IT Operations Troubleshooting Guide

Table of Contents

CPU Utilization Issues

Symptoms

- System running slowly or becoming unresponsive
- High CPU usage (>80% consistently)
- Applications taking longer to respond
- Server overheating warnings

Diagnostic Steps

Windows

```cmd
# Check CPU usage
tasklist /svc
wmic process get processid,parentprocessid,name,executablepath

# Performance monitoring
perfmon
typeperf "\\Processor(_Total)\\% Processor Time" -sc 10
```

Linux

```bash
```

```
# Check CPU usage
top -c
htop
ps aux --sort=-%cpu | head -10

# Detailed process analysis
pidstat 1 5
sar -u 1 5
```

Common Causes and Solutions

1. Runaway Processes

- **Cause**: Application bugs, infinite loops, or malware
- **Solution**:
  - Identify the process using `top` or Task Manager
  - Kill the problematic process: `kill -9 [PID]` or End Task
  - Check application logs for errors
  - Update or reinstall the problematic application

2. Insufficient Hardware Resources

- **Cause**: Too many applications running simultaneously
- **Solution**:
  - Scale up (add more CPU cores)
  - Scale out (distribute load across multiple servers)
  - Implement load balancing

3. Background Services

- **Cause**: Unnecessary services consuming CPU
- **Solution**:
  - Review running services: `systemctl list-units --type=service` (Linux) or `services.msc` (Windows)
  - Disable non-essential services
  - Schedule resource-intensive tasks during off-peak hours

Prevention

- Implement CPU usage alerts (threshold: 80% for 5+ minutes)
- Regular performance baseline reviews
- Capacity planning based on growth projections

Memory Leaks

Symptoms

- Gradually increasing memory usage over time
- System becoming slower
- Out of memory errors
- Application crashes
- Swap usage increasing

Diagnostic Steps

Windows

```cmd
# Memory usage monitoring
tasklist /m
wmic process get processid,name,workingsetsize,privatebytes

# Performance counters
typeperf "\\Memory\\Available MBytes" -sc 20
typeperf "\\Process(*)\\Private Bytes" -sc 10
```

Linux

```bash
# Memory monitoring
free -h -s 5
vmstat 1 10
cat /proc/meminfo

# Per-process memory usage
ps aux --sort=-%mem | head -10
pmap -x [PID]
smem -s uss
```

Common Causes and Solutions

1. Application Memory Leaks

- **Cause**: Poor programming practices, unreleased objects
- **Solution**:
  - Restart the affected application

- Contact application vendor for patches
- Implement application restarts on schedule
- Use application profilers for development

2. Cache Growth

- **Cause**: Unbounded caches in applications or databases
- **Solution**:
  - Configure cache limits
  - Implement cache eviction policies
  - Clear caches: `echo 1 > /proc/sys/vm/drop_caches` (Linux)

3. Memory Fragmentation

- **Cause**: Long-running processes with frequent allocations
- **Solution**:
  - Restart services periodically
  - Tune memory allocators
  - Monitor memory fragmentation metrics

Memory Leak Investigation Tools

- **Windows**: Process Explorer, Application Verifier, Debug Heap
- **Linux**: Valgrind, AddressSanitizer, tcmalloc

Prevention

- Set memory usage alerts (threshold: 85% physical memory)
- Regular application restarts for known leaky applications
- Code reviews focusing on memory management
- Automated memory testing in CI/CD pipelines

Database Connection Errors

Common Error Messages

- "Connection timeout"
- "Too many connections"
- "Connection refused"
- "Database server not found"

- "Authentication failed"

Diagnostic Steps

```sql
-- Check active connections (MySQL)
SHOW PROCESSLIST;
SHOW STATUS LIKE 'Threads_connected';
SHOW VARIABLES LIKE 'max_connections';

-- Check active connections (PostgreSQL)
SELECT * FROM pg_stat_activity;
SELECT count(*) FROM pg_stat_activity;

-- Check active connections (SQL Server)
SELECT * FROM sys.dm_exec_sessions;
SELECT COUNT(*) FROM sys.dm_exec_sessions WHERE is_user_process = 1;
```

Network Connectivity Tests

```bash
# Test database port connectivity
telnet [db_server] [port]
nc -zv [db_server] [port]

# DNS resolution check
nslookup [db_server]
dig [db_server]
```

Common Causes and Solutions

1. Connection Pool Exhaustion

- **Cause**: Application not releasing connections, connection leaks
- **Solution**:
  - Increase connection pool size temporarily
  - Audit application code for connection leaks
  - Implement connection timeouts
  - Monitor connection pool metrics

2. Database Server Overload

- **Cause**: Too many concurrent connections, resource exhaustion

- **Solution**:
  - Increase `max_connections` setting
  - Optimize slow queries
  - Implement connection pooling (PgBouncer, MySQL Proxy)
  - Scale database resources

3. Network Issues

- **Cause**: Firewall rules, network partitions, DNS issues
- **Solution**:
  - Verify firewall rules allow database port
  - Check network latency: `ping [db_server]`
  - Verify DNS resolution
  - Test with IP address instead of hostname

4. Authentication Problems

- **Cause**: Invalid credentials, account lockouts, privilege issues
- **Solution**:
  - Verify username/password
  - Check account status and permissions
  - Review database authentication logs
  - Test connection with database client tools

Database-Specific Troubleshooting

MySQL

```sql
-- Check error log location
SHOW VARIABLES LIKE 'log_error';

-- Check connection limits
SHOW VARIABLES LIKE '%connection%';

-- Kill problematic connections
KILL [connection_id];
```

PostgreSQL

```sql
```

```sql
-- Check configuration
SHOW config_file;
SHOW hba_file;

-- Terminate connections
SELECT pg_terminate_backend(pid) FROM pg_stat_activity WHERE datname = 'database_name';
```

SQL Server

```sql
-- Check connection info
SELECT @@SERVERNAME, @@VERSION;

-- Kill session
KILL [session_id];
```

Prevention

- Connection pool monitoring and alerting
- Regular database maintenance (statistics updates, index maintenance)
- Connection timeout configuration
- Database performance monitoring

Network Connectivity Issues

Symptoms

- Timeouts when accessing services
- Intermittent connectivity
- Slow response times
- DNS resolution failures

Diagnostic Commands

Basic Connectivity

```bash
```

```
# Test basic connectivity
ping [hostname/IP]
traceroute [hostname/IP]  # Linux
tracert [hostname/IP]     # Windows

# Test specific ports
telnet [hostname] [port]
nc -zv [hostname] [port]  # Linux
```

DNS Testing

```bash
# DNS lookup
nslookup [hostname]
dig [hostname]
host [hostname]

# Check DNS servers
cat /etc/resolv.conf  # Linux
ipconfig /all         # Windows
```

Network Interface Analysis

```bash
# Interface status (Linux)
ip addr show
ifconfig
netstat -i

# Interface status (Windows)
ipconfig /all
netsh interface show interface
```

Common Issues and Solutions

1. DNS Resolution Problems

- **Symptoms**: "Host not found" errors
- **Solutions**:
  - Check DNS server configuration
  - Flush DNS cache: `ipconfig /flushdns` (Windows), `sudo systemctl restart systemd-resolved` (Linux)
  - Try alternative DNS servers (8.8.8.8, 1.1.1.1)

- Check hosts file for conflicts

2. Firewall Blocking

- **Symptoms**: Connection timeouts, specific ports blocked
- **Solutions**:
  - Check firewall rules: `iptables -L` (Linux), `netsh advfirewall show allprofiles` (Windows)
  - Temporary disable firewall for testing
  - Add specific rules for required ports

3. Network Congestion

- **Symptoms**: High latency, packet loss
- **Solutions**:
  - Monitor bandwidth usage: `iftop`, `nethogs` (Linux)
  - Check for network loops
  - Implement Quality of Service (QoS) policies

Advanced Network Diagnostics

```bash
# Packet capture
tcpdump -i any -n host [hostname]
wireshark  # GUI tool

# Network statistics
netstat -s
ss -s    # Linux modern alternative

# Bandwidth testing
iperf3 -c [server]  # Client
iperf3 -s           # Server
```

Disk Space and I/O Problems

Symptoms

- "Disk full" errors
- Slow file operations
- Application crashes due to inability to write files
- System becoming unresponsive during I/O operations

Disk Space Diagnostics

Check Disk Usage

```bash
# Linux
df -h                # File system usage
du -sh /*            # Directory usage
du -h --max-depth=1  # Subdirectory sizes
find / -size +100M   # Find large files

# Windows
dir /s               # Directory sizes
fsutil volume diskfree C:  # Free space
```

I/O Performance Monitoring

```bash
# Linux
iostat -x 1 5        # I/O statistics
iotop                # Per-process I/O
hdparm -tT /dev/sda  # Disk speed test

# Windows
typeperf "\\PhysicalDisk(*)\\% Disk Time" -sc 10
diskpart             # Disk management
```

Common Solutions

1. Clean Up Disk Space

```bash
# Linux cleanup
sudo apt autoremove    # Remove unused packages
sudo apt autoclean     # Clean package cache
journalctl --vacuum-time=7d  # Clean systemd logs

# Log rotation
sudo logrotate -f /etc/logrotate.conf

# Windows cleanup
cleanmgr             # Disk Cleanup utility
sfc /scannow         # System file checker
```

2. Identify Space Consumers

```bash
# Find largest directories
du -h / | sort -rh | head -20

# Find old files
find /var/log -name "*.log" -mtime +30 -ls

# Find duplicate files
fdupes -r /home/
```

3. I/O Performance Issues

- Check for failing drives: `smartctl -a /dev/sda`
- Monitor disk queue lengths
- Consider SSD migration for high I/O workloads
- Implement proper backup and archiving strategies

Service and Application Failures

Service Management

Linux (systemd)

```bash
# Check service status
systemctl status [service_name]
systemctl list-units --failed

# Service logs
journalctl -u [service_name] -f
journalctl -u [service_name] --since "1 hour ago"

# Service operations
systemctl start [service_name]
systemctl stop [service_name]
systemctl restart [service_name]
systemctl enable [service_name]
```

Windows

```cmd
```

```
# Service management
sc query [service_name]
sc start [service_name]
sc stop [service_name]

# Event logs
eventvwr.msc
wevtutil qe System /c:10 /rd:true /f:text
```

Application Troubleshooting Steps

1. Check Application Logs

- Application-specific log files
- System event logs
- Error patterns and frequency

2. Verify Dependencies

- Database connectivity
- External service dependencies
- Required files and permissions

3. Resource Availability

- Memory usage
- Disk space
- Network connectivity
- License availability

4. Configuration Issues

- Configuration file syntax
- Environment variables
- Path settings
- Security permissions

Common Recovery Actions

1. Service Restart

```
bash
```

```
# Graceful restart
systemctl restart [service_name]

# Force restart if needed
systemctl kill [service_name]
systemctl start [service_name]
```

2. Clear Temporary Files

```
bash

# Clear application temp files
rm -rf /tmp/app_temp/*
rm -rf /var/cache/app/*
```

3. Reset Configuration

- Backup current configuration
- Restore known good configuration
- Gradually apply changes to identify issues

Performance Monitoring Best Practices

Essential Metrics to Monitor

System Level:

- CPU utilization (per core and aggregate)
- Memory usage (physical and swap)
- Disk I/O (IOPS, throughput, latency)
- Network utilization (bandwidth, packets, errors)

Application Level:

- Response times
- Request throughput
- Error rates
- Queue depths

Monitoring Tools

Open Source

- **Nagios**: Infrastructure monitoring

- **Zabbix**: Network and application monitoring
- **Prometheus + Grafana**: Metrics collection and visualization
- **ELK Stack**: Log aggregation and analysis

Commercial

- **SolarWinds**: Comprehensive infrastructure monitoring
- **Datadog**: Cloud-scale monitoring
- **New Relic**: Application performance monitoring

Alerting Guidelines

Critical Alerts (Immediate Response):

- Service down
- Disk space >95%
- Memory usage >95%
- Database connection failures

Warning Alerts (Response within hours):

- CPU usage >80% for 10+ minutes
- Memory usage >85%
- Disk space >80%
- High error rates

Info Alerts (Response within days):

- Performance degradation trends
- Capacity planning thresholds
- Security events

Documentation Standards

Runbooks Should Include:

- Step-by-step troubleshooting procedures
- Common symptoms and solutions
- Escalation procedures
- Required tools and access
- Success criteria for each step

Incident Documentation:

- Date/time and duration
- Symptoms and impact
- Root cause analysis
- Resolution steps taken
- Prevention measures implemented

Emergency Contacts and Escalation

Internal Contacts

- **Level 1 Support**: [Phone/Email]
- **Level 2 Support**: [Phone/Email]
- **Database Administrator**: [Phone/Email]
- **Network Administrator**: [Phone/Email]
- **Security Team**: [Phone/Email]
- **Management**: [Phone/Email]

External Vendors

- **Hardware Vendor**: [Contact Info]
- **Software Vendor**: [Contact Info]
- **ISP/Network Provider**: [Contact Info]
- **Cloud Provider**: [Contact Info]

Escalation Criteria

- **Immediate**: Complete system outage, security breach
- **Within 1 Hour**: Service degradation affecting users
- **Within 4 Hours**: Performance issues, minor outages
- **Within 24 Hours**: Maintenance issues, planning items

Quick Reference Commands

System Information

```bash
```

```
# Linux
uname -a        # System information
lscpu           # CPU information
free -h         # Memory information
lsblk           # Block devices
ip addr         # Network interfaces

# Windows
systeminfo      # System information
wmic cpu get name # CPU information
wmic memorychip get capacity # Memory information
```

Process Management

bash

```
# Linux
ps aux          # Process list
top             # Real-time processes
kill -9 [PID]   # Force kill process
killall [name]  # Kill by name

# Windows
tasklist        # Process list
taskmgr         # Task Manager
taskkill /PID [PID] # Kill process
```

Network Troubleshooting

bash

```
# Universal
ping [host]     # Test connectivity
traceroute [host] # Trace route
netstat -an     # Network connections
nslookup [host]  # DNS lookup
```

This troubleshooting guide should be regularly updated based on new issues encountered and lessons learned. Keep local copies updated and ensure all team members have access to the latest version.

Document Version: 1.0
Last Updated: August 2025
Next Review: February 2026