

GridAdmin

Dan Alexandru

Faculty of Computer Science, Iasi, Romania

`dan.alexandru@info.uaic.ro`,

web home page: <https://github.com/xR86/rc-project>

Abstract. GridAdmin offers a solution for orchestration of VMs - both on managed servers and cloud servers (like AWS), with a client-server architecture, the server acting as a command and control server, and the client offering a web interface...

Keywords: networking, Linux, C, Python, AWS

1 Introduction

GridAdmin offers a solution for network administration by sending commands through a Python GUI to a C intermediary server that passes along data to the Python server, that uses ssh to pass the commands to the slave machines (or respective API available for cloud platforms).

Targeted functionality similar to Python's Fabric (<http://www.fabfile.org/>).

2 Technologies used

2.1 Server-side

The project uses C for the concurrent TCP server. Use of libssh (<https://www.libssh.org/>) can be employed on the server (master), along with openssh-server on the slave, but since libssh is unstable/not functional with certain OS/openssh versions, it will require the use of the Paramiko ssh library.

If needed, the C server could be extended with python modules.

2.2 Client-side

The project uses Python and PyQt4, the interface being modeled with web technologies such as HTML, CSS, Javascript, Angular.js. This will grant easier interface creation, and a more rich interface, (example of frameworks doing this: Apache Cordova - <https://cordova.apache.org/> / Ionic, Electron). One example of web app that runs on desktop is Slack (<https://slack.com/downloads/windows>).

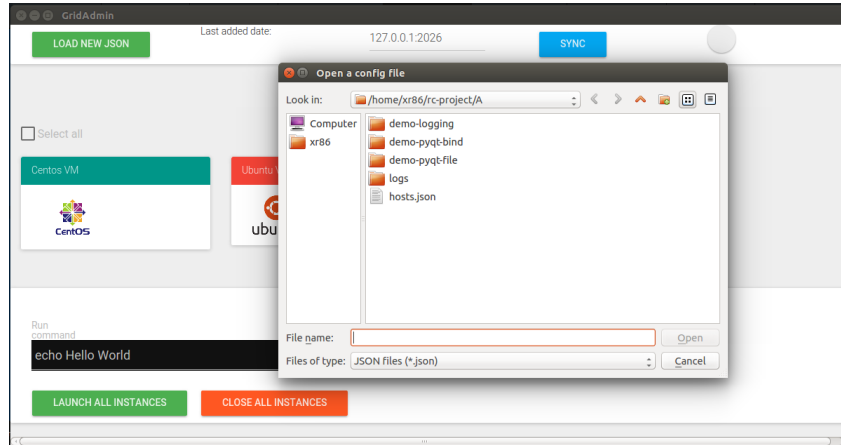


Fig. 1. Client demo

3 Application Architecture

The application makes use of the client-server model for interfacing and master-slave model for control.

3.1 Client-server model

In this project, the **client** is the Python GUI that can be remote or on the same machine as the server. The client connects to the **C server** to run bulk commands (to multiple machines) or to run one or more commands on one machine. The server will ssh to various machines, execute commands from client and return the output to the client.

3.2 Master-slave model

In this project, the **master-slave model** acknowledges that there is a server that has ssh access to various machines (public-key authentication and/or password authentication), and can run almost anything on them (if root mode allowed in `sshd_config`).

3.3 Application layer diagram

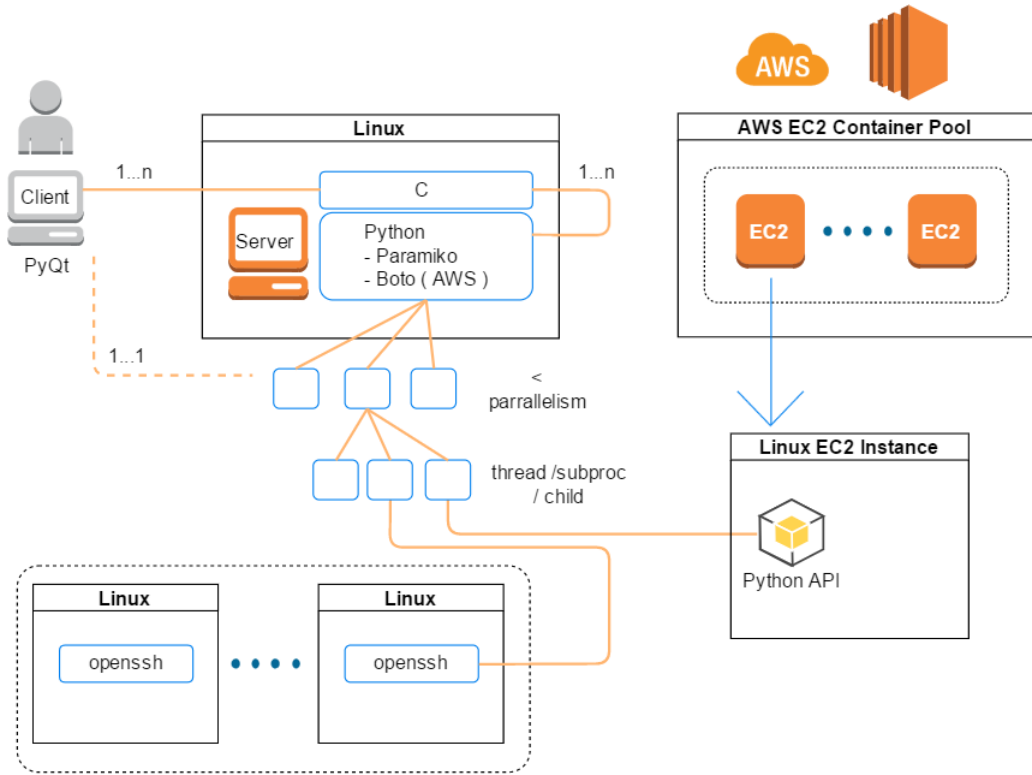


Fig. 2. Redesigned Application Layer architecture - The client exchanges data with the server, that is C and Python separately (C data proxy and Python Paramiko server) , the server then connects either to various machines (with openssh pre-installed) or to a AWS EC2 instance (through the API).

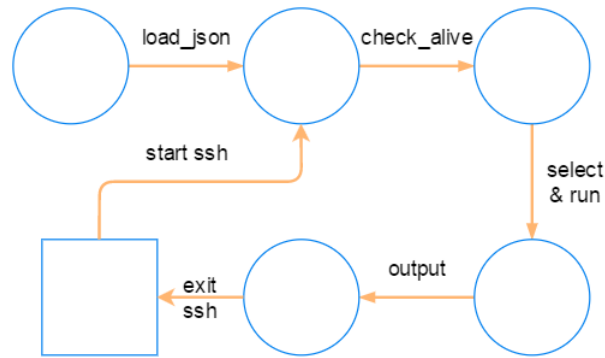


Fig. 3. Automata for GridAdmin.

4 Implementation details

4.1 Application flow

4.2 Relevant code

5 Conclusions

The solution could be extended to be:

- **cloud provider agnostic** (especially IaaS), offering support for Google Cloud Platform and/or Microsoft Azure)
- duplicating the **server and hosting it in different geographical locations** and infrastructure backbones - for availability / legal reasons
- offering **specific commands in the interface** such as on-the-fly snapshots, backup rotation of logs (backup rotation scheme), bulk close/sleep/open, grouping after type (VM/instance/etc)

Solution will be made available at the following address:

<https://github.com/xR86/rc-project>