

IBM Power Systemsにおける マルチテナント型仮想化ホスティング・サービスの設計技法

山口 星志 青山 真巳 沢橋 松王

Design Method for Multi-tenant Virtualized Hosting Service Using IBM Power Systems

Seiji Yamaguchi, Manami Aoyama and Matsuo Sawahashi

複数のお客様が仮想区画を共用するマルチテナント型仮想化ホスティング・サービスを提供するためには、個々のセキュリティーやパフォーマンスだけではなく、多様なお客様への要件を満たす柔軟性や可用性を考慮する必要がある。本論文では、シェアード・サービスのためのセキュリティー、論理区画（LPAR）のサイジング技法などのテクニックだけではなく、安全で安価な保守を意識した構成設計など運用面にも配慮した、IBM Power Systemsを使用したマルチテナント型仮想化ホスティング・サービスを実現するシステム設計技法を提案する。本設計技法を用いるとクラウド・コンピューティング・サービス事業者はAIXのOSイメージを提供するPaaS、もしくはIaaS型のサービスを展開することが可能となる。またアウトソーシング事業者も同様に複数のお客様向けにシェアード型のIT資源提供サービスを展開することが可能となる。

When delivering a multi-tenant virtualized hosting service providing virtual machines to multiple customers, we have to consider security and performance for each customer. In addition, we also have to consider flexibility and availability. This paper presents IBM Power Systems' design method, which not only provides security for shared service and sizing techniques, but also a design technique which takes into consideration operational aspects such as safety configuration and that reflects an awareness of the need to provide reliable maintenance at low cost. When applying this design method, cloud computing service providers can develop PaaS or IaaS services using AIX OS image. Outsourcing service providers can also deploy shared IT resources for multiple customers.

Key Words & Phrases : 仮想化, マルチテナント, システム設計, クラウド, サイジング
virtualization, multi-tenant, system design, cloud computing, sizing

1. はじめに

近年、仮想化技術の躍進により、クラウド・サービスに対するお客様の興味は年々高まっている。経済産業省が発表した報告書では2020年までに累計40兆円超のクラウド・サービス市場が世界で創出されるとの予測をしている[1]。国内でも現在300億円の売上有るクラウド・サービスは、5年後には1,400億円以上の市場規模になると予想されており、特に、PaaS (Platform as a Service) もしくはIaaS (Infrastructure as a Service) への注目が集まっている [2] [3]。

IBMではこの領域へのIBM Power Systems^{*1}を使ったクラウド・サービスが待たれていたが、2010年7月にSO (Strategic Outsourcing: 戦略的アウトソーシング) 向けのIaaSとしてMCCS (Managed Cloud Computing Services) にPower Systemsを利用するサービスメニュー

を追加した [4]。本論文では、このMCCS設計の経験を元にマルチテナント型仮想化ホスティング・サービスの設計手法を提案する。実環境を用いたテスト結果を提示し、その有効性を示す。

2. マルチテナント型仮想化ホスティング・サービスの課題

マルチテナント型仮想化ホスティング・サービスの共用環境の設計では、個別のお客様向けの設計とは異なり、それぞれのお客様が利用するシステムの論理区画間^{*2} (LPAR: Logical PARTitioning) のセキュリティーやパフォーマンスを考慮するだけではなく、多様なお客様への要件を満たす柔軟性や可用性を考慮する必要がある。以下にそれぞれの課題について述べる。

提出日:2010年9月6日 再提出日:2011年6月14日

*1 POWERプロセッサベースのUNIX機器
*2 物理筐体を論理的な区画に分割する技術

2.1 共用環境における可用性

共用環境において、サービスを提供しているすべてのお客様へ同一の停止時間を調整するという事は現実的ではない。特に予防保守などの際にサービス停止が発生することのない可用性が必要である。また筐体障害時の影響を最小化する必要がある。

2.2 LPAR間の分離性

シェアード・サービスにおける仮想化設計では、セキュリティ上の観点から、完全に LPAR 間の分離性を保つことが必須である。ネットワークとしては LPAR 間での通信を完全に遮断し、ディスク・アクセスについても、各お客様環境を分離できる構成とすることが必要である。

2.3 リソース競合の問題

限りある CPU リソースを有効に、かつ他の区画の影響を受けないようにするためには工夫が必要である。キャパシティを保証するだけでなく、ピークや低稼働率の場合に他の区画とのリソースの融通を図る必要がある。

限られたネットワーク帯域においても最大限のパフォーマンスを出せるだけでなく、他の区画への影響が少ない構成を取る必要がある。

2.4 多様なお客様要件を満たす柔軟性

共用インフラ・サービスとして提供するためには、多様なお客様環境の要件を満たす必要がある。システム設計時に想定したお客様環境だけではなく、さまざまなピーク特性を持ちエンド・ユーザーへのサービス内容もまったく異なるお客様環境に対しても、サービス提供ができる必要がある。

エンタープライズ向けのクラウド・コンピューティング基盤の設計手法は数多くあるが [5]、それらの手法は主にセキュリティやシステムの可用性に注目し、限られたリソースの有効活用や定期保守のための計画停止など、安全かつ安価なサービスを提供するためのシステム設計についての考慮は少ない。本手法では、こうしたマルチテナント型仮想化ホスティング・サービスにおける数多くの考慮点を、有効的に解決する設計手法を提案する。

3. マルチテナント型仮想化ホスティングサービスの設計手法

この章では MCCS で選択した手法を元に、マルチテナント型仮想化ホスティング・サービスとしての推奨設計について述べる。

3.1 全体構成

共用環境の障害や保守時の影響を最小化するために、クラウド・コンピューティング・システムのコンポーネントは、徹底した二重化が求められる。MCCS ではサーバーやネットワークだけではなく、ストレージ筐体も二重化した。通常はストレージのコントローラーが二重化されているために問題ないとされるが、例えば、コントローラーが二重化されていてもストレージ筐体全体の障害が発生したら、すべてのお客様に致命的な影響を与えてしまいかねない。またサービス停止を伴わない保守が実現できる設計が必要である。これらを解決するために、SVC (SAN Volume Controller)^{*3} を用いた仮想ディスク環境下におけるストレージ筐体の二重化を推奨する。

3.2 ディスク構成

各 LPAR の分離性を保ちつつ、パフォーマンスを考慮したストレージ設計が必要となる。

SVC を利用した仮想ディスク環境によって LPAR 間の分離性を保証する方法として、お客様ごとに Mdisk Group をアサインする方法と、Vdisk をアサインする方法がある。どちらの方法でも IBM のセキュリティ・ポリシーである ITCS104 の要件を満たしており、お客様ごとの分離性は保たれる。

Mdisk Group は個々のディスク筐体ごとの論理ディスクである LUN (Logical Unit Number) の集合体であり、Vdisk は Mdisk Group から均一にエクステント (論理的に連続した記憶域) を選択した仮想ディスクである [6]。

Mdisk Group を構成する場合、できるだけ大きいサイズでかつ同じサイズの LUN を組み合わせると、アクセス効率が高くパフォーマンスが高くなるが、多様なお客様環境に対応する必要があるため、設計時に LUN の数やサイズを見積もることはできない。そのため個々のお客様ごとに Mdisk Group を割り当て、アクセスを分離する方法は、LUN を効率的に使うことができずパフォーマンスがあまり見込めない構成となる。

パフォーマンスがよく、さらには個々のお客様を受け入れる際の作業も少なく済むため、お客様ごとに Vdisk をアサインしアクセスを分離する方法をとり、図1に示す構成とすることを、この種のサービスにおける設計指針として広く推奨する。

また、Vdisk を切り出す元となる Mdisk Group の構成方法としては、筐体の能力および回転速度とサイズの同じ物理ディスク上に構成した LUN を組み合わせることに

*3 ストレージ機器の仮想化を実現する機器

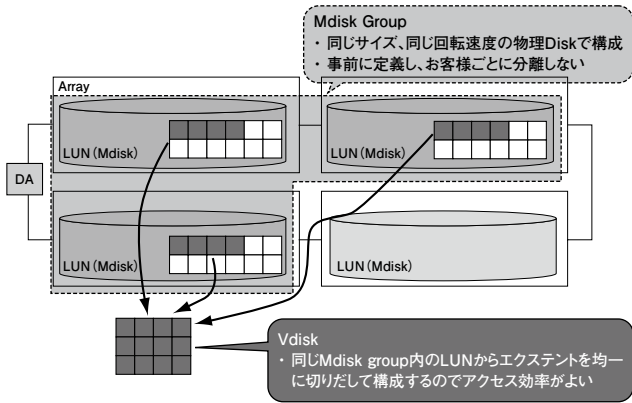


図1. Mdisk GroupとVdisk設計の考慮点

より、パフォーマンスのよい構成をとることができる。

シェアード環境ではさまざまな種類のディスク・アクセスが混在し事前に想定することはできない。高価で高速なディスクを用意できればよいが安価なサービスを提供するためには最適化が必要である。そこで推奨するのは、SVCを用いた仮想ディスク環境下で高速ディスクと低速ディスクを組み合わせ、用途に応じて使い分ける設計である。

3.3 サーバー構成

(1) VIOSのCPU設計

Power Systems の仮想環境を提供する VIOS (Virtual I/O Server)^{*4} は、ネットワークとストレージ I/O を集中管理するため、二重化は必須である。

シェアード・サービスとして提供する場合、さまざまな負荷ピークを持った複数のお客様が利用する可能性がある。そのため、初期設計での負荷見積りは困難であり、片方の VIOS に負荷が偏る可能性がある。2 区画の VIOS で利用するための専用 SPP (Shared Processor Pool)^{*5} を構成し、Shared^{*6}、Uncapped^{*7} でアサインすることにより、急な VIOS 間での負荷の偏りにも対応できる設計を推奨する。

(2) VIOCのCPU割り当て

VIOC (Virtual I/O Client)^{*8} の CPU 割り当てについては、想定されるお客様の CPU 使用状況のデータを取得し、VIOC 全体の CPU 使用量のピーク時の必要能力値を算出する。この方法により、Micro-Partitioning [7] の技術を利用して、各 VIOC が効率的に CPU を確保できるように計算できる。

CPU リソースの競合を起こさないようにするために、VIOC 上の SPP は同一プールかつ Shared の割り当てとし、重みづけもすべての VIOC で同等とする。また Desired パラメーター^{*9} で CPU キャパシティーを保証しな

がら、Uncapped 属性によってコアの上限までバーストを許すことができる。

VIOS で使用する SPP とは分けることにより、VIOS の負荷による影響を各 VIOC に与えないようにする。

(3) LPMによる可用性

共用環境におけるサーバー筐体保守時の可用性を確保するため LPM (Live Partition Mobility) 機能 [8] が有効である。2 筐体以上同時に停止することはない前提とすれば、1 筐体分の VIOC 資源に相当するリソースを筐体プールの中に確保しておけばよい。1 台に寄せておいてもよいし、全台で均等に確保してもよい。

(4) VLANによるVIOSの振り分け

VIOC のネットワークにおける冗長性を確保する方法として、SEA (Shared Ethernet Adapter) Failover と SEA NIB (Network Interface Backup) という方法がある。前者は2つの VIOS で同じ VLAN^{*10} に属する VE (Virtual Ethernet)^{*11} からなる SEA を構成し、通常時は優先度が高い方の VIOS を経由して筐体の外と通信し、通常時の経路の障害時には、もう一方の VIOS に Failover して通信をする方法である。後者は2つの VIOS で異なる VLAN に属する VE を VIOC に割り振り、NIB^{*12} を構成するという方法である [9]。

それぞれの構成におけるメリット・デメリットの概要を表 1 にまとめた。

表1. SEA Failover – SEA NIB構成比較

	利点	難点
SEA Failover構成	VIOCで意識しない切り替え・切り戻しが可能	確保できるネットワーク帯域が半分となる
SEA NIB構成	多くの帯域を使うことが可能	SEAへの振り分けは各VIOCでの設定が必要

冗長性・メンテナンス性を確保するために VIOS 2 区画から構成される SEA Failover 構成を推奨する。SEA Failover 構成は Active-Standby で冗長性を確保するた

^{*4} ディスク I/O や通信に必要な物理アダプターを複数の論理区画から共有して使用することを可能にする専用論理区画
^{*5} CPU 資源を確保するリソース・プールのこと。論理区画は当該リソースに所属させることにより効率的なリソース配置が可能
^{*6} CPU リソース割り当て方法の一つ。Shared は割り当てられた SPP にて、CPU を他の論理区画と共用する。
^{*7} 論理区画の CPU リソース設定属性。オンに設定することにより、リソースの空き状況によっては設定値以上の CPU 値を割り当てられる。
^{*8} VIOS からリソース提供を受ける仮想マシン
^{*9} 論理区画への CPU リソース設定方法。SPP に余裕がある場合は Desired に設定した値のリソースが割り当てられる。
^{*10} 仮想 LAN のこと。物理接続ではなく、仮想的な LAN を設定すること。
^{*11} ハイパーバイザー層に設定する仮想的な Ethernet アダプターのこと。
^{*12} AIX によるネットワーク・インターフェース・カードの冗長化機能

め、各 VIOC は優先度が高い VIOS のみを経由して通信することになる。

そのため構成している VIOS 2 区画のうち、優先度が高い VIOS に負荷が偏ってしまう可能性がある。そこで図 2 に示す通り VIOS での負荷の偏りを軽減するため、VLAN 番号の偶数・奇数でそれぞれ A 系、B 系の SEA を構成し、Network I/O による負荷の調整・分散ができる設計を、この種のサービスの設計方針として推奨する。

(5) Etherchannelによる帯域確保

ネットワークのスループットを出すために、各 VIOS には複数のイーサネット・ポートをアサインし、Etherchannel^{※13}を構成することを推奨する。バックアップ用チャンネルは構成せず、障害時には手動で SEA Failover を実行することにより、帯域の減少時間を最小限に抑えた運用が可能となる。

(6) 最大VLAN数の事前設定

VIOS の仕様として、付与した VLAN の追加・削除の際には VIOS の再起動を伴う SEA の再構成が必要である。よって、サービス停止を発生することなく柔軟な要件を満たすために、設定可能な最大数の VLAN を用意した方がよい。VE あたり最大 20 の VLAN (1PVID および追加で最大 19 VID 付与)、VE の最大数は 16 であることから、19x16=304 個の VLAN を事前に用意するとよい。(なお 1LPAR あたり最大 256VE、HMC V7R7.2 以降では再起動なしに DLPAR で追加・削除が可能である。)

(7) VIOCにおけるディスクの可用性

ディスクの冗長化設計として、VIOC に仮想 Fiber をアサインし直接ディスクを見せる NPIV (N Port ID Virtualization) による方法と、VIOS で VIOC へ提供するディスクへのパスを冗長化する MPIO (PV-PV MPIO) の 2 つの方法がある。前者は VIOC 側で Disk I/O をラウンド・ロビン設定することにより I/O のスループットが上がる可能性がある。また、MPIO に比べ VIOS 側の負荷が少ないため限られたリソースを有効に使うことが可能である。しかし、外部ストレージのマイクロコード更新などの

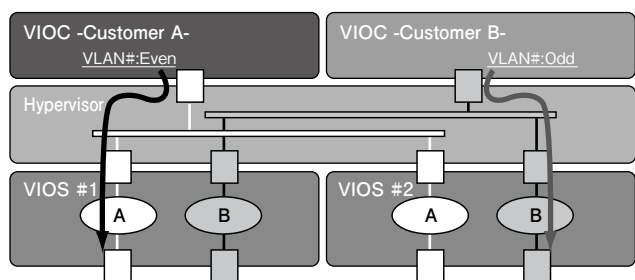


図2. VIOS-VLAN分散構成

際に、VIOC でのドライバーの対応が必要になる可能性があり、共用環境としての柔軟性要件を満たしていないため、MPIO 方式を推奨する。それぞれの構成におけるメリット・デメリットを表 2 にまとめた。

表. 2 MPIO - NPIV比較

	利点	難点
MPIO	サーバーに接続しているStorageの影響をVIOCは受けない	Active-standby構成になるためVIOCでは負荷分散しない
NPIV	VIOSへの負荷が低い VIOCでの負荷分散が可能	外部Storageの影響がある (VIOC-Storageドライバーの依存性)

VIOC の MPIO では I/O をラウンド・ロビンなどの負荷分散構成にすることはできず、active-standby による冗長構成となる。そこで、VIOC における MPIO のパス優先度を設定し VIOS の負荷分散設計を行う必要がある。

4. 設計検証と結果

おのおの設計方法について検証を行い、問題なく設計通りの動きをしていたかを確認する必要がある。ここではテスト環境で実際にテストしたテスト・ケースに基づいて設計の妥当性を検証する。

4.1 CPU

CPU について、同じ筐体上の区画で負荷を与え、想定された保証レートやバースト属性を発揮できるかを検証した。

検証区画 (LPAR) の CPU リソースは、Shared、Uncapped で設定し、同じ SPP にアサインした。SPP には 4 core 分の CPU を割り当てた。それぞれの LPAR に対して、CPU の重みも同じ値とした。与える CPU リソースは以下表 3 の通りである。

表3. CPUリソース・アサイン (Desired値)

	PU ^{※14}	VP ^{※15}	Memory
LPAR#1	0. 10	1	1GB
LPAR#2	0. 20	1	2GB
LPAR#3	0. 49	1	5GB

※ 13 Etherchannel とは複数のイーサネット・リンクを束ねて利用する技術。帯域の増加と可用性の向上が可能。

※ 14 Processing Units の略。1 core = 1.00 として 0.01 単位で論理区画に割り当てるリソース値

※ 15 Virtual Processors の略。仮想マシンに割り当てるプロセッサ数

以下表 4 に、各 LPAR に同時にかけた CPU 負荷のパターンを記載し、表 5 にその結果の PU 値を載せる。表 4 内の数値は PU に対する負荷の割合を示している。

表4. CPU負荷パターン

	Pattern 2	Pattern 3	Pattern 4
LPAR#1	200%	500%	1000%
LPAR#2	100%	250%	500%
LPAR#3	100%	140%	200%

表5. 負荷をかけた場合のPU値

	Pattern 2	Pattern 3	Pattern 4
LPAR#1	0.20	0.50	1.00
LPAR#2	0.20	0.50	1.00
LPAR#3	0.49	0.69	0.98

同時に負荷をかけた場合でも、十分なリソースがある場合、それぞれの LPAR において CPU は負荷を満たすバースト値を示した。

次に、リソースが逼迫している時の動作を検証するために、1 core の SPP を定義し、Uncapped で 0.2 PU の LPAR を2つアサインした。図 3 に示すように、初めに一方の LPAR に対してのみ負荷をかけ、少し時間が経過した後、別の LPAR にも負荷を与えて SPP 内でどのような CPU の取り合いをするかを検証した。

負荷をかけた LPAR_A は、初めは SPP 内のリソースをすべて使っていたが、LPAR_B に負荷がかかり LPAR_B へのリソース要求があると 2 つの LPAR で分け合う動きをしていることが分かる。

つまり、ある LPAR が高負荷状態で SPP 内のリソースが不足している状況でも、他の LPAR が稼働する際にはリソースを分け合う動きをする。また、リソースが使用されていない状態では、高負荷な LPAR に対してリソースが割り当てられ有効活用される。

同一の SPP にある他の区画と分け合う形になるため、業務のピークが重なる時期・時間は、想定しているバースト値にはならない。さらに、他の区画の影響を受け、検証時に測定したパフォーマンス値とはならない可能性があるため、十分に考慮する必要がある。

以上のことを考慮することにより、共用サービスにおいて柔軟性と高パフォーマンス要件を満たすことができる。

4.2 ネットワーク

Etherchannel によるネットワーク帯域の検証と VLAN 番号による割り振りの検証を行った。検証環境では、

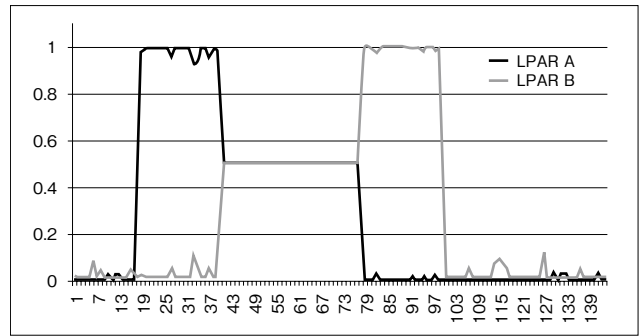


図3. 同一SPP内のCPUリソース配分

(注) 図中の横軸は経過時間 (秒) を示し、縦軸は PU を示す。各 LPAR の重みは同値である。

VIOS は 2 区画構成 (VIOS#1, VIOS#2) としそれぞれ A 系, B 系の VLAN を配置した。

仮想お客様 LPAR, CustA, CustB を作成し、それぞれ A 系, B 系の VLAN にアサインした。CustA, CustB それぞれに対して筐体外の区画からのネットワーク通信 (FTP 転送) を実施して、CustA, CustB でのネットワーク流量を計測した。

さらに SEA Failover を実施し A 系, B 系の active-standby 構成を入れ替え、障害時にも問題なく帯域を確保できるかを検証・確認した。その検証結果を表 6 に示す。

表6. ネットワーク検証値結果

VLAN	状態	帯域 (Gbps)
VIOS#1 A系	Main	2. 274972
VIOS#1 B系	Sub	2. 335883

上記結果から、VIOS に構成したそれぞれの Etherchannel において 2Gbps 以上のスループットが出ることが確認できたことから、想定通りの帯域を確保できていたことが分かる。

3 つの 1Gbps 物理ポートでの Etherchannel を構成した場合、本環境と同様に 75% の帯域を確保可能であると仮定すると、256Kbps の帯域を保証するには論理区画は 9 区画まで構成可能となり、複数 Etherchannel 構成することにより帯域を有効に使えることが分かる。

4.3 Mdisk Groupによるパフォーマンスの違い

本環境では、要件によりアサインするディスクを変えて運用する設計とした。SVC 配下で、高速・低速それぞれの Disk 筐体の Mdisk Group から作成された Vdisk に対して、ブロック・サイズの異なる書き込みアクセスを実施し、どの程度のパフォーマンスの違いが発生するのか

を検証し、その結果を表 7 に示す。

結果から、本環境では IBM System Storage DS3400 と DS8100 の Mdisk Group の違いにより約 70% 程度のパフォーマンスが異なることが分かった。この結果を参照し、個々の要件と照らし合わせてディスク装置をアサインする設計とした。SVC を使用した 1 つの仮想 Disk 環境として全体を利用することができる一方で、物理的に使用する Disk 筐体の違いによりパフォーマンスに違いがあることに注意する必要がある。パフォーマンス値については SAN を構成する環境によって異なるため、実測して設計を検討いただきたい。

表7. Mdisk Groupごとのパフォーマンス

Block Size	DS3K (MB/s)	DS8K (MB/s)	DS3K/DS8K (%)
4KB	239. 06	341. 93	69. 9
8KB	271. 15	382. 51	70. 9
64KB	315. 22	432. 04	73. 0
1024KB	318. 44	444. 95	71. 6

4.4 MPIOの優先度による負荷分散

MPIO によるディスク・アクセスについて優先度による IO 割り振りが可能であることを確認した。VIOS 2 区画からの MPIO 構成とした仮想ディスクをアサインした検証用 VIOC を作成した。この VIOC からは 1 つの仮想ディスクに対して 2 つの仮想 SCSI アダプター^{*16} から接続することになる。

この区画に対してアダプターごとに Disk I/O を計測した。検証結果を図 4 に示す。

上記検証結果から、優先度を高く設定した SCSI (vscsi0) からの I/O のみ計測されていることが分かる。

NPIV を利用して VIOC での IO 割り振りを常時ラウンド・ロビンで使用する設計も可能だが、マルチテナント型のサービスにおいては、障害時の復旧容易性や保守性を考慮すると、VIOS 上でディスク・アクセスを制御できる仮想 SCSI 構成が推奨される。NPIV を利用しない場合にも、MPIO による設計にて優先度を設定することにより、ディスク・アクセスの負荷を分散することが可能であることがこの検証結果より分かる。共用環境における VIOC のディスク・アクセス検討の際には MPIO による負荷分散方法を推奨する。

4.5 LPM (Live Partition Mobility)

LPAR を稼働した状態での LPM 時に、大量の Memory 転送が VIOS 間で発生するため、VIOC のパ

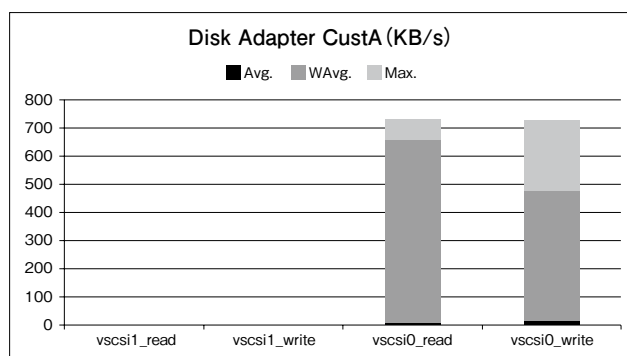


図4. VIOCのディスク Adapter I/O

フォーマンス劣化が発生する可能性がある。実際にどの程度の劣化があるかを検証した。

LPM 検証として、2 通りの方法を実施した。1 つ目は、表 8 に示すように通常時と LPM 時でそれぞれ負荷をかけて、Disk I/O (Write Only) と Memory (ops/s) パフォーマンスにおいてどの程度の差異があったかを確認した。2 つ目は、表 9 に示すように負荷がある場合とない場合での LPM にかかる時間を計測した。

表8. LPM時のパフォーマンス比較I

	通常	LPM時	比較 (%)
Disk IO (64KB)	562. 89	364. 39	64. 74
Memory (ops/s)	10358. 7	6414. 73	65. 85

表9. LPM時のパフォーマンス比較II

	通常 (sec)	高負荷時 (sec)	比較 (%)
LPM time	212	246	109. 0

結果から、LPM 時には Disk IO, Memory のパフォーマンスは通常時に比べて 7 割弱に減少し、実行時間は 10% 増加することが分かる。

検証した VIOC へは 4Gbytes のメモリーをアサインした。よって、高負荷をかけなかった場合の LPM 時には、19.3Mbps (4 × 1024 (M) / 212 (s)) の帯域を使用していたことが分かる。LPM 時の VIOC ディスク I/O パフォーマンス減少は、VIOS 上の CPU 負荷が増えたため VIOC へ提供する I/O 処理に影響がでたと考えられる。同様に、VIOC メモリーのパフォーマンス減少も、計測時に LPM 最後に発生する VIOS 上のメモリー処理一時停止の時間を含んでいたためだろう。一般には、LPM 最中には VIOC のパフォーマンスに大きな劣化はないと言われている。

*16 接続ディスクを仮想化して、VIOC で共有するためのアダプターのこと。

検証結果から LPM 自体の時間も増加することが分かった。運用時には LPM 実行時間を検討し、稼働状況への影響がないかを考慮して実行計画を検討いただきたい。

5. おわりに

本論文では、IBM Power Systems を使用して複数のお客様に仮想区画を提供する、マルチテナント型仮想化ホスティング・サービスにおけるシステム設計技法をさまざまな観点から設計・考察し、所望の動作をしているかについて検証を実施した。検証結果から、セキュリティー・パフォーマンスと可用性において設計通りの動作を確認することができた。この設計技法を用いて MCCS はサービスを開始しているので信頼性が高いと言える。また IBM 内、CIO 直下のセキュリティー管理組織からの IES 認定^{※17}も受けており、LPAR/STORAGE/NETWORK のセキュリティー上の分離についても問題ないことが証明されているため、安心して本設計を利用してほしい。

なお、本論文における IBM Power Systems の機能や性能に関する記述は執筆時点のものである点にご留意いただきたい。

謝辞

当論文の作成にあたっては、pMCCS の構築・運用メンバー、CPU リソースのサイジングの際 Bruno Blanchard さんに助言をいただき、各種検証についてもご協力いただきました。あらためて深謝いたします。

参考文献

- [1] 「クラウドコンピューティングと日本の競争力に関する研究会」報告書, <http://www.meti.go.jp/press/20100816001/20100816001-3.pdf>, p16.
- [2] IDC Japan 国内クラウドサービス市場予測, <http://www.idcjapan.co.jp/Press/Current/20100928Apr.html>
- [3] 丸山不二夫: "クラウド時代への備え", IT アーキテクト, Vol. 21, pp24-43 (2009).
- [4] IBM マネージド・クラウド・コンピューティング・サービス, <http://www.ibm.com/services/jp/index.wss/offerfamily/so/b1333005>
- [5] エンタープライズ向けクラウドサービス基盤「EINS/SPS」における設計ポイント, http://www.intec.co.jp/itj/itj11/contents/itj11_22-27.pdf
- [6] An IBM Redbooks publication, Implementing the IBM System Storage SAN Volume Controller V5.1, <http://publib-b.boulder.ibm.com/abstracts/sg246423.html>

※ 17 IBM 外部との企業間サービス / 接続は Inter-Enterprise Service (IES) と呼ばれ、これを実現するためには、IBM Corporation が規定する IES 認証プロセスを経て Global レベルでの正式承認を受ける必要がある。

- [7] マイクロ・パーティショニング, (Micro-Partitioning), <http://www.ibm.com/systems/jp/power/techinfo/lpar/micropartitioning.html>
- [8] An IBM Redbooks publication, IBM PowerVM Live Partition Mobility, <http://www.redbooks.ibm.com/abstracts/sg247460.html>
- [9] An IBM Redbooks publication, PowerVM Virtualization on IBM System p Introduction and Configuration Fourth Edition. <http://www.redbooks.ibm.com/abstracts/sg247940.html>



日本アイ・ビー・エム株式会社
グローバル・テクノロジー・サービス
サービス・デリバリー・アウトソーシング
IT スペシャリスト
※論文執筆時の情報であり、現在はエムエルアイ・システムズ株式会社に意向中。

山口 星志 Seiji Yamaguchi

[プロフィール]

2007 年、日本 IBM 入社。金融のお客様における SO デリバリーを担当し、システム基盤構築などを経験。2010 年、MLI に出向し分散系システム基盤運用を担当。IT スペシャリスト。
E34189@jp.ibm.com



日本アイ・ビー・エム株式会社
グローバル・テクノロジー・サービス
サービス・デリバリー・アウトソーシング
アソシエイト・サーティファイド・アーキテクト

青山 真巳 Manami Aoyama

[プロフィール]

1999 年、日本 IBM 入社。2000 年より e-ビジネスサービス部門での金融・流通・製造業のお客様のホスティングサービスの設計・構築・運用を担当。データセンターにおける担当お客様の仮想化システムへのリニューアル時の仮想化設計の経験を経て、2010 年より IBM MCCS for UNIX の全体設計および詳細設計を担当。
MANAMIAO@jp.ibm.com



日本アイ・ビー・エム株式会社
グローバル・テクノロジー・サービス
サービス・デリバリー・アウトソーシング
シニア・サーティファイド・アーキテクト

沢橋 松王 Matsuo Sawahashi

[プロフィール]

1991 年、日本 IBM 入社。ネットワーク、サーバー設計・構築、アプリケーション開発と幅広く経験し、2001 年より e-ビジネスサービス部門にて運用業務に従事。2009 年よりリード・アーキテクトとして SO のお客様向けパブリック・クラウド・サービスである IBM MCCS の企画・設計・開発を指揮。
MATSUOS@jp.ibm.com