International Journal of Innovative Research in Engineering
Volume 4, Issue 1 (January-February 2023), PP: 148-150.
www.theijire.com ***ISSN No: 2582-8746***

Application Performance Monitoring in Production Server

Barathkumar[1]
[1]Computer Engineering, Bannari Amman Institute Of Technology, Erode, Tamil Nadu.

How to cite this paper:
Barathkumar[1]. "Application Performance Monitoring in Production Server", IJIRE-V4I01-148-150.

Abstract: *A server is a computer used by multiple users for a specific application or multiple functions. The benefits of servers include centralized location, abilityto control the air conditioning, consistent data archiving and speed. For example,a print server can eliminate the need for each user to have a personal printer. This would improve the efficiency of using network printer and reduce the cost of maintaining multiple printers. Application servers, while similar to file servers, are unique in that they run executable applications from the central location. By using application servers, costs of application software are reduced. Web Server performance is cardinal to effective and efficient Information communication. Performance measures include response time and service rate, memory usage, CPU utilization among others. A review of various studies indicates a close comparison among different web servers that included Apache, IIS, Nginx and Lighttpd among others. The results of various studiesindicates that response time,CPU utilization and memory usage varied with different web servers depending on the model used. However, it was found that Nginx out performed Apache on many metrics that included response time, CPU utilization and memory usage.*

Key Word: *Server, CPU Utilization, Application Server, Apache, Lighttpd.*

I.INTRODUCTION

In the modern Information technology environment, the functions like websites, web-based application, a centralized computing system, mobile apps, e-commerce application or even cloud computing, be subsided with the concept of client-server. The client-server system is a distributed computing between two types of independent and autonomous entities known as server and client. The client-server computing places a vital role in data or information access form remotely stored lactations among the majority. The client-server system plays a significant role in IT evolution. The components involved in the client-server system divided into two major sections physical and logical components. Physical components are servers, client devices, input/output devices, networking, and power supply. Logical components are web pages, data, programming scripts, protocols, e.g., http, https, telnet, IP and API, e.g., ODBC, JDBC

II.MONITORING

Application monitoring is the process of collecting log data in order to help developers track availability, bugs, resource use, and changes to performance in applications that affect the end-user experience (ux). application monitoring tools provide alerts to live anomaly events, and through distributed tracing provide a means of seeing which events form a causal chain that led to them across multiple services. also known as application performance management (apm), application monitoring tools provide a visual means of seeing how events are connected through dependency and flow mapping. application monitoring can be accomplished by dedicated tools to monitor apps, or by collecting and analyzing logs using log management tools. with application monitoring, the end goal is to maximize availability and give customers the best experience.

III. GARBAGE COLLECTION

3.1 Java

Garbage collection in java is the automated process of deleting code that's no longer needed or used. This automatically frees up memory space and ideally makes coding java apps easier for developers.

Java applications are compiled into byte code that may be executed by a JVM. Objects are produced on the heap (the memory spaceused for dynamic allocation), which are then monitored and tracked by garbage collection operations. Most objects used in Java codeare short-lived and can be reclaimed shortly after they are created. The garbage collector uses a mark-and-sweep algorithm to markall unreachable objects as garbage collection, then scans through live objects to find objects that are still reachable.

Automatic garbage collection means you don't have control over whether and when objects are deleted. This is in contrast to languages like C and C++, where garbage collection is handled manually. However, automatic garbage collection is popular for goodreason—manual memory management is cumbersome and slows down the pace of application development.

During the garbage collection process, the collector scans different parts of the heap, looking for objects that are no longer in use. Ifan object no longer has any references to it from elsewhere in the application, the collector removes the object, freeing up memory in the heap. This process continues until all unused objects are successfully reclaimed.

Sometimes, a developer will inadvertently write code that continues to be referenced even though it's no longer being used. The garbage collector will not remove objects that are being referenced in this way, leading to memory leaks. After memory leaks are created, it can be hard to detect the cause, so it's important to prevent memory leaks by ensuring that there are no references to unused objects.

To ensure that garbage collectors work efficiently, the JVM separates the heap into separate spaces, and then garbage collectors usea mark-and-sweep algorithm to traverse these spaces and clear out unused objects. Let's take a closer look at the different generationsin the memory heap, then go over the basics of the mark-and-sweep algorithm.

3.2Python

When a programmer forgets to clear a memory allocated in heap memory, the memory leak occurs. It's a type of resource leak or wastage. When there is a memory leak in the application, the memory of the machine gets filled and slows down the performance ofthe machine. This is a serious issue while building a large scalable application.

Request: The requests library is an integral part of Python for making HTTP requests to a specified URL. Whether it be REST APIsor Web Scrapping, requests are must be learned for proceeding further with these technologies. When one makes a request to a URI,it returns a response. Python requests provide inbuilt functionalities for managing both the request and response.

Gc Module: Module gc is a python inbuilt module, that provides an interface to the python Garbage collector. It provides features to enable collector, disable collector, tune collection frequency, debug options and more.

In lower-level languages like C and C++, the programmer should manually free the resource that is unused i.e write code to managethe resource. But high-level languages like python, java have a concept of automatic memory manager known as Garbage collector.Garbage collector manages the allocation and release of memory for an application.

3.3Provisioning Servers

Network provisioning can include setting up a network to be accessed by users, servers, containers, and IoT devices, amongother things. In today's connected world, there are many different types of items that are network consumers.

Network provisioning has frequently been used by the telecommunications industry as a way to refer to providing a telecommunications service to a user, including the required equipment and wiring. It may also include service activation of a wireless environment for a use.

In the past, IT infrastructure provisioning was typically handled manually, including setting up the physical servers and configuringhardware to the desired settings. And if you needed additional capacity, you would have to order more hardware, wait for it to arrive,and then set it up and provision it. Today, infrastructure is now often defined in software, and virtualization and containers have sped up the provisioning process while eliminating the need for frequent hardware provisioning and management. Provisioning can also be handled through automation. Virtual infrastructure has increased the scale and capacity of many enterprise environments as well, which comes with its own challenges. Developers still need to provision virtual infrastructure for each new deployment, and manualprovisioning is time consuming and prone to human error. It's difficult to track changes and control versions, and avoid errors and inconsistencies when provisioning is managed manually by developers for each deployment. Infrastructure as code (IaC) offers a solution and makes it possible to automate your infrastructure.

IaC is the managing and provisioning of infrastructure through code instead of through manual processes.

Automating infrastructure provisioning with IaC means that developers don't need to manually provision and manage servers, operating systems, storage, and other infrastructure components each time they develop or deploy an application. With IaC,configuration files are created that contain your infrastructure specifications. This takes away the majority of provisioning work fromdevelopers, who just need to execute a script to have their infrastructure ready to go. IaC ensures that you provision the same environment every time. Deploying your infrastructure as code also means that you can divide your infrastructure into modular components that can then be combined in different ways through automation.

IV.PROBLEMS IN PRODUCTION SERVER

- Process Hanging while the programs are severing to customers in run time
- Garbage collector unable to run to collect the unused space in RAM
- Out Of Memory Process is tend to kill the actual running process tin the production server
- Machine hangs without any external or internal parameters affecting the production service
- Unable to scale the api calls when there is huge incoming calls serving.

V.OBJECTIVES

Study on the web server and application server in the production environment

Working with the existing monitoring metrics for the application server

Learning the web servers with its implementation apache and ngnix for my project implementation

Threading and multiprocessing modules in the app server – python implementation

Case study on threading module used for the memory analysis in production servers

Identify the memory leaks in the running application

Run garbage collector when system unable to run implicitly

Record the api calls

Log or append the result to the dashboard built for monitoring

View the thread stack when system about to down (when 90% memory filled)

VI.APPROACHES

6.1 Out Of Memory Handling

The "OOM Killer" or "Out of Memory Killer" is a process that the Linux kernel employs when the system is criticallylow on memory. This situation occurs because processes on the server are consuming a large amount of memory, and the system requires more memory for its own processes and to allocate to other processes. When a process starts it requests a block of memory from the kernel. This initial request is usually a large request that the process will not immediately or indeed ever use all of. The kernel, aware of this tendency for processes to request redundant memory, over allocates the system memory. This means that when the system has, for example, 8GB ofRAM the kernel may allocate 8.5GB to processes.

The solution that the Linux kernel employs is to invoke the OOM Killer to review all running processes and kill one ormore of them in order to free up system memory and keep the system running.

Steps to Avoid killing the process

The vm_over commit_memory variable memory can be controlled with the following settings :

0: Setting the variable to 0, where the kernel will decide whether to over commit or not. This is the default value formost versions of Linux. (Not Preferred).

1: Setting the variable to 1 means that kernel will always over commit. This is a risky setting because the kernel willalways over commit the memory to processes. This can lead to kernel running out of memory because there is a good chance that processes can end up using the memory committed by the kernel. (Existing).

2: Setting the variable to 2 means that kernel is not supposed to over commit memory greater than the over commit_ratio. This over commit_ratio is another kernel setting where you specify the percentage of memory kernel can over commit. If there is no space for over commit, the memory allocation function fails and over commit is denied. This is the safest option and recommended value for Memory consuming process.

Swappiness adjusting:

Swappiness can have a value between 0 and 100. A value of 0 instructs the kernel to aggressively avoid swappingout for as long as possible. A value of 100 will aggressively be swapping processes out of physical memory.

A lower value will make the kernel try to avoid swapping whenever possible while a higher value means the kernel will try to use the swap space more aggressively.

Accessing swap memory is much slower than accessing physical memory directly. A lower value for the swappiness parameter will most likely improve overall system performance. For regular desktop installation, a value of 10 isrecommended. A swappiness value of 0 or 1 is recommended for most database servers.

The optimal Swappiness value depends on your system workload and the size of the RAM memory. You should adjustthis parameter in small increments to find an optimal value.

6.2 Profiling Memory and CPU

Accounting the python language and its framework formulated the approach to view the memory and monitor the performance of the server and its process Monitor function to be executed to record the memory and stack framescontinuously with 1 second interval using the sys._current_frames acquiring the current function execution with code After fetching the results it is converted to key value pairs with thread id (thread responsible for the execution) as key and stack frame details as the value. new_threads acquires the current stack frames at the instant where the old_threads acquires the total stack frames record

API Alert:

The api can be processed by any one of the worker thread in cherrypy application, since the /blog method was handled by the above mentioned thread, as remaining threads are in waiter. acquire function as waiting to acquire the api request.

Note: As when the memory exceeds the threshold printing the stack frames can identify the memory leaks occurring spot.

Here in the case memory exceeds more than 100 mb at the time we are capturing the stack frames , hence the function,code present in the stack frame at the time will be responsible for the memory hike or memory leaks.

Advantages:

Can find the Code with function and line number which responsible for the memory leak or memory hike Can view the stack frame respect to the thread id where each frames (function) handled by the threads available in the cherry py application

Overall code specific approach

Decorated implementation

References

[1]. https://workdrive.zohopublic.com/writer/open/89g9qc778777244d647bbacb78cb96658a89e.