RES-MATCH Fall 2023

Final Report

Distributed Computation for Light Propagation Modeling

| Student Researcher: Prajwal S Venkateshmurthy |
| Faculty Mentor: Nik Sultana |
Nov 30, 2023

I. Introduction

This research project aimed to develop extensions to the Co-Design of Exascale Storage Architectures (CODES)/Rensselaer's Optimistic Simulation System (ROSS) simulation system to enable distributed computation for accurate and efficient modeling of light propagation. The goals were to enhance programmability of network switches specifically used in the FatTree network topology using C/C++ and Python and to leverage parallelism, and prototype light propagation simulations.

II. Methods

The research methodology involved three primary phases:

Phase 1 - Phase 1 involved augmenting the FatTree router model in CODES/ROSS to enable custom function executability and state accessibility during packet routing. The core methodology was implementing virtual functions that allow each switch to register and execute C/C++ functions upon receiving packets. As a proof of concept, functions were developed to count sent and received packets at the switch level. Packets were then injected into the network based on predefined parameters and routing tables. Testing focused on comparative analysis of per-switch counters to verify expected equivalency between aggregate packets sent and received across the system. This approach of executable virtual functions and state accessibility laid the groundwork for advanced modeling and control.

Phase 2 - involved integrating a Python interpreter within CODES to enable flexible Python-based modeling and control. The core methodology leveraged Python's C API and the Python.h library to load and execute custom Python functions from within the FatTree router model. During simulation, whenever a switch's registered function is triggered upon receiving a packet, router state and packet data gets converted to JSON. This information then passes into the Python environment as dictionary objects. The

Python code can analyze and manipulate the data, applying advanced logic like firewall policies. Updated state returns to the router model, synchronizing any changes. As an example, functions were developed to implement rule-based packet filtering. Packets were injected based on predefined parameters to test determinism across repeated simulations. The Python firewall either allowed or dropped packets using division rules on packet IDs.

Phase 3 - Prototyping a light propagation simulation using the enhanced programming capabilities. In this phase we implemented a Python code to perform a Monte Carlo simulation to model the propagation of photons through a participating medium. The key goal is to simulate and track the trajectories of photons as they undergo absorption and scattering events. By modeling individual photon "packets" and their interactions probabilistically, the code can analyze and aggregate spatial distribution and patterns of photon path.

III. Results

Phase 1 testing validated the enhanced programmability and accessibility of the FatTree router model in CODES/ROSS. Execution of registered C/C++ functions during packet routing was successful, as evidenced by the accurate per-switch packet counting. Testing showed strong determinism - packets consistently reached their intended destinations across repeated simulations based on routing tables. Comparative analysis verified precise equivalence between aggregate packets sent and received system-wide.

In Phase 2, the integrated Python interpreter enabled robust functionality and coordination between scripts and the router model. Manipulation of router state and packet data by firewall policy logic in Python properly synchronized back to CODES/ROSS. The firewall use case exhibited deterministic allowance/blocking of packets in line with division rules applied to packet IDs in the Python code.

Finally in Phase 3, simulation coordinates were successfully generated reflecting the probabilistic trajectory of photon packets. Statistical analysis of photon spatial distributions showed spreading patterns - showcasing proper modeling of light propagation phenomena. Computation times also improved by leveraging parallelised execution across distributed simulation nodes.

IV. Discussion

This project successfully enhanced the programming capabilities of the CODES/ROSS platform to enable advanced modeling of complex physical systems. Phase 1 laid groundwork by making the FatTree router model highly customizable and accessible. Phase 2 then integrated a Python interpreter for flexible control. Finally, the light propagation prototype in Phase 3 validated the framework for distributed photonics computations. Outcomes aligned well with objectives around precision, performance, and usability. Packet routing determinism, Python/C++ coordination, and statistical improvements were achieved. With these simulation capabilities, researchers can now pursue innovations in light matter interactions, medical physics, atmospheric optics, and astronomy. This research project advances an important code asset. By opening up new possibilities for distributed modeling, it could have a significant research impact across large-scale storage systems, I/O workloads, HPC network fabrics, distributed science systems, and data-intensive computation environments.

V. Conclusion

In conclusion, the project successfully met all defined milestones across the three phases within the program timeframe. Major progress was made towards enabling accurate and efficient distributed computation for light propagation simulations. Specific achievements include augmented programmability, accessibility, and determinism in the CODES/ROSS router model to support advanced simulations. Additionally, the integration of a Python interpreter now enables flexible scripting and control. Finally, the fully completed Phase 3 led to an accurate prototype for simulating light propagation in parallel. The model was thoroughly tested and validated. Together, these outcomes provide a feature-rich platform primed for tackling a wide range of research questions through high performance distributed computing. The project has opened up new possibilities for the research community.

VI. Overall Impression

Participating in RES-MATCH and working with leading researchers from Argonne National Laboratory and professor Dr. Nik Sultana was an invaluable learning experience. The research project provided hands-on software engineering practice including background on distributed science systems, data-intensive computation environments, large-scale storage systems, I/O workloads, & photonics modeling. As the program grows, offering scholarships specifically for computer science students could enable even greater cross-disciplinary synergies. With more computer science participation, researchers could leverage methodologies like machine learning to push boundaries. The flexible structure and coordinator support fostered an excellent research experience for me.