Online Metrics Prediction in Monitoring Systems

Matthieu Caneill, Noël De Palma, Ali Ait-Bachir
Bastien Dine, Rachid Mokhtari, Yagmur Gizem Cinar

April 16, 2018

IEEE INFOCOM 2018
The 8th International Workshop on Big Data in Cloud Performance (DCPerf'18)

Table of contents

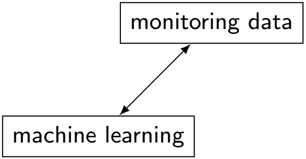
1. Introduction
2. Metrics prediction
3. Evaluation
4. Conclusion

System goal: anticipate failures

monitoring data

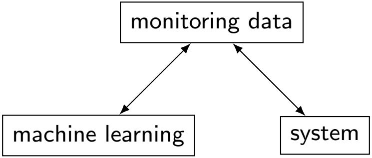
- Monitoring insights

System goal: anticipate failures



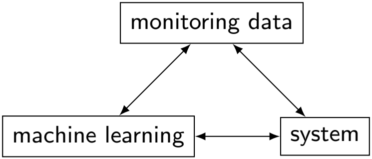
- Monitoring insights
- Failure prediction

System goal: anticipate failures



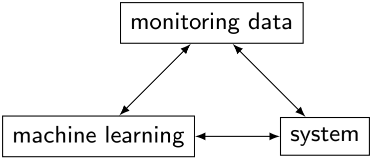
- Monitoring insights
- Failure prediction
- Infrastructure scaling

System goal: anticipate failures



- Monitoring insights
- Failure prediction
- Infrastructure scaling

System goal: anticipate failures



- Monitoring insights
- Failure prediction
- Infrastructure scaling
- More server uptime

Desired properties

- Scalable infrastructure: at least up to a few servers (around 150 CPU cores)

Desired properties

- Scalable infrastructure: at least up to a few servers (around 150 CPU cores)
- End-to-end fault tolerance: metrics can never be lost

Desired properties

- Scalable infrastructure: at least up to a few servers (around 150 CPU cores)
- End-to-end fault tolerance: metrics can never be lost
- Performances: “fast” to compute metrics predictions (low latency)

Metrics

- Monitoring metric: observation point on a server in a datacenter

Metrics

- Monitoring metric: observation point on a server in a datacenter
- System metrics: CPU load, memory, open sockets

Metrics

- Monitoring metric: observation point on a server in a datacenter
- System metrics: CPU load, memory, open sockets
- Service metrics: database status, web server response time

Metrics

- Monitoring metric: observation point on a server in a datacenter
- System metrics: CPU load, memory, open sockets
- Service metrics: database status, web server response time
- Reported by agents, processed, and stored

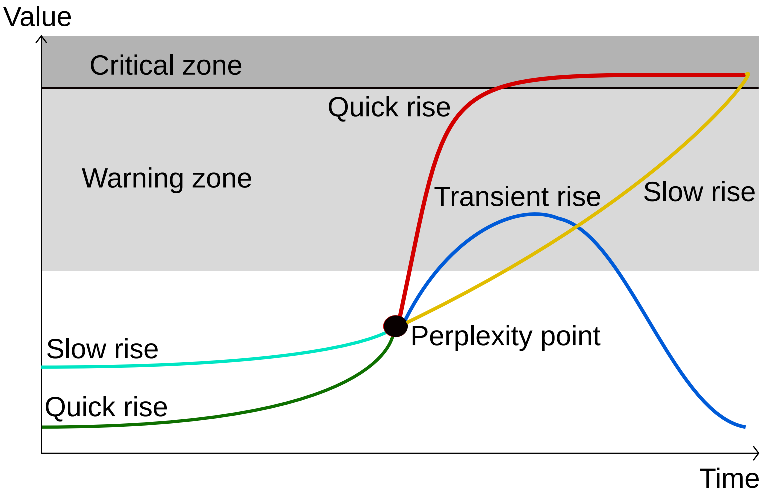
Metrics

- Monitoring metric: observation point on a server in a datacenter
- System metrics: CPU load, memory, open sockets
- Service metrics: database status, web server response time
- Reported by agents, processed, and stored
- Computed as time-series

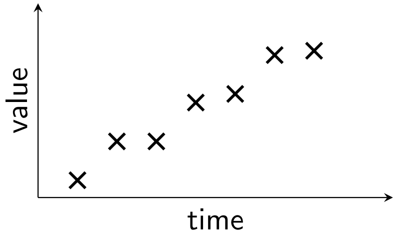
Metrics

- Monitoring metric: observation point on a server in a datacenter
- System metrics: CPU load, memory, open sockets
- Service metrics: database status, web server response time
- Reported by agents, processed, and stored
- Computed as time-series
- Associated to thresholds: warning and critical

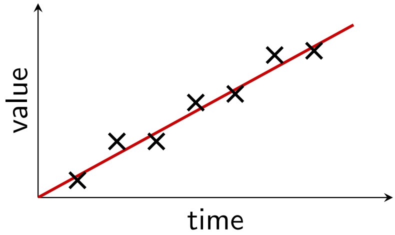
Metrics behaviour: 6 scenarios



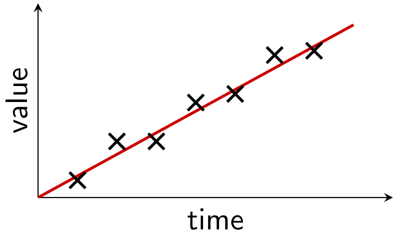
Linear regression



Linear regression

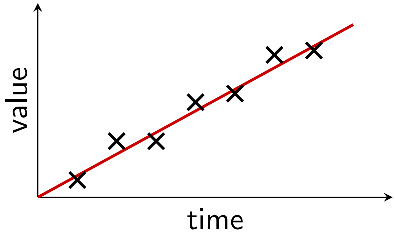


Linear regression



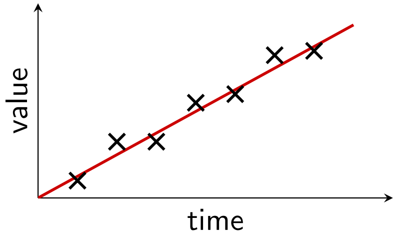
- Ability to identify local trends (few hours)

Linear regression



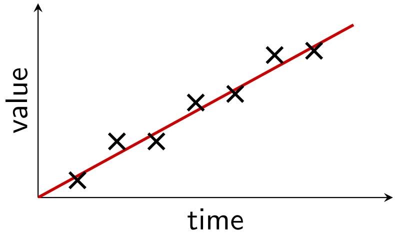
- Ability to identify local trends (few hours)
- Fast to compute

Linear regression



- Ability to identify local trends (few hours)
- Fast to compute
- Good candidate to avoid false positives (peaks)

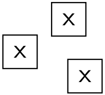
Linear regression



- Ability to identify local trends (few hours)
- Fast to compute
- Good candidate to avoid false positives (peaks)
- Library: MLlib (part of Apache Spark)

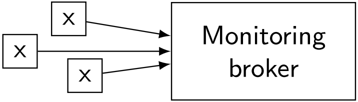
Metrics prediction

System architecture



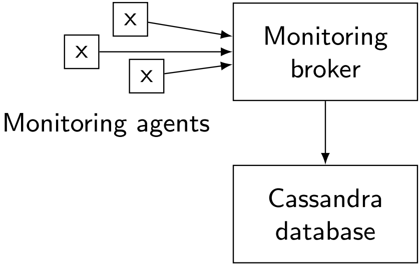
Monitoring agents

System architecture

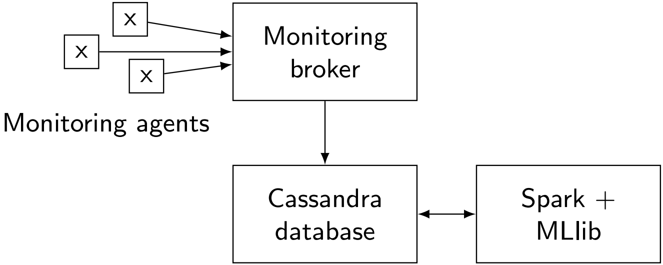


Monitoring agents

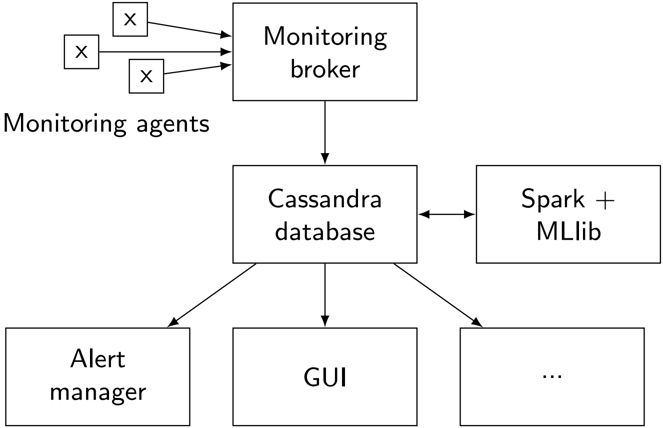
System architecture



System architecture



System architecture



Metric blacklisting

- Some metrics are too volatile and hard to predict

Metric blacklisting

- Some metrics are too volatile and hard to predict
- To avoid false positives/negatives, and save resources, they are blacklisted

Metric blacklisting

- Some metrics are too volatile and hard to predict
- To avoid false positives/negatives, and save resources, they are blacklisted
- Root Mean Square Error evaluated weekly

Metric blacklisting

- Some metrics are too volatile and hard to predict
- To avoid false positives/negatives, and save resources, they are blacklisted
- Root Mean Square Error evaluated weekly
- Metrics (temporarily) blacklisted if their RMSE $>$ threshold

Metric blacklisting

- Some metrics are too volatile and hard to predict
- To avoid false positives/negatives, and save resources, they are blacklisted
- Root Mean Square Error evaluated weekly
- Metrics (temporarily) blacklisted if their RMSE $>$ threshold
- 58.5% of the metrics have a low RMSE $\rightarrow$ good predictions

Example

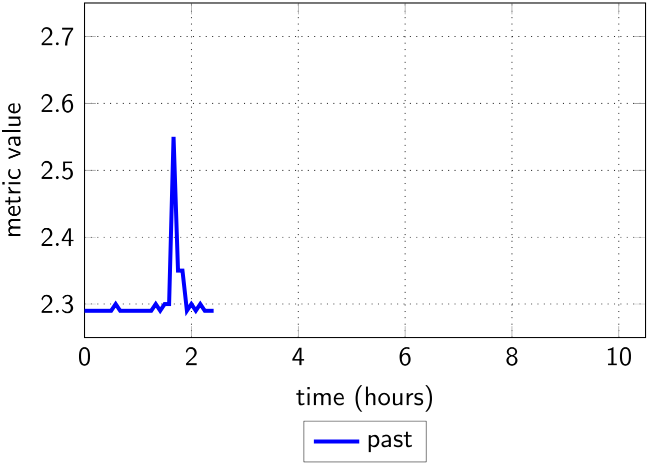


Figure: swap memory

Example

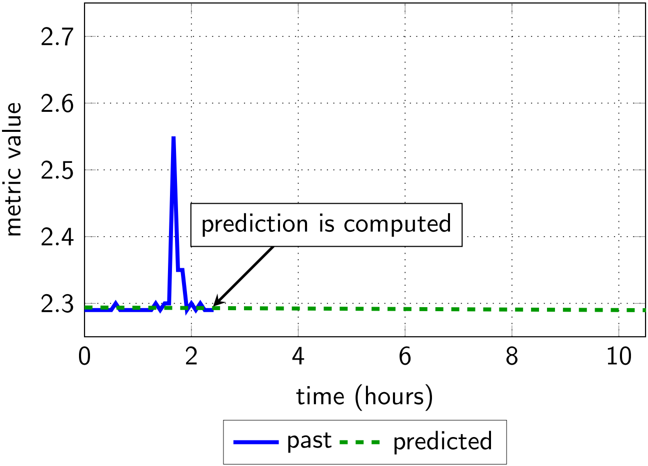


Figure: swap memory

Example

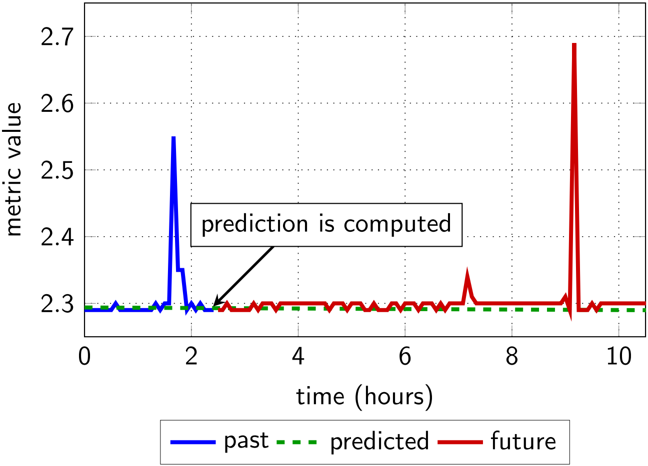


Figure: swap memory

Example

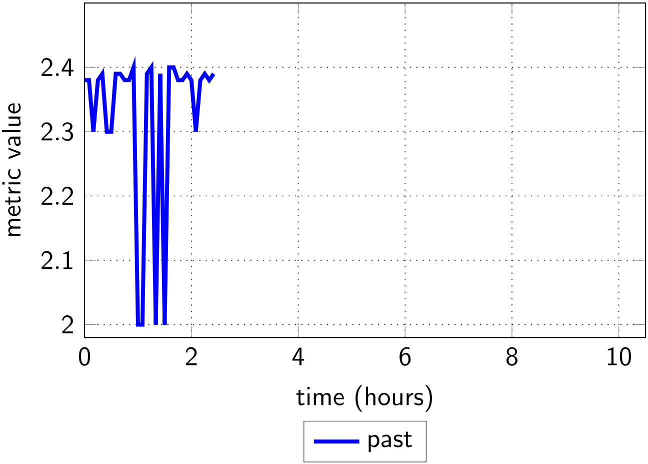


Figure: physical memory

Example

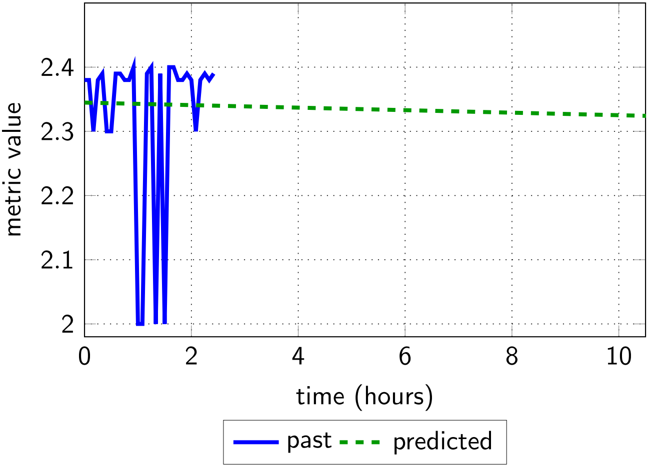


Figure: physical memory

Metrics prediction

Example

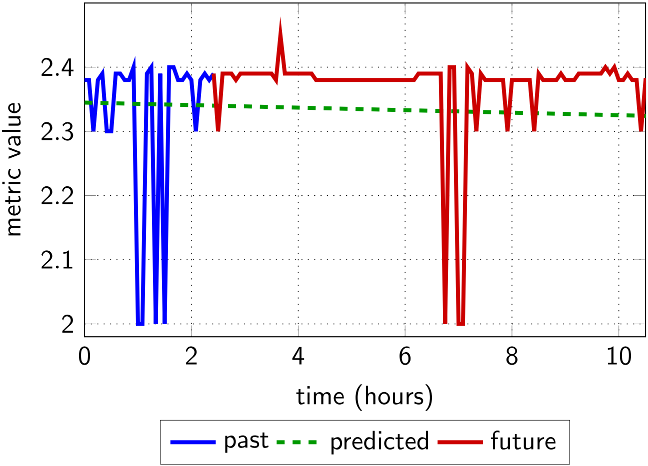


Figure: physical memory

Example

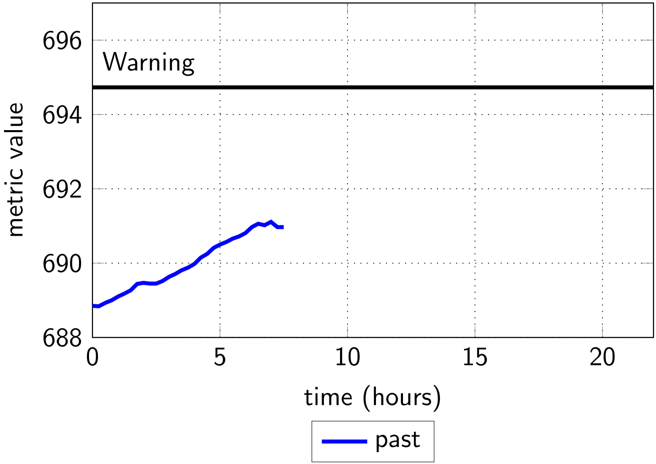


Figure: disk partition

Example

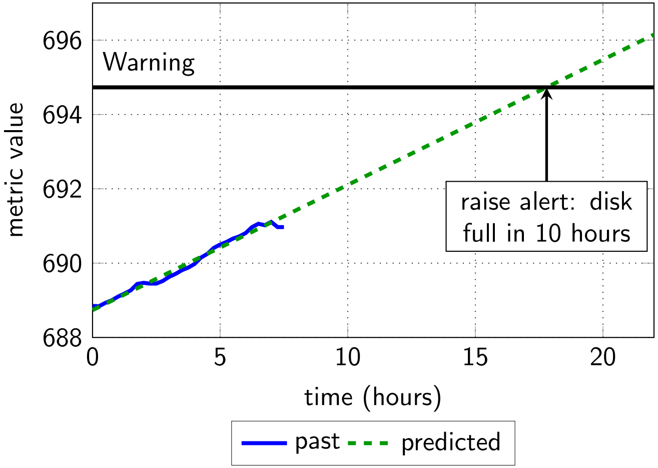


Figure: disk partition

Example

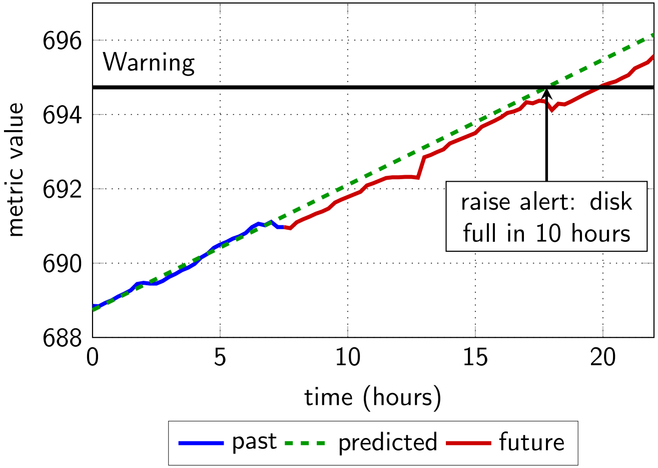
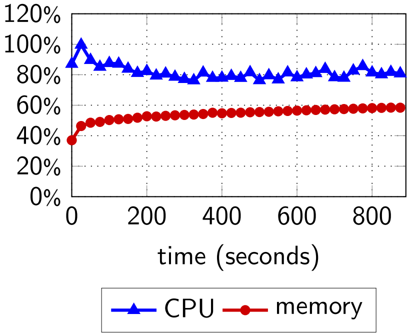


Figure: disk partition

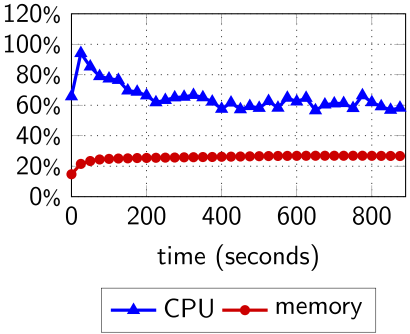
Setup

- Hardware: 4 servers (16–28 cores, 128–256 GB RAM)
- Dataset: Replay on production data recorded at Coservit
- 424 206 metrics, 1.5 billion data points monitored on 25 070 servers

CPU load and memory consumption



(a) master



(b) slave-1

Figure: Running on 4 machines and 100 cores for 15 minutes.

Time repartition

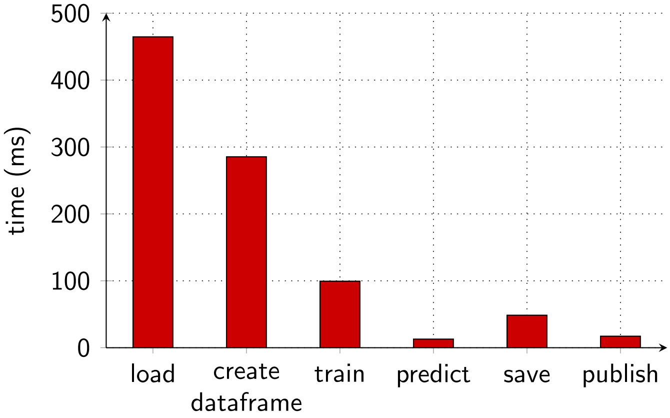


Figure: Time repartition for predicting a metric.

Load handling

- End-to-end process for the prediction of 1 metric: 1 second.

Load handling

- End-to-end process for the prediction of 1 metric: 1 second.
- One monitoring server (with 24 cores) can handle the load of 1440 metrics (at worst), which is 85 servers on average.

Load handling: linear scaling

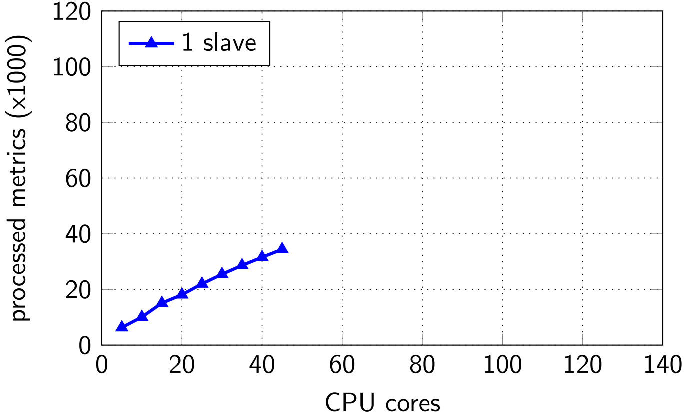


Figure: Amount of metrics handled in 15 minutes.

Load handling: linear scaling

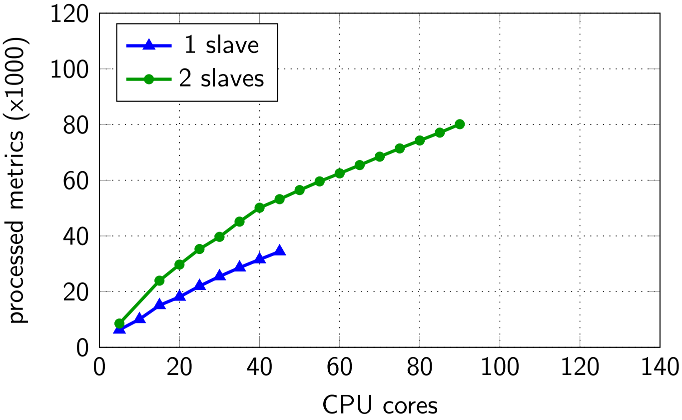


Figure: Amount of metrics handled in 15 minutes.

Load handling: linear scaling

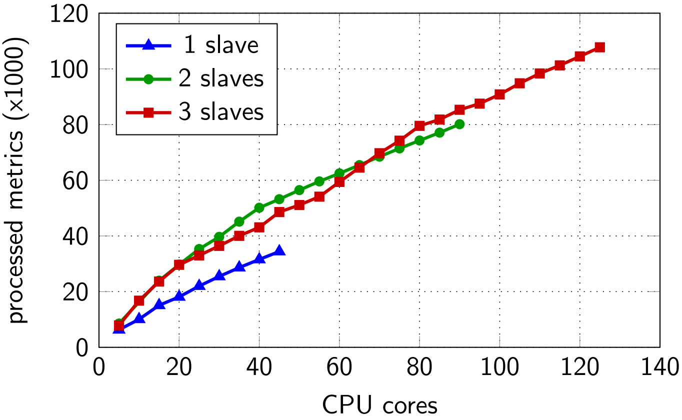


Figure: Amount of metrics handled in 15 minutes.

Related work

Positioning

No published work exhibits the same system (end-to-end system for monitoring metrics prediction, storage and blacklisting).

Prediction models

- Hardware failures [CAS12]

Related work

Positioning

No published work exhibits the same system (end-to-end system for monitoring metrics prediction, storage and blacklisting).

Prediction models

- Hardware failures [CAS12]
- Capacity planning (e.g. Microsoft Azure [mic])

Related work

Positioning

No published work exhibits the same system (end-to-end system for monitoring metrics prediction, storage and blacklisting).

Prediction models

- Hardware failures [CAS12]
- Capacity planning (e.g. Microsoft Azure [mic])
- Datacenter temperature (e.g. Thermocast [LLL+11])

Related work

Positioning

No published work exhibits the same system (end-to-end system for monitoring metrics prediction, storage and blacklisting).

Prediction models

- Hardware failures [CAS12]
- Capacity planning (e.g. Microsoft Azure [mic])
- Datacenter temperature (e.g. Thermocast [LLL+11])
- Monitoring metrics (e.g. Zabbix [zab] with manual tuning)

Future work

- Experiment with more complex ML algorithms
- Predictions on long-term global trends
- Link with ticketing mechanism

Thanks! Questions?

T. Chalermarrewong, T. Achalakul, and S. C. W. See.
Failure prediction of data centers using time series and fault tree analysis.
In *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, pages 794–799, Dec 2012.

Lei Li, Chieh-Jan Mike Liang, Jie Liu, Suman Nath, Andreas Terzis, and Christos Faloutsos.
Thermocast: A cyber-physical forecasting model for datacenters.
In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1370–1378, New York, NY, USA, 2011. ACM.

Bibliography II

Microsoft cloud azure.
https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice.

Zabbix prediction triggers.
https://www.zabbix.com/documentation/3.0/manual/config/triggers/prediction.