A Software Failure Detection Mechanism for Web Server Software using On-line Update of Bayesian Network with Auto-Selection of Training Data

Akira Tameoka, Takashi Ishio, Makoto Matsushita, Katsuro Inoue

Osaka University

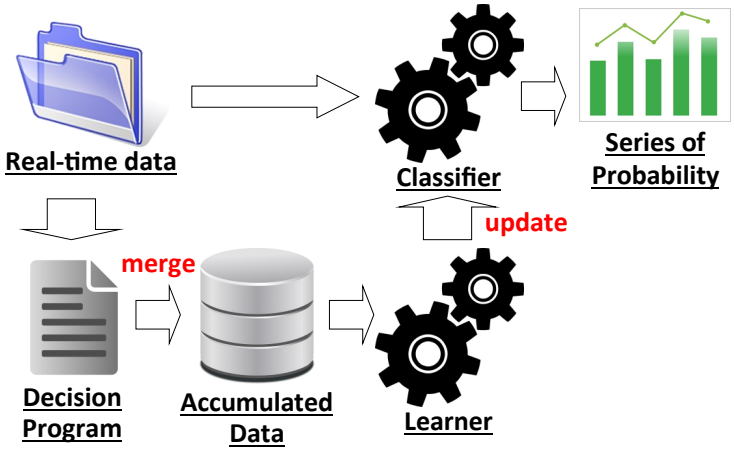
Background

- System administrators would like to make a decision to detect software failure with individual metrics (ex. CPU utilization, Memory usage, Disk I/O).
- The failure, for example, is defined that the maximum response time of the web server is more than three seconds.

Machine learning for failure detection

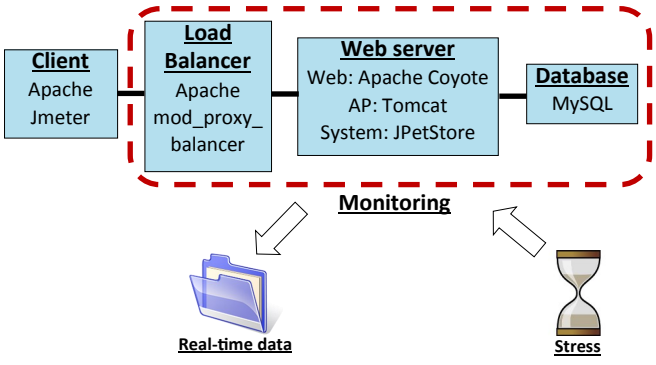
- Machine learning is used for failure detection.
- Learning data selection is needed for real-time update, but it is difficult since raw data retrieved from system would be too much.

Approach



- Decision Program gets Real-Time Data and decide whether the data is usable or not (using clustering mechanism). Usable data is merged to the Accumulated Data.
- Accumulated Data is input to learner.
- Classifier gets Real-Time Data and Learner, and outputs Series of Probability of failure.
- Using this mechanism, it is possible to update the training data within one minute.

Environment

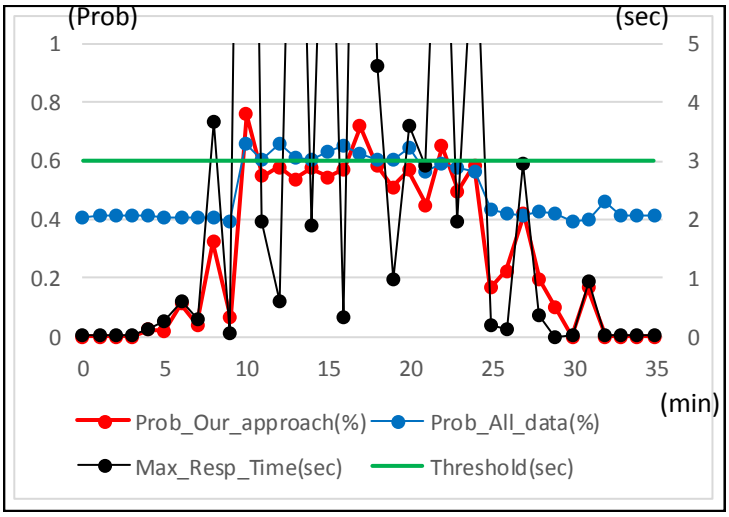


Correcting system data in real-time

Resources	Monitoring metrics
CPU	Utilization(%)
Memory	Usage (bytes)
Network	Send and receive (bytes/sec)
Disk	I/O operation (ops/sec)
Web access	Request, max/avg response time

We collect web access data from Load Balancer only.

Results



When the max response time is above the threshold, we define the interval as failure.

Evaluation

Element	All data	Our approach
1) # of failures	107	
2) Detected failures	60	90
2) / 1)	0.561	**0.841**

Our approach's value is much better than all data's.

Discussion

Element	All data	Our approach
3) # of alerts raised	64	102
4) True alerts	57	90
5) 4) raised just on time	22	**48**
4) / 3)	0.891	**0.882**

As a result, we confirmed that we could find software failure in advance with our approach.