Extracting Log Patterns from System Logs in LARGE

Yining Zhao

Computer Network Information Center
Chinese Academy of Sciences

HPBDC, 27 May 2016

Content

- ScGrid in CAS
- The System of LARGE
- Log Patterns – Why?
- 2 Algorithms and Optimizations of Extracting
  - Identical Word Rate
  - Tree-matching
  - Optimizations
- Comparisons on Performances
- Conclusion

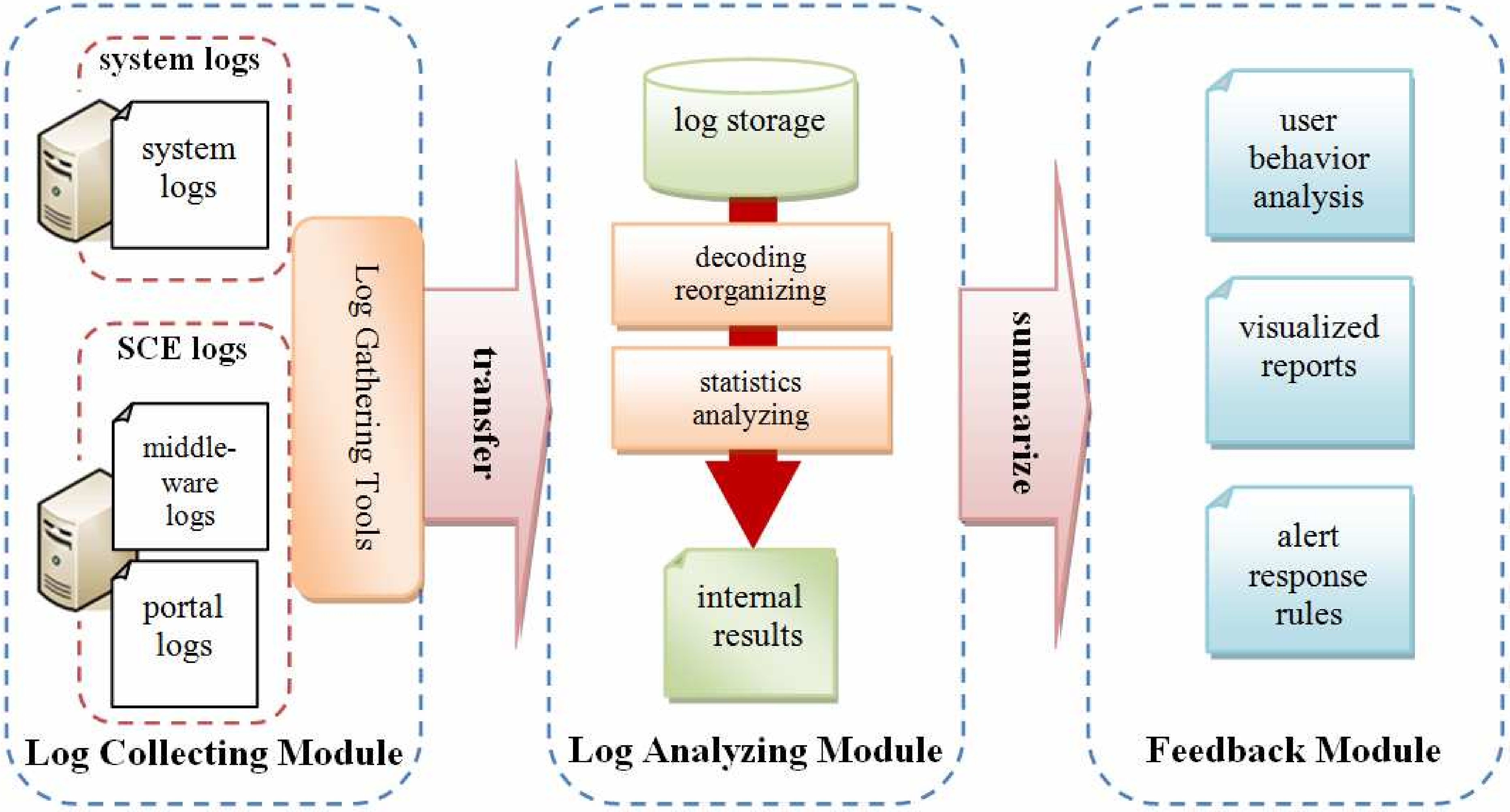
ScGrid in CAS

- Scientific Computing Grid Environment
- Integrated by many supercomputing centers in China
  - CNIC the head center
  - 8 national centers
  - 18 institute centers
- Using SCE middleware developed by SCCAS
- Provide computing resources
  to researches in various fields
  - Meteorology, Metal Forging,
    Fluid Mechanics, High Energy Physics,
    Computational Chemistry, Astrology…

The System of LARGE

- Log Analyzing fRamework in Grid Environments
- Processing logs produced by the environment
  - gathering logs
  - processing, doing statistics and analyzing
  - producing feedbacks
- Two major types of logs
  - system logs – by log service in operating systems
  - SCE logs – by SCE middleware and job scheduling processes
- Helping the environment run correctly and steadily
  - generate alerts for particular patterns of logs
  - provide data for system analysis and maintenance
    - ➢ user behavior report
    - ➢ system errors and faults

The System of LARGE



Log Patterns – Why?

- 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 algorithms of extracting log patterns 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 IWR

- 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
  - 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
- Logs with different length has IWR of ZERO!

```
It is a good day
It isn’t a bad day
```

→

t = 0.66

IWR = 60%

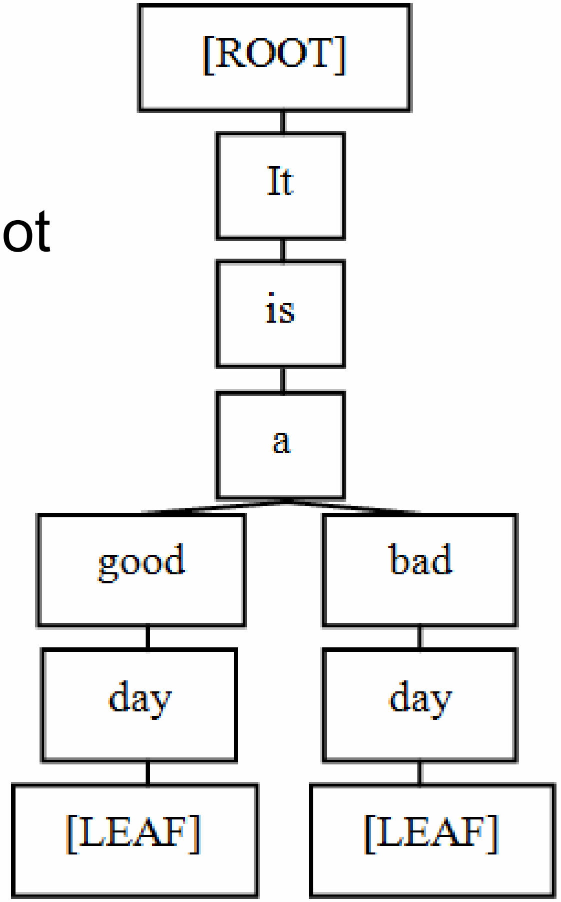
NOT in the same pattern!

Algorithm of IWR

- 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*
- *P* will be affected by order of input
  - 3 logs, *L1:{a, b, c} L2:{a, b, d} L3:{d, b, c}*, *t* = 0.6
  - in order of *l1, l2, L3*, only 1 pattern left
  - in order of *l2, l1, l3*, 2 patterns left
- Not ideal in complexity
  - $O(n^2)$

Algorithm of Tree-matching

- Different in storing and matching structure with IWR
- Words stored orderly in a tree
  - branches for different words
- When performing the algorithm
  - compare each word in the incoming log from root
  - if successfully matched to leaf, check next
  - if unmatched found, create a new branch
- To get pattern set, use depth-first traverse
- Better complexity, but worse result
  - O(n *log*(n))
  - unacceptable number of patterns
  - need optimizations

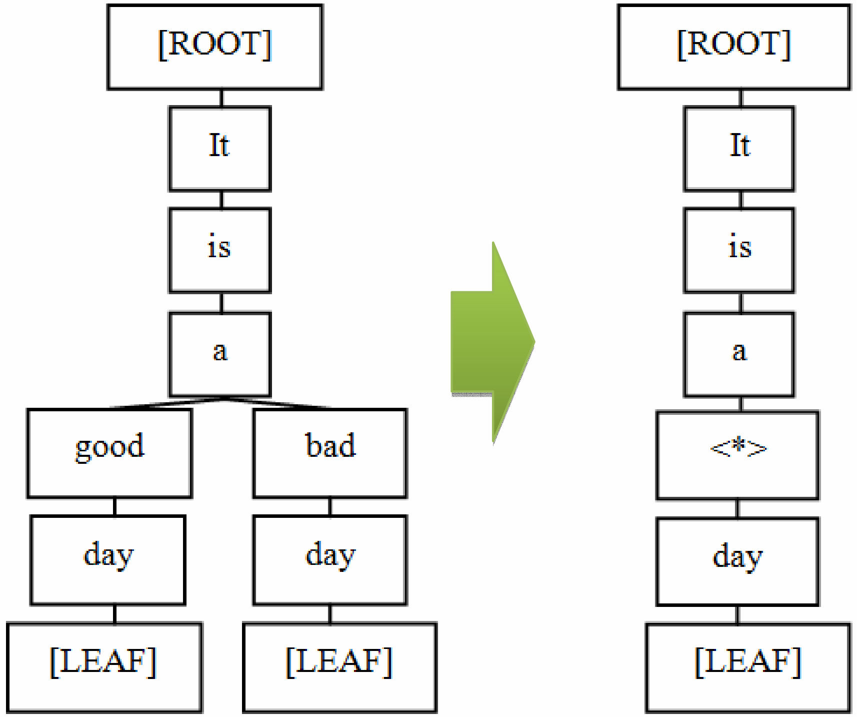


Algorithm Optimizations

- Tree pruning
  - if two nodes has same subtrees, merge to a key node (<*>)
  - after previous step, merge all key nodes to keep uniqueness
  - key node can be matched to any word, but only the last option
  - we can do this because…

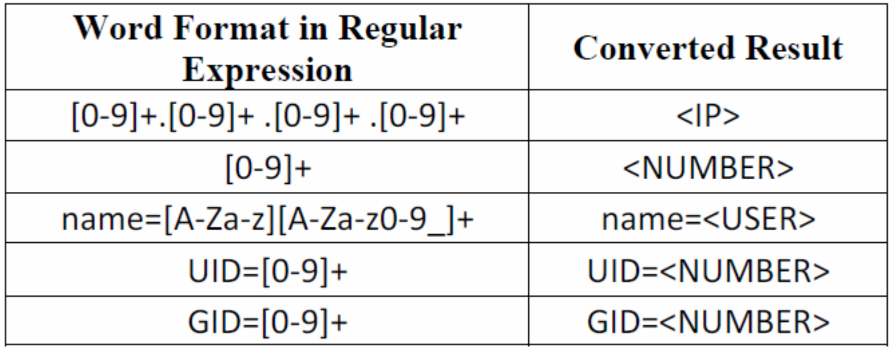
```
Login failed from user alice: password failed
Login failed from user bob: password failed
Login failed from user chris: password failed
```

key information position



Algorithm Optimizations

- Word converting function
  - a preprocessing optimization
  - converting commonly appeared expressions to predefined strings
- We can do this because…
  - At this stage, IPs, usernames, etc. are not essential
  - could be a distraction for extracting and enlarge pattern set
- Helpful for tree-matching, but not IWR
  - IWR has higher tolerance for differences of words
  - Tree-matching is more sensitive

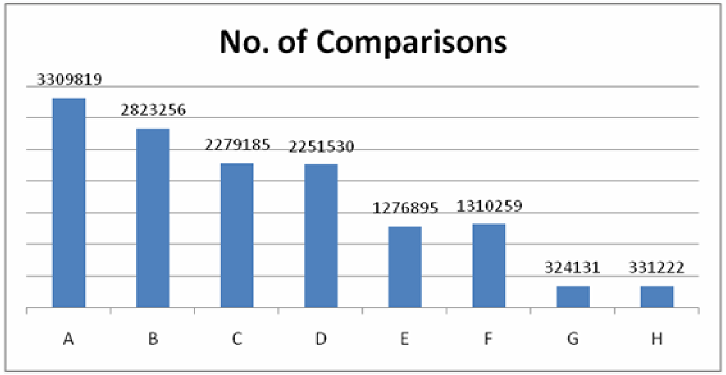


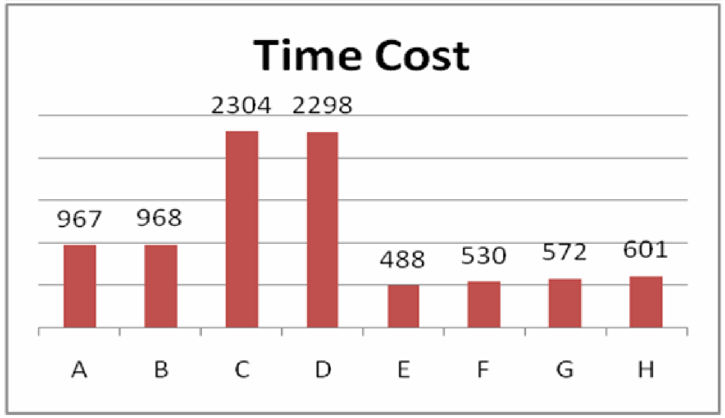
Word Format in Regular Expression	Converted Result
[0-9]+.[0-9]+ .[0-9]+ .[0-9]+	<IP>
[0-9]+	<NUMBER>
name=[A-Za-z][A-Za-z0-9_]+	name=<USER>
UID=[0-9]+	UID=<NUMBER>
GID=[0-9]+	GID=<NUMBER>

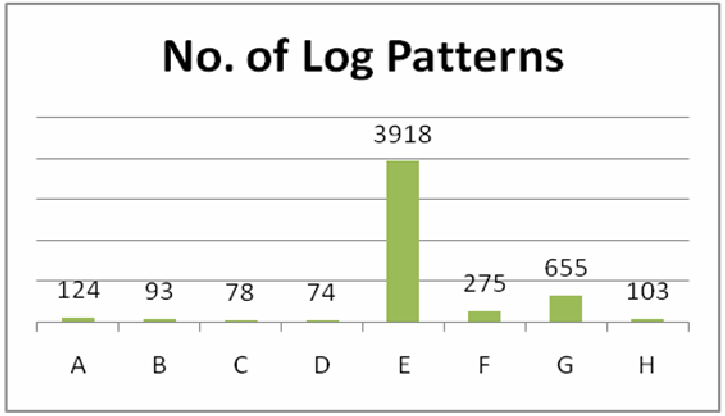
Comparisons on Performances

original: 49079 input logs

A. IWR, t = 0.66
B. IWR, t = 0.6
C. IWR + word converting, t = 0.66
D. IWR + word converting, t = 0.6
E. Tree-matching
F. Tree-matching + pruning
G. Tree-maching + word converting
H. Tree-matching + both







Comparisons on Performances

Tree-matching has significantly reduced the number of comparisons

o

A. IWR, t = 0.66

B. IWR, t = 0.6

C. IWR + word converting, t = 0.66

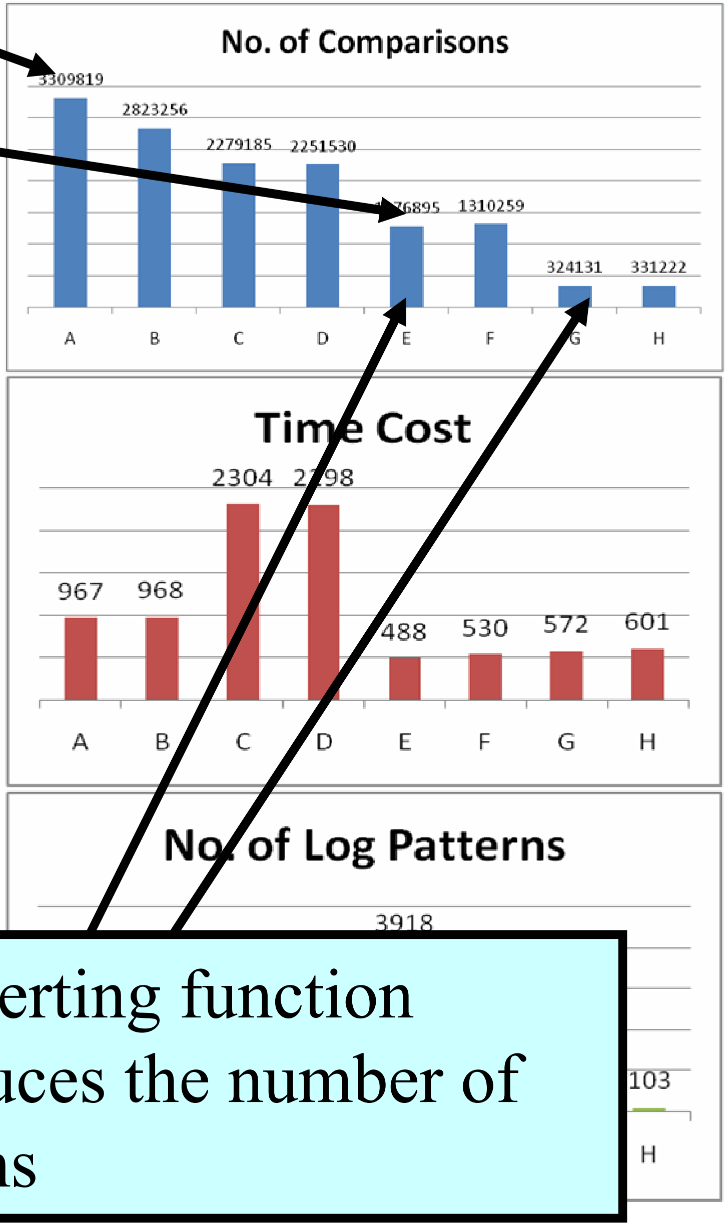
D. IWR + word converting, t = 0.6

E. Tree-matching

F. Tree-matching + pruning

G. Tree-maching + word converting

H. Tree-matching + both



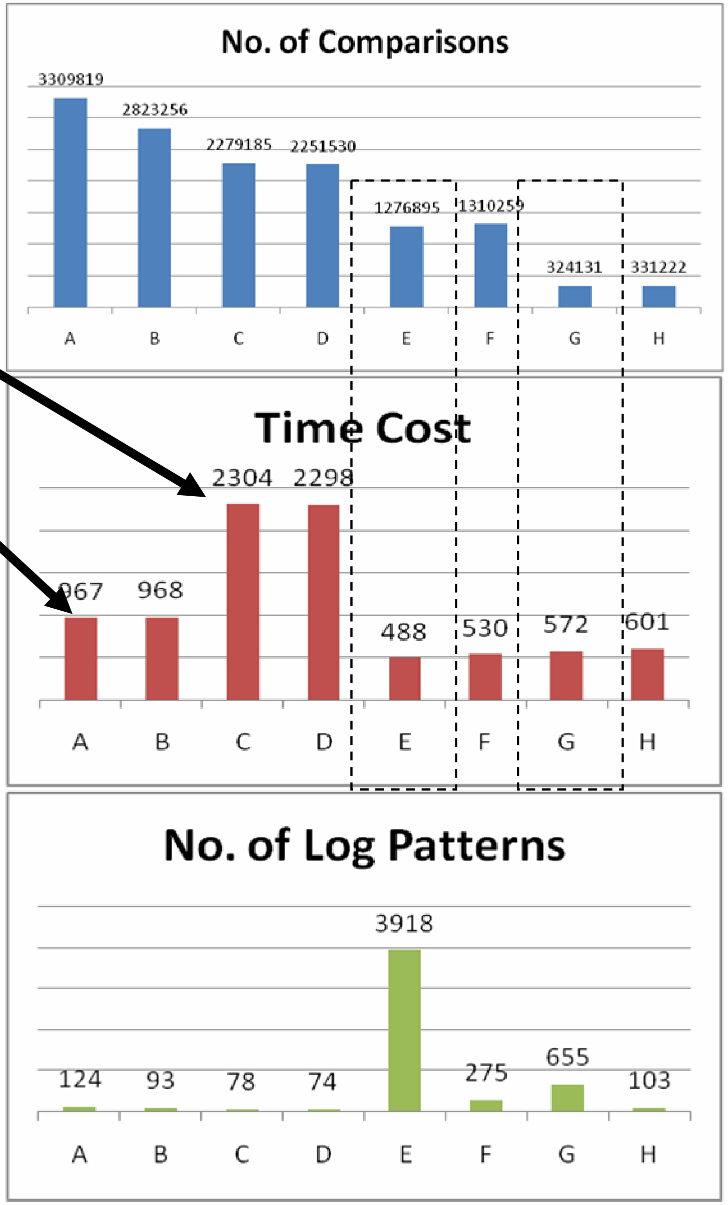
Word converting function further reduces the number of comparisons

Comparisons on Performances

origi

Word converting function is quite time costly

A. IWR, t = 0.66
B. IWR, t = 0.6
C. IWR + word converting, t = 0.66
D. IWR + word converting, t = 0.6
E. Tree-matching
F. Tree-matching + pruning
G. Tree-maching + word converting
H. Tree-matching + both



Comparisons on Performances

orig

Extracting efficiency (time and comparison costs): tree-matching is better

A.

B. IWR, t = 0.6

C. IWR + word converting, t = 0.66

D. IWR + word converting, t = 0.6

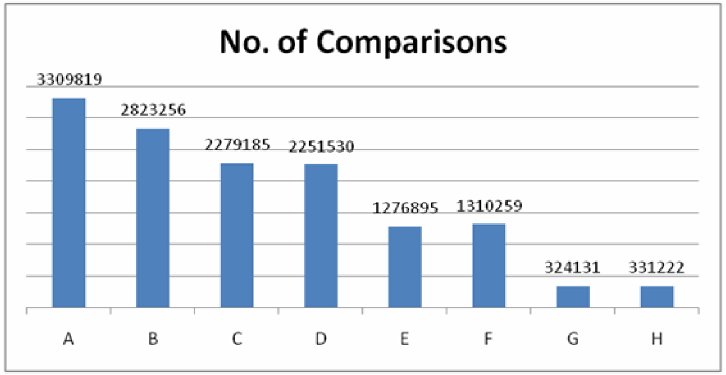
E. Tree-matching

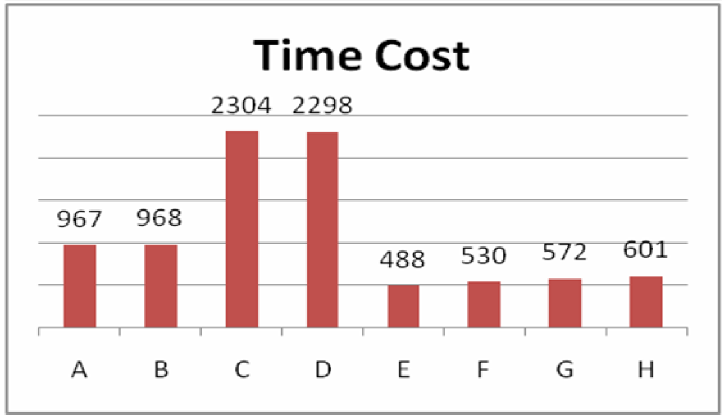
F. Tree-matching + pruning

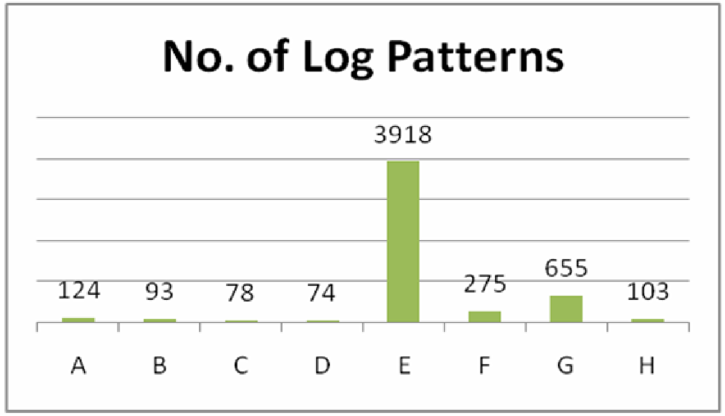
G. Tree-maching + word converting

H.

Extracting effect (number of patterns): IWR is better





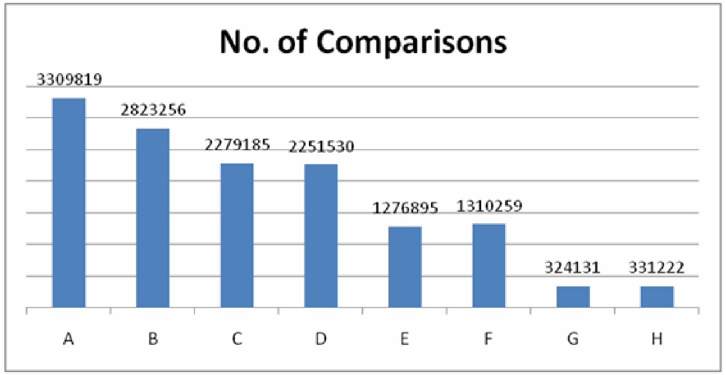


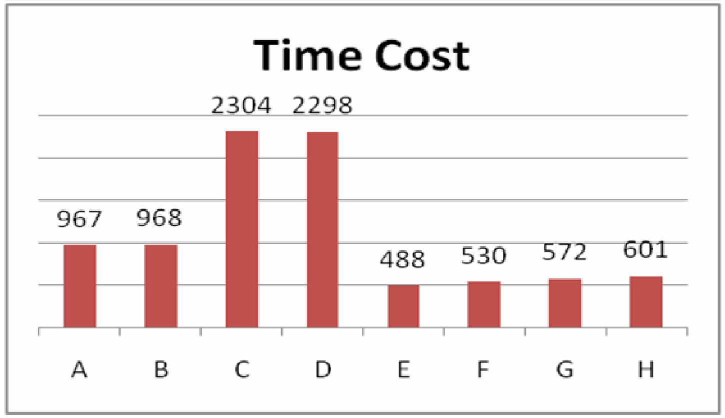
Comparisons on Performances

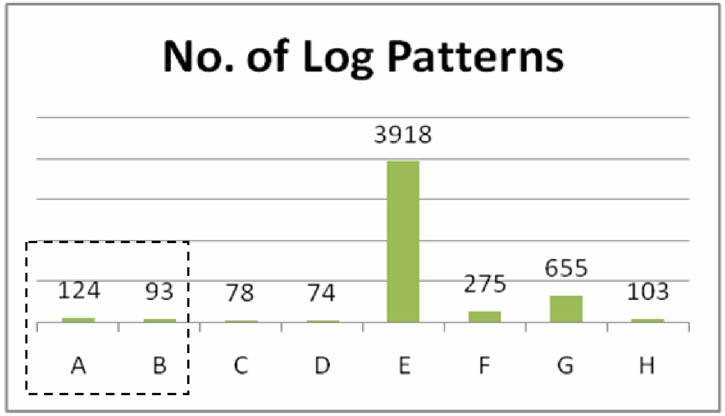
o

Lower threshold gives lesser patterns in IWR

A. IWR, t = 0.66
B. IWR, t = 0.6
C. IWR + word converting, t = 0.66
D. IWR + word converting, t = 0.6
E. Tree-matching
F. Tree-matching + pruning
G. Tree-maching + word converting
H. Tree-matching + both





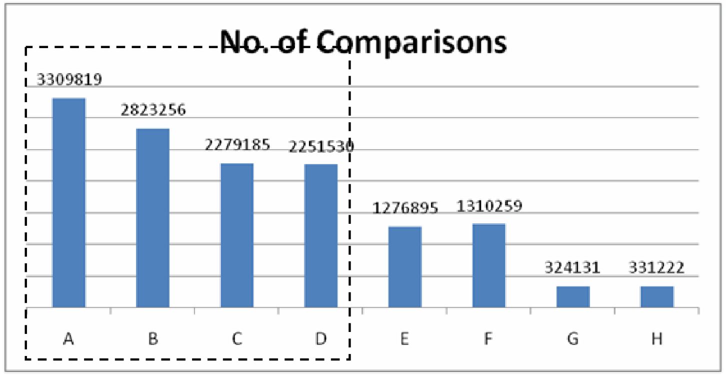


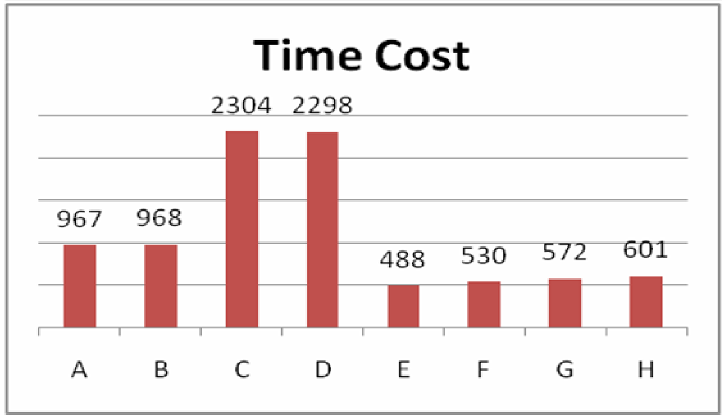
Comparisons on Performances

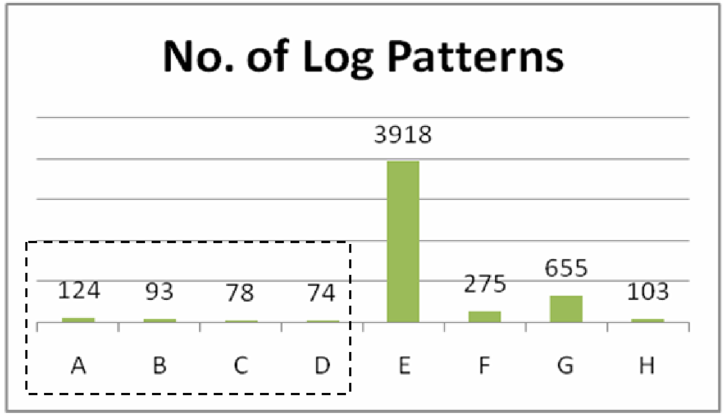
Number of comparisons has nearly direct proportion to the number of extracted patterns

o

A. IWR, t = 0.66
B. IWR, t = 0.6
C. IWR + word converting, t = 0.66
D. IWR + word converting, t = 0.6
E. Tree-matching
F. Tree-matching + pruning
G. Tree-maching + word converting
H. Tree-matching + both







Comparisons on Performances

Pure tree-matching:

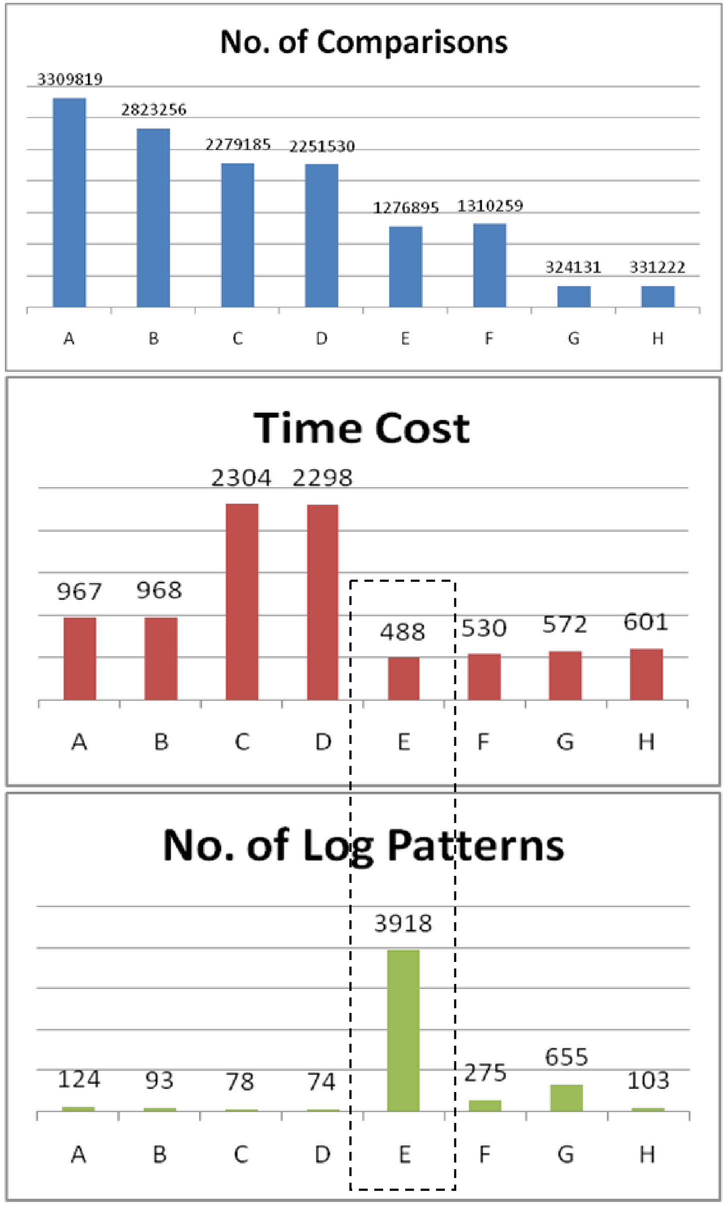
good in extract efficiency

very bad in extracting effect

o

A

B. IWR, t = 0.6
C. IWR + word converting, t = 0.66
D. IWR + word converting, t = 0.6
E. Tree-matching
F. Tree-matching + pruning
G. Tree-maching + word converting
H. Tree-matching + both



Comparisons on Performances

- Pruning has greatly improved the extracting effect

A. IWR, t = 0.66

B. IWR, t = 0.6

C. IWR + word converting, t = 0.66

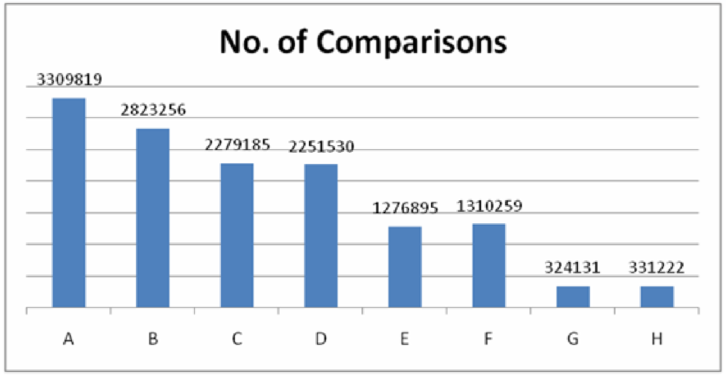
D. IWR + word converting, t = 0.6

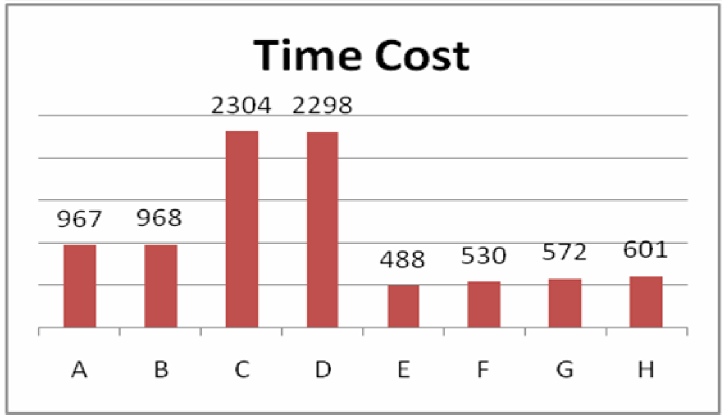
E. Tree-matching

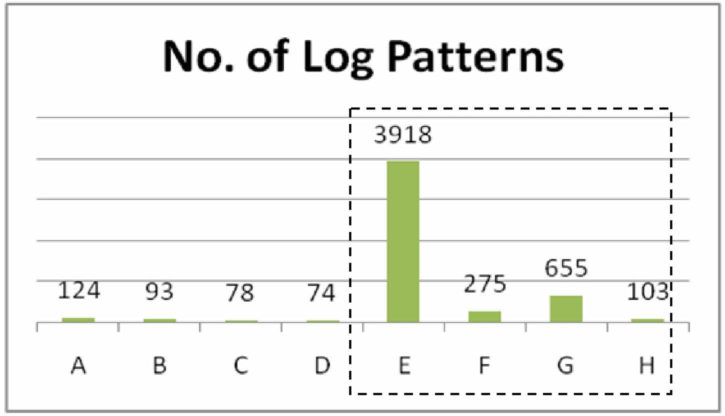
F. Tree-matching + pruning

G. Tree-maching + word converting

H. Tree-matching + both







Comparisons on Performances

original: 49079 input logs

Tree-matching with both optimizations: good in extracting efficiency and effect, satisfying algorithm!

A. I

B. I

C. I

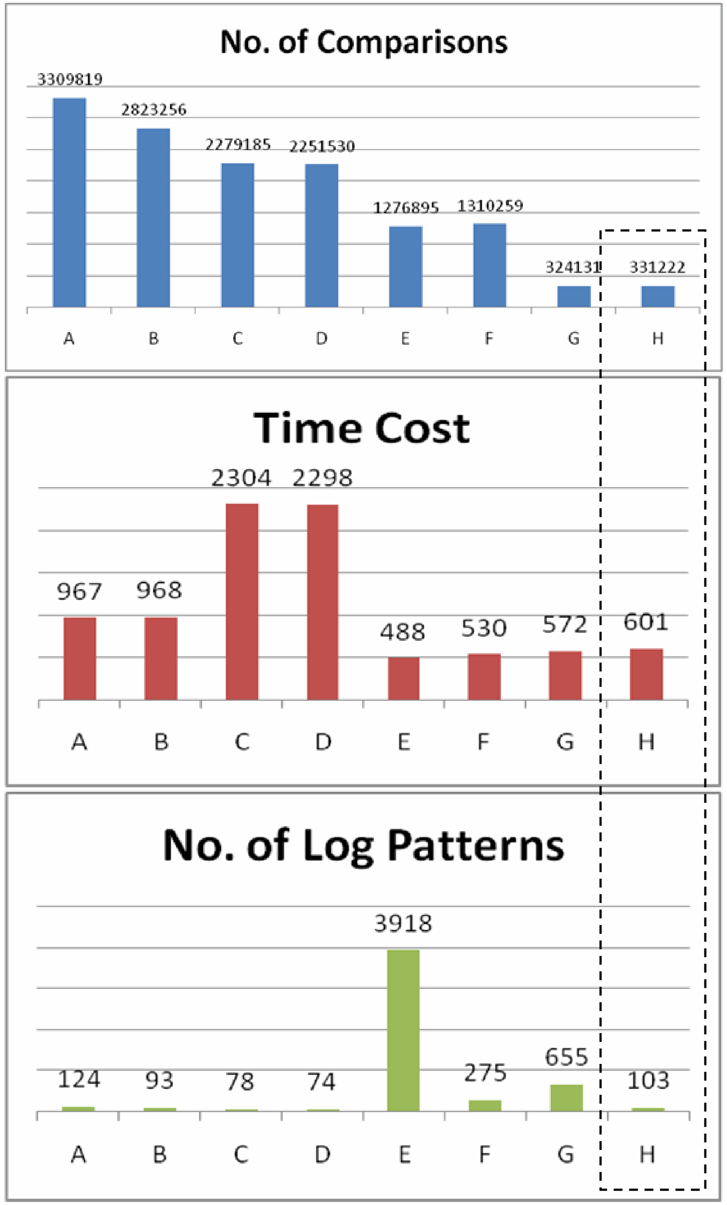
D. IWR + word converting, t = 0.6

E. Tree-matching

F. Tree-matching + pruning

G. Tree-maching + word converting

H. Tree-matching + both



Conclusion

- LARGE is a log analyzing system
- When monitoring system logs, we need to extract log patterns
- Two algorithms: IWR and tree-matching, plus optimizations
  - tree-matching with two optimizations looks good

- Future work
  - what if more than one key position in tree-matching algorithm?
  - we may use parallel computing to accelerate log processing

Thank you!

微重力效应模拟生物力学平台

微重力效应模拟生物力学平台是应用计算流体力学与生命科 的虚拟平台。采用模型化方法对各种生物培养器的微重力效应模 重力效应模拟运控模式，为空间生命科学研究和空间生物技术发 导地面实验设计。

Tasks are listed. Press "More" to add new and "Delete" to remove. Press "Select task" to save and modify the task you want to change.

IDName	DicRef	Description		
0 Geometry Optimizatic	matgrid.default	Default task	More	Delete
2 Density of State	matgrid.dos	density of state calculation	More	Delete
3 Band Structure	matgrid.bandStru	Band Structure	More	Delete
4 Band Gap	matgrid.bandgap	Band Gap	More	Delete