プライベートクラウド構築に挑戦

手島　史綱、長屋　貴量、岩橋　建輔，澤　昌孝、水谷　文保

自然科学研究機構　分子科学研究所　技術課

1. 概要

最近「クラウドコンピューティング」（以下、クラウドとする）というのをよく耳にする。AmazonEC2 などの商用サービスが利用者を伸ばしている。また、最近になって、オープンソースのクラウドミドルウェアが出始め、個人でもクラウドを構築することができるようになってきた。クラウドを利用すれば、物理的リソースを意識することなく高可用性なサーバが素早く稼働出来ると言われている。我々は、これらの利点に興味をもち、調査を始めオープンソースのクラウドミドルウェア（Eucalyptus、openNebula）を使用してプライベートクラウドを構築することに挑戦している。

2. クラウドとは何か

クラウドと言う言葉は、具体的にはインターネットそのものを指す。これは従来から一般的にコンピュータ関連のイメージ図を書く時にネットワークを雲の図で表す場合が多く、それが由来でインターネット自体をクラウドというようになったと言われている。これまでのコンピュータの利用方法は、個人や企業などがコンピュータのハードウェアやソフトウェア、データなどを自前で保有と管理を行ってきた。ところが、そのような保有、管理をすると、ハードウェアやソフトウェアのメンテナンスも自前でやらないといけない。そこで、ハードウェアやソフトウェア、サービスをプロバイダー等に任せ、インターネットから操作し利用しようという流れが生まれてきた。このような「コンピュータ資源（ネットワーク、サーバ、ストレージ、アプリケーション、サービスなど）を、インターネットを介して操作し、必要な分だけ利用する。」をクラウドコンピューティングという。今までもインターネットを利用してコンピュータ資源は利用されていたが、関連技術の進歩によって変化した新しい形のインターネットサービス全般を指すものといえる。

クラウドと言えるものの特徴として次の３つが挙げられる。

- 複数のコンピュータ資源を１つのコンピュータ資源と見なす。
- 必要な時に必要な時だけ使用できる。
- インターネットが利用出来る環境下なら、いつでもどこでも使用できる。

クラウドは、次の３つのモデルに大きく分類される。

表 1　クラウドコンピューティングモデル一覧

モデル名	概　　要
SaaS	(Software as a Service) インターネット経由のソフトウェアパッケージの提供。 従来は ASP（Application Service Provider）と言われていた。 アプリケーション提供ベンダーが用意する専用 URL へアクセスし、アプリケーションをインストールすることなく WEB ブラウザから利用する形態である。 Google の「Google Apps」（Gmail など）や salesforce.com の「Salesforce　CRM」がこれに該当する。
PaaS	(Platform as a Service) インターネット経由のアプリケーション実行・開発用プラットフォームの提供。仮想化されたアプリケーションサーバやデータベースなど、ユーザが自分のアプリケーションを配置して運用で

	きる。 Google の「Google App Engine」や Microsoft の「Windows Azure」がこれに該当する。
IaaS (HaaS)	(Infrastructure as a Service) インターネット経由のハードウェアやインフラの提供。仮想化サーバや共有ディスクなど、ユーザが自分で OS などを含めてシステム導入・構築できる。 従来はホスティングサービスや VPS（Virtual Private Server）と言われていた。 Amazon の「Amazon Web Service」がこれに該当する。

提供されるサービスの形態には、次４つに分類される。

表 2　クラウドサービスの形態

クラウドの種類	概　　要
パブリッククラウド	Amazon Web Services に代表されるインターネット経由の一般利用者を対象に提供される（一般向けサービス）。リードタイム、従量課金性、膨大な資源の貸し出しなどがあり、すぐに使え、必要な時に必要なだけ使い、不要になったら止めるというメリットがある。
コミュニティクラウド	同様のミッションを持つ特定企業群によって形成する「コミュニティ」で共同運用されるデータセンターの共同利用などの形態。パブリッククラウドのようなセキュリティに対する懸念を解消しつつ、プライベートクラウドのようなある程度の柔軟性とコスト削減効果が期待できる。
プライベートクラウド	企業が自社内でクラウドを構築し、企業内の部門やグループ会社などに対してサービスを提供する（ファイヤーウォール内向けサービス）。 一般に利用率が高く、長期間使用するのであれば自前で保持し運用したほうが長期的な TCO (Total Cost of Ownership)は安くなることがある。データ保管等のセキュリティ面で安全かもしれない。
ハイブリッドクラウド	パブリッククラウドとプライベートクラウドを組み合わせたもの

クラウドの実情を見ると、AmazonEC2 や Google Apps（Gmail やカレンダー、グループなど）は多くの方が利用している。個人宅でも光インターネット回線が普及し、PC の性能も格段にあがるなどインターネットが利用しやすくなった背景や高機能高価格な小売店アプリケーションよりも、必要な機能があり低価格（無料のものも）で必要な時にすぐ利用出来る SaaS 型クラウドが人気である。また、企業などにおいては、社内ツールの統一やデータ共有などのために PaaS 型クラウドを利用する企業が増えている。また、従来からのホスティングサービスは、IaaS 型クラウドになり、素早くユーザのニーズ（メモリを今増やしたい。HDD を今増やしたい）に答えられるようになってきている。これらのサービスは全てパブリッククラウドやコミュニティクラウドの形態として提供されている。なお、TCO やセキュリティ面を気にする場合やコンピュータ資源のより柔軟・迅速・効率的な割当や共有をしたい場合、技術研鑽の場合などは、企業内や組織内で閉じてその利点を享受するププライベートクラウドが利用される。

クラウドは、複数のコンピュータをプライベートネットワークで接続し、仮想的に１つのコンピュータにする。この時のコンピュータをクラスタノード（以下、ノードとする）という。当然ながら、これらのノードを管理するコンピュータが必要になり、それをここではフロントエンドノード（以下、フロントエンドとする）と呼ぶ。

3. オープンソースのクラウドミドルウェア

最近になって、オープンソースの IaaS 型クラウドミドルウェアの開発が活発である。調べた範囲で一覧にまとめてみた。

表 3　オープンソースのクラウドミドルウェアー覧

クラウド名	概　　要
Eucalyptus	Elastic Utility Computing Architecture for Linking YourPrograms To Useful Systems カルフォルニア大学サンタバーバラ校が開発。現在は、その開発者達によって創設した Eucalyptus Systems が開発や保守を行っている。 IaaS 機能の提供。AmazonEC2/S3 と機能・API 互換性を提供 仮想化は Xen、KVM に対応。 対応 OS：Linux ライセンス：GPL v3 http://www.eucalyptus.com/ 日本ユーザグループ http://eucalyptus-users.jp/
openNubula	マドリード・コンプルテンセ大学（Universidad Complutense de Madrid）の Distributed Systems Architecture Research Group が開発(2002～)。EU 政府の支援あり。 IaaS 機能の提供 仮想化は Xen、KVM と VMWare に対応。AmazonEC2 や英 ElasticHosts などとのパブリッククラウドとの連携も可能。 対応 OS：Linux、Mac OS　X ライセンス：Apache License 2 http://www.opennebula.org/
Nimbus	シカゴ大学（University of Chicago）とアルゴンヌ研究所（Argonne National Laboratory）が中心となって開発。 IaaS 機能の提供 仮想化は Xen に対応。近々KVM にも対応予定。 Grid 系から出発した取り組み。Nimbus 自体はかなり長い歴史あり。 対応 OS：Linux ライセンス：Apache License 2 http://www.nimbusproject.org/
NASA Nebula	Eucalyptus を使用していた NASA だが、業務で要求される膨大なデータ量、処理ノード数を支えるスケーラビリティと開発プロセスのオープン性の問題から独自に開発。 openNebula とは全く別物。 NASA エイムズ研究所のクラウド http://nebula.nasa.gov/
openStack	Rackspace Hosteing（クラウド事業者）と NASA が中心になって開発（初盤は 2010 年 10 月 21 日にリリースされる）。 その他、Dell、Citrix、NTT データ、Intel、AMD などが参加表明。 「OpenStack Compute」と「OpenStack Object Storage」の２つのプロジェクト ライセンス：Apache License 2.0 「OpenStack Compute」は、NASA が開発し利用している Nebula をベースにしている。大規模インスタンスの展開を可能にする IaaS 機能を備えており、インスタンス間の連携によるスケー

	ラビリティにも対応。 http://www.openstack.org/ 日本ユーザ会（http://openstack.jp/）
CloudStack	CloudCom 社製品の高可用性機能などを除いた Community Edition。 http://www.cloud.com/community
Wakame-vdc	あくしゅ社（http://axsh.jp/）が開発中（国産） http://wakame.axsh.jp/vdc.html/

4. 動機

我々が配属されている計算科学研究センター（以下、センターとする）は、大学共同利用施設として、全国の分子科学研究者にコンピュータ利用サービスを行っている。そのため、センターには、WEB サーバを始め、メールサーバや計算サーバなどがあり、これらの管理運用を技術職員で行っている。その中で、センターで運用しているメールサーバの OS（RedHat）サポートがもうすぐ切れるということが分かり、マシンも古いことから OS とマシンの更新をすることに決まった。

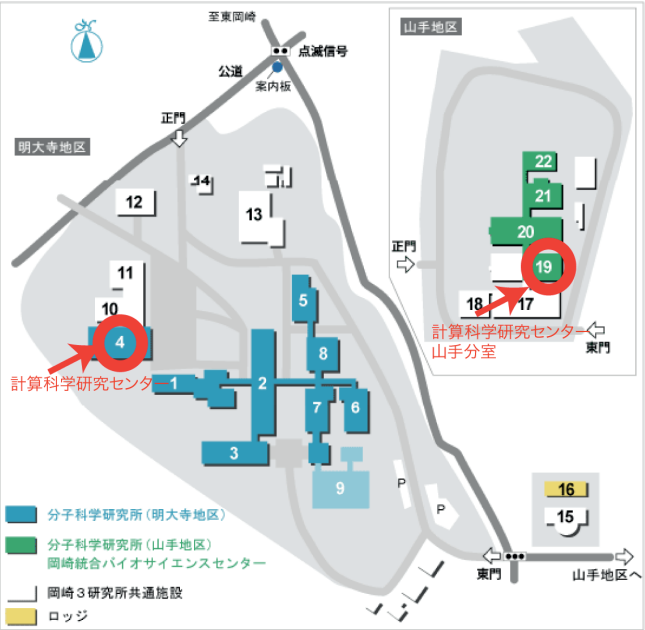

図 1　キャンパス図

今までのメールサーバは１台で運用しており、障害が発生すると業務が停滞していた。また、分子科学研究所は図１のように明大寺地区と山手地区の２つのキャンパス（徒歩１５分程度の距離）で構成されており、計算科学研究センターは、明大寺地区と山手地区に主機室がある。現メールサーバは明大寺地区に設置されている。それぞれの地区では、年に１回電気設備点検の為の計画停電があるため、メールサーバはこの停電の時は停止させている。しかし、明大寺地区が停電の時は、山手地区は停電（通電されている）ではないので、メールサーバを停電ではない地区に持っていけば、停止することなく運用が続けられる。しかし、サーバ自身を持ち運ぶのは非効率的であり故障の原因にもなりかねない。そこで、高可用性なサーバを構築する必要がある。

仮想サーバの Live Migration 機能を使用すれば、高可用性が実現できるはずであるが、最近になって、オープンソースのクラウドミドルェアが公開されているのを知り、それらを使用して、将来的にはその他のサーバもプラベートクラウド環境下におき柔軟で拡張性の高い運用ができると判断し情報収集を始めた。当初、クラウドと仮想サーバの違いは正直分からなかった。その違いは、仮想サーバは１台毎に閉じた中でゲスト OS を構築していくことになるので、当然、メモリや HDD が足りなくなって来た場合は、仮想サーバ自身を停止させて（ゲスト OS もサービス停止となる）拡張作業などを行うことになる。しかし、クラウドでは、複数台のノードを１台の仮想サーバと見立てるので、複数台のメモリ、HDD が１つとなり、複数のゲスト OS に使われる。ここで大きく違ってくるのは、仮想サーバが運用中でもノードを追加することが出来き、ゲスト OS への追加も設定ファイルを書き換えてゲスト OS をリブートするだけで、運用停止時間はかなり短縮できる。このことから仮想サーバより可用性が高いものである。

雑誌や、WEB などで取り上げられており ドキュメントが比較的多かったので Eucalyptus を選択して構築を始めた。

5. 構想

構想をまとめておく。明大寺地区、山手地区の計算科学研究センター主機室にオープンソースクラウド（IaaS）のノード群をそれぞれ設置しそれぞれの地区でグループ化する。明大寺地区、山手地区にそれぞれフロントエンドを置き、1 台を主、もう 1 台を副とする。これは、ノードを管理するフロントエンドは当然ながらクラウド環境外にあり、単なるサーバであるので、バックアップやホット・スタンバイが必要である。

明大寺地区で計画停電が行われる場合、明大寺地区で稼働していたサーバ（図 2 の場合ではメールサーバと WEB サーバ）を停止させることなく山手地区に移動させる（Live Migration 機能）。そして、明大寺地区フロントエンドを停止させて山手地区フロントエンドを主にする。これにより、高可用性なサーバが維持出来る。現時点ではかなり理想の高い構想となっている。

我々が目指しているクラウドは、IaaS 型のプライベートクラウドである。

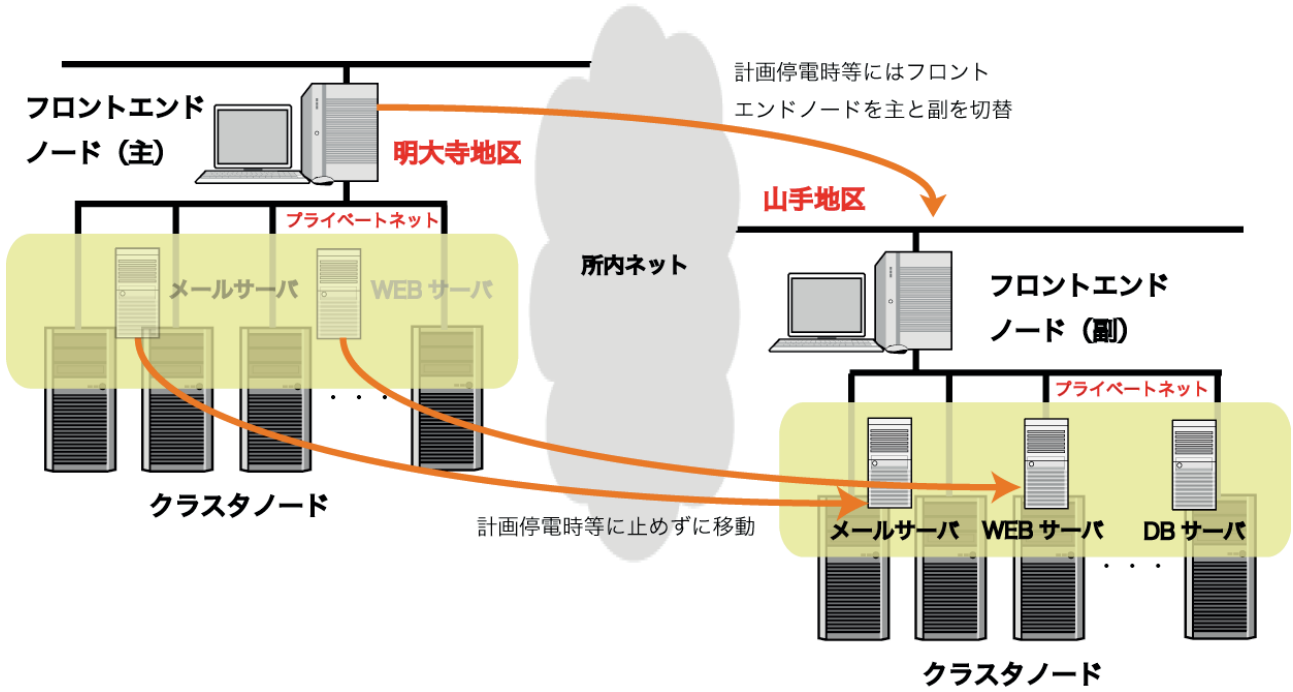

図 2　クラウド構想図

6. 構築

●Eucalyptus での構築方法

PC 4 台（スペックについては表 4 を参照）を使用して構築した。4 台のマシンスペックは全く同じである。ただし、フロントエンドだけは NIC を 1 枚追加している。その他は Onbord の NIC を使用している。4 台の PC 接続は図 3 を見てもらうと分かるが、フロントエンドは、グローバルネット（グルーバル IP アドレスを割当）とノードが接続されているプラベートネット（プライベート IP アドレスを割当）に接続されている。各々ノードは当然ながらプライベート IP アドレスのみが割り当てられている。

表 4　マシンスペック

CPU	Intel Core i3 540 3.06GHz（4MB L3 Cahche）
Memory	4GB（2GB×2）　Dual chanel DDR3-1333SDRAM
HDD	320GB SATA（7,200rpm）

4 台の PC には、CentOS5.5（Xen ベース）をインストールし、Eucalyptus1.6.2 を使用している。実は、構築当初は、Ubuntu Server 9.0 と Ubuntu Enterprise Cloud(Eucalyptus ベース)を使用して構築を始めたが、作業が難航した。そんな時に、「クラウド Watch」（http://cloud.watch.impress.co.jp/docs/column/eucalyptus/index2010.html）という WEB ページで CentOS+Eucalyptus の作業手順が分かりやすくあったので、途中からこちらに切り替えたという経

緯がある。作業手順の詳細はこちらの WEB ページを参照して頂きたい。

インスタンスというのは、ノード上に置かれる GuestOS である。このインスタンスに割り当てる IP アドレスはグローバル IP アドレスである。これは、Eucalyptus の設定ファイルにて割当範囲を設定した（DHCP、固定 IP アドレス設定も可能）

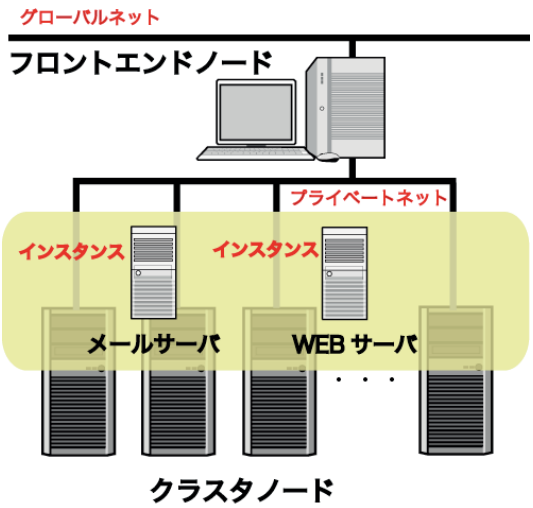

図 3　Eucalyptus による構築図

構築当初、ノードのネットワーク接続がブリッジ接続となっており、フロントエンドと通信が出来ない状態になっていた。static になるように各々ノードの/etc/modules と/etc/network/interfaces を編集して、NIC を eth0 に設定し、フロントエンドとの通信が出来るようになった。インスタンスの設定まで作業が進んだところでインスタンスを起動してみたが、起動出来なかった。原因は、このネットワーク設定をブリッジから static に変更したためだった。ノードのネットワーク設定を当初のブリッジ設定に戻したところ問題なくインスタンスを起動することができた。これは、ノードが既に Xen による仮想ネットワーク上に置かれているのに、それを無視してブリッジ接続を止めたため、Xen 上の Eucalyptus がノードを認識出来ずインスタンスを起動出来なかったものである。仮想化の知識が足りなかった故の失敗である。

起動出来たインスタンスは、停止するとインスタンス上の情報は失われてしまう。よって、インスタンス全体をマシンイメージ化して保存する必要がある。これらの作業もなんとか出来てマシンを停止しても、同じインスタンスを起動出来ることを確認できた。しかし、作業と並行して情報収集していたところ、残念なことに Eucalyptus は Live Migration に未対応ということが分かった。そこで、他にオープンソースなクラウドミドルウェアはないのか再度調査すると、当初はあまり見つけられなかった情報が目につくようになり、いろいろと情報が集まった。その中で openNebula が Live Migration に対応している情報があったので、早々に Eucalyptus による構築作業を止め、openNebula を使用したシステム構築を始めた。

●OpenNebula での構築方法

Ubuntu Server 10.04 をインストールして、Ubuntu の WEB ページ（https://help.ubuntu.com/community/OpenNebula）を参考にしながら作業を進めた（仮想化は KVM を使用）。WEB ページには「Setting up a Private Cloud in 5steps」と書いてあるので、簡単に作業が完了するはずであるが、全く 5steps では構築できていない。先に使用した Eucalyptus では、ノードはフロントエンドから IP アドレスが割り振られていたので、openNebula も同じだと考えていたが、ノードに IP アドレスが割り振られない。設定をどこで行うのか WEB ページの説明を読んでも分からなかった。結局、フロントエンドからは IP アドレスは割り振られず、悩んで末に技術班のミィーティングで相談した結果、「openNebula」はプライベートネットワーク上に DHCP サーバなどが既にある、ネットワークとして整った環境が前提なのでは」という意見をいただいたので、早速、ノード 3 台の内 1 台を DHCP サーバとして設定したところ、フロントエンドからもノードが認識されるようになった。

現報告書作成時点では、フロントエンドから各々ノードへの ssh 接続が出来き、フロントエンドからノードの status 状態を確認出来るコマンドを実行するとノードの状態（STAT）は on になっているが、その他の情報は表示されない状態のところで作業は止まっている。おそらく何らかの設定が不足しているものと考えている。これがうまく認識できれば、この後の作業として

・インスタンスを設定、稼働
・ゲスト OS イメージを登録して稼働

を行うことになる。詳しい情報は openNebula の HP にあるようだが、私の英語力が低い為に作業は難航している。あと、KVM の知識も必要だと感じている。openNebula は知識が薄い人には難しいシステムかもしれない。

以下、に自分が行ってきたインストール手順を記す。

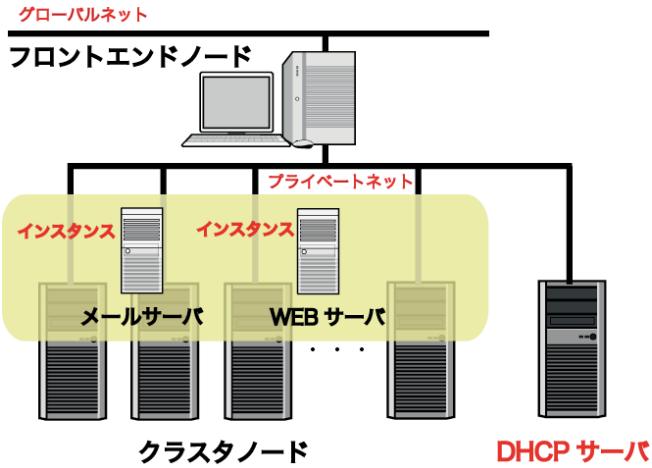

図 4　OpenNebula による構築図

DHCP サーバの設定方法は、Ubuntu の一般的方法で設定している。

% sudo apt-get install dhcp3-server

% sudo vi /etc/dhcp3/dhcpd.conf（表 4 参照）

% sudo service dhcp3-server start

% sudo sysc-rc-conf dhcp3-server on

ノードにおいて、ifconfig コマンドを実行して、IP アドレスが割り当てられているか確認する。仮想化で KVM を使用するので、マシンにハードウェア的仮想化支援機構（vt-d）が必要である。それがあるか確認をする。

■Intel の場合

% egrep vmx /proc/cpuinfo

表 5　/etc/dhcp3/dhcpd.conf の内容

```
ddns-update-style none;
subnet 192.168.0.0 netmask 255.255.255.0{
        option routers                  192.168.0.1;
        option subnet-mask              255.255.255.0;

        option domain-name              cloud;
        option domain-name-servers      192.168.0.1;

        range dynamic-bootp             192.168.0.101 192.168.0.4200;
        default-lease-time              21600;
        max-lease-time                  43200;
}
```

■AMD の場合

% egrep svm /proc/cpuinfo

■すべてのパッケージを最新にする（フロントエンド、各々ノード）

% sudo apt-get update

% sudo apt-get upgrade

■KVM パケージのインストール

% sudo apt-get install kvm libvirt-bin

■apparmor パッケージのアンインストール

% sudo apt-get remove apparmor

■PPA 登録

/etc/apt/sources.list に下記 PPA を登録

deb http://ppa.launchpad.net/opennebula-ubuntu/ppa/ubuntu intrepid main

■前提動作環境（次のものをインストールしておく）

ruby >=1.8.6 and < 1.9.0

sqlite3 >= 3.5.2

sqlite3-dev >= 3.5.6-3 (aptitude)

sqlite3-ruby (aptitude)

xmlrpc-c >= 1.06 (aptitude)

openssl >= 0.9

libxmlrpc-c >=1.06 (aptitude)

scons >= 0.97

g++ >= 4

ssh

■フロントエンドに openNebula パッケージインストール

% sudo apt-get install opennebula

■フロントエンドに ssh でログインして見る。

% ssh –u oneadmin rontend

■システムにクラスタノードを追加

% onehost add cluster02 im_kvm vmm_kvm tm_ssh

■ssh アクセス設定

% sudo –u oneadmin ssh cluster02

■ノードに openNebula パッケージインストール

% sudo apt-get install opennebula-node

■ネットワーク設定

$ sudo vi /etc/network/interface

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual

auto br0
iface br0 inet dhcp
    bridge_ports eth0
    bridge_fd 9
    bridge_hello 2
    bridge_maxage 12
    bridge_stp off
```

% sudo /etc/init.d/networking restart

現在のところ完了しているのはここまでである。構想に対して、フロントエンドとノードのプライベートネットが出来、フロントエンドからノードが認識出来ている程度である。いろいろと試し整理しながら作業を地道に少しずつ進めている。

7. まとめ

オープンソースのクラウドミドルウェアは、まだまだ発展途上であると感じるが、今後の計算機システム構築の中では確実に重要なものである。クラウドを作っていく上で、UNIX サーバの知識だけではなく、仮想化やネットワークの知識をも熟知していないと構築できないことも分かった。まだまだ、構築が始まったばかりであるが、切磋琢磨して完成を目指したい。また、Live Migration の実現とその操作性も今後検証を行っていく予定である。