

# 拡張性を考慮した小規模仮想化基盤の構築

松浦 知史<sup>1,a)</sup> 森 健人<sup>1,b)</sup> 金 勇<sup>1,c)</sup> 友石 正彦<sup>1,d)</sup>

**概要：**大学内のログ情報を扱う組織において構築した小規模な仮想化基盤について紹介する。少額の投資から始められ、段階的に機器を導入する事が可能な構成を目指した。ベースは HP Proliant および HDD エンクロージャとし、その他に実験用マシン (Mac mini) を組み合わせて初期段階の構成とした。現段階では vSphere 5.5 および仮想マシンのバージョン 10 という組み合わせが安定運用に適していることが分かった。環境構築に当たってはソフトウェアのバージョンの組み合わせで不具合が発生することも多く、その様な失敗談も踏まえて環境構築の過程を記す。

## 1. はじめに

ICT 基盤はあらゆる組織の活動を支える一方で、基盤整備にかかるコストやその基盤を維持するための運用コストが問題となっている。大学もその例外では無く、コスト削減や将来の拡張性を考慮し仮想化基盤の構築やクラウド環境を利用した ICT 基盤の見直しが進んでいる [1], [2], [3]。

本論文はこのような大学全体における ICT 基盤の戦略を示すものではなく、学内の一組織が小規模な仮想化基盤環境構築した事例を示すものである。組織に寄っては扱うデータの性質や量が特殊であり、全学向けに整備された仮想化基盤の利用が困難な場合がある。その様なケースの一助となるために具体的な機器や設定手順を要求要件と共に整理することが、本論文の主な目的である。

2 章では要求要件や予算などの制約条件を整理し、3 章ではそれら要件を満たすハードウェアおよびソフトウェア構成を記す。4 章では環境構築過程で遭遇したトラブルや簡単な性能評価を紹介する。

## 2. 小規模仮想化基盤の要件と設計

仮想化基盤を導入する組織における制約、要求要件を整理し、設計方針を示す。予算の制約が厳しく高額なサーバ等を導入する事は出来ないため、少額規模の投資を繰り返す事で規模を拡張できるシステムを念頭に設計を行う。

### 2.1 制約と要求要件

学内における様々なネットワーク機器等のログを扱う組織において、仮想化基盤を構築することを想定する。ストレージ、仮想ホスト、予算に関する要件を下記に整理する。

#### 2.1.1 ストレージ

まず、ストレージに関する要求を列挙する。

- 容量：数百 TB まで拡張可能なシステムである事
- データ保護：RAID10 をサポートする事
- 接続方式：iSCSI をサポートする事
- NIC：10Gbps のインターフェースを備えること

現状で利用しているログのデータ量は合計で数 TB 程度であり一般的な 1U サイズのサーバマシンが備える容量で十分である。しかし今後はネットワーク機器だけでなく、セキュリティ機器等のログやネットワークフォレンジック用のログ等も想定されるため扱うログの種類も数倍程度増加すると見込まれる。また対外線の性能も 10Gbps から 100Gbps に増加することが計画されており、必要に応じて数十 TB から数百 TB まで拡張可能なストレージシステムが求められる。

データ保護のために RAID の採用は必須である。RAID10 はミラーリングとストライピングを組み合わせた手法で容量は半分になるが、速度も確保されている。採用の一番の理由は障害の局所性である。RAID5/6 では障害の影響が複数の HDD に及び、データ復旧時に長時間サービスが停止してしまう可能性がある。また全てのパリティチェックをするために大きな負荷がかかり続け、そのタイミングで復旧中の HDD が破損しデータ復旧に失敗するという危険性も存在する。RAID10 は設置する HDD が増加してもミラーしている 2 台の HDD の中だけで障害の影響が抑えられるという局所性を持ち、リビルドもミラーリングしてい

<sup>1</sup> 東京工業大学 Tokyo Institute of Technology, Meguro, Tokyo 152-8550, Japan

a) matsuura@gsc.titech.ac.jp

b) mori@cert.titech.ac.jp

c) yong@gsc.titech.ac.jp

d) tomoishi@noc.titech.ac.jp

る HDD から情報をコピーするだけである。出来るだけシンプルな構成を採る方が良いと判断し RAID10 を要件とした。

ライブマイグレーションの様な仮想化環境特有の機能を実現するためには共有ストレージが欠かせない。FC/iSCSI/NFS が代表的な接続方法である。機器構成がシンプルで一定の性能も期待出来るため iSCSI を要件とした。

現在有線 LAN 用途の NIC は 1Gbps が主流である。これは現状のログ情報を収集するには十分な性能である。一方で 1Gbps は市販の HDD(SATA) の速度を数割程度下回る性能であり、ネットワーク接続された共有ストレージとして使用する場合は性能低下を招く。また RAID10 による速度向上や SSD の採用を考えると一層の性能低下が懸念される。その為に 10Gbps の NIC を備える必要がある事を要件とした。

### 2.1.2 ネットワーク

ログの収集・分析が主目的であり、通信記録等は外部からアクセス出来ない環境で転送しなくてはならない。そのため、学内に閉じたログの転送経路が必要である。また共有ストレージを利用するための経路も別セグメントで用意しておくことが望ましい。加えて、ログの分析結果等を学内全域に掲示する場合はグローバルなセグメントが必要になる。現状の扱っているログの量を考慮すると、転送性能は 1Gbps で十分サービス可能である。一方で将来的なログの増大を考慮すると共有ストレージを接続するセグメント内およびログ転送に利用するセグメントの上流側は 10Gbps に拡張できる事が望ましい。

### 2.1.3 仮想ホスト

仮想ホストに求められる事項としては”Windows 端末からも容易に利用できるシステムである事”が挙げられる。これは想定される利用環境による事項で、教員以外にも事務職員が仮想化基盤環境を利用するためである。事務職員は基本的に Windows 端末を利用しており、そこからシステムの制御を行いやすい環境を構築する必要がある。

### 2.1.4 予算

予算規模は 100-200 万円程度を想定する。また単年度でなく複数年度の予算が割り当てられているものとする。そのため、機材等の購入時は基本的に 100 万円以下、可能であれば 50 万円以下である事を要件とする。

## 2.2 システム設計

前節の要件を満たすようにシステム設計を行う。

### 2.2.1 ネットワーク構成

まずネットワークおよび物理構成を図 1 に示す。前節で示したネットワークに対する要求事項を満たすために、ネットワークは 3 つのセグメントによって構成される。グローバルセグメントはグローバル IP アドレスからなるセグメントであり学内外からのアクセスを想定するホストが

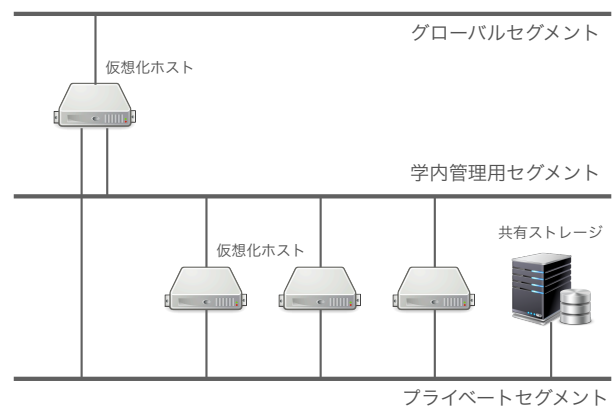


図 1 ネットワークおよび物理構成

接続される。現状では学内全域に向けてサービスを提供する事を想定しており、集約されたデータの提示 (WEB サーバ) など全学向けのサービスを提供する仮想マシンを配置する事を想定している。学内管理用セグメントは学内に閉じたプライベート IP アドレスからなるセグメントである。このセグメントはセキュリティやネットワーク機器等の存在するセグメントと接続可能で有り、ログの収集経路として利用される。また機器の設定等の管理を行う場合もこのセグメントを利用する。プライベートセグメントは今回構築する小規模仮想化基盤の内側に存在するネットワークである。仮想化基盤を支えるためのセグメントであり共有ストレージの設置が想定されている。またライブマイグレーションの様な仮想化基盤特有のデータもこのセグメントを通してやり取りされる。

### 2.2.2 仮想化ホストのマシン構成

前節で示したストレージに関する要求事項を満たすように、また拡張性や予算面も考慮しながらマシン構成の設計を行った。容量に関しては、HDD ディスクエンクロージャを採用することで対応する事にした。iSCSI に対応したネットワークストレージ等も多く販売されているが、予算の制約上導入が難しく、単にホストマシンのディスク容量を増加させる HDD エンクロージャを選択した。そのため iSCSI の環境は別途準備する必要が生じる。分散ストレージの技術も多数提案されており、サーバ台数を増やすことで容量を確保する方法も存在するが、分散ストレージを運用するには一定水準の技術力が求められ、また選択した分散ストレージ特有の制約を受けながらアプリケーション開発を行わなくてはならない。技術力の高い担当者の異動を機にシステム運用が継続出来ないといった事態を避けるため、サーバの台数を増やして容量を増やすのではなく、HDD ディスクエンクロージャを増やすことで容量を増やす方法を選んだ。

## 3. 仮想化基盤の実装

前章の要件及び設計を受け、本章では具体的なハード

ウェアおよびソフトウェアを示しながら仮想期基盤の実装手順を示す。

### 3.1 ハードウェア構成

全体のハードウェア構成を図2に示し、下記に用途毎の詳細を記す。

#### 3.1.1 仮想ホスト用サーバ

下記に仮想化ホストとして利用しているマシンスペックを示す。全体で6台のサーバを用意した。いずれのサーバもCPUの数は1つである。HPサーバには8GB\*4枚のメモリ、4TB\*4個のディスクを搭載した。Mac miniサーバには8GB\*2枚のメモリ、240GB\*2個のディスクを搭載した。HPサーバのみがRAID10となっているが、これはハードウェアRAID機能の有無が関係している。また、2.1.4項で示した予算の要求を満たす形でサーバの選択を行った。

- HP proliant DL320e gen8 (2台)
  - CPU: Xeon E3-1240v2 (4 core) 3.4 GHz
  - MEM: 32GB ADTEC ADS12800D-E8G4
  - HDD: 8TB (RAID10) HGST 0S03361
  - PCIe: HP Ethernet 10Gb 2ポート 530T
  - PCIe: HP Smart アレイ P431/2GB FBWC (1台のみ)
- Mac mini server 2012 late (4台)
  - CPU: Core i7 (4 core) 2.6 GHz
  - MEM: 16GB Transcend JM1600KSH-16GK
  - SSD: 480GB (240GB\*2) Intel SSDSC2CW240A3

P431を備えるサーバについて補足しておく。P431はD3600(HDDディスクエンクロージャ)と接続するためのボードである。10GBASE-Tを2ポート備える530TとP431を同時にDL320e gen8のサーバに設置する事は一見出来ないように思われる。これはDL320eのシステム構成図[4]を参照する事で解決した。HPオンラインストアではサーバ購入時に様々なオプションを選択可能であるが、システム構成図にはより網羅的なオプションに関する記述が存在する。実際にはDL320e(サーバ)と530TとP431(オプション)は別に購入し、別途取り付け作業を行った。

#### 3.1.2 HDD ディスクエンクロージャ

HPサーバと接続するHDDディスクエンクロージャのスペックは下記の通りである。D3600はサーバと直接接続するタイプのエンクロージャであり、上記で示したHP Smartアレイ P431を持つHPサーバと接続される。D3600は12個のドライブベイを備えており、今回は4TB(SATA)のHDDを12個搭載しRAID10を採用することで24TBの容量を確保した。下記にD3600のスペックを示す。

- HP D3600 Enclosure (1台)
  - HDD: 24TB (RAID10) HGST 0S03361 (12個)
  - IF: SAS 12Gbps

D3600は最大8台までカスケード接続が可能であり、現在の構成であれば合計で192TB(RAID10)の容量を確保す

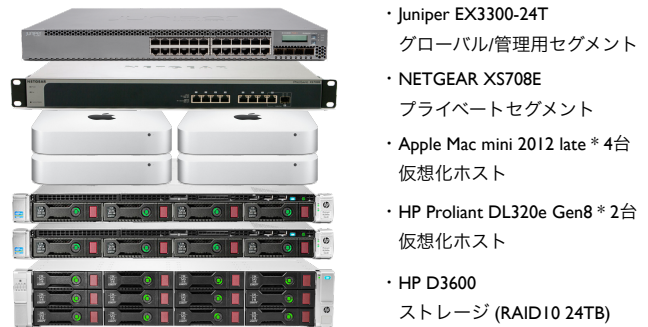


図2 ハードウェア構成

ることが出来る。インターフェイスはSAS 12Gbpsと記しているが、今回はSATAのHDDを利用しており、半分の6Gbpsで接続される。SATAをSASのディスク(SAS 3.0)と置き換えることで本来のSAS 12Gbpsで接続することが可能である。上記のハードウェア構成を採ることにより、2.1.1項に示した4つの要求要件の中でiSCSIによる接続方式以外は全て満たすことが可能である。

#### 3.1.3 ネットワークスイッチ

ネットワーク機器は下記のスイッチを利用している。Juniperスイッチは上流と仮想化基盤を繋ぐ役割を果たしている。2.2.1項で示した様に仮想化基盤の環境では3つのセグメントを想定しており、VLANによってJuniper配下にグローバルおよび学内管理用セグメントを実現している。NETGEARスイッチは共有ストレージに接続するためだけに利用するスイッチである。このようなネットワーク機器の構成を採ることによって2.1.2項に示したネットワーク対する要求要件を満たす事が可能である。

- Juniper EX3300-24T (1台)
  - 10GBASE-T 24ポート
  - SFP+ 4ポート
- NETGEAR XS708E-100AJS (1台)
  - 10GBASE-T 8ポート
  - SFP+ 1ポート

### 3.2 ソフトウェア構成

ソフトウェアに関しては基本的に運用担当者の変更に耐えられるように、利用者が多いものを選択した。仮想化ホストにはVMWare vSphere 5.5を採用し、その上で動作する仮想マシンには基本的にubuntu Linux 14.04 LTSを採用した。

#### 3.2.1 仮想化ソフトウェア

仮想化のソフトウェアとしてVMWare vSphere 5.5を採用した。今回構築する仮想化基盤環境は教員や技術職員に加えて事務職員も利用する。事務職員はWindows端末を利用しており、そのような端末からでも操作が可能なVMwareを選択した。また複数のvSphereを統合的に管理するvCenterも利用した。vSphere、vCenterおよび仮想

マシンのバージョンを下記に記す。

- VMware vSphere Ver. 5.5
- VMWare vCenter Ver. 5.5
- VMWare 仮想マシン Ver. 10

今回のハードウェア構成では上記のバージョンが最も安定して動作した。付録 A.1 にインストール手順を記す。また詳細は次章に記すが、現在最新の Ver. 6.0 ではディスク性能が大幅に低下した。将来的には VMware 社の対応状況を確認し、vSphere/vCenter は Ver. 6.0 に、またそれに合わせて仮想マシンのバージョンも更新する計画である。

### 3.2.2 iSCSI のサポート

iSCSI に対応し、10GT の NIC を備える機器は高価であり、予算制約である 100 万円以下を満たすことが出来なかった。そのため、自前で iSCSI の環境を構築することとした。HDD エンクロージャが接続されている HP Proliant に vSphere をインストールし、その仮想マシン (ubuntu Linux) 上に iSCSI target(LIO) を立ち上げて、仮想化環境における共有ストレージとして利用した。図 3 に iSCSI の構成を示す。図中右側にある複数の仮想化ホストは各々の VMware ESXi が備える iSCSI software adapter を通して遠隔のディスク (HDD エンクロージャが備えるディスク) をローカルに存在するディスクと同じように利用することが可能である。下記にソフトウェアとそのバージョンを記す。

- ubuntu 14.04.02 LTS server
- targetcli 2.1-1

上記の仮想マシンに 32TB の HDD を thin provisioning 形式で追加し、これを iSCSI 用のディスクとして割り当てた。HDD エンクロージャは 24TB の容量しか持たないが、使用する段階でディスク領域を割り当てる thin provisioning 形式を選択することでこの様な割り当ても可能となる。targetcli を利用し、32TB の追加ハードディスクを iSCSI を通して共有出来るように設定した。他の仮想ホストから iSCSI を利用してディスクをマウントし、仮想マシンの作成やライブマイグレーションなどを行い良好に動作した。参考までに vSphere 5.5 では 62TB の上限がある。また 60TB 程度を割り当てた場合は動作が不安定で、共有ストレージとして利用する事は出来なかった。付録 A.2 に iSCSI 環境の構築手順を記す。

上記の様な設定を行い、iSCSI のサポートを行った。また、2 章で挙げた要求要件の全てを本章を通して満たすことが可能である事を示した。

## 4. 失敗事例および簡易な性能評価

### 4.1 ESXi 6.0 に関する問題

#### 4.1.1 HP proliant における書き込み速度

仮想化マシンのバージョン 11(ESXi 6.0) を利用すると書き込み速度が非常に遅くなる (20MB/s 程度と USB メモリ程度の速度)。これは VMware 社も認識している問題で

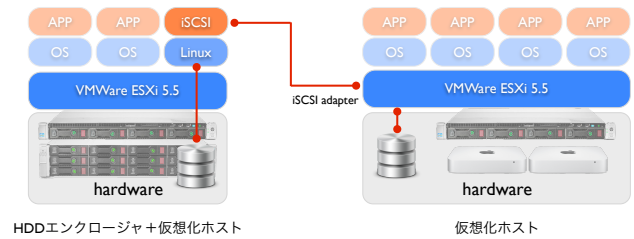


図 3 iSCSI による共有ストレージの構築

有り、CPU が XEON E3、E5、E7 の v2 以上であればサポートされる予定である。

ESXi 6.0 上で仮想マシンバージョン 10 および 8 を試したが、問題は出なかった。また、バージョン 10 の方が全体的にパフォーマンスが良かった。ESXi 6.0 は比較的新しいバージョンなので、安定運用も考慮し ESXi 5.5 を利用する事とした。5.5 以上であれば仮想マシンのバージョン 10 が利用可能である。

#### 4.1.2 mac mini の筐体温度に関する問題

Mac mini はそもそも ESXi で正式にサポートされている訳では無いが、ESXi 6.0 からインストールが非常に楽になり、6.0 インストール用の ISO イメージを利用する事で事前の設定等無しでインストールが可能である。ただファンが正常に動作しないよう室内の温度が 27 度程度上昇したときに半数以上の ESXi が反応しなくなり、再起動を余儀なくされた。ESXi 5.5 ではファンは回転してはいるが一定で風量も少なく手で筐体を触れても熱く感じるほどである。USB のファン (TIMELY USB ファン BIGFAN120U for Men) を利用し、熱対策を取ることで温度が下がり動作が安定した。

HP Proliant におけるディスク性能、mac mini の熱対策を考慮すると ESXi 5.5 かつ仮想マシン ver10 が妥当であると判断した。

### 4.2 HDD のホットスワップについて

HP proliant は HDD の挿入口が 4 つ有り、今回は RAID10 の構成を採った。この機器はホットスワップ対応製品で実際に動作中の HDD をマウンタごと引き抜き、新品の HDD と取り替えて指し直した所、正常に動作した。つまり HDD リビルド中もホスト及び仮想マシンは動作し続け、6-8 時間程度かけてリビルドが完了した。

このホットスワップ作業を素早く行おうと考え別途マウンタと HDD を準備した。マウンタと HDD をセットで事前に準備しておくことで、トラブル時の作業コストを低減する事を狙っている。しかし、HDD だけでなくマウンタも含めて交換したところ、正しくリビルドが開始された様に思えたが数分後には HP proliant の電源が落ちた状態になった。ここで電源を投入したところリビルドが開始され、最終的には正常な状態に復帰した。マウンタには ID

が振られているのか、認識するために再起動が必要なようである。HDD 交換するときはマウンタを引き抜き、HDD だけを交換した状態で元に戻すのが良いと考えられる。

#### 4.3 仮想マシン ver8 および iSCSI に関する問題

仮想マシン (ubuntu14.04) 上に iSCSI target(LIO) を立ち上げて、仮想化環境における共有ストレージとして利用した。また、ローカルセグメント (1Gbps) のみを用いて検証した。ここで iSCSI target を仮想マシン ver8 かつ HDD を使用した場合に問題が生じた。具体的な症状としてはこの共有ストレージに接続している ESXi のホストがアクセス不能になってしまった。アクセス不能になるまではホストによって差があり、2日から5日程度であった。設定の見直し、再起動や再インストールを繰り返したが状況は改善されなかった。一度この不具合が発生すると再起動中に停止してしまう。起動プロセスとしては8割、9割程度は完了している様で、最後に”swapobj loaded successfully”とメッセージが表示されそれ以上起動プロセスが進行しない状態であった。この原因は詳細までは判明しなかったが iSCSI target のパフォーマンスに影響されていると予想される。問題が起きた環境では書き込み速度が 40MB/s - 125MB/s とパフォーマンスが安定しなかった。ver8 であっても mac mini(SSD を利用) 上で仮想マシンを立ち上げそこで iSCSI target を動かした場合は書き込み速度は 120MB/s - 125MB/s と 1Gbps のネットワークをほぼ埋め尽くしながら安定したパフォーマンスを発揮した。また、上記の様なホストへのアクセス不能および再起動不能といった状況は発生しなかった。同様に仮想マシンを ver10 にし、HP proliant(HDD を利用) 上で iSCSI target を動かした場合も書き込み速度は 1Gbps のネットワークを埋める形で安定したパフォーマンスを示し、問題も発生しなかった。そのため、仮想マシンのバージョンは 8 ではなく 10 を採用した。

#### 4.4 ESXi 上の仮想マシンに 4TB 以上のディスクを追加する場合の注意点

ESXi5.5 の仮想マシンは 62TB までハードディスクを割り当てることが可能であるが、追加のハードディスクには 4TB の制限がある。しかしこれは vSphere Client を利用した場合の制約で、vSphere WEB Client を利用すれば 4TB を越える追加のディスクを作成することが可能である。今回は上記の方法で iSCSI 用に 32TB の追加ディスクを割り当てた。

#### 4.5 thin / thick provisioning について

一般的に thin は遅いがスペースを節約でき、thick は早いスペースを多く消費すると言われている。これは事前に領域を確保するかしないかの違いで、OS から見ると同

じ 100GB でも thin の場合は問い合わせがあってから確保してその領域を返すために速度は低下するがスペースの利用効率は高い。thick はその逆である。実際にどの程度の違いが出るか確認してみた。用意した環境は下記の通りである。

- server-A: 1CPU 1GB メモリ 300GB (thin)
- server-A: 1CPU 4GB メモリ 16TB (thin)
- server-B: 1CPU 1GB メモリ 100GB (thick lazy)
- server-B: 1CPU 1GB メモリ 100GB (thick eager)
- server-B: 1CPU 4GB メモリ 16TB (thin)

OS はすべて ubuntu 14.04.2 LTS server 版である。評価したのは dd を利用した簡易なシーケンシャル write の性能である。コマンドは下記を利用した。

```
# time dd if=/dev/zero of=/tmp/hdparm.tmp \
ibs=1M obs=1M count=1024
```

性能の結果は thin が必ずしも遅いわけでは無かった。これは VMWare が thin のパフォーマンス向上のために色々と工夫を凝らしているためと思われる。結果を示すと以下の通りである。

- 85MB/s A: 1CPU 1GB メモリ 300GB (thin)
- 300MB/s A: 1CPU 4GB メモリ 16TB (thin)
- 85MB/s B: 1CPU 1GB メモリ 100GB (thick lazy)
- 150MB/s B: 1CPU 1GB メモリ 100GB (thick eager)
- 300MB/s B: 1CPU 4GB メモリ 16TB (thin)

結果的には thin で容量を大きく確保したものが一番早かった。ちなみに利用した日立製の HDD は 130-140MB/s 程度の性能である。速度が向上しているのは RAID10 のためと思われる。書き込み性能自体は仮想化しているからと言ってそれほど低下しているわけでは無く、場合に寄っては差が殆ど無い。参考までに 5K iMac に付属している 1TB SSD は 700MB/s の性能を示した (mac の dd では ibs, obs 等の値に M は使えない 1000000 と書く)。

thin provisioning を採用したときの挙動を観察すると、論理的に設定した容量の 1-2%程度は事前に確保しておくようである。つまり 16TB のボリュームを作成すると OS 自体の容量は 1GB 程度で何もしていなくても 140GB 以上の容量が消費されてしまう。この事前に確保された領域は thin といえども高速に読み書き出来るようで、最高性能を発揮した。また 1GB のファイルを何度も作成および消去したが、消費される容量は変化しなかった。一方 300GB の thin では性能が低かった。とはいえ thick lazy とほぼ同じである。ここでは事前に 8GB が消費されており、1GB のファイルを作成/消去するたびに消費される容量も 1GB ずつ増加した。これは thin の正常な動作で有り、OS から見るとファイルを消去して容量の空きが増えるが thin の領域は開放されない。設定した上限に近づくと最適化がかかる様である。

thin が何処まで使えるか疑問を持っており簡単な性能評

価を行ったが、予想に反して thin が高いパフォーマンスを示した。書き込みや削除が多く発生する環境ではどの程度のパフォーマンスを発揮できるか、またスペースの節約に繋がるかまだまだ疑問は残るが、十分に利用できるシーンはありそうである。

capped volume を持つ DB を運用する場合は thick eager 辺りで最初に容量を確保し、ひたすらログが増え続けるようなサーバは thin で大きな容量を取るのが良いかも知れない。ある程度長期で運用しないと結論づけられ無いと考える。

## 5. おわりに

学内の一組織が管理する小規模な仮想化基盤の構築過程について述べた。少額の予算を複数年度に渡って執行する事を想定し、予算制約の下でも将来的な拡張が可能なシステム構成を提案した。具体的には HP ProLiant を仮想ホスト用のマシンとし、ディスクとしてはデイジーチェーン可能な HP D3600 を採用した。また検証の結果、仮想化ソフトウェアとしては VMware vSphere 5.5、仮想マシンのバージョンとしては ver.10 が安定している事が分かった。

現状では本仮想化基盤は検証段階である。今後はハードウェア的な拡張を行うと同時に、ログの収集や分析などを行って行く計画である。この様な本番環境における各種パフォーマンス計測を行い、より現実に即した具体的な評価を行う予定である。

**謝辞** 本研究は JSPS 科研費 15K00115 の助成を受けたものである。

## 参考文献

- [1] 柏崎礼生, 宮永勢次, 森原一郎: 大阪大学における仮想化基盤の増強とクラウド戦略, インターネットと運用技術シンポジウム 2014 論文集, Vol. 2014, pp. 93-100 (2014).
- [2] 前田香織, 末松伸朗, 北村俊明: 広島市立大学における情報ネットワークシステムのクラウド移行, インターネットと運用技術シンポジウム 2015 論文集, Vol. 2015-IOT-28, No.19, pp. 1-6 (2015).
- [3] 柏崎礼生: スモールスタートで始める大学の仮想化基盤の構築と運用の実情, インターネットと運用技術シンポジウム 2012 論文集, Vol. 2012, pp. 94-101 (2012).
- [4] HPE ProLiant DL320e Gen8 構成図 (online), 入手先 <[http://h50146.www5.hp.com/products/servers/proliant/system\\_pdf/dl320egen8.pdf](http://h50146.www5.hp.com/products/servers/proliant/system_pdf/dl320egen8.pdf)> (2015.12.10).

## 付 録

### A.1 vSphere (ESXi) インストール手順

HP ProLiant に vSphere (ESXi) 5.5 をインストールする手順を示す。様々なインストール方法があるが、USB メモリにインストール用の ISO イメージをファイルして (USB メモリ自体を起動ディスクにする必要は無い) コピーして

おき、インストーラーに従って ISO イメージを選択する方法が最も簡易と思われる。

#### A.1.1 環境

- HP DL320e Gen8 HP 4LFF モデル (6755970-ARIM)
  - HP Dynamic Smart アレイ B120i コントローラー
- vSphere standard 5.5

#### A.1.2 RAID10 の設定

BIOS 起動時に [F9] を押して設定する。'Enable Dynamic HP Smart Array B120i RAID Support' を選択する。再起動時に RAID コントローラの初期化プロセスで [F5] を押すとセットアップが可能となる。GUI が立ち上がりマウスも利用して設定を進める。4TB \* 4 個の HDD を選択し、RAID1+0 で論理ボリュームを作成する。特に必要が無ければオプションはデフォルトのまま構わない。

#### A.1.3 HP ProLiant のファームウェアアップデート

起動時に [F10] を押して"メンテナンスの実行" - "Intelligent Provisioning" と辿って自動的にネットワークの設定を行う。その後"ファームウェアの更新"を実行する。DHCP 等で IP アドレスを振っても上記の通りネットワークの設定を行っておかないと、ファームウェア更新時にネットワーク接続エラーで hp.com にアクセス出来ないといったエラーが発生してしまう。

#### A.1.4 VMware ESXi 5.5 のインストール

ESXi の起動ディスクを準備するためには dd だけでなく fdisk 等も利用した作業が必要であり、デバイスドライバも別途インストールする必要がある。ESXi には HP サーバ用のカスタムインストールイメージが用意されており、この ISO イメージを手元の USB メモリにファイルとしてコピーインストールを行う。この ISO イメージを利用する事で、デバイスドライバのインストールも自動で行われる。

BIOS 起動時に [F10] を押すとカスタム CD を利用したインストール画面が立ち上がる。これは GUI でマウスを利用して操作する。基本的にデフォルトのまま進めれば良い。インストールディスクを選択するページで ISO イメージを含んだ USB を差し込み、"Refresh" ボタンを押すと新規に差し込んだ USB がフォルダ形式で追加される。その中から ISO イメージを選択することでインストールを進めることができる。今回使用した ISO イメージのファイル名は下記の通りである。

VMware-ESXi-5.5.0-Update2-2068190-HP-5.77.3-Nov2014.iso

ISO を選択すると後は RAID コントローラのドライバ等も含めてインストールされる。今回の構成では 20-30 分程度かかった。

### A.1.5 vCenter のインストール

複数台の ESXi ホストを束ねるために vCenter が存在する。vCenter によって管理が一元化されるだけでなく、ESXi をクラスタ化したり、ライブマイグレーションやレプリケーション、障害時におけるリソース退避等々が可能となる。vCenter は仮想マシンの一つとしてインストールした。

VMware-vCenter-Server-Appliance-5.5.0.20400-2442330\_OVF10.ova

OVA 形式の上記ファイルを利用した。これは SUSE Linux をベースにした vCenter の仮想化ファイルで、vSphere Client を利用してファイルを設置するだけですぐに vCenter を起動することが可能である。ちなみに vcenter 5.5 は 100 ホスト/3000VM まで管理することができる。また vCenter にはブラウザで 5480 番ポートにアクセスする事でログインできる。user/pass = root/vmware が初期設定である。また vSphere WEB Client は証明書の関係で最初接続しようとするエラーが出る。5480 番でログインした時に admin の項目でパスワードを変更すると共に”certificate regeneration enabled”を YES にする。これで vSphere WEB Client が利用できる。

### A.1.6 個別の仮想マシンのインストール

vSphere Client のあるマシン上に ISO イメージがあれば簡単にインストール可能である。下記 2 点を抑えておくとスムーズにインストール可能である。ただし、仮想マシンのバージョン 10 をインストールするときは vSphere WEB Client を利用する必要があり別途注意が必要である。下記の方法は vSphere (ESXi) 5.0、仮想マシンのバージョン 8 以下の環境で有効である。

- オプションで起動時に強制的に BIOS 画面になるように設定する
  - これは電源投入後に時間を稼ぎその間に ISO をドライブとして割り当てるためである
- ISO が正しくドライブとして認識されたかを確認する
  - たまに接続中となり認識できない事があるので vSphere client を起動し直す

#### A.1.6.1 仮想マシン ver10 のインストール手順

vSphere WEB Client と vSphere Client には機能面で違いがある。vSphere WEB Client の方が高機能であり、設定出来る項目も多い。ESXi 5.5 をインストールした場合に仮想マシン ver10 を作成するには WEB Client の利用が必須となる。また ver8 よりも上のバージョンの設定を行う場合は WEB Client を利用する方が良い。注意すべきはテンプレート化する場合で、ver10 の仮想マシンを vSphere Client でテンプレート化しようすると処理中の状態が続き完了しない (通常は数分)。

仮想マシンのインストール時はコンソールを利用する必要があるが、こちらは vSphere Client の機能を利用したい

(WEB Client では追加のライセンスが必要)。しかし ver10 の仮想マシンを作成するとインストール用の ISO を利用する事が出来ない。この問題を解消するために、下記の手順に従うと良い。

- (1) WEB Client ver10 の仮想マシンを作成する
- (2) インストール用 ISO イメージを仮想ホストにアップロードする
- (3) WEB Client を利用して強制的に BIOS が起動するように設定する
- (4) 仮想マシンの電源を入れる
- (5) WEB Client を利用して ISO を CD/DVD ドライブとして設定する
- (6) vSphere Client を利用してコンソールから BIOS に入り、再起動する
- (7) vSphere Client を利用してコンソール上でインストールを進める

## A.2 ubuntu での iSCSI target 構築手順

### A.2.1 検証環境

- ESXi 5.5
- ubuntu 14.04.02 LTS server
- targetcli 2.1-1
- ローカルセグメント (1Gbps) のみを利用

### A.2.2 ubuntu のインストールと追加 HDD の割り当て

ESXi 上に ubuntu をインストールする。スペックは下記の通り。

- HDD1 : 20GB (thin プロビジョニング) OS(ubuntu)
- HDD2 : 1TB (thin プロビジョニング) iSCSI target
- MEM : 8GB

4TB 以上の HDD を追加する場合は注意が必要。基本的に web client から編集して、メモリ量の変更、HDD の追加を行うと良い。

### A.2.3 targetcli のインストール

```
> sudo apt-get -y install targetcli
```

apt でインストールするのが良い。依存関係の範囲が広く 1GB 程度容量を消費する。ソースからインストールしようとするこの依存関係の問題で非常に苦勞する。公式サイトでも apt 等の利用を推奨している。

### A.2.4 targetcli を使って iSCSI target を設定する

ディスクが正しく割り当たっているか確認する。ディスクを割り当てただけなので、最初はフォーマットもされていないし、マウントもされていない (下記状態が正しい)。

```
> sudo fdisk -l /dev/sdb
Disk /dev/sdb: 1099.5 GB, 1099511627776 bytes
ヘッド 255, セクタ 63, シリンダ 133674, 合計 2147483648 セクタ
Units = セクタ数 of 1 * 512 = 512 バイト
セクタサイズ (論理 / 物理): 512 バイト / 512 バイト
I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト
ディスク識別子: 0x00000000
```

ディスク /dev/sdb は正常なパーティションテーブルを含んでいません

targetcli は sudo で実行する。CLI という名前が付いているようにコマンドを使って操作する。また targetcli はコンテキストが重要で、どのディレクトリにいるかでコマンドの意味合いが変わってくる事を意識する必要がある。

#### A.2.4.1 HDD の割り当て

target に設定する領域を指定する。今回はディスクをそのまま選ぶので iblock に移動する。file を利用する場合は fileio、SCSI デバイスを利用する場合は pscsi に移動する。name は適当に決めれば良い。全体の状況を知りたい場合はルートに移動して ls すれば良い。

```
> sudo targetcli
/> cd backstores/iblock
/backstores/iblock> create name=block_backend dev=/dev/sdb
Generating a wwn serial.
Created iblock storage object block_backend using /dev/sdb.
/backstores/iblock> cd /
/> ls
o- / ..... [....]
o- backstores ..... [....]
| o- fileio ..... [0 Storage Object]
| o- iblock ..... [1 Storage Object]
| | o- block_backend ..... [/dev/sdb deactivated]
| o- pscsi ..... [0 Storage Object]
| o- rd_dr ..... [0 Storage Object]
| o- rd_mcp ..... [0 Storage Object]
o- ib_srpt ..... [0 Targets]
o- iscsi ..... [0 Targets]
o- loopback ..... [0 Targets]
o- qla2xxx ..... [0 Targets]
o- tcm_fc ..... [0 Targets]
```

#### A.2.4.2 iSCSI の作成

```
> cd iscsi
/iscsi> create
Created target iqn.2003-01.org.linux-iscsi.ubuntu.x8664:sn.820c2baa3d89.
Selected TPG Tag 1.
Successfully created TPG 1.
```

#### A.2.4.3 LUN、IP アドレス、イニシエータ、CHAP 認証の設定

上記で設定した iSCSI target に対して設定を加えていく。まずは tpg1 ディレクトリに移動する。LUN の設定で最初に割り当てた領域とターゲットを関連づけ、続いて IP アドレスを設定する。

```
/iscsi/iqn.20...baa3d89/tpgt1> luns/ create /backstores/iblock/block_backend
Selected LUN 0.
Successfully created LUN 0.
/iscsi/iqn.20...baa3d89/tpgt1> portals/ create 192.168.100.123
Using default IP port 3260
Successfully created network portal 192.168.100.123:3260.
```

次にこの iSCSI ターゲットにアクセスしてくるイニシエータを設定する。ESXi の場合、仮想ホスト→構成→ストレージアダプタ→iSCSI software adapter と辿り iSCSI の項目にイニシエータの値が書かれている。下記の様な番号だが実際はもっと長い。またローカルセグメントだけでは利用しないので接続確認のため一旦 CHAP 認証を無効化しておく。

```
/iscsi/iqn.20...baa3d89/tpgt1> acls/ create iqn.1998-01.com.vmware:54ed98b8-0442e4e5
Successfully created Node ACL for iqn.1998-01.com.vmware:54ed98b8-0442e4e5
Created mapped LUN 0.
/iscsi/iqn.20...baa3d89/tpgt1> set attribute authentication=0
Parameter authentication is now '0'.
```

#### A.2.4.4 設定の保存と設定内容の確認

```
> cd /
/> saveconfig
/> ls
o- / ..... [....]
o- backstores ..... [....]
| o- fileio ..... [0 Storage Object]
| o- iblock ..... [1 Storage Object]
| | o- block_backend ..... [/dev/sdb activated]
| o- pscsi ..... [0 Storage Object]
| o- rd_dr ..... [0 Storage Object]
| o- rd_mcp ..... [0 Storage Object]
o- ib_srpt ..... [0 Targets]
o- iscsi ..... [1 Target]
| o- iqn.2003-01.o...isc2baa3d89 ..... [1 TPG]
| | o- tpgt1 ..... [enabled]
| | o- acls ..... [2 ACLs]
| | o- iqn.1998-01.com.vmware:54ed98b8-0442e4e5 ..... [1 Mapped LUN]
```

```
| | o- mapped_lun0 ..... [lun0 (rw)]
| o- iqn.1998-01.com.vmware:54eed449-22be5782 ..... [1 Mapped LUN]
| | o- mapped_lun0 ..... [lun0 (rw)]
o- luns ..... [1 LUN]
| o- lun0 ..... [iblock/block_backend (/dev/sdb)]
o- portals ..... [1 Portal]
| o- 192.168.100.123:3260 ..... [OK, iser disabled]
o- loopback ..... [0 Targets]
o- qla2xxx ..... [0 Targets]
o- tcm_fc ..... [0 Targets]
```

iSCSI target の設定はこれで完了である。今回使用した環境では追加で設定する項目はなかった。必要であれば/etc/init.d/target status 等を参考に起動時の制御等を行う。

### A.2.5 ESXi から iSCSI サーバを利用する

#### A.2.5.1 VMKernel ネットワークの設定

vSphere client で仮想ホストの”構成” - ”ネットワーク” - ”ネットワークの追加”と辿り、VMKernel を選択し vSwitch0 上に作成する。また vMotion のチェックも入れておく。vMotion を行いたいホストは全てこの vSwitch0 上 (同じセグメント上) に存在するとする。他のホストでも同様の設定を行う。これはあくまでテスト環境での設定で、本番環境では vMotion を行うネットワークは分離しておくべきである。

#### A.2.5.2 iSCSI の認識

仮想ホストで”構成” - ”ストレージアダプタ” - ”追加”を選択し、ソフトウェア iSCSI アダプタを追加する。追加したアダプタの”プロパティ” - ”ネットワーク構成” - ”追加”と辿り、先に設定した VMKernel のネットワークを選択する。”動的抽出”のタブを選択し、iSCSI target の IP アドレスを入力する。他のホストでも同様の手順で iSCSI を認識させる。

#### A.2.5.3 iSCSI データストアの作成

仮想ホストで”構成” - ”ストレージ” - ”ストレージの追加”を選択し、iSCSI を選んでデータストアを作成する。1TB (thin プロビジョニング) を作成した。他の仮想ホストでもこのデータストアが自動的に認識される。このデータストアを利用して、仮想マシンを作成することが可能である。

### A.2.6 書き込みパフォーマンス

今回の検証は Mac mini (256GB SSD\*2 個) 上に ESXi をインストールし、20GB および 1TB の仮想ディスクを割り当てて環境構築を行った。ローカルネットワークは全て 1G の有線で接続されている。他のマシン (HP Proliant DL320e) 上に仮想マシンを作成し、ディスクは今回作成した iSCSI を利用した。この仮想マシン上で dd を用いて書き込みを行ったところ 125MB/s 程度の性能が出た。つまり 1Gbps のネットワーク性能の限界まで利用し書き込みが成功している事が分かる。ちなみに iSCSI サーバに該当する仮想マシンのメモリを 4GB に設定し直すと 90MB/s の書き込み性能であった。