# POSTER: Keeping deep learning GPUs well fed using object storage

Or Ozeri
IBM Research
Haifa, Israel
oro@il.ibm.com

Effi Ofer
IBM Research
Haifa, Israel
effio@il.ibm.com

Ronen Kat
IBM Research
Haifa, Israel
ronenkat@il.ibm.com

## ABSTRACT

In recent years, machine learning and deep learning techniques such as deep neural networks and recurrent neural networks have found uses in diverse fields including computer vision, speech recognition, natural language processing, social network analysis, bioinformatics and medicine, where they have produced results comparable to and in some cases surpassing human experts. Machine learning requires large amount of data for training its models with much of this data residing in object storage, an inexpensive and scalable data store. Also, deep learning make use of state of the art processing capabilities from high-end GPUs and accelerators, such as Google Tensor Processing Units (TPUs), which enable parallel and efficient execution. The throughput that such GPUs can support is very high.

This however constitutes an impedance mismatch as the object storage is not designed for high performance data transfers and standard practices for feeding deep learning models from the object storage can result in poor training performance. Furthermore, the typical deep learning framework uses a file access interface, and object storage support a REST based interface with different APIs and semantics than a file system [2]. To fully take advantage of these GPUs and operate at full utilization, frameworks, such as TensorFlow, Cafe, and Torch, needs to deliver data as fast as possible to keep the GPUs busy. This becomes a significant challenge when the training data does not reside in the same machine as the GPUs, as is the case when using object storage, resulting in a utilization challenge for the expensive processing units.

To solve the impedance mismatch and keep the processing units fully utilized, we have added a FUSE based file system, S3fs [1], to our deep learning stack. S3fs translates POSIX file API requests into REST API against the object storage. It is an open source project which, as part of this work, we optimized so that read requests are performed using new innovative logic that translates the requests into multiple concurrent range reads requests against the object storage. This enables us to obtain higher throughput from the object storage than is possible using the naive approach. Reads are cached in memory and are served back to the deep learning framework asynchronously. Since deep learning frameworks often run their training in multiple epochs the in memory cache speed is highly beneficial.

Our FUSE based architecture has been implemented in the Deep Learning as a Service offering on the IBM Cloud, and our S3fs enhancements have been contributed to the S3fs project repository. Using our architecture we are able to speed up deep learning performance many folds and keep expensive GPUs fully utilized.

## REFERENCES

[1] 2018. S3fs git repository. (2018). https://github.com/s3fs-fuse/s3fs-fuse
[2] Gil Vernik, Michael Factor, Elliot Kolodner, Pietro Michiardi, Effi Ofer, and Francesco Pace. 2018. Stocator: Providing High Performance and Fault Tolerance for Apache Spark over Object Storage. *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)* (2018).


*SYSTOR, 2018, Haifa, Israel*

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4