**NOVA:**

1. **Image precaching service**

BP: <https://blueprints.launchpad.net/nova/+spec/compute-image-cache>

nova-specs: [https://review.openstack.org/#/c/85792](https://review.openstack.org/%23/c/85792)

nova-codes: [https://review.openstack.org/#/c/86393/](%20https:/review.openstack.org/%23/c/86393/) (WORKINPROGRESS)

功能介绍：

镜像预缓存服务，主要用来把热点镜像提前缓存到计算节点，管理员api，后续计划支持调度器的过滤功能，优先选择已经缓存有本次创建所用镜像的计算节点启动虚拟机，主要目标是加快虚拟机创建速度，减少创建过程中IO、网络带宽消耗。

使用场景：

公有云，并且基础镜像数量比较多的情况。对于私有云，镜像数量和计算节数量都有限，创建一定数量的虚拟机之后就没有这个问题了，另外也可以通过手工拷贝镜像到计算节点方式来实现同样的功能。

1. **Next steps in live upgrade**

主要是讨论了nova的在线升级功能的历史、现状以及后续设计演进路线，由于conductor的加入，I版本开始部分支持从H版本在线升级到I版本，J版本及后续版本会尽力做到数据库及nova服务的在线升级（完全不停服），但目前需要做的工作还很多。

要注意的地方是，如果我们后续要使用在线升级功能，必须支持conductor，目前我们没有用（使用的是local方式，直接访问数据库），主要是有一些自研功能没有支持conductor，需要修改。

1. **Scheduler hints for VM life cycle**

BPs:

<https://blueprints.launchpad.net/nova/+spec/persist-scheduler-hints>

<https://blueprints.launchpad.net/nova/+spec/validate-targethost-live-migration>

nova-specs:[https://review.openstack.org/#/c/88983/](https://review.openstack.org/%23/c/88983/)

功能介绍：

主要是用来持久化调度参数，把本来仅在创建时生效的调度参数（scheduler hints）持久化保存到nova数据库，让这些参数在后续的各类需要选择目标节点的操作中也能继续生效（如resize、cold migration、live migration等），防止此类操作违反用户的最初选择目标节点的规则。

尤其是live migration，用户指定了目标节点后，只会用目标节点的nova-compute进行是否可以迁移的检查，并不会对调度参数进行二次验证，这就有可能选到一个不符合用户创建虚拟机时的规则要求的目标节点上，即使用户不指定目标节点，让调度器选择，也有相同的问题，创建时的scheduler hints也不会被用来选择目标节点。

这个功能会在nova show的时候显示出用户的scheduler hints，方便用户确认。

使用场景：

这个功能主要是为了解决虚拟机生命周期过程中，二次节点选择与创建时首次选择的规则不一致问题，这个问题目前来说对我们影响不大，我们的节点类型比较单一，并且resize等操作不能跨AZ。

但对于指定目标节点的各类迁移操作（如指定节点离线迁移功能、以及后续可能要支持的在线迁移功能）都有类似问题，我们目前对于这两个操作其实是可以跨AZ的。

总的来说对我们的影响不大，主要是rds会在创建虚拟机的时候用到很多的互斥规则，然后如果要迁移rds的虚拟机，是有可能违反创建的规则，但只要不跨AZ影响都不大，另外指定目标节点的操作都是管理员权限，管理员选节点的时候注意一下就可以了。

1. **Simultaneous Scheduling for Server Groups**

BP：<https://blueprints.launchpad.net/nova/+spec/simultaneous-server-group>

依赖BP：<https://blueprints.launchpad.net/nova/+spec/instance-group-api-extension>

nova-specs: [https://review.openstack.org/#/c/88080](https://review.openstack.org/%23/c/88080)

功能介绍：

实现虚拟机组的同时调度功能，具体来说就是如果用户创建的是一组虚拟机（需要虚拟机组功能的支持），那么调度器就会同时对这组虚拟机进行调度，看nova-specs里面的描述应该会把这一组的几台虚拟机调度到同一个物理节点上（同一组里面的虚拟机可以使用不同的镜像和规格），基本上就是当成一台虚拟机来调度。

使用场景：

假设有一批宿主机，其中几台在线（开机状态），某一台为了省电处于关机离线状态（上面没有虚拟机），在线状态的宿主机不能同时满足这一组虚拟机的全部所需资源，但是某几台宿主机可以满足这一组虚拟机所需资源，也即这组虚拟机如果使用目前的调度策略可能会被分散调度到不同宿主机上，但如果有了这个功能，就能保证他们被一起调度到关机那台宿主机上。（应该主要是为了性能考虑）

1. **Common no DB Scheduler**

BP：<https://blueprints.launchpad.net/nova/+spec/no-db-scheduler>

功能介绍：

1）调度器不再访问数据库，通过memcache方式来存放节点资源使用情况的数据

2）多个项目的调度器集中到一个项目，如nova\cinder等项目的scheduler集成到一个项目Gantt中

3）实现联合调度功能（用户可以创建虚拟机到计算资源和存储资源都充足的节点上）

使用场景：

联合调度功能对我们看起来有用，其实我们用了云硬盘，用处就不大了，另外对于大规模部署的环境，数据库访问可能是瓶颈，也可能有用，我们的规模较小，目前没看到有用到的可能。

另外该功能依赖的其他功能较多，估计J版本合入的可能性比较小。

1. **Future of Gantt APIs and interfaces**

nova调度器独立出来的新项目，后续同步api方式来执行调度流程，而不是目前的rpc方式，这个分享主要是讨论Gantt项目的api定义和功能列表。计划是J版本开始做这个项目，估计J版本不会替换目前的nova-scheduer。

wiki：<https://wiki.openstack.org/wiki/Gantt>

1. **Hypervisor power management**

BP：<https://blueprints.launchpad.net/nova/+spec/hypervisor-power-management>

功能介绍：

可以关掉没有虚拟机运行的宿主机，已达到省电的目的，并在需要的时候通过nova-scheduer发送上电请求给宿主机使其上电运行。

该功能需要增加特定的hypervisor driver来支持（硬件不同driver也不同）。

对我们用处不大，我们的宿主机都有很多虚拟机在运行，通常都是资源不够用，不会多余。

1. **Implementing state management for quotas**

功能介绍：

主要是为了解决nova的配额系统跟企业内部的财务系统的融合问题，比如企业里的私有云，有很多部门在用，需要考虑成本问题，某个部门要增加配额，要看该部门的预算等是否超支之类的考虑，还有就是要考虑整个云平台的容量和成本，也就是说用户的配额申请不一定能全部满足，但是可以慢慢的随着系统的扩容，资金的到位等逐步满足，这个功能就是改进nova的配额管理功能，让配额支持状态管理，比如soft配额、hard配额，soft是期望配额量，hard是当前实际能提供的配额量。

对我们来说是有一定的用处的，但是功能定位目前还不够清晰，后续开发起来也会比较耗时，到目前连BP页面都没有。

1. **limited conductor API**

功能介绍：

主要是限制计算节点访问conductor的权限，不能随便更新数据库甚至是帮其他计算节点更新数据库，也即每个计算节点只能更新他自己相关的数据库记录。目前的conductor是不校验更新申请的权限和来源等信息的。

这个议题只是讨论一下后续的解决方案，没有实际的开发动作。有一个关联的BP可以供我们参考，他们建议是先实现这个BP再根据问题情况决定下一步动作。

关联BP：<https://blueprints.launchpad.net/oslo.messaging/+spec/trusted-messaging>

目前看来这个功能对我们的用处不大。

1. **Flavor storage re-vamp**

功能介绍：

主要是解决虚拟机创建后flavor的存储问题，后续会把instances表里面的instance\_type\_id字段给去掉，目前是存储在system\_metadata表里面，但是不包含extra\_specs里面的信息，另外也不支持flavor的分组，无法支撑后续flavor相关功能的开发，需要讨论下一步的方案。

这个只是个初期讨论，还没有相关BP提出来，但对我们还是有用处和影响的，因为我们的ecu、ssd等信息都是存在extra\_specs里面，都跟flavor有关，需要注意这个问题的动向。

1. **Docker driver - features & testing**

介绍：

这个议题是讨论Docker的后续发展方向和测试方法，因为Docker已经从nova中移走（到stackforge的heat项目里面了，通过heat来调用Docker api），移走的原因是如果仅仅把Docker作为nova一个virt driver来看待，它的很多特性不能使用（Docker本身有很多api可用，但是nova不支持这些api）。

对我们来说lxc的支持看起来只有libvirt一种了，因为heat我们没用到。

1. **Improve performance of live migration on KVM**

调优kvm的live migration性能，但是没看到ppt，不知道具体的调优方法，要等待ppt放出再继续分析。

1. **Clustered hypervisor support in Nova**

nova对集群hypervisor支持的不好，目前是每个hypervisor都要运行一个nova-compute才能使用，但是对裸机节点还有其他集群类型的hypervisor（如Hyper-V、VMWare集群等），并不会每个节点都运行nova-compute，所以要讨论一下nova要怎么处理这种部署情况。

对我们的用处不大，后续如果我们要支持VMWare集群的话，会有用处。

1. **Data transfer service plug-in**

BP：<https://blueprints.launchpad.net/nova/+spec/data-transfer-service-plugin>

nova-specs：[https://review.openstack.org/#/c/87207/](https://review.openstack.org/%23/c/87207/)

用来跨机房传输数据的功能插件，计划支持FTP、BitTorrent协议。

对我们用处不大。

1. **Image Upload Plug-in**

BP：<https://blueprints.launchpad.net/nova/+spec/image-upload-module-plugin>

nova-specs：[https://review.openstack.org/#/c/84671/](https://review.openstack.org/%23/c/84671/)

类似上面的功能，直接通过FTP等协议上传镜像到glance，对我们用处不大。

1. **SR-IOV support**

网卡直通功能，neutron和nova都有改动，目前对我们用处不大，后续有需要再跟踪分析。

1. **Base feature requirements for compute drivers**

主要讨论compute drivers需要支持哪些基础功能，以及如何进行测试。

对我们的用处不大。

**Glance:**

1. **Glance Store and future features**

BP：<https://blueprints.launchpad.net/glance/+spec/create-store-package>

把镜像存储后端适配代码从glance库里面移出去，成为一个单独的项目，类似把nova里面的配置项管理相关代码独立出来形成oslo.config，以后nova直接访问镜像相关信息就可以使用这个项目进行访问了，应该是打快照时上传快照到镜像存储后端时会用到。

1. **Signed Images**

BP：<https://blueprints.launchpad.net/glance/+spec/signed-images>

Glance认证镜像支持，主要是用来在使用镜像前，检查镜像的合法性的，比如镜像是一个tar/tgz压缩包，解压缩之后可以拿到一个镜像文件和一个校验用的认证文件，之后用户或者镜像使用方（包括nova等）就可以根据这个认证文件来检查镜像的合法性。

**Keystone：**

1. **Create multi-attribute based endpoint groups**

BP：<https://blueprints.launchpad.net/keystone/+spec/multi-attribute-endpoint-grouping>

这个功能应该是为了创建endpoint group，支持CRDU操作，目前endpoint没有group的概念，有了组概念之后，就可以按创建各种类型的group，比如云主机服务可以按地域来归类形成一个endpoint group，各种admin服务也可以按地域来形成类似USAdmin, EuropeAdmin and AsiaAdmin groups，user和project也可以关联到一个endpoint group，关联之后就只能看到这个group的endpoint了。某些存储（volume）服务也可以和主机服务形成组（比如按照地域），之后主机服务就可以只看到组内的（本地的）存储服务的endpoint了。

1. **Discoverable and Hierarchical Catalog**

BP：<https://blueprints.launchpad.net/keystone/+spec/discoverable-hierarchical-catalog>

支持服务项的层级索引，用户可以方便的找到各种服务的endpoint。

1. **Identity IDs in a multi-backend or Federated world**

BP：<https://blueprints.launchpad.net/keystone/+spec/multi-backend-uuids>

keystone不同后端有不同的用户ID形式，如SQL后端是32位随机字符，LDAP是64位，还有其他类型，所以这个功能就是想统一一下ID形式，有两种方案，一个是建个mapping表，映射一下，另外一个是encode一下这些ID，以便统一形式。

1. **Password Policy and Lifecycle Management**

这个功能很有用，就是常见的密码生命周期管理功能，比如：密码过期功能、下次登陆必须修改密码功能、密码错误次数达到一定限制后锁定账号功能、找回密码功能（通过密码问题等）、密码复杂度检查功能、密码回滚功能（感觉应该是定期修改密码功能吧）。

但是还没有BP，代码更是没有，估计还要两三个版本才能发布。

1. **Federation within a private cloud**

支持多个部署环境的keystone之间互相验证token，比如有两个环境，各有一个keystone部署，并且使用不同的后端存储token，这两个keystone各自生成的token，可以互相交叉认证。感觉很符合我们的需求，另外有人评论说，token\_format=PKI已经可以做到这点，只要多个keystone都把这个用户的签名证书（signing certificate）导入即可。

1. **Running Keystone in different environments**

讨论如何在各种运行环境下运行keystone，比如java环境用Jython或者使用IronPython运行在.Net-based环境下，而不需要把keystone代码从python移植到java等其他编程语言。

1. [**VPC Modeling**](http://summit.openstack.org/cfp/details/421)

wiki：https://wiki.openstack.org/wiki/Blueprint-VPC

BP：https://blueprints.launchpad.net/keystone/+spec/vpc

openstack的VPC架构，主要是讨论keystone相关设计。

1. **Support for Virtual Organisations**

BP：<https://blueprints.launchpad.net/keystone/+spec/vo-management>

Keystone虚拟组织功能，多个不想改的users可以组成一个虚拟的组织。

1. **Future Enhancements to Keystone Federation Support**

Keystone的多后端支持。

1. **Keystone support for One Time Password ( OTP)**

Keystone一次性密码支持。

**Ceilometer:**

1. [**Metering Network Services**](http://summit.openstack.org/cfp/details/251)

BPs：<https://blueprints.launchpad.net/ceilometer/+spec/ceilometer-meter-lbaas>

<https://blueprints.launchpad.net/ceilometer/+spec/ceilometer-meter-vpnaas>

<https://blueprints.launchpad.net/ceilometer/+spec/ceilometer-meter-fwaas>

支持各种网络服务的监控，如vpn服务、lbs服务、firewall服务，对我们的监控计费可能会有参考价值。

**CInder：**

1. **What is a Cinder Driver**

主要讨论cinder是如何定义driver这个概念的，有些新增cinder后端需要增加一些cinder和第三方服务之间的代码才能支持。

1. **3rd Party Driver Certification / CI Systems**

跟上面的议题类似，也是讨论怎么能让Cinder支持更多的后端存储。