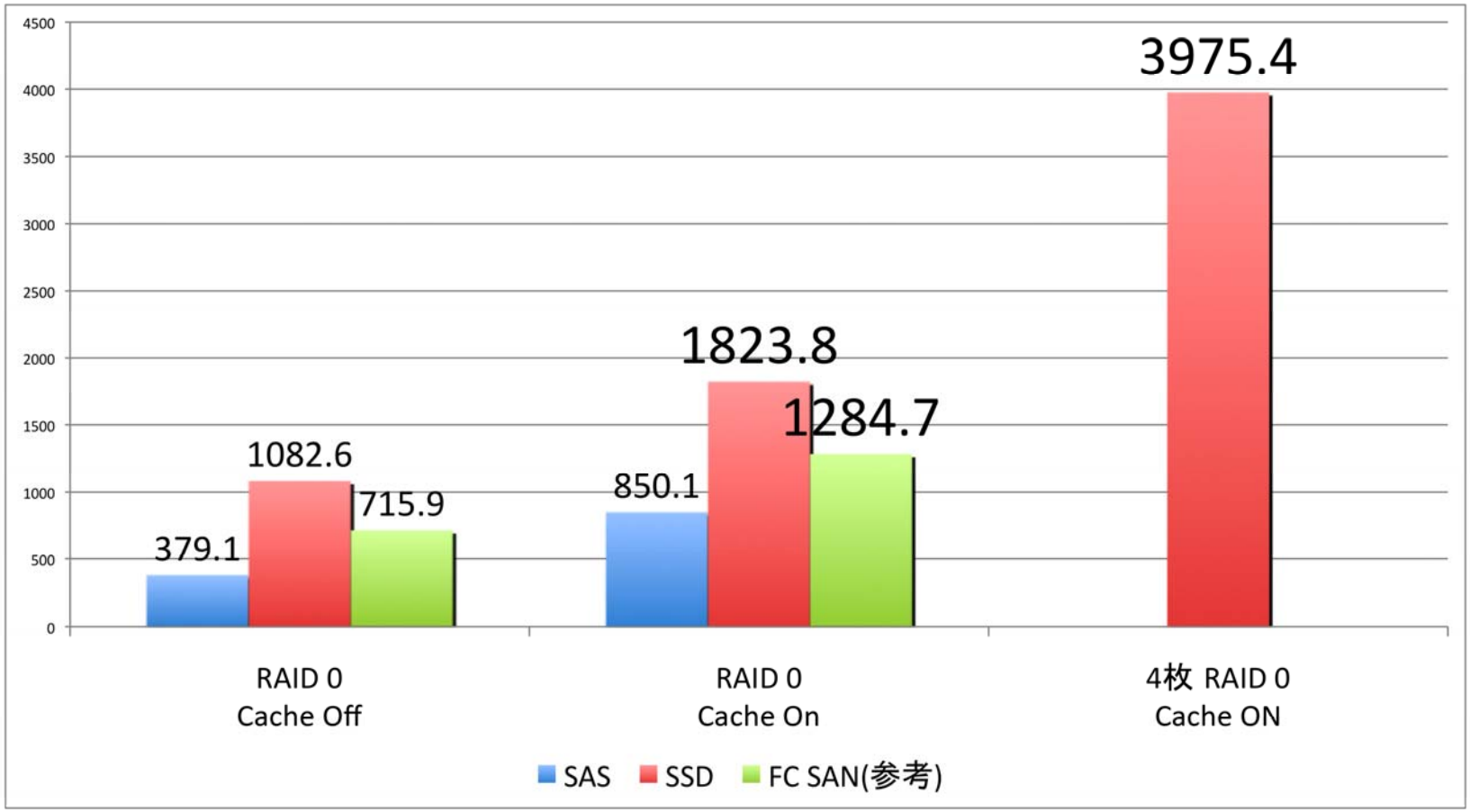
TPC-B ベンチマーク結果



VircualTech Japan

参考

TPC-B ベンチマーク結果



VirtualTech Japan

参考

ベンチマーク実行環境

- SASディスク：2.5” 36.4GB 15krpm × 2台
- SSD：Intel X25-E(SLC) 32GB × 2台
  - RAIDコントローラーのキャッシュはOff
  - ディスク内蔵のキャッシュをOn/Offし測定
- FC SAN：HP MSA1000
  - SCSI 146GB 10krpm × 14台によるRAID 5
  - コントローラーに512MBキャッシュ
    - Cache Off：R50%/W50%
    - Cache On：R0%/W100%
- PostgreSQL 8.3.7によるベンチマーク
  - pgbench -c 20 -t 3000
  - 20回実行し、11回～20回の平均値を記録

まとめ

- 分散が多いクラウド的な技術の中では、仮想化は集中的なお話
- リソース競合が発生しない限り、性能設計は比較的容易
- ボトルネックがI/O部分に発生しやすい
- 事前ベンチマークで性能値を測定するのが重要

お問い合わせ先

「仮想化環境を構築したいが、どこに相談すればいいの？」

まずは我々にご相談ください

日本仮想化技術株式会社

http://VirtualTech.jp/
sales@VirtualTech.jp
050-7571-0584

VirtualTech Japan