

# OpenStack云计算与云数据库方案



技术创新，变革未来

# 主要内容

- 云计算与OpenStack
- 云计算与云数据库
- OpenStack Trove介绍
- OpenStack Trove实践
- 与典型云数据库的对比
- 总结

# 云计算与云计算平台

- 云计算是一种**IT资源的运作与交付形式**，资源多个层次IaaS/PaaS/SaaS，最大特征是**极具弹性与按需而变**。
- 云计算与虚拟化关系：虚拟化不是云计算必须的（比如有些云计算平台也提供物理机资源），但是**虚拟化大大增强了云计算能力**，没有虚拟化的云计算正如没有钢筋的楼房。
- 云计算基础三大关键要素：**计算(CPU)、网络(系统总线)与存储(Mem&Disk)**，每个要素也都有多种实现及体系。
- 云计算平台是实现云计算的基础平台，如AWS，GCE，阿里云，腾讯云，RackSpace等等，往往**综合了IaaS和PaaS**，甚至包含SaaS。

# OpenStack生态

- **Openstack**是应用最广，兼容最好的开源云平台实现，制定了云平台整个框架和API标准，一开始以兼容AWS API为出发点，整体上高内聚松耦合。
- OpenStack之于云计算（对AWS的开源实现） <===> Hadoop之于大数据处理（对Google大存储/数据的开源实现）；
- 由OpenStack基金会管理，董事会+技术委员会；平台厂商+金牌会员基本覆盖了大部分著名的IT公司



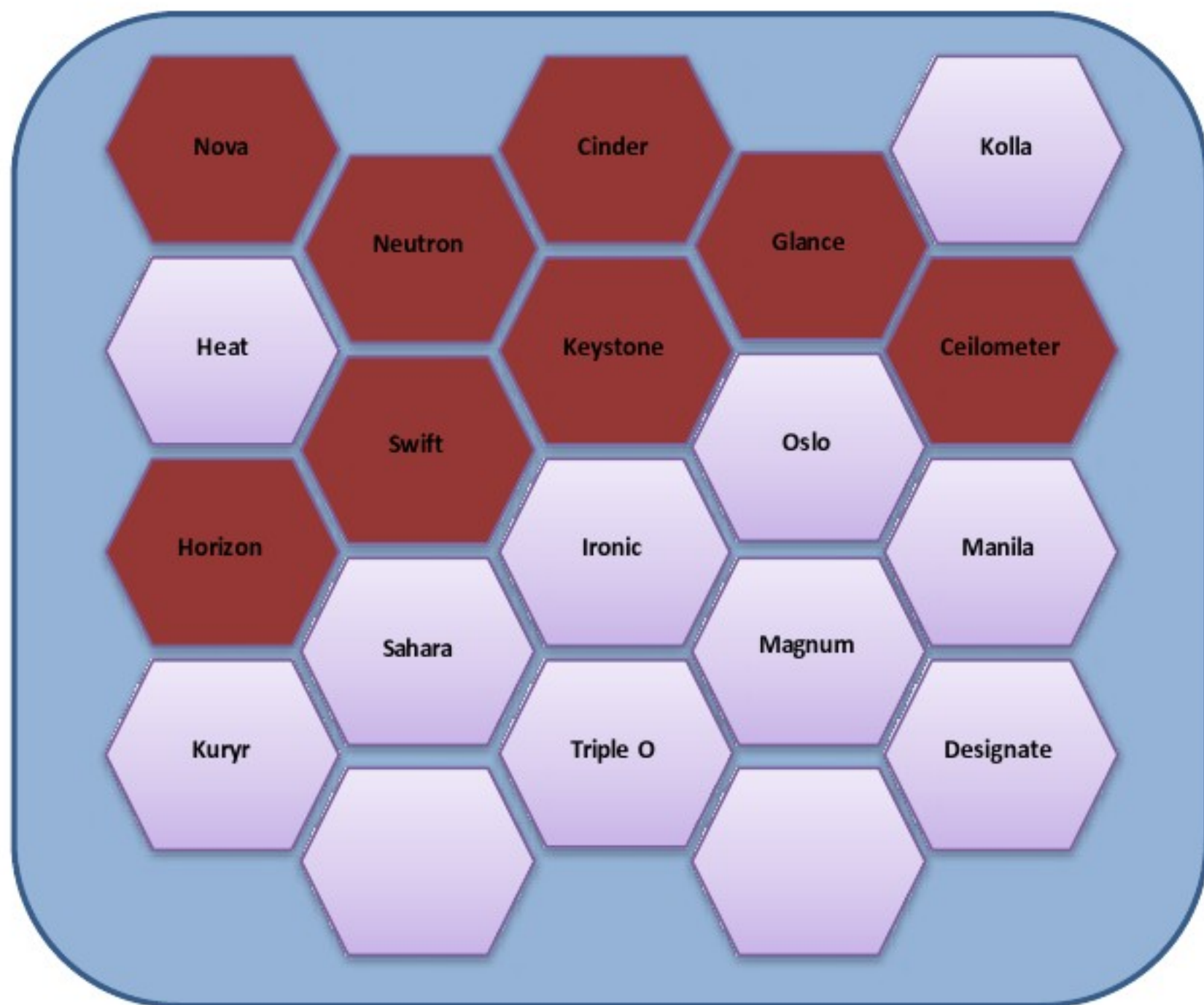
Rackspace

# OpenStack版本历史

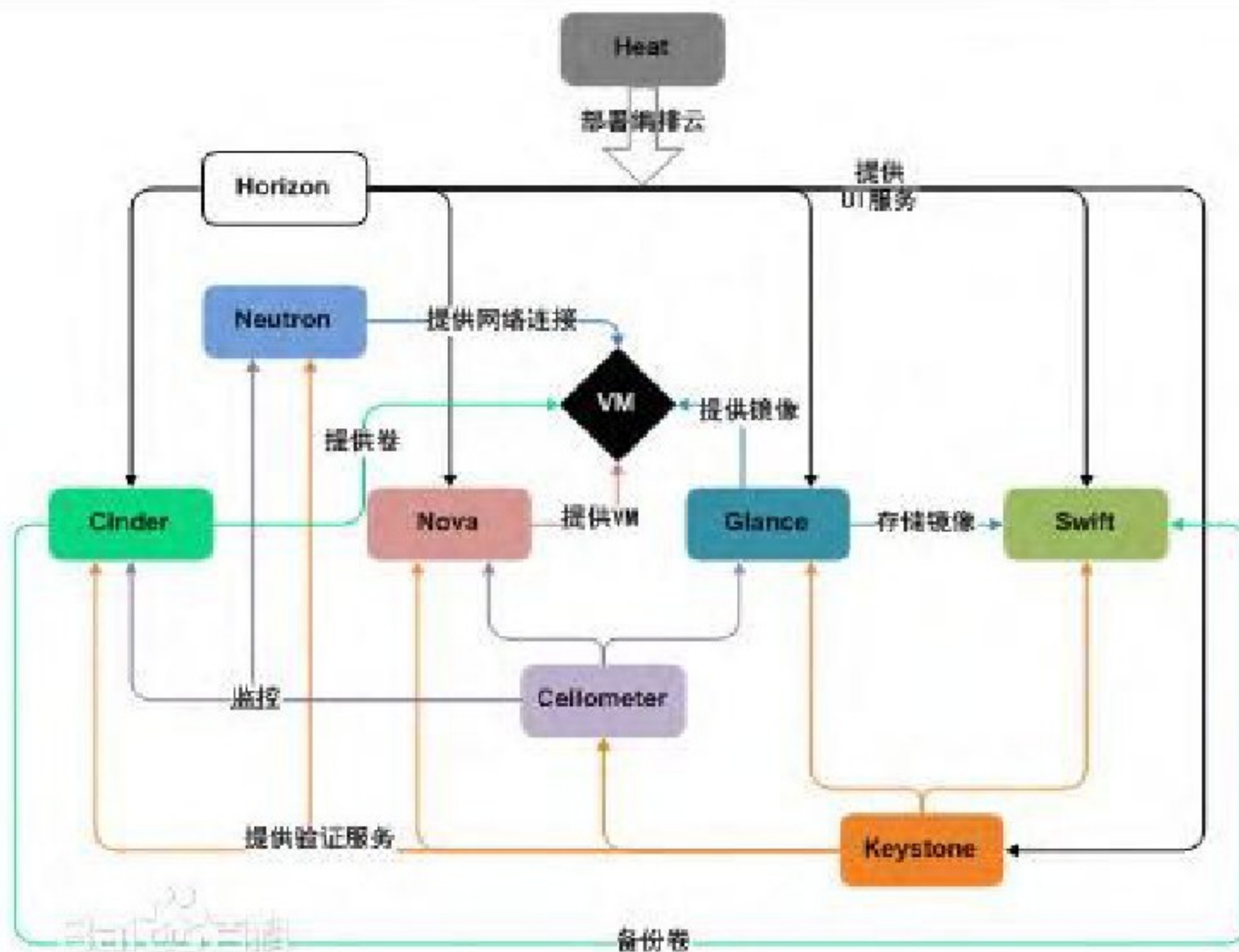
<b>Austin</b>	<b>October 2010</b>	<b>Nova, Swift</b>	<b>Grizzly</b>	<b>April 2013</b>	-
<b>Bexar</b>	February 2011	+Glance	<b>Havana</b>	October 2013	+Heat, Ceilometer
<b>Cactus</b>	April 2011	-	<b>Icehouse</b>	April 2014	<b>+Trove (数据库管理)</b>
<b>Diablo</b>	September 2011	-	<b>Juno</b>	October 2014	+Sahara (大数据管理)
<b>Essex</b>	April 2012	+Horizon, Keystone	<b>Kilo</b>	April 2015	+Ironic (物理机管理)
<b>Folsom</b>	September 2012	+Quantum, Cinder	<b>Liberty</b>	October 2015	+Zaqar, Manila, Designate, Barbican, Searchlight

以字母序为版本命名，每半年发布一个版本，每个版本几乎都有新服务。

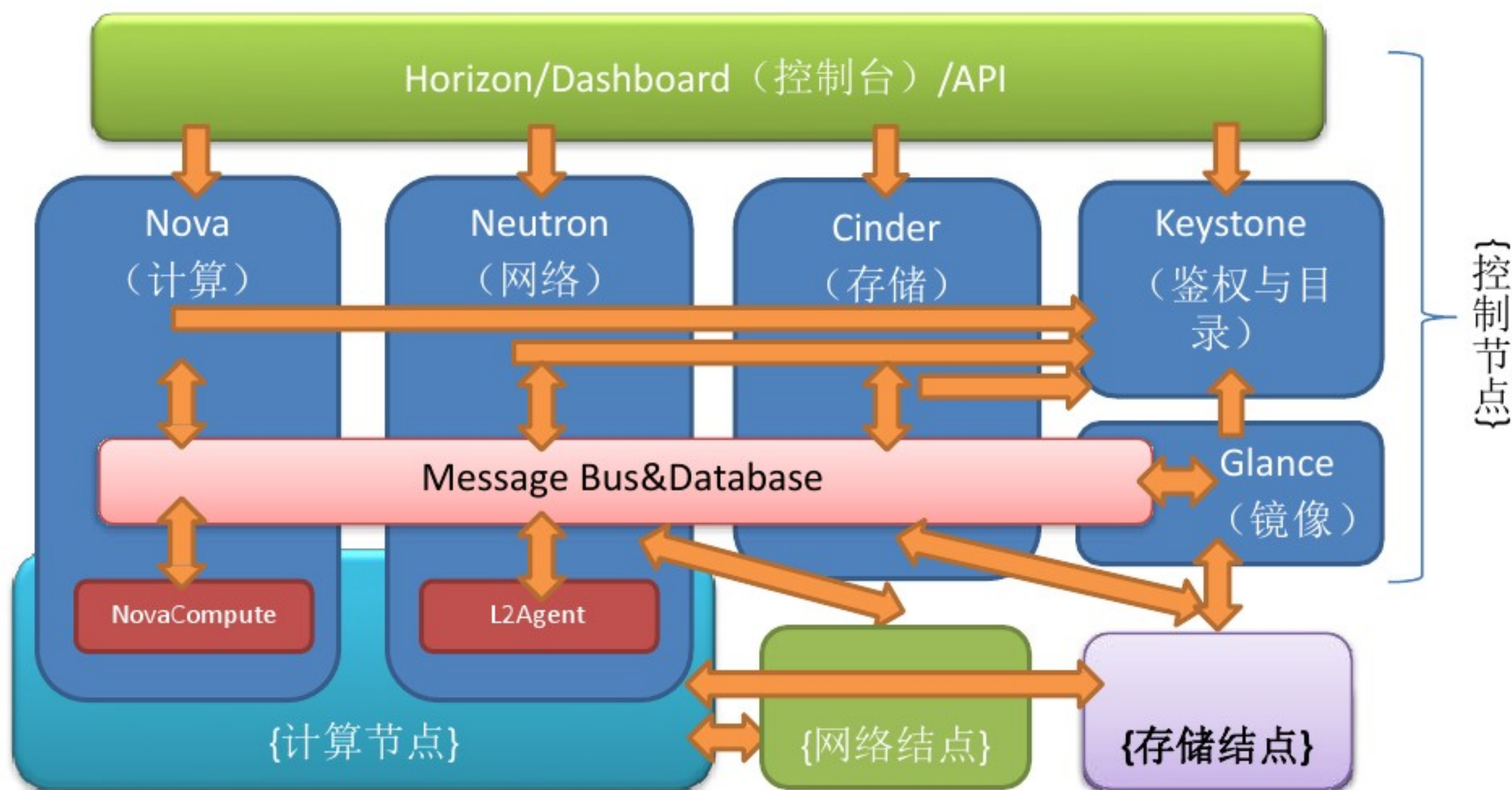
# 2016.4-Mitaka版本项目预览



# Openstack核心流程架构



# OpenStack核心架构





# Openstack的设计特征

- **大量采用成熟的开源组件**
  - kvm/qemu、openvswitch、ceph。。。
  - mysql、rabbitmq、memcached、haproxy。。。
- **功能强大，扩展性强**
  - 除了几个核心项目，不断加入新项目实现其他服务场景（数据库、大数据、DNS、容器、消息队列。。。）
  - 大部分项目都采用插件形式来实现底层的细节，具备很强的扩展性和定制性
- **清晰的服务栈，避免重复造轮子**
  - 共享的消息队列，鉴权服务，虚拟化服务
  - 服务堆叠，快速实现上层服务

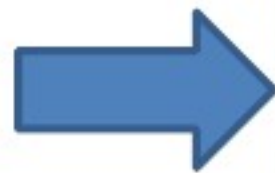
# Openstack的产品化

- Openstack跟大多数的商业方案一样，解决的问题基本类似的，在大道理上往往是**殊途同归**，细节处理上**有得有失**。
- Openstack从开始**更加规范，更加统一**，开源的好处在于集成**大众智慧**，且更快发展。
- 正如Linux内核与Linux操作系统的关系一样，只有内核是不够的，需要有完整的用户态环境，这样才能真正运行业务。
- Openstack**拉低了核心技术门槛**，但是**产品化的要求却没有降低**，随着用户需求的增长反而更高。



# 数据库与云平台

- 在云计算中数据库是更上层的资源，而且是大部分**应用的核心资源**
- 数据库与数据库管理的关系比虚拟机与虚拟机管理的关系**更贴近应用**
- 数据库管理的发展路径：手工管理->托管平台->云数据库平台



# 云数据库应具备的能力

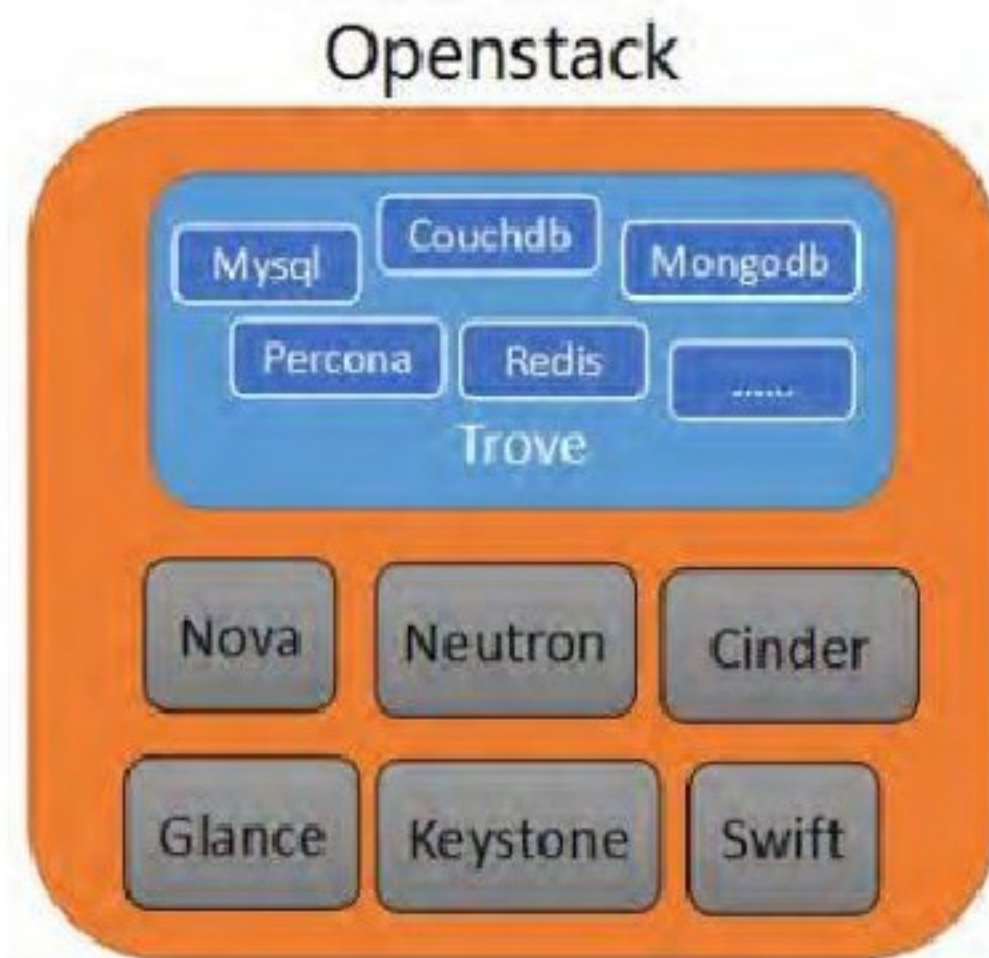
- **受控易用**：基本的管控能力与配置变更能力
- **弹性伸缩**：性能Scale Up/Down，规模Scale Out/in
- **高可用**：数据冷/热备，分布式集群，网络健壮性
- **多租户**：资源有效隔离，细粒度计量
- **资源有效利用**：充分利用和均衡各种软硬件资源，避免资源闲置与过度使用，良好的成本性价比
- **扩展与开放**：支持不同的数据库引擎，具备较完整的二次开发接口调用，便于集成

# Openstack的云数据库

- 自Icehouse版本推出了Trove项目
  - 实现DBaaS，解决用户使用与管理DB遇到的复杂繁琐问题
  - 依托OpenStack核心，提供对多种流行开源数据库的支持
  - **Trove本身就是Openstack云平台运用的一个绝佳的案例**
- 经历I、J、K、L四个版本，功能越来越强，支持的数据库类型越来越多

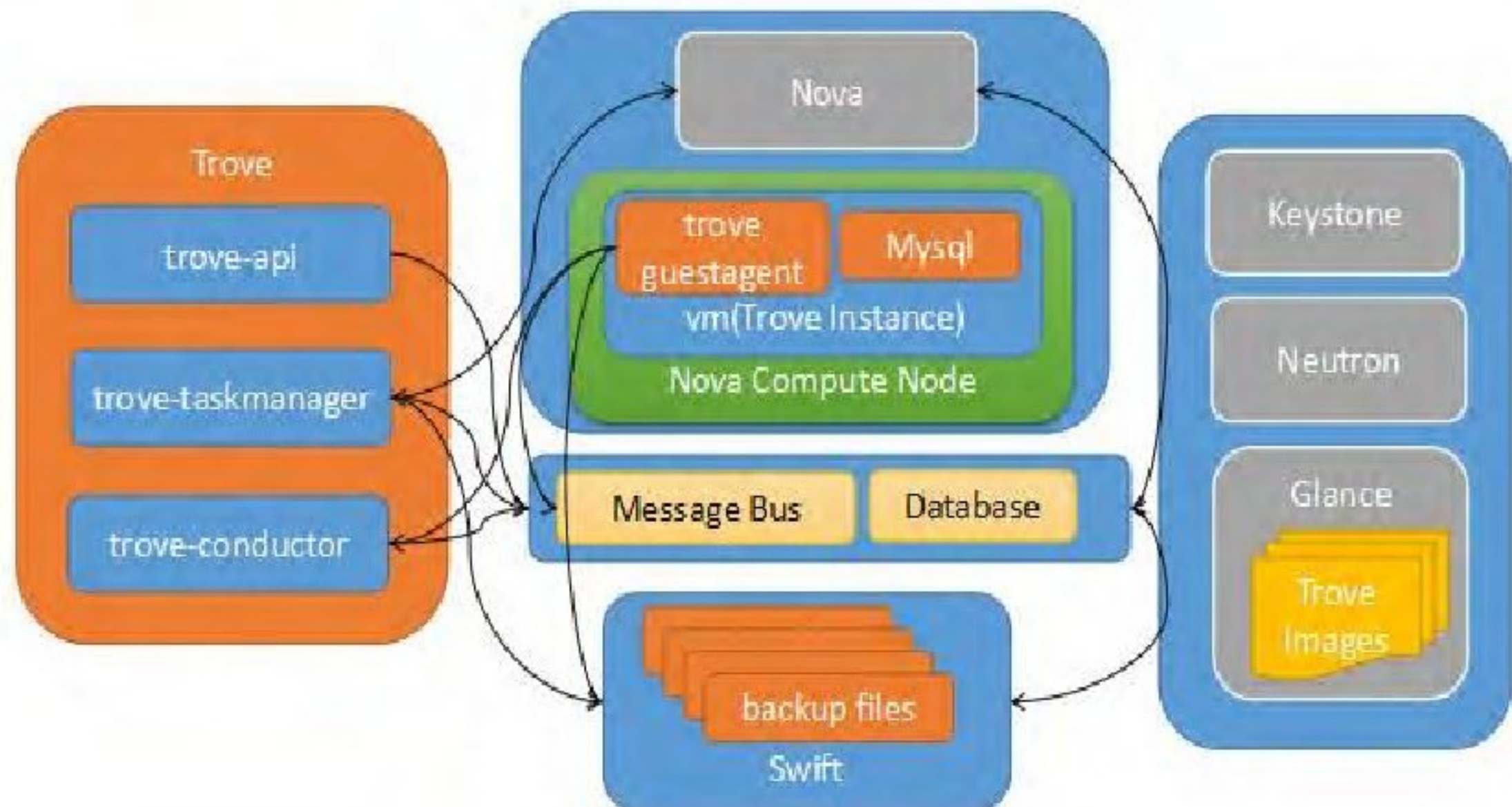
# Trove设计目标

- 完全运行在OpenStack之上
  - 基于OpenStack的核心服务之上构建
  - 遵从OpenStack框架与API规范
  - Trove自身属于OpenStack体系
- 支持关系与非关系型数据库
  - 进行更高层次的抽象，减少重复造轮子
  - 支持mysql, redis, mongodb等等



# Trove整体架构

- Trove自身的架构组织是非常薄的；充分利用Openstack各服务
- VM实例 就是 数据库实例；不同数据库有不同的VM镜像定制

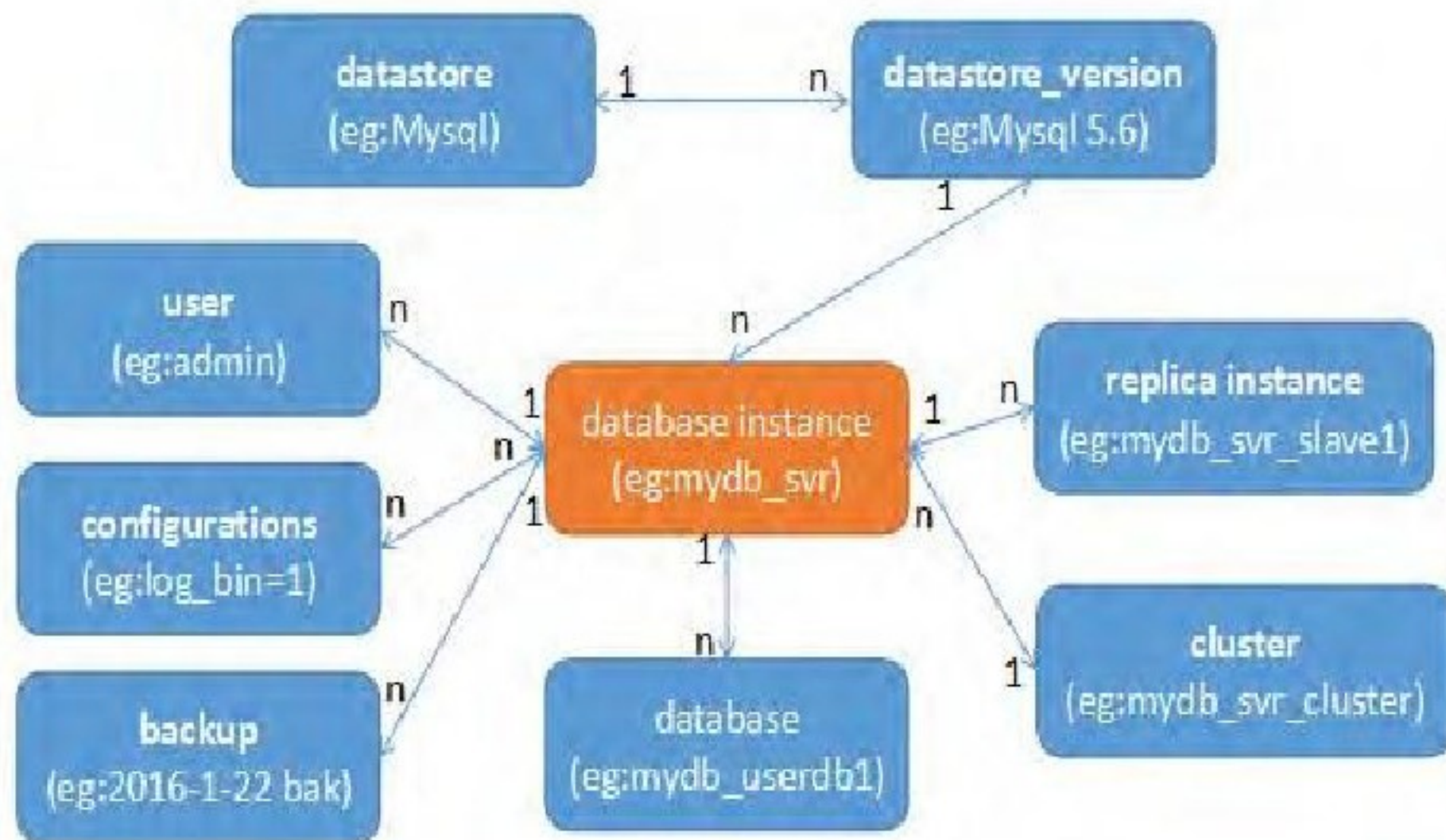


# Trove概念与功能

- 核心概念

- 主要功能

- 实例管理
- 数据库管理
- 主从管理
- 备份管理
- 配置管理
- 集群管理





# Trove实践1

- 平台环境建立
  - 包含Trove的OpenStack环境
  - Trove特定的VM镜像
  - 采用Trove-Integration项目完成以上两个工作
- Trove-Integration构建
  - 详细参考<https://wiki.OpenStack.org/wiki/Trove/trove-integration>
  - 注意的地方：一定需要先构建trove专用的vm镜像并导入glance
  - Trove也提供了强大的控制台trove命令行工具

# Trove实践2

## 建立mysql一主二从的实例运行环境

### 1) 创建主实例

命令：`trove create my_inst_master 8 --size 10 --database my_inst_db --users admin:admin123 --datastore mysql --datastore_version 5.6`

解释：实例名字为my\_inst\_master，实例规格ID是8（512MB内存），硬盘卷大小是10GB，并且创建数据库my\_inst\_db和用户admin（密码是admin123），数据库引擎类型是mysql，版本是5.6版本。

### 2) 创建主实例的备份（先随便在创建的数据库实例里创建表和插入一些数据）

命令：`trove backup-create my_inst_master my_bak.0001`

解释：创建数据库实例my\_inst\_master的当前的备份，备份名字为my\_bak.0001。

# Trove实践3

## 3) 从主实例的备份创建两个从实例，并且建立主从关系

命令：`trove create my_inst_slave 8 --size 10 --backup my_bak.0001 --replica_of my_inst_master--replica_count 2`

解释：创建两个数据库实例，名字以mysql\_inst\_slave开头，实例规格ID为8，硬盘卷大小是10GB，并且用备份名为my\_bak.0001的备份导入数据，且建立到实例my\_inst\_master的主从复制关系。

## 4) 动态Resize主实例规格

命令：`trove resize-instance my_inst_master 2`

解释：动态调整实例my\_inst\_master的instance规格为2（内存2GB大小）

命令：`trove resize-volume my_inst_master 20`

解释：动态调整实例my\_inst\_master的硬盘卷大小为20GB

# 与典型云数据库的对比1

数据库服务对比	计算资源	存储资源	实例部署	实例伸缩能力	支持数据库类型
OpenStack Trove	虚拟机	网络块设备	实例与虚拟机一一对应，基于成熟的虚拟机平台进行隔离与调度	采用云平台成熟的虚拟机和网络卷技术动态调整实例规格	支持主流关系型和非关系型数据库，接口统一
典型云平台 DBaaS服务	物理机	本地磁盘	多实例共享单个物理机，基于cgroups做资源隔离	主要通过迁移实现规格调整，效率不高	主要支持mysql或postgresql或redis，接口各异
Trove的优势	搭建非常灵活，而且完全基于云平台，充分利用共享资源	网络存储成本较低，天然超卖；容灾能力好；随机大并发吞吐量高	隔离性好，环境一致性强，与用户的虚拟机同网络同子网部署。	可靠成熟，效率较高	丰富的数据库类型，扩展性强，方便迁移
Trove的劣势	虚拟机性能有一定的减损	有一定的网络传输延迟，对网络的可靠性依赖较高	实例数越多虚拟机越多，会增大平台运营压力；另外小实例过多也不经济。		每种数据库类型的支持不够细化

# 与典型云数据库的对比2

数据库服务对比	主从HA支持	分布式集群支持	数据备份与回档	配置更新能力	监控统计
OpenStack Trove	暂时不支持	支持 Percona, Mariadb, Mongodb 等分布式集群	对于mysql类数据库, 使用xtrabackup的物理备份(全量和增量); mongodb则是dump出数据文件然后再拷贝; 基于卷的快照备份也是一种方式	具备广泛的动态更新配置的能力	暂时没有
典型云平台 DBaaS服务	通过LVS类似的接入代理支持主从透明切换	暂时不支持	对于mysql类数据库, 使用dump方式的逻辑备份或xtrabackup的物理备份	具备一定的配置管理能力	有比较好的监控统计, 并配合HA容灾
Trove的优势		依赖底层数据库类型的集群技术支持	除了各种数据库类型自身的备份方式外, 还能够基于卷快照做备份, 速度很快		
Trove的劣势	要尽量避免主从实例所在的虚拟机在同一个宿主主机上		基于卷快照备份无法恢复到任意时间点。		当实例发生故障的时候, 需要自己判断并进行切换

# 总结

- Trove的应用情况介绍

- 功能基本够用，稳定性依赖OpenStack基础服务
- ebay、Rackspace、HP等公司已经大规模使用
- 有27%的OpenStack用户在生产环境使用Trove

- Trove的当前不足

- 统计与监控能力
- 自动容灾能力

- 未来M和N两个版本的计划

- New UI panel added to Horizon
- New datastore management API for operators

### Mitaka (1 specs/blueprints as of 10/13/2015)

- Self-healing cluster for MySQL
- Cassandra clusters and enhancements

### "N" Release

- TBD, themes are not prevalent yet either

# 参考资料

- OpenStack

<http://www.OpenStack.org/>

- Trove

<http://www.OpenStack.org/software/releases/liberty/components/trove>

- Trove-Integration

<https://github.com/OpenStack/trove-integration>

- Building Guest Images for OpenStack Trove

[http://docs.OpenStack.org/developer/trove/dev/building\\_guest\\_images.html](http://docs.OpenStack.org/developer/trove/dev/building_guest_images.html)