CERN

Distributed Machine Learning with Apache Spark and Keras

Joeri Hermans (Technical Student)
Maastricht University

CERN IT Hadoop Service

- **Hadoop production service**
  - Multiple Hadoop / Spark clusters to satisfy demanding requirements from the experiments and accelerator communities.

- **Challenges**:
  - Running critical applications on Hadoop and Spark.

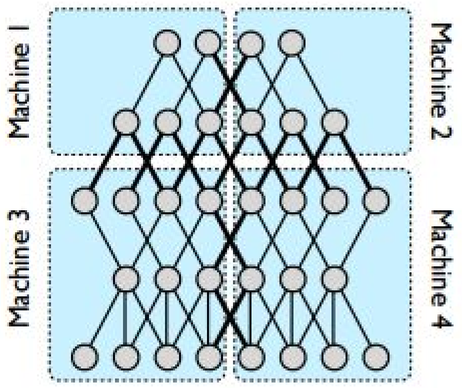
- **Notable projects**:
  - ACCLOG
  - IT Monitoring
  - ATLAS EventIndex

Problem Statement

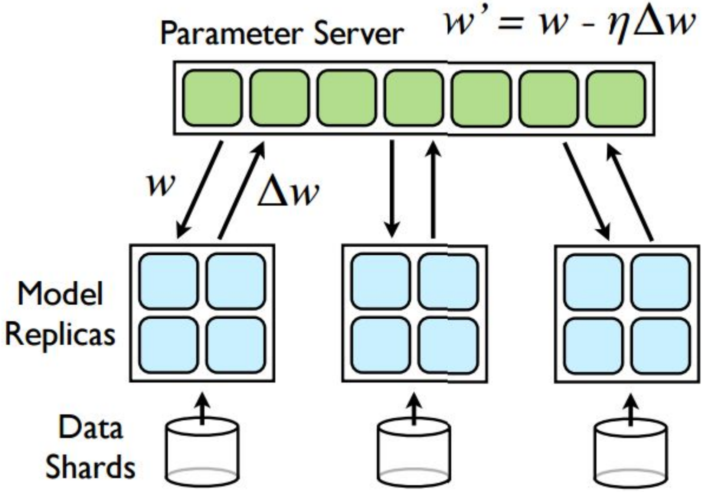
- **Specific question from CMS:**
    - Equivalent amount of CPU cores compared to a single GPU?
- CMS use-case requires a lot of data (**Spark** would be a good candidate).
- **Keras** (modeling framework in Python, interfacing TensorFlow or Theano).
- Solutions exist which fit the above, but:
    - **Not working on cluster** (26/09/2016).
    - Very naive gradient optimization.
    - Not compatible with Spark DataFrames.
    - **Not easily extendable** (e.g., new distributed optimizers).

Main approaches

Model parallelism



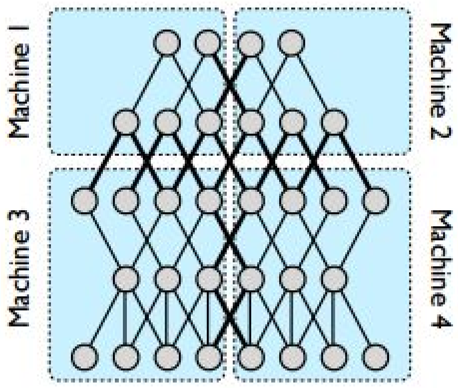
Data parallelism



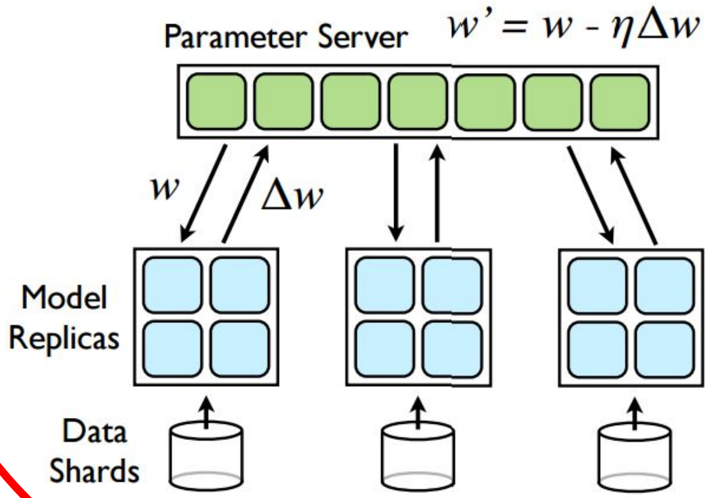
Source: Large scale distributed Deep Networks [1]

Main approaches

Model parallelism

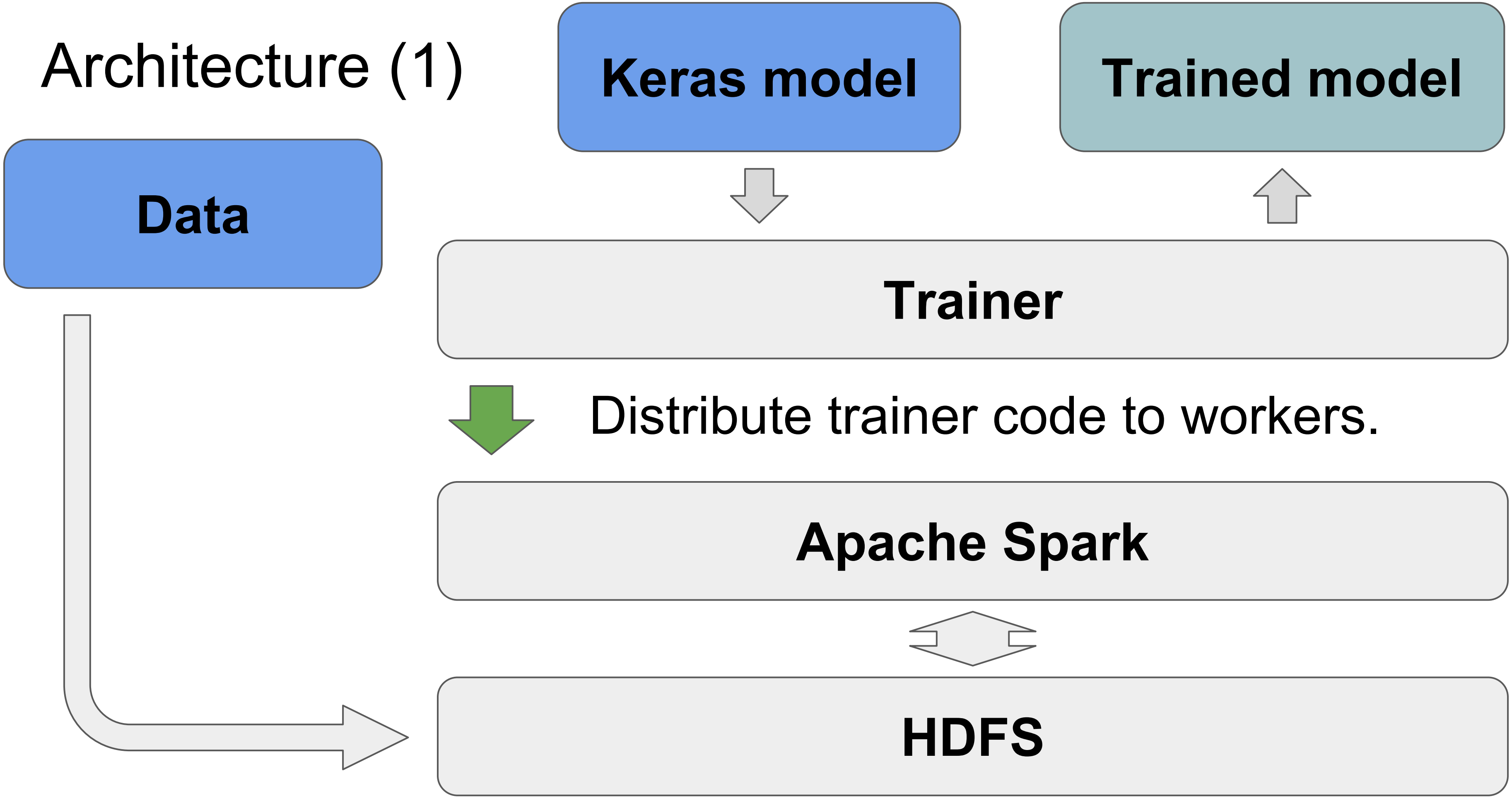


Data parallelism

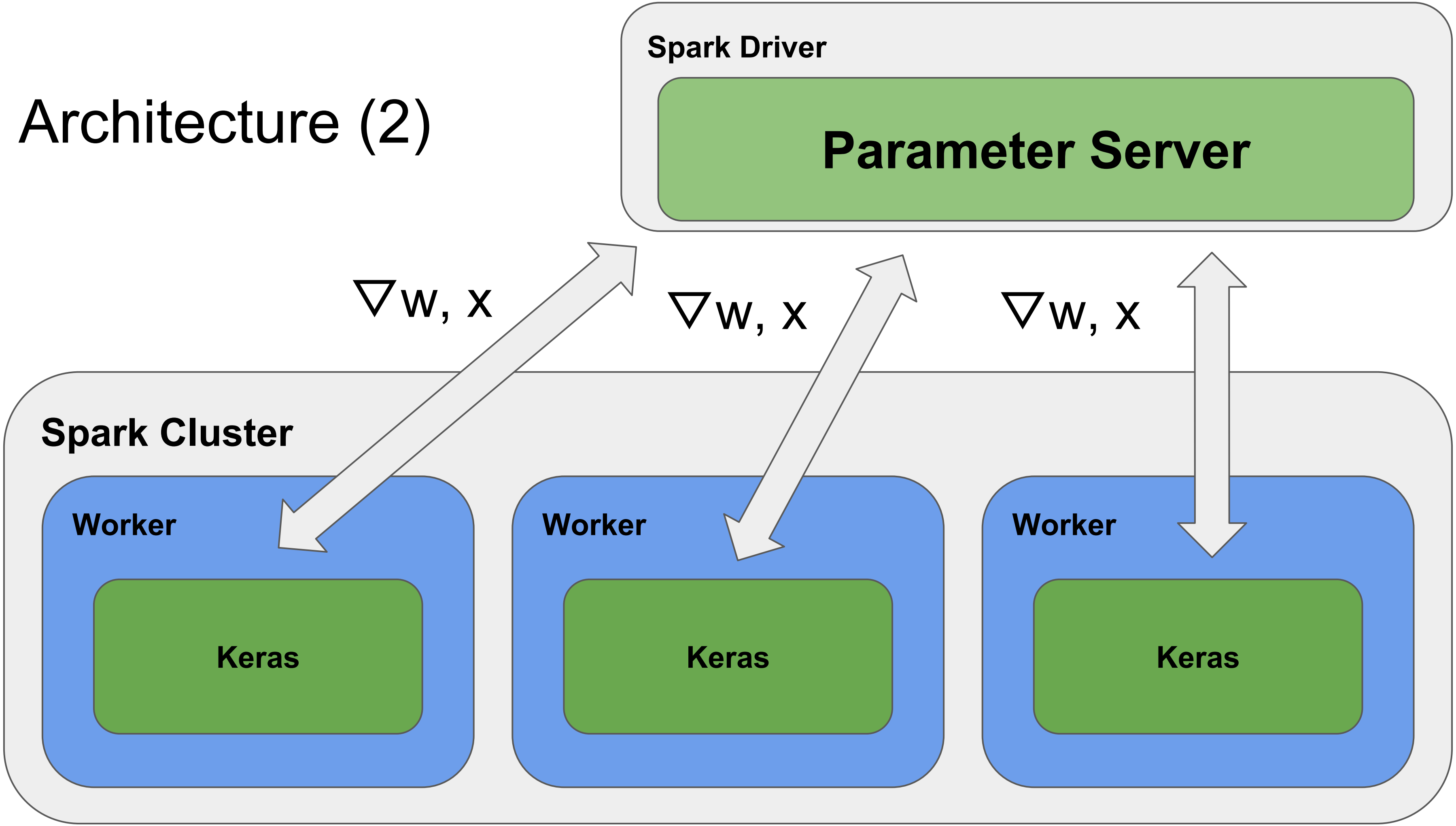


Source: Large scale distributed Deep Networks [1]

Architecture (1)



Architecture (2)



Architecture (3)

```
def train(self, data):
    self.start_service()  # Start parameter server
    worker = self.allocate_worker()
    numPartitions = data.rdd.getNumPartitions()
    if numPartitions > self.num_workers:
        data = data.coalesce(self.num_workers)
    else:
        data = data.repartition(self.num_workers)
    data.rdd.mapPartitionsWithIndex(worker.train).collect() # Train models
    self.stop_service()
```

Worker code as a Spark mapping function.

Source: github.com/JoeriHermans/dist-keras/blob/master/distkeras/distributed.py

Advantages of using Apache Spark

- **Scale out** on cluster level (more cores, faster training).
- Commodity hardware.
- Spark will handle “**bigger than memory**” issue and **parallelization**.
- Easy to provide compatibility with Spark API’s / Python:
  - SparkSQL (DataFrames)
  - Distributed preprocessing of dataset.
  - Evaluation metrics.
  - …

Related Work (1)

- Elephas (https://github.com/maxpumperla/elephas):
    - **Fits requirements** (Keras, Apache Spark).
    - Patched to work with Keras 1.0+, but testing indicates that it does not work on a cluster.
    - Distributed **gradient update is naive** (but could not test).
    - Does not support Spark 2.0, nor it supports Spark DataFrames.
    - **Not easily extendable** to implement new algorithms with different communication protocols.

Related Work (2)

- SparkNet (https://github.com/amplab/SparkNet)
  - Built using **Caffe**
  - TensorFlow supported
  - A lot of **freedom**, **but** distributed optimizer needs to be implemented with every model.

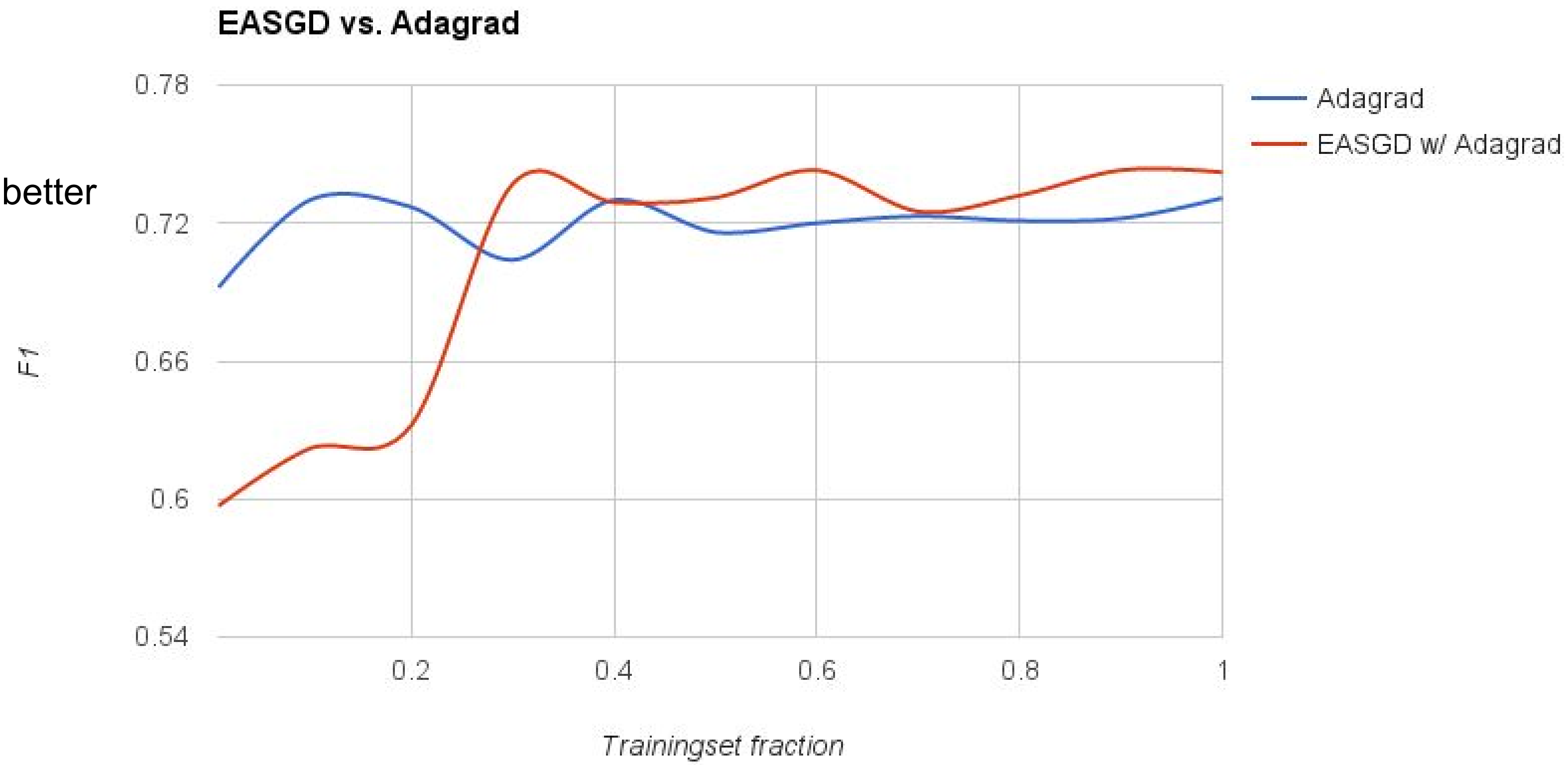
- Deeplearning4j (http://deeplearning4j.org/spark)
  - Java API
  - Again, **naive gradient update** (averaging)

Experimental setup

- Evaluating EASGD [1] (Elastic Averaging SGD)
- nb_epoch = 1, 24 cores used when running EASGD
- **Model:** 380402 trainable parameters
- **Data:** only 250000 instances
    - **Training:** 90% (variable)
    - **Test:** 10%

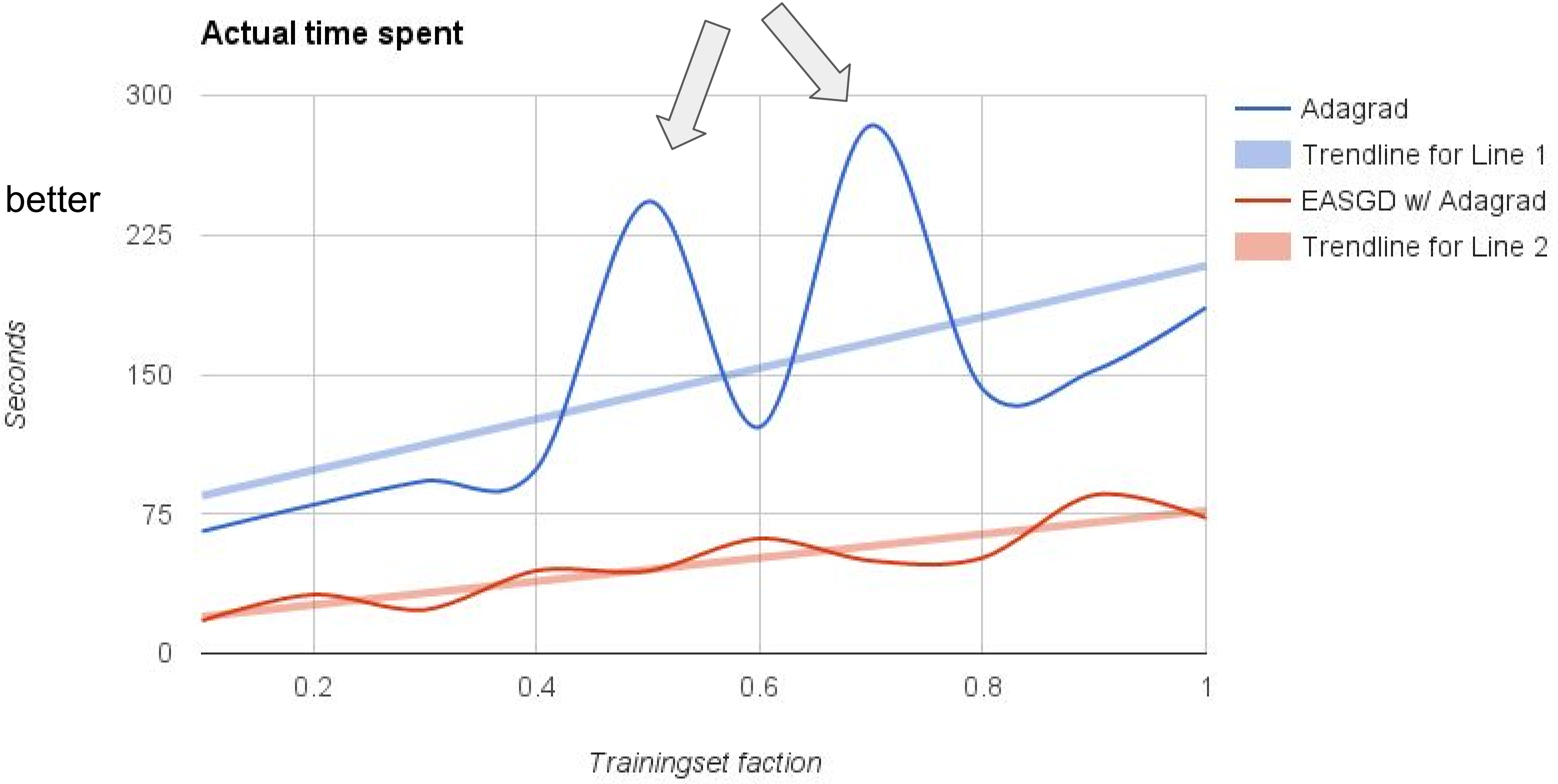
- Evaluation criteria:
    - Convergence
    - CPU usage
    - Scaling

Higher is better

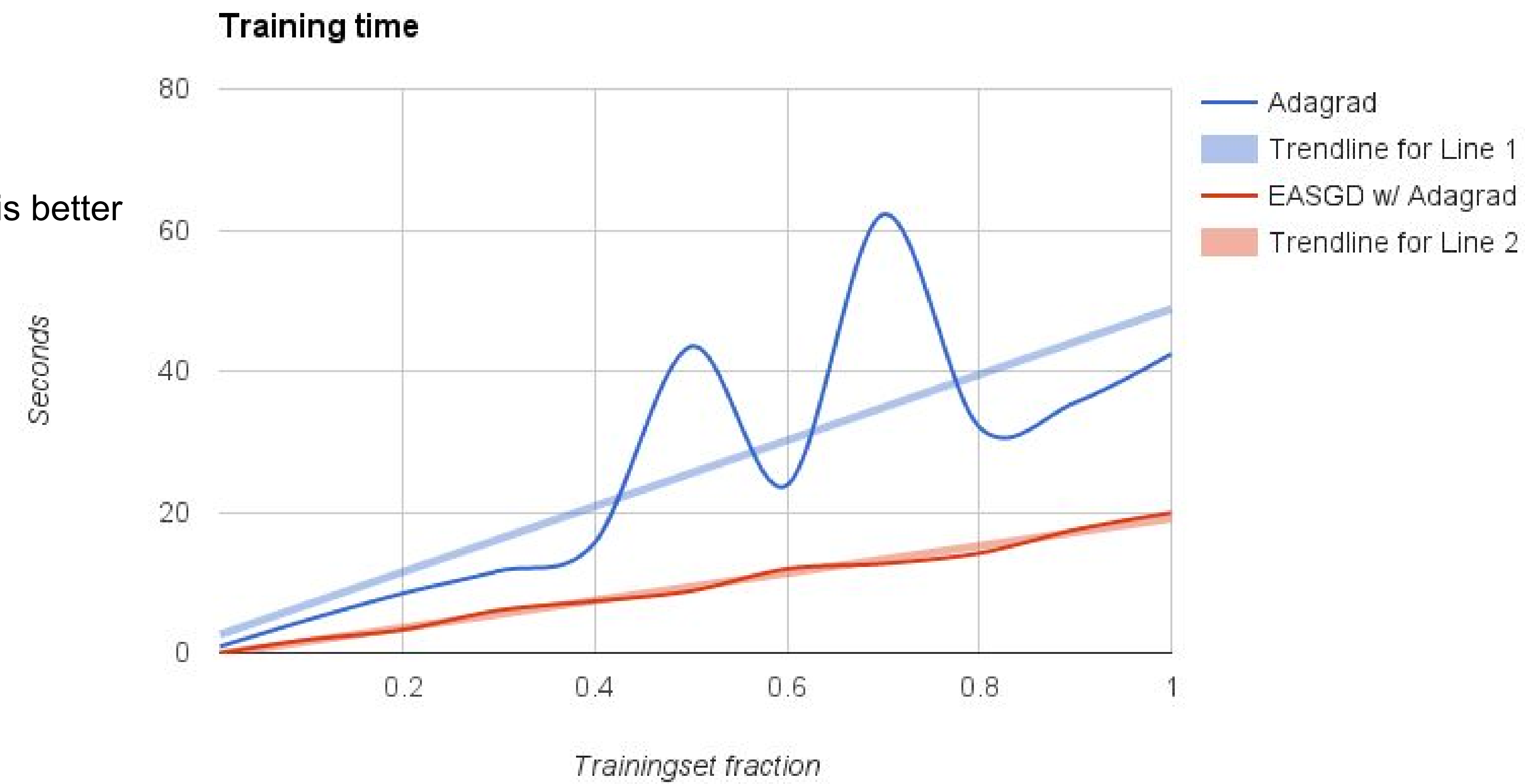


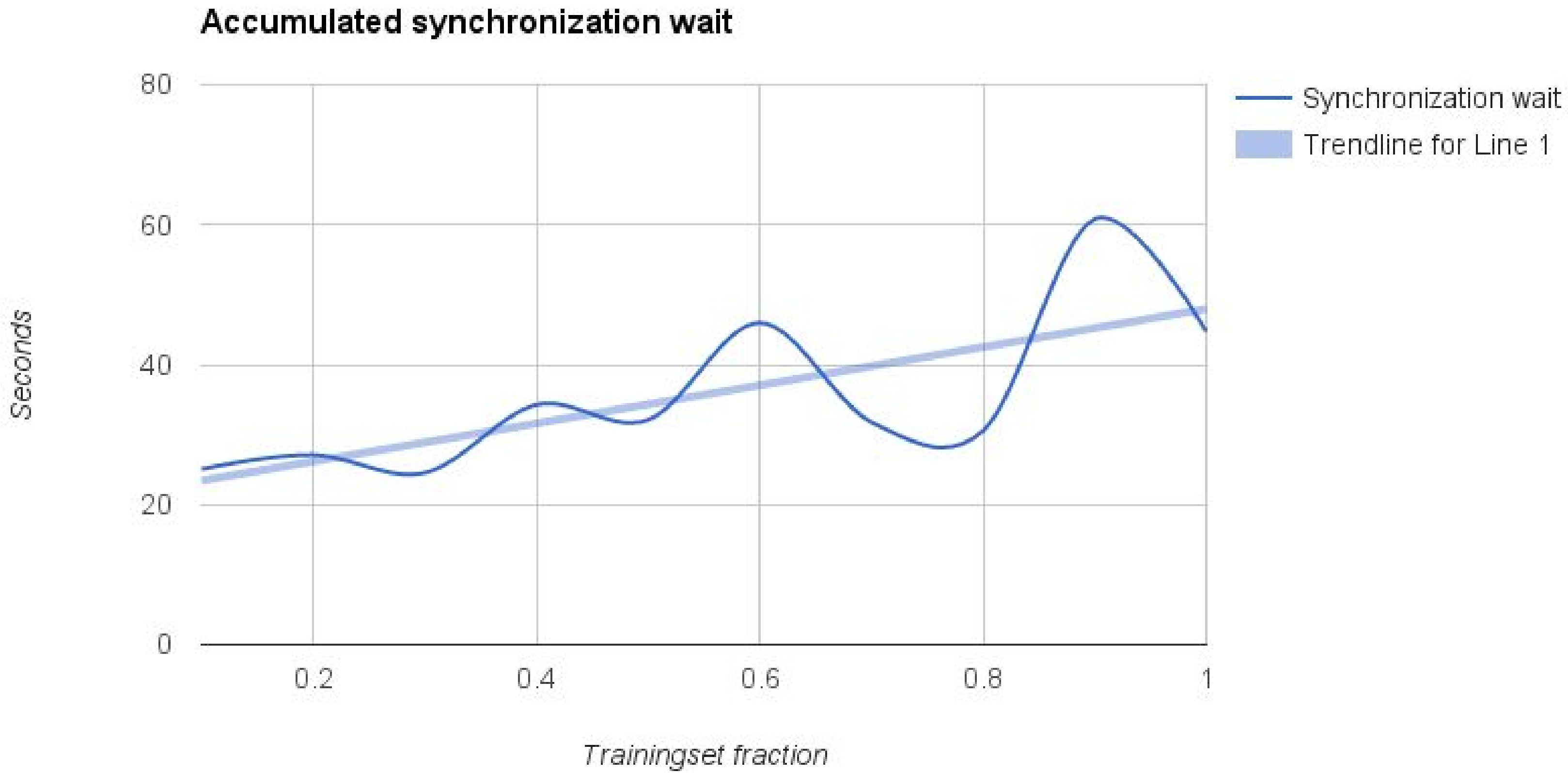
Outliers (loaded clusternode)

Lower is better



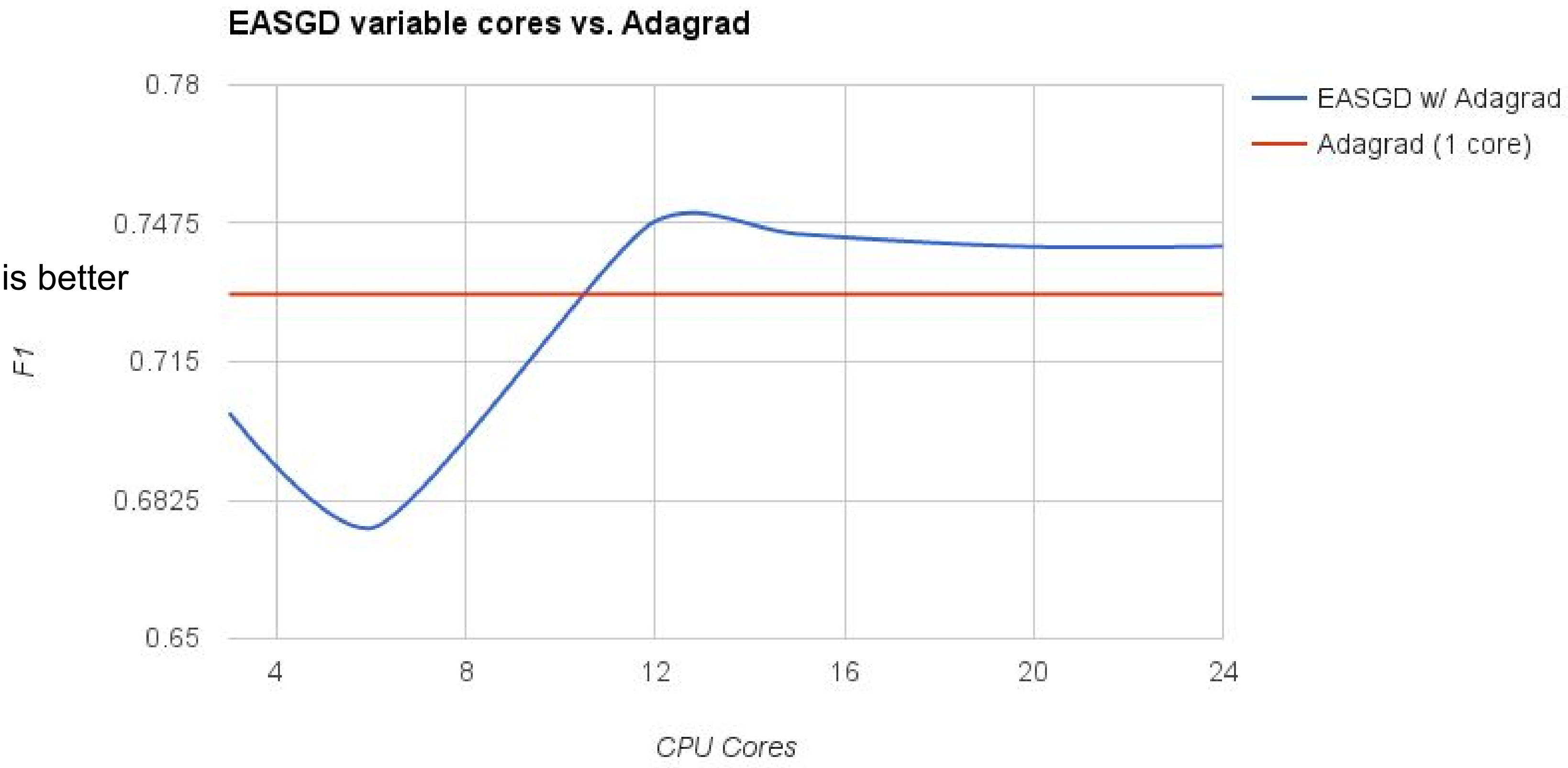
Lower is better





A lot of time is spent on the synchronization of the workers!

Higher is better



Conclusion / Future Work

- Source code of framework is available as Open Source software.
- EASGD implemented and evaluated. Claims of paper are confirmed.
- Abstraction to easily experiment new distributed algorithms.

- More experiments to understand scaling and convergence (10F-CV).
- Asynchronous algorithms
- Experiments with parameterization of EASGD.
- **Other features requested by community?**

References

- *Large Scale Distributed Deep Networks*. Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang and Andrew Y. Ng. NIPS 2012. [1]

- Zhang, S., Choromanska, A. E., & LeCun, Y. (2015). Deep learning with elastic averaging SGD. In *Advances in Neural Information Processing Systems* (pp. 685-693). [2]

- https://github.com/JoeriHermans/dist-keras/ [3]

- SparkNet: Training Deep Networks in Spark. http://arxiv.org/pdf/1511.06051v4.pdf [4]

Questions?