JSON Dataset Reader

- TensorFlow I/O

Project Proposal for Google Summer of Code 2019

Mentor: Yong Tang (GitHub Linkedin) and Yuan Tang (GitHub Linkedin Twitter)

1. Student Information

Background

Name: Jiacheng Xu
University: Swiss Federal Institute of Technology in Lausanne (EPFL)
Major: Master student in Computational Science and Engineering

Open source experience:
Google Summer of Code Student in 2018, CoreDNS, CNCF

- Designed and built an etcd-based CoreDNS plugin for identifying nodes in a cluster without domain name collisions which can be used for deploying distributed TensorFlow.
- Related work: idetcd
- CNCF blog: https://www.cncf.io/blog/2018/10/30/gsoc-jiacheng-xu/

Contributor of CoreDNS

- #16(out of 150) on the CoreDNS contributor page.

Contact Information

Email: xjcmaxwellcjx@gmail.com
Phone: +41 786642289
Github: https://github.com/jiachengxu
Linkedin: https://www.linkedin.com/in/jiacheng-xu-389530128/

About Me

My name is Jiacheng Xu, and I am a third-year master student at the Swiss Federal Institute of Technology in Lausanne (EPFL), majoring in Computational Science and Engineering. I am passionate about the distributed system and distributed machine learning. I am a Gopher, but I also have rich experience in other programming languages such as Python, C++, and Scala.

I am definitely an open source lover! As I mentioned before, I was a GSoC student in 2018, I was working on CoreDNS during the GSoC, and my task was about designing and

implementing a CoreDNS plugin for deploying a cluster without domain name collisions. It also provides a possible approach to deploying distributed TensorFlow and makes it much easier to replace broken nodes in distributed TensorFlow without terminating the whole system.

I am also a speedcuber, I really like solving puzzles. I can solve 3x3 Rubik's Cube around 15 seconds and 7x7 for around 4 minutes. I also go to competitions for that.

Availability

I will be based in Baden, Switzerland during the summer. Therefore, I will be working in GMT +2 time zone. And I will be available **35 - 40 hours per week**. Since our summer vacation is from June to September, so I believe that I have enough time to complete the project.

2. Synopsis

TensorFlow is one of the most popular machine learning frameworks and is widely used in fields beyond machine learning and data science. The architecture of TensorFlow has been elegantly designed such that it is possible to be extended in big data, medical imaging, and physical sciences. Supporting different format of data is the necessary step for communities beyond machine learning to adopt TensorFlow, as data is always as the entry point or edge node of the TensorFlow's graph. Importing data with different formats natively in TensorFlow allows users to build their systems or applications without the need of additional conversion infrastructure.

TensorFlow I/O is a SIG which focuses on providing various data format supports for TensorFlow, and many data formats are already supported, like Apache Kafka stream-processing, Amazon Kinesis data streams and also LMDB format and MNIST format, etc. However, the generic JSON format hasn't been supported yet. It is quite necessary to support JSON format since JSON files are widely used in machine learning and data science.

In this project, I will be working on providing JSON support in the TensorFlow I/O so that it will be possible to read JSON files into Tensorflow.

3. Project Goals

Objectives

- Implement a C++ library JSON parser since the TensorFlow I/O is written in C++.
- Design and Write a Python API wrapper for the JSON parser.
- Add unit and integration tests for the JSON dataset reader.

- Add documentation for JSON dataset reader.

Tasks

- Implement a C++ library JSON parser.
    - Determine a good JSON library to implement the C++ JSON parser.
        - jsoncpp can be an option, but there are some other options.
        - JSON for Modern C++
        - Picojson
    - Implement the data operations for the C++ kernel library.
        - Read Operation: Read JSON as the input.
        - Convert Operation: Extract a set of specific fields from the JSON streams into TensorFlow dataset.
    - Register the operations into TensorFlow.
- Design and Write a Python API wrapper for the JSON parser.
    - Write Python API wrapper by invoking the data operations in the C++ library.
- Add unit and integration tests for the JSON data reader.
    - Write tests for both the Python APIs and the C++ library.
- Add documentation for JSON dataset reader.
    - Provide detailed Python examples about how to use JSON dataset reader.

4. Timeline

Timespan	**Activity**
6 May - 27 May	*Community Bonding Period:* ● Getting familiar with the structure and source code of TensorFlow I/O. ● Doing research and benchmarks and communicating with the mentor for choosing the JSON c++ library.
Actual Start of the Work Period	
28 May - 24 June	● Implement the C++ library for JSON dataset reader. ● Start writing unit tests. ● Start writing documentation.
24 June - 28 June	First evaluations: ● C++ library of the JSON dataset reader should be implemented.
28 June - 22 July	● Write Python APIs for the JSON Dataset. ● Keep writing tests and documentation.
22 July - 26 July	Second evaluations: ● Python APIs of the JSON dataset reader should work.

26 July - 19 August	● Add more tests and documentation. ● Add examples for how to use JSON dataset reader.
19 August - 26 August	● Submit Code and Evaluations.

5.Deliverables

- A C++ JSON dataset parser for TensorFlow I/O.
- A Python API wrapper for JSON dataset reader.
- Useful tests for JSON dataset reader.
- Detailed examples about how to use JSON dataset reader.