Performance Analysis and Online Resource Management of a VM System

GRS Project - Team 23

Priyanshu Kumar Rai, Aman Yadav, Ritesh Gupta and Vanshika Jain

Abstract

- Investigates deployment, performance analysis, and resource management of VMs.
- Uses QEMU/KVM and libvirt on a Linux host.
- Benchmarks include miniFE, miniQMC, HPCG, and others.
- Focus on dynamic resource allocation for optimization.

Problem Statement

The project focuses on centralized monitoring, performance analysis, and dynamic resource management in virtualized environments. By studying three Alpine Linux VMs on a Linux host, the system aims to:

- Monitor real-time metrics for CPU, memory, network, and disk I/O.
- Execute performance benchmarks to analyze resource utilization under varying workloads.
- Dynamically adjust CPU and memory allocation to optimize performance and resource efficiency.

Project Goals

Monitoring

- Centralized monitoring of CPU, memory, network, and disk I/O.
- Real-time metrics collection and visualization.

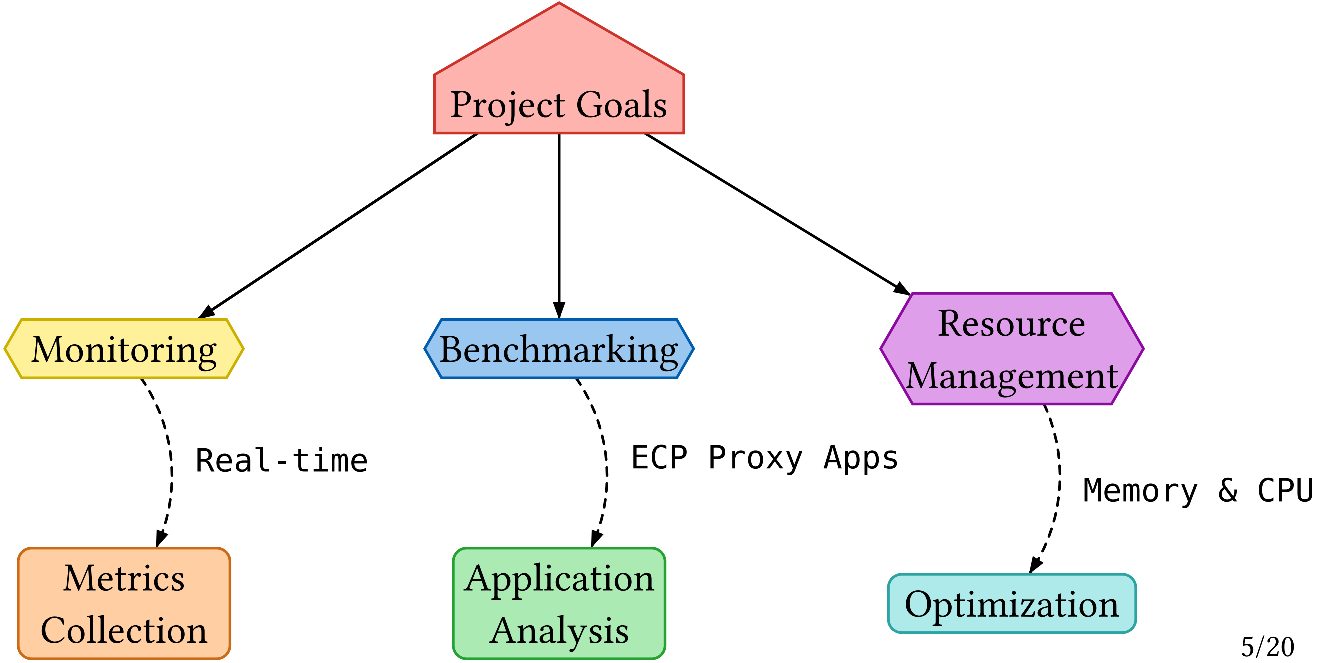
Benchmarking

- Execute benchmarks like HPCCG, XSBench, miniFE, and HPCG.
- Analyze performance under resource constraints.

Resource Management

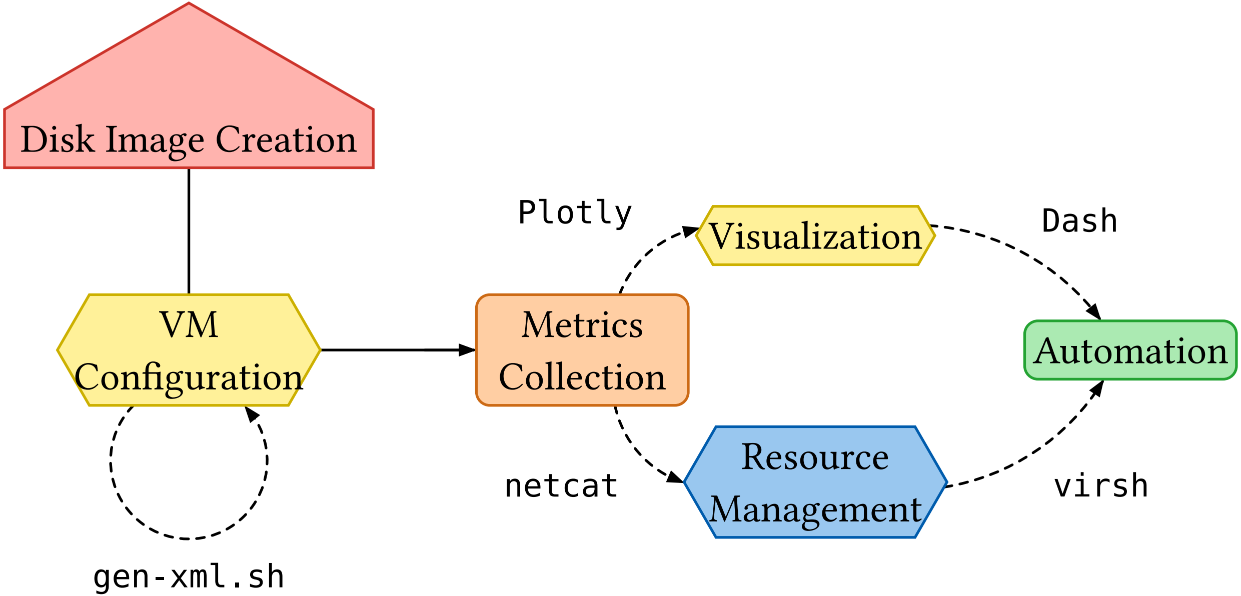
- Dynamic memory ballooning and CPU core allocation.
- Optimize storage using pooling and sharing strategies.

For the eyes


Project Goals
Monitoring
Benchmarking
Resource Management
Real-time
ECP Proxy Apps
Memory & CPU
Metrics Collection
Application Analysis
Optimization

Project Structure

Summary



Disk Image Creation

VM Configuration

Metrics Collection

Visualization

Resource Management

Automation

`gen-xml.sh`

`Plotly`

`netcat`

`Dash`

`virsh`

Components

File Name	Purpose
`alpine-base.qcow2`	Base Image
`alpine-1.qcow2`, `alpine-2.qcow2`, `alpine-3.qcow2`	VM Images
`alpine.xml`	Base XML Template
`alpine-vm-1.xml`, `alpine-vm-2.xml`, `alpine-vm-3.xml`	VM Configurations
`gen-xml.sh`	XML Generation Script
`report-memory.sh`	Metrics Script (VM ⟶ Host)

Components (ii)

`plot_adv.py`	Listener, Visualizer and Monitor
`runall.sh`	Runner Script (VM)
`memory_hog.c`	Workload Simulation (VM)
`metrics_data/`	Metrics Storage (Host)
`miniQMC/`	Quantum Monte Carlo Simulation
`MiniFE/`	Finite Element Solver
`HPCG/`	High-Performance Computing Graph

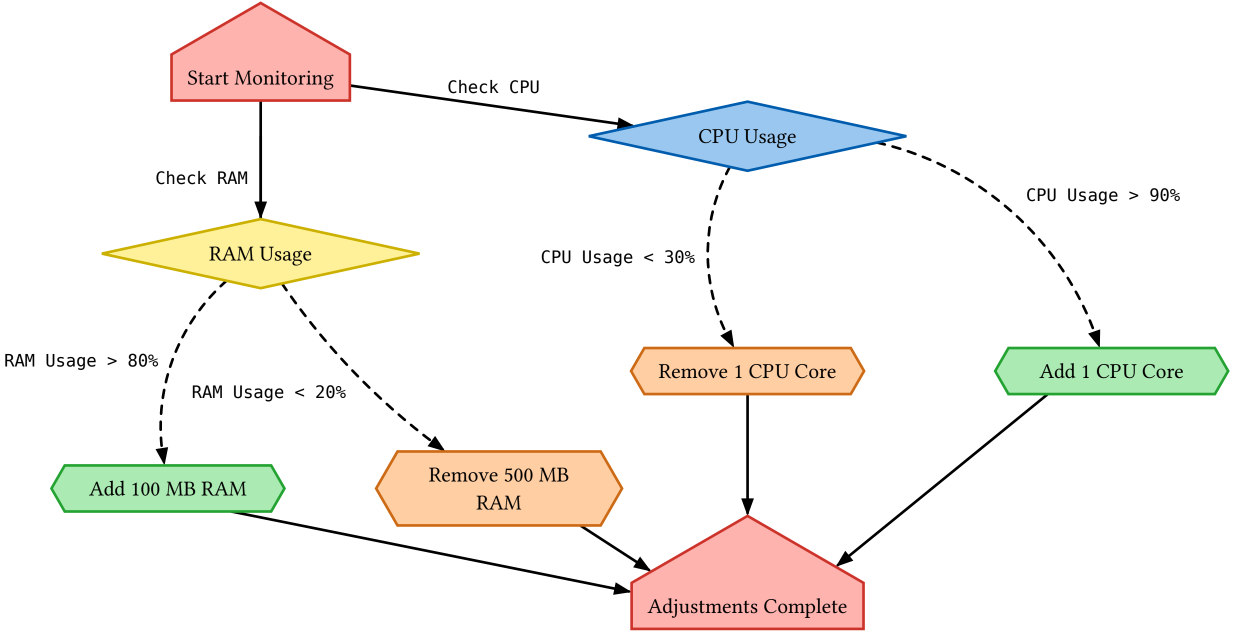
Resource Management

Dynamic RAM Adjustment

- Increase RAM by 100 MB if usage > 80%.
- Decrease RAM by 500 MB if usage < 20%.
- Uses libvirt APIs for live adjustments.

Dynamic CPU Core Adjustment

- Add CPU core if usage > 90%.
- Remove CPU core if usage < 30%.
- Adjusts cores dynamically (up to 2 cores).

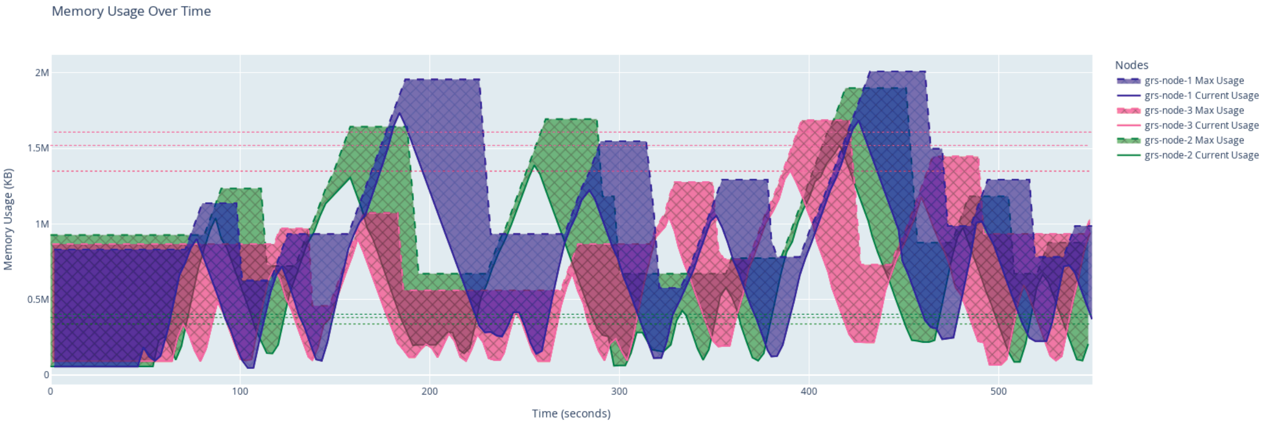

Start Monitoring
Check CPU
CPU Usage
Check RAM
RAM Usage
CPU Usage > 90%
CPU Usage < 30%
RAM Usage > 80%
RAM Usage < 20%
Remove 1 CPU Core
Add 1 CPU Core
Add 100 MB RAM
Remove 500 MB RAM
Adjustments Complete

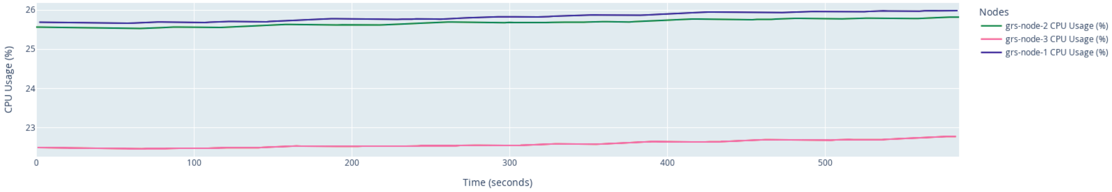
Metrics Reporting

- Memory usage (total and used).
- CPU usage (% utilization).
- Disk I/O (read/write activity).
- Network usage (bytes received/transmitted).

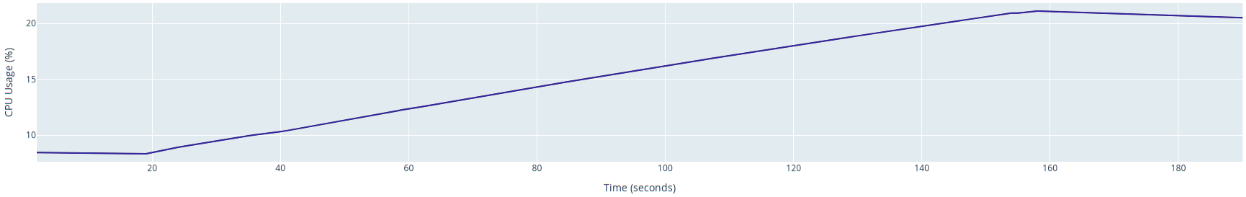
We used:

- Real-time graphs for memory, CPU, disk I/O, and network usage.
- Threshold indicators for memory and CPU.
- Historical data stored in CSV files.

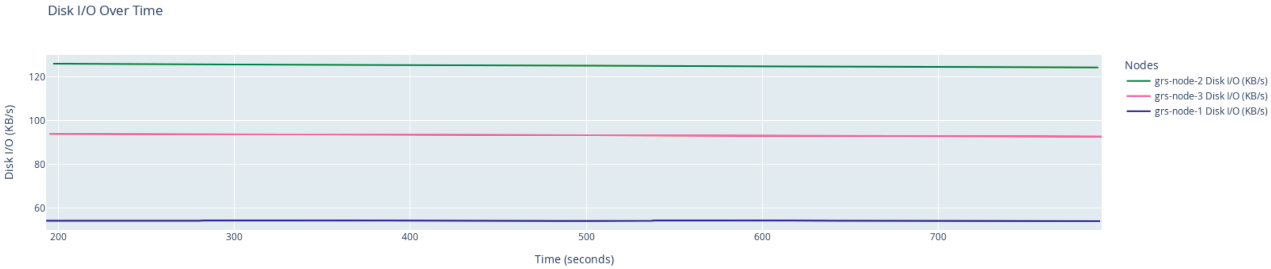

Memory Usage Over Time
Memory Usage (KB)
0
0.5M
1M
1.5M
2M
0
100
200
300
400
500
Time (seconds)
Nodes
grs-node-1 Max Usage
grs-node-1 Current Usage
grs-node-3 Max Usage
grs-node-3 Current Usage
grs-node-2 Max Usage
grs-node-2 Current Usage



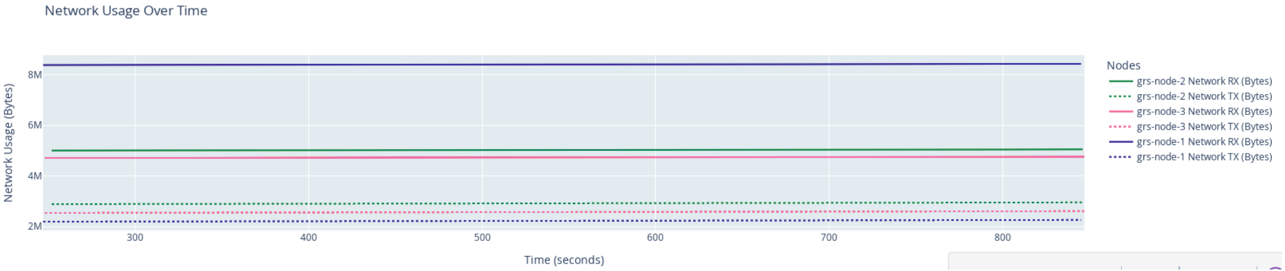
CPU Usage Over Time



CPU Usage Over Time

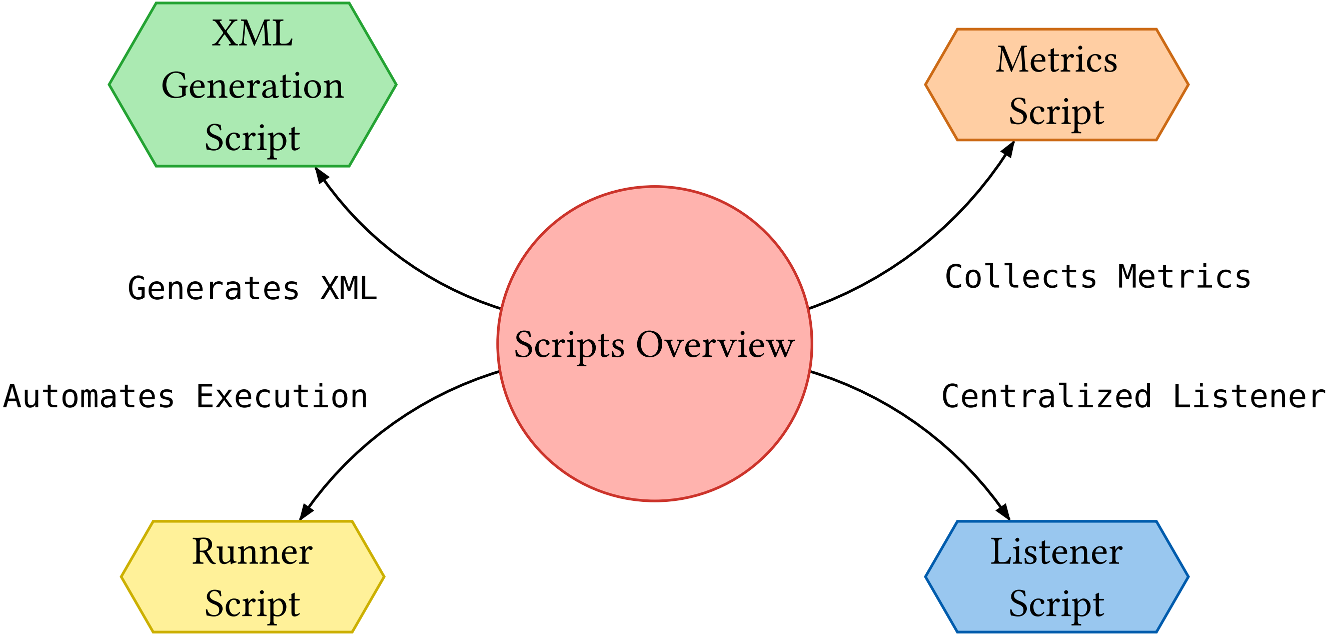


Disk I/O Over Time



Network Usage Over Time

Automation


XML Generation Script
Metrics Script
Generates XML
Collects Metrics
Scripts Overview
Automates Execution
Centralized Listener
Runner Script
Listener Script

Project Outcomes

- Automated VM Deployment.
- Real-time Monitoring.
- Dynamic Resource Management.
- Performance Benchmarking.
- Visualization Dashboard.
- Optimized Resource Allocation.

Future Work

- Extend resource types: Support storage and network bandwidth optimization.
- Advanced Benchmarking: Integrate more comprehensive benchmarking tools.
- Machine Learning: Predict resource usage and optimize allocation.

Thank You