Fast Log Analysis Made Easy by Automatically Parsing Heterogeneous Logs

Biplob Debnath and Will Dennis

NEC Laboratories America, Inc.

{biplob,wdennis}@nec-labs.com

LISA17

October 29–November 3, 2017 | San Francisco, CA
www.usenix.org/lisa17 #lisa17

Log Analysis

Internet of Things (IoT)



Software Systems



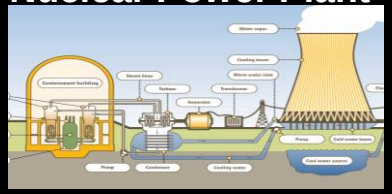
Computer Systems



Smart Cities



Nuclear Power Plant



Auto Production Line



- Log ubiquitously exists in many complex systems.
- Most of the logs include massive amount of transaction or status data.
- Tons of logs are generated, but difficult to manually investigate.

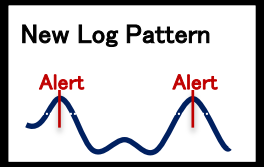
Log Analysis: Example

Raw Log Streams

L1
L2
L3
L4
.
.
.
Ln

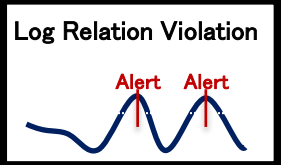
Log Analyzer

Analyzes logs using various features and reports alerts









Sample Alert

{"message":"\" uag412cjme107 \" , \" ZU4JNWG412PUC025 . 2013 / 09 / 24 15 : 00 : 17 \" , \" JNWSTART \" , \" - \" , \" 2013 / 09 / 25 00 : 00 : 17 . 000 (東京 (標準時)) \" , \" - \" ","@version":"1","@timestamp":"2015-06-02T21:21:19.360Z","type":"No.08_Learning_file.log","host":"am14-06","seq":23}

{"message":"\" uag412cjme107 \" , \" ZU4JNWG412PUC025 . 2013 / 09 / 24 15 : 00 : 17 \" , \" JNWSTART \" , \" - \" , \" 2013 / 09 / 25 00 : 00 : 17 . 000 (東京 (標準時)) \" , \" - \" ","@version":"1","@timestamp":"2015-06-02T21:21:19.364Z","type":"No.08_Learning_file.log","host":"am14-06","seq":24}

"anomaly_reason":"IP","anomaly_score":1.00,{"message":"\" IOROI - 1 \" , \" ZJd582YYx . 2015 / 03 / 31 15 : 01 : 00 \" , \" LOGIN ID = IOROI - 1 \" , \" 10 . 110 . 4 . 218 \" , \" 2015 / 04 / 01 00 : 01 : 00 . 000 (東京 (標準時)) \" , \" - \" ","@version":"1","@timestamp":"2015-04-01T04:01:00.000Z","type":"No.11_Learning_file.log.Jianwu","host":"am14-06","P1F1":"1","P1NS1":"ZJd582YYx","ts1":"2015 / 03 / 31 15 : 01 : 00","P1F2":"1","IP":"10 . 110 . 4 . 218","mts":"2015 / 04 / 01 00 : 01 : 00","P1F3":"000","tags":["pattern","1"],"seq":1}

"anomaly_reason":"In between 2015 / 04 / 01 00 : 10 : 41 and 2015 / 04 / 01 00 : 20 : 41 pattern 15 exceeds previous maximum count of 1. Current count: 6",{"message":"\" NAGAMATSU - 2 \" , \" hkW351ciQ . 2015 / 03 / 31 15 : 13 : 17 \" , \" LOGIN ID = NAGAMATSU - 2 LOGIN FAILED \" , \" 10 . 110 . 4 . 107 \" , \" 2015 / 04 / 01 00 : 13 : 17 . 000 (東京 (標準時)) \" , \" 2015 / 04 / 01 00 : 13 : 17 . 000 (東京 (標準時)) \" ","@version":"1","@timestamp":"2015-04-01T04:13:17.000Z","type":"No.13_Learning_file.log.Jianwu","host":"am14-06","P15W1":"NAGAMATSU","P15F1":"2","P15NS1":"hkW351ciQ","ts1":"2015 / 03 / 31 15 : 13 : 17","P15W2":"NAGAMATSU","P15F2":"2","IP":"10 . 110 . 4 . 107","mts":"2015 / 04 / 01 00 : 13 : 17","P15F3":"000","ts3":"2015 / 04 / 01 00 : 13 : 17","P15F4":"000","tags":["pattern","15"],"seq":55}

"anomaly_reason":"Concurrency Violation for Pattern 1",{"message":"\" KITANO1 \" , \" cpA091acZ . 2015 / 03 / 31 15 : 01 : 42 \" , \" LOGIN ID = KITANO1 \" , \" 10 . 110 . 4 . 109 \" , \" 2015 / 04 / 01 00 : 00 : 48 . 000 (東京 (標準時)) \" , \" \" ","@version":"1","@timestamp":"2015-04-01T04:00:48.000Z","type":"No.12_Learning_file.log.Jianwu","host":"am14-06","P1NS1":"cpA091acZ","ts1":"2015 / 03 / 31 15 : 01 : 42","IP":"10 . 110 . 4 . 109","mts":"2015 / 04 / 01 00 : 00 : 48","P1F1":"000","tags":["pattern","1"],"seq":2}

Log Parsing is the Core Step to any type of Log Analysis

Log Parsing: Example

Mar 3 16:30:04 envctl APC_PDU_LEGAMPS: [INFO] PDU=pdu2z04-am-rack4f Leg=3 Amps=2.5



Parsing Pattern ???

DATETIME envctl APC_PDU_LEGAMPS: [INFO] PDU=NOTSPACELeg=NUMBER Amps=NUMBER

```
{
  “message”: “Mar 3 16:30:04 envctl APC_PDU_LEGAMPS: [INFO]  PDU=pdu-2z04-am-rack4f Leg=3 Amps=2.5”,
  "timestamp" : 2017-03-03T21:30:04.000Z,
   "PDU" : "pdu-2z04-am-rack4f",
   "Leg" : "3",
   "Amps" : "2.5“
}
```

Heterogeneous Log Formats

Oct 23 13:53:39 am12-09 kernel: [448260.543317] sd 6:0:0:0: [sdc] tag#0 FAILED Result: hostbyte=DID_ERROR driverbyte=DRIVER_SENSE
Oct 23 13:53:39 am12-09 kernel: [448260.543324] sd 6:0:0:0: [sdc] tag#0 Sense Key : Hardware Error [current] [descriptor]
Oct 23 13:53:39 am12-09 kernel: [448260.543335] sd 6:0:0:0: [sdc] tag#0 Add. Sense: No additional sense information
Oct 23 13:54:09 am12-09 ntpd[1603]: Soliciting pool server 2001:67c:1560:8003::c8

Syslog

2017-10-04T12:39:44.269-0400 I NETWORK [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
2017-10-04T12:39:44.269-0400 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory '/data/db/mongodb/diagnostic.data'
2017-10-04T12:39:44.270-0400 I NETWORK [initandlisten] waiting for connections on port 27017
2017-10-04T16:47:33.264-0400 I INDEX [conn6] build index on: testdb.testcol properties: { v: 1, key: { timestamp: -1 }, name: "timestamp_-1", ns: "testdb.testcol" }

MongoDB

2014-11-12 16:43:33.061 13625 INFO nova.virt.libvirt.driver [-] [instance: 9d9ffa30-b827-4330-8d8a-4bff5a782475] Instance destroyed successfully. 172.16.4.121 /var/log/nova/nova-compute.lo
2014-11-12 16:43:33.681 13625 INFO nova.compute.manager [-] [instance: 9d9ffa30-b827-4330-8d8a-4bff5a782475] During sync_power_state the instance has a pending task.
Skip. 172.16.4.121 /var/log/nova/nova-compute.log
2014-11-12 16:44:02.446 13625 AUDIT nova.compute.resource_tracker [-] Auditing locally available compute resources 172.16.4.121 /var/log/nova/nova-compute.log

OpenStack

(0):TimerRoutine():(w3wp.exe,0x1bd8,74):0[25 21:39:21] ---------- Process Director Timer Routine COMPLETE ---------- 457016, seconds=0.0781404
(0):GetUserBySID():2[25 21:39:21] SQL SELECT TABLE: tblUser WHERE: tblUser.guidSessionID=N'0b8b6ff1-5e87-48e4-9221-e5fe61a92cae'
(0):GetObject():2[25 21:39:21] SQL SELECT TABLE: tblContent WHERE: tblContent.oID='3b676138-0fe6-4dbd-a20b-49fe5b07725d'
(0):GetColumns():2[25 21:39:21] SQL SELECT TABLE: tblKViewColumn WHERE: tblKViewColumn.oKVID='3b676138-0fe6-4dbd-a20b-49fe5b07725d' ORDER BY tblKViewColumn.nColumnNum ASC

Custom Application

Pattern Generation: Challenges

- Various log formats that can change over time
- Extremely huge amount of system logs
- Limited domain knowledge

Goal: Generate patterns with no or minimal human involvement.

Outline

- Pattern Generation Algorithm
- Demo Use-cases

Outline

- Pattern Generation Algorithm
- Demo Use-cases

Pattern Generation: Problem Statement

- Input
  - ❑A set of logs
  - ❑Optional
    - ❑ Tokenization Delimiter
    - ❑ Max Pattern limit
- Output
  - ❑A set of patterns to parse all input logs

Sample Input Logs

1. 2017/02/24 09:01:00 login 127.0.0.1 user=bear12
2. 2017/02/24 09:02:00 DB Connect 127.0.0.1 user=bear12
3. 2017/02/24 09:02:00 DB Disconnect 127.0.0.1 user=bear12
4. 2017/02/24 09:04:00 logout 127.0.0.1 user=bear12

5. 2017/02/24 09:05:00 login 127.0.0.1 user=bear34
6. 2017/02/24 09:06:00 DB Connect 127.0.0.1 user=bear34
7. 2017/02/24 09:07:00 DB Disconnect 127.0.0.1 user=bear34
8. 2017/02/24 09:08:00 logout 127.0.0.1 user=bear34

9. 2017/02/24 09:09:00 login 127.0.0.1 user=bear#1
10. 2017/02/24 09:10:00 DB Connect 127.0.0.1 user=bear#1
11. 2017/02/24 09:11:00 DB Disconnect 127.0.0.1 user=bear#1
12. 2017/02/24 09:12:00 logout 127.0.0.1 user=bear#1

Pattern Count?

Is it possible to give an option to the users to cherry-pick pattern-set?

Pattern-Tree: Providing Cherry-Picking Options

1. 2017/02/24 09:01:00 login 127.0.0.1 user=bear12
2. 2017/02/24 09:02:00 DB Connect 127.0.0.1 user=bear12
3. 2017/02/24 09:02:00 DB Disconnect 127.0.0.1 user=bear12
4. 2017/02/24 09:04:00 logout 127.0.0.1 user=bear12

5. 2017/02/24 09:05:00 login 127.0.0.1 user=bear34
6. 2017/02/24 09:06:00 DB Connect 127.0.0.1 user=bear34
7. 2017/02/24 09:07:00 DB Disconnect 127.0.0.1 user=bear34
8. 2017/02/24 09:08:00 logout 127.0.0.1 user=bear34

9. 2017/02/24 09:09:00 login 127.0.0.1 user=bear#1
10. 2017/02/24 09:10:00 DB Connect 127.0.0.1 user=bear#1
11. 2017/02/24 09:11:00 DB Disconnect 127.0.0.1 user=bear#1
12. 2017/02/24 09:12:00 logout 127.0.0.1 user=bear#1

Input Logs

1. DATETIME login IPV4 user = WORD
2. DATETIME DB Connect IPV4 user = WORD
3. DATETIME DB Disconnect IPV4 user = WORD
4. DATETIME logout IPV4 user = WORD
5. DATETIME login IPV4 user = NOTSPACE
6. DATETIME DB Connect user = NOTSPACE
7. DATETIME DB Disconnect IPV4 user = NOTSPACE
8. DATETIME logout IPV4 user = NOTSPACE

Level 1

9. DATETIME login IPV4 user = NOTSPACE
10. DATETIME DB Connect IPV4 user = NOTSPACE
11. DATETIME DB Disconnect IPV4 user = NOTSPACE
12. DATETIME logout IPV4 user = NOTSPACE

Level 2

13. DATETIME WORD IPV4 user = NOTSPACE
14. DATETIME DB WORD IPV4 user = NOTSPACE

Level 3

15. DATETIME * WORD IPV4 user = NOTSPACE

Level 4

Pattern-Tree

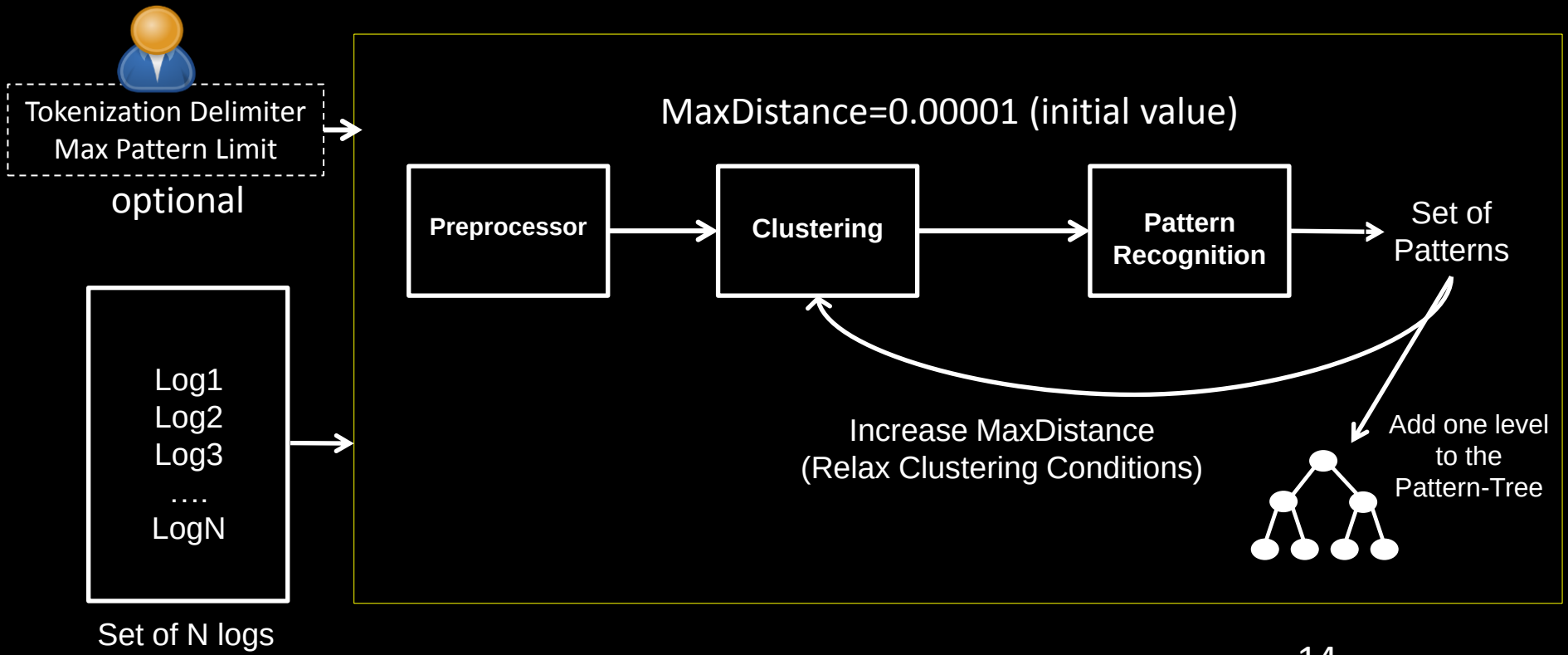
LogMine: An Automatic Pattern Generator

- LogMine: Fast Pattern Recognition for Log Analytics [CIKM 16]
  - Link: *http://www.nec-labs.com/~biplob/Papers/LogMine.pdf*

- Logmine generates a pattern-tree from a set of input logs
  - User can cherry-pick any set of patterns from this tree
  - It is a super-Fast algorithm
    - Scans log data only once
    - Scalable
      - Time complexity O(# of logs)
      - Memory complexity O(# of clusters)

- LogMine leverage the following facts
  - Logs are not randomly generated, rather generated by computing devices
  - Logs having same formats are very similar

LogMine: Workflow



LogMine: Three Steps

1. Preprocessing
   a) Tokenization
   b) Datatype identification

2. Pattern-Tree Generation (Iterative Process)
   a) Starts with a small similarity distance threshold
   b) Form clusters based similarity among logs
   c) If any cluster has multiple logs, merge them together to form one log.
   d) Add clusters in the pattern-tree
   e) If more than one clusters formed
      - Increase similarity distance threshold
      - Repeat Step a, b, c, d, e

3. Output Final Patterns from the Pattern-Tree
   a) If user inputs a max pattern limit, then outputs the first level closer to Level-0 satisfying user's limit
   b) Otherwise, outputs the tree level having minimum patterns with no wildcards (i.e., cost = 0)

Pre-Processing: Tokenization

1. 2017/02/24 09:01:00 login 127.0.0.1 user=bear12
2. 2017/02/24 09:02:00 DB Connect 127.0.0.1 user=bear12
3. 2017/02/24 09:02:00 DB Disconnect 127.0.0.1 user=bear12
4. 2017/02/24 09:04:00 logout 127.0.0.1 user=bear12
5. 2017/02/24 09:05:00 login 127.0.0.1 user=bear34
6. 2017/02/24 09:06:00 DB Connect 127.0.0.1 user=bear34
7. 2017/02/24 09:07:00 DB Disconnect 127.0.0.1 user=bear34
8. 2017/02/24 09:08:00 logout 127.0.0.1 user=bear34
9. 2017/02/24 09:09:00 login 127.0.0.1 user=bear#1
10. 2017/02/24 09:10:00 DB Connect 127.0.0.1 user=bear#1
11. 2017/02/24 09:11:00 DB Disconnect 127.0.0.1 user=bear#1
12. 2017/02/24 09:12:00 logout 127.0.0.1 user=bear#1



1. 2017/02/24 09:01:00 login 127.0.0.1 user = bear12
2. 2017/02/24 09:02:00 DB Connect 127.0.0.1 user = bear12
3. 2017/02/24 09:02:00 DB Disconnect 127.0.0.1 user = bear12
4. 2017/02/24 09:04:00 logout 127.0.0.1 user = bear12
5. 2017/02/24 09:05:00 login 127.0.0.1 user = bear34
6. 2017/02/24 09:06:00 DB Connect 127.0.0.1 user = bear34
7. 2017/02/24 09:07:00 DB Disconnect 127.0.0.1 user = bear34
8. 2017/02/24 09:08:00 logout 127.0.0.1 user = bear34
9. 2017/02/24 09:09:00 login 127.0.0.1 user = bear#1
10. 2017/02/24 09:10:00 DB Connect 127.0.0.1 user = bear#1
11. 2017/02/24 09:11:00 DB Disconnect 127.0.0.1 user = bear#1
12. 2017/02/24 09:12:00 logout 127.0.0.1 user = bear#1

- Each Individual word is a token
- Split token by tokenization delimiter if specified by the user.
  - In this example "=" is the delimiter.
- Consecutive spaces are consolidated

Preprocessing: Data-Type Identification

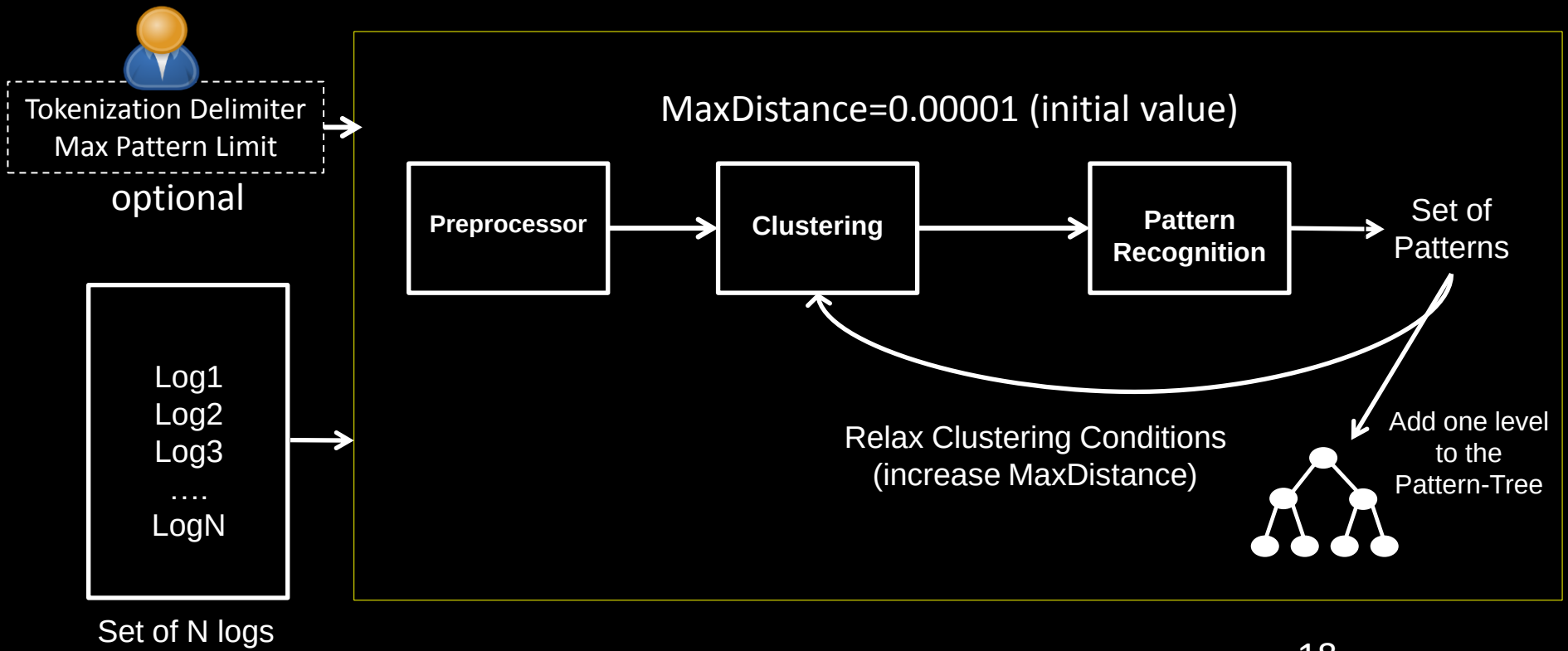
1. 2017/02/24 09:01:00 login 127.0.0.1 user = bear12
2. 2017/02/24 09:02:00 DB Connect 127.0.0.1 user = bear12
3. 2017/02/24 09:02:00 DB Disconnect 127.0.0.1 user = bear12
4. 2017/02/24 09:04:00 logout 127.0.0.1 user = bear12
5. 2017/02/24 09:05:00 login 127.0.0.1 user = bear34
6. 2017/02/24 09:06:00 DB Connect 127.0.0.1 user = bear34
7. 2017/02/24 09:07:00 DB Disconnect 127.0.0.1 user = bear34
8. 2017/02/24 09:08:00 logout 127.0.0.1 user = bear34
9. 2017/02/24 09:09:00 login 127.0.0.1 user = bear#1
10. 2017/02/24 09:10:00 DB Connect 127.0.0.1 user = bear#1
11. 2017/02/24 09:11:00 DB Disconnect 127.0.0.1 user = bear#1
12. 2017/02/24 09:12:00 logout 127.0.0.1 user = bear#1



1. DATETIME login IPV4 user = WORD
2. DATETIME DB Connect IPV4 user = WORD
3. DATETIME DB Disconnect IPV4 user = WORD
4. DATETIME logout IPV4 user = WORD
5. DATETIME login IPV4 user = WORD
6. DATETIME DB Connect IPV4 user = WORD
7. DATETIME DB Disconnect IPV4 user = WORD
8. DATETIME logout IPV4 user = WORD
9. DATETIME login IPV4 user = NOTSPACE
10. DATETIME DB Connect IPV4 user = NOTSPACE
11. DATETIME DB Disconnect IPV4 user = NOTSPACE
12. DATETIME logout IPV4 user = NOTSPACE

- Identify following datatypes: DATETIME, IPV4, NUMBER, WORD, NOTSPACE
  - ❑ Exception: Symbols and token having only alphabets
- Intuition: Developers uses meaningful words to interpret log messages ☺

LogMine: Workflow (Recap)



LogMine: Pattern-Tree Formation

Level 1

1. DATETIME login IPV4 user = WORD
2. DATETIME DB Connect IPV4 user = WORD
3. DATETIME DB Disconnect IPV4 user = WORD
4. DATETIME logout IPV4 user = WORD
5. DATETIME login IPV4 user = NOTSPACE
6. DATETIME DB Connect user = NOTSPACE
7. DATETIME DB Disconnect IPV4 user = NOTSPACE
8. DATETIME logout IPV4 user = NOTSPACE

Level 2

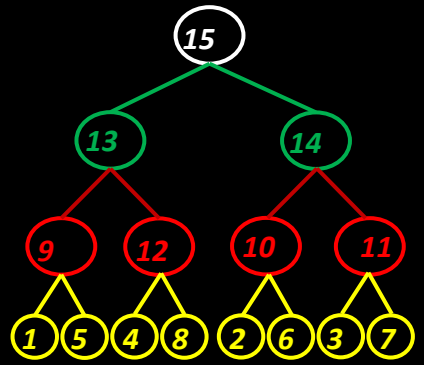
9. DATETIME login IPV4 user = NOTSPACE
10. DATETIME DB Connect IPV4 user = NOTSPACE
11. DATETIME DB Disconnect IPV4 user = NOTSPACE
12. DATETIME logout IPV4 user = NOTSPACE

Level 3

13. DATETIME WORD IPV4 user = NOTSPACE
14. DATETIME DB WORD IPV4 user = NOTSPACE

Level 4

15. DATETIME * WORD IPV4 user = NOTSPACE



Pattern Tree

Memory Usage depends on Level 1 Size

Optimization: Fast Level 1 Formation

1. DATETIME login IPV4 user = WORD
2. DATETIME DB Connect IPV4 user = WORD
3. DATETIME DB Disconnect IPV4 user = WORD
4. DATETIME logout IPV4 user = WORD
5. DATETIME login IPV4 user = WORD
6. DATETIME DB Connect IPV4 user = WORD
7. DATETIME DB Disconnect IPV4 user = WORD
8. DATETIME logout IPV4 user = WORD
9. DATETIME login IPV4 user = NOTSPACE
10. DATETIME DB Connect IPV4 user = NOTSPACE
11. DATETIME DB Disconnect IPV4 user = NOTSPACE
12. DATETIME logout IPV4 user = NOTSPACE

Preprocessed Logs



1. DATETIME login IPV4 user = WORD
2. DATETIME DB Connect IPV4 user = WORD
3. DATETIME DB Disconnect IPV4 user = WORD
4. DATETIME logout IPV4 user = WORD
5. DATETIME login IPV4 user = WORD
6. DATETIME DB Connect IPV4 user = WORD
7. DATETIME DB Disconnect IPV4 user = WORD
8. DATETIME logout IPV4 user = WORD
9. DATETIME login IPV4 user = NOTSPACE
10. DATETIME DB Connect IPV4 user = NOTSPACE
11. DATETIME DB Disconnect IPV4 user = NOTSPACE
12. DATETIME logout IPV4 user = NOTSPACE

Preprocessed Logs

Level 1 could be formed without clustering

Hint: Identify unique log lines.

Optimization: Fast Level 1 Formation

1. DATETIME login IPV4 user = WORD
2. DATETIME DB Connect IPV4 user = WORD
3. DATETIME DB Disconnect IPV4 user = WORD
4. DATETIME logout IPV4 user = WORD
5. DATETIME login IPV4 user = WORD
6. DATETIME DB Connect IPV4 user = WORD
7. DATETIME DB Disconnect IPV4 user = WORD
8. DATETIME logout IPV4 user = WORD
9. DATETIME login IPV4 user = NOTSPACE
10. DATETIME DB Connect IPV4 user = NOTSPACE
11. DATETIME DB Disconnect IPV4 user = NOTSPACE
12. DATETIME logout IPV4 user = NOTSPACE

Preprocessed Logs



1. DATETIME login IPV4 user = WORD
2. DATETIME DB Connect IPV4 user = WORD
3. DATETIME DB Disconnect IPV4 user = WORD
4. DATETIME logout IPV4 user = WORD
5. DATETIME login IPV4 user = NOTSPACE
6. DATETIME DB Connect IPV4 user = NOTSPACE
7. DATETIME DB Disconnect IPV4 user = NOTSPACE
8. DATETIME logout IPV4 user = NOTSPACE

Level 1 (Unique lines)

Level 1 could be formed without clustering
Hint: Identify unique log lines.

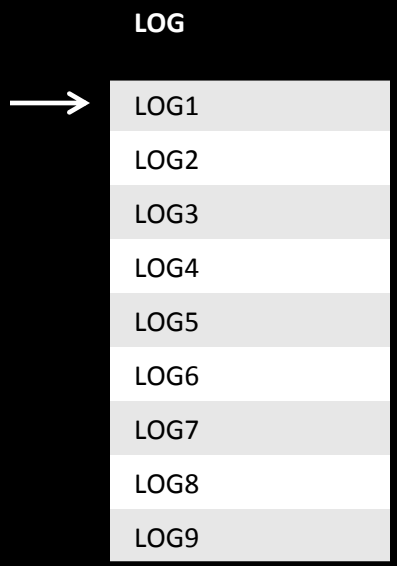
Clustering: Example

Sample LOG

LOG1
LOG2
LOG3
LOG4
LOG5
LOG6
LOG7
LOG8
LOG9

MaxDistance = 0.01

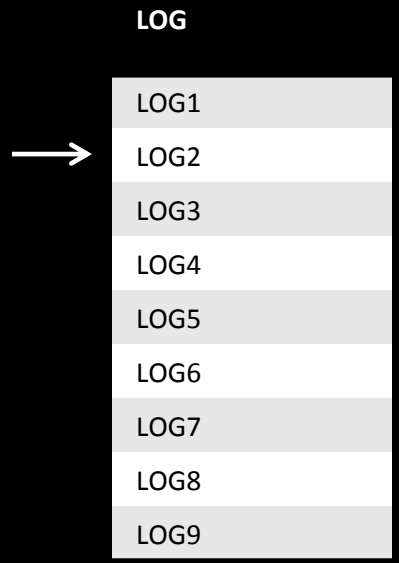
Clustering: Example



MaxDistance = 0.01

LOG1

Clustering: Example



MaxDistance = 0.01

Dist (LOG1 , LOG2) = 0.2

LOG1

LOG2

Example: Distance Calculation

Log1: DATETIME login IPV4 WORD NUMBER

Log2: DATETIME login IPV4 WORD

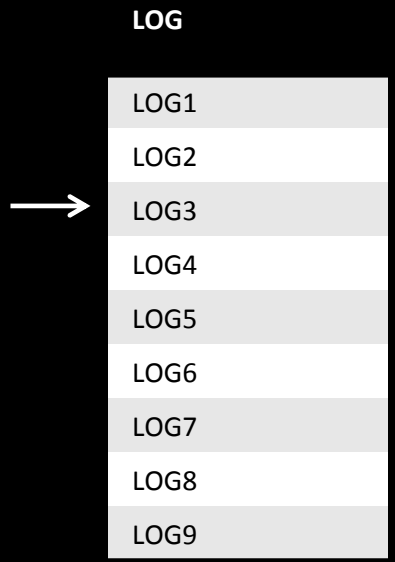
Similarity (Log1, Log2) = 4/5 = 0.80

Distance (Log1, Log2) = 1 – Similarity = 1 – 0.80 = 0.20

$$Dist(P,Q) = 1 - \sum_{i=1}^{Min(|P|,|Q|)} \frac{Score(Pi,Qi)}{Max(|P|,|Q|)}$$

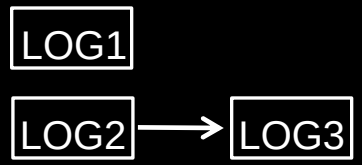
$$Score(X,Y) = \begin{cases} 1 & If\ X = Y \\ 0 & Otherwise \end{cases}$$

Clustering: Example

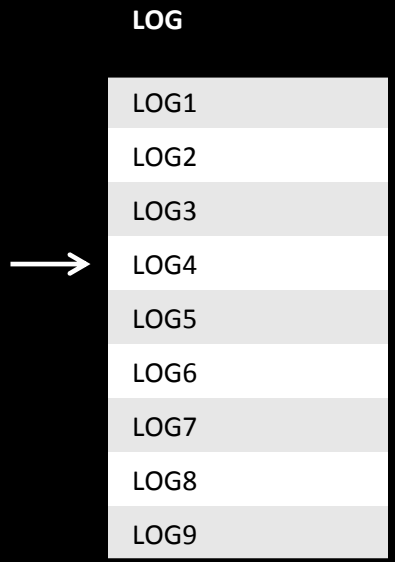


MaxDistance = 0.01

Dist (LOG3 , LOG1) = 0.8
Dist (LOG3 , LOG2) = 0.001

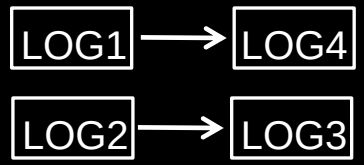


Clustering: Example

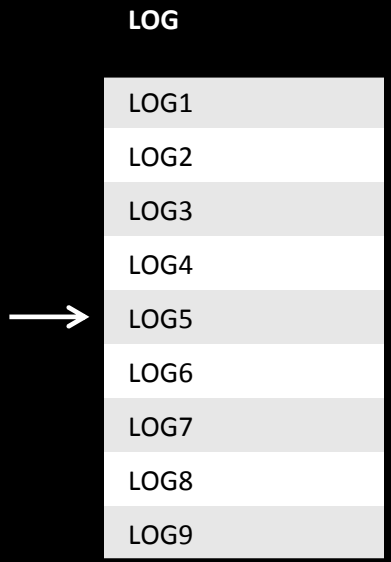


MaxDistance = 0.01

Dist (LOG4 , LOG1) = 0

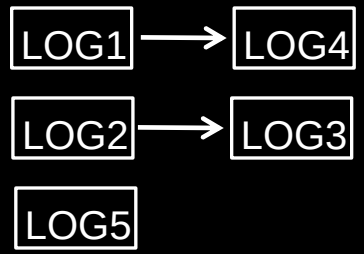


Clustering: Example



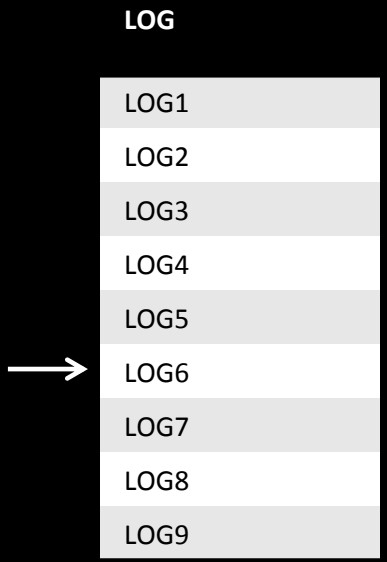
MaxDistance = 0.01

Dist (LOG5 , LOG1) = 0.2
Dist (LOG5 , LOG2) = 0.5



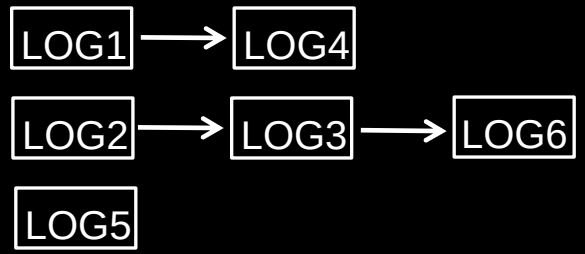
Optimization: Compare only with the first log in a cluster

Clustering: Example



MaxDistance = 0.01

Dist (LOG6 , LOG1) = 0.3
Dist (LOG6 , LOG2) = 0.001



Clustering: Example

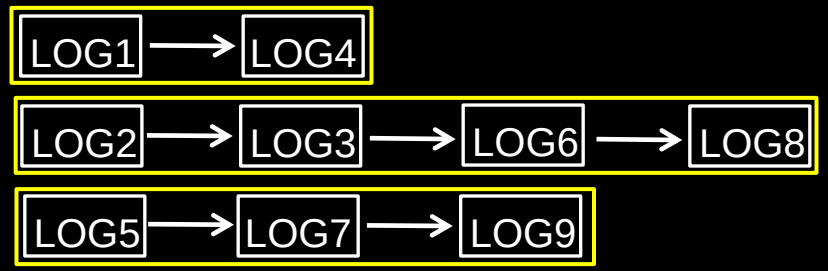
LOG
LOG1
LOG2
LOG3
LOG4
LOG5
LOG6
LOG7
LOG8
LOG9

Max Distance = 0.01

Cluster1

Cluster2

Cluster3



Pattern @ Cluster (Smith–Waterman Algorithm)

- Logs in a cluster are merged together to produce one final pattern per cluster.
- Optimization: No need to follow any merging order.

LOG1: DATETIME WORD IPV4 user = NOTSPACE

LOG2: DATETIME DB WORD IPV4 user = NOTSPACE



Alignment

LOG1: DATETIME **---** WORD IPV4 user = NOTSPACE

LOG2: DATETIME DB WORD IPV4 user = NOTSPACE



Merge the Alignments

Result: DATETIME ***** WORD IPV4 user = NOTSPACE

LogMine: Output Pattern-Set

Level 1

1. DATETIME login IPV4 user = WORD
2. DATETIME DB Connect IPV4 user = WORD
3. DATETIME DB Disconnect IPV4 user = WORD
4. DATETIME logout IPV4 user = WORD
5. DATETIME login IPV4 user = NOTSPACE
6. DATETIME DB Connect user = NOTSPACE
7. DATETIME DB Disconnect IPV4 user = NOTSPACE
8. DATETIME logout IPV4 user = NOTSPACE

Level 2

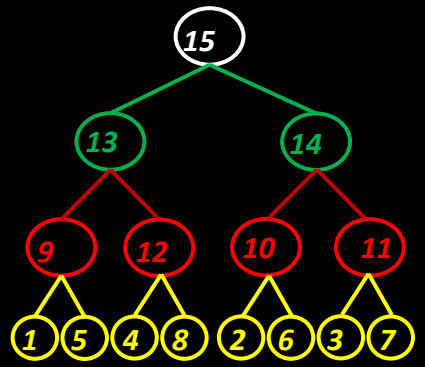
9. DATETIME login IPV4 user = NOTSPACE
10. DATETIME DB Connect IPV4 user = NOTSPACE
11. DATETIME DB Disconnect IPV4 user = NOTSPACE
12. DATETIME logout IPV4 user = NOTSPACE

Level 3

13. DATETIME WORD IPV4 user = NOTSPACE
14. DATETIME DB WORD IPV4 user = NOTSPACE

Level 4

15. DATETIME * WORD IPV4 user = NOTSPACE



Pattern Tree

Cost: 12

Cost: 0

Cost: 0

Cost: 0

Max Pattern Limit = 5, then Level 2 will be the output

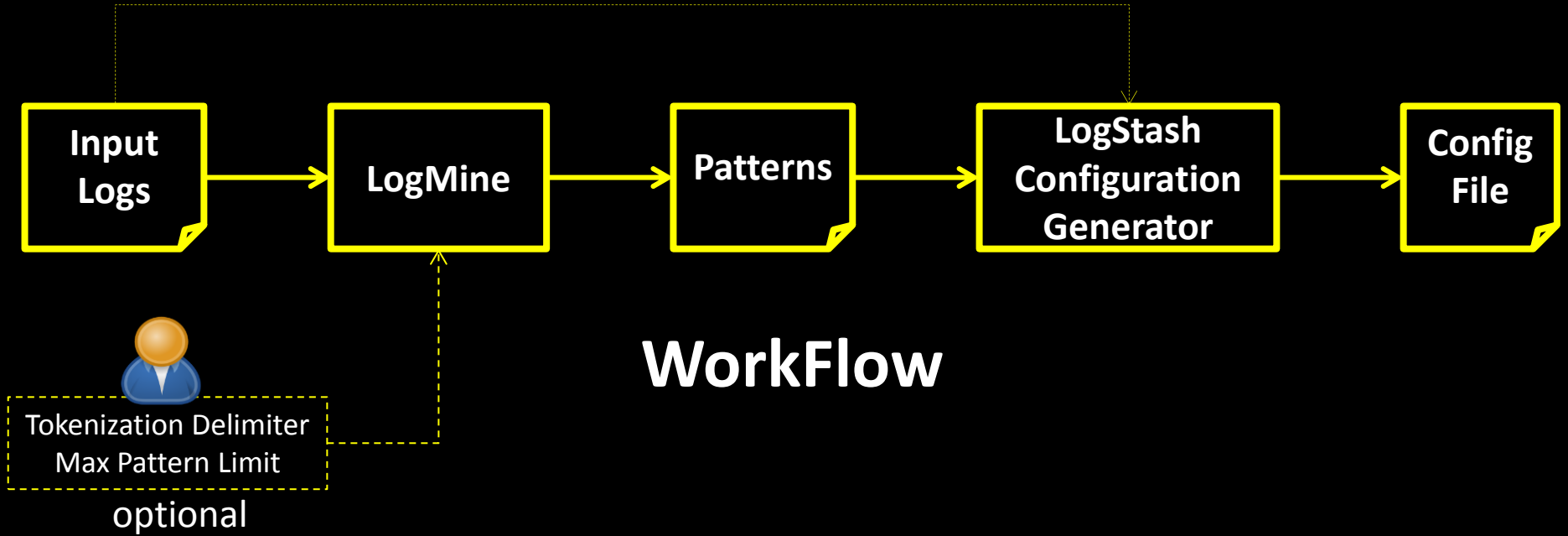
Cost is Calculated based on WildCards and covered Logs Count. For example, pattern 15 contains one wildcard and it covers 12 logs – in this case cost is estimated as 1*12= 12.

If user gives no Max Pattern limit on pattern count, then Level 3 will be the final output as it has the minimum cost with minimum number of patterns.

Outline

- Pattern Generation Algorithm
- Demo Use-cases

LogMine + LogStash : Configuration Generation



Demo 1: Datacenter Power Usage

```
Mar  3 16:25:03 envctl APC_PDU_LEGAMPS: [INFO] PDU=pdu-2g04-apc-rack3a Leg=1
Amps=4.6
Mar  3 16:25:03 envctl APC_PDU_LEGAMPS: [INFO] PDU=pdu-2g04-rack1a Leg=1
Amps=4.1
.
.
.
Mar  3 16:30:03 envctl APC_PDU_LEGAMPS: [INFO] PDU=pdu-2g04-apc-rack1a Leg=2
Amps=1.2
Mar  3 16:30:03 envctl APC_PDU_LEGAMPS: [INFO] PDU=pdu-2g04-apc-rack1a Leg=3
Amps=0.0
```

Sample Logs

Input Logs: 14976 lines, Tokenization Delimiter : “=“

Patterns: 1, Time Taken: 9 seconds (Single core)

Demo 1: LogStash Parsing Configuration

```
filter {
       mutate {
         add_field => { "raw_input" => "%{message}" }
       }

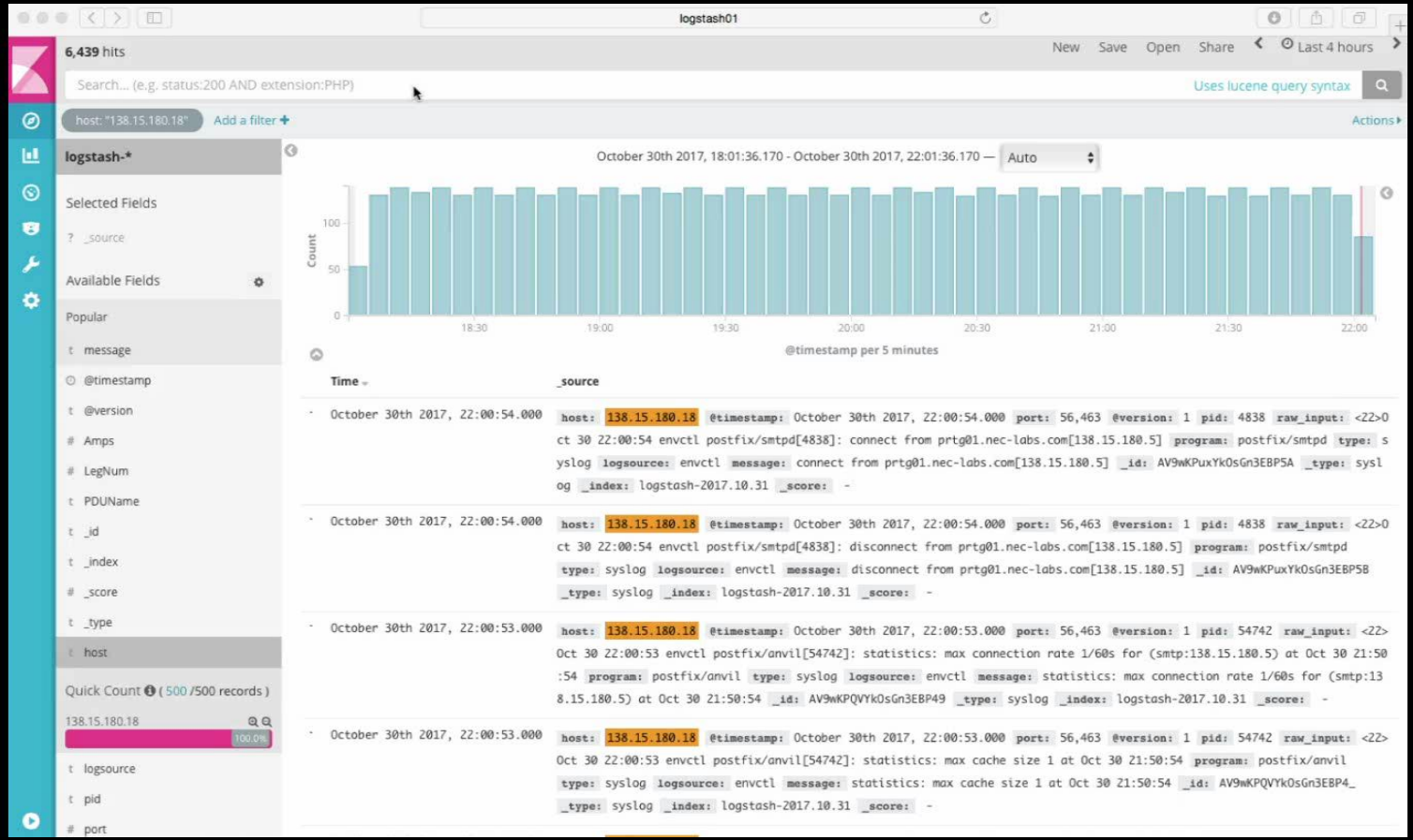
       mutate {
         gsub => [
           "message", "=", " \0 ",
           "message", "\s+", " "
         ]
       }

       grok {
         match => { "message" => "^(?<logTime>%{MONTH:month}
%{MONTHDAY:day} %{HOUR:hour}:%{MINUTE:minute}:%{SECOND:second}) envctl
APC_PDU_LEGAMPS: \[INFO\] PDU = %{NOTSPACE:PDUName} Leg =
%{NUMBER:LegNum:int} Amps = %{NUMBER:Amps:float}$"}
        }

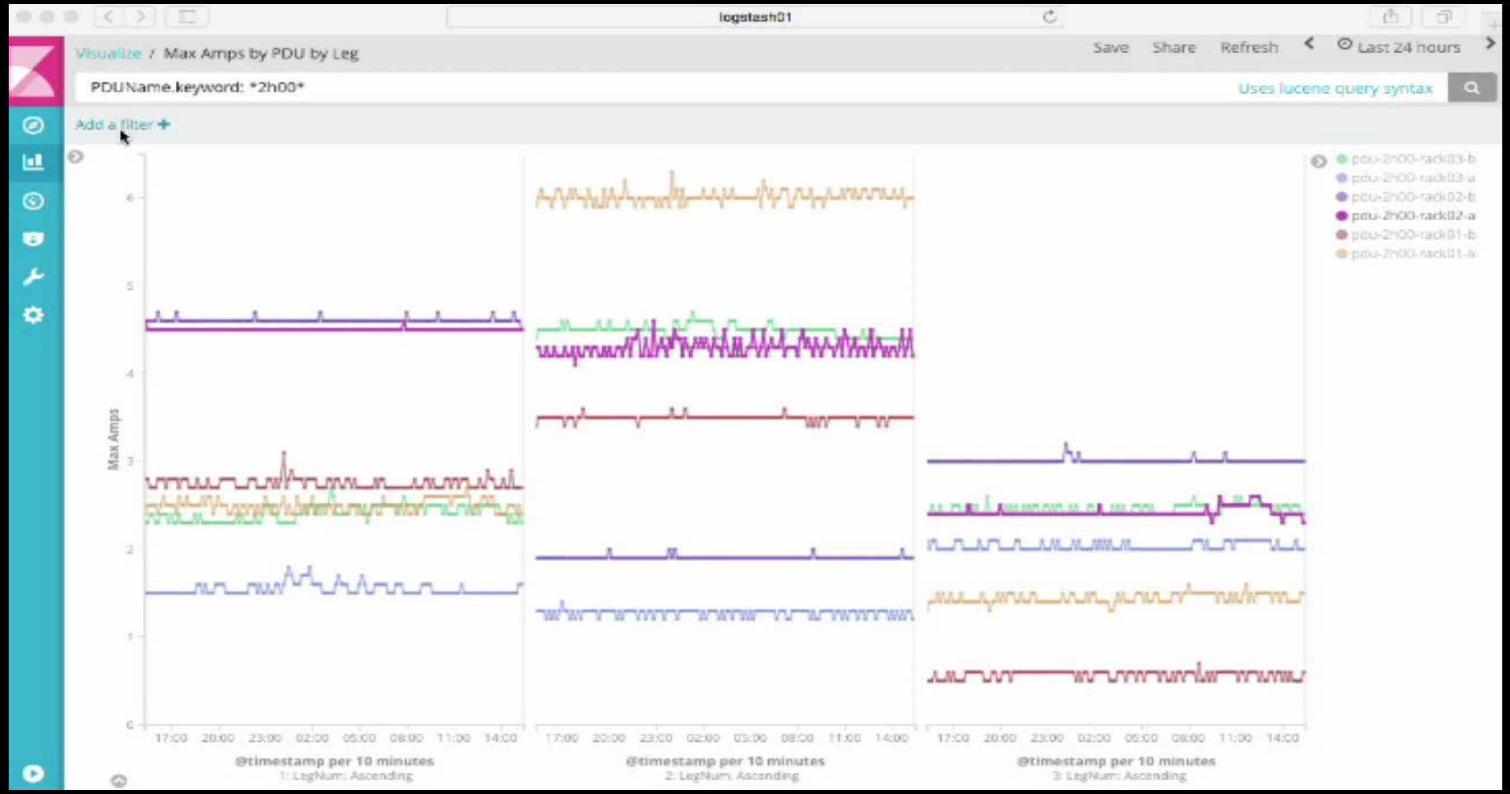
        date {
          match => [ "logTime", "MMM dd HH:mm:ss" ]
          target => "@timestamp"
        }

        mutate {
          remove_field => ['year']
          remove_field => ['month']
          remove_field => ['day']
          remove_field => ['hour']
          remove_field => ['minute']
          remove_field => ['second']
          remove_field => ['logTime']
        }
}
```

Demo 1: Datacenter Power Usage



Demo 1: Datacenter Power Usage



Demo 2: Custom Application Log

```
(0):GetFormControl():2[25 21:39:21] SQL SELECT TABLE: tblFormControl WHERE: tblFormControl.oFCID='6a602aaa-9afd-4e2c-95e9-ee900dde4b50'

(0):_GetObjects():1[25 21:39:21] KVIEW running: 'My Releases Needing Followup'

(0):_GetObjects():2[25 21:39:21] SQL SELECT TABLE: tblContent WHERE:  oPID='ad1aa290-01ae-4edd-989c-1cee2ba63707'  AND ( ( (  ( ( ( oID IN (SELECT oID
FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='6a602aaa-9afd-4e2c-95e9-ee900dde4b50' AND ((tblFormData.tValue IS NOT NULL AND
tblFormData.tValue > '1799-01-01T00:00:00.000' AND tblFormData.tValue < '2200-01-01T00:00:00.000')) ))) AND (( (nType!=15 OR oID IN (SELECT oFORMINSTID
FROM tblFormInstance WHERE tblFormInstance.oFORMID='3ebee358-2087-43d4-908b-df9ed04e74cc')) AND (nType!=14 OR tblContent.oID='3ebee358-2087-43d4-908b-
df9ed04e74cc') )) AND ( ( oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='7e68b547-0869-4a56-a664-26b32d0b5804' AND
((tblFormData.tValue<='2017-10-26T03:59:59.000' OR tblFormData.tValue IS NULL)) ))) AND ( ( oID IN (SELECT oID FROM tblFormData WHERE
oFORMINSTID=tblContent.oID AND oFCID='e28c6d82-532d-4618-a0a8-d62a15e00731' AND (tblFormData.sValue=N'dadf4506-2995-42c4-8616-cb43786fa382') ))) AND (
( oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='2a004b8d-16ef-4973-8ec8-be7db392e436' AND ((tblFormData.sValue<>N'Y'
OR tblFormData.sValue IS NULL)) ))) ) ) )  AND (nType!=15 OR oID IN (SELECT oFORMINSTID FROM tblFormInstance WHERE oFORMID='3ebee358-2087-43d4-908b-
df9ed04e74cc') ) AND 1=1 AND (  nType=15 ) ) AND  (oID IN (SELECT oID FROM tblPerm WHERE (oGrantID='dadf4506-2995-42c4-8616-cb43786fa382' OR
oGrantID='[Authenticated]' OR oGrantID='[Anonymous]' OR oGrantID IN (SELECT oParent FROM tblMembership WHERE oChild='dadf4506-2995-42c4-8616-
cb43786fa382')) AND fRead=1) )  AND (nSubType!=2 AND nSubType!=1 AND nSubType!=4 AND nSubType!=5) AND (nType!=15 OR nVersion!=0)

(0):_GetObjects():2[25 21:39:21] SQL SELECT TABLE: tblFormData WHERE: oFORMINSTID = '418f38ce-a35e-47db-8e1c-88fc7eb09de3' AND oFCID IN ('fe53e626-
13ae-4206-8bc7-178cbc69b866', '6a602aaa-9afd-4e2c-95e9-ee900dde4b50', '1bfb5785-4f29-488b-8d09-c42faef48fee', '611e6c07-c8ba-44c5-b745-485e9faddcb4',
'3d8dfd3d-2c62-4c19-8cb5-a3bec8bf729b', '7e68b547-0869-4a56-a664-26b32d0b5804')

(0):bpPage_PreInit():(w3wp.exe,0x1bd8,17):1[25 21:40:22] Page: http://app.1nec-labs.com/kv_right.aspx?kvid=3b676138-0fe6-4dbd-a20b-
49fe5b07725d&showmax=1, request URL: http://app1.nec-labs.com/kv_right.aspx?kvid=3b676138-0fe6-4dbd-a20b-49fe5b07725d&showmax=1, remote IP:
138.15.207.74
```

Sample Logs

Input Logs: 242232 lines, Tokenization Delimiter: “[=]“

Patterns: 339, Time Taken: 27 seconds (Single core)

Demo 2: Custom Application Log

```
(0):GetFormControl():2[25 21:39:21] SQL SELECT TABLE: tblFormControl WHERE: tblFormControl.oFCID='6a602aaa-9afd-4e2c-95e9-ee900dde4b50'

(0):_GetObjects():1[25 21:39:21] KVIEW running: 'My Releases Needing Followup'

(0):_Ge:2[25 21:39:21] SQL ontent WHERE:  oPID='ad1aa290-01ae-4edd-989c-1cee2ba63707'  AND ( ( (  ( ( ( oID IN (SELECT oID
FROM tbl FCID='6a602aaa-9afd-4e2c-95e9-ee900dde4b50' AND ((tblFormData.tValue IS NOT NULL AND
tblFormData.tValue > '1799-01-01T00:00:00.000' AND tblFormData.tValue < '2200-01-01T00:00:00.000')) ))) AND (( (nType!=15 OR oID IN (SELECT oFORMINSTID
FROM tblFormInstance WHERE tblFormInstance.oFORMID='3ebee358-2087-43d4-908b-df9ed04e74cc')) AND (nType!=14 OR tblContent.oID='3ebee358-2087-43d4-908b-
df9ed04e74cc') )) AND ( ( oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='7e68b547-0869-4a56-a664-26b32d0b5804' AND
((tblFormData.tValue<='2017-10-26T03:59:59.000' OR tblFormData.tValue IS NULL)) ))) AND ( ( oID IN (SELECT oID FROM tblFormData WHERE
oFORMINSTID=tblContent.oID AND oFCID='e28c6d82-532d-4618-a0a8-d62a15e00731' AND (tblFormData.sValue=N'dadf4506-2995-42c4-8616-cb43786fa382') ))) AND (
( oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='2a004b8d-16ef-4973-8ec8-be7db392e436' AND ((tblFormData.sValue<>N'Y'
OR tblFormData.sValue IS NULL)) ))) ) ) )  AND (nType!=15 OR oID IN (SELECT oFORMINSTID FROM tblFormInstance WHERE oFORMID='3ebee358-2087-43d4-908b-
df9ed04e74cc') ) AND 1=1 AND (  nType=15 ) ) AND  (oID IN (SELECT oID FROM tblPerm WHERE (oGrantID='dadf4506-2995-42c4-8616-cb43786fa382' OR
oGrantID='[Authenticated]' OR oGrantID='[Anonymous]' OR oGrantID IN (SELECT oParent FROM tblMembership WHERE oChild='dadf4506-2995-42c4-8616-
cb43786fa382')) AND fRead=1) )  AND (nSubType!=2 AND nSubType!=1 AND nSubType!=4 AND nSubType!=5) AND (nType!=15 OR nVersion!=0)

(0):_GetObjects():2[25 21:39:21] SQL SELECT TABLE: tblFormData WHERE: oFORMINSTID = '418f38ce-a35e-47db-8e1c-88fc7eb09de3' AND oFCID IN ('fe53e626-
13ae-4206-8bc7-178cbc69b866', '6a602aaa-9afd-4e2c-95e9-ee900dde4b50', '1bfb5785-4f29-488b-8d09-c42faef48fee', '611e6c07-c8ba-44c5-b745-485e9faddcb4',
'3d8dfd3d-2c62-4c19-8cb5-a3bec8bf729b', '7e68b547-0869-4a56-a664-26b32d0b5804')

(0):bpPage_PreInit():(w3wp.exe,0x1bd8,17):1[25 21:40:22] Page: http://app1.nec-labs.com/kv_right.aspx?kvid=3b676138-0fe6-4dbd-a20b-
49fe5b07725d&showmax=1, request URL: http://app1.nec-labs.com/kv_right.aspx?kvid=3b676138-0fe6-4dbd-a20b-49fe5b07725d&showmax=1, remote IP:
138.15.207.74
```

Sample Logs

Input Logs: 242232 lines, Tokenization Delimiter: “[=]“

Patterns: 339, Time Taken: 27 seconds (Single core)

Demo 2: Pattern-Tree Stats

Level	Patterns	Cost
1	2582	0
2	339	0
3	256	297396
4	231	327373
5	230	327397
6	219	336073
7	217	336127
8	214	336153
9	213	336175
10	211	338036
11	210	338074
12	195	340653
13	176	353253
14	132	456600
15	121	641338
16	106	655485
17	82	669673
18	57	742742
19	52	745047
20	48	749203
21	39	761506
22	34	794590
23	30	795275
24	24	1171696
25	20	1333119
26	9	939438
27	1	726696

We put no upper limit on pattern count., therefore Level 2 is selected as the final output because it has the minimum cost with minimum number of patterns.

Demo 2: Sample Parsing Pattern

(0):_GetObjects():2[25 21:39:21] SQL SELECT TABLE: tblContent WHERE: oPID='ad1aa290-01ae-4edd-989c-1cee2ba63707' AND ((((((oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='6a602aaa-9afd-4e2c-95e9-ee900dde4b50' AND ((tblFormData.tValue IS NOT NULL AND tblFormData.tValue > '1799-01-01T00:00:00.000' AND tblFormData.tValue < '2200-01-01T00:00:00.000'))))) AND (((nType!=15 OR oID IN (SELECT oFORMINSTID FROM tblFormInstance WHERE tblFormInstance.oFORMID='3ebee358-2087-43d4-908b-df9ed04e74cc')) AND (nType!=14 OR tblContent.oID='3ebee358-2087-43d4-908b-df9ed04e74cc'))) AND ((oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='7e68b547-0869-4a56-a664-26b32d0b5804' AND ((tblFormData.tValue<='2017-10-26T03:59:59.000' OR tblFormData.tValue IS NULL))))) AND ((oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='e28c6d82-532d-4618-a0a8-d62a15e00731' AND (tblFormData.sValue=N'dadf4506-2995-42c4-8616-cb43786fa382')))) AND ((oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='2a004b8d-16ef-4973-8ec8-be7db392e436' AND ((tblFormData.sValue<>N'Y' OR tblFormData.sValue IS NULL)))))))) AND (nType!=15 OR oID IN (SELECT oFORMINSTID FROM tblFormInstance WHERE oFORMID='3ebee358-2087-43d4-908b-df9ed04e74cc')) AND 1=1 AND (nType=15)) AND (oID IN (SELECT oID FROM tblPerm WHERE (oGrantID='dadf4506-2995-42c4-8616-cb43786fa382' OR oGrantID='[Authenticated]' OR oGrantID='[Anonymous]' OR oGrantID IN (SELECT oParent FROM tblMembership WHERE oChild='dadf4506-2995-42c4-8616-cb43786fa382')) AND fRead=1)) AND (nSubType!=2 AND nSubType!=1 AND nSubType!=4 AND nSubType!=5) AND (nType!=15 OR nVersion!=0)

```
grok {
        match => { "message" => "^%{NOTSPACE:P199NS1}\[(?<logTime>%{MONTHDAY:day} %{HOUR:hour}:%{MINUTE:minute}:%{SECOND:second})\]
SQL SELECT TABLE: tblContent WHERE: oPID = 'ad1aa290\-01ae\-4edd\-989c\-1cee2ba63707' AND \( \( \( \( \( \( oID IN \(SELECT oID FROM tblFormData WHERE oFORMINSTID
= tblContent\.oID AND oFCID = '6a602aaa\-9afd\-4e2c\-95e9\-ee900dde4b50' AND \(\(tblFormData\.tValue IS NOT NULL AND tblFormData\.tValue > '1799\-01\-
01T00:00:00\.000' AND tblFormData\.tValue < '2200\-01\-01T00:00:00\.000'\)\) \)\)\) AND \(\( \(nType! = %{NUMBER:P199F1:int} OR oID IN \(SELECT oFORMINSTID FROM
tblFormInstance WHERE tblFormInstance\.oFORMID = '3ebee358\-2087\-43d4\-908b\-df9ed04e74cc'\)\) AND \(nType! = %{NUMBER:P199F2:int} OR tblContent\.oID =
'3ebee358\-2087\-43d4\-908b\-df9ed04e74cc'\) \)\) AND \( \( oID IN \(SELECT oID FROM tblFormData WHERE oFORMINSTID = tblContent\.oID AND oFCID = '7e68b547\-0869\-
4a56\-a664\-26b32d0b5804' AND \(\(tblFormData\.tValue< = %{NOTSPACE:P199NS2:int} OR tblFormData\.tValue IS NULL\)\) \)\)\) AND \( \( oID IN \(SELECT oID FROM
tblFormData WHERE oFORMINSTID = tblContent\.oID AND oFCID = 'e28c6d82\-532d\-4618\-a0a8\-d62a15e00731' AND \(tblFormData\.sValue = %{NOTSPACE:P199NS3} \)\)\)
AND \( \( oID IN \(SELECT oID FROM tblFormData WHERE oFORMINSTID = tblContent\.oID AND oFCID = '2a004b8d\-16ef\-4973\-8ec8\-be7db392e436' AND
\(\(tblFormData\.sValue<>N'Y' OR tblFormData\.sValue IS NULL\)\) \)\)\)\) \) \) \) AND \(nType! = %{NUMBER:P199F3:int} OR oID IN \(SELECT oFORMINSTID FROM
tblFormInstance WHERE oFORMID = '3ebee358\-2087\-43d4\-908b\-df9ed04e74cc'\) \) AND %{NUMBER:P199F4:int} = %{NUMBER:P199F5:int} AND \( nType =
%{NUMBER:P199F6:int} \) \) AND \(oID IN \(SELECT oID FROM tblPerm WHERE \(oGrantID = %{NOTSPACE:P199NS4} OR oGrantID = ' \[ Authenticated \] ' OR oGrantID = ' \[
Anonymous \] ' OR oGrantID IN \(SELECT oParent FROM tblMembership WHERE oChild = %{NOTSPACE:P199NS5} AND fRead = 1\) \) AND \(nSubType! =
%{NUMBER:P199F7:int} AND nSubType! = %{NUMBER:P199F8:int} AND nSubType! = %{NUMBER:P199F9:int} AND nSubType! = 5\) AND \(nType! = %{NUMBER:P199F10:int} OR
nVersion! = 0\) $"}
        remove_field => [ "tags" ]
        add_tag => [ "pattern199" ]
        tag_on_failure => []
    }
```

```
date {
      match => [ "logTime", "dd HH:mm:ss" ]
      target => "@timestamp"
    }
```

Demo 2: Sample Parsed Output

(0):_GetObjects():2[25 21:39:21] SQL SELECT TABLE: tblContent WHERE: oPID='ad1aa290-01ae-4edd-989c-1cee2ba63707' AND ((((((oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='6a602aaa-9afd-4e2c-95e9-ee900dde4b50' AND ((tblFormData.tValue IS NOT NULL AND tblFormData.tValue > '1799-01-01T00:00:00.000' AND tblFormData.tValue < '2200-01-01T00:00:00.000'))))) AND (((nType!=15 OR oID IN (SELECT oFORMINSTID FROM tblFormInstance WHERE tblFormInstance.oFORMID='3ebee358-2087-43d4-908b-df9ed04e74cc')) AND (nType!=14 OR tblContent.oID='3ebee358-2087-43d4-908b-df9ed04e74cc'))) AND ((oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='7e68b547-0869-4a56-a664-26b32d0b5804' AND ((tblFormData.tValue<='2017-10-26T03:59:59.000' OR tblFormData.tValue IS NULL))))) AND ((oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='e28c6d82-532d-4618-a0a8-d62a15e00731' AND (tblFormData.sValue=N'dadf4506-2995-42c4-8616-cb43786fa382')))) AND ((oID IN (SELECT oID FROM tblFormData WHERE oFORMINSTID=tblContent.oID AND oFCID='2a004b8d-16ef-4973-8ec8-be7db392e436' AND ((tblFormData.sValue<>N'Y' OR tblFormData.sValue IS NULL)))))))) AND (nType!=15 OR oID IN (SELECT oFORMINSTID FROM tblFormInstance WHERE oFORMID='3ebee358-2087-43d4-908b-df9ed04e74cc')) AND 1=1 AND (nType=15)) AND (oID IN (SELECT oID FROM tblPerm WHERE (oGrantID='dadf4506-2995-42c4-8616-cb43786fa382' OR oGrantID='[Authenticated]' OR oGrantID='[Anonymous]' OR oGrantID IN (SELECT oParent FROM tblMembership WHERE oChild='dadf4506-2995-42c4-8616-cb43786fa382')) AND fRead=1)) AND (nSubType!=2 AND nSubType!=1 AND nSubType!=4 AND nSubType!=5) AND (nType!=15 OR nVersion!=0)


```
{
   "type": "lisa-demo2",
   "tags": ["pattern199"],
   "P199NS1": "(0):_GetObjects():2",
    "@timestamp": "2017-01-25T21:39:21.000Z",
   "P199F1": "15",
   "P199F2": "14",
   "P199NS2": "'2017-10-26T03:59:59.000'",
   "P199NS3": "N'dadf4506-2995-42c4-8616-cb43786fa382')",
   "P199F3": "15",
   "P199F4": "1",
   "P199F5": "1",
   "P199F6": "15",
   "P199NS4": "'dadf4506-2995-42c4-8616-cb43786fa382'",
  "P199NS5": "'dadf4506-2995-42c4-8616-cb43786fa382'))“,
   "P199F7": "2",
   "P199F8": "1",
   "P199F9": "4",
   "P199F10": "15"
}
```

Summary

1. LogMine is fast and scalable ☺
2. It can work with no (or minimal) human Involvement ☺
3. It is flexible because user can cherry-pick any desired pattern-set ☺
4. It is the basic component of a real product ☺
5. Paper: *http://www.nec-labs.com/~biplob/Papers/LogMine.pdf*

All these benefits come from the fact that
LogMine is customized for logs.


Thank
you