



Monitoring HPC Clusters Using Sqr1

Delaney Gill-Sommerhauser, Danielle Larson, Christopher Moussa



Abstract

Sqr1 is a computer inventory application written in Golang that gathers and exports extensive system information to an outside source for analysis and visualization of the data. The purpose of this project is to provide accessible/scalable data to users who are interested in finding out information about their Linux systems in order to better understand efficient utilization of available nodes in a cluster.

Introduction

In large clusters of computers, it can be difficult to extract information concerning the physical node; however, this process can be automated through a program. Previously existing packages provided select information about the node, such as network name, number of cores, free and used memory, hard drive type, kernel version, and type of operating system, but none contained exhaustive system specifications. Our goal was to create a single program that provided all of this information, called sqr1. We aim to research the possibilities of implementation of sqr1 in large-scale systems to find the most efficient and practical way to monitor systems.

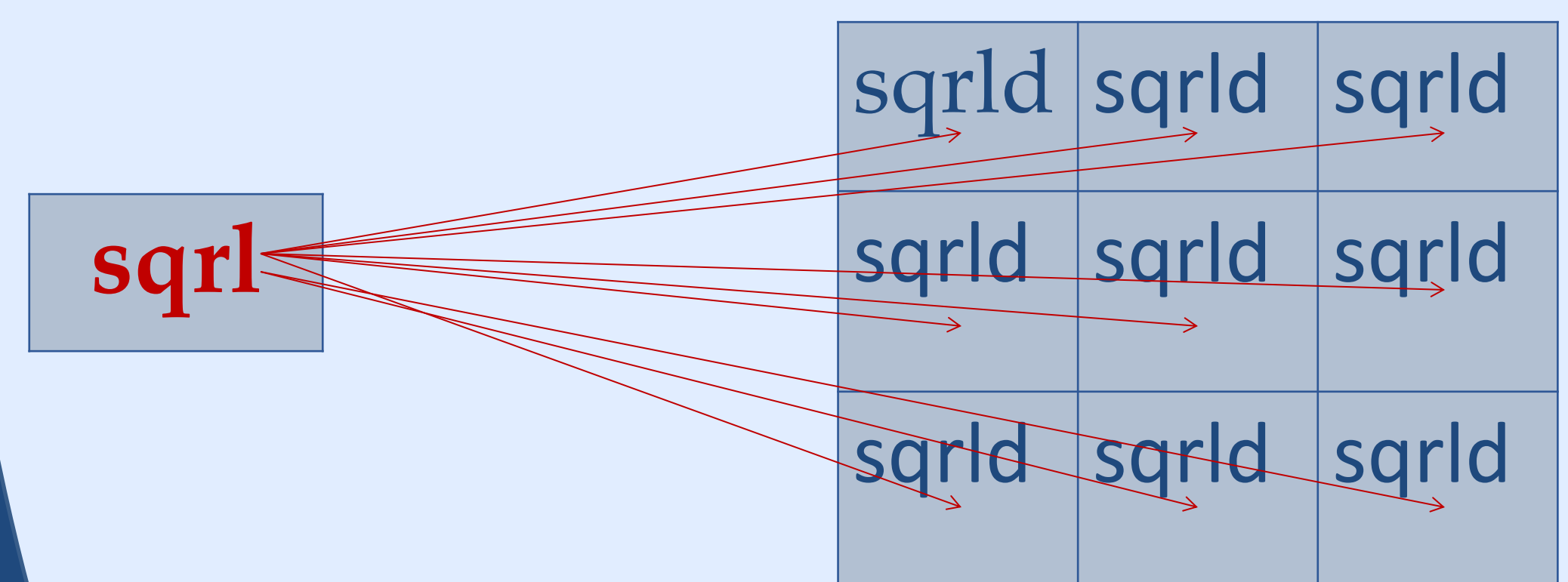


Figure 1: sqr1 can be easily run on all nodes of a cluster with one command. With much larger clusters having thousands of nodes, this program alleviates the need to address each node individually.

Methods

We leveraged previously existing Go packages in order to create sqr1. Our completed application allows the user to customize what is logged and exported.

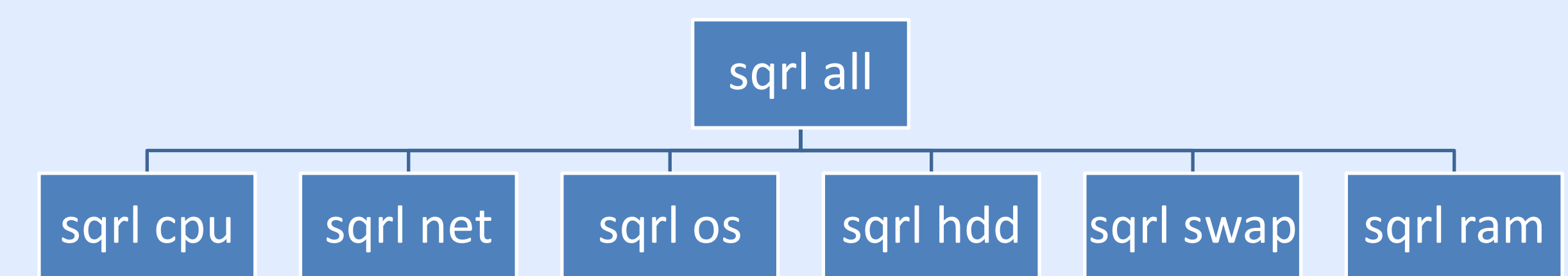


Figure 2: The commands and subcommands of sqr1 that allow the user to customize logs and output

We looked into different ways sqr1 could be automated and scheduled to run:

- Cron (through the operating system)
- Internal task scheduling
- HTTP server

After exploring pros and cons of each process, we decided on a combination of both internal task scheduling and an HTTP server.

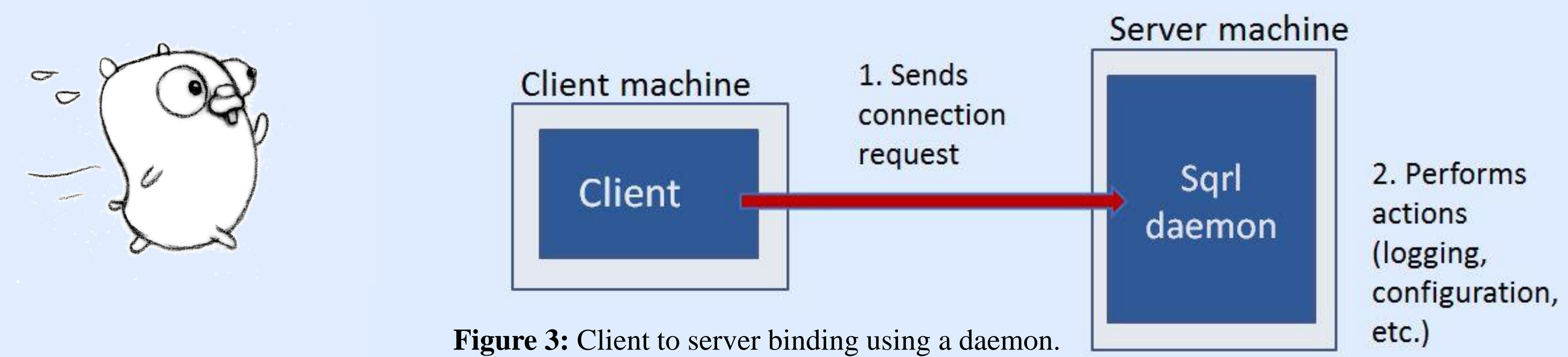
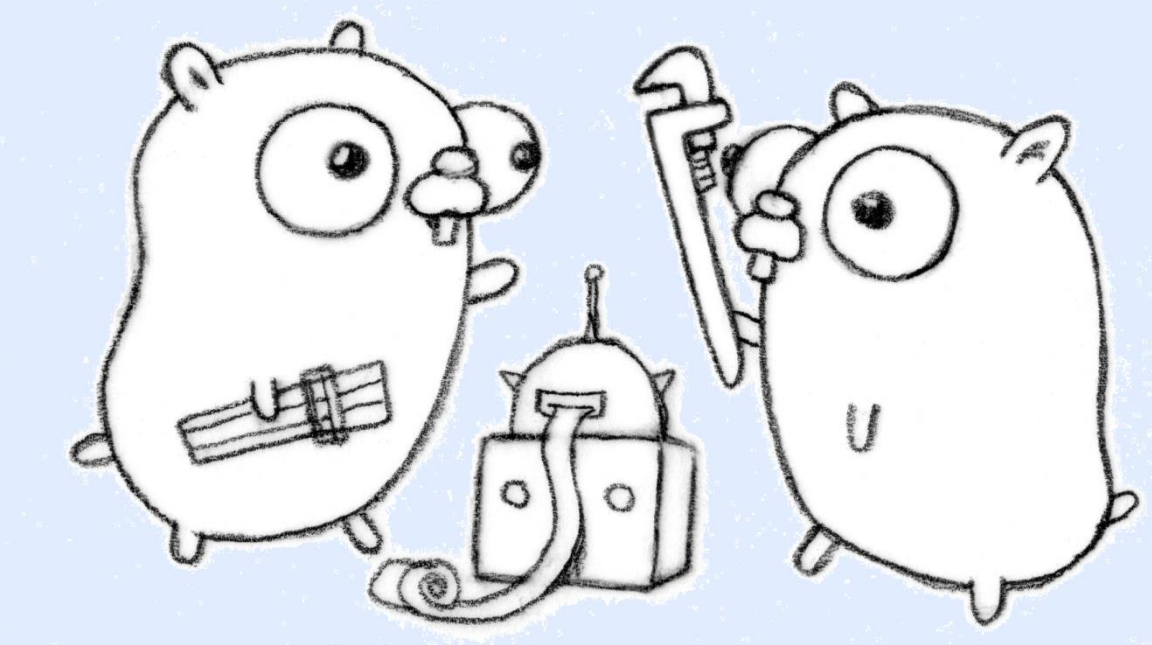
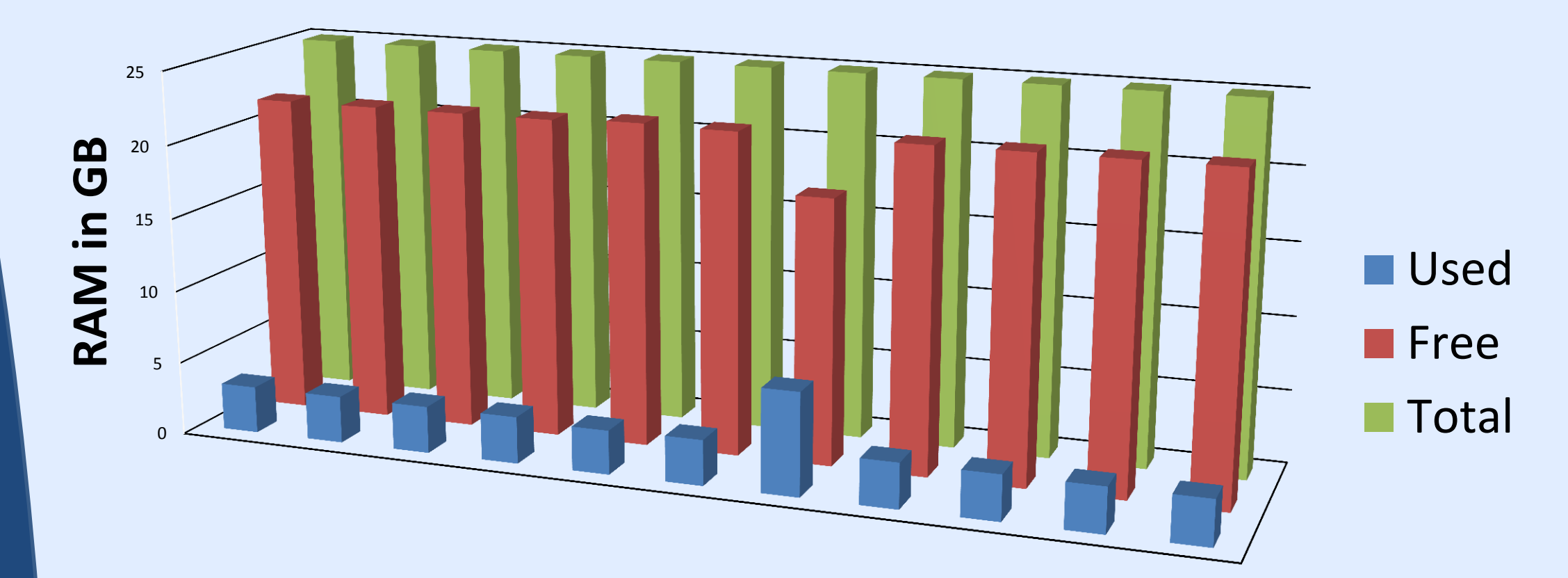


Figure 3: Client to server binding using a daemon.

Why We Chose It

Internal task scheduling and an HTTP server provide many benefits for continuously and efficiently logging system information. While our internal scheduler will update and log information on a set time (versus the operating system's own cron scheduler), using an HTTP server provides the ability to run sqr1 at any time. Sqr1, when running in daemon mode, allows the application to accept requests that could update the frequency of logging and verbosity of information. It also provides scalability; whether there are five machines or 5,000, each individual machine logs its own information, allowing an application like Splunk to collect and visualize that information. In case of an emergency, the HTTP server in sqr1 allows the user to make configuration changes to all of the nodes in a cluster.

A Selection of Compute Nodes from the Cab Cluster



Individual Nodes

Figure 4: Graph of memory usage on a selection of nodes in the Cab cluster.

Discussion

The data that sqr1 provides can be used in many different applications. For example, sqr1 could provide the data needed to recognize correlations between running jobs and memory usage, allowing users to better utilize system resources. Depending on the memory requirement of a particular job, sqr1's exported data could be evaluated using analytic tools to estimate the most efficient node-to-memory distribution for certain jobs. Because sqr1 is automatically run, it is easily accessible and provided in real-time to users.

While the combination of an HTTP server and daemon is optimal, there are still some downfalls to this method. The downside to the HTTP server is its lack of security; as of now, anyone can access the server if needed. Consistent reliability of the network also raises another concern because users would not be able to access all of the machines through a management node if a network connection is broken.

To extend this project, a point to address would be security. For example, adding authentication would ensure possession of how sqr1 runs is limited to a select number of privileged users. Another area for extension would be logging dynamic information each time the program is run, while the static information, such as operating system info, is only logged per specific request, removes unnecessary logs of information.

