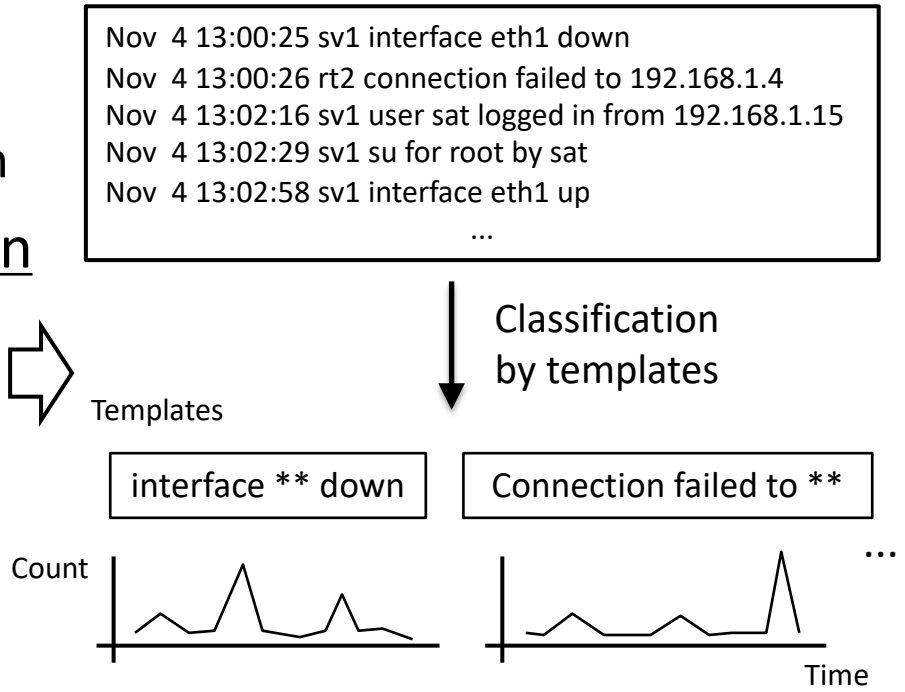
amulog: A General Log Analysis Framework for Diverse Template Generation Methods

Satoru Kobayashi[1], Yuya Yamashiro[2],
Kazuki Otomo[2], Kensuke Fukuda[1]
1: National Institute of Informatics, 2: The University of Tokyo

CNSM 2020, Poster A: Data Mining / Management
Nov 4, 2020

Background

- Automated system log analysis
  - Helpful in daily network operation
- Requires log template generation
  - To classify log messages for time-series analysis
  - To apply natural language processing approaches [1]



[1] W. Meng, et al. “Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs,” in IJCAI, 2019, pp. 4739–4745.

Log template generation methods

- More than 50 different methods [2]
- Diverse assumptions
  - Some methods classify logs, others do not
  - The methods use different segmentation rules
  - ➢ Difficult to compare or combine multiple methods
- We need general framework to use these methods uniformly

[2] M. Landauer, et al. “System log clustering approaches for cyber security applications: A survey”, *Computers and Security*, *92*(101739), 1–17, 2020.

Goal

- Design and Implement a general framework for diverse log template generation methods
  - For easier evaluation
    - Comparing log template generation methods in the same manner
  - For flexible and practical use
    - Combining multiple log template generation methods
    - Importing / Exporting templates

Requirements for general framework

A) Preprocessing logs uniformly

– Preprocessing should depend on data (NOT template methods)

➢ For constant comparison of methods

B) Matching log templates and their instances

– Messages with known templates should be processed fast

➢ For flexible template use

C) Storing parsed data into database

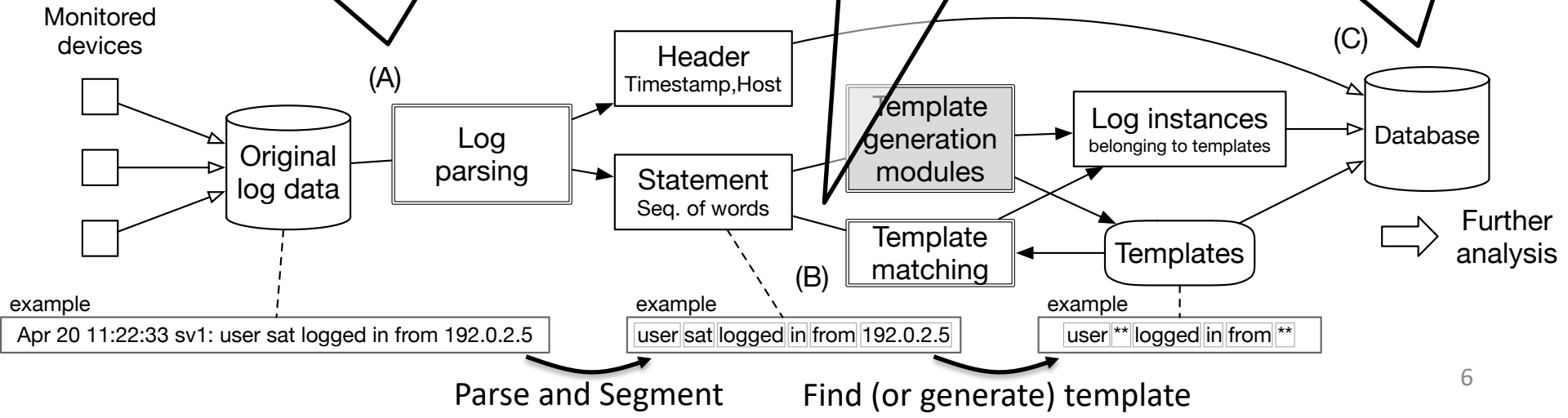
➢ For further analysis

amulog's design

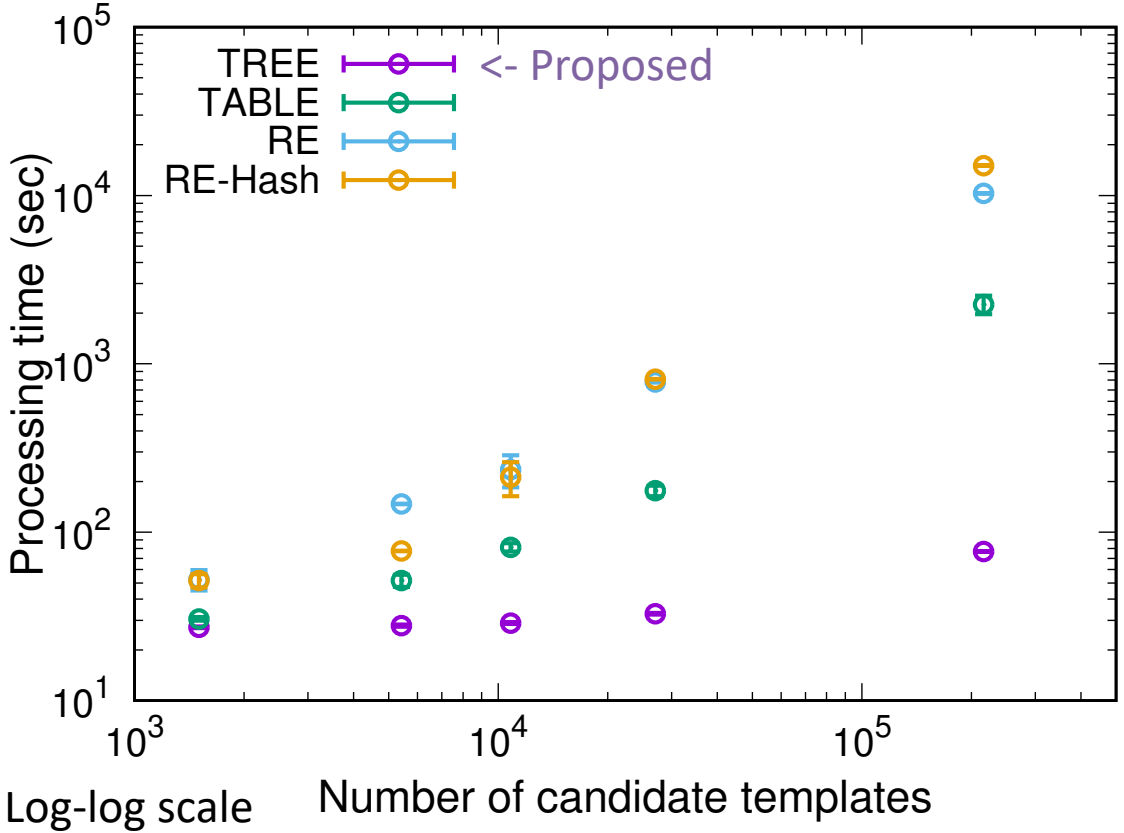
(A) Uniform preprocessing
-> Rule-based customizable parser to segment message (log2seq)

(B) Template matching
-> Tree-based fast search method (not estimation)

(C) Database store
-> Available in both online and offline use



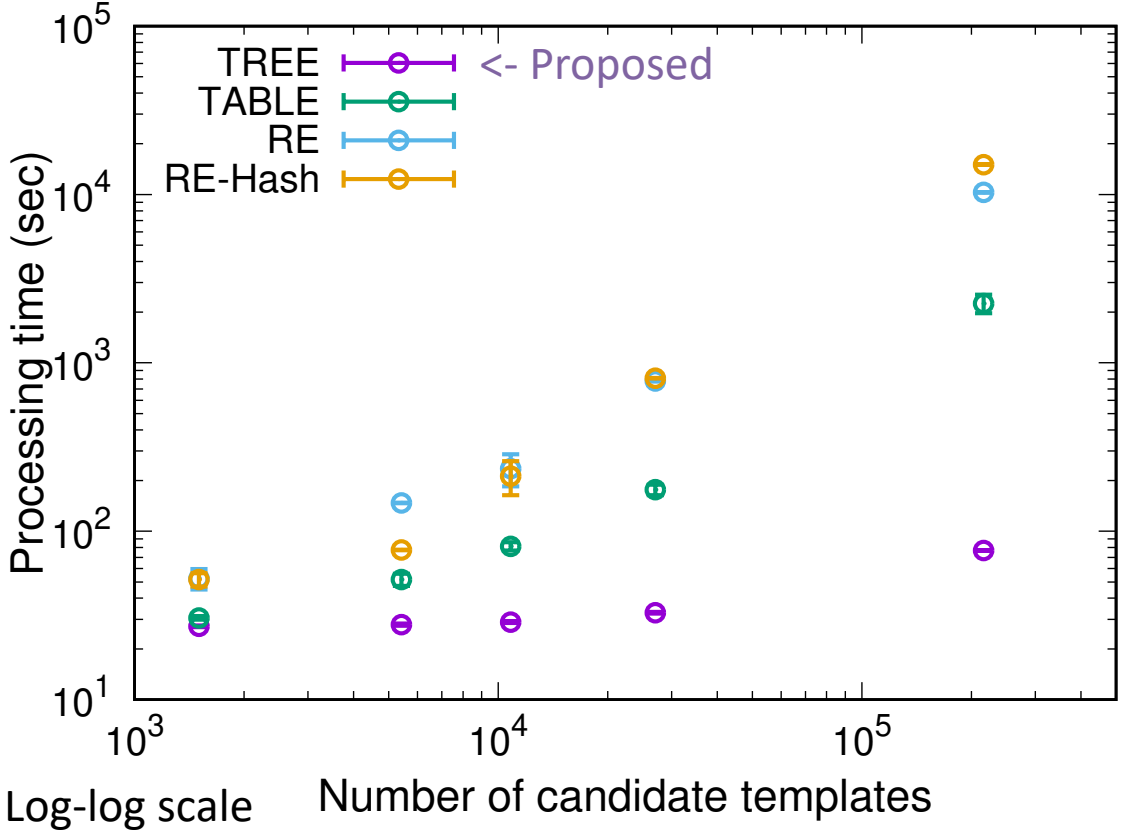
Evaluation of template matching algorithm



- Compare processing time to classify 1-day log messages (76,719)
  - Using SINET4 [3] log messages
  - Give log templates generated by 5 different methods

[3] S. Urushidani, et al. “Highly available network design and resource management of sinet4,” Telecomm. Systems, vol. 56, pp. 33–47, 2014.

Evaluation of template matching algorithm



Findings

- **TREE** and **TABLE** (both using segmentation) are faster than others
  - ➢ Segmentation makes template matching fast

- **TREE** is fast even with over 10^5 given log templates
  - ➢ amulog is scalable

Conclusion

- amulog: A general log analysis framework for diverse log template generation methods
  - Combination or comparison of methods in constant manner
  - Flexible and practical use with template matching
- amulog is fast and scalable in template matching
- https://github.com/cpflat/amulog

This work is supported by JSPS KAKENHI JP19K20262 and MIC/SCOPE #191603009.