

# Tuning system parameters for better performance in centos

Topics : [Centos Server](#)

Written on [March 05, 2024](#)

Tuning system parameters is essential for optimizing performance and improving the efficiency of your CentOS system. Here are some key system parameters you can tune for better performance:

## 1. Kernel Parameters:

- Adjust kernel parameters in `/etc/sysctl.conf` to optimize kernel behavior and system performance.
- Common kernel parameters to tune include:
  - `vm.swappiness`: Adjusts the tendency of the kernel to swap out memory pages. Lower values reduce swapping.
  - `vm.dirty_ratio` and `vm.dirty_background_ratio`: Control the percentage of system memory used for dirty pages (unflushed data). Adjust these values based on your workload and available memory.
  - `net.core.*`: Tune network-related parameters such as TCP buffer sizes, maximum socket backlog, and connection tracking settings.
  - `fs.*`: Tune filesystem-related parameters such as read-ahead settings, inode and directory cache sizes, and filesystem write behavior.
- Apply changes using the `sysctl -p` command or `sysctl --system` to reload the configuration.

## 2. Filesystem Optimization:

- Optimize filesystem parameters to improve disk I/O performance.
- Choose appropriate filesystem types (e.g., ext4, XFS) based on your workload and requirements.
- Adjust filesystem mount options (e.g., `noatime`, `nodiratime`) to reduce unnecessary disk writes.

## 3. Disk I/O Scheduler:

- Choose the appropriate disk I/O scheduler to optimize disk I/O performance.
- Common I/O schedulers include CFQ (Completely Fair Queuing), Deadline, and NOOP.
- Test different schedulers to find the most suitable one for your workload.

## 4. Network Configuration:

- Tune network parameters to optimize network performance and throughput.
- Adjust TCP/IP stack parameters such as TCP window size, TCP congestion control algorithms, and network buffer sizes.

- Use tools like `ethtool` to optimize network interface settings (e.g., offload features, interrupt moderation).

## 5. Memory Management:

- Optimize memory management settings to improve system responsiveness and memory utilization.
- Adjust swappiness to control the balance between using swap space and system memory.
- Configure transparent huge pages (THP) to improve memory efficiency for memory-intensive workloads.

## 6. Security Configuration:

- Balance security and performance by tuning security-related parameters.
- Adjust SELinux settings, firewall rules, and security modules based on your security requirements and performance impact.
- Fine-tune authentication and authorization mechanisms to minimize overhead.

## 7. Hardware Optimization:

- Optimize hardware settings and configurations to improve system performance.
- Ensure that hardware components (CPU, memory, disks, network interfaces) are properly configured and utilized.
- Monitor hardware health and performance metrics to identify potential issues.

## 8. Monitoring and Optimization:

- Regularly monitor system performance using tools like `top`, `vmstat`, `sar`, and `iostat`.
- Analyze performance metrics and identify bottlenecks that may require further tuning.
- Continuously review and adjust system parameters based on workload changes and performance requirements.

## 9. Testing and Benchmarking:

- Test and benchmark system performance after making changes to system parameters.
- Use tools like `sysbench`, `iperf`, or `bonnie++` to measure performance improvements and validate changes.

## 10. Documentation and Best Practices:

- Document system parameter changes and performance optimizations for future reference.
- Follow best practices and guidelines provided by CentOS documentation, community resources, and industry standards.