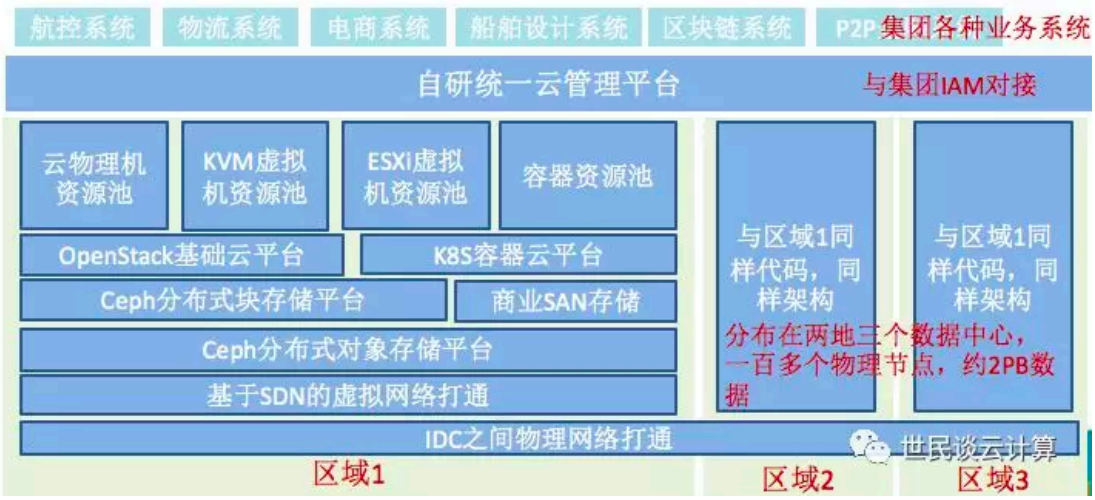
OpenStack's eight-year itch

Original:Sammy Liu

In October 2010, OpenStack released the first version. Last month, it released its 18th version of Rocky. Not as many posts about the new releases as before, there are few posts only. After the announcement of the Rocky version, several short translated articles appeared in the OpenStack group. From time to time, the circle is floating out. "Is OpenStack dead? "Who is the one who replaced all OpenStack with Kubernetes" message. Why is this a turning point in just a few years? As an OpenStack user, in this article, I will reflect from the user's perspective on how it has gone in the past eight years. I will also try to predict OpenStack's status in eight years, good or not, or even will have died.

What kind of user are we?

As part of the HH Group Cloud Platform team, we built the basic cloud platform as shown below in the group:



Its main features are as follows:

- Compute: Supports three resource pools such as KVM, ESXi, and bare metal servers.
- Networking: Virtual network is implemented with Neutron + VLAN + OVS.
- Storage: Block storage is implemented using Ceph and SAN, and object storage is implemented using Ceph.

- Region: Three regions are deployed in three computer rooms in two cities. Each region is divided into resource pools, and the resource pools are divided into available zones by rack. All three levels are visible to the user and can be selected as needed. In addition, we also tried to engage in a small public cloud region.
- Functions: Utilize components such as Glance, Nova, Neutron, Cinder, Keystone, Heat, Telemetry, OVSvAPP, Trove, Ironic in the Mitaka.
- Management: Manage multiple regions with self-developed cloud management platform.
- Container Cloud Platform: The container cloud platform based on Kubernetes runs on a physical machine that it manages.
- Team: Up to 8 people in the OpenStack R&D team and 3 people in the operation and maintenance team.

Some feelings:

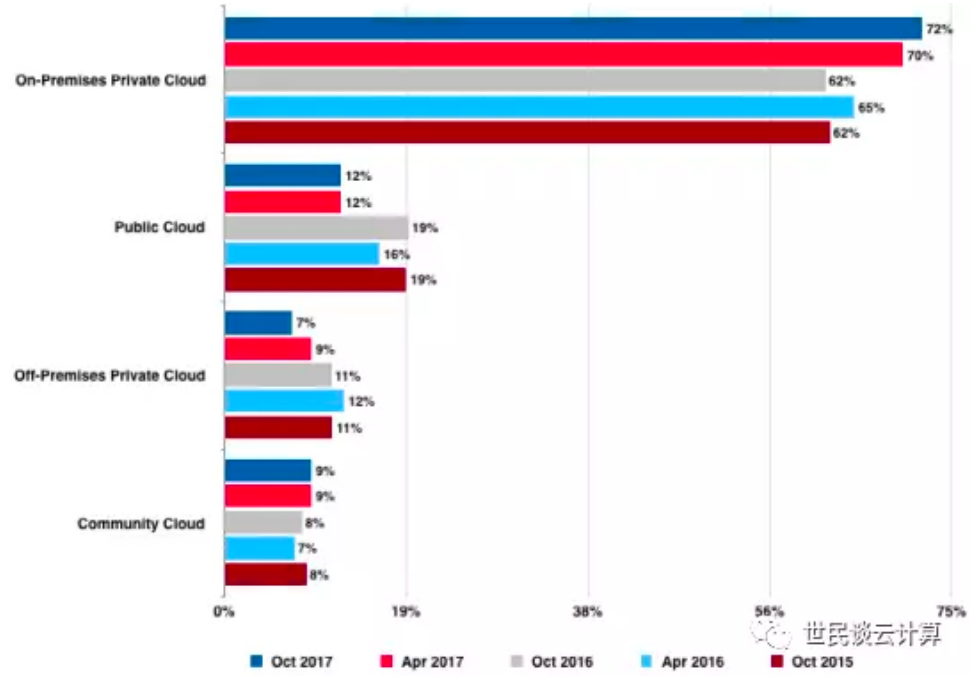
- In general, the operation is quite good. We have done a good job in technology and product selection, R&D, operation and maintenance, etc. The team is very strong, the development cycle is short, and the iteration is fast. Now it supports hundreds of systems of large and small groups, and it is very stable, and the operational pressure is relatively small. Here, I would also like to thank the little friends who fought side by side.
- There have also been some stability issues: For example, Neutron VR occasionally switches automatically (we also have a small public cloud environment, using Neutron + VR + OVS architecture); KVM virtual machine occasionally automatically restarts or even downtime; KVM support for Windows Poor, occasionally inexplicable problems, such as disk offline, blue screen, unable to start.
- The monitoring components and log components are not complete, and we need to make big changes or build them from scratch.
- In addition to the core modules, the other modules are almost half-pull projects. Taking Trove as an example, we spent a lot of

time rewriting almost half of the code, which enabled the creation and management of the most basic database instances.

- The gap between OpenStack and public cloud demand is too far.

What is the positioning and benchmarking of OpenStack?

The original mission of the OpenStack community in 2010 was to "provide an open source cloud computing platform that meets the needs of public and private clouds." At that time, the private cloud had no reference, so it can be considered that the earliest mission of OpenStack was to do another AWS in open source. This is really a grand goal, how exciting it is, and even the hearts of VMware and AWS have been layered! However, from the user survey results from 2014, OpenStack can't do public cloud, private cloud is the main battlefield of OpenStack, because the two private cloud environments add up to 80%, and the ratio of public cloud is only 12% in 2017. And it is constantly shrinking. Therefore, the actual positioning of OpenStack is in the private cloud, which is beyond doubt.



In the enterprise private cloud environment, VMware is the real boss. Therefore, OpenStack is going to be the goal of a private

cloud, saying that it is good to learn from VMware; to say that it is hard to replace VMware. VMware vSphere provides only a virtualized environment, so OpenStack's target object should be "VMware's virtualization function" + "AWS's Cloud function, mainly cloud API". However, because the OpenStack target was AWS at the beginning, and AWS is a public cloud, this led to many problems later, which will be detailed later in the article.

In the two parts of "VMware virtualization" + "AWS Cloud function", because OpenStack is the benchmark AWS at the beginning, the "Cloud" part should be said to be very good, or cloned. This is from the user survey "Why does the organization choose OpenStack? Part of the answer can also be seen, that is, the standardization of open platforms and APIs is the first business driver.

What about the "VMware virtualization" benchmarking part? Take a look at the comparison of the basic features of VMware vSphere and OpenStack:

VMware features	Description	The corresponding OpenStack function
vMotion	Live migration of a running virtual machine from one physical server to another can be achieved with zero downtime and continuously available services. vSphere 6.0 supports vMotion across data centers.	Virtual live migration can be achieved with the KVM live migration feature, but with third-party tools.
DRS (Distributed Resource Scheduling)	Cross-resource pools continuously monitor utilization and intelligently allocate available resources across multiple virtual machines based on predefined rules that reflect business needs and chan	not support.

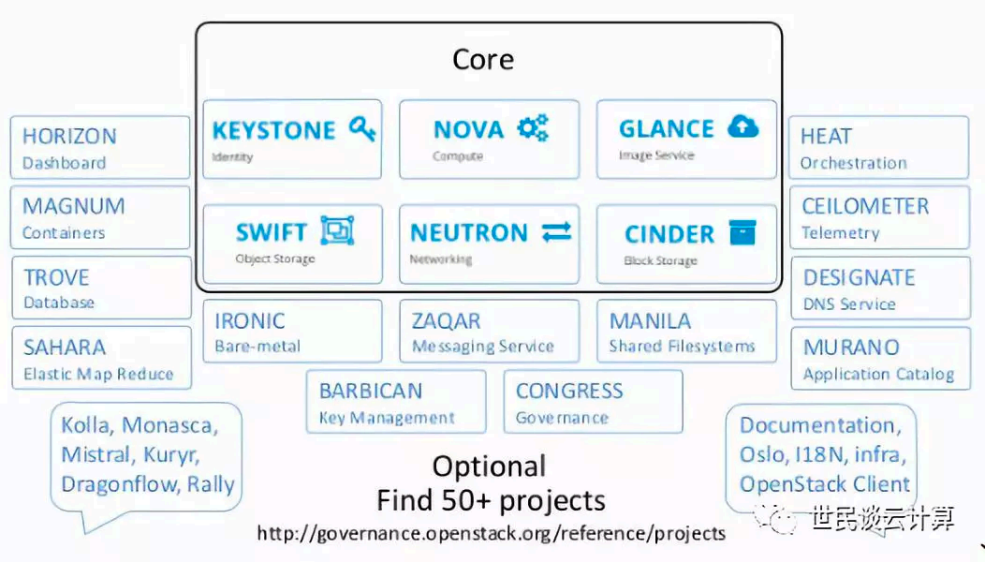
	ging priorities.	
Distributed Power Management (DPM)	DPM provides the goal of saving power by dynamically adjusting cluster capacity to match virtual machine resource requirements. DPM automatically consolidates virtual machines onto fewer ESXi hosts and performs more resource utilization than ESXi hosts in a given period. Power outage, if resource requirements increase, the ESXi host powers back to the cluster and the virtual machine is reassigned to all available ESXI hosts in the cluster.	not support.
HA	Continuously monitor all physical servers in the resource pool and restart virtual machines affected by server failures. You can also monitor and detect virtual machine guest operating system failures and automatically power on virtual machines after user-specified intervals	not support.
FT	Provide continuous availability to virtual machines by creating and maintaining a secondary virtual machine that is equivalent to the primary virtual machine and can replace the primary virtual machine in the event of a failover	not support.
vShield	VMware Secure Virtual Appliance Suite	Neutron's security groups and firewalls implement some of vShield's features
vDS (distributed virtual switch)	Simplify virtual machine network connectivity by allowing users to set up virtual machine access exchanges for the entire data center from a centralized interface.	Neutron implements some functions with OVS
Storage API		Cinder

SRM	Site disaster recovery	There is a Freezer project, but it is not enough to enter the production environment.

As you can see from the above table, most of the vSphere features OpenStack are not implemented, or only a little. The only result is that OpenStack does not have the ability to replace VMware, and it can't drive users to abandon VMware and switch to OpenStack.

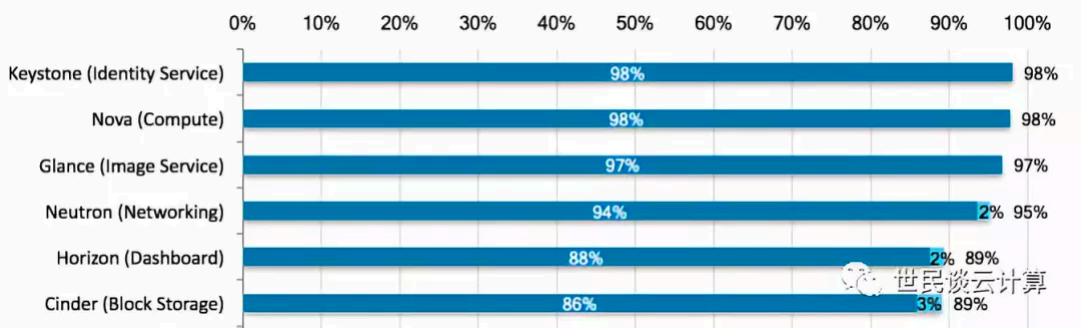
Where is the problem with the big tent mode?

In 2015, the OpenStack community began using the "big tent" model. This model divides OpenStack projects into two categories: core projects and non-core projects. There are only six core projects, and the rest are non-core projects.



According to personal understanding, I simply explain some of the problems of this figure:

1. The six core services have developed really well, but the problems are still quite a lot. On the one hand, as in the April 2017 user survey results, the usage rate of the first few core projects exceeded 90%. On the other hand, users have not stopped complaining the core projects, and there are several pages in the user survey report that record the user's complaining.



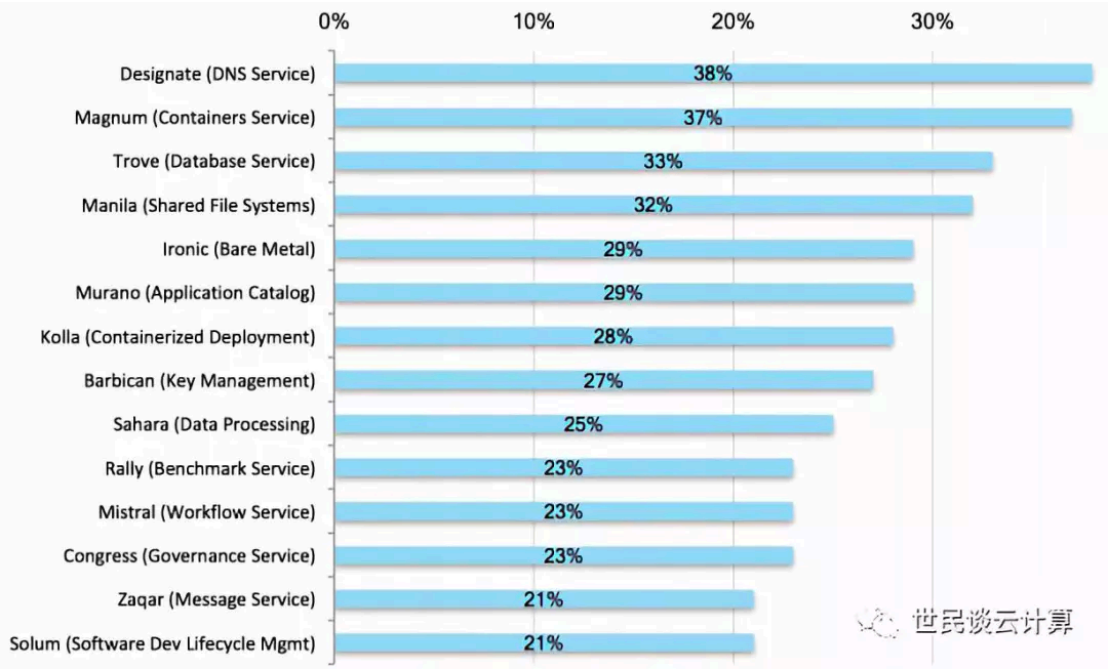
2. Comparing either VMware or AWS, the scope of OpenStack core services is too small, resulting in the lack of some of the mandatory features. I think at least the following services need to go to the core service list:

- Orchestration service Heat: Orchestration service is one of the basic services of the cloud. As a result, users can create and destroy cloud resources themselves through orchestration services. Second, many secondary services can be provided to users by providing orchestration templates. Thirdly, they can interface with third-party cloud management platforms and tools to foster their ecology.
- Ceilometer: A cloud production environment is inseparable from a strong monitoring service. So far, the Ceilometer project has been problematic, such as scale issues, performance issues, and feature coverage issues.
- Bare-metal service Ironic: Bare-metal machines have many application scenarios in the private cloud, such as running databases, big data platforms, container platforms, and so on. If OpenStack does the Ironic, it will be a big advantage over VMware, and it will be a support platform for applications that need to use bare metal. The current Ironic project is too heavy and too complex, too deeply associated with physical network devices. However, if you can be as flexible and lightweight as Linux's kickstart and cobbler, this process, such as vmware physical machine can deploy ESXI in batches, then ESXI management, you can use all the services in VC, such The process is more reasonable.

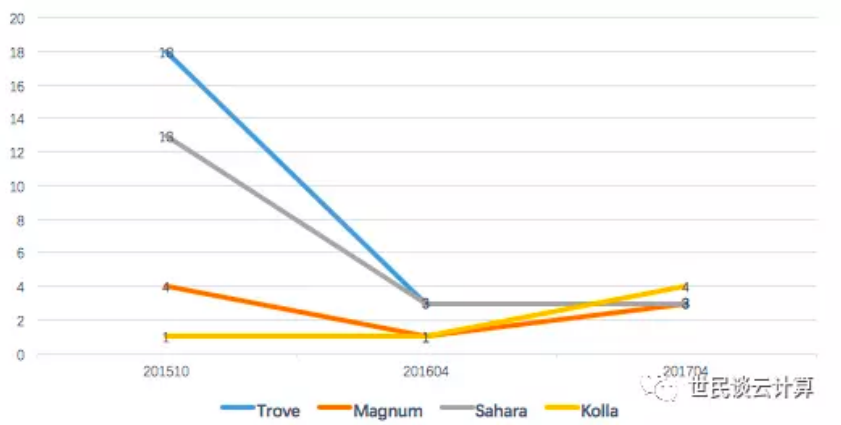
- Log service: Like the monitoring service, the log service is also a basic service of the cloud platform, just like AWS's CloudWatch and all projects are open.Unfortunately, until now, OpenStack does not have a native log service project.
- Deploying services: Deployment is important to private clouds. OpenStack needs a service that provides a graphical one-click deployment tool like Mirantis Fuel.

3. The OpenStack community has spent too much effort on services that look promising but are actually more sloppy, such as the container service Magnum, the big data service Sahara, the database service Trove, and the containerized deployment service Kolla. Ok, I know that you may have different opinions. I don't want to argue. Let’s look at the data in the user survey report.

- On the one hand, users are very interested in these projects. I think there are at least three reasons. One is that people are curious about new things, the other is the vigorous publicity of the OpenStack community, and the third is the ardent expectation. The following data is from the 201704 User Survey Report:



- But the deployment of these services in real-world production environments is very rare and less frequent:



(Note: the numbers in the figure are percentages)

- So what is the reason for these new services to be called? I think there are several reasons:

  (1) Differences in demand for cloud platforms between private and public clouds.

  The following picture is a private cloud environment that I think is typical:

  It has several features:

- Only the underlying physical machine management system is unified, and the above multiple platforms are separate. On the public cloud, the cloud platform is unified.
- The platform is separate. There may be several reasons for this. First, management factors, each platform is often managed and used by different departments; second, the operation and maintenance factors, put the platforms together, the operation and maintenance team can not operate the operation and maintenance of this single platform, It must be divided and ruled; the third is the

technical factor. The private cloud field has not yet seen a unified cloud platform like AWS and AliCloud that can manage these platforms together. Fourth, it is limited to the need for security and security in some enterprises. A large business needs to monopolize the resource pool/dedicated resource pool.

- In addition to the basic cloud platform is to achieve multi-tenancy outdoor at the virtual machine level, other platforms often only achieve multi-tenancy at the management platform level, or the business level itself implements multi-tenancy, and the following is one or several large resource pools.

  In a private cloud environment and a public cloud environment, the creation and management of these services (which should be called application services separately from the fundamental services) are very different. In a public cloud environment, because of multi-tenant needs, cloud providers need to provide the creation and management services for these services, allowing users to create, manage, and destroy these environments themselves. However, there are not so many requirements in a private cloud, and it is necessary to repeatedly create and destroy the operating environment of these services. Therefore, the need to implement automated creation and destruction of container platforms and big data platforms in OpenStack is less intense and can even be considered a pseudo-demand(fake requirement). For these new applications, OpenStack's mission should first be to "run well" on its own platform, rather than "create the runtime environment."

  The reason, I think this is related to the mission of the early OpenStack, because at the beginning OpenStack is to make an open source AWS, OpenStack service will grow like AWS'. The problem is that OpenStack has not paid much attention to the difference between private and public clouds, or has not been able to pay attention to it, because it is relatively easy to implement a set of OpenStack in accordance with the various services of

AWS. Moreover, when OpenStack is in full swing, it is a glorious thing to be able to open a new project. The PR draft will be much better.

Then why shouldn’t you waste so much time on these projects, or should the community not take the wrong direction?

- Or the positioning of OpenStack is not clearly and timely corrected. What should OpenStack do in the face of these emerging new applications? Is it a mind to do your own one-acre three-point land, while satisfying their needs for themselves, to achieve good support for them, or to insert a leg anyway? I think the original should be chosen, but the community actually chose the latter.
- The native deployment tools for these applications are better. The corresponding projects on OpenStack have not been able to create and manage the environment of these applications from the very beginning. With the release of new versions of these applications, the gap will only become larger and larger, and in the end, only some will be left unmaintained. There is no user's half pull item.
- These projects in the OpenStack community are immature projects that cannot enter the production environment, and the cost of the changes is quite high. Take Trove as an example. After modifying almost half of the code, we will implement the basic database instance creation and management functions, which is still far from the actual production requirements.
- OpenStack's learning of AWS only stays on the surface of "shape" and does not learn "God". Despite having more than a hundred services on AWS, what we see is that AWS is doing its basics in a down-to-earth manner. Let me give you a few examples. Blockchain is very hot right now, AWS currently only provides CloudFormation templates for users to orchestrate the cloud resources running the blockchain; Kubernetes is also very

hot now, but AWS does not even manage the K8S cluster interface. provide.

What kind of attitudes and directions does OpenStack have for these new applications? I think it should be two points:

- Focus on fundamental/core projects. The running infrastructure environment of these new applications is made, so that these services can run well in virtual machines/physical machines, networks and storage managed by OpenStack.
- Optimize Heat service, and provide a good template as AWS does. When the user needs it, the administrator uses these templates to excel the environment and then hand it over to ordinary users.

Why does OpenStack have a middle-aged crisis in youth?

I think there are several reasons for this. Of course, those are part of reasons.

(1) The appearance of containers has a great impact on OpenStack. However, we also have to see that the emergence of containers does not make VMware and IaaS cloud service providers represented by AWS a bitter. What OpenStack should think about why OpenStack failed to do the underlying architecture of the container.

- Take AWS as an example. It has two container-related projects, one is its self-developed ECS , which is a Docker container management service, and the container runs on the EC2 host. The other is EKS , a creation and management service for the Kubernetes runtime environment. AWS has done a few things to support the container:

  1. Created the amazon-ecs-cni-plugin project, making the container work well in the VPC.
  2. Open user permissions, users canlog in to

the Kubernetes environment using the AWS account.

3. Implemented a set of Docker container management services, as well as K8S management nodes.

- Let’s see OpenStack's support for containers, it mainly does a few things. One is to make a big deal with the Magnum project, and it takes a lot of effort to make the K8S environment. The other is that there are several network related projects, but it seems that no one is using them.
- As a result, in the OpenStack environment, the K8S environment is not well programmed (of course, do not want to create and manage K8S clusters in the private cloud, discussed earlier), K8S also does not work well in the OpenStack environment (Because the network and storage for the K8S are not supported well.) Therefore, I believe that OpenStack did not support K8S in time, which led to the separation of K8S and OpenStack .

(2) The community did not plan and manage the development direction of OpenStack, and wasted valuable time and resources in the key development stage. As mentioned earlier, the OpenStack community has not been able to position itself and focus on the underlying core services to make the bottom solid. On the contrary, like a young boy, when he was young, he didn't study hard and was absorbed by the outside world. He did not do business all the time. When he was an adult, he found that he could not cultivate his basic competitiveness. In addition, when the problem occurred, the community failed to turn the tide and failed to correct the development direction in time.

(3) Some OpenStack startups are too impetuous to do very good product development and service. At the peak, some startups are pursuing the contribution of the community, regardless of the quality of the contribution, or even game the statistics; the pursuit of the number of users, at any cost less than the cost, regardless of whether the project can be done, Users will not be satisfied; the

pursuit of PR articles and various hype, but failed to seriously do user cases. In short, products and services are not well done, and users' reputation and confidence in OpenStack has not been established. In contrast, some companies that are serious about making products have developed their OpenStack cloud business very well. This shows that OpenStack can be done well and users are willing to use it.

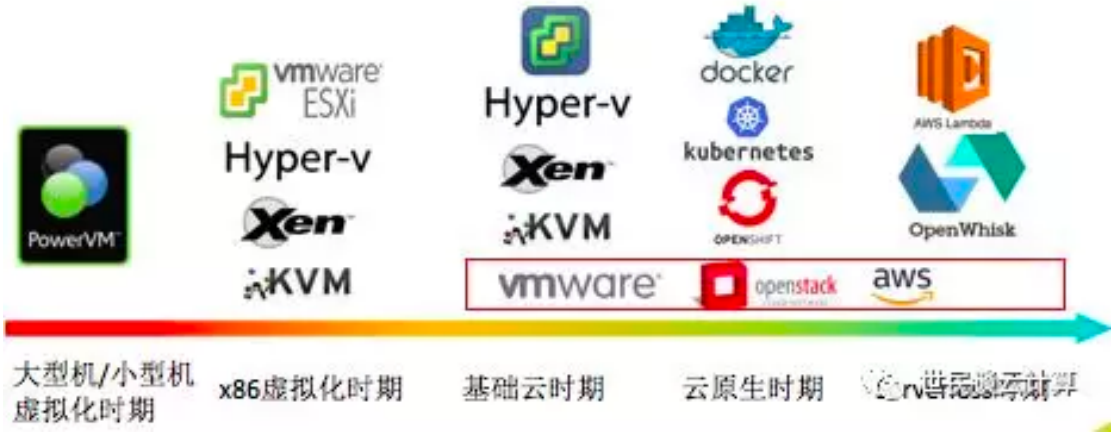
(4) Many customers, especially most traditional enterprises, actually use VMware virtualization, which is enough. The company's O&M system, resource delivery system, and application development, operation, and design architecture are all in the era of virtualization, so VMware supports existing applications. This can be seen from the continued growth in revenue from VMware's earnings. So how much motivation these customers can get from VMware to OpenStack is actually a big problem.

What will happen to the future of OpenStack?

Personally think that the future of OpenStack will have two paths:

- One is that OpenStack only takes the path of the KVM virtual machine and the choreographer of the Ceph storage volume. This way, it will inevitably go to the same end as the open source cloud platform like CloudStack, that is, it has really faded before it really rises.

- The other is OpenStack's path to AWS and even VMware, becoming the supporting platform for the underlying cloud, native cloud and future Serverless cloud. In this case, its path will be long.



However, unfortunately, from the current situation, OpenStack should be on the first road, perhaps this is why many people think that OpenStack is dying or even dead.

My feelings for OpenStack

I have a deep feeling for OpenStack. It is, let me know what is the cloud, how the cloud is built, how it works, how it is maintained, and so on. Starting from researching it, I started to enter the cloud field from the traditional software field. I also started the long history of blogging, and I also got to know a lot of friends through it. Therefore, when I saw someone deliberately smashing it, even when it was down, it was not a taste. In fact, I feel that it is not only me, but the entire IT field should be thanked for OpenStack, and its emergence has greatly accelerated the evolution of IT architecture. The previous content, perhaps the composition of the spray is mostly, but please understand my mood, because OpenStakc can be developed better, after all, it has had a good time, location and people. From the actual situation, if the enterprise has an OpenStack R&D team, or find a reliable external supplier, and the scale is not particularly large, the business is not so complicated,

there are several powerful operation and maintenance, OpenStack private cloud can still Run very well. At least in China, OpenStack has become one of the main representatives of the self-controllable private cloud cloud platform, which is glowing in all walks of life.

Regardless of its ending, OpenStack will leave a lot of emphasis on the history of IT development. Here, I would like to thank the OpenStack project, thanks to OpenStack for every line of code and every document, the OpenStack community, and all the companies and people who have contributed to OpenStack.