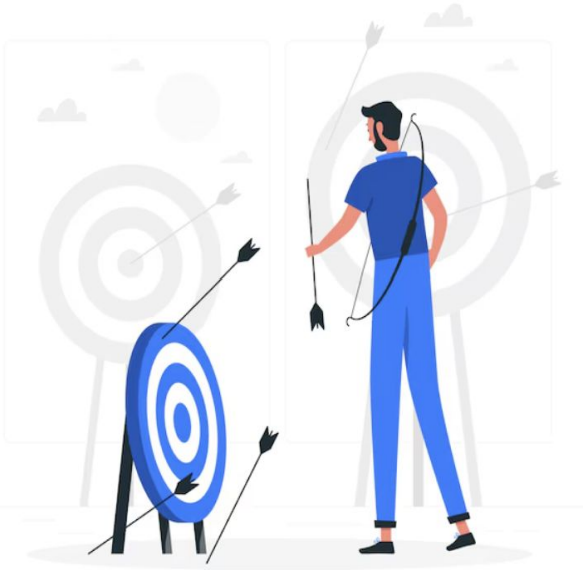
PW SKILLS | DevOps and Cloud Computing

Advanced System Optimization and Resource Allocation in Linux

Objective

- Identify CPU, memory, and I/O bottlenecks using system metrics.
- Recognize and resolve memory issues such as high swap usage and low free memory.
- Utilize sar for system activity reporting and understand th basics of long-term monitoring with Prometheus.
- Optimize CPU usage through workload offloading and resource balancing.
- Tune kernel parameters using sysctl for networking and resource allocation.
- Apply disk partitioning strategies for optimizing read-heavy and write-heavy workloads.



PW SKILLS

System load averages and their interpretation using uptime.

System load averages

- System Load Average is a critical metric in Linux that indicates how busy the CPU is with tasks currently being processed and those waiting in line for processing.
- It reflects the average number of processes that are either actively using the CPU or waiting to use it over specified time intervals—typically 1, 5, and 15 minutes.
- This metric is distinct from CPU usage, which provides a snapshot of CPU activity at a specific moment.

System load averages

How Load Average is Calculated:

Load averages are calculated by averaging the number of active and waiting processes over the past minute, five minutes, and fifteen minutes. For example, when you execute the ‘uptime’ command, it displays three load average values:

- 1-minute average
- 5-minute average
- 15-minute average

System load averages

Interpreting Load Average Values:

1. Normal Load
2. Overloaded system
3. Understanding zero values

System load averages

Using the Uptime Command:

The ‘uptime’ command provides valuable insights into system performance:

```
$ uptime
10:14:14 up 60 days, 17:42, 1 user, load average: 0.44, 0.28, 0.25
```

In this output:

- The first number (0.44) represents the load average for the last minute.
- The second number (0.28) represents the load for the last five minutes.
- The third number (0.25) represents the load for the last fifteen minutes.

Pop Quiz

Q. What does the load average represent in a Linux system?

A

The average number of processes waiting to be executed

B

The total CPU usage percentage

Pop Quiz

Q. What does the load average represent in a Linux system?

A

The average number of processes waiting to be executed

B

The total CPU usage percentage

PW SKILLS

Identifying CPU, memory, and I/O bottlenecks with real-world examples.

CPU Bottlenecks

To identify CPU bottlenecks, you can use the top or htop command. A high percentage of CPU usage indicates that the CPU is under heavy load. Specifically, look for the following:

- User %
- System %
- iowait%

```
top
```

Example Command:

- If you see a user % nearing 100%, it indicates your processes are CPU-bound, meaning they are limited by CPU availability

Memory Bottlenecks

Memory bottlenecks can be identified using the free command or vmstat. Key indicators include:

- Available memory
- Swap Usage

Example Command:

```
free -m
```

- If the available memory is consistently low and swap usage is high, your system may be experiencing memory pressure

I/O Bottlenecks

I/O bottlenecks can be diagnosed using tools like iostat, iotop, and vmstat. Key metrics to monitor include:

- %Iowait
- %util
- Await

Example Command:

1. To check I/O statistics:

```
iostat -x -d 2 5
```

2. To monitor real-time I/O usage by processes:

```
iotop -o
```

If you observe a high %iowait alongside high await times, it indicates an I/O bottleneck.

Real-World Scenario

Imagine you are running a web server on a Linux machine experiencing slow response times. You can use these commands to diagnose the issue:

- Run top to check if the CPU is maxed out.
- Use free -m to ensure there's sufficient memory available.
- Finally, run iostat -x -d 2 5 to check if disk I/O is causing delays.

Pop Quiz

Q. What does a high percentage of 'iowait' indicate when monitoring CPU performance?

A

The CPU is fully utilized

B

The CPU is waiting for I/O operations to complete

Pop Quiz

Q. What does a high percentage of 'iowait' indicate when monitoring CPU performance?

A

The CPU is fully utilized

B

The CPU is waiting for I/O operations to complete

PW SKILLS

Discussing memory issues

Impact of high swap usage and how to reduce it

Impact of High Swap Usage:

1. Performance Degradation
2. Increased CPU usage
3. Application Instability

How to Reduce High Swap Usage:

1. Adjust Swappiness: The swappiness parameter controls how aggressively the kernel swaps memory pages.

```
echo 10 | sudo tee /proc/sys/vm/swappiness
```

To make this change permanent, add ‘vm.swappiness=10’ to ‘/etc/sysctl.conf’

Impact of high swap usage and how to reduce it

1. Increase Physical RAM
2. Optimize Applications
3. Use swapoff Temporarily
4. Monitor Memory Usage

Pop Quiz

Q. What is a primary consequence of high swap usage in a Linux system?

A

Slower response times for applications

B

Improved application performance

Pop Quiz

Q. What is a primary consequence of high swap usage in a Linux system?

A

Slower response times for applications

B

Improved application performance

Identifying Low free memory scenarios and their implications.

PW SKILLS

Low Free Memory Scenarios:

1. Available Memory Near Zero:

Implication: When the available memory (as shown by free -m or free -h) is close to zero, the system may struggle to allocate memory for new processes or applications.

- This can lead to performance degradation as the kernel attempts to manage memory more aggressively.

Identifying Low free memory scenarios and their implications.

2. **Increased Swap Usage:**

Implication: As free memory decreases, the system starts using swap space more frequently. High swap usage can slow down system performance since accessing data from disk is significantly slower than accessing it from RAM.

- This situation can also lead to increased CPU usage as the kernel manages swapping.

3. **Out of Memory (OOM) Conditions:**

Implication: If the system runs out of available memory, it may trigger the OOM killer, which terminates processes to free up memory.

- This can result in unexpected application crashes and loss of data.

Identifying Low free memory scenarios and their implications.

4. Performance Thrashing:

Implication: In scenarios where memory is heavily utilized, the system may experience thrashing, where processes are frequently swapped in and out of memory.

- This leads to high latency and reduced responsiveness, making the system appear sluggish.

5. Memory Leaks:

Implication: Applications with memory leaks can gradually consume all available memory, exacerbating low free memory conditions.

- This often goes unnoticed until performance issues arise, complicating troubleshooting efforts.

PW SKILLS

The sar command for collecting and reporting system activity.

‘sar’ command

The sar (System Activity Report) command is a powerful tool in Linux used for collecting, reporting, and analyzing system activity data.

- It is part of the sysstat package and provides valuable insights into various aspects of system performance.

Key Features of the sar Command:

1. Comprehensive Monitoring
2. Real-Time & historical data
3. Customizable Reporting

‘sar’ command

PW SKILLS

4. Installation:

The sar command is not installed by default on many Linux distributions. It can be installed via package managers:

- For Ubuntu: sudo apt install sysstat
- For Red Hat/CentOS: sudo dnf install sysstat

5. Usage Syntax: The basic syntax for using sar is:

```
sar -[options] [time_interval] [number_of_times]
```

‘sar’ command

For example, to monitor CPU usage every second for five seconds:

```
sar -u 1 5
```

Implications of Using sar:

- Performance Analysis
- Historical Insights

Pop Quiz

Q. Which package must be installed to use the sar command?

A

sarutils

B

sysstat

Pop Quiz

Q. Which package must be installed to use the sar command?

A

sarutils

B

sysstat

PW SKILLS

How to use sar for monitoring CPU, memory, and I/O metrics.

How to use ‘sar’

1. Monitoring CPU Usage

To monitor CPU usage statistics, you can use the -u option. This command displays CPU utilization at specified intervals.

Example Command:

```
sar -u 1 5
```

How to use ‘sar’

2. Monitoring Memory Usage

To check memory utilization, use the -r option. This will provide statistics on used and free memory.

Example Command:

```
sar -r 1 5
```

How to use ‘sar’

3. Monitoring I/O Activity

For monitoring I/O statistics, the -b option can be used to report on transfer rates and block device activity.

Example Command:

```
sar -b 1 5
```

Pop Quiz

Q. What does the sar command primarily do in Linux?

A

Collect and report system performance data

B

Manage user accounts

Pop Quiz

Q. What does the sar command primarily do in Linux?

A

Collect and report system performance data

B

Manage user accounts

PW SKILLS

Introduction to Prometheus for long-term metrics collection:

Introduction to Prometheus

Prometheus is an open-source monitoring and alerting toolkit designed for reliability and scalability, particularly suited for cloud-native environments. It excels in collecting and storing time-series data, making it a popular choice for long-term metrics collection.

Key Features of Prometheus:

1. Data model
2. Metrics collection
3. Time series database
4. Query Language (PromQL)
5. Integration with Visualization Tools

Introduction to Prometheus

Long-Term Storage Considerations:

- While Prometheus is effective for short-term monitoring, it has limitations regarding long-term storage, typically retaining data for only about 15 days by default. To address this limitation, several solutions exist:

1. Thanos
2. Cortex & Mimir

Basics of setting up Prometheus and collecting system data.

To set up Prometheus for collecting system data, follow these basic steps:

1. Download and Install Prometheus

- Visit the Prometheus downloads page and download the latest version for your operating system.

- **Extract the downloaded archive:**

```
tar xvfz prometheus-*.tar.gz
cd prometheus-*
```

Basics of setting up Prometheus and collecting system data.

2. Configure Prometheus

Create or modify the prometheus.yml configuration file in the extracted directory. This file defines how Prometheus scrapes metrics from targets.

Example Configuration:

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']
```

This configuration tells Prometheus to scrape its own metrics every 15 seconds.

Basics of setting up Prometheus and collecting system data.

3. Start Prometheus

- Run the Prometheus server using the following command:

```
./prometheus --config.file=prometheus.yml
```

- By default, Prometheus will be accessible at http://localhost:9090.

4. Collect System Metrics

- To collect system metrics, you can use an exporter like Node Exporter. Install Node Exporter on the same machine or another target machine.
- Start Node Exporter, which typically runs on port 9100.

Basics of setting up Prometheus and collecting system data.

- **Update prometheus.yml to include Node Exporter:**

```
scrape_configs:
  - job_name: 'node'
    static_configs:
      - targets: ['localhost:9100']
```

5. **Access the Web Interface**

- Open a web browser and navigate to **http://localhost:9090** to access the Prometheus web interface. Here you can query metrics and visualize data.

Take A 5-Minute Break!

- Stretch and relax
- Hydrate
- Clear your mind
- Be back in 5 minutes

PW SKILLS

Demonstrating CPU usage optimization

CPU usage optimization

To optimize CPU usage effectively, consider the following strategies based on best practices and techniques:

Techniques for CPU Usage Optimization:

1. Close Unused Applications
2. Manage Startup Programs
3. Update Drivers and Software
4. Optimize Background Processes
5. Adjust Virtual Memory Settings
6. Implement Code Optimizations
7. Reboot Regularly

Offloading workloads to other systems.

PW SKILLS

What is Workload Offloading?

Workload offloading refers to the process of relocating computational tasks from a device with limited processing power (e.g., mobile devices or edge devices) to a more powerful remote processing node (RPN), such as cloud servers or dedicated hardware accelerators.

Benefits of Workload Offloading:

1. Enhanced Performance
2. Resource Optimization
3. Scalability
4. Improved User Experience

Offloading workloads to other systems.

Challenges in Workload Offloading:

1. Network Latency
2. Data Security
3. Dynamic Task Management
4. Cost Implications

Balancing CPU-intensive processes for efficient resource utilization.

Strategies for Balancing CPU-Intensive Processes:

1. Load Balancing
2. Task Prioritization
3. Resource Allocation
4. Dynamic Scaling
5. Monitoring and Analysis
6. Automation Techniques
7. Use of Optimization Software

PW SKILLS

Introducing sysctl for kernel tuning

sysctl for kernel tuning

‘sysctl’ is a command-line utility in Linux that allows system administrators to configure and tune kernel parameters at runtime without the need for a reboot.

Key Features of sysctl:

1. Dynamic Configuration
2. Access to Kernel Parameters
3. Temporary and Permanent Changes
4. Listing Parameters

sysctl for kernel tuning

Example Commands:

- To view all kernel parameters:

```
sysctl -a
```

- To temporarily set a parameter:

```
sysctl -w net.ipv4.ip_forward=1
```

- To make a change permanent:

```
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
sysctl -p
```

Key parameters for networking performance and resource allocation.

PW SKILLS

To optimize networking performance and resource allocation in Linux, several key parameters can be tuned using the sysctl command.

1. TCP Buffer Sizes
2. TCP Read and Write Memory
3. TCP Fast Open
4. TCP Connection Timeout
5. Network Device Backlog
6. ARP Cache Limits

Examples of tuning net.ipv4.tcp-rmem and vm.swappiness

1. Tuning ‘net.ipv4.tcp-rmem’:

The net.ipv4.tcp-rmem parameter controls the size of the TCP receive buffer. It consists of three values: minimum, default, and maximum sizes (in bytes) for the receive buffer used by TCP sockets.

Example Configuration:

- To set the TCP receive buffer to specific values, you can use the following command:

```
sudo sysctl -w net.ipv4.tcp_rmem="4096 87380 16777216"
```

Examples of tuning net.ipv4.tcp-rmem and vm.swappiness

2. Tuning vm.swappiness

The vm.swappiness parameter controls how aggressively the kernel will swap memory pages. It ranges from 0 to 100, where a lower value favors keeping applications in RAM, while a higher value allows more frequent swapping to disk.

Example Configuration:

- To adjust swappiness, you can use the following command:

```
sudo sysctl -w vm.swappiness=10
```

PW SKILLS

Disk partitioning for performance

Disk partitioning

Disk partitioning is the process of dividing a physical disk into separate logical sections, known as partitions.

- Disk partitioning can significantly impact performance by optimizing how data is stored and accessed on a hard drive.

Key Strategies for Disk Partitioning for Performance:

1. Short-Stroking
2. Separate Partitions for OS and Data
3. Optimal Partition Sizing
4. File System Selection
5. Defragmentation
6. Partition Alignment

Strategies for optimizing read-heavy workloads

To optimize read-heavy workloads effectively, several strategies can be employed to enhance performance and resource utilization.

Strategies for Optimizing Read-Heavy Workloads:

1. Database Indexing
2. Read Replicas
3. Caching
4. Content Delivery Networks (CDNs)
5. Load Balancing
6. Denormalization
7. Vertical and Horizontal Scaling

Considerations for write-heavy workloads, including journaling and RAID.

When managing write-heavy workloads, several key considerations are essential for ensuring optimal performance, reliability, and scalability.

Key Considerations for Write-Heavy Workloads:

1. Journaling
2. RAID (Redundant Array of Independent Disks)
3. Database Selection
4. Batch Processing
5. Asynchronous Processing
6. Partitioning and Sharing
7. Caching Strategies

Pop Quiz

Q. What is the primary benefit of partitioning a disk for performance?

A

Reduced seek time for frequently accessed files

B

Increased disk space

Pop Quiz

Q. What is the primary benefit of partitioning a disk for performance?

A

Reduced seek time for frequently accessed files

B

Increased disk space

PW SKILLS

Time for case study!

Important

- Complete the post-class assessment
- Complete assignments (if any)
- Practice the concepts and techniques taught in this session
- Review your lecture notes
- Note down questions and queries regarding this session and consult the teaching assistants

Thanks !