

openstack

Open source software to build public and private clouds.

リソース（仮想マシン） 割り当ての裏側 ～ nova-scheduler(1) ～

2012.08.04

日本オープンスタックユーザ会
Hideki Saito / @saito_hideki



Agenda



- 自己紹介
- はじめに
- 仮想マシンを起動する
- リソースのスケジューリング(1)
- まとめ

- 氏名: 齊藤 秀喜 (さいとう ひでき)
- TwitterID: @saito_hideki
- 所属: 日本オープンスタックユーザ会 / 某ISP

- 仕事:
 - クラウド基盤の開発・運用
 - 次世代クラウド基盤の調査・実証実験

- 趣味: OpenStack (嗜む程度)

はじめに (今回ご紹介する内容)



このセッションは、OpenStack Computeがhypervisorの上で仮想マシンを起動するまでの流れと、それに関わる重要な要素である、

nova-scheduler

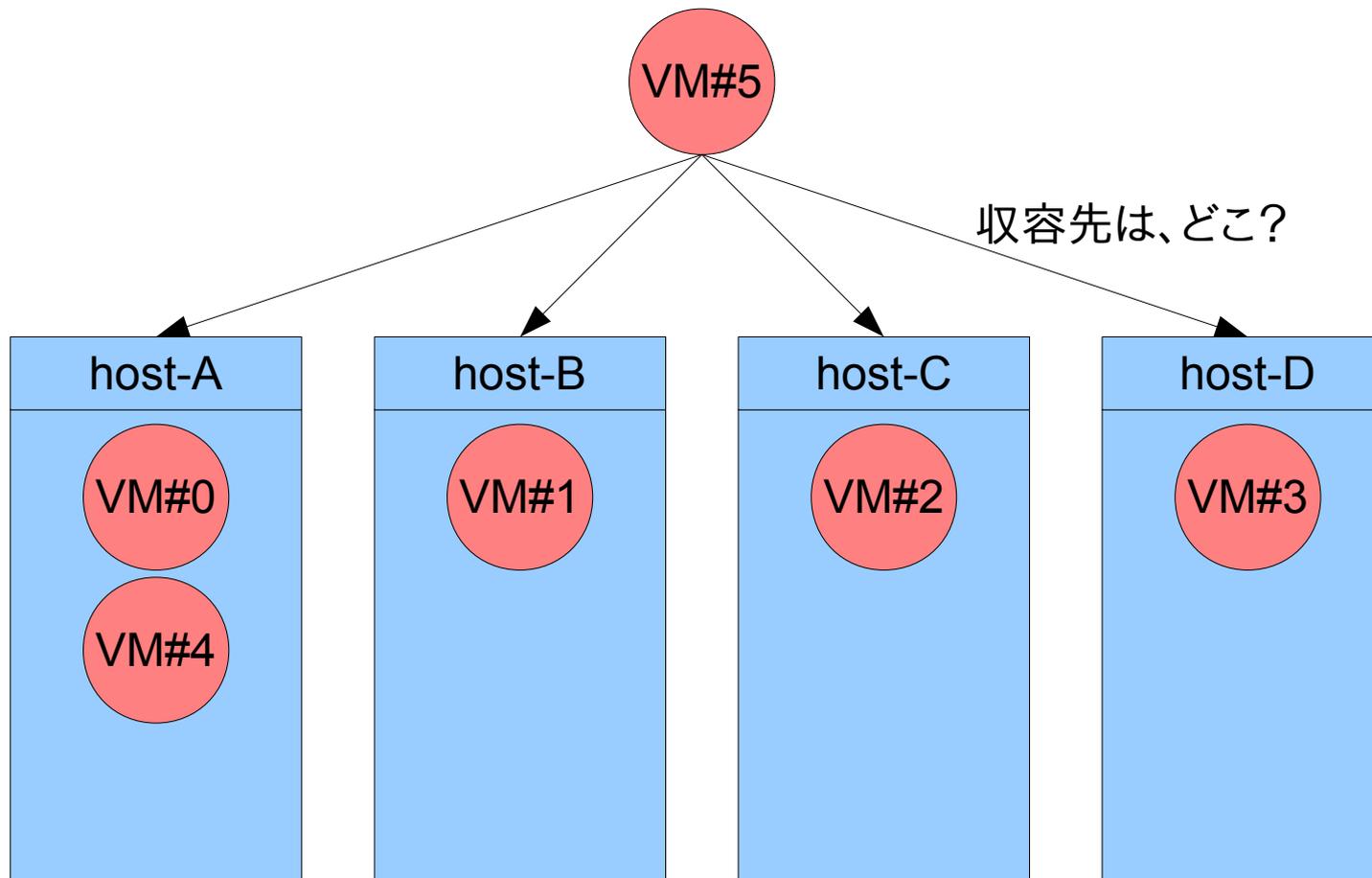
の仕組み(概要)についてご紹介する20分15分のセッションです。

※nova-schedulerに関する公式な資料は...
ほとんどないのが特徴です:)



はじめに (動機)

この仮想マシン、いったいどのcomputeで動くんだらう? ...どこかに情報ないかな...

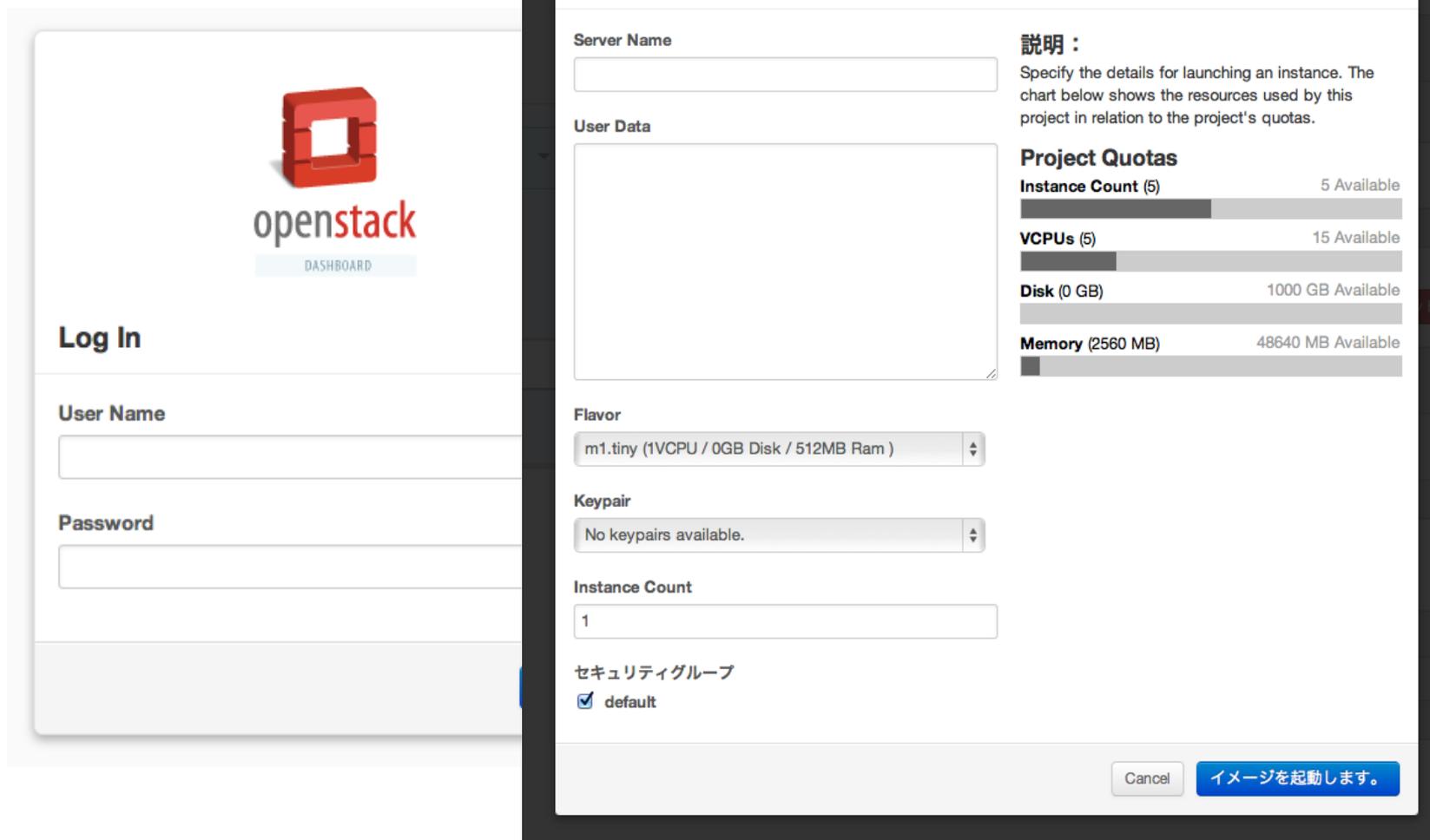


結局ね...
ソースコードを
読むしかないって
ことですよ。

調べてみました。
まずは仮想マシンの
作成&起動を
の流れから。

仮想マシンを起動する

ここでは、WebUIであるhorizonから仮想マシンを作成・起動するまでを簡単に紹介します。



The screenshot displays the OpenStack Horizon dashboard on the left and a 'Launch Instances' dialog box on the right. The dashboard includes the OpenStack logo, a 'Log In' section with 'User Name' and 'Password' fields, and a 'DASHBOARD' button. The 'Launch Instances' dialog box contains the following fields and sections:

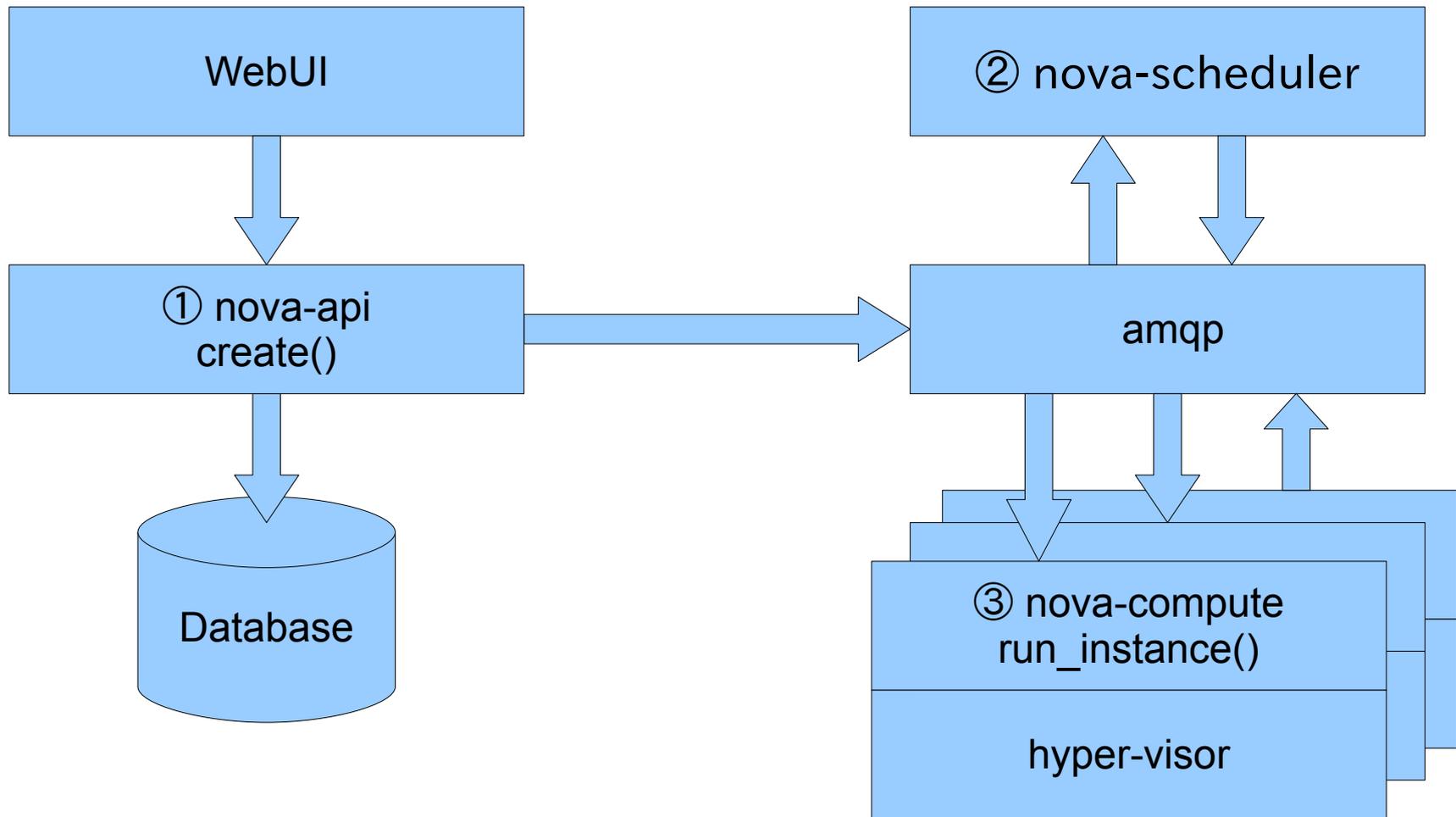
- Server Name:** An empty text input field.
- User Data:** A large empty text area.
- Flavor:** A dropdown menu showing 'm1.tiny (1VCPU / 0GB Disk / 512MB Ram)'.
- Keypair:** A dropdown menu showing 'No keypairs available.'
- Instance Count:** A text input field containing the number '1'.
- セキュリティグループ (Security Group):** A checked checkbox next to 'default'.
- Project Quotas:** A section with a '説明:' (Description) and a chart showing resource usage. The chart data is as follows:

Resource	Used	Available
Instance Count (5)	0	5 Available
VCPUs (5)	0	15 Available
Disk (0 GB)	0	1000 GB Available
Memory (2560 MB)	0	48640 MB Available

At the bottom of the dialog box, there are two buttons: 'Cancel' and 'イメージを起動します。' (Launch Image).

仮想マシンを起動する

WebUIから”イメージを起動します”を選択すると、以下のようなフローで仮想マシンが作られます。



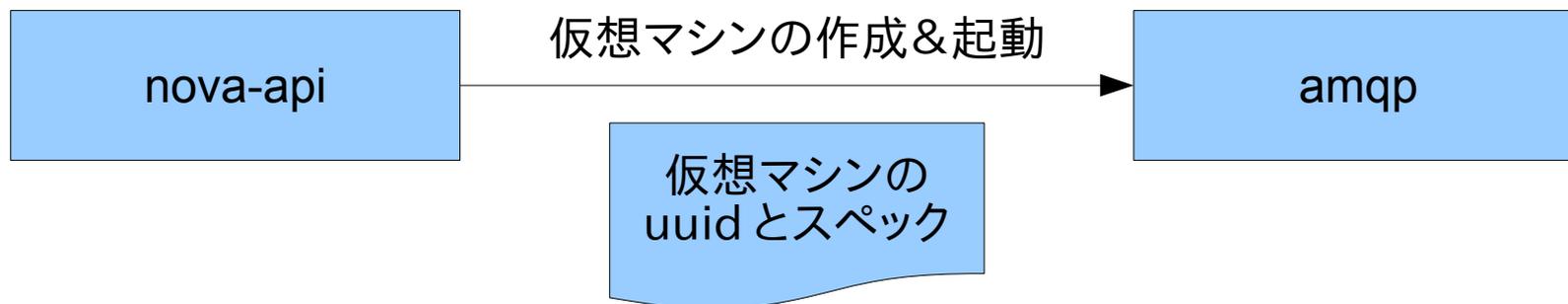
超訳：仮想マシンを起動する

OpenStack側の登場人物は、nova-api/nova-scheduler/nova-computeの3名。

① nova-api

WebUIからの起動リクエストを受け取り、渡されたスペックに従って仮想マシンの作成&起動を行うRPCをamqpに投入します。

この時点では、どのcomputeに仮想マシンを起動させるかは指示されていない。

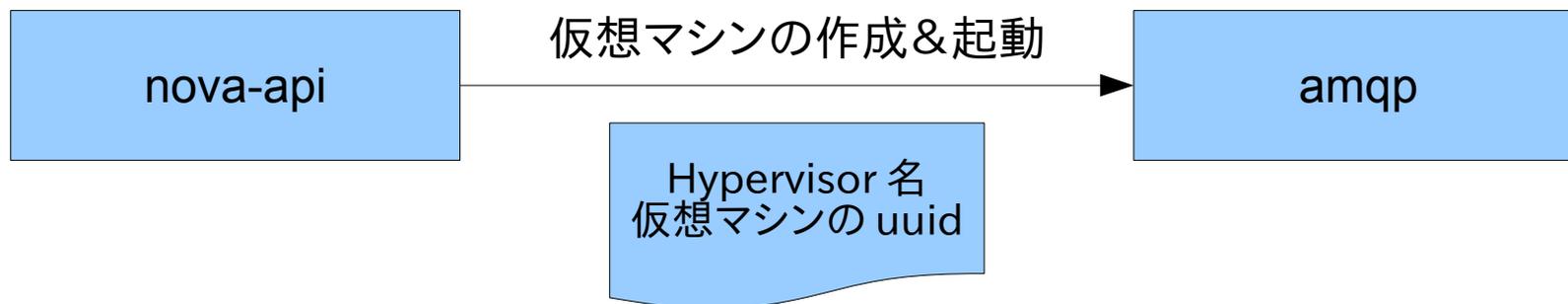


超訳：仮想マシンを起動する

② nova-scheduler

amqpのキューに投入された仮想マシンの作成&起動メッセージを実行するhypervisorを選択して対象hypervisorに向けた仮想マシンの作成&起動RPCをamqpに投入します。

ここで初めて仮想マシンをどのhypervisorで起動するかが決まります。

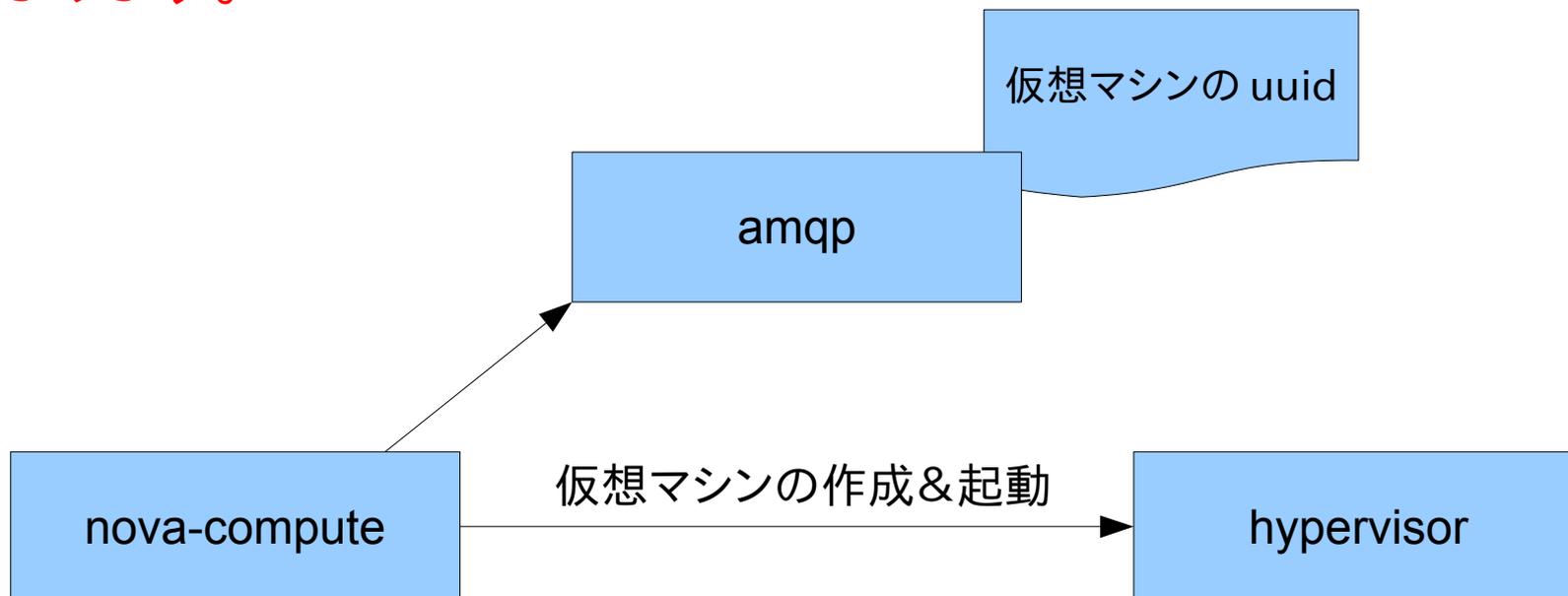


超訳：仮想マシンを起動する

③ nova-compute

amqp のキューに投入された、仮想マシンの作成・起動メッセージを実行する compute ノードを選択、そこに向けた仮想マシンの作成 & 起動 RPC を amqp に投入します。

ここで初めて仮想マシンをどの compute ノードで起動するかが決まります。



リソースのスケジューリング

nova-schedulerは、OpenStackの管理下にあるcomputeノード/volumeノードの中から、さまざまな要件により、リソースをどのノードに割り当てるかを決定するための機構を提供します。

設定ファイル(nova.conf)内での指定は3箇所。

- /etc/nova/nova.conf

```
scheduler_driver="nova.scheduler.multi.MultiScheduler"  
compute_scheduler_driver="nova.scheduler.filter_scheduler.SimpleScheduler"  
volume_scheduler_driver="nova.scheduler.chance.ChanceScheduler"
```

- ① scheduler_driver (scheduler全体の挙動を制御するドライバ)
- ② compute_scheduler_driver (computeノード選択用ドライバ)
- ③ volume_scheduler_driver (volumeノード選択用ドライバ)

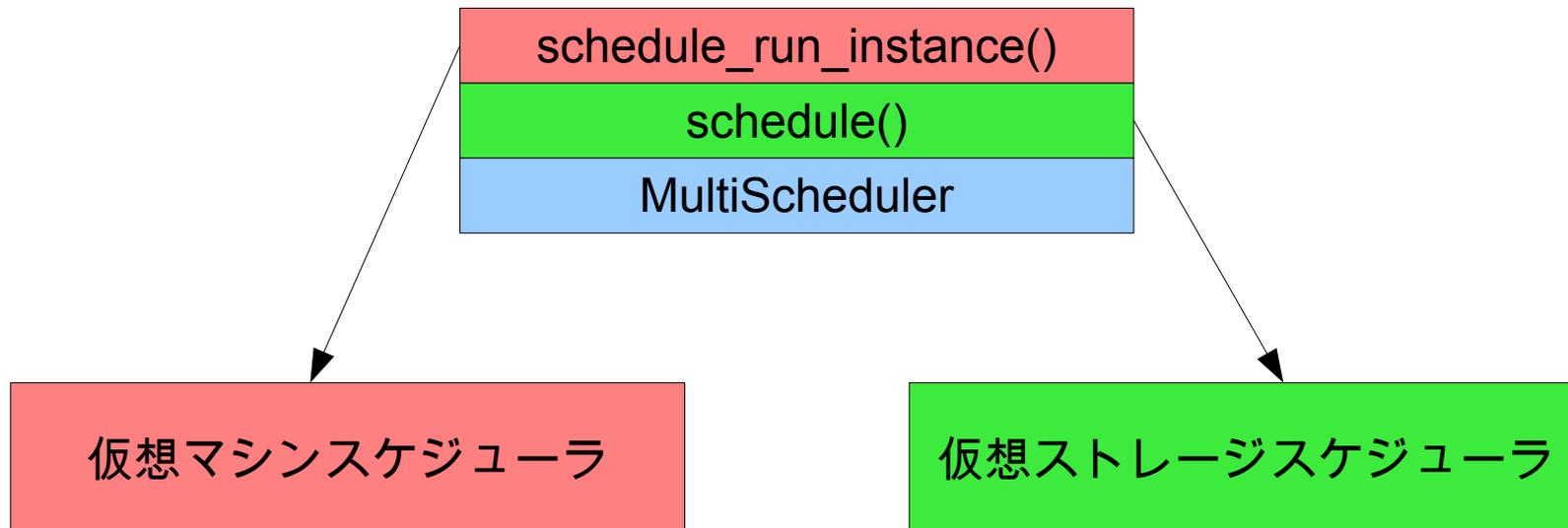
リソースのスケジューリング

スケジューリングドライバには、以下の4種類があります。
MultiSchedulerは、scheduler全体の動きを決めるscheduler_driverとして使用します。基本ルールはノードリストを優先度順でソート。

- ① MultiScheduler
scheduler全体の動きを決める。サブスケジューラとしてcompute/volumeそれぞれ別のドライバを指定可能。
- ② SimpleScheduler
仮想マシンのスケジュールに利用される場合、使用済みコア数で優先度が付けられる。割り当て済みコア数が多いほど優先度が低くなる。
- ③ ChanceScheduler
MultiSchedulerのvolumeノード用デフォルトサブスケジューラ。
- ④ FilterScheduler
MultiSchedulerのcomputeノード用デフォルトサブスケジューラ。
解析中なう! OSC2012-Tokyo/Fallを待て!

MultiScheduler

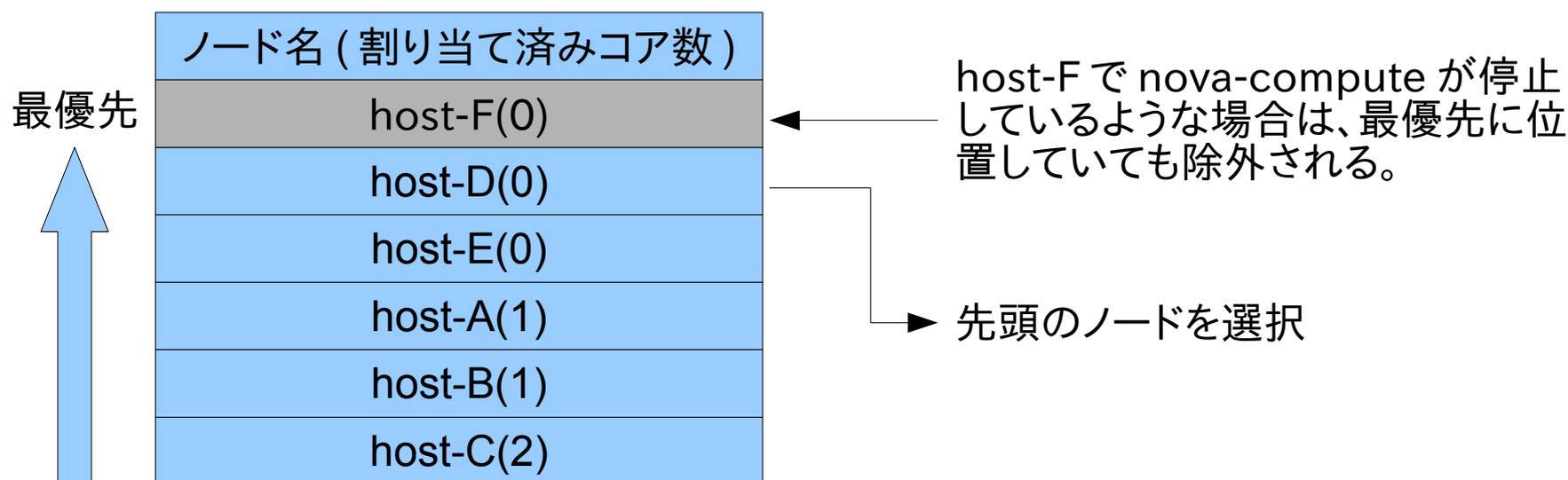
nova-scheduler全体の動きを制御します。
 実際のスケジューリングはサブスケジューラに丸投げしますが、サブスケジューラのドライバ毎にある差異を吸収するためのラッパーの役割も果たします。



SimpleScheduler

仮想マシンと仮想ストレージのスケジューリングを行う。
仮想マシンのスケジューリングアルゴリズムにフォーカスすると、動きは以下ようになります。(仮想ストレージの話は、またそのうちに...)

- ① computeノード一覧を作成
割り当て済みコア数の少ない順にリストアップする。コア数以外の割り当て済みメモリサイズなど、その他の要素は考慮されない。
- ② 無効なノード(nova-computeが停止しているノードなど)を除外
- ③ リストの先頭となった1台を選択



ChanceScheduler

MultiScheduler が利用する volume ノード選択用デフォルトサブスケジューラですが、compute ノード選択用にも利用することができます。compute/volume 共通のアルゴリズムでノードを選択します。

- ① ノード一覧を作成
サービス (compute または volume) ノード一覧を作成
- ② 無効なノード (nova-compute が停止しているノードなど) を除外
- ③ リストの先頭となった1台を選択
ノード選択は `host[int(random.random() * len(hosts))]`

ノード名 (割り当て済みコア数)
host-F(0)
host-A(1)
host-C(2)
host-D(0)
host-B(1)
host-E(0)

host-F で nova-compute が停止しているような場合は、最優先に位置していても除外される。

→ ③ のアルゴリズムでノードを選択

MultiScheduler が利用する compute ノード選択用デフォルトサブスケジューラです。

いろいろできるらしいです。
結構まじめに調べてます。
戦いは続く...

次回 (OSC2012 Tokyo/Fall あたり) を待て!

① リソーススケジューラを使うメリット

=> 仮想マシンや仮想ストレージの配置をルールに従って自動で行ってくれます。

数十台、数百台規模になったら配置を都度指定して作成... そんな運用してられません...

なので、自動スケジューラは重要な機能なのです。

② リソースってなに？

=> ここでは仮想マシンと仮想ストレージのことです。

③ リソーススケジューラはどのような時に利用されるのか？されないのか？

=> リソースが新規に作成される際に利用されます。

=> 仮想マシンのライブマイグレーションについても利用したいところですが、現状は利用できません。

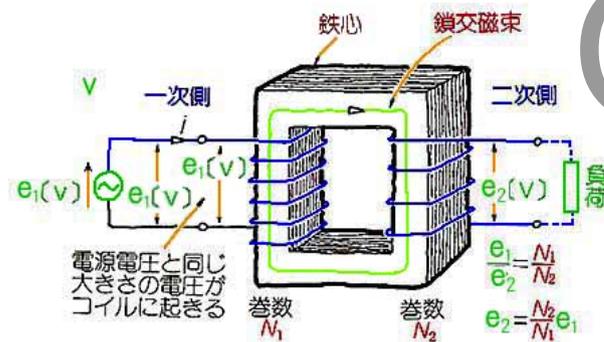
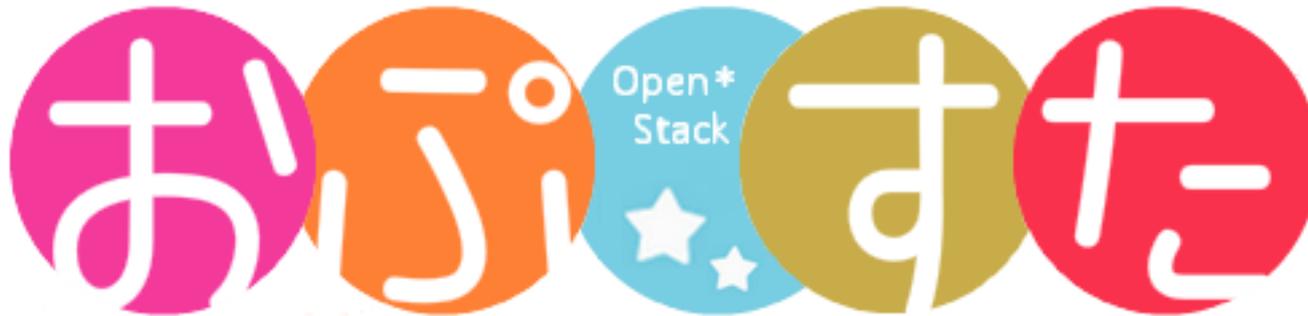
はあ！？スケジューラ！？

まずはインストールだろ？というアナタのための参考URL

- 本家
<http://www.openstack.org/>
- 日本オープンスタックユーザ会
<http://www.openstack.jp/>
- いしかわさんの2done
<http://2done.org/openstack/index.html>

Special Thanks

ご清聴ありがとうございました m(__)m



Openstack
JAPAN