Improvement of Log Pattern Extracting Algorithm Using Text Similarity

ZHAO Yining

Computer Network Information Center,
Chinese Academy of Sciences

in HPBDC18, 2018/05/21

Content

- CNGrid & LARGE
- Why Log Patterns & Extracting Algorithm
- Algorithm of Identical Word Rate
- Text Similarity Based Approach
  - Improved Extracting Formation & LCS
  - Experiment Result
- Modified Log Comparing Model
- Summary & Future Work

CNGrid & LARGE

❖ China National HPC Environment

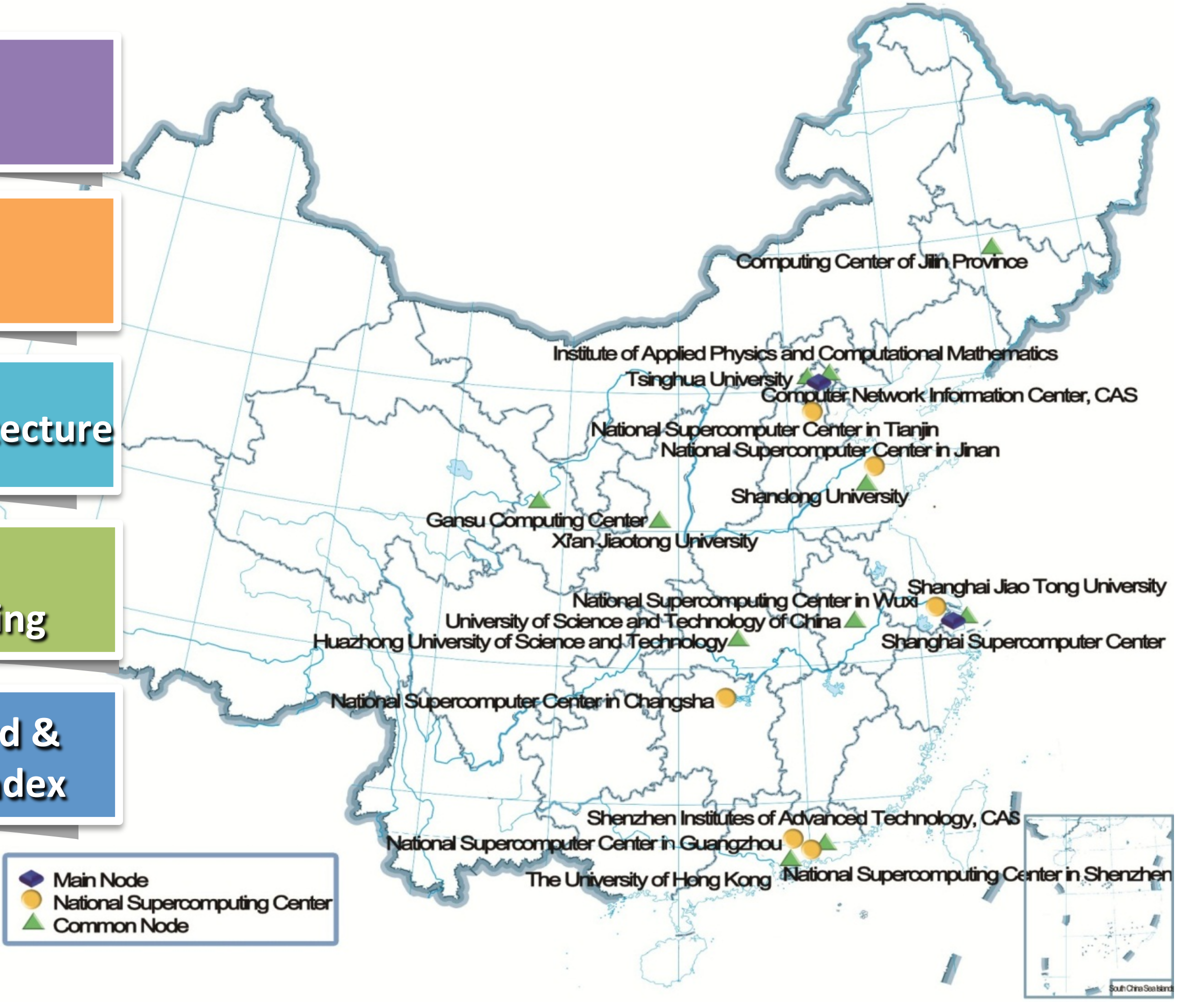
2 Operating Centers
(Beijing / Hefei)

19 Sites
(200PF + 162PB)

Portal with Micro-Service Architecture

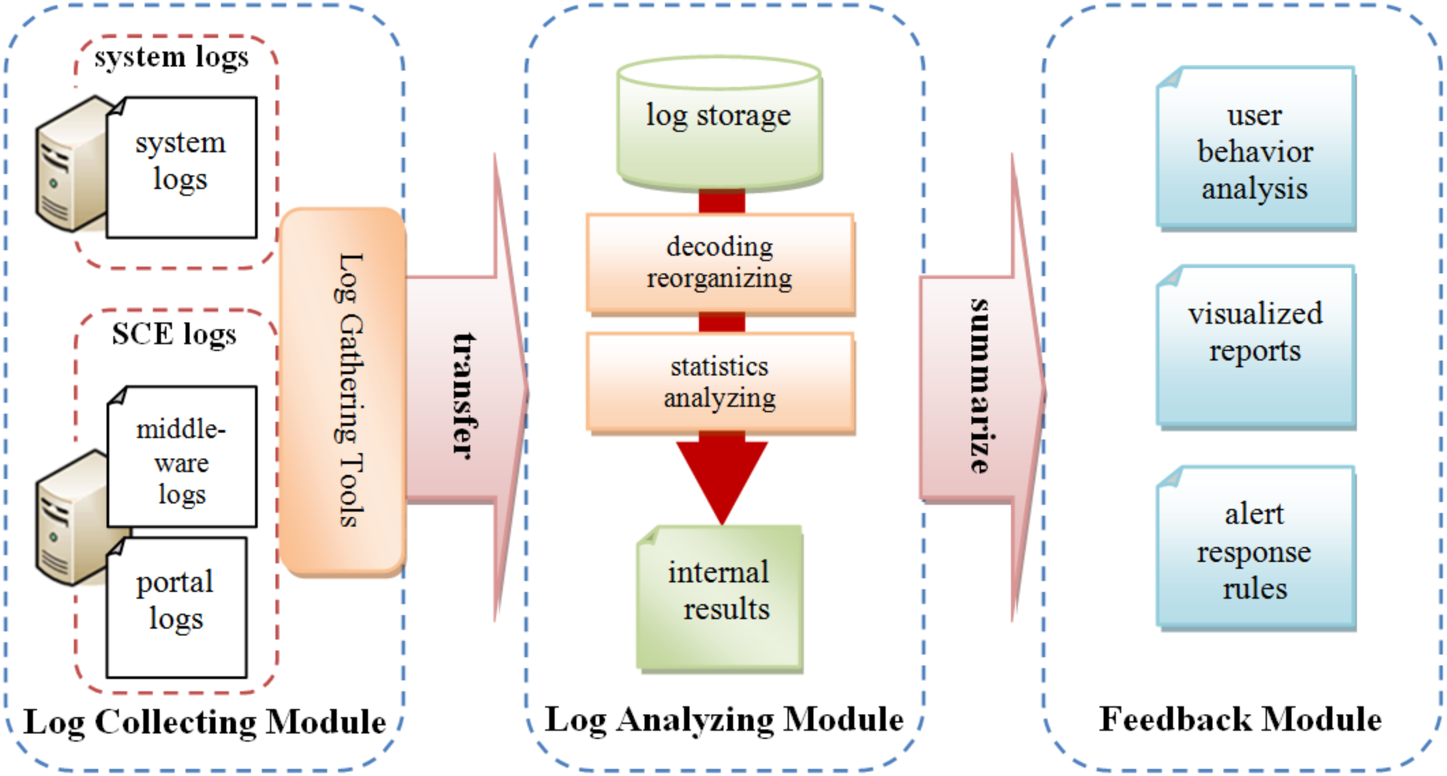
Application oriented
Global Scheduling & Predicting

Resource Evaluation Standard &
Comprehensive Evaluation Index


Computing Center of Jilin Province
Institute of Applied Physics and Computational Mathematics
Tsinghua University
Computer Network Information Center, CAS
National Supercomputer Center in Tianjin
National Supercomputer Center in Jinan
Shandong University
Gansu Computing Center
Xi'an Jiaotong University
Shanghai Jiao Tong University
National Supercomputing Center in Wuxi
University of Science and Technology of China
Huazhong University of Science and Technology
Shanghai Supercomputer Center
National Supercomputer Center in Changsha
Shenzhen Institutes of Advanced Technology, CAS
National Supercomputer Center in Guangzhou
The University of Hong Kong
National Supercomputing Center in Shenzhen
South China Sea Islands
Main Node
National Supercomputing Center
Common Node

CNGrid & LARGE

- Log Analyzing fRamework in Grid Environment



Log Patterns & Extracting Algorithm

- We want to be alerted for logs in certain patterns, but…
  - too many logs for human to read
  - need to summarize patterns before defining alert rules
- Set of log patterns in our context:
  - patterns are different from each other
  - covering all logs in original set
  - significantly less than original
- The process of using log patterns
  - filter and remove frequent normal logs
  - use log pattern extraction algorithms to get the set of patterns
  - manually check the set and pick out abnormal patterns
  - define rules to generate alerts for these patterns

Algorithm of Identical Word Rate

- Algorithm of identical word rate – a straight forward way
  - identical words
    - 2 words that are identical
    - and in the same position in 2 original logs

$$\text{identical(i,l,l')} = \begin{cases} 1, & \mathrm{w_i} = \mathrm{x_i} \\ 0, & \mathrm{w_i} \neq \mathrm{x_i} \end{cases}$$

  - identical word rate
    - (number of identical words) / (total words)
    - predefined threshold t
    - If IWR is greater than t, the two logs are in one pattern

$$\text{r(l, l')} = \begin{cases} \frac{\sum_{\mathrm{i=1}}^{\mathrm{n}} identical(i,l,l')}{\mathrm{n}}, & n = m \\ 0, & n \neq m \end{cases}$$

- Process of algorithm of IWR
  - set threshold t and initial empty pattern set P
  - for each new incoming logs, compute IWR with each pattern in P
  - if pattern matched, skip to next; if none matched, add to P
- Significant Limitation
  - Logs with different length has IWR of ZERO!

Text Similarity Based Approach (1)

- Using Text Similarity to resolve the problem
  - S = P x O
  - S: similarity, P: propotion of common words, O: order factor

- Two logs l_1 and l_2, L_1 and L_2 are word sets respectively
  - define P: $P(l_1, l_2) = (|L_1 \cap L_2| \times 2) / (|L_1| + |L_2|)$
  - define O: $O(l_1, l_2) = SeqSim(l_1, l_2) / |L_1 \cap L_2|$
  - hence S: $S(l_1, l_2) = (SeqSim(l_1, l_2) \times 2) / (|L_1| + |L_2|)$

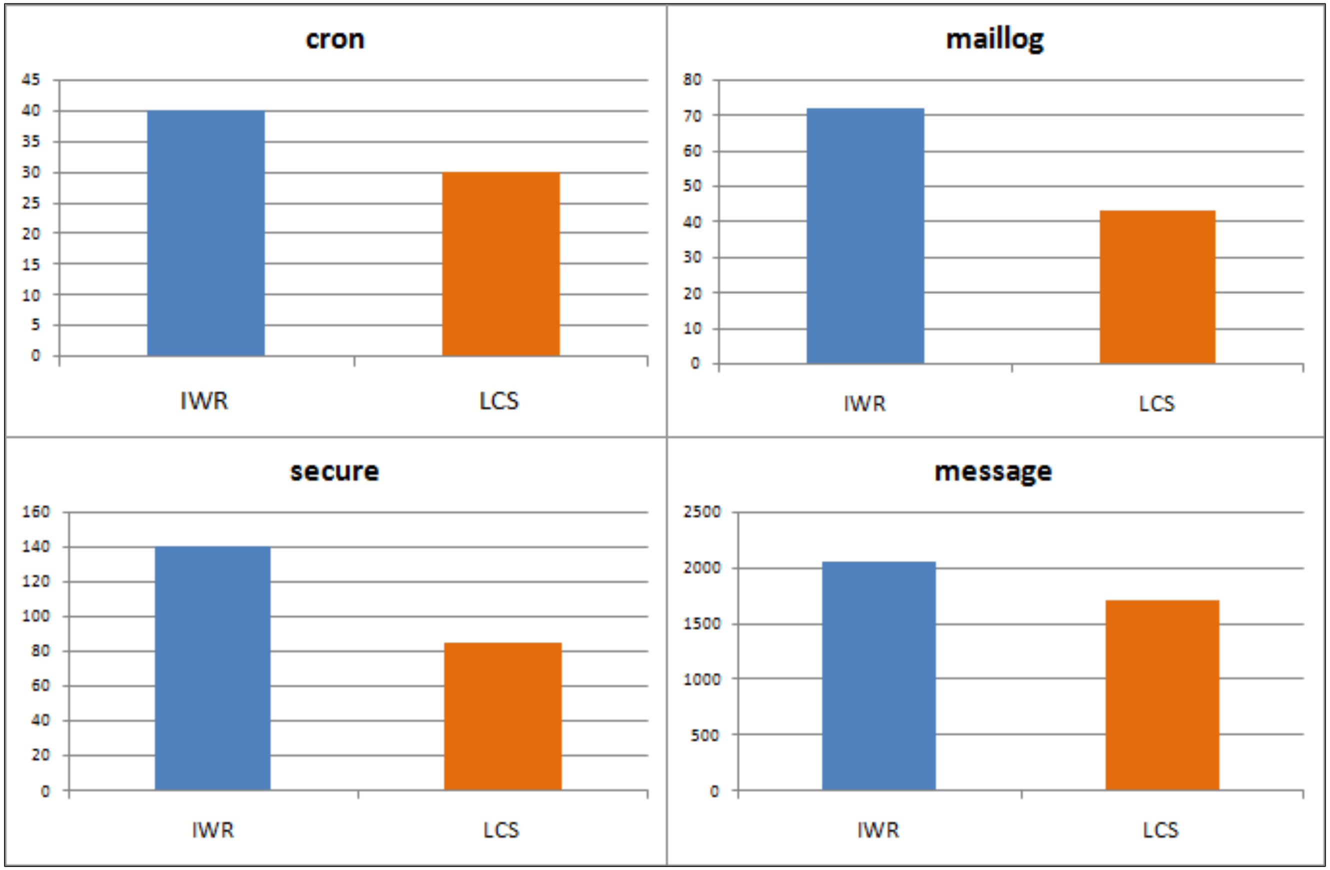
- By this, logs in different lengths can be compared

Text Similarity Based Approach (2)

- Using Longest Common Subsequence to define SeqSim(l_1,l_2)
  - $S(l_1, l_2) = (|LCS(l_1, l_2)| \times 2) / (|L_1| + |L_2|)$
  - Same pattern if $S(l_1, l_2) \geq t$, where t is the predefined threshold
- The process of improved log pattern extracting algorithm
  - set the threshold value t. Set the initial log pattern set P to be an empty set
  - for a new log l appearing from the input log set L, compute $S_i(l, p_i)$ between l and every $p_i \in P$ using a LCS algorithm
  - if there is no $S_i(l, p_i) \geq t$, add l to P
  - after all logs in L have been checked, return P
- Increase time cost for single comparison
  - but reduce total number of comparisons
  - can be offset by choosing a better LCS algorithm

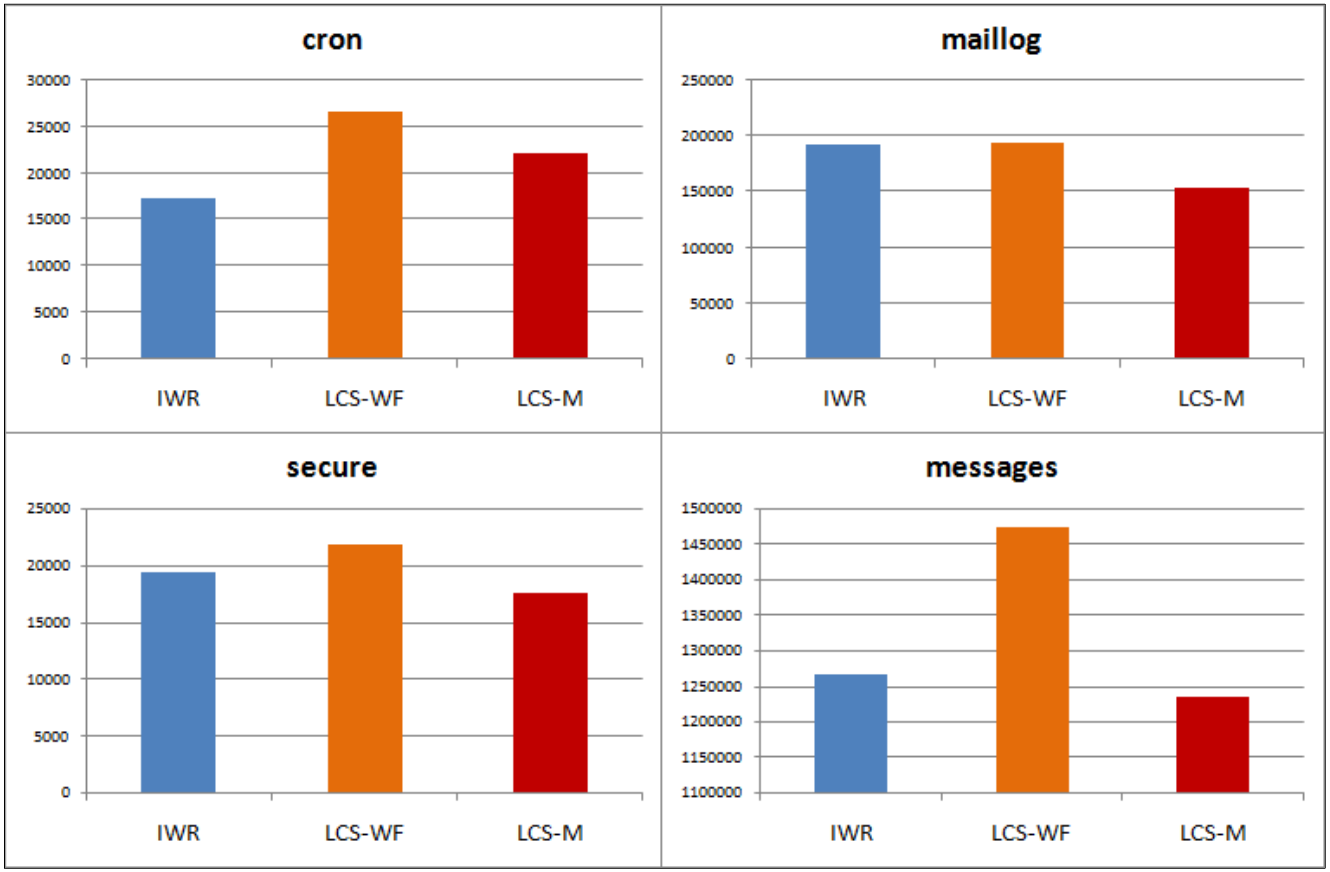
Text Similarity Based Approach (3)

- Experiment result
  - numbers of extracted patterns



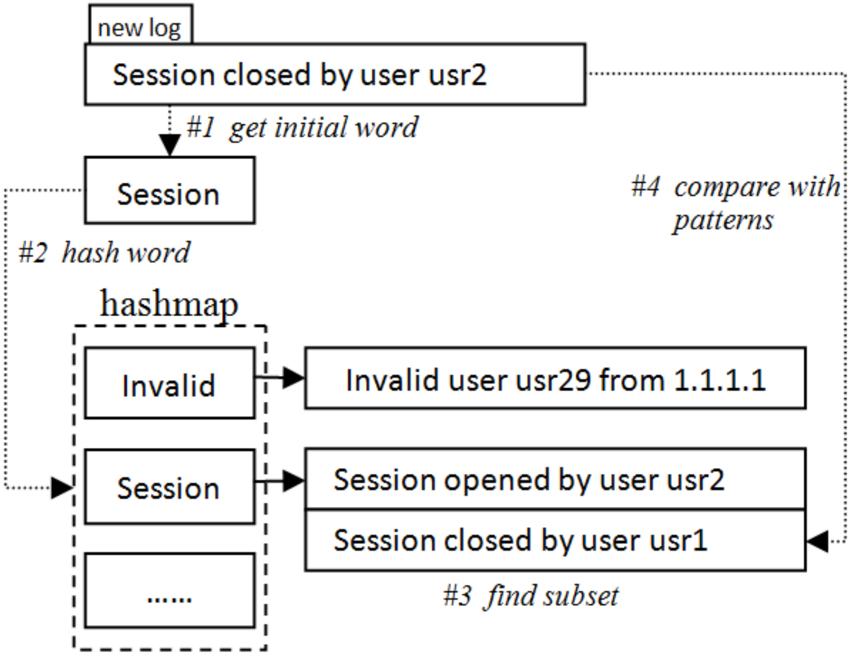
Text Similarity Based Approach (3)

- Experiment result
  - time costs of candidate algorithms (in milliseconds)



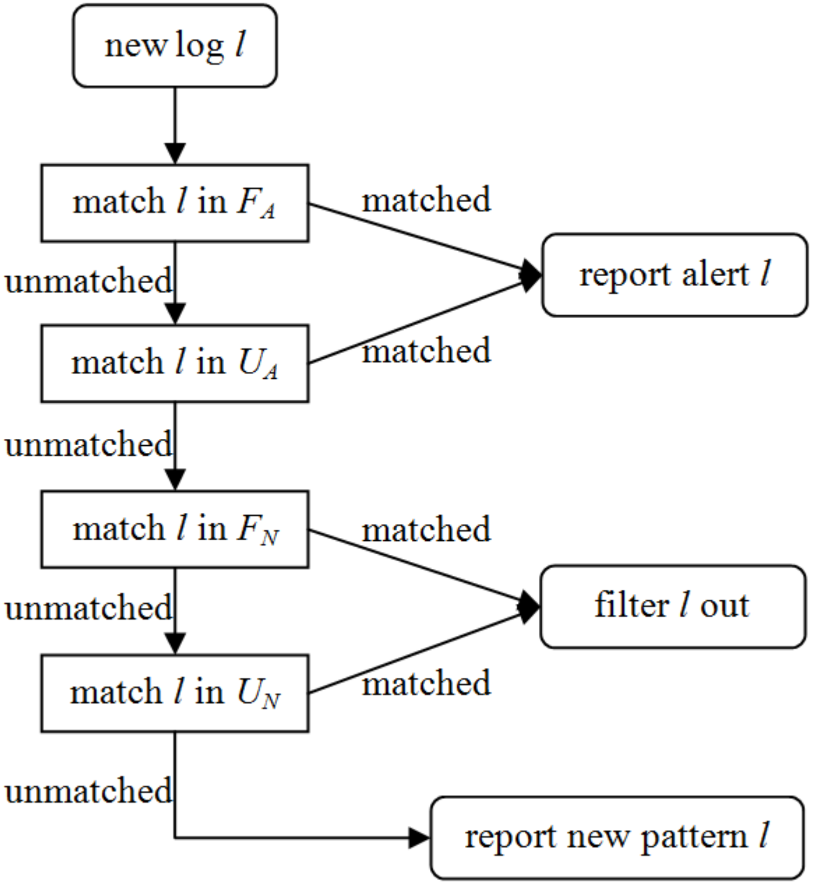
Modified Pattern Comparing Model (1)

- The original model is bad in time cost of searching patterns
  - has to visit all patterns until the one is met
- Use hashmap to accelerate the matching
  - divide pattern set into subsets by initial words
  - skip majority of patterns in irrelevant subsets
- Matching process :
  1. get initial word of the log
  2. hash the word
  3. find desired subset in hashmap
  4. compare with patterns in the subset


new log
Session closed by user usr2
#1 get initial word
Session
#4 compare with patterns
#2 hash word
hashmap
Invalid
Invalid user usr29 from 1.1.1.1
Session
Session opened by user usr2
Session closed by user usr1
......
#3 find subset

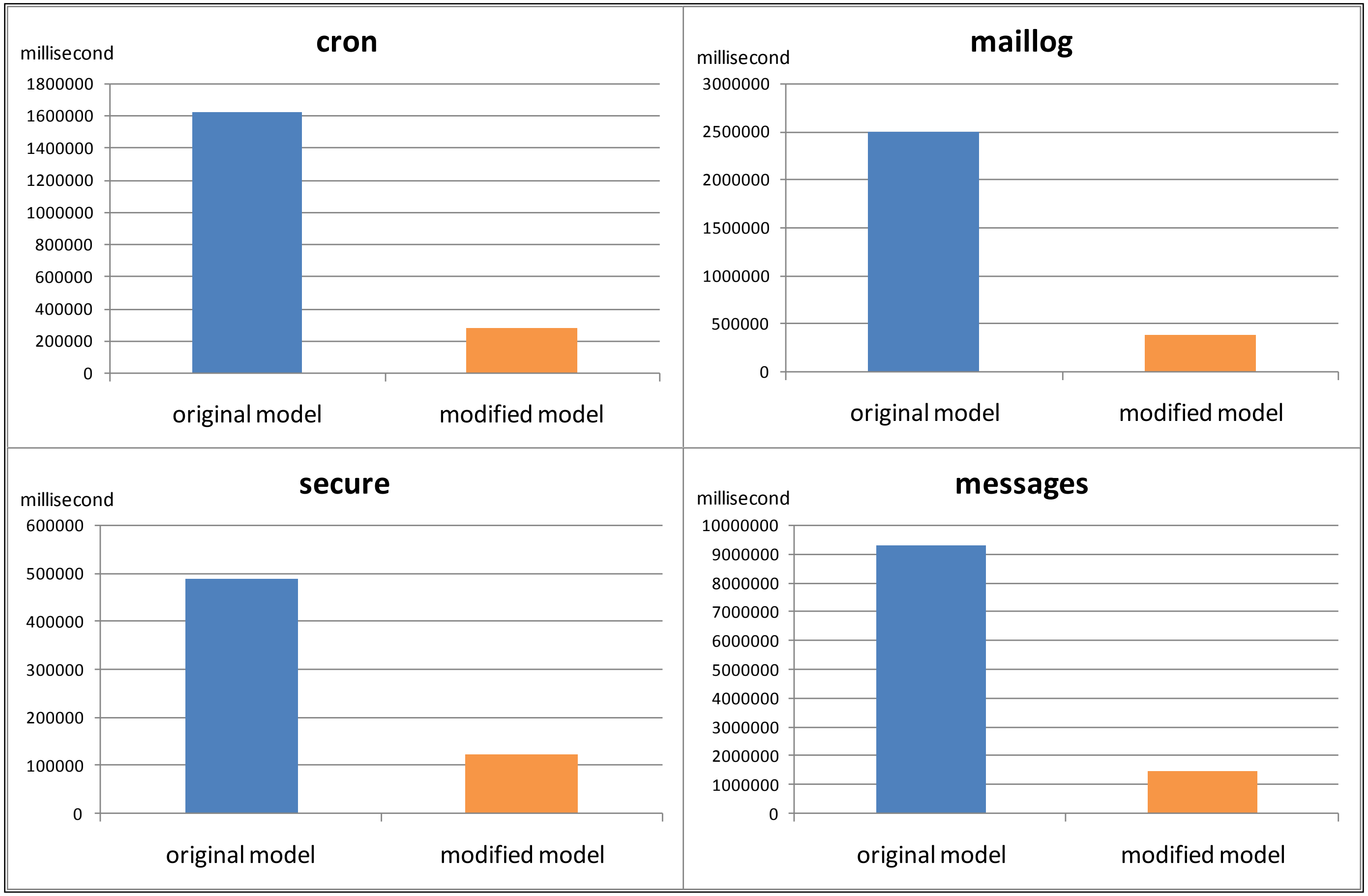
Modified Pattern Comparing Model (2)

- This approach cannot deal with patterns with unfixed initials
  - build an unfixed pattern set
- In real system, we split pattern set in 4 parts:
  - fixed alert pattern set
  - unfixed alert pattern set
  - fixed normal pattern set
  - unfixed normal pattern set
- When a new log comes, it is compared in the 4 sets in turn to decide processing methods



Modified Pattern Comparing Model (3)

- Real time cost comparison between original & modified models



Summary & Future Work

- Log patterns: used to build log recognition
- Algorithm of IWR isn’t capable to match logs in different lengths
- Using the idea of text similarity and LCS to improve the algorithm
- Modify log comparing model to accelerate the process

- Future work: log pattern based analyses in CNGrid
  - log pattern associations
  - log flow feature modeling