



# Machine Learning for CI

Kyra Wulffert

[kwulffert@yahoo.com](mailto:kwulffert@yahoo.com)

Matthew Treinish

[mtreinish@kortar.org](mailto:mtreinish@kortar.org)

Andrea Frittoli

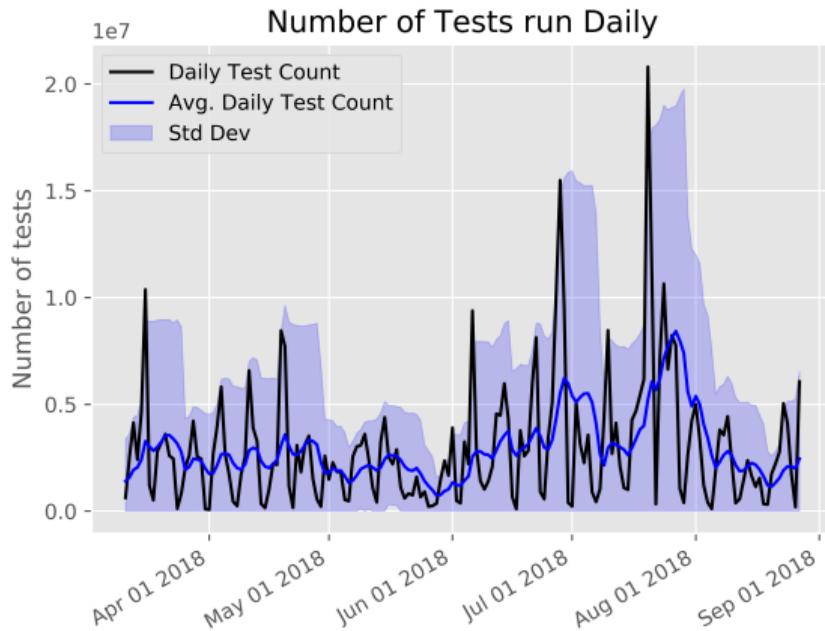
[andrea.frittoli@gmail.com](mailto:andrea.frittoli@gmail.com)

August 29, 2018

[https://github.com/afrittoli/ciml\\_talk](https://github.com/afrittoli/ciml_talk)



# CI at Scale



Source: subunit2sql-graph dailycount

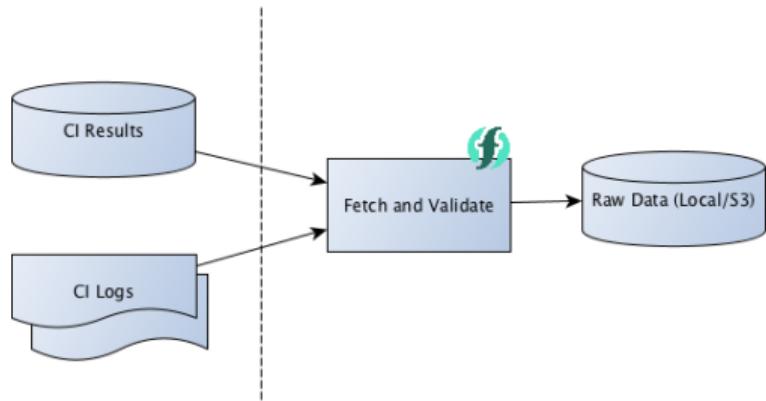
- ▶ Continuous Integration
- ▶ Continuous Log Data
- ▶ Lots of data, little time
- ▶ Triaging failures?
- ▶ AI to the rescue!

## The OpenStack use case

- ▶ Integration testing in a VM
- ▶ System logs, application logs
- ▶ Dstat data
- ▶ Gate testing
- ▶ Not only OpenStack

Normalized system average load for different examples

# Collecting data



Data caching diagram

- ▶ Automation and repeatability
- ▶ Light-weight data validation
- ▶ Object storage for data
- ▶ Periodic Action on OpenWhisk

# Training workflow

- ▶ Visualize data
- ▶ Define a dataset
- ▶ Define an experiment
- ▶ Run the training
- ▶ Collect results
- ▶ Visualize data

```
# Build an s3 backed dataset
ciml-build-dataset --dataset cpu-load-1min-dataset \
--build-name tempest-full \
--slicer :2000 \
--sample-interval 10min \
--features-regex "(usr|1min)" \
--class-label status \
--tdt-split 7 0 3 \
--data-path s3:// cimlrawdata \
--target-data-path s3:// cimldatasets
```

```
# Define a local experiment
ciml-setup-experiment --experiment dnn-5x100 \
--dataset cpu-load-1min-dataset \
--estimator tf.estimator.DNNClassifier \
--hidden-layers 100/100/100/100/100 \
--steps $(( 2000 / 128 * 500 )) \
--batch-size 128 \
--epochs 500 \
--data-path s3:// cimldatasets
```

## Training workflow

- ▶ Sharing data, experiments and results
- ▶ Helm deployment
- ▶ Ffdl <- s3 input
- ▶ Experiment / dataset in Cloudant / via OpenWhisk API
- ▶ TBD build a diagram for the training pipeline

## Infrastructure for prediction

- ▶ MQTT - near real time
- ▶ Trained model on storage
- ▶ Prediction in storage
- ▶ OpenWhisk trigger -> comment on Gerrit/Github
- ▶ Build a diagram for the NRT pipeline

## Data Selection

- ▶ What is dstat data
- ▶ Experiment Reproducibility
- ▶ Dataset selection
  - ▶ Feature selection
  - ▶ Sampling selection

## Data Normalization

- ▶ Unrolling
- ▶ Normalizing
- ▶ Graphs of normal and normalized features

## Building the dataset

- ▶ Split in training, dev, test
- ▶ Store normalized data on s3
- ▶ Input function for training
- ▶ Input function for evaluation

## DNN - Binary Classification

- ▶ Varying feature selection
- ▶ TBD Graph of accuracy with different features

## DNN - Binary Classification

- ▶ Varying data sampling
- ▶ Looking for Configurations
- ▶ TBD Graph of accuracy with different sampling

## DNN - Changing test job

- ▶ Train with a CI Job
- ▶ Evaluating with another CI Job (as well)

## DNN - Multi Class

- ▶ Detecting the Cloud Provider
- ▶ Growing back number of features
- ▶ Reducing down-sampling

## DNN - Multi Class

- ▶ Playing with the network topology

## DNN - Multi Class

- ▶ Reducing the number of classes
- ▶ What does that mean
- ▶ Why did it work

## DNN - Changing test job

- ▶ Train with a CI Job
- ▶ Evaluating with another CI Job (as well)

# Conclusions

- ▶ Summary on DNN single class
- ▶ Summary on DNN multi class
- ▶ Collect data
- ▶ Know your data
- ▶ Work with cloud tools

## Future Work

- ▶ Complete setup of the pipeline
- ▶ Human curated dataset for supervised training
- ▶ Making our life easier
- ▶ Integrate with real life CI system
- ▶ Explore job portability
- ▶ Tune optimization for quick convergence

Thank you!  
Questions?