Real-time Exascale Data Analytics for Scientific Workflows: Resource and Information Abstraction

Student: Patrick Buehler
Advisor: Richard Yang

1 Summary

The Large Hadron Collider generates massive amounts of data gathered from a variety of sensors when particles collide. This project focuses on data gathered from one of these detectors in particular, the Compact Muon Solenoid, which uses a large magnet to bend the paths of particles from collisions in the LHC.

Hundreds of organizations and thousands of scientists are a part of the CMS experiments, working on the CERN data at research institutions around the world. It is a major technological challenge to host, coordinate, and share the experiment data with all the member researchers.

Professor Yang is working with Caltech and CERN to design a distributed computing network to perform real-time analytics on exascale data generated from CMS. My capstone project will focus on the software-defined network controllers used in this network, which are per-domain controllers serving as an intermediate layer between the end user (researchers, labs and universities) and a centralized global resource orchestrator. In particular, the SDN controllers will support the following two functions: [1] to allow the resource orchestrator to query network constraints, link speeds, etc. from the end users, and [2] to provide optimal local routing information across end-user specific platforms and protocols. My objective is to design and build an interface between the global resource orchestrator and the domain-specific SDN controllers for the next-hop queries.

After designing and implementing these SDN controllers, the culmination of the project will be setting up and configuring the SDN domain controllers for a demo presentation of our distributed computing network at the ACM/IEEE SC17 conference in Denver, Co. this November.

2 Project Overview

The larger CMS project is broken up into four parts: a global resource orchestrator, SDN domain controllers, end user set-up, and general infrastructure.
2.1 Global Resource Orchestrator

The global resource orchestrator will receive a stream of jobs from end users, and query the SDN domain controllers for routing information and network constraints in order to [1] efficiently allocate the tasks to different virtual machines and [2] optimize where different virtual machines will get the data from.

2.2 SDN Domain Controllers

2.2.1 Next-Hop Routing

This is the piece I focus on. It is treated in greater detail in §4 SDN Domain Controllers.

2.2.2 Network Constraints: Set of Inequalities [Thomas]

The global resource orchestrator will need to be aware of current network constraints in order to efficiently delegate tasks and jobs. To make this happen, the GRO will be able to query the SDN domain controllers for linear inequalities describing the current network constraints for a set of flows. Thomas will focus on this aspect of the project.

2.3 End-User Systems

The end-user systems will be responsible for enforcing the rates of computation and data transmission expected from the global resource orchestrator. Each individual domain, or site, may have varying security interests and different requirements. In this way, the system is more complicated than a distributed computing network run by Facebook or Google, which can implement top-down policies in their closed system that eliminate some of these concerns. To implement this layer, many changes will be made to open virtual switches in each of the domains.

3 SDN Domain Controllers

Software defined networking promises programmability and central control over the whole network infrastructure. In the context of this project, SDN will allow the global resource orchestrator to manage the end user’s routing, resource dedication, etc.

3.1 Objectives

In order to support this centralized control over the network, the SDN domain controllers will manage specific domains as intermediary levels between the global resource orchestrator and end users. They will use a bottom-up method of gathering
data, abstract it, and provide the global resource orchestrator with next-hop routing information and an abstraction of the various network constraints in its domain in the form of a set of linear inequalities. These, then, are the major deliverables of the project. My contributions to the project will center on the next-hop routing information.

3.1.1 Routing Information & Next Hops [Patrick]

The first task is to design a RESTful API to allow the global resource orchestrator to query the location of the next hop for a set of flows in an abstract, modular way. RESTfulness allows communication with multiple SDN platforms in a scalable way, essential for exascale data analytics.

Once the API has been specified, the next step will be implementation. Given the scope of a distributed system of exascale analytics, as well as the permissions concerns from linking together disparate labs and universities, using Border Gateway Protocol (BGP) will be necessary for its emphasis on both scalability and security. In order to use BGP in each domain, we may have to configure BGP ourselves and set it up on each SDN controller to work alongside the controller’s routing information base (RIB) in planning the next hop. This step will require querying the SDN database to understand the domain topology; we will focus on edge devices, the egress points of the domain, and plan the next hop from there.

Different SDN controllers may use different platforms, so API implementation will be done in a layered manner and ensure compatibility with at least three industry SDN platforms: OpenDaylight, Kytos, and ONOS.

3.2 Technologies to Consider

The technologies I will use in this project may include:

3.2.1 RESTful API

The interfaces between the various layers of the larger distributed computing infrastructure will most likely be through RESTful API’s. API’s provide a level of abstraction for the different layers that allows the designers to modularize parts of the system for easier extensibility and portability. This will allow us to support many different devices and protocols. RESTful API’s in particular use stateless HTTP requests to provide four core functionalities: GET, PUSH, POST, and DELETE.
3.2.2 Border Gateway Protocol

The border gateway protocol, or BGP, is probably the most widely used routing protocol. It divides a network up into different autonomous systems. It can be used for routing within the AS’s, i.e. internal BGP or iBGP, as well as between AS’s, i.e. external BGP or eBGP. Throughout the project, it will be important to understand BGP’s routing decision making process, modifiable attributes, etc.

3.2.3 YANG Modeling

YANG Modeling is a data modeling language useful for modeling both the configuration data and the state data of network elements. As YANG is protocol independent, one of its key advantages is that it can easily be converted into either XML or JSON, depending on the needs of the user.

3.2.4 Protocol Buffers (Google)

Google’s language-neutral, platform-neutral, extensible mechanism for serializing structured data. Developers can define the format of data, and Google’s generated source code makes it easy to read and write the data into a variety of streams and formats.

3.2.5 Thrift (Facebook)

Thrift is a protocol definition software framework that enables language-agnostic communication and Remote Procedure Calls between a server and a client. Compiler generated code makes it easy to serialize/deserialize messages and manage sockets.

4 Timeline & Deliverables

<table>
<thead>
<tr>
<th>Deliverable</th>
<th>Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal</td>
<td>September 25</td>
</tr>
<tr>
<td>Routing Information API Design (for review)</td>
<td>October 6</td>
</tr>
<tr>
<td>Version 1 Prototype</td>
<td>October 27</td>
</tr>
<tr>
<td>API Backend Implementation and Basic Visualization</td>
<td>November 9</td>
</tr>
<tr>
<td>SC17 Conference: Presentation and Demo</td>
<td>November 16</td>
</tr>
<tr>
<td>Final write-up and code, polished visualization</td>
<td>December 20</td>
</tr>
</tbody>
</table>
5 Signatures and Approval

Patrick Buehler: ________________ 9-25
Richard Yang: ________________ 9-25
James Aspnes: ________________