The Monitoring Method Research of Docker Vessel Performance

Song Yao

Jinnan District of Tianjin City Road No. 1 Yaguan Haihe Education Park, Tianjin, 300350, China

yaosong@126.com

Keywords: Docker, vessel, performance monitoring, Linux

Abstract. Docker has made another better choice for the establishment and arrangement of the applied operation environment which is different from the traditional virtual machine technology through its own features, however, the imperfect performance monitoring function has constrained the application and development of Docker platform. This article will make a further research on the monitoring target and method of Docker platform based on the existed developmental practice and functions of Docker platforms, which has a certain reference meanings for researching the operation systems and operation monitoring of Docker platforms.

Introduction

The development of Internet indicates that all the applied systems should be developed by a series of the perfectly-defined agreement stack mode which is based on the middle ware, the database and operating systems and arranged by the method of adopting the special function server, which has led to the more clumsiness of the applied systems, that is to say that the applied systems are not able to work normally once being separated from the original environment, which will also lead to the increase of the difficulty degree of the system arrangement, management and maintenance with the increase of the complex degree of the applied systems. This mode has not been able to satisfy the needs of the current increasingly complex Internet application.

In face of the issues of the increasingly large application systems, the unfamiliar systems and more complex operations, Even though we can utilize the methods to deal with through adopting the quick developmental mode from starting small and continuous improvement or harnessing the enterprise-level framework to adapt the huge business systems. However, we will still develop, arrange, maintain a series of actual issues face-to-face. Therefore, if there is a mode like the application system of packing container which can separate its mutual influences between them, the application can be made to move swiftly under the different operating environment and between the different platforms. This will solve the challenges of the huge systems, difficult arrangements and operations.

The Research Background

The arrival of the cloud computer age has brought the hopes for solving all those above challenges. The cloud computing is divided into the software that is the service (SAS), the basic infrastructure service (IaaS), the platform that is the service (PaaS) according to its service mode. The operating environment and developmental environment have been treated as a cloud computing mode for serving for the network by Paas. The Paas provider can provide the platform-level products of the operating systems and applied developmental environment for the Paas users through the network by means of service. Paas has explained the developmental group and operating group, which has explained the developmental group and operating group, which has made the applied system development and the specialized operating division more possible. This has altered the traditional arrangements and operating modes of the applied system, which has greatly improved the efficiency of the software delivery and operation.

One of the cores of Paas technology is the virtualization. The virtualization is a method from which a certain user and applied programs can be easier to benefit to express the process of the computer materials instead of expressing them according to the special methods of the resources

realization, geographical positions or the physical packages. In another words, it has provided a logical view for the database computing abilities, the storage resources and other resources other than the physical view. The virtualization is divided into the virtual technologies based on the hardware and software. Docker belongs to the virtualization of the operating systems level and it is based on the software.

Docker is a senior container engine based on LXC and opened by dot Cloud. PaaS provider. It can create a lighter transplantable and self-sufficient container for any application with great relaxation. The container will entirely use the sandbox mechanism without any conjunction between each other. Docker has obtained the endorsement and focus from huge companies like RedHat, IBM. Google, Baidu, Alibaba etc since it was released in 2013.

The aim of Docker is to realize "One package, run everywhere."through the management of all life circles of the packaging distribution, arrangement and operations or the applied systems and assemblies. That is to say, Docker is like the container which can package applied systems and interdependence and the applied systems can be made to operate under any Docker environment.

The traditional virtualization systems are generally based on the hardware virtualization, which adopts the instruction virtualization method to fully virtualize the complete features of a set of the physical host including CPU, the internal storage, the disk and network card to present to each virtualized environment. Each virtualized environment has its own operating system respectively. The corresponding operating environment will be configured on this foundation. As each virtualized environment is required to be configured by the respective operating environment. Therefore, the maintainability of the system is not very so good and it will also consume the resources of the host computer.

Compared with the traditional virtualized machine, as Docker adopts the virtualization of the operating system level, which oeprates above the host operating system. All the containers mutually share the core within one system even the public library. The container engine provides the process level isolation, which enables each container to operate on a separate system. Therefore, Docker has some features of being light with lower system resource consumption and being easy to manage and maintain. Therefore, Docker is suitable to establish the private cloud within the Unit and arrange the applied systems rapidly so as to accomplish the purpose of reducing the operating cost.

The Index of Docker Performance Monitoring

The basic theory of Docker performance monitoring originates from the difference between the vessel monitoring and traditional virtualized machine monitoring. The traditional virtualized machine monitoring scheme will obtain the index from each sever and its operating application. These severs and applications operating on the severs are generally static with very long operating time. However, the vessels on the host computer is generally behaved as the process with their own environment, the virtual network and different storage management and mutually share the resources of the underlying host. The short-term living batch command and the long-term living process will be possible to be dispatched on the same vessel. Therefore, the performance monitoring of Docker should include the two aspects of performance index monitoring based on the host computer and performance index monitoring based on vessels.

(1) The index based on the host computer

The Docker host computer is operating for long-term compared with vessel. Therefore, the operating status of the host computer should be monitored so as to manage and prioritize various resources and processing abilities.

CPU storage load and internal storage load: We can timely master the load status of the host computer through the monitoring or the two index and fully display the capabilities of the host computer. When CPU load appears to reduce , which means the service of a certain vessels has been disrupted when CPU load has exceeded the maximum (such as 90%), which means that the load ability of the host computer has reached the limitation. We can draw the attentions of the managers if the two above circumstances have occurred.

The disk space of the host computer: As the host operating system, vessel image will all consume the disk space. The usage condition of disk space has become a constraint index of bearing the vessel amount by the host computer. It will be a good operating practice when the more disk spaces will be emptied out and the more vessels will be born by clearing the disk control and removing the unusable vessels at regular interval. As the disk reading frequency has a great impact on the process and the overall operating conditions of the host computer. Therefore, the disk reading frequency should be monitored in order to avoid the host computer block caused by the arrangement of the application of several high disk reading frequency on the same host computer.

The vessel amount of operating on the host computer. Under the circumstance of the static usage, knowing about the current and historical vessel amounts will assist us to ensure all the functions with the same operating status as the previous arrangements upon the upgrade.

(2) The index based on the vessel

The resource sharing needs to have the reasonable quota for the vessel, the usage conditions of the vessel resources should be also seen vice-verse. Therefore, the behavior which needs the vessels is to be monitored and further adjusted accordingly.

Vessel CPU: The index of total amount of CPU time of the vessel provides the basic information of setting CPU sharing correctly. The rise of this index means the CPU processing abilities needed by one or several containers have been beyond the ability scope provided by the host computer.

The internal storage use, the internal storage exchange of the vessel and the failure counter of the vessel internal storage: The rise of these index means the internal storage amount needed by the vessel has exceeded the value distributed to them. The application will be ensured not to utilize too much internal storage by monitoring and restricting the top limit of the vessel usage, internal storage so as to avoid influencing other vessels on the same host computer.

The disk reading frequency of the vessel: The same host computer resources can be used simultaneously by several vessels. Monitoring the vessel disk reading frequency is conducive to allocating the higher throughput to he key application, such as the database storage or Web server. The batch processing operating can also be shunted by the disk I/O.

The network circulation of the vessel: For the inter-related vessels. It is very important to monitor the virtual network, such as the container load equalizer. The discarded data packet needs to be followed up. Furthermore, the network circulation can reflect the usage conditions of the client application. If it appears a higher value peak, which probably means the appearance of refusing the service attack, load testing or the fault produced during the clients application.

The Method of Performance Monitoring of Docker

The Docker vessel mainly operates on the Linux system, therefore, this article only discusses about the performance monitoring based on 64 digits Linux system.

(1) The systematic performance monitoring method of the host computer

A kind of documentary systems has been provided within the core of Linux through proc in order to visit the internal data structure in the core, alter the mechanism of core setting during the operation. Proc is a kind of pseudo file systems which is different from the common documentary systems. It stores a series of special documents during the current core operating status. The users can only check its relevant systematic hardware and the current operating information and they can even alter some core operating status by updating some documents among them.

The above-mentioned particularities are based on proc documentary systems. The documents among them are frequently called as the virtual documents with some particular features, such as some documents will return a large amount of information when they are checked by the checking orders. However, the size of the document itself will be shown as zero byte. In addition, the time and date attribute in a majority of documents among these particular documents are generally deemed as the current system time and data, which is also related to the renovation at any time.

In order to check and use conveniently, these documents will be also generally classified according to its relevance and stored in different catalog or subcatalog. For example, all the relevant information of SCSI equippments in the current system is being stored in the proc/scsi catalog and

the relevant information stored in proc/N is the relevant information which is in the current operating. Among them , N is the processing ID, currently, the relevant catalogs will also disappear at the end of the process.

Meanwhile, the resource status of the entireness of the host computer systems is being shown through the proc documents as below and the relevant documents of the host computer performance monitoring are also as below:

(1) /proc/cpuinfo: The relevant information of CPU host computer

(2) /proc/diskstats. The disk I/O calculated information lists on each disk equipment, the Linux version supports this function after the internal core of 2.5.69.

(3) /proc/meminfo: The information related to the current inter storage utility conditions etc in the system.

(4) /proc/stat: Timely tracking the calculating information of the systematic operation since the previous launch of the systems:

a. The eight values after CPU line respectively indicate 1/100 of a second Unit system, running statistical values in user mode, a low priority user mode and transport system user mode, idle mode, I/O waiting mode time etc.

b. "intr"line has provided the disrupted information. The first is all the disrupted times since the establishment of the system. Then each number will correspond with a specific disrupted times occurring since the launch of the system.

c. "ctxt"provides the context exchange times of the CPU occurrence since the launch of the system.

d."btime"provides the time from the launch of the system to the present, the unit is the second.

e."processes (total_forks) the established task number since the launch of the system

f ."procs_running"The current number of running the parade task

g."procs_blocked"The number of the current blocked task

The main performance index monitoring can be realized on the host computer through timely reading the data in the above-mentioned documents.

(2) The performance monitoring method of Docker vessel

Docker vessels rely on cgroups. The vessel process, CPU storage. ID bar and the measurement information of the network usage condition can be tracked through cgroups. The groups will expose the data through a pseudo file systems the same as the proc cgroups lies in the catalog of the operating systems like sys/fs/cgroup, each catalog under it represents the cgroups levels of different equipment resources. For each vessels operating in Docker will produce a documentary catalog under each cgroups level corresponding with the vessel with the catalog of ID vessel. However, these subcatalogs are created with the vessel operation and disappear with the vessel elimination.

All the inner storage information can be obtained under the catalog osys/fs/cgroup/memory/Docker/<long_id>, during the operation of Docker vessel. CPU metric information can also be obtained under the catalog of /sys/fs/cgroup/cpu,cpuacct/Docker/<long_id>.

Although the resource usage of the Docker vessel can be monitored by cgroups, but it will not only consume more processing materials to use the programs to analyze the database documents of different formats, but it will be also very troublesome as well. As a matter of fact, Docker provides a group of rest long-rang API. These API has corresponded with the Docker operating instructions and has been dispatched by adopting the form of weburl. The methods for sending the request of API are HTTP, GET and HTTP POST. Different API will adopt different methods to send requests and they cannot replace each other. API returns the database result in the form of json. The general json data analytic method can be adopted to analyze the obtained data.

The documents of Docker background default listener unix//var/run/Docker.sock. In the systems of Ubuntu and Debian, you can modify the default listener port via.

THe API related to the vessel performance monitoring is as below:

(1) Listing the vessels:

Method: GET /containers/json

The main parameter: Whether all lists out all the vessels or with the value is 1/true or 0/false. All the vessels including the ones stopped operation has been listed out, otherwise only the vessels which are operating currently are being listed out with O as the default value.

The return value examples are as below:

{"Id":"7dbea7a80cea8579c3d7c09a8886b553ac94279f6464042dad5fda9f8f0b5a79","Names":["/big_jepsen"],"Image":"google/cadvisor:latest","ImageID":"sha256:4bc3588563b107ed7f755ddbbdf09cccdb19243671d512810da3ed0fef6f7581","Command":"/usr/bin/cadvisor-logtostderr","Created":1471148811,"Ports":[{"IP":"0.0.0.0","PrivatePort":8080,"PublicPort":8080,"Type":"tcp"}],"Labels":{},"Status":"Up3minutes","HostConfig":{"NetworkMode":"default"},"NetworkSettings":{"Networks":{"bridge":{"IPAMConfig":null,"Links":null,"Aliases":null,"NetworkID":"","EndpointID":"a84a0c35eb4c15f0fda913084f568442a89ad6fe257744073df806f1c0877e6a","Gateway":"172.17.0.1","IPAddress":"172.17.0.2","IPPrefixLen":16,"IPv6Gateway":"","GlobalIPv6Address":"","GlobalIPv6PrefixLen":0,"MacAddress":"02:42:ac:11:00:02"}}}}

(2) Obtaining the designated vessels operating status and resource usage situations.

Method: POST /containers/{containerId}/stats

{containerId} is the length of the designated vessels.

The data value corresponding with the key words" ID" in the return data of the previous instruction.

The main parameter: Whether the stream obtains the resource usage status of the designated vessels continuously with the value of 1/true or 0/false usage status. When the value is 1/true , the resource usage status of the vessel will be continuously circulated and output. Otherwise, one time is output with 1 as the default value .

The return value examples are as below:

{"read":"2016-08-14T12:37:37.348882228+08:00","precpu_stats":{"cpu_usage":{"total_usage":20187610993,"percpu_usage":[5945448710,5530703989,4371533350,4339924944],"usage_in_kernelmode":9390000000,"usage_in_usermode":9700000000},"system_cpu_usage":43125510000000,"throttling_data":{"periods":0,"throttled_periods":0,"throttled_time":0}},"cpu_stats":{"cpu_usage":{"total_usage":20212004616,"percpu_usage":[5955954596,5542576512,4371873248,4341600260],"usage_in_kernelmode":9400000000,"usage_in_usermode":9720000000},"system_cpu_usage":43129420000000,"throttling_data":{"periods":0,"throttled_periods":0,"throttled_time":0}},"memory_stats":{"usage":26882048,"max_usage":29249536,"stats":{"active_anon":26718208,"active_file":32768,"cache":172032,"dirty":0,"hierarchical_memory_limit":9223372036854771712,"inactive_anon":0,"inactive_file":131072,"mapped_file":0,"pgfault":48569,"pgmajfault":0,"pgpgin":7769,"pgpgout":1717,"rss":26710016,"rss_huge":2097152,"total_active_anon":26718208,"total_active_file":32768,"total_cache":172032,"total_dirty":0,"total_inactive_anon":0,"total_inactive_file":131072,"total_mapped_file":0,"total_pgfault":48569,"total_pgmajfault":0,"total_pgpgin":7769,"total_pgpgout":1717,"total_rss":26710016,"total_rss_huge":2097152,"total_unevictable":0,"total_writeback":0,"unevictable":0,"writeback":0},"failcnt":0,"limit":8279924736},"blkio_stats":{"io_service_bytes_recursive":[{"major":8,"minor":0,"op":"Read","value":131072},{"major":8,"minor":0,"op":"Write","value":0},{"major":8,"minor":0,"op":"Sync","value":0},{"major":8,"minor":0,"op":"Async","value":131072},{"major":8,"minor":0,"op":"Total","value":131072}],"io_serviced_recursive":[{"major":8,"minor":0,"op":"Read","value":1},{"major":8,"minor":0,"op":"Write","value":0},{"major":8,"minor":0,"op":"Sync","value":0},{"major":8,"minor":0,"op":"Async","value":1},{"major":8,"minor":0,"op":"Total","value":1}],"io_queue_recursive":[],"io_service_time_recursive":[],"io_wait_time_recursive":[],"io_merged_recursive":[],"io_time_recursive":[],"sectors_recursive":[]},"pids_stats":{},"networks":{"eth0":{"rx_bytes":9268,"rx_packets":55,"rx_errors":0,"rx_dropped":0,"tx_bytes":1340,"tx_packets":17,"tx_errors":0,"tx_dropped":0}}}

It should be particularly illustrated that the statistical data of CPU network circulation and the disk reading and writing are all deemed as cumulative value during the above-mentioned return value. The data reflects the overall situations of the resource consumption since its operation instead of the instant pressure.

Conclusion

Docker, as an emerging lightweight Paas platforms, has broken off the Paas forms based on the traditional virtualized machine technology, which enables the Docker to have the features of rapidity, high resource utility rate, low performance expenditure and comparatively lower operating cost based on the vessel technology pattern. However, in the aspect of performance monitor, the Docker platform only provides the simple consultation instruction with the insufficient functions, which controls the application, development of Docker platform. This article has made an investigation on the performance monitoring targets and methods based on the current existing developmental practice and the current functions of Docker platform, which enables the application arranged on the Docker platform to operate steadily and reliably and also provides a foundation for the performance priority of Docker vessels. Meanwhile, it also has some referencing meanings for researching and developing the operating systems of Docker platform.

References

[1] "Docker development and practice," compiled by Zengjinglong, Xiaoxinhua, Liuqing, China publishing group, the people's Posts and telecommunications publishing house.

[2] "Docker vessels and vessel clouds," written by SEL lab of Zhejiang university, China publishing group, the people's Posts and telecommunications publishing house.

[3] "Docker source code analysis," written by Sunhongliang mechanical industrial press.

[4] "The first book of Docker," written by James Turnbull (Australia), Translated by Lizhaohai Liubing Juzheng the people's Posts and telecommunications publishing house.

[5] "The skills and practice entry battle of Docker," written by Yaobaohua Daiwangjian Caoyalun mechanical industrial press.

[6] "The Progress and practice of Docker," written by Huawei Docker practice group mechanical industrial press.