TRADEMOB

Real-Time Monitoring of Distributed Systems

Buzzwords, Berlin, 2015/06/02 – Nakul Selvaraj & Tobias Kuhn

Measure Anything.
Measure Everything!

Trending Topic

#AnomalyDetection

Requirements

- (Near) Real-**Time**
- **Isolation**
- Low Footprint
- Extensible

Anna-Molly

Scope Metrics

- Application Metrics
- System Metrics

Scopes:

- Instance
- Cluster
- Service
- Data-Centre

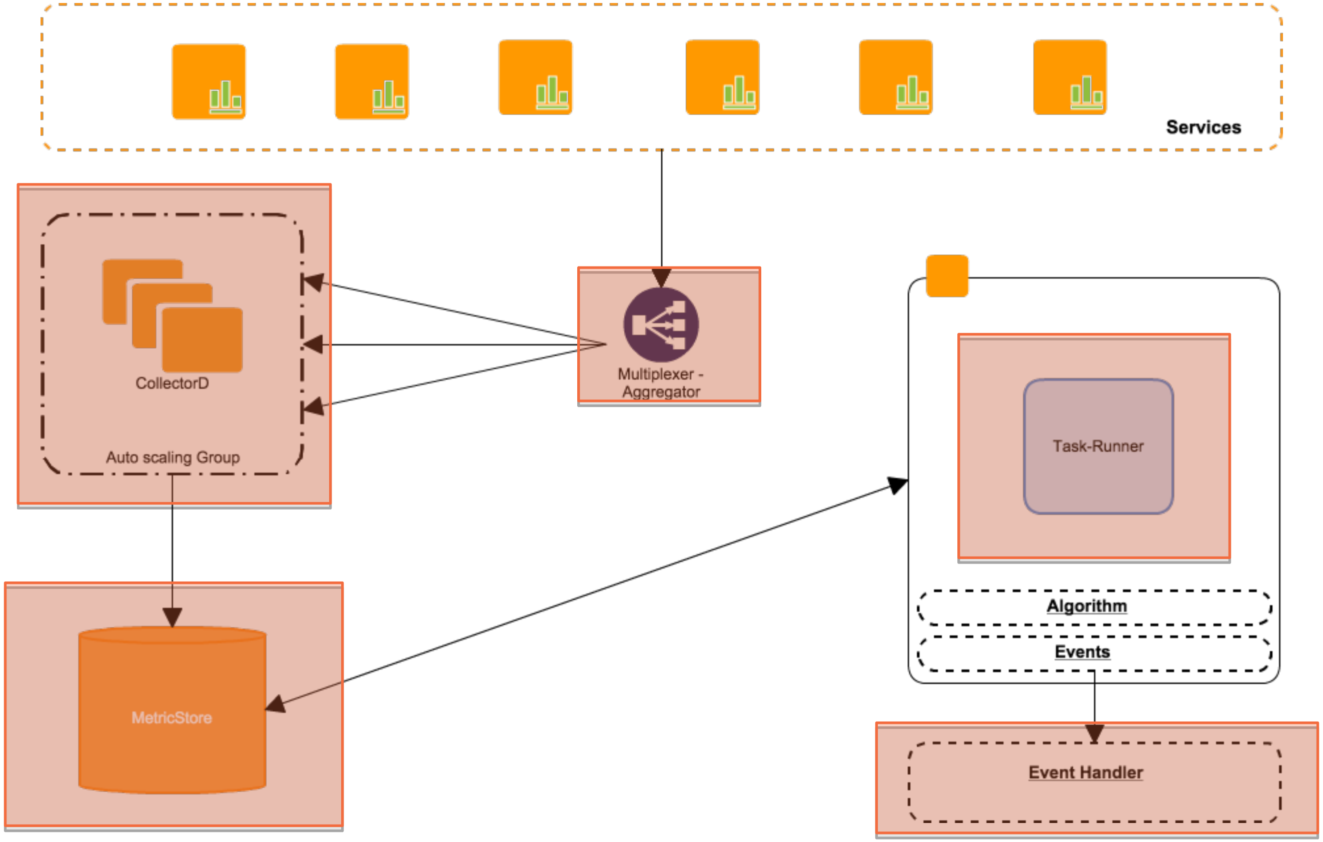
Example

host-10-0-x-x:eu-west-1:az-c:asg-20141112:service-378fc7c:cpu

Yields:

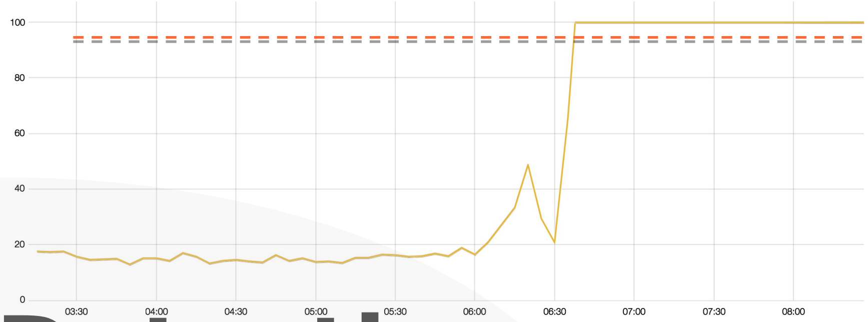
- **host-10-0-x-x:cpu**
- **eu-west-1:cpu**
- **eu-west-1:az-c:cpu**
- **eu-west-1:asg-20141112:cpu**
- **service-378fc7c:cpu**
- **service-378fc7c:eu-west-1:cpu**
- **...**

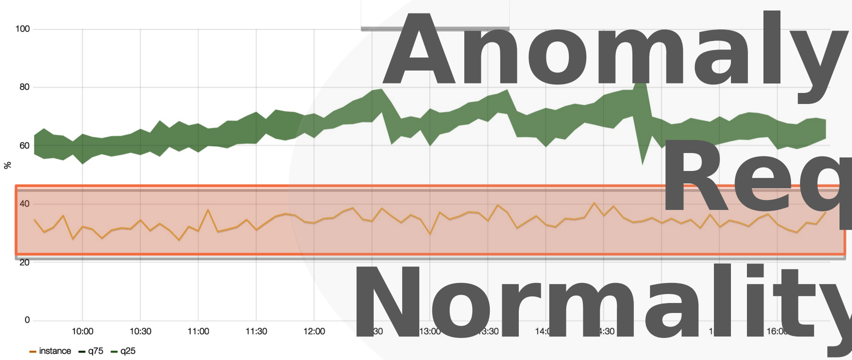
Anna-Molly


Services
CollectorD
Auto scaling Group
Multiplexer - Aggregator
Task-Runner
Algorithm
Events
MetricStore
Event Handler

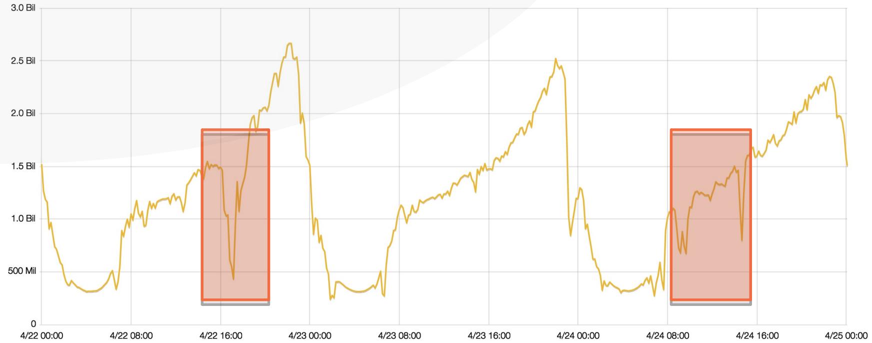
Algorithms

Anomalies

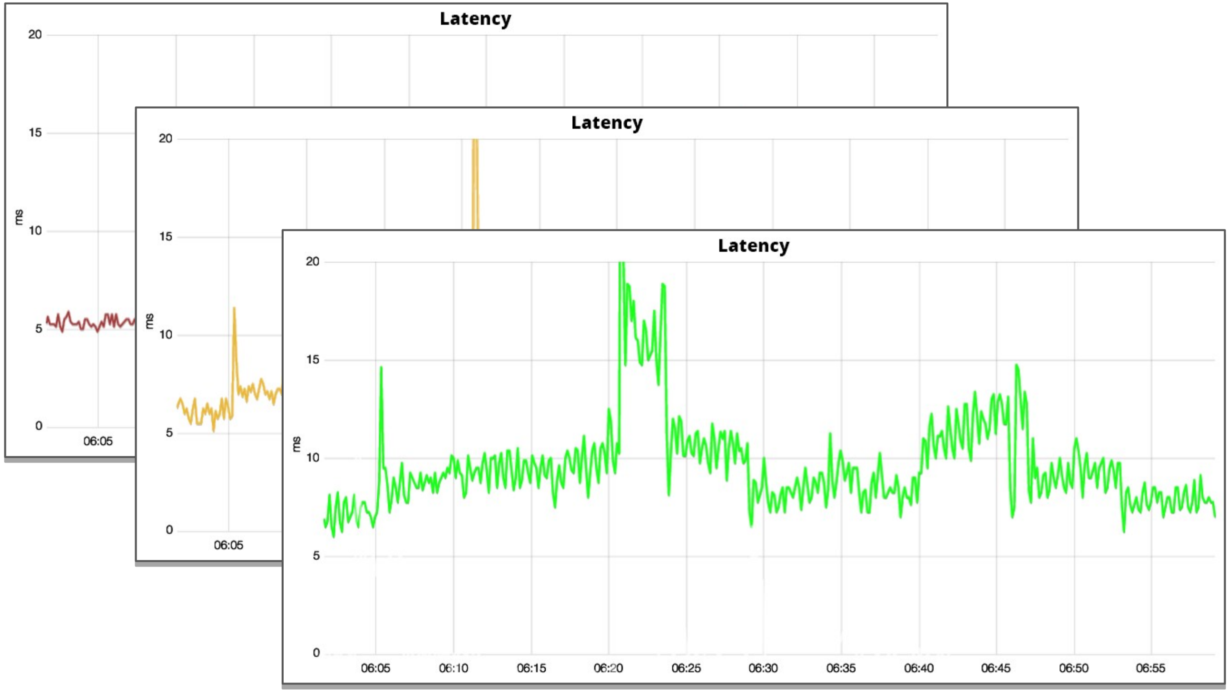




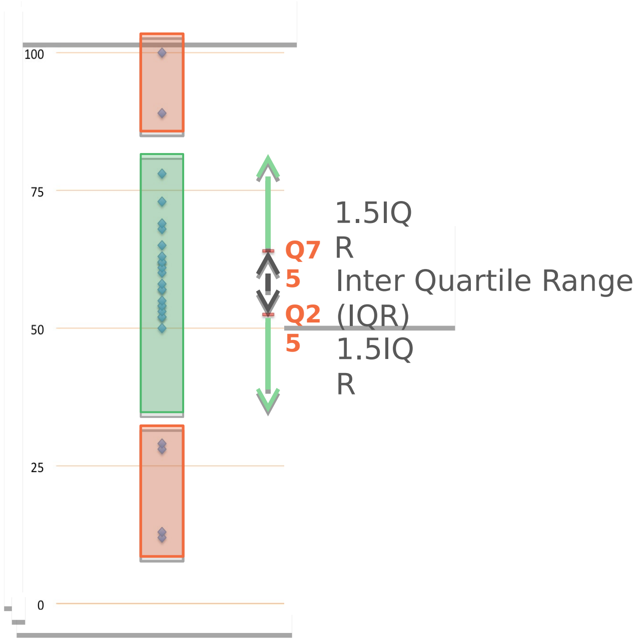
Anomaly Detection Requires Normality Definition



Distributed System Behaviour



Tukey’s Outlier Filter



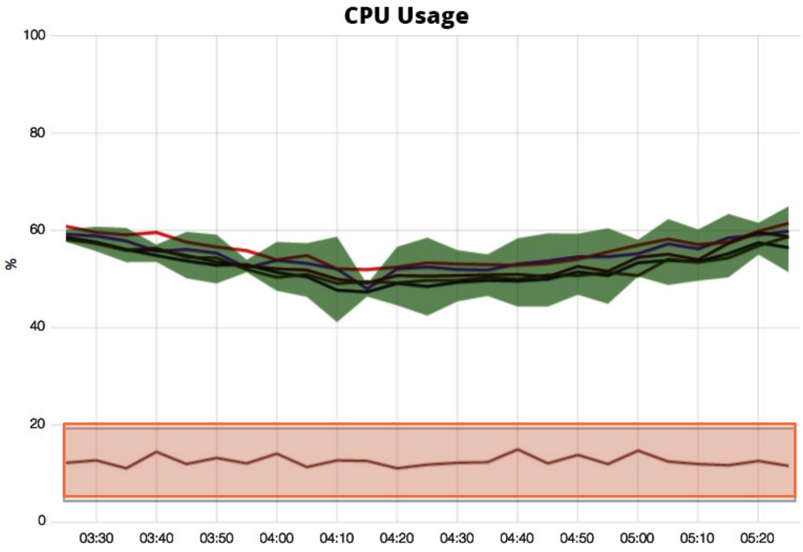
1.5IQR

Q75

Inter Quartile Range (IQR)

Q25

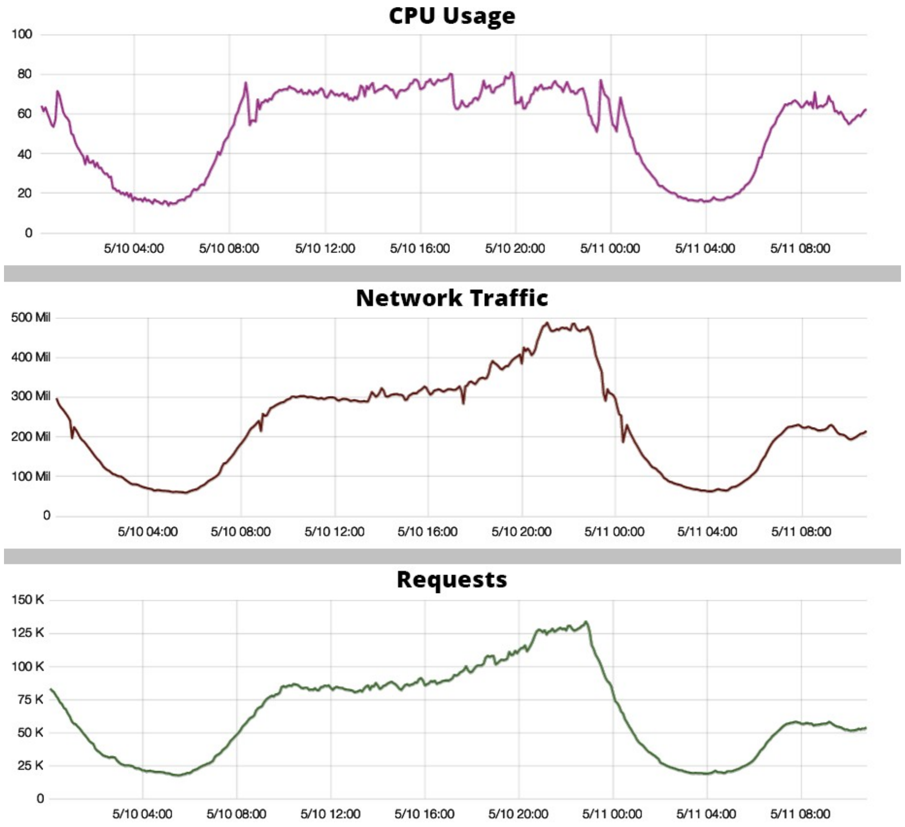
1.5IQR



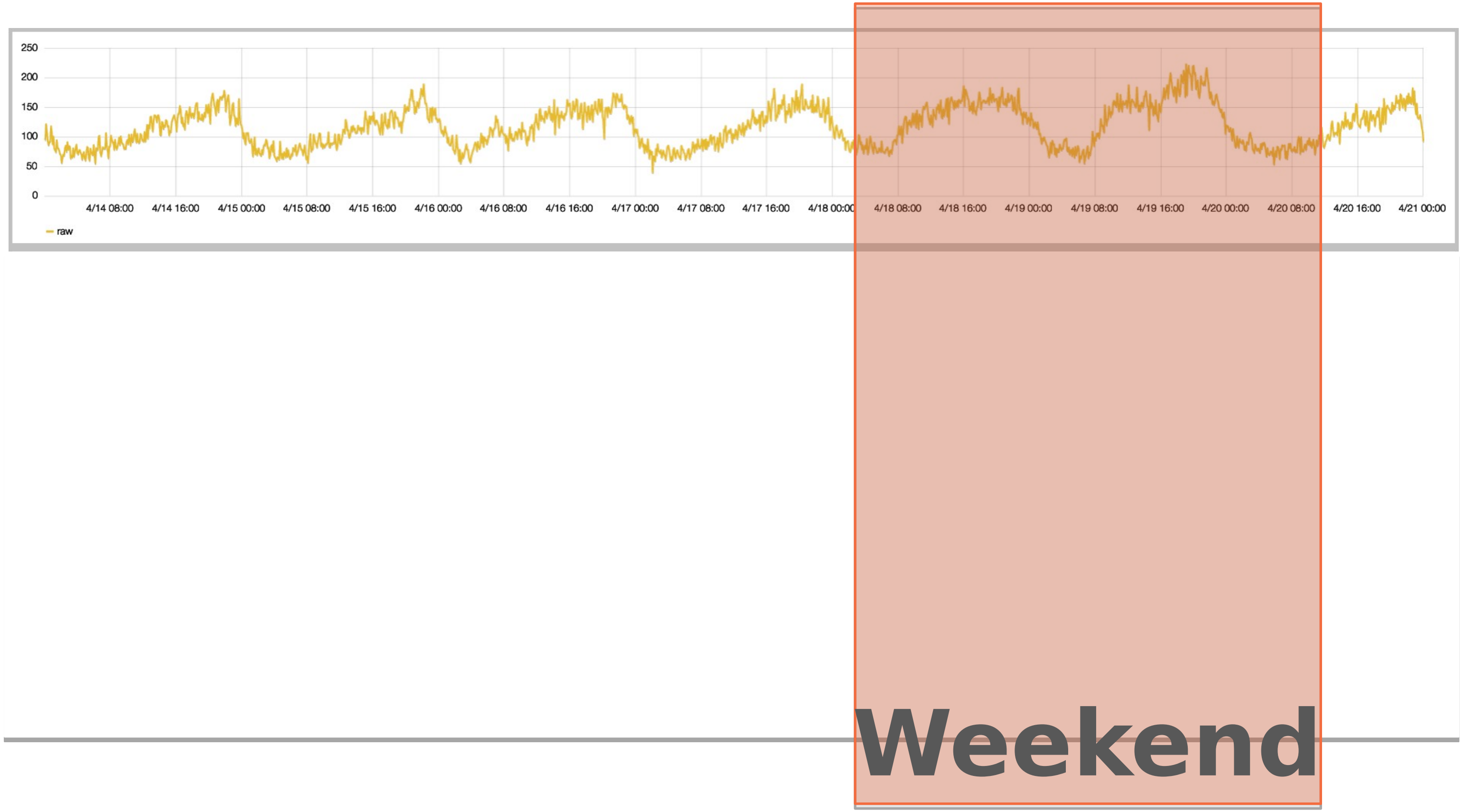
Purpose Identifying auto scaling issues (e.g. memory leaks)

Restrictions Healthy range needed

Overall System Performance



Seasonal Trend Decomposition – Algorithm

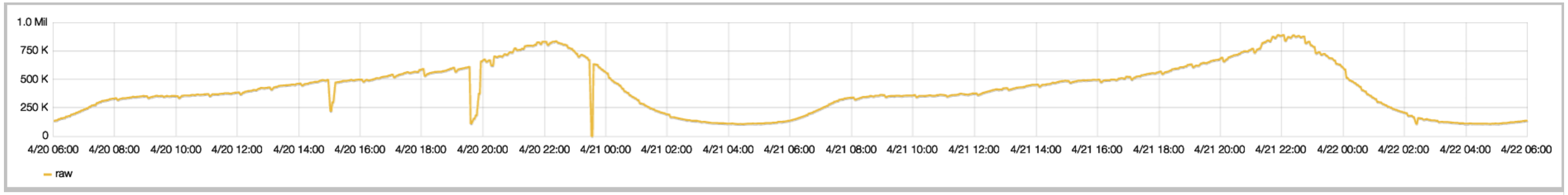


Weekend

Seasonal

Trend

Seasonal Trend Decomposition – Example

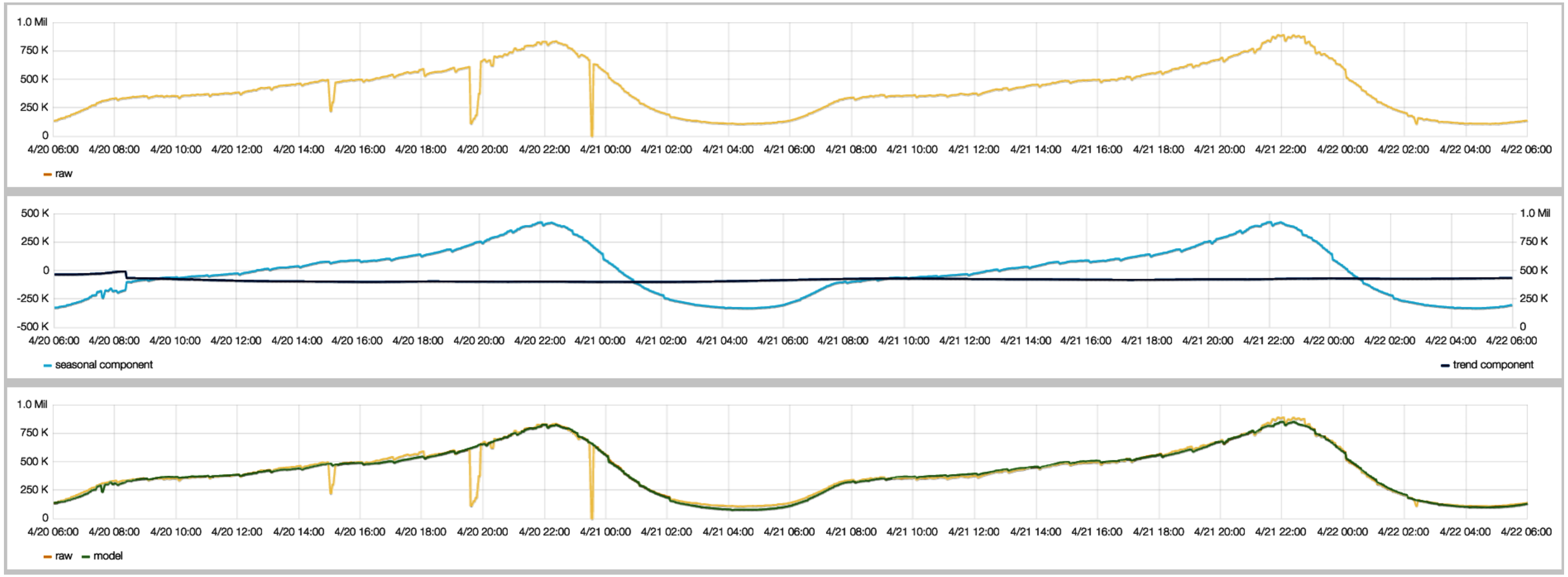


Input

Seasonal
Trend

Model

Seasonal Trend Decomposition – Example



Input

Seasonal

Trend

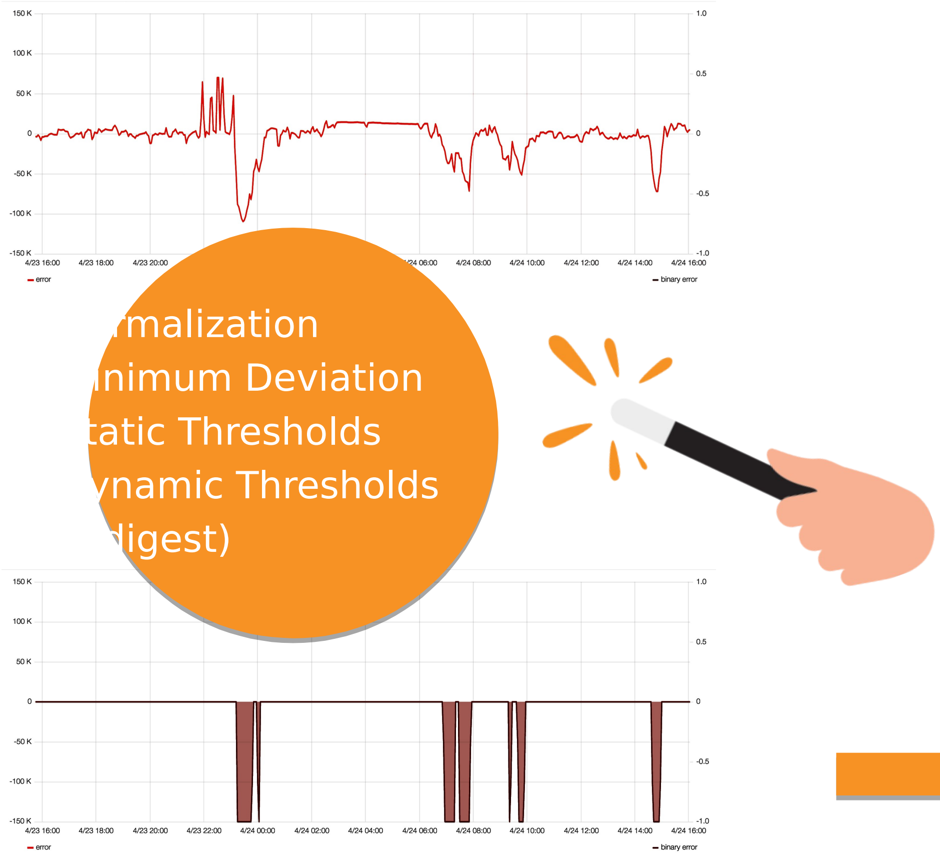
Input

Model

Deviation

Evaluation

Algorith
m



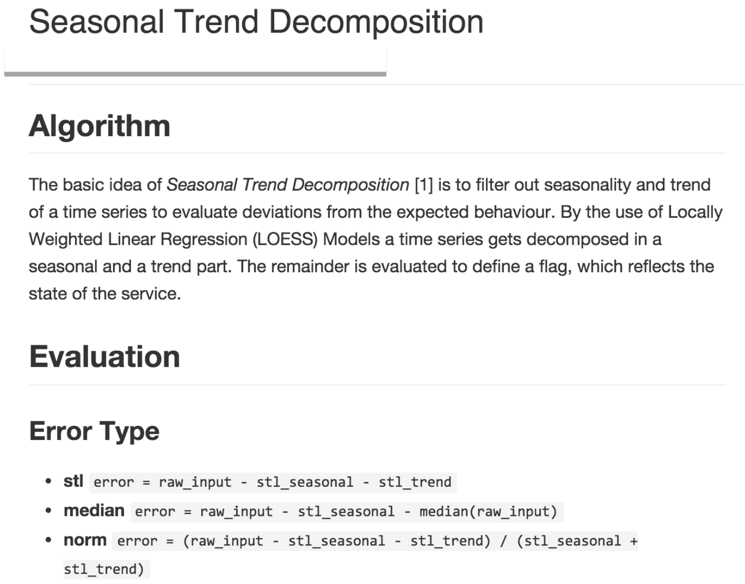
rmalization
inimum Deviation
tatic Thresholds
ynamic Thresholds
digest)

Actions

TRADEMOB

June 2015

https://github.com/trademob/anna-molly



Seasonal Trend Decomposition

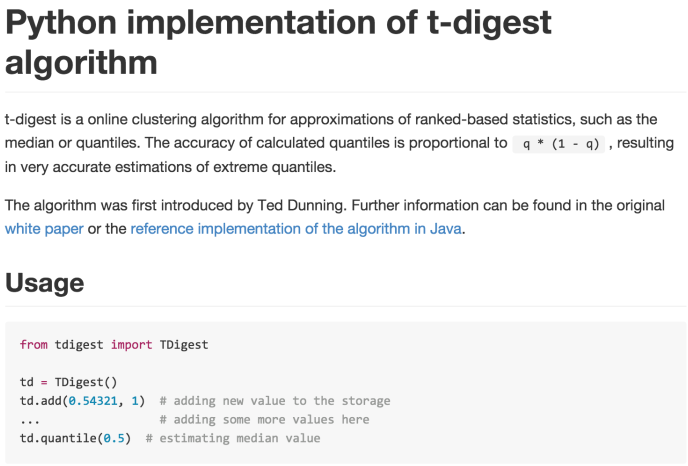
Algorithm

The basic idea of *Seasonal Trend Decomposition* [1] is to filter out seasonality and trend of a time series to evaluate deviations from the expected behaviour. By the use of Locally Weighted Linear Regression (LOESS) Models a time series gets decomposed in a seasonal and a trend part. The remainder is evaluated to define a flag, which reflects the state of the service.

Evaluation

Error Type

- **stl** `error = raw_input - stl_seasonal - stl_trend`
- **median** `error = raw_input - stl_seasonal - median(raw_input)`
- **norm** `error = (raw_input - stl_seasonal - stl_trend) / (stl_seasonal + stl_trend)`



Python implementation of t-digest algorithm

t-digest is a online clustering algorithm for approximations of ranked-based statistics, such as the median or quantiles. The accuracy of calculated quantiles is proportional to `q * (1 - q)` , resulting in very accurate estimations of extreme quantiles.

The algorithm was first introduced by Ted Dunning. Further information can be found in the original white paper or the reference implementation of the algorithm in Java.

Usage

```
from tdigest import TDigest

td = TDigest()
td.add(0.54321, 1)  # adding new value to the storage
...                 # adding some more values here
td.quantile(0.5)  # estimating median value
```

https://github.com/trademob/t-digest

Questions?

trademob.com

@trademob

Trademob GmbH