

SimGrid Platform File Visualizer and Converter

Haochen Wang

Advisor: Henri Casanova, University of Hawai'i at Manoa

Spring 2024



INFORMATION & COMPUTER SCIENCES

UNIVERSITY of HAWAI'I at MĀNOA

Context

Computer science researchers face challenges when assessing distributed system architectures and applications (HPC, cloud, edge, peer-to-peer, etc.)

SimGrid is a framework for developing simulators of distributed systems for rapid and reproducible assessments of arbitrary hardware configurations

A requirement for any SimGrid-based simulator is user specification of the hardware platform to be simulated (network zones, hosts, routes, disks,, links,, etc.)

Motivation

Traditionally, SimGrid allows users to describe simulated hardware platform configurations using XML files

However, XML files have limited expressive power and can pose challenges for users to analyze

The SimGrid S4U API has addressed XML limitations by enabling users to programmatically describe simulated platforms in C++

SimGrid has a large user community and many XML platforms that has been developed needs to be:

- inspected for correctness via visualizations
- replaced by C++ code

Objective

Develop a tool for:

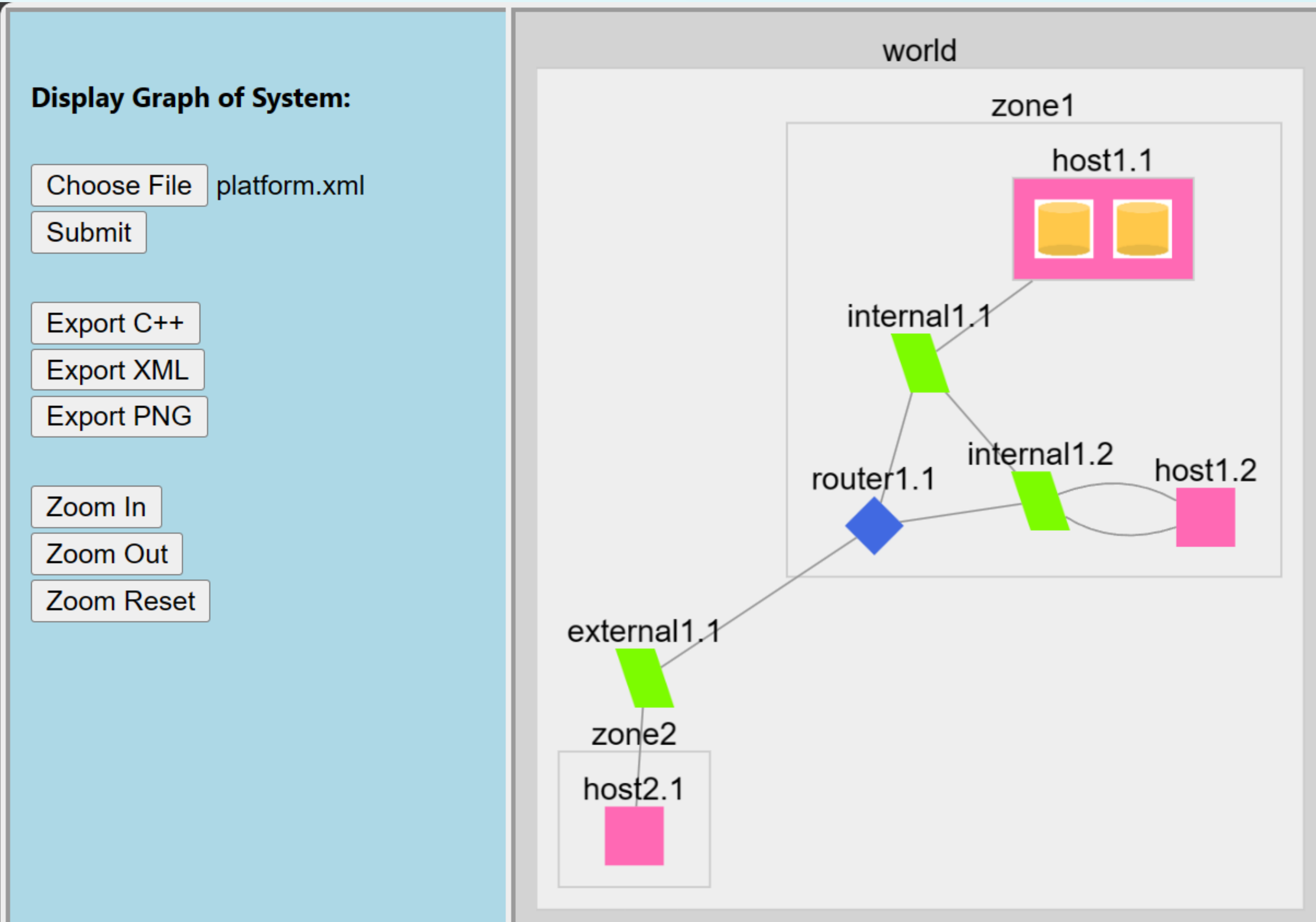
- visualizing any SimGrid XML platform file
- editing the SimGrid platform files
- automatic generation and export of the equivalent C++ code based on the S4U API

Methodology

- Develop a front-end that allows users to upload a SimGrid platform description XML file
- Parse the XML file to recursively generate a visualization of the hardware platform as a hierarchy of network zones displayed using force-directed layouts
- Implement modals for users to inspect and edit the specifications of the simulated hardware resources
- Implement an XML-to-C++ converter that generates a standalone C++ package that is ready for use in any SimGrid simulator

Visualizer and Converter

The image depicted below displays the visualization of a SimGrid hardware platform. The uploaded platform configuration file is parsed and then graphed using the Cytoscape library.



The image depicted below displays the conversion of the platform configuration file from XML to C++. The C++ package is downloaded from the Visualizer.

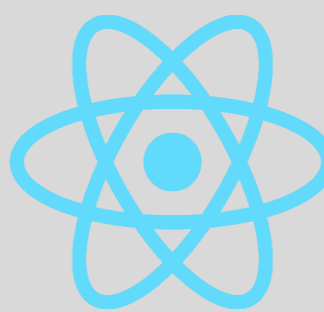
```
<platform version="4.1">
  <zone id="world" routing="Full">
    <zone id="zone1" routing="None">
      <host id="host1.1" speed="1Gf">
        <disk id="hard_drive1" read_bw="100MBps" write_bw="100MBps">
          <prop id="size" value="5000GiB"/>
        </disk>
        <disk id="hard_drive2" read_bw="200MBps" write_bw="150MBps">
          <prop id="size" value="1000GiB"/>
        </disk>
      </host>
    </zone>
  </zone>
</platform>
```

```
#include <simgrid/s4u.hpp>
namespace sg4 = simgrid::s4u;
extern "C" void load_platform(const sg4::Engine &e);
void load_platform(const sg4::Engine &e) {
  auto world = sg4::create_full_zone("world");
  auto zone1 = sg4::create_full_zone("zone1");
  zone1->set_parent(world);
  auto host1.1 = zone1->create_host("host1.1", "1Gf");
  host1.1->set_core_count(10)->seal();
  auto hard_drive1 = host1.1->create_disk("mydisk", "100Mbps", "100Mbps");
  hard_drive1->set_property("size", "5000GiB");
}
```

Tech Stack



Javascript:
Back-end development



React:
Front-end development



Cytoscape:
Library for graph visualization



C++:
S4U interface



SimGrid:
Framework for developing
simulators of distributed systems

Future Work

Implement algorithms for analyzing the vertices and edges of the simulated platform's network graph for inspection of network routes

Add advanced visual editing capabilities by which platform elements can be added or removed

Deployment in production on the SimGrid website

Learning

Deepened knowledge in developing a full-stack web application

Expanded understanding of distributed systems

Acquired additional experience working with visualization and graph visualization APIs

Gained more proficiency in recursive algorithm design and implementation