

エンジニアの皆さん、 パブリッククラウドだけで満足ですか？

～ハイブリッドクラウドで実現する新しいインフラのかたち～

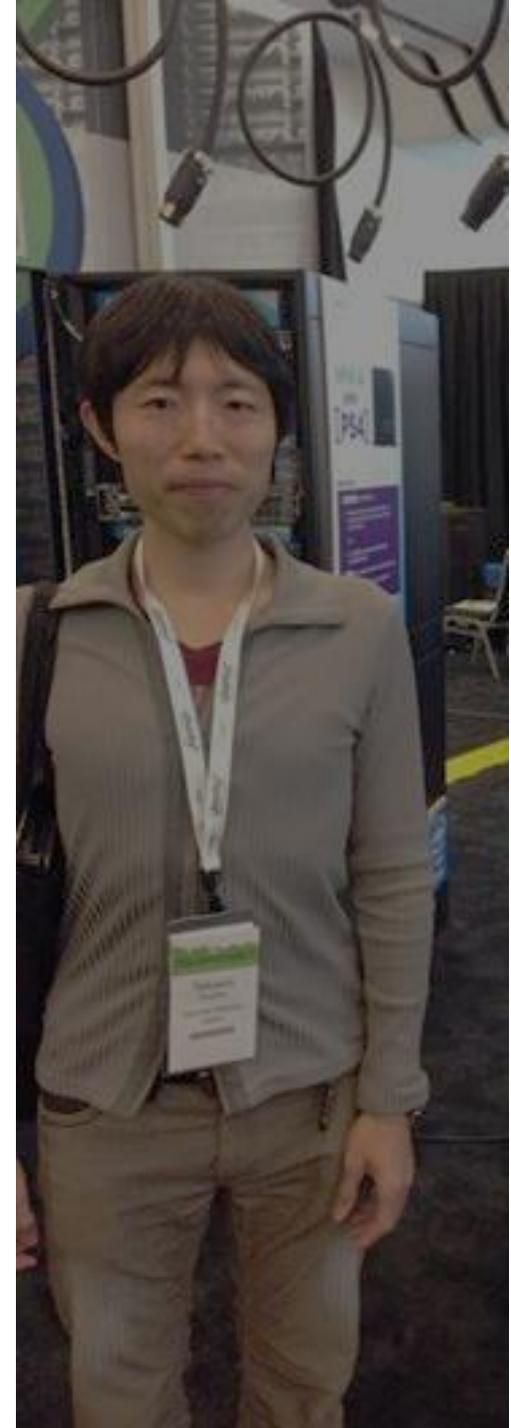
Nov 28, 2014

Internet Initiative Japan Inc.

Takashi Sogabe

Who am I?

- Takashi Sogabe
 - @rev4t
 - Software Engineer, Internet Initiative Japan Inc.
 - 最近の注力分野
 - OpenStack
 - 関わっているプロジェクト
 - OpenContrail, mruby, Riak, etc.



クラウドの先にあるもの

- サーバ資産を変動費用にできる?
- コストが下がる?
- Agilityが上がる?

ソフトウェア化が進んだ世界(1)

- 20 years from now, labor demand for lots of skill sets will be substantially lower.
 - Bill Gates
 - <http://bitly.com/1tKH7Q2>

ソフトウェア化が進んだ世界(2)

- Rapid improvements in artificial intelligence, for instance, will make computers and robots adept at most jobs. Given the chance to give up work, nine out of 10 people “wouldn’t want to be doing what they’re doing today”
 - Larry Page
 - <http://on.ft.com/1ouoOzA>

クラウドの先にあるもの(2)

- コンピュータがジャブジャブに使える世界
 - AI, BigData, IOT, etc.
- クラウドのユーザ
 - 世界中の誰でも、何処からでもコンピュータが使える
- クラウド提供者
 - 大量のサーバを全世界的に自動制御にて管理

開発スタイルの変化に対応したい

- Agile IT
 - 分散システムが大前提
 - クラウド必須
 - スケールアップ -> スケールアウト
 - これからのアプリ開発に必須
 - IaaSがあれば実現できるが..

クラウドだけあれば十分?

- Traditional IT: Do it Right

- 従来のエンタープライズ型IT運用・管理モデル
- 効率、セキュリティ、コンプライアンス、信頼性を重視

- Agile IT: Do it Fast

- 新しい時代の戦略性の高いITモデル
- クラウド、モバイル、アナリティクス、ソーシャルなどの動きに沿う

プライベートクラウドも欲しいのです

- 開発環境は堅牢なサービスでなくて良い
 - 開発環境向けにコスト最適化
 - 退役サーバ、余りサーバ、etc. の有効利用
- 社外に出したくないデータ
- IaaSをプライベート環境でも使いたい
 - 伸縮自在, Self Manage, APIによるプログラム制御

Agile ITと Traditional ITの棲み分け

- インフラ
 - ハイブリッドクラウドを駆使してクラウドの恩恵を受けつつ Agile ITのインフラと Traditional IT のインフラを共有
- 開発・運用
 - 開発スタイルが全く違うので、無理に一本化はせず別々のチームで開発・運用する

コスト面では圧倒的に不利なHybrid Cloud

- IaaSに最適化されたコモディティサーバ
- 人力作業を最大限に排除できるCloud OS
- 余分な電力を消費しないファシリティ

サーバ、Cloud OS、ファシリティのコスト構造を
可能な限り Public Cloud に近づけることが
Hybrid Cloud 導入には必要

他業種での取り組み事例

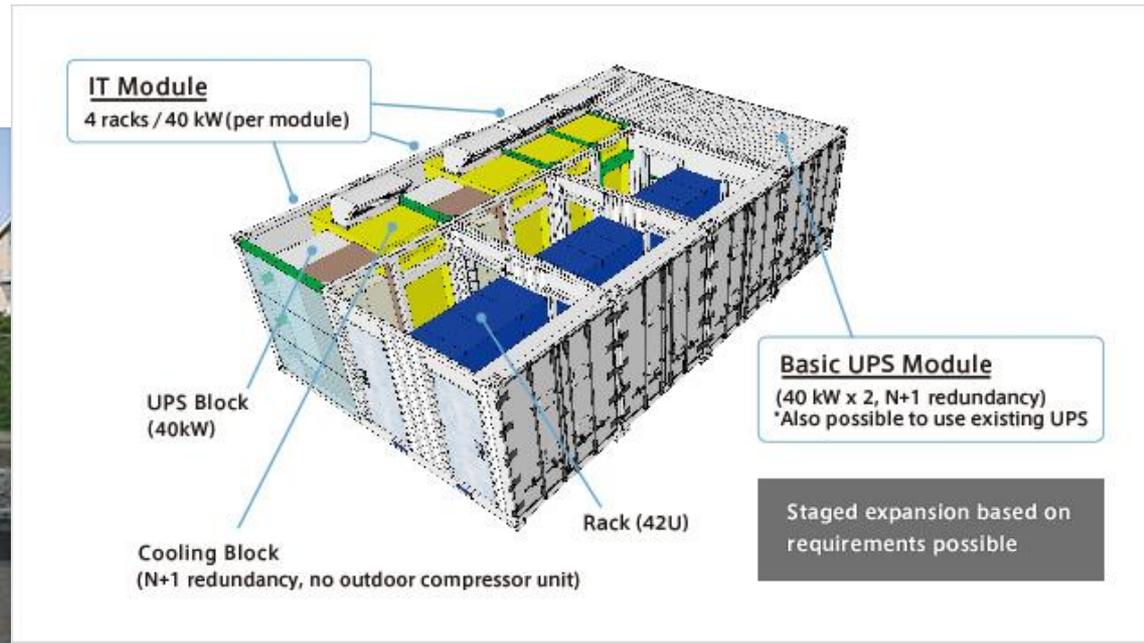
- コスモスベリーズ
 - 町の電気屋さんでも、量販店と同じ価格で電気製品を販売できる
 - 仕入れなどの流通システムはヤマダ電機のもの共有
 - 地域密着、Face to Face 等の強みを活かして従来の量販店がカバーできなかったユーザを開拓

Reference Design: Hybrid Cloud

- ファシリティ
 - co-lzmo/l
- ラック、サーバ、スイッチ
 - Open Compute Project (OCP)
- Cloud OS
 - OpenStack

Ready-Made Data Centers

- co-IZmo/I(コイズモアイ)
 - モジュール型のコンテナDC
 - <http://www.iiij.ad.jp/DC/en/products/index.html>



高い電力消費効率、伸縮自在なDC

- 高い電力消費効率
 - PUE 1.17
- 伸縮自在
 - 1コンテナ当たり4rack
 - 連結することで拡張ができる

ラック、サーバ、スイッチ

- Open Compute Project
 - コスト効率の高いコンピューティングインフラを設計するためのプロジェクト
 - Facebook社が取り組みを始め、現在では多数の会社が参画
 - オープンソースの開発モデルをハードウェア設計に適用



OPEN
Compute Project

Cloud OS

- OpenStack
 - IaaSを誰でも無償で構築できるソフトウェア
 - Compute(Nova), Block Storage(Cinder), Object Storage(Swift)等、多岐にわたるコンポーネントが提供されている



システム構成

- コンポーネントの差し替えが可能
 - 自由にコンポーネントを選ぶこともできる

データセンタ(都市型DC, colzmo/Iなど)

Cloud OS

DCIM

Federation

Compute
Servers

Storage
Servers

Switches

Connectivity

PoC環境

- OpenStack Controller, Compute nodes
 - Winterfell, etc.
- Object Storage
 - Swift, Swift/Kinetic(テスト中)
- Networking
 - Cumulus, ARISTA

OCP Rack

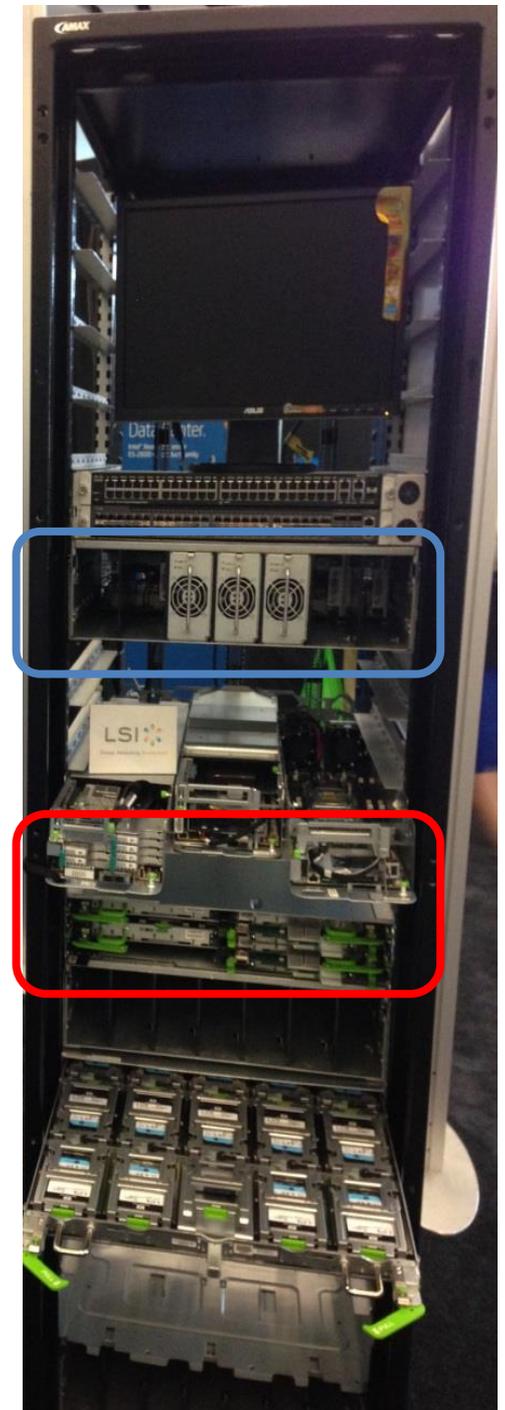
- 直流給電
 - 各ラックにn+1電源モジュールを搭載
 - 電源レールを用いてサーバにDC給電
- 前面吸気、後面排気
 - ホットアイル、コールドアイル型のデータセンタに最適
- 21inch幅
 - サーバの横幅を大きくできる
 - Open Rack自体は、横幅がスリム



電源バー

電源モジュール

WinterFell



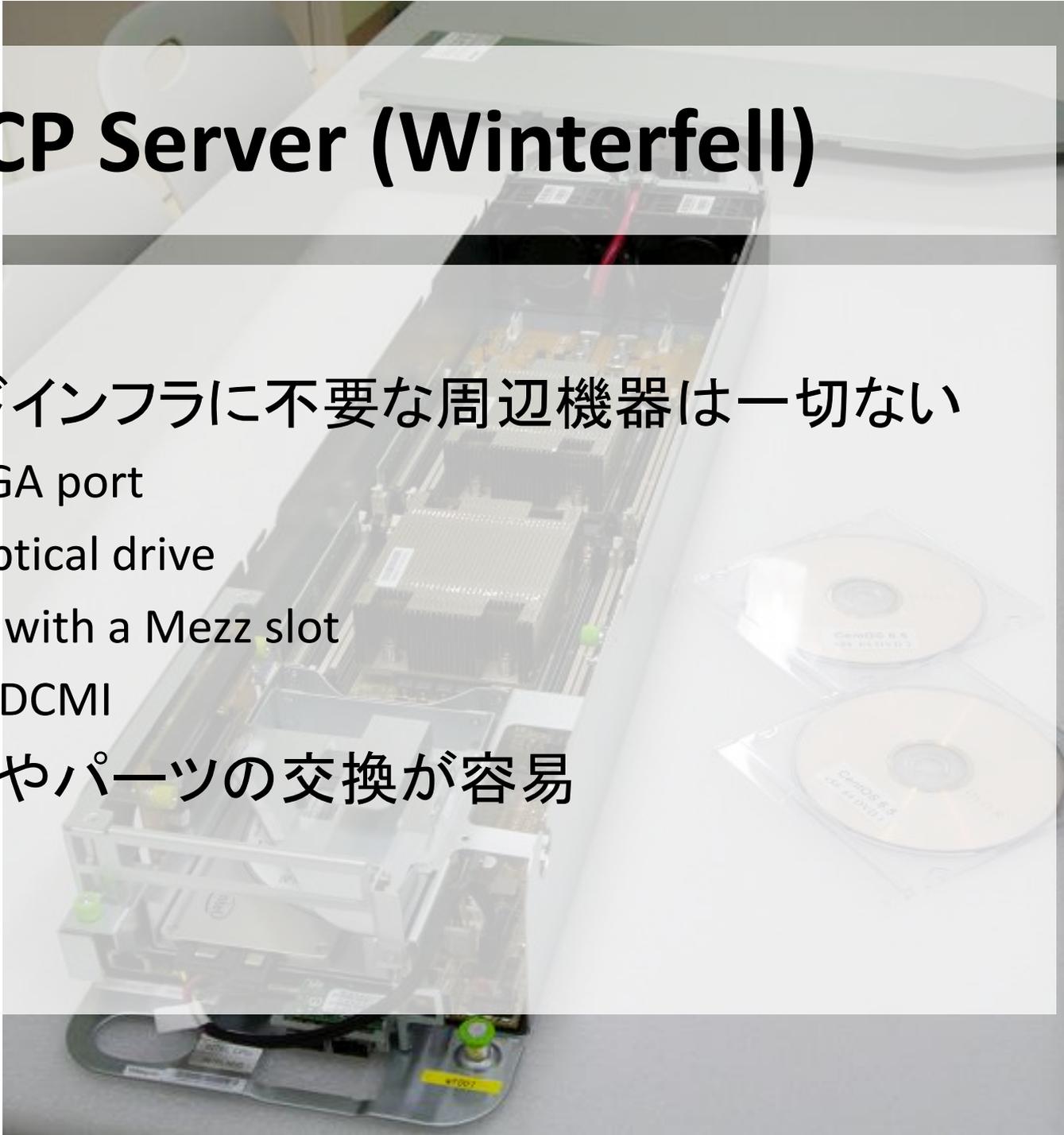
OCP Server (Winterfell)

- Simple

- クラウドインフラに不要な周辺機器は一切ない

- No VGA port
- No Optical drive
- 1 NIC with a Mezz slot
- IPMI/DCMI

- サーバやパーツの交換が容易



OCP JBOD (Knox)

- 2Uのシャーシに 30台のHDDを搭載
- RAIDコントローラ内蔵、SAS Interface で Winterfellに接続できる



シンプルなことの利点 (1)

- 起動時間がとても早い
 - Winterfell
 - 約30sec
 - 汎用サーバ
 - Over 120sec

シンプルなことの利点 (2)

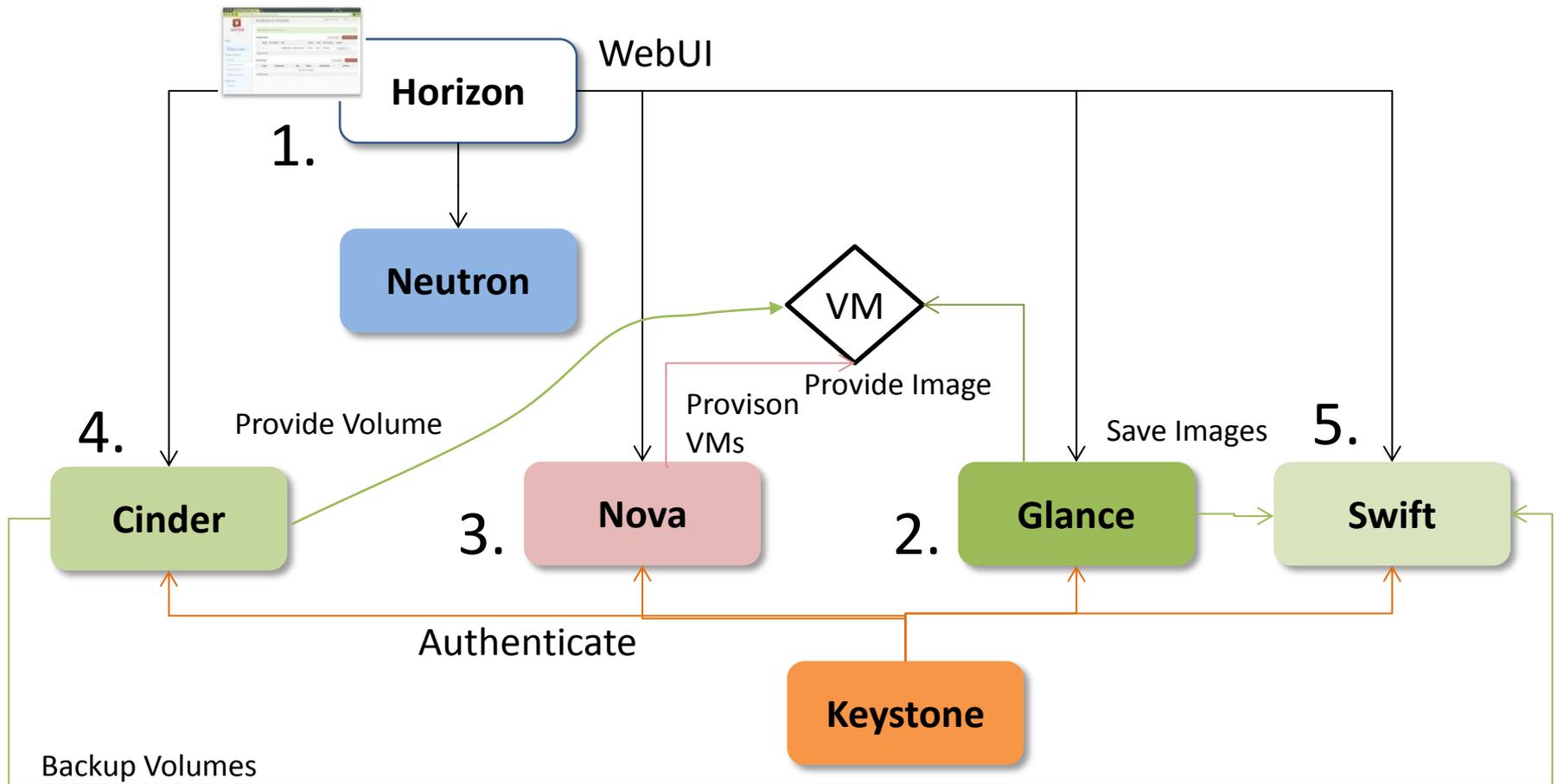
- OpenStack Integration
 - IaaSに不要な周辺機器を買う必要がない
 - コモディティサーバを束ねることで、計算能力やストレージ容量を自在に増減できる

OpenStack

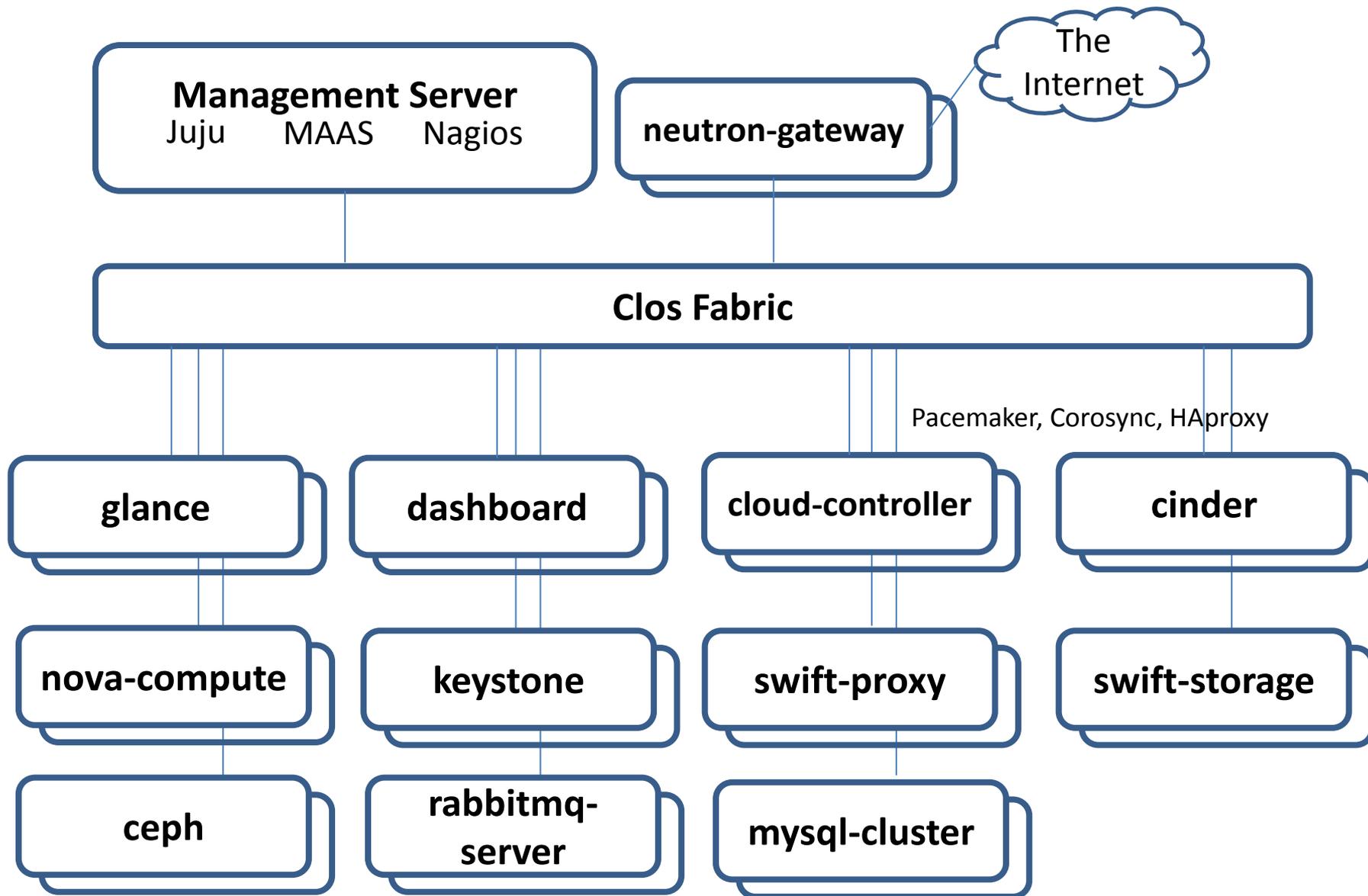
- 現在の課題
 - プロダクション環境向けのインストールは大変
 - 商用ディストリビューションを使うことで緩和できる
 - Metacloud, Mirantis, Piston, RedHat, etc.
 - Open Source実装を使う場合
 - TripleO (OpenStack on OpenStack)
 - Juju/MAAS



OpenStack 主要コンポーネント



PoC環境のコンポーネント



Bare Metal Provisioning

- MAAS
 - Metal as a Service
 - ver.1.7では他社Distro(CentOS等)にも対応
- Cobbler
 - 昔から広く使われているOS Deployment Tool
 - 商用OpenStackのインストーラとして良く使われている



Juju

- Deployment tool
 - Chef, Puppet, Ansible 等に似たツール
 - MAASと組み合わせて使える
 - Charm例
 - OpenStack
 - Hadoop
 - MySQL



Networking (1)

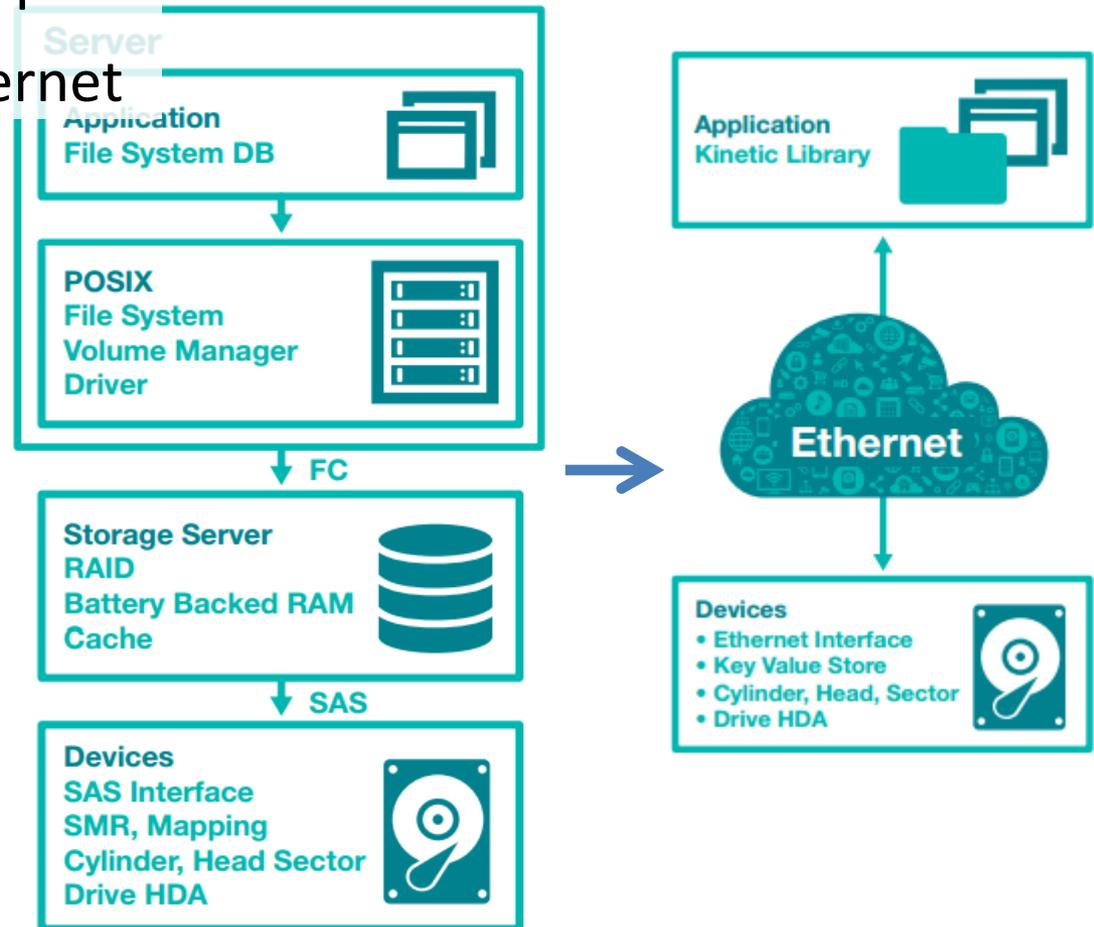
- Flat Network
 - Simple
 - スケールアウトさせるのが難しい
- CLOS Topology
 - Scalable
 - ECMPでロードバランスできる
 - Overlay networkが必要
 - GRE, VXLAN, etc.

Networking (2)

- 一元管理
 - DevOpsに向けた管理方法の確立
 - Chef/Puppet/Ansible, CI
 - Network機器、実は既にLinuxベースが主流
 - であれば、User Interface も Linuxで良いはず!
 - 余計なUser Interfaceがあるとソフトウェアによる一元管理の妨げになる

Storage(1)

- Seagate Kinetic
 - ServerからHDDを分離
 - Key-Value Store API
 - Protobuf with Ethernet
 - 利点
 - 柔軟性
 - スケールアウト性
 - 効率



Storage (2)

- Swift with Kinetic
 - <https://github.com/swiftstack/kinetic-swift>
 - OpenStack(Swift)との組み合わせも可能

```
$ swift-ring-builder kinetic.builer
kinetic.builder, build version 31
1024 partitions, 3.000000 replicas, 1 regions, 1 zones, 4 devices, 0.00 balance
The minimum number of hours before a partition can be reassigned is 1
Devices:  id region zone  ip address  port  replication ip  replication port  name weight partitions balance meta
          7     1     1    127.0.0.1  6010    127.0.0.1          6010 10.174.251.101:8123  1.00      768  0.00
          8     1     1    127.0.0.1  6020    127.0.0.1          6020 10.174.251.102:8123  1.00      768  0.00
          9     1     1    127.0.0.1  6030    127.0.0.1          6030 10.174.251.103:8123  1.00      768  0.00
         10     1     1    127.0.0.1  6040    127.0.0.1          6040 10.174.251.104:8123  1.00      768  0.00

$ swift upload mycontainer test
test
$ swift download mycontainer test -o -
awesome
$ for i in {1..4}; do kcmd -H 10.174.251.10${i} list objects; done
objects.202f47d75a808c212d43c6dda051f39c.1407379571.83189.data.89a32569-fbbf-4ad3-8e8a-f46e1d632cbb
objects.202f47d75a808c212d43c6dda051f39c.1407379571.83189.data.dd7a79e0-3c92-43fb-937a-10fd11a28c32
objects.202f47d75a808c212d43c6dda051f39c.1407379571.83189.data.df9f0a7f-d2da-4d52-aa06-55e96a1fdcd7
```

Container対応への取り組み

- Mesosphere, Kubernetes など、分散コンピューティングに最適なコンテナ環境の開発が急速に進んでいる
- IJは、OCP Japan の PoC Working Group にて OCP Server を用いたOpenStack のコンテナ環境について検証中
 - <http://opencomputejapan.org/community/working-group>

課題のまとめ

- Agile IT と Traditional IT、全く別の開発・運用チームが必要だが インフラについてはHybrid Cloud を用いて共用するのが望ましい
- Hybrid Cloudを導入するためには、サーバ、ファシリティ、Cloud OS の3つを Public Cloudのように最適化する必要がある

取り組みのまとめ

- Hybrid Cloudのリファレンスデザインを作成し、PoC環境を構築した
- ContainerやKineticなど、新しい技術への取り組みを進めている