# ironic部署方案

[ironic部署方案](#header-c1)  
[术语](#header-c17)  
[1. 概述](#header-c56)  
 [1.1 功能描述](#header-c57)  
[2. 设计原理](#header-c67)  
 [2.1 关键技术](#header-c68)  
 [进度条](#header-c69)  
 [linux 进度条方案](#header-c70)  
 [windows进度条](#header-c89)  
 [raid配置（openstack-ironic无成熟功能）](#header-c93)  
 [方案一](#header-c94)  
 [方案二（要修改openstack-ironic社区代码流程）](#header-c98)  
 [PXE](#header-c154)  
 [IPMI](#header-c157)  
 [2.3 系统扩展](#header-c161)  
 [2.4 系统性能](#header-c165)  
 [2.5 兼容性](#header-c169)  
[3. 系统架构](#header-c174)  
 [3.1 架构说明](#header-c175)  
 [3.1.1 硬件网络平面](#header-c176)  
 [3.1.2 部署网络切换（交换机及vlan操作）](#header-c179)  
 [部署前](#header-c180)  
 [部署后](#header-c186)  
 [3.1.3 软件逻辑](#header-c192)  
 [3.1.4 软件组网](#header-c195)  
 [3.1.4.1 简单组网](#header-c196)  
 [3.1.4.2 静态组网](#header-c199)  
 [3.1.4.3 祼机虚机共管组网](#header-c218)  
 [3.2 配置说明](#header-c237)  
 [3.2.1 neutron配置](#header-c238)  
 [3.2.2 ironic配置](#header-c249)  
 [3.2.3 管理员工程设置](#header-c267)  
 [3.2.4 管理员界面配置（provider界面）](#header-c278)  
 [3.3 部署说明](#header-c302)  
 [3.3.1 部署时序（从nova boot开始，由provider触发）：](#header-c303)  
 [3.3.2 部署流程](#header-c306)  
 [3.3.3 ks定制(根据用户需求确认)](#header-c347)  
[4. 协作说明](#header-c373)  
 [4.1 组件协作流程](#header-c374)  
 [4.2 需要各组件开发的功能](#header-c416)  
 [4.2.1 raid配置](#header-c417)  
 [4.2.2 进度查询](#header-c451)  
 [4.2.3 flavor创建](#header-c468)  
 [4.2.4 主机发现](#header-c476)  
 [4.2.5 ks定制](#header-c490)  
 [4.2.6 部署裸机](#header-c501)  
[5.组件说明](#header-c519)  
 [5.1 ironic修改](#header-c520)  
 [5.2 provider修改](#header-c534)  
[6. 接口说明](#header-c539)  
 [6.1 ironic接口](#header-c540)  
 [6.1.1 进度查询：](#header-c541)  
 [6.1.2 其它ironic restful api接口](#header-c546)  
[7. 设计局限性](#header-c552)

# 术语

|  |  |  |
| --- | --- | --- |
| 术语/定义 | 英文 | 说明 |
| provider |  | 硬件管理界面 |
| ironic |  | 裸机部署服务 |
| PXE | Preboot Execution Environment | 预启动执行环境 |
| IPMI | Intelligent Platform Management Interface | 智能平台管理接口 |
| TFTP | Trivial File Transfer Protocol | 简单文件传输协议文件传输协议 |
| DHCP | Dynamic Host Configuration Protocol | 动态主机设定协定 |
| VLAN | Virtual Local Area Network | 虚拟局域网 |
| TECS | Tulip Elastic Cloud System | 郁金香弹性云系统 |

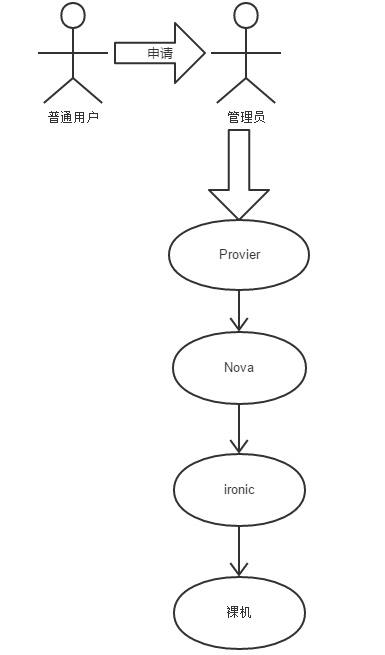
# 1. 概述

## 1.1 功能描述

本文提供了解决大集群裸机的部署和管理的方案。通过openstack-ironic组件来替代传统的手工部署模式。

本方案主要是完成物理裸机的发现，部署，管理，分配，删除等功能。

普通用户可以像管理员申请裸机资源，管理员通过操作provider web来完成裸机的部署和分配，具体用例图如下图所示。



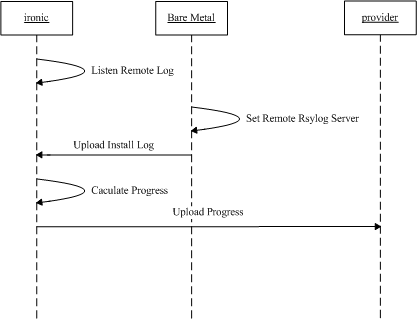
# 2. 设计原理

## 2.1 关键技术

### 进度条

#### linux 进度条方案

1. 裸机下载的anaconda小系统配置日志发送到ironic-conductor服务器；
2. 在ironic-conductor节点启动rsyslog服务，接受裸机发过来的日志信息；
3. 根据系统类型分析安装日志；
4. 上传进度，在provider页面显示；
5. 进度显示时序图如下：



#### windows进度条

**待验证**

### raid配置（openstack-ironic无成熟功能）

#### 方案一

用户自己提前配置好raid；这种限制比较多，不适合数量较多的情况。

#### 方案二（要修改openstack-ironic社区代码流程）

使用小系统配置raid，这种方式适合大量部署，由于raid控制器型号不同，实际使用前需要兼容性验证。

目前小系统配置raid支持范围：

* 中兴华为E9000
* dell R730
* 联想 X3650
* HP C7000

如果这一部分需要实现，需要provider提供UI。主要流程如下：

1. 用户在provider界面设置raid信息（硬盘、槽位号、raid类型）；
2. provider通知ironic配置raid，并传递raid配置信息；
3. ironic准备好raid配置的小镜像，配置好DHCP服务器；
4. 设置裸机从pxe启动；
5. 复位裸机；
6. 裸机发送DHCP请求；
7. 裸机下载raid小系统；
8. 裸机配置raid；
9. 通过ipmi关闭裸机；
10. 通知provider raid配置结果。

### PXE

目标祼机需要支持PXE网络启动方式。

### IPMI

目标祼机需要有IPMI管理。

## 2.3 系统扩展

可以通过增加新的ironic部署代理的方式，提高系统并行部署多台祼机的能力。

## 2.4 系统性能

可以通过增加新的ironic部署代理的方式，提高系统并行部署多台祼机的能力。

## 2.5 兼容性

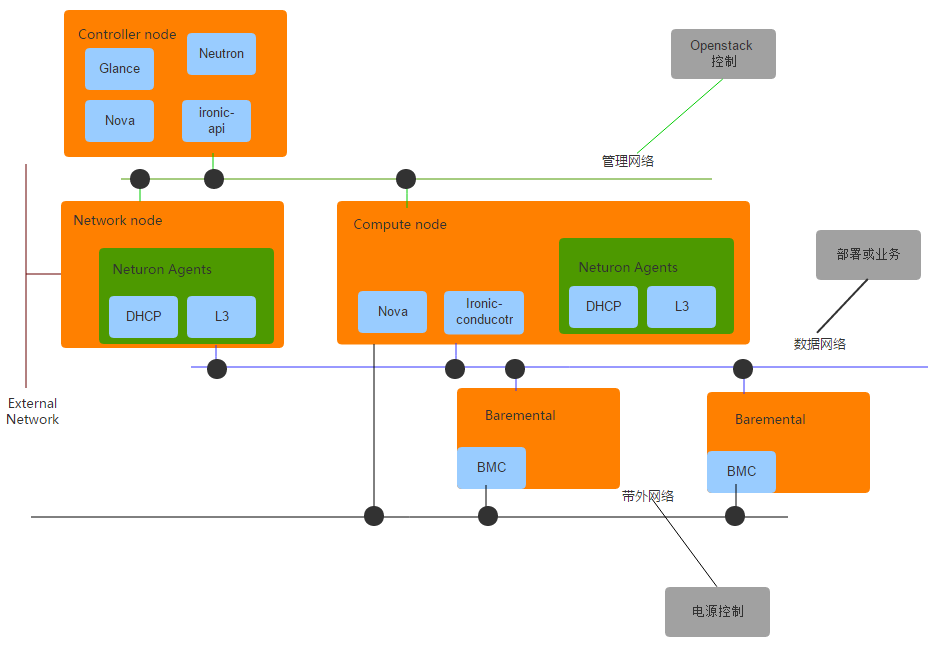
与 Openstack Mitaka 版本兼容。

与TECS 3.0版本兼容。

# 3. 系统架构

## 3.1 架构说明

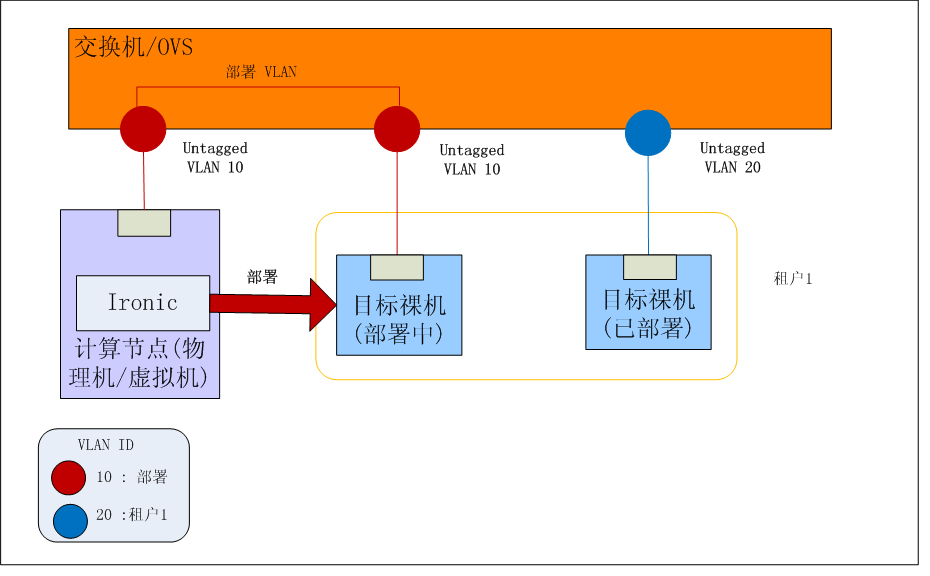
### 3.1.1 硬件网络平面



### 3.1.2 部署网络切换（交换机及vlan操作）

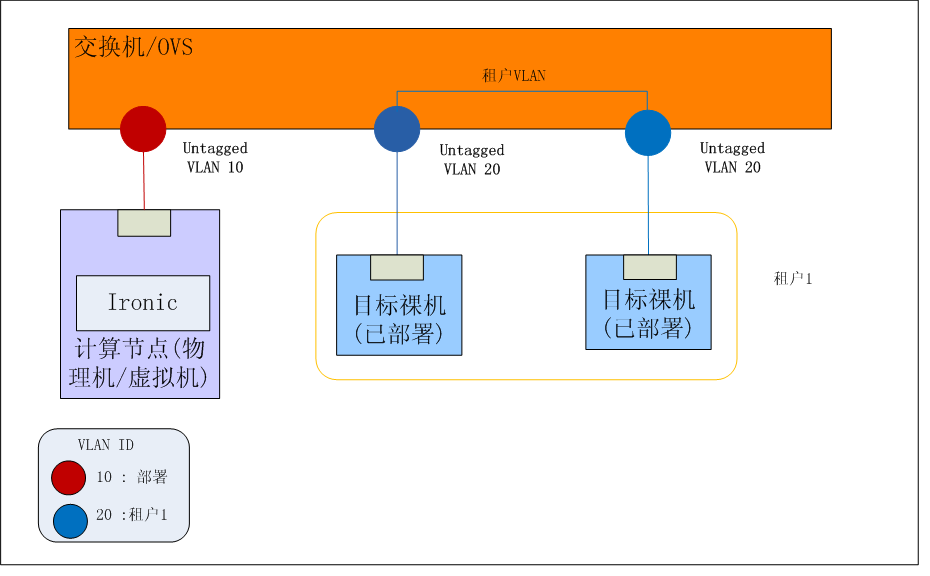
#### 部署前

在部署前，用户需要保证待部署祼机与ironic部署节点的网络连通性。所有裸机网口必须设置为允许从pxe启动。 如果在VLAN网络中， 则需要保证待部署祼机与ironic部署节点在同一个部署VLAN中。 如下图所示：

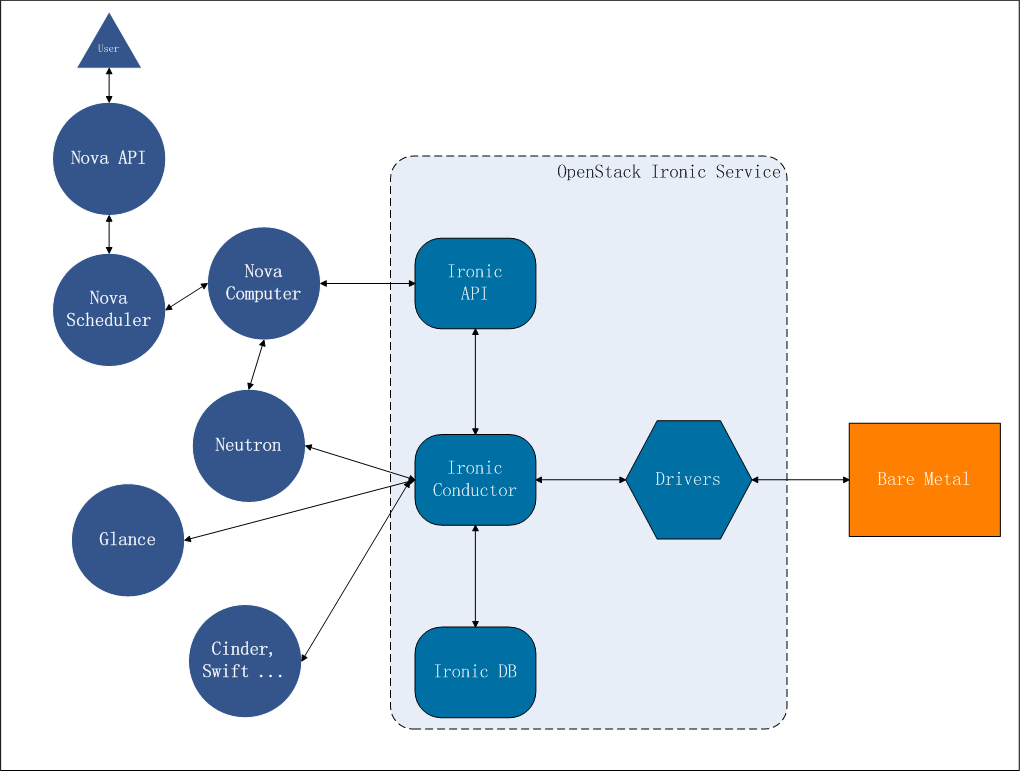


#### 部署后

在部署完成后， 部署完成的祼机可以断开与ironic部署节点的网络连接。 如果在VLAN网络中， 部署完成后用户可以将裸机加入任意其他VLAN中。如下图所示：

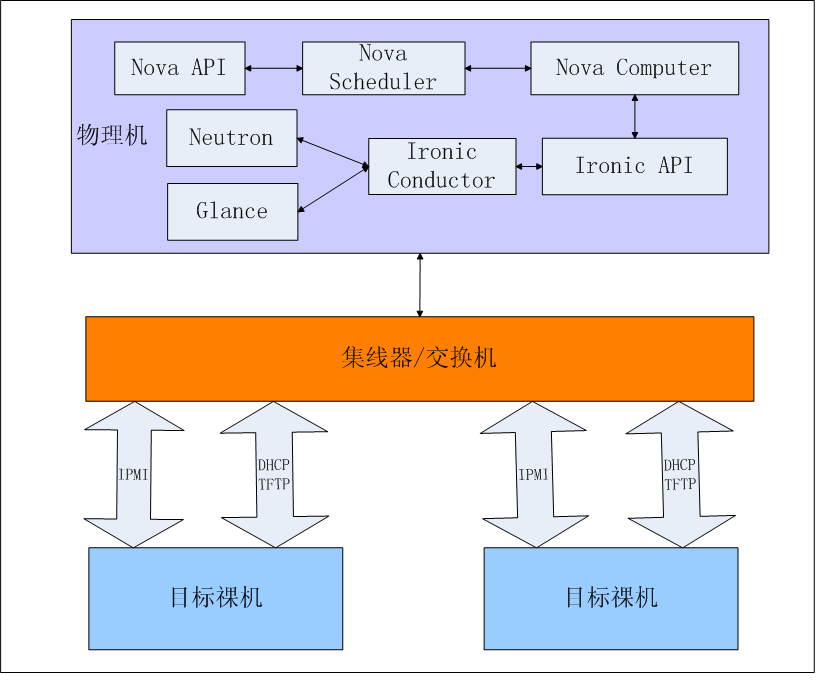


### 3.1.3 软件逻辑



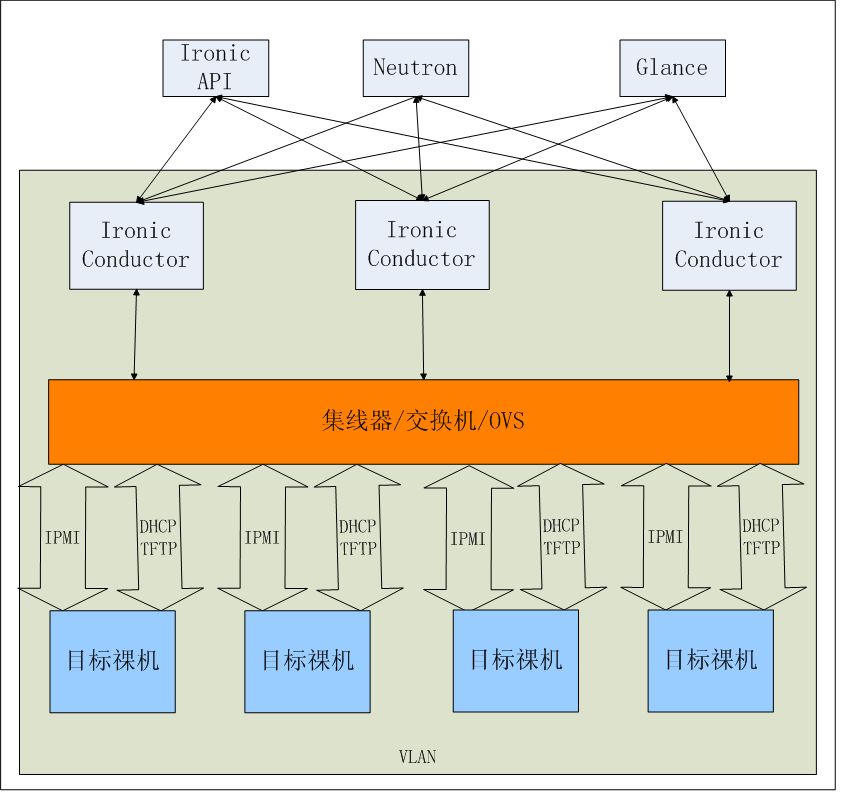
### 3.1.4 软件组网

#### 3.1.4.1 简单组网



#### 3.1.4.2 静态组网

* 管理员将要部署的裸机加入到指定的vlan(部署使用的vlan)
* 每个祼机需要提供一个专门的网口用于ironic部署**（用户保证）**
* 所有的祼机的部署专门的网口都需要在部署专用flat网络里**（用户保证）**
* 当部署完成后， 管理员可将网口接入业务vlan；
* 部署专用的flat网络中可以有多个ironic condutor 用于部署祼机



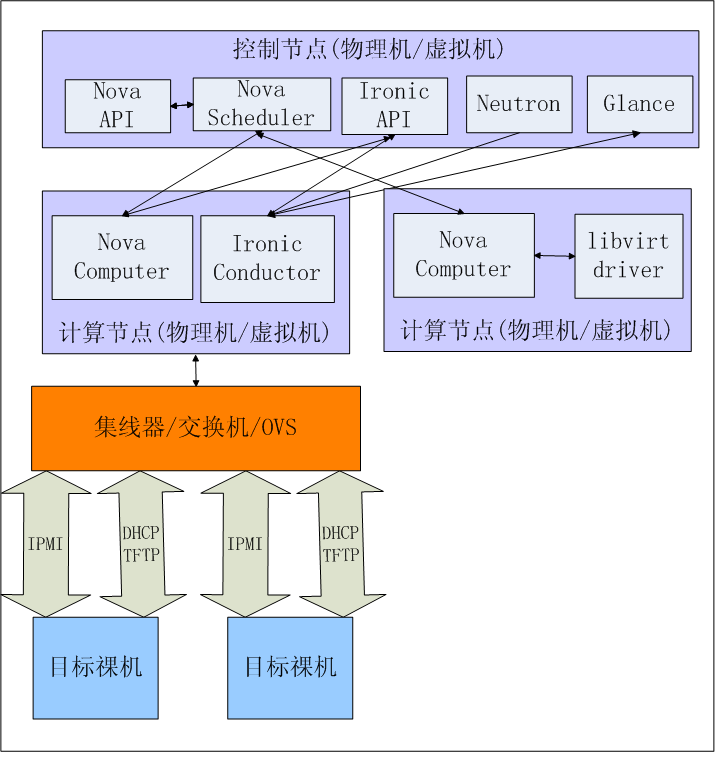
#### 3.1.4.3 祼机虚机共管组网

由于一个nova-compute一次只能使用一种驱动，起虚机是用的libvirt驱动，而部署物理机是需要使用ironic驱动。

目前nova已经提供了共管的方式来同时管理裸机和虚机，参考：[nova共管配置](http://wiki.zte.com.cn/pages/viewpage.action?pageId=31065166)

需要说明的是共管方式是:

* 共管方式只是用拿出一个节点提供ironic驱动，专门用来部署裸机
* 提供ironic驱动的机器不能用来起虚机
* 裸机使用的flavor必须要设置hw:compute\_type=ironic属性



## 3.2 配置说明

### 3.2.1 neutron配置

* 用户创建部署专用网络和子网。
* neutron需要在部署网络里要有dhcp-agent，并且使用的部署网络绑定到该agent；
* 管理员指定ironic-conductor的ip（需要和部署网络在同一个网段，且不占用dhcp ip地址池内ip）

### 3.2.2 ironic配置

* 安装ironic-conductor和ironic-api;
* 修改配置文件(ironic， nova， neutorn);
  + 修改neutron配置文件，增加flat类型；
  + 修改ironic-conductor节点的nova驱动（使用ironic驱动）;
  + 设置共管方式 (裸机和虚机共存)；

### 3.2.3 管理员工程设置

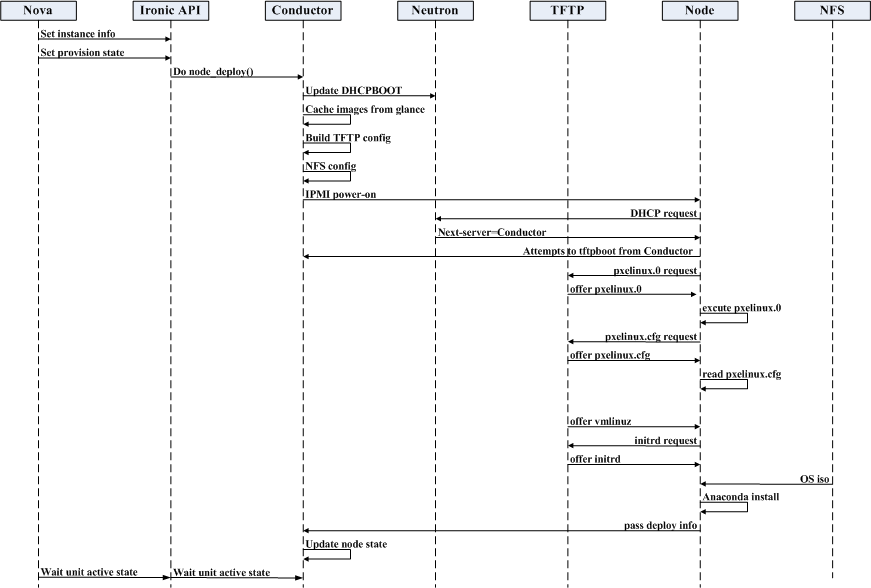
* 目标机所有网口设为允许pxe启动。（**管理员**工程设置）
* **管理员**将裸机加入到裸机部署网络。（vlan环境需要加入到部署vlan网络）
* neutron网络保证DHCP网口和目标机目标pxe网口通。（**管理员**工程设置）

### 3.2.4 管理员界面配置（provider界面）

* 管理员在界面操作发现主机，由provider创建裸机对应的ironic node和port
  + 从provider获取IPMI地址用户名密码，目标机硬件信息（内存、CPU、硬盘）。
  + 创建对应物理机的ironic node，添加信息。（provider自动完成）
* glance中加入需要部署的系统镜像。
* 如果管理员勾选部署裸机，则由他根据ironic node信息表，创建目标机硬件信息对应的nova flavor，由nova设置flavor属性：nova flavor-key flavorID set hw:compute\_type=ironic
* 选择要安装的操作系统。
* 界面配置目标系统参数（见ks定制章节）。

## 3.3 部署说明

### 3.3.1 部署时序（从nova boot开始，由provider触发）：



### 3.3.2 部署流程

1. 假设准备工作已经完成，用户在页面上选择好了物理机，flavor，和要部署的镜像等，点击开始部署，进入nova boot。
2. nova将要部署的节点信息发送给Ironic API，通知Ironic进入部署状态。
3. API调度conductor执行部署。
4. neutron根据mac地址，在指定的子网下建立mac对应的port，并分配指定网段内ip地址。
5. conductor从glance缓存镜像信息，配置tftp服务器。这里，根据nova boot传过来的镜像信息，解压iso，拷贝到tftp指定目录下，生成node UUID的文件夹，包含config、vmlinuz、initrd.img。在pxelinux.cfg目录下，建立对应mac地址为名字的软链接，链接到前面的UUID文件夹。
6. 准备NFS服务器。
7. 目标机上电设置网络启动，向服务器发送dhcp请求。
8. 服务器响应请求，分配一个ip，并且告诉目标机，要通过tftp下载pxelinux.0。
9. 目标机得到响应后，就通过tftp到服务器的tftp路径下去下载pxelinux.0文件。
10. 目标机下载到pxelinux.0文件后，执行这个文件，指导下一步的动作。pxelinux.0让目标机到服务器的tftp路径下载default文件，这里会优先下载pxelinux.cfg下的带有mac地址的文件。
11. 下载到了带mac地址的default程序后，里面会有内核文件的路径kernel vmlinuz、最小系统的路径initrd=initrd.img以及ks文件的路径，然后就去下载vmlinux和initrd.img。
12. 执行vmlinuz启动内核，引导启动initrd.img，小系统里引导anaconda程序，anaconda程序会根据default文件中的ks文件路径去服务器下载kickstart文件，解析kickstart文件，并且开始安装。
13. 安装完成后，通知conductor已经安装完毕。conductor自己修改ironic node的状态为active，nova改变节点状态为active，到此裸机部署结束。

### 3.3.3 ks定制(根据用户需求确认)

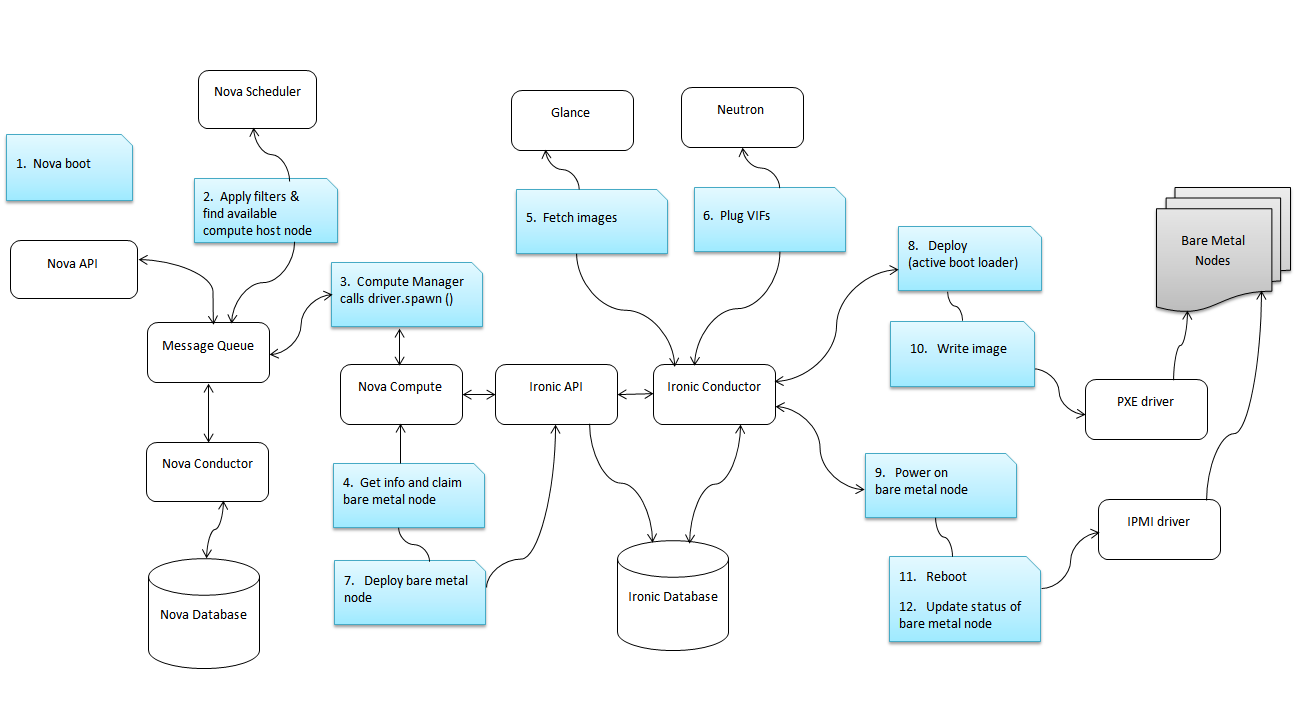
目前安装操作系统采用定制ks文件的方式，提供的每一种操作系统定制一个kickstart文件，ks文件中包括：

* 用户创建（用户名及密码提供默认值root:root，**细化讨论**）
* 时区设置
* 分区（**需要根据用户需求再细化**，默认值：第一个物理盘作为系统盘）
* ssh key生成（可选）
* hostname配置
* 用户自定义脚本

以上配置也可以提供默认值，系统装完后由用户配置。

# 4. 协作说明

## 4.1 组件协作流程



1. nova-api接收到用户的部署请求，接着发送消息通知nova-scheduler。
2. nova-scheduler使用调度器选择合适的计算节点。
3. 计算节点的nova-compute服务启动一个协程来实现部署动作。
4. nova-compute获取ironic数据库中对应的node，并将该node资源占用住。
5. 从Glance下载物理机部署需要的镜像到ironic-conductor所在的节点。
6. 在ironic-conductor节点上通过neutron启动DHCP服务来支持PXE/TFTP部署（目前支持neutron的flat和vlan网络）。
7. nova-compute通过ironic-api发送请求给ironic-conductor来部署物理机。
8. ironic的PXE驱动配置一个tftp文件。
9. ironic的IPMI驱动发送命令让物理机上电，这次是让物理机从网络启动。
10. 物理机以网络安装方式启动，通过DHCP服务获取到ip后，再根据TFTP服务去获取安装镜像，安装系统。
11. ironic的IPMI驱动会再次重启物理机，这次是将物理机从硬盘启动。
12. 更新ironic的数据库中node状态与nova数据库中instance的状态。

## 4.2 需要各组件开发的功能

### 4.2.1 raid配置

1. provider提供raid配置 UI
   * 显示raid信息UI (硬盘槽位号， raid类型，raid卡型号)
   * 设置页面
2. provider 点击raid配置后，会将参数以json格式传给ironic
   * ironic 向neutron创建裸机的port
   * 解析json字符串
   * 裸机走pxe流程，下载raid配置小映像
   * 关闭裸机
   * 向provider返回结果
3. provider向用户显示raid配置

### 4.2.2 进度查询

1. 裸机把安装日志发送到ironic-conductor节点；
2. provider在通过定时发送http请求裸机安装进度；
3. ironic-api接收到proivder的请求后，调用ironic-conductor；
4. ironic-conductor根据node\_uuid找到对应裸机日志，分析安装日志，并返回进度；
5. provider在页面显示进度；

### 4.2.3 flavor创建

1. 管理员在provider页面查看可用的裸机；
2. 根据裸机信息调用nova-api创建flavor，设置flavor属性：nova flavor-key flavorID set hw:compute\_type=ironic，这里最好指定flavor 的uuid和node的uuid保持一致；

### 4.2.4 主机发现

1. 管理员保证要部署的裸机加入到部署vlan，并开启网口的dhcp功能；
2. provider提供发现UI，管理员点击对应裸机的发现按钮；
3. provider调用ironic-api，创建ironic node和ironic port(其中指定node的驱动，ipmi信息，启动方式，映像文件，硬件信息)；
4. provider显示已经发现的裸机；

### 4.2.5 ks定制

1. provider界面提供系统参数设置的页面。
2. 用户设置完系统参数后，生成ks文件，随映像一起传给ironic-api。
3. ironic-api在部署过程中，传给conductor，配置tftp服务器，在安装过程中使用该定制ks。

### 4.2.6 部署裸机

1. provider提供部署页面，供用户选择要部署的裸机；
2. 用户选择要部署的操作系统映像文件，选择非iso映像时提示错误；
3. ironic-conductor从映像文件中提取vmlinuz和initrd，替换tftp服务器中的映像；
4. ironic完成裸机的部署工作；
5. 更新ironic node状态。

# 5.组件说明

### 5.1 ironic修改

1. ironic需要集成到TECS， 完成ironic安装部署; 修改nova配置, neutron配置， 数据库配置；
2. 修改社区ironic安装流程，去除裸机与ironic api 交互过程；
3. 用自有系统替换社区部署镜像， 并进行定制开发验证；
4. 新增部署流程支持用户系统采用Windows， 开发并验证；

### 5.2 provider修改



* 1. 用户通过provider硬件管理页面选择刀片或者架式服务器设置为“物理裸机”
  2. Ironic通过查询接口获取所有设置为“物理裸机”类型的物理机列表。

# 6. 接口说明

## 6.1 ironic接口

### 6.1.1 进度查询：

method GET  
URI /api/v1/ironic/nodes/uuid/progress  
  
# 这里的token是通过keystone认证获取到的  
headers = {'X-Auth-Token': token}

Response

{  
 "status": "OK/ERROR"，  
 "err": "msg",  
 "porgress": 0  
}

### 6.1.2 其它ironic restful api接口

这里provider主要需要根据裸机的硬件信息创建ironic的node和port，node的uuid使用provider硬件管理主机的uuid，这样方便后续操作。具体api接口参考： [restful api接口](http://developer.openstack.org/api-ref/baremetal/)

关于node使用的驱动默认采用"pxe\_ipmitool"

## 6.1provider接口

###Get /api/v1.0/hardware/nodes?user=Ironic###

### direction POSD/Daisy&ironic -> hardware ###

Request body is empty.

\*\*Successful\*\* HTTP response will be \*\*200\*\*, Response body will be a list of board list to the client.

Response body for example:

{

"nodes": [

{

"id": "6CU521YHFA",

"serialNo": "6CU521YHFA",

"podId": "",

"cpuCore": 16,

"cpuFrequency": "2.60GHz",

"memory": 64,

"disk": 279,

"hardwareType": "BladeServer: HP-C7000: BL460c",

"hardwareStatus": "normal",

"bios":

{

"biosBootMode": "unknown",

"cpuVTxMode": "unknown",

"nicPXEMode": "unknown",

"nicSRIOVMode": "unknown"

},

"ipmiInfo":

{

"user": "Administrator",

"password": "Aa888888",

"address": "10.62.40.222",

"port": 623

},

"interfaces":

[

{

"name": "Port 2",

"mac": "8c:dc:d4:b6:0e:d9",

"bandwidth": "10Gb",

"adapterName": "Mezzanine 1",

"description": "HP Ethernet 10Gb 2-port 560M Adapter"

},

{

"name": "Port 1",

"mac": "8c:dc:d4:b6:0e:d8",

"bandwidth": "10Gb",

"adapterName": "Mezzanine 1",

"description": "HP Ethernet 10Gb 2-port 560M Adapter"

},

{

"name": "NIC 2",

"mac": "5C:B9:01:8A:7E:79",

"bandwidth": "10Gb",

"adapterName": "FLB Adapter 1",

"description": "HP Ethernet 10Gb 2-port 560FLB Adapter"

},

{

"name": "NIC 1",

"mac": "5C:B9:01:8A:7E:78",

"bandwidth": "10Gb",

"adapterName": "FLB Adapter 1",

"description": "HP Ethernet 10Gb 2-port 560FLB Adapter"

}

],

"self": "/api/v1.0/hardware/nodes/6CU521YHFA",

"links":

[

{

"uri": "/api/v1.0/hardware/nodes/6CU521YHFA/allocation",

"rel": "allocation"

},

{

"uri": "/api/v1.0/hardware/nodes/6CU521YHFA/restart\_actions",

"rel": "restart"

},

{

"uri": "/api/v1.0/hardware/nodes/6CU521YHFA/one\_time\_boot",

"rel": "oneTimeBoot"

}

]

},

{

"id": "6CU521YHFC",

"serialNo": "6CU521YHFC",

"podId": "",

"cpuCore": 16,

"cpuFrequency": "2.60GHz",

"memory": 64,

"disk": 279,

"hardwareType": "BladeServer: HP-C7000: BL460c",

"hardwareStatus": "normal",

"bios":

{

"biosBootMode": "unknown",

"cpuVTxMode": "unknown",

"nicPXEMode": "unknown",

"nicSRIOVMode": "unknown"

},

"ipmiInfo":

{

"user": "Administrator",

"password": "Aa888888",

"address": "10.62.40.106",

"port": 623

},

"interfaces":

[

{

"name": "Port 2",

"mac": "8c:dc:d4:b6:0f:0d",

"bandwidth": "10Gb",

"adapterName": "Mezzanine 1",

"description": "HP Ethernet 10Gb 2-port 560M Adapter"

},

{

"name": "Port 1",

"mac": "8c:dc:d4:b6:0f:0c",

"bandwidth": "10Gb",

"adapterName": "Mezzanine 1",

"description": "HP Ethernet 10Gb 2-port 560M Adapter"

},

{

"name": "Port 2",

"mac": "5C:B9:01:8A:8D:05",

"bandwidth": "10Gb",

"adapterName": "FLB Adapter 1",

"description": "HP Ethernet 10Gb 2-port 560FLB Adapter"

},

{

"name": "Port 1",

"mac": "5C:B9:01:8A:8D:04",

"bandwidth": "10Gb",

"adapterName": "FLB Adapter 1",

"description": "HP Ethernet 10Gb 2-port 560FLB Adapter"

}

],

"self": "/api/v1.0/hardware/nodes/6CU521YHFC",

"links":

[

{

"uri": "/api/v1.0/hardware/nodes/6CU521YHFC/allocation",

"rel": "allocation"

},

{

"uri": "/api/v1.0/hardware/nodes/6CU521YHFC/restart\_actions",

"rel": "restart"

},

{

"uri": "/api/v1.0/hardware/nodes/6CU521YHFC/one\_time\_boot",

"rel": "oneTimeBoot"

}

]

}

]

}

# 7. 设计局限性

* 由于没有控制物理交换机，不能动态调整VLAN。
* 在VLAN环境下, 管理员需要在部署前保证祼机在ironic部署网络的VLAN中。
* 目前无法指定特定的 condutor 用于部署指定的祼机。
* 管理员需要保证祼机与ironic网络的网络连通性