

Performance Tuning Guide

Overview

This guide provides comprehensive instructions for optimizing the performance of the HX-Infrastructure-Ansible automation platform, including system-level optimizations, Ansible performance tuning, and application-specific improvements.

System Performance Optimization

CPU Optimization

CPU Scaling and Frequency

```
# Check current CPU governor
ansible all -i inventory.yml -m shell -a "cat /sys/devices/system/cpu/cpu0/cpufreq/
scaling_governor"

# Set performance governor
ansible all -i inventory.yml -m shell -a "echo performance | tee /sys/devices/system/
cpu/cpu*/cpufreq/scaling_governor"

# Install and configure cpufrequtils
ansible-playbook playbooks/performance/configure_cpu_scaling.yml -i inventory.yml \
  --extra-vars "cpu_governor=performance"

# Verify CPU frequency
ansible all -i inventory.yml -m shell -a "cpufreq-info"
```

CPU Affinity and Process Binding

```
# Set CPU affinity for critical processes
ansible all -i inventory.yml -m shell -a "taskset -cp 0-3 $(pgrep nginx)"

# Configure CPU isolation
ansible-playbook playbooks/performance/configure_cpu_isolation.yml -i inventory.yml \
  --extra-vars "isolated_cpus=4-7"

# Monitor CPU usage per core
ansible all -i inventory.yml -m shell -a "mpstat -P ALL 1 5"
```

Memory Optimization

Memory Management

```
# Configure swap settings
ansible-playbook playbooks/performance/configure_swap.yml -i inventory.yml \
  --extra-vars "swappiness=10 vfs_cache_pressure=50"

# Optimize memory allocation
ansible all -i inventory.yml -m sysctl -a "name=vm.overcommit_memory value=1"
ansible all -i inventory.yml -m sysctl -a "name=vm.overcommit_ratio value=80"

# Configure huge pages
ansible-playbook playbooks/performance/configure_hugepages.yml -i inventory.yml \
  --extra-vars "hugepage_size=2048 hugepage_count=1024"
```

Memory Monitoring and Analysis

```
# Monitor memory usage
ansible all -i inventory.yml -m shell -a "free -h && cat /proc/meminfo | grep -E
'MemTotal|MemFree|MemAvailable|Cached|Buffers'"

# Check memory fragmentation
ansible all -i inventory.yml -m shell -a "cat /proc/buddyinfo"

# Analyze memory usage by process
ansible all -i inventory.yml -m shell -a "ps aux --sort=-%mem | head -20"
```

Storage Performance

Disk I/O Optimization

```
# Configure I/O scheduler
ansible-playbook playbooks/performance/configure_io_scheduler.yml -i inventory.yml \
  --extra-vars "scheduler=deadline"

# Optimize filesystem mount options
ansible-playbook playbooks/performance/optimize_filesystem.yml -i inventory.yml \
  --extra-vars "mount_options=noatime,nodiratime,data=writeback"

# Configure disk read-ahead
ansible all -i inventory.yml -m shell -a "blockdev --setra 4096 /dev/sda"
```

Storage Monitoring

```
# Monitor disk I/O
ansible all -i inventory.yml -m shell -a "iostat -x 1 5"

# Check disk usage and performance
ansible all -i inventory.yml -m shell -a "iotop -ao -d 1 -n 5"

# Analyze disk latency
ansible all -i inventory.yml -m shell -a "ioping -c 10 /var/lib/mysql"
```

Network Performance

Network Tuning

```
# Optimize network buffers
ansible-playbook playbooks/performance/optimize_network.yml -i inventory.yml \
  --extra-vars "
    net_core_rmem_max=134217728
    net_core_wmem_max=134217728
    net_ipv4_tcp_rmem='4096 87380 134217728'
    net_ipv4_tcp_wmem='4096 65536 134217728'
  "

# Configure TCP congestion control
ansible all -i inventory.yml -m sysctl -a "name=net.ipv4.tcp_congestion_control
value=bbr"

# Optimize network interface settings
ansible all -i inventory.yml -m shell -a "ethtool -G eth0 rx 4096 tx 4096"
```

Network Monitoring

```
# Monitor network performance
ansible all -i inventory.yml -m shell -a "iftop -t -s 10"

# Check network connections
ansible all -i inventory.yml -m shell -a "ss -tuln | wc -l"

# Analyze network latency
ansible all -i inventory.yml -m shell -a "ping -c 10 8.8.8.8"
```

Ansible Performance Optimization

Ansible Configuration Tuning

ansible.cfg Optimization

```
# /etc/ansible/ansible.cfg
[defaults]
# Increase parallelism
forks = 50
host_key_checking = False
gathering = smart
fact_caching = jsonfile
fact_caching_connection = /tmp/ansible_fact_cache
fact_caching_timeout = 86400

# Optimize SSH connections
[ssh_connection]
ssh_args = -o ControlMaster=auto -o ControlPersist=60s -o ControlPath=/tmp/ansible-
ssh-%h-%p-%r
pipelining = True
retries = 3
```

Inventory Optimization

```
# Use static inventory for better performance
ansible-playbook playbooks/performance/optimize_inventory.yml -i inventory.yml

# Group hosts efficiently
# In inventory.yml:
[webservers]
web[01:10].example.com

[databases]
db[01:03].example.com

[all:vars]
ansible_ssh_common_args='-o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null'
```

Playbook Performance Optimization

Task Optimization

```
# Optimize task execution
- name: Install packages efficiently
  package:
    name:
      - nginx
      - mysql-server
      - redis-server
    state: present
# Instead of multiple package tasks

- name: Use async for long-running tasks
  command: /usr/bin/long-running-command
  async: 3600
  poll: 0
  register: long_task

- name: Check async task status
  async_status:
    jid: "{{ long_task.ansible_job_id }}"
  register: job_result
  until: job_result.finished
  retries: 30
  delay: 10
```

Fact Gathering Optimization

```
# Disable fact gathering when not needed
- hosts: all
  gather_facts: no
  tasks:
    - name: Simple task without facts
      ping:

# Gather only required facts
- hosts: all
  gather_facts: yes
  tasks:
    - name: Gather minimal facts
      setup:
        filter: ansible_os_family
```

Loop Optimization

```
# Use efficient loops
- name: Create users efficiently
  user:
    name: "{{ item.name }}"
    groups: "{{ item.groups }}"
    state: present
  loop:
    - { name: user1, groups: developers }
    - { name: user2, groups: admins }
# Instead of with_items for better performance

# Use loop control for large datasets
- name: Process large dataset
  debug:
    msg: "Processing {{ item }}"
  loop: "{{ large_list }}"
  loop_control:
    pause: 1
    label: "{{ item.name }}"
```

Parallel Execution

Serial Execution Control

```
# Control parallel execution
- hosts: webservers
  serial: 2 # Process 2 hosts at a time
  tasks:
    - name: Rolling update
      service:
        name: nginx
        state: restarted

# Percentage-based serial execution
- hosts: webservers
  serial: "30%" # Process 30% of hosts at a time
  tasks:
    - name: Update application
      command: /usr/local/bin/update-app
```

Strategy Optimization

```
# Use free strategy for independent tasks
- hosts: all
  strategy: free
  tasks:
    - name: Independent task
      command: /usr/bin/independent-command

# Use linear strategy for dependent tasks
- hosts: all
  strategy: linear
  tasks:
    - name: Dependent task
      command: /usr/bin/dependent-command
```

Application Performance Optimization

Web Server Optimization

Nginx Performance Tuning

```
# Deploy optimized Nginx configuration
ansible-playbook playbooks/performance/optimize_nginx.yml -i inventory.yml \
  --extra-vars "
    worker_processes=auto
    worker_connections=4096
    keepalive_timeout=65
    client_max_body_size=100m
  "

# Configure Nginx caching
ansible-playbook playbooks/performance/configure_nginx_caching.yml -i inventory.yml

# Monitor Nginx performance
ansible all -i inventory.yml -m uri -a "url=http://localhost/nginx_status"
```

Apache Performance Tuning

```
# Optimize Apache configuration
ansible-playbook playbooks/performance/optimize_apache.yml -i inventory.yml \
  --extra-vars "
    max_request_workers=400
    threads_per_child=25
    server_limit=16
  "

# Enable Apache modules for performance
ansible all -i inventory.yml -m apache2_module -a "name=mod_deflate state=present"
ansible all -i inventory.yml -m apache2_module -a "name=mod_expires state=present"
```

Database Performance

MySQL Optimization

```
# Deploy optimized MySQL configuration
ansible-playbook playbooks/performance/optimize_mysql.yml -i inventory.yml \
  --extra-vars "
    innodb_buffer_pool_size=2G
    innodb_log_file_size=256M
    query_cache_size=128M
    max_connections=200
  "

# Optimize MySQL queries
ansible all -i inventory.yml -m mysql_query -a "
  login_user=root
  login_password={{ mysql_root_password }}
  query='ANALYZE TABLE user_table'
"

# Monitor MySQL performance
ansible all -i inventory.yml -m shell -a "mysqladmin -u root -
p{{ mysql_root_password }} extended-status | grep -E 'Queries|Threads|Connections'"

```

PostgreSQL Optimization

```
# Optimize PostgreSQL configuration
ansible-playbook playbooks/performance/optimize_postgresql.yml -i inventory.yml \
  --extra-vars "
    shared_buffers=256MB
    effective_cache_size=1GB
    work_mem=4MB
    maintenance_work_mem=64MB
  "

# Analyze PostgreSQL performance
ansible all -i inventory.yml -m postgresql_query -a "
  db=mydb
  query='SELECT * FROM pg_stat_activity'
"

```

Caching Optimization

Redis Configuration

```
# Deploy optimized Redis configuration
ansible-playbook playbooks/performance/optimize_redis.yml -i inventory.yml \
  --extra-vars "
    maxmemory=2gb
    maxmemory_policy=allkeys-lru
    save_config='900 1 300 10 60 10000'
  "

# Monitor Redis performance
ansible all -i inventory.yml -m shell -a "redis-cli info stats"

```

Memcached Configuration

```
# Optimize Memcached settings
ansible-playbook playbooks/performance/optimize_memcached.yml -i inventory.yml \
  --extra-vars "
    memory_limit=512
    max_connections=1024
    threads=4
  "

# Monitor Memcached performance
ansible all -i inventory.yml -m shell -a "echo 'stats' | nc localhost 11211"
```

Performance Monitoring and Analysis

Performance Metrics Collection

System Metrics

```
# Deploy performance monitoring
ansible-playbook playbooks/performance/deploy_performance_monitoring.yml -i inventory.yml

# Collect system performance data
ansible-playbook playbooks/performance/collect_performance_data.yml -i inventory.yml

# Generate performance baseline
./scripts/performance/generate_baseline.sh
```

Application Metrics

```
# Monitor application performance
ansible-playbook playbooks/performance/monitor_application_performance.yml -i inventory.yml

# Collect application metrics
curl http://app-server:8080/metrics

# Analyze response times
ansible all -i inventory.yml -m uri -a "url=http://localhost/api/health" -m time
```


Performance Testing

Load Testing

```
# Deploy load testing tools
ansible-playbook playbooks/performance/deploy_load_testing.yml -i inventory.yml

# Run load tests
ansible-playbook playbooks/performance/run_load_tests.yml -i inventory.yml \
  --extra-vars "
    target_url=http://app-server
    concurrent_users=100
    test_duration=300
  "

# Analyze load test results
./scripts/performance/analyze_load_test_results.sh
```

Stress Testing

```
# Run CPU stress test
ansible all -i inventory.yml -m shell -a "stress --cpu 4 --timeout 60s"

# Run memory stress test
ansible all -i inventory.yml -m shell -a "stress --vm 2 --vm-bytes 1G --timeout 60s"

# Run I/O stress test
ansible all -i inventory.yml -m shell -a "stress --io 4 --timeout 60s"
```

Performance Analysis Tools

Profiling Tools

```
# Install profiling tools
ansible-playbook playbooks/performance/install_profiling_tools.yml -i inventory.yml

# Profile application performance
ansible all -i inventory.yml -m shell -a "perf record -g ./application"
ansible all -i inventory.yml -m shell -a "perf report"

# Use strace for system call analysis
ansible all -i inventory.yml -m shell -a "strace -c -p $(pgrep nginx)"
```

Performance Visualization

```
# Generate performance graphs
ansible-playbook playbooks/performance/generate_performance_graphs.yml -i inventory.yml

# Create performance dashboard
ansible-playbook playbooks/performance/create_performance_dashboard.yml -i inventory.yml

# Export performance data
./scripts/performance/export_performance_data.sh
```

Performance Optimization Strategies

Capacity Planning

Resource Forecasting

```
# Analyze resource trends
ansible-playbook playbooks/performance/analyze_resource_trends.yml -i inventory.yml

# Generate capacity forecast
./scripts/performance/generate_capacity_forecast.py

# Plan resource scaling
ansible-playbook playbooks/performance/plan_resource_scaling.yml -i inventory.yml
```

Scaling Strategies

```
# Implement horizontal scaling
ansible-playbook playbooks/performance/implement_horizontal_scaling.yml -i inventory.yml

# Configure auto-scaling
ansible-playbook playbooks/performance/configure_auto_scaling.yml -i inventory.yml \
  --extra-vars "
    scale_up_threshold=80
    scale_down_threshold=20
    min_instances=2
    max_instances=10
  "
```

Performance Optimization Workflow

Continuous Performance Monitoring

```
# Set up continuous monitoring
ansible-playbook playbooks/performance/setup_continuous_monitoring.yml -i inventory.yml

# Configure performance alerts
ansible-playbook playbooks/performance/configure_performance_alerts.yml -i inventory.yml

# Automated performance optimization
ansible-playbook playbooks/performance/automated_optimization.yml -i inventory.yml
```

Performance Regression Testing

```
# Run performance regression tests
ansible-playbook playbooks/performance/run_regression_tests.yml -i inventory.yml

# Compare performance baselines
./scripts/performance/compare_baselines.sh

# Generate performance regression report
./scripts/performance/generate_regression_report.sh
```

Troubleshooting Performance Issues

Common Performance Problems

High CPU Usage

```
# Identify CPU-intensive processes
ansible all -i inventory.yml -m shell -a "top -bn1 | head -20"

# Analyze CPU usage patterns
ansible all -i inventory.yml -m shell -a "sar -u 1 10"

# Optimize CPU-intensive tasks
ansible-playbook playbooks/performance/optimize_cpu_tasks.yml -i inventory.yml
```

Memory Issues

```
# Identify memory leaks
ansible all -i inventory.yml -m shell -a "ps aux --sort=-%mem | head -10"

# Analyze memory usage over time
ansible all -i inventory.yml -m shell -a "sar -r 1 10"

# Optimize memory usage
ansible-playbook playbooks/performance/optimize_memory_usage.yml -i inventory.yml
```

I/O Bottlenecks

```
# Identify I/O bottlenecks
ansible all -i inventory.yml -m shell -a "iotop -ao -d 1 -n 5"

# Analyze disk performance
ansible all -i inventory.yml -m shell -a "iostat -x 1 5"

# Optimize I/O performance
ansible-playbook playbooks/performance/optimize_io_performance.yml -i inventory.yml
```

Performance Debugging

Application Performance Issues

```
# Debug application performance
ansible-playbook playbooks/performance/debug_application_performance.yml -i inventory.yml

# Analyze application logs
ansible all -i inventory.yml -m shell -a "grep -E 'slow|timeout|error' /var/log/application.log"

# Profile application code
ansible all -i inventory.yml -m shell -a "python -m cProfile application.py"
```

Database Performance Issues

```
# Analyze slow queries
ansible all -i inventory.yml -m mysql_query -a "
    login_user=root
    login_password={{ mysql_root_password }}
    query='SELECT * FROM mysql.slow_log ORDER BY start_time DESC LIMIT 10'
"

# Optimize database queries
ansible-playbook playbooks/performance/optimize_database_queries.yml -i inventory.yml

# Monitor database performance
ansible all -i inventory.yml -m shell -a "mysqladmin -u root -
p{{ mysql_root_password }} processlist"
```

Best Practices

Performance Optimization Guidelines

- **Measure before optimizing:** Establish baselines
- **Focus on bottlenecks:** Identify and address the biggest constraints
- **Test changes:** Validate optimizations in staging environment
- **Monitor continuously:** Track performance metrics over time
- **Document changes:** Keep records of optimization efforts

Performance Testing Best Practices

- **Realistic test data:** Use production-like data volumes
- **Gradual load increase:** Ramp up load gradually
- **Multiple test scenarios:** Test different usage patterns
- **Environment consistency:** Use consistent test environments
- **Regular testing:** Perform regular performance tests

Monitoring and Alerting

- **Key performance indicators:** Monitor critical metrics
- **Proactive alerting:** Set up alerts for performance degradation
- **Trend analysis:** Analyze performance trends over time
- **Capacity planning:** Plan for future resource needs
- **Performance budgets:** Set performance targets and budgets

Related Documentation

- [Monitoring Guide](#) (MONITORING_GUIDE.md)
- [Troubleshooting Guide](#) (../runbooks/TROUBLESHOOTING_GUIDE.md)
- [Deployment Runbook](#) (../runbooks/DEPLOYMENT_RUNBOOK.md)
- [Security Procedures](#) (../runbooks/SECURITY_PROCEDURES.md)