All in One & One for All: Technical Advancing of “Kuaijie” Cloud Hosting

From: UCloud Kuaijie Cloud Hosting Team

Original Chinese Article: All in one & One for all, “快杰”云主机的技术进阶之路

Nearly half a year has passed since UCloud released the “Kuaijie” cloud hosting at the **“Think In Cloud”** (TIC) Conference on May 28. In this half year, “Kuaijie” has taken its own unique path in the cloud hosting market with its superior performance and high cost performance. Behind these achievements, we use two sentences to elaborate the technical concept of "Kuaijie": **All in One & One for All**.

All in One & One for All

As the cornerstone product of cloud computing, the core characteristics of the cloud hosting determine the expansion of other capabilities on the cloud, and also directly relate to the user experience. The starting point for users to choose cloud computing is simplicity, speed and economy. However, due to the diversity of scenarios of Internet and it services, most of the manufacturers in the industry are launching different types of cloud hosting to adapt to different scenarios, which also brings the complexity of user operation, maintenance and procurement.

What is "All in One"?

"Kuaijie" defines itself as a "simple" product. Simple not only means easy to use, but also means the integration of multiple hardware and software technologies, so as to provide users with ultra-high product performance. In other words, when you are faced with different performance requirements of

CPU, network and storage in various business scenarios, "Kuaijie" can be satisfied without considering too many factors.

This is a set of merits from “Kuaijie”:

- **Fully equipped with the latest generation of Intel Cascade Lake processor**
- **Equipped with 25G basic network and adopting new network enhancement 2.0 scheme**
- **Support RDMA-SSD cloud disk**
- **Network performance up to 10 million PPS**
- **Storage performance up to 1.2 million IOPS**

At the beginning of the product launch, we conducted a run score test on "Kuaijie". The test results show that under the same specification configuration, the performance of "Kuaijie" is significantly better than that of the same type of cloud hosting products on the market. For example, in the same 8-core 16GB configuration, the network performance of "Kuaijie" is more than three times higher than that of other offerings, and the storage performance is nearly four times different.

However, under such a high configuration, the price of "Kuaijie" does not increase by more than 20%. The price of some configuration models is basically the same or slightly lower than that of ordinary models. The price of cloud disk is only 60% or lower than that of similar products in the market. For **"Kuaijie" uses the price bonus of "One for all" to give back all technologies to users.**

"Rome was not built in a day". Whether “All in one” or “One for all”, these data cannot be separated from UCloud's continuous exploration and accumulation in technology. Next, let's talk about the technology advancing behind “Kuaijie”.

——01——

Network Enhancement 2.0: 4x Performance Improvement + 3x Latency Reduction

The network channel is one of the bottlenecks to severely reduce the performance of the cloud hosting. Here, it is worth mentioning that "Kuaijie" has made a technological breakthrough in the network enhancement by 25GbE smart network card.

Hardware Level Network Card Acceleration

Based on the demand of improving the performance of cloud hosting network, 25GbE network is becoming a trend. However, due to the performance bottleneck of the traditional software virtual

switch scheme, when the physical network card receives the message, it sends it to the Vhost thread according to the forwarding logic, and then the Vhost is transferred to the virtual machine. **The processing ability of Vhost becomes the key to the performance of virtual machine network.**

After investigating the mainstream smart network card schemes in the industry, we finally adopted the OpenvSwitch open-source scheme based on TC Flower Offload, which provides hardware level network card acceleration for "Kuaijie".The network card of virtual machine can be directly offloaded to the hardware, bypass the host kernel, and realize the direct data access from virtual machine to network card. Compared with the traditional scheme, the performance of the package forwarding from the new smart network card scheme in the whole switch is 24M PPS for small packet, and the performance of package receiving of a single VF is 15M PPS, which improves the overall performance of the network card by more than 10 times. When applied to the cloud hosting, the network capacity of "Kuaijie" is improved at least **4 times, and 3 times decrease of latency.**

Technical Breakthrough: Live Migration of Virtual Machine

At the time of the implementation of the scheme, we encountered a technical problem: **the live migration of virtual machine**. Since each vendor's SmartNIC is based on a VF passthrough solution, and VF's non-migration makes virtual machine migration difficult. Here we would like to share our solution.

Through the research, we found that users do not need to manually set the binding operation or make specific images which can properly solve the problem of user intervention. Inspired by this, we adopted the method of **VF+Standby and Virtio-net for virtual machine migration.** The specific migration process is as follows:

1. Create the Virtio-net network card of the virtual machine, then select a VF on the host as a Hostdev network card, set the same MAC address as the Virtio-net network card, attach to the virtual machine, so that the virtual machine will automatically form the similar bonding function to

the Virtio-net and VF network card, at this time, there are two network data planes for the virtual machine on the host;

2. The tap device of Virtio-net backend is automatically added to the OpenvSwitch bridge of the Host when the virtual machine is started. When the virtual machine network card is switched, the datapath also needs to be switched. Attach VF to the virtual machine and replace the tap device with VF_repr on the OpenvSwitch bridge.

In addition, other technical innovations of UCloud for 25GbE smart network card can be viewed at: UCloud's practice: based on OpenvSwitch offload via high-performance 25GbE intelligent network card

——02——

RDMA-SSD Cloud Disk: Providing 1.2 Million IOPS Storage Capacity

In terms of cloud disk optimization, we mainly completed the technical realization of RDMA-SSD cloud disk from three aspects: IO access layer performance optimization, RDMA network acceleration and back-end storage node improvement, and finally provided "Kuaijie" with 1.2 million IOPs storage capacity.

Performance Optimization of IO Access Layer Based on SPDK

The following figure indicates the traditional QEMU Virtio solution. In step 3, **when the driver layer in QEMU listens through "gate" to forward IO requests from Unix domain socket, there is extra copy overhead,** thus becoming the performance bottleneck of IO access layer.

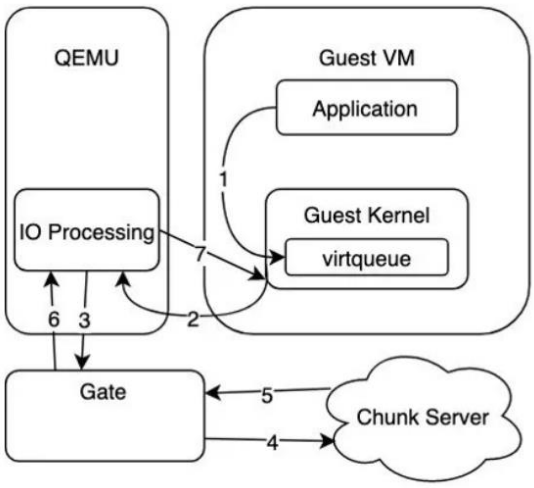

Figure: QEMU Virtio Scheme

To address this issue, UCloud uses SPDK Vhost to optimize the virtualization IO path.

(1) SPDK Vhost: Realize Zero Copy of Forwarding IO Requests

SPDK (Storage Performance Development Kit) provides a set of tools and libraries for writing high-performance, scalable, user state storage applications. The basic components are user space, polling, asynchronous, lockless NVMe driver, providing zero copy and highly parallel access to SSDs directly from user space applications.

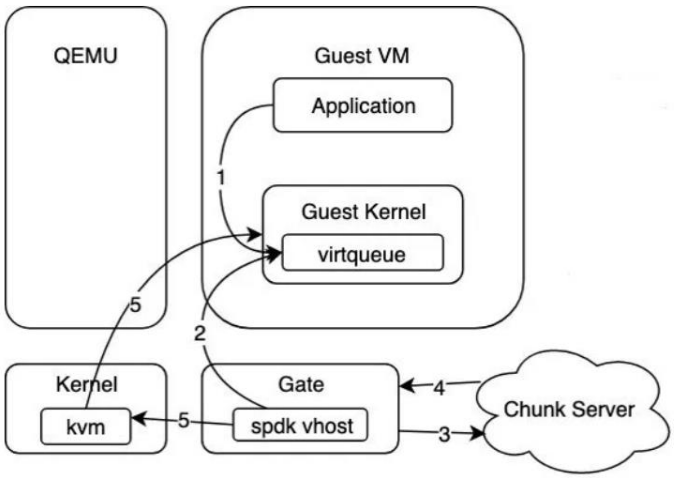

Figure: SPDK Vhost Scheme

As shown in the figure above, after integrating the SPDK Vhost solution, the IO path flow is as follows:

1. Submit IO to virtqueue;
2. Poll virtqueue to process the new IO;
3 + 4. Backend storage cluster processes IO requests from Gate;
5. Notify Guest IO to complete via irqfd.

In the end, SPDK Vhost can quickly transfer IO requests between VM and Gate by sharing large pages of memory. **In this process, there is no need to copy memory**. It is entirely pointer transfer, so it greatly improves the performance of IO path.

As shown in the following table, we test and compare the performance of new and old gates. It can be seen that after the integration of SPDK Vhost, the latency and IOPS are significantly optimized, and the **latency is reduced by 61us, IOPS is increased by 58%.**

Test item	QEMU Virtio + Old Gate		SPDK Vhost + New Gate	
	average latency	IOPS	average latency	IOPS
QD=1, 4KB, Randread	403us	2445	342us	2874
QD=128，4KB，Randread	2596us	49.3k	1631us	78.3k

(2) Breakthrough of Open Source Technology Difficulties: Live Upgrade of SPDK Vhost

When we use SPDK, we find that it lacks an important feature - **live upgrade**. We cannot 100% guarantee that the SPDK process will not crash at all time. Although the user space SPDK back-end is restarted from crashed very quickly, IO in the front-end QEMU will be stuck sometimes, even after the SPDK is restarted, it cannot be recovered.

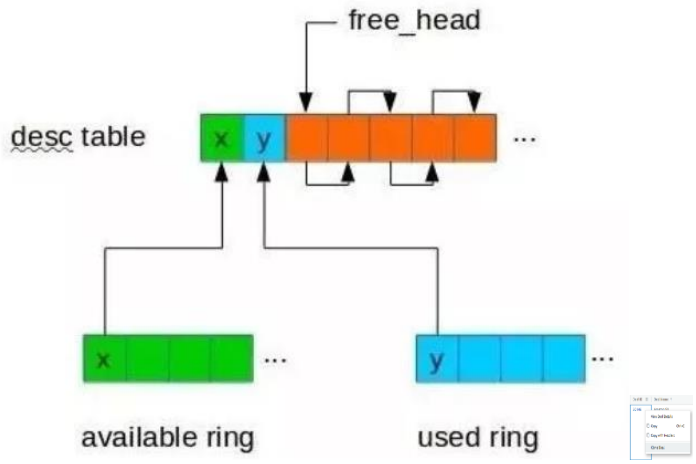

Figure: The Mechanism of Virtio Vring

Through in-depth study of the mechanism of virtio vring, we found that when SPDK exits normally, it will ensure that all IO has been processed and completed before exiting. That is to say, the virtio vring is clean. In case of unexpected crash, this guarantee cannot be made. In case of unexpected crash, some IO in virtio vring has not been processed. Therefore, after SPDK recovers, you need to scan virtio vring to send the unprocessed requests.

To solve this problem, we allocate a new piece of shared memory for each virtio vring in QEMU and send it to SPDK during initialization**. SPDK will record the status of each virtio vring request in this memory when handling IO, and can use this information to find out the request that needs to be re-**

issued after the unexpected crash recovery, so as to realize the live upgrade (or also called live migration) of SPDK. For details, please refer to [SPDK IO path optimization practice that can realize 1.2 million IOPS of RSSD cloud hard disk.]

RDMA Network Acceleration

(1) TCP bottleneck

After solving the IO path optimization problem, we continue to look for the key points to improve the IO read-write performance of cloud disk. At the protocol level, we found the following problems when using TCP protocol:

- There are interrupt overhead from network card and data copy overhead from kernel space to user space during TCP transmitting and receiving data;
- TCP is based on streaming transmission, so generally, network framework (i.e., libevent) will use a buffer to hold data temporarily, and it will not be removed from the buffer until the data reaches the processable length. Similarly, in order to simplify the exception caused by TCP buffer full during the sending process, the network framework will also have a send buffer, so a second data copy will be generated here.

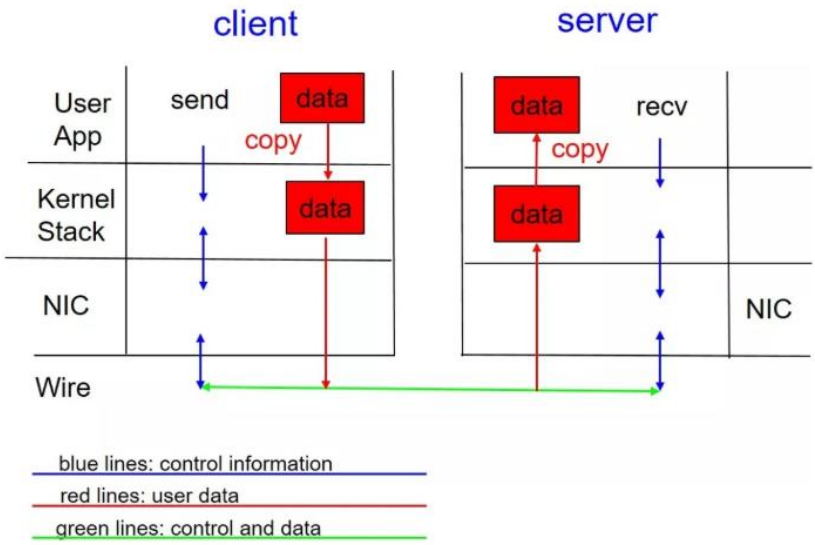

Figure: Principle of TCP Protocol

To solve this problem, we use RDMA protocol instead of TCP protocol to improve IOPS and latency.

(2) RDMA Replaces TCP

RDMA technology is called remote direct memory access, which is to solve the latency of data processing on the server side in network transmission.

The advantages of using RDMA instead of TCP are as follows:

① RDMA' s data path is kernel bypass, data is DMA by network card in the process of transmission, there is no data copy problem.

② There is no context switch in the process of RDMA receiving and sending. When sending data, post_send the data to SQ, and then inform the network card to send. After sending, a CQE will be generated in CQ; there are some differences in the process of receiving, RDMA needs to post_recv some buffers in advance, and the network card will write directly to the buffer when receiving packets, and a CQE will be generated in CQ.

③ RDMA is message transmission, that is, assuming that the sender sends a packet with a length of 4K, and if the receiver receives it, the packet length is 4K, and there is no case that only a part of it is received. This capability provided by RDMA can simplify the packet receiving process. It does not need to judge whether the data has been received completely or not like TCP, and there is no buffer required by TCP.

④ The protocol stack of RDMA is implemented by the network card, and the data plane is offloaded to the network card, which frees up the CPU and brings better latency and throughput.

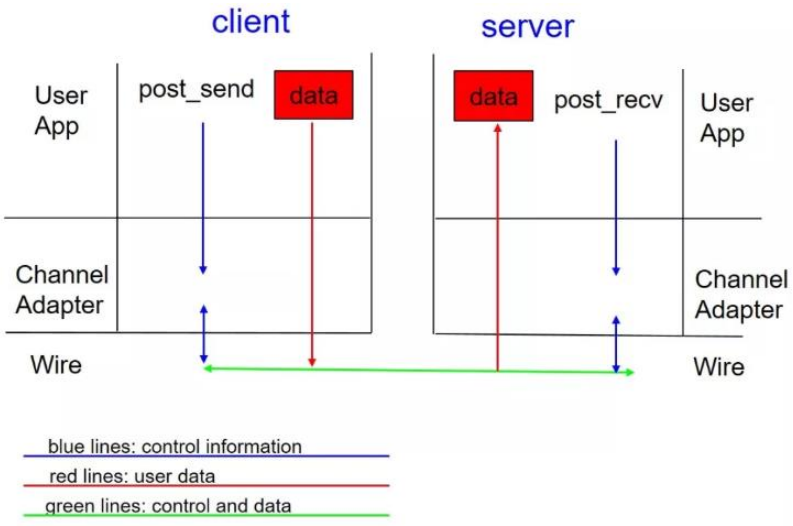

Figure: Schematic Diagram of RDMA Protocol

The Backend Storage Node's IO Path is Accelerated

In addition to improving the IO path access and transmission protocol, UCloud also optimizes the cloud disk backend storage nodes.

For the original Libaio with Kernel Driver, **SPDK NVMe Driver** is adopted to replace it. The figure below is Fio comparing and testing of the single core performance of the two drivers. It can be seen that the performance has been greatly improved after applying SPDK NVMe Driver.

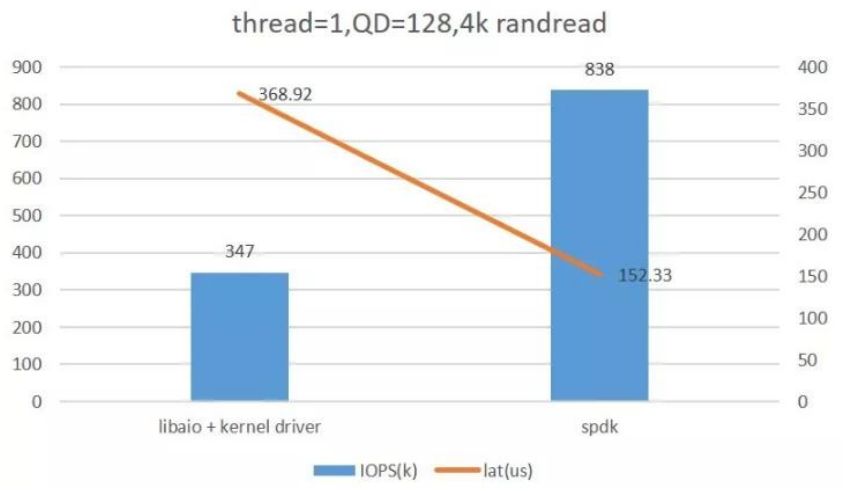

Figure: Single Core Performance Comparison of Libaio with Kernel Driver & SPDK NVMe Driver

In addition, using polling mode, SPDK NVMe driver can cooperate with RDMA to unleash the best performance of backend storage.

To sum up, we have realized the overall optimization of cloud disk: **SPDK Vhost** is used instead of QEMU to realize zero copy of data from virtual machine to storage client. Use high performance RDMA as the communication protocol of backend storage, realize the offload of receiving and sending packets to the hardware, make RSSD cloud disk's latency reduced to 0.1 millisecond, the experience is almost consistent with the local disk; Storage engine by SPDK instead of libaio, in the high concurrency situation, it can still maintain a low latency**. With the full 25G of the underlying physical network**, the random read and write performance of RDMA-SSD cloud disk can reach the best, achieving 1.2 million IOPS.

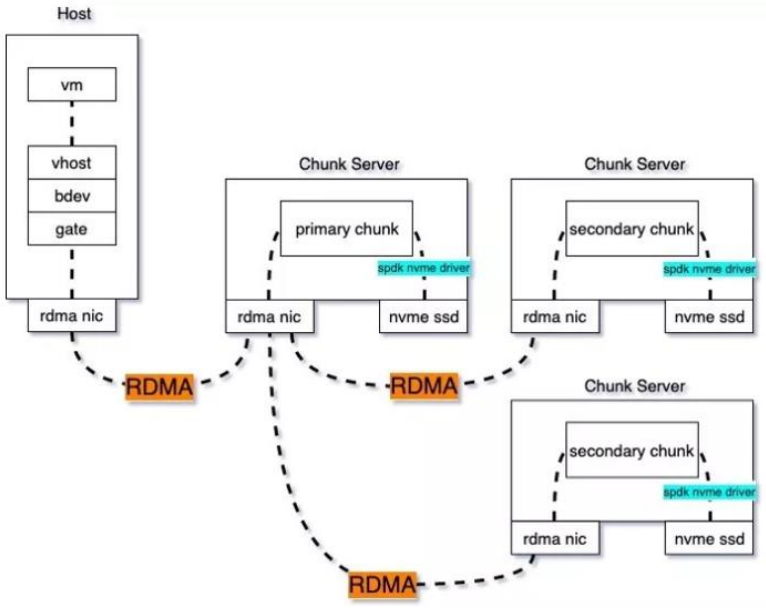

Figure: Schematic Diagram of RDMA-SSD Cloud Disk

——03——

Kernel Tuning: 10% Improvement in Overall Product Performance

When it comes to virtual machine, more people think of computing, storage and network, and few people pay attention to the kernel. However, kernel construction is the core work of the cloud hosting, which is responsible for managing the process, memory, device driver, file and network system of the system, which is very important to the performance and stability of the cloud hosting.

Before optimization, we conducted benchmark performance test for specific business scenarios in the cloud hosting. In the process of testing, using perf, systemtap, eBPF and other dynamic tracking technologies, at different observation levels such as host kernel, KVM and guest kernel, the factors that affect performance are analyzed at the instruction level.

On this basis, we focused on the kernel's enhancement and optimization work.

CPU Enhancements & Bug Fixes

We have added support for a new generation of Intel Cascade Lake CPUs for QEMU and KVM, comparing to previous Skylake, Cascade Lake CPU adds the clflushopt, pku, axv512vnni and other instruction sets to perform better in specific scenarios. In addition, regarding the CPU vulnerability, we solved Meltdown, MDS, L1TF and other vulnerabilities by the hardware, and added a less cost Spectre_v2 patch from Enhanced **IBRS Enhanced repair mechanism** to repair the vulnerability at the virtualization level. Finally, we give "Kuaijie" the ability to take advantage of the new hardware, so that the cloud hosting can avoid the Guest kernel's necessity to fix the security holes at the software level in VM, and eliminate the performance overhead and the decrease in business indicators caused by this.

CPU Memory Read-Write Optimization

For the CPU to optimize the memory reading and writing ability, we mainly from two aspects to achieve.

First of all, based on the hardware memory virtualization (Intel EPT), we add the support of customized hugepage memory, so as to avoid the manager / allocator's overhead and page change delay in the

memory virtualization, greatly reduce the page table size and TLB miss, and ensure that the cloud hosting's memory are isolated from other applications and system software to avoid impact each other.

Secondly, we enhanced the use of **NUMA affinity**. As we all know, the overhead of accessing memory across sockets is far greater than that generated by local access. To solve this problem, we use reasonable resource isolation and allocation to bind the VCPU and memory of the cloud hosting to the same socket. In addition, for large scale cloud hosting, there may be insufficient resources on a single socket. We allocate the cloud hosting to two sockets and expose the topology of the sockets to the Guest kernel, so that the cloud hosting can more easily use NUMA features to schedule and manage key businesses.

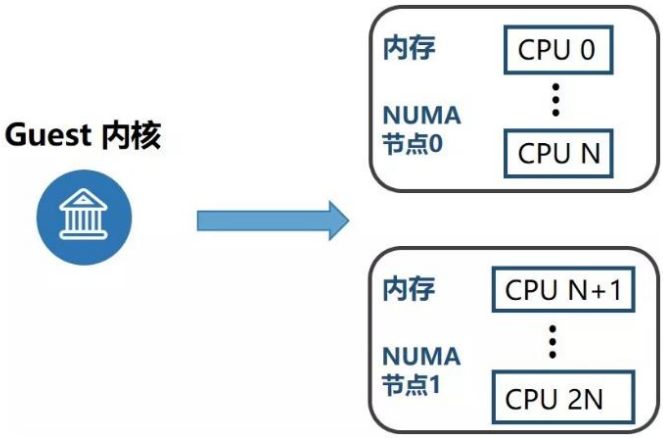

Figure: The Use of NUMA Affinity

Host Kernel & KVM Optimization

Combined with the performance analysis data, we also made a lot of optimizations to the Host kernel and KVM.

In VCPU scheduling, we find that CFS scheduler will use O (n) algorithm in critical region, which leads to high overhead of scheduler, decreased Guest's computing time and increased scheduling delay. We fix this problem in CFS.

In addition, during **the Host/Guest context switch**, we found that the context maintenance code for some registers introduced some overhead, so we eliminated the overhead caused by the maintenance code while ensuring the correctness of the register context switch.

During a cloud hosting run, a large number of **Inter-Processor Interrupts (IPI)** occur, each IPI causes a VM-Exit event. We introduced two new features in the virtualization layer: KVM-PV-IPI and KVM-PV-TLB-Flush. Through the Send-IPI Hypercall provided by KVM, the cloud hosting's kernel can use PV-IPI operation to eliminate a large number of VM-Exit, so as to reduce the IPI overhead. When the cloud hosting updates the TLB, VCPU as the initiator will wait for other VCPU to complete TLB Shootdown. The

cloud hosting’s kernel greatly reduces the cost of waiting and waking up other VCPU through PV-TLB-Flush.

These are some important optimizations, and won't go over the other kernel, KVM, QEMU enhancements, and stability improvements. After overall evaluation, through kernel tuning, it can help "Kuaijie" to achieve more than 10% of the overall ability.

——04——

Analysis of Three Application Scenarios

Based on the powerful performance, "Kuaijie" can easily meet the use scenarios of high concurrent network clusters, high-performance databases and massive data applications. We have selected three application scenarios of **Nginx cluster, TiDB** and **ClickHouse database** respectively. Now let's see the performance of "kuaijie":

Scenario 1: Build The Nginx Cluster to Break The Network Restrictions

Appfactory is a company engaged in advertising DSP (demand side platform) business. Due to business needs, Appfactory has very high requirements for high concurrency of network clusters. Finally, Appfactory chooses to use "Kuaijie" to build Nginx cluster as API gateway to provide services to its terminal customers.

Nginx is a lightweight HTTP reverse proxy Web server that supports approximately 25% of the world's busiest Web sites, including Dropbox, Netflix, Wordpress.com, and more, according to its official website. Its characteristic is the concurrency ability is strong, and "Kuaijie" further enhanced its concurrency ability.

"Kuaijie" breaks through the network limitation based on cloud hosting. As shown in the figure below, the usage of "Kuaijie" enables the number of hosts in the original cluster of Appfactory to be greatly reduced, and **the cost is halved under the same service capacity.**

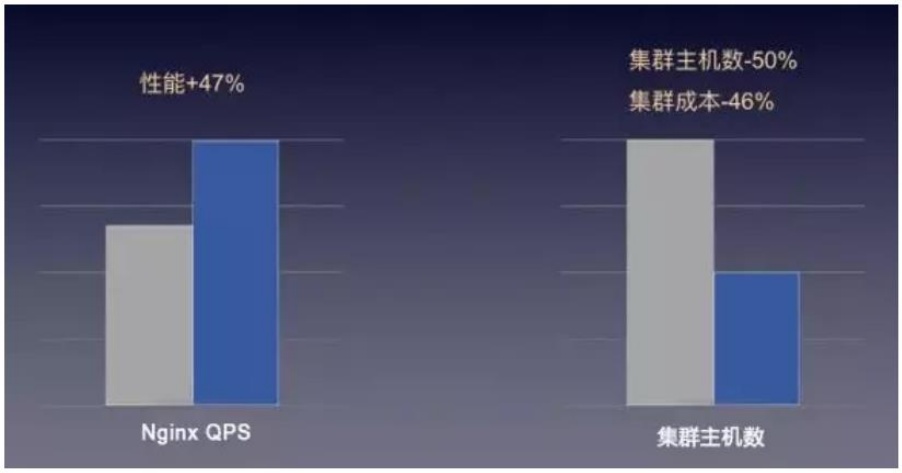

Figure: The Performance of "Kuaijie" in The High Concurrent Network Cluster Scenario

Scenario 2: Build TiDB and Breakthrough IO Performance Bottleneck

The PingCAP TiDB is a popular open-source distributed relational database. It is designed for the needs of high concurrent real-time writing, real-time query, real-time statistical analysis, etc. in the era of big data, and the requirements for IO performance are undoubtedly very high. Generally, the underlying layer of TiDB is required to use NVMe SSD local disk to support its performance, but “Kuaijie” cloud hosting can meet the high requirements of TiDB through RSSD cloud disk, and its performance is also recognized by PingCAP engineers.

At present, many UCloud customers have used “Kuaijie” cloud hosting to build TiDB, breaking through the previous database performance bottleneck.

Figure: Performance of "Kuaijie" in High Performance Database Scenario

In addition to TiDB, the measured performance of "Kuaijie" can effectively improve the performance of various databases by **more than 20%.**

Scenario 3: Build ClickHouse to Increase Data Throughput by 2 times

TT is a special voice tool for mobile game players. Because of the business requirements, it is necessary to collect the embedded data of App into the big data cluster for analysis. TT uses “Kuaijie” to build ClickHouse database as the core of the whole big data cluster. Comparing to previous solution, with “Kuaijie”, the daily increment of operation reaches 800 million records.

In addition to the ClickHouse scenario, “Kuaijie” can also optimize the big data ecology in an all-round way, as shown in the figure below, data throughput can be increased up to 2 times, which will help the development of big data business of enterprises.

Figure: Performance of "Kuaijie" in Big Data Application Scenarios

——05——

Conclusion

Based on the concept of "Co-design of software and hardware", “Kuaijie” has achieved a great technical advancing in network enhancement 2.0, RSSD cloud disk optimization, kernel tuning and other aspects, bringing breakthrough cloud hosting performance improvement for users.

For the improving of “Kuaijie” technology, the change and upgrade of technology can be described with words. However, behind the realization of technology, it represents the persistence and pursuit of UCloud to create core value for users.