SentiLog: Anomaly Detecting on Parallel File Systems via Log-based Sentiment Analysis

Di Zhang[1], Dong Dai[1], Runzhou Han[2], Mai Zheng[2]

[1]University of North Carolina at Charlotte

[2]Iowa State University

Outline

1. Background
2. Challenge
3. SentiLog Design
4. Evaluation
5. Conclusion

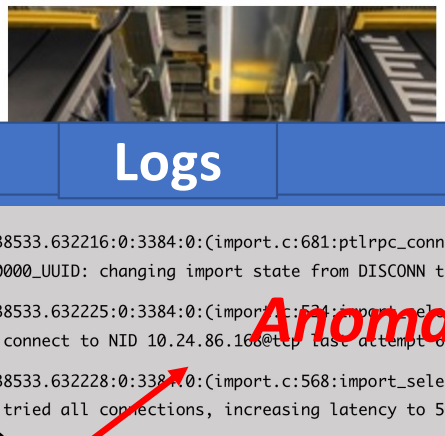
Outline

1. Background
2. Challenge
3. SentiLog Design
4. Evaluation
5. Conclusion

Importance of Anomaly Detection

Fugaku

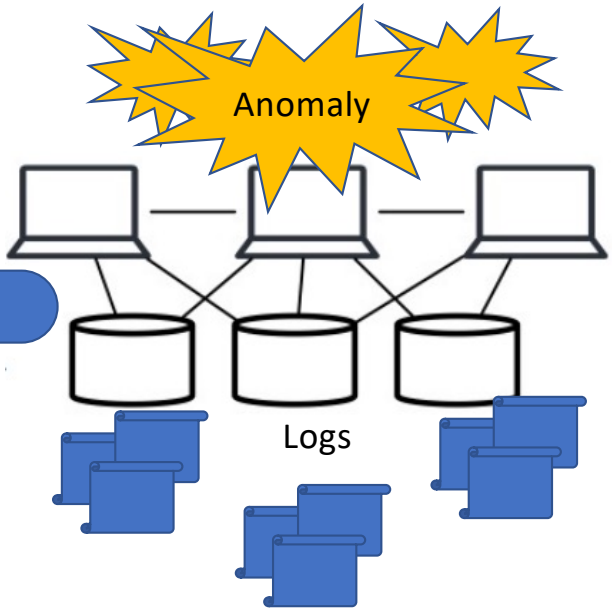
Summit



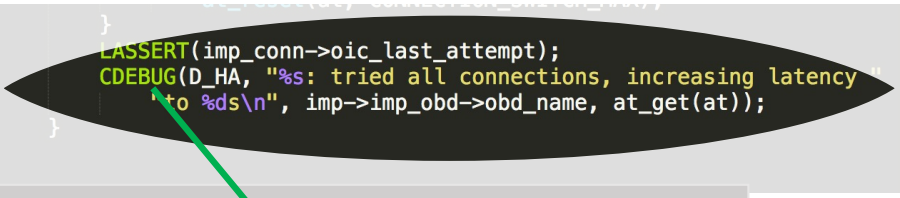
Logs

```
00000100:00080000:0.0:1583538533.632216:0:3384:0:(import.c:681:ptlrpc_connect_import())
ffff8f3dc0323000 lustre-MDT0000_UUID: changing import state from DISCONN to CONNECTING

00000100:00080000:0.0:1583538533.632225:0:3384:0:(import.c:531:import_select_connection())
lustre-MDT0000-lwp-OST0002: connect to NID 10.24.86.168@tcp last attempt 6597178959

00000100:00080000:0.0:1583538533.632228:0:3384:0:(import.c:568:import_select_connection())
lustre-MDT0000-lwp-OST0002: tried all connections, increasing latency to 50s

00000100:00080000:0.0:1583538558.632332:0:3384:0:(pinger.c:217:ptlrpc_pinger_process_import())
lustre-MDT0000-lwp-OST0002_UUID->lustre-MDT0000_UUID: level CONNECTING/4 force 0 force_next 0
deactive 0 pingable 1 suppress 0

00000100:00080000:0.0:1583538558.632342:0:3384:0:(pinger.c:230:ptlrpc_pinger_process_import())
lustre-MDT0000-lwp-OST0002_UUID->lustre-MDT0000_UUID: not pinging (in recovery or recovery
disabled: CONNECTING)

00000100:00080000:0.0:1583538583.[illegible]:3384:0:(import.c:568:import_select_connection())
lustre-MDT0000-lwp-OST0002: tried all [illegible]ctions, increasing latency to 60s
```

Anomaly!



Logs of PFSes



Log Content

Log Level

Log Index

```
00000100:00080000:0.0:1583538533.632225:0:3384:0:(import.c:524:import_select_connection())
lustre-MDT0000-lwp-OST0002: connect to NID 10.24.86.168@tcp last attempt 6557138959
```

1

```
00000100:00080000:0.0:1583538533.632228:0:3384:0:(import.c:568:import_select_connection())
lustre-MDT0000-lwp-OST0002: tried all connections, increasing latency to 50s
```

CDEBUG

```
00000100:00080000:0.0:1583538558.632332:0:3384:0:(pinger.c:217:ptlrpc_pinger_process_import())
lustre-MDT0000-lwp-OST0002_UUID->lustre-MDT0000_UUID: level CONNECTING/4 force 0 force_next 0
deactive 0 pingable 1 suppress 0
```

2

```
00000100:00080000:0.0:1583538558.632342:0:3384:0:(pinger.c:230:ptlrpc_pinger_process_import())
lustre-MDT0000-lwp-OST0002_UUID->lustre-MDT0000_UUID: not pinging (in recovery or recovery
disabled: CONNECTING)
```

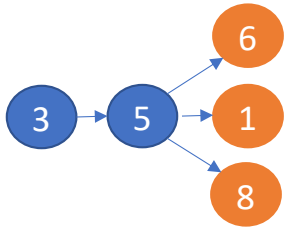
1

```
00000100:00080000:0.0:1583538583.632228:0:3384:0:(import.c:568:import_select_connection())
lustre-MDT0000-lwp-OST0002: tried all connections, increasing latency to 60s
```

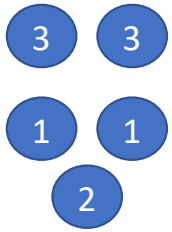
Existing Work: Three Different Ways

Log Index

DeepLog[1]



IM[2]



# of 1 = 2
# of 2 = 1
# of 3 = 2
Invariant:
# of 1 == # of 3

…

Log Level

```
Log1: lustre-MDT0000: transno 4295489106
is committed
Log2: can't lstripe objid
[0x200000402:0x93d5:0x0]: have 2 want 3
```

Look source code for Log Level

```
Log1: lustre-MDT0000: transno 4295489106
is committed DEBUG
Log2: can't lstripe objid
[0x200000402:0x93d5:0x0]: have 2 want 3
ERROR
```

Log Content

Search keyword,
e.g., “error”, “exception”.
Compare synonyms and antonyms,
e.g., LogAnomaly[3]

Can we go further?

[1] Deeplog: Anomaly detection and diagnosis from system logs through deep learning.
[2] Mining Invariants from Console Logs for System Problem Detection.
[3] LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs

Outline

1 Background

2 Challenge

3 SentiLog Design

4 Evaluation

5 Conclusion

Challenge 1: Difficult to build appropriate sessions

Lustre

```
00000100:00080000:0.0:1607448618.327577:0:2290:0:(recover.c:58:ptlrpc_initiate_recovery
()) lustre-OST0000_UUID: starting recovery

00000100:00080000:0.0:1607448618.327580:0:2290:0:(import.c:681:ptlrpc_connect_import())
ffffa139cab87800 lustre-OST0000_UUID: changing import state from DISCONN to CONNECTING

00000100:00080000:0.0:1607448618.327589:0:2290:0:(import.c:524:import_select_connection
()) lustre-OST0000-osc-MDT0000: connect to NID 10.0.0.8@tcp last attempt 4296114409

00000100:00080000:0.0:1607448618.327593:0:2290:0:(import.c:568:import_select_connection
()) lustre-OST0000-osc-MDT0000: tried all connections, increasing latency to 11s
```

HDFS

```
081109 203518 143 INFO dfs.DataNode$DataXceiver: Receiving block blk_-
1608999687919862906 src: /10.250.19.102:54106 dest: /10.250.19.102:50010

081109 203518 35 INFO dfs.FSNamesystem: BLOCK* NameSystem.allocateBlock:
/mnt/hadoop/mapred/system/job_200811092030_0001/job.jar. blk_-1608999687919862906

081109 203519 143 INFO dfs.DataNode$DataXceiver: Receiving block blk_-
1608999687919862906 src: /10.250.10.6:40524 dest: /10.250.10.6:50010

081109 203519 145 INFO dfs.DataNode$PacketResponder: PacketResponder 1 for block blk_-
1608999687919862906 terminating
```

Challenge 1: Difficult to build appropriate sessions

DeepLog:

Dependency: 

Session 1 

Session 2 

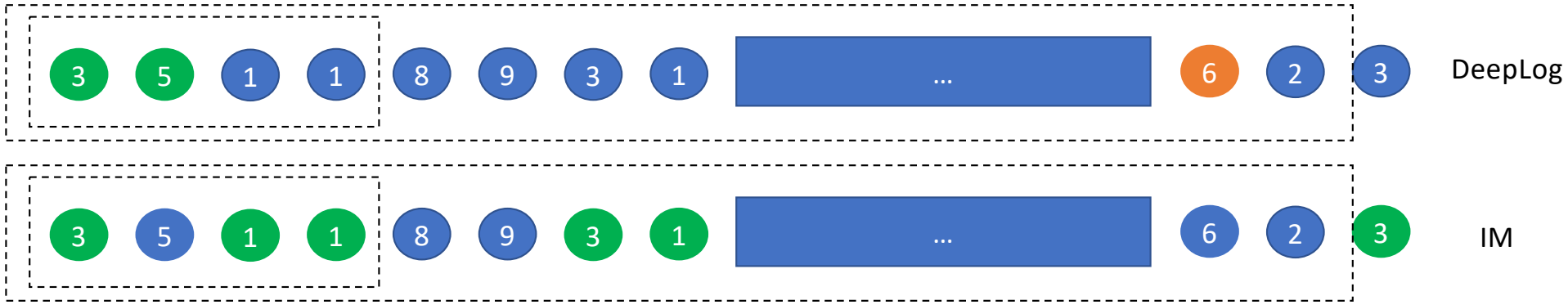
IM:

Invariant: # of  == # of 

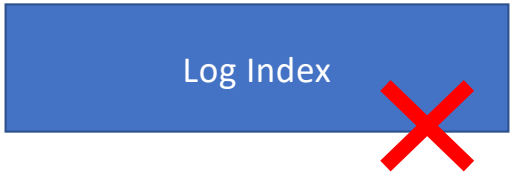
Session 1 

Session 2 

Challenge 1: Difficult to build appropriate sessions



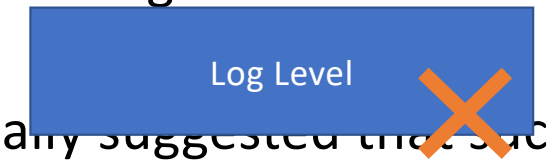
- Not able to build sessions based on their identifiers.
- Can only build sessions based on timestamps:
  - But, how to choose suitable time windows?
  - A small window may not include the relevant indices.
  - A large window have too many indices, which makes it difficult to discover the dependencies or invariants.



Challenge 2: Log level may be inaccurate

```
Log_CRITICAL:Dec14 23:06:03 Main [App] » BeeGFS Helper Daemon Version: 7.2
Log_WARNING:Dec15 16:12:27 Main [App] » LocalNode: beegfs-mgmtd osboxes [ID:1]
Log_WARNING:Dec15 15:58:37 Worker1 [Node registration] » New node: beegfs-client 435-5FD9237D-osboxes [ID: 2]; Source: 10.0.0.121:59206
```

- The three log entries above are simply reporting variable values, but they are labeled as ‘Warning’ or ‘Critical’ instead of normal by the developers.
- Previous study[1] actually suggested that such variable printing logs were reported as normal level in 95% of the time in multiple open-source software.



[1] DeepLV: Suggesting Log Levels Using Ordinal Based Neural Networks.

Challenge 3: Labeled data is difficult to obtain

PFS Cluster

Methods depending on labeled data:

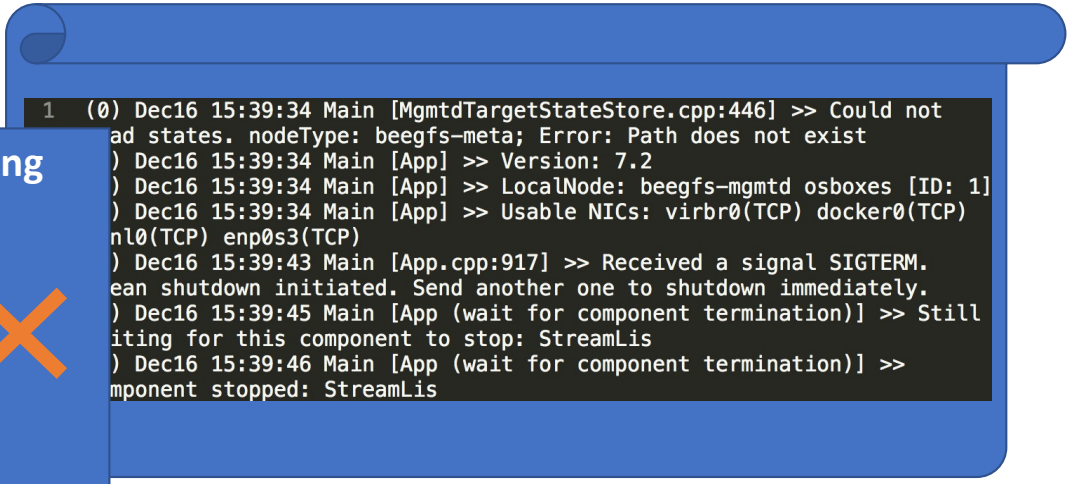
DeepLog

LogAnomaly

Decision Tree

SVM

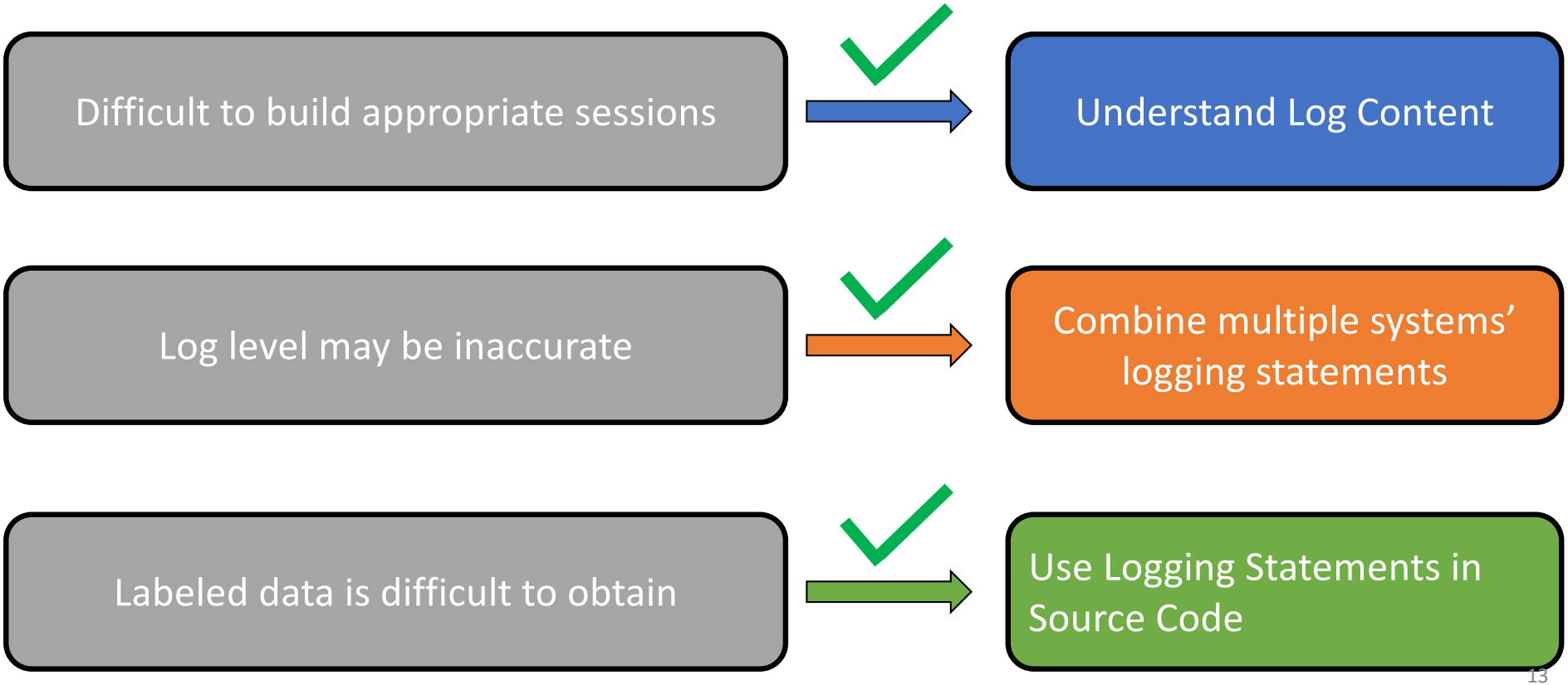
…



```
1  (0) Dec16 15:39:34 Main [MgmtdTargetStateStore.cpp:446] >> Could not
   ad states. nodeType: beegfs-meta; Error: Path does not exist
   ) Dec16 15:39:34 Main [App] >> Version: 7.2
   ) Dec16 15:39:34 Main [App] >> LocalNode: beegfs-mgmtd osboxes [ID: 1]
   ) Dec16 15:39:34 Main [App] >> Usable NICs: virbr0(TCP) docker0(TCP)
   nl0(TCP) enp0s3(TCP)
   ) Dec16 15:39:43 Main [App.cpp:917] >> Received a signal SIGTERM.
   ean shutdown initiated. Send another one to shutdown immediately.
   ) Dec16 15:39:45 Main [App (wait for component termination)] >> Still
   iting for this component to stop: StreamLis
   ) Dec16 15:39:46 Main [App (wait for component termination)] >>
   mponent stopped: StreamLis
```

- Are there any ano
- Which lines are associated with anomalies and which are not?

How does SentiLog solve these challenges?



Outline

1 Background

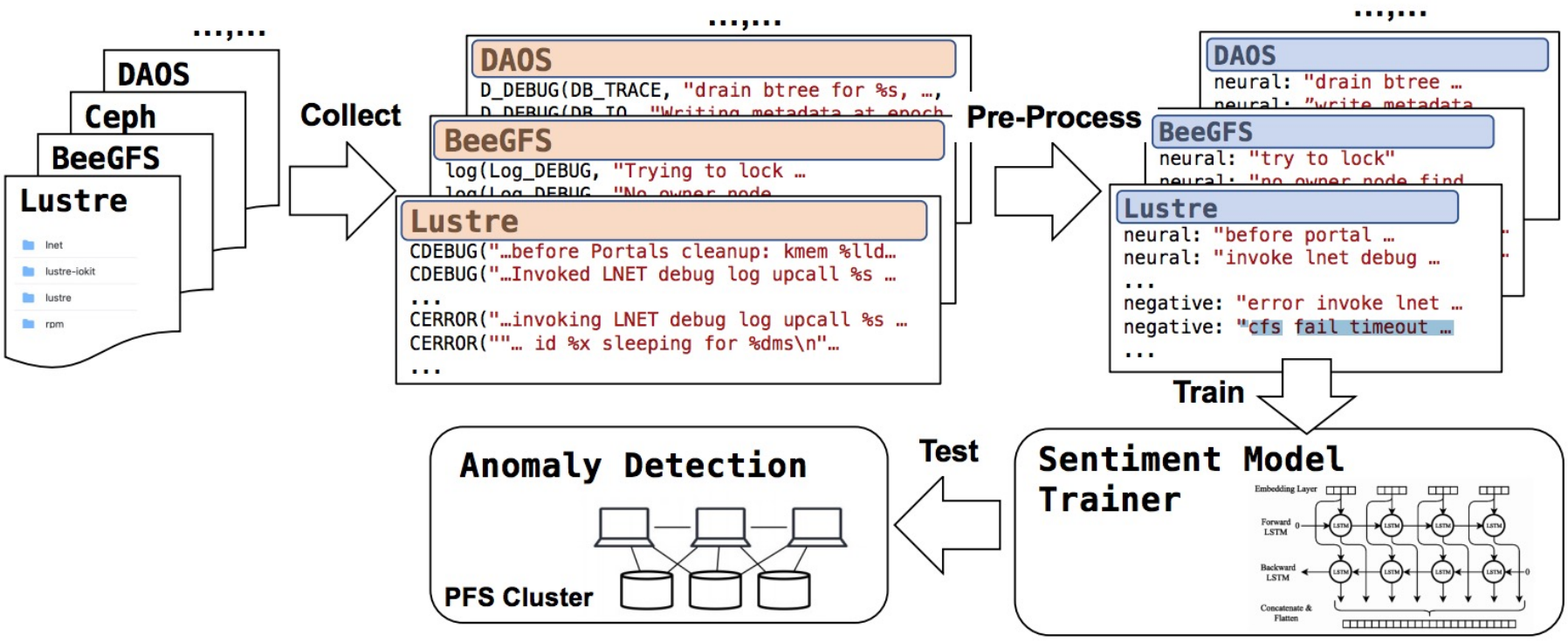
2 Challenge

3 SentiLog Design

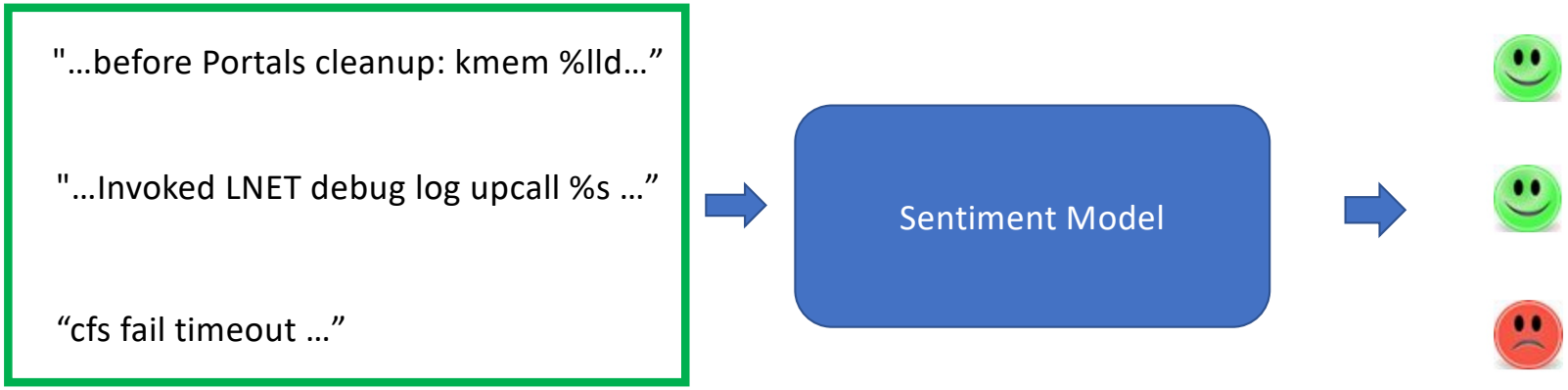
4 Evaluation

5 Conclusion

SentiLog Overview

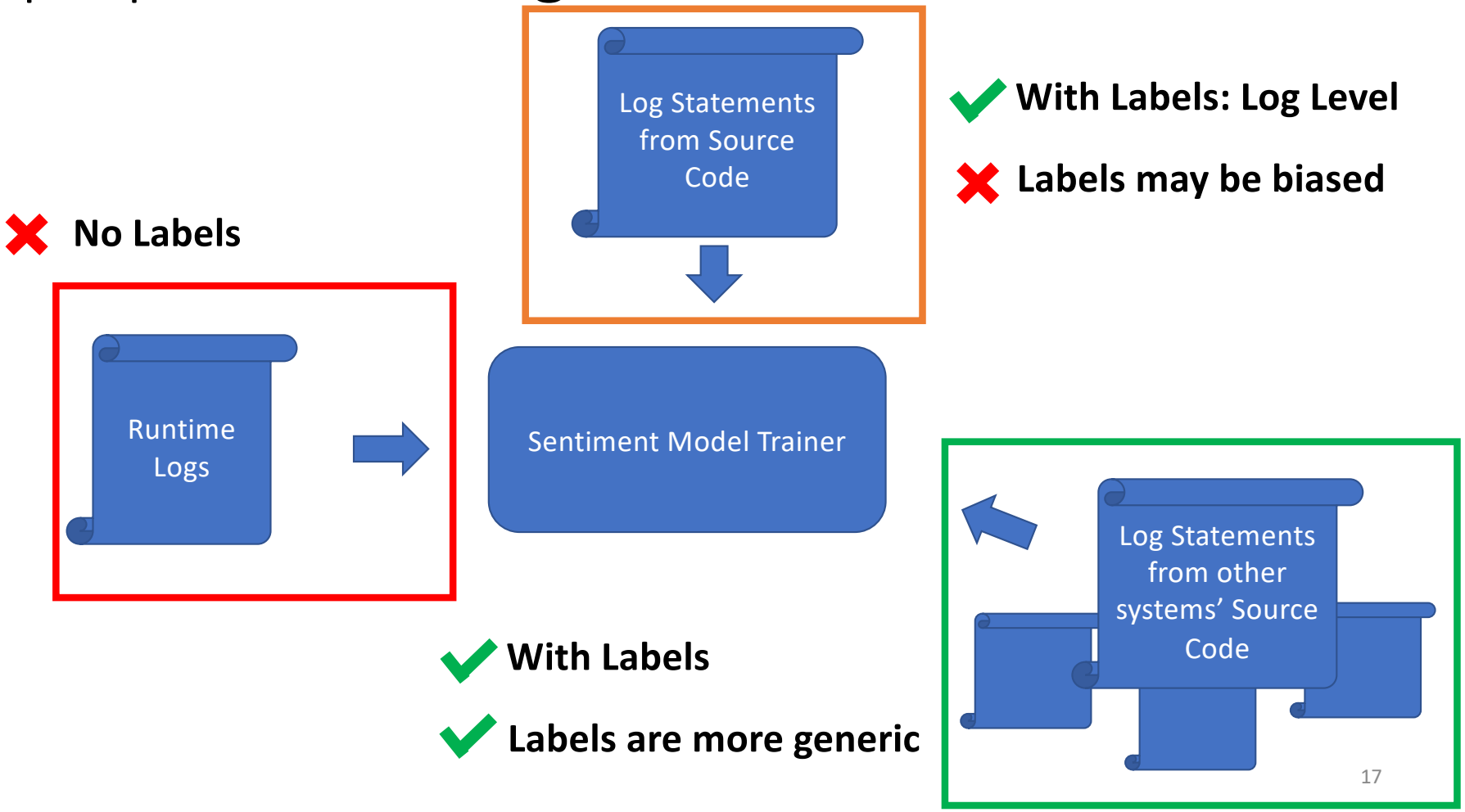


Sentiment Analysis

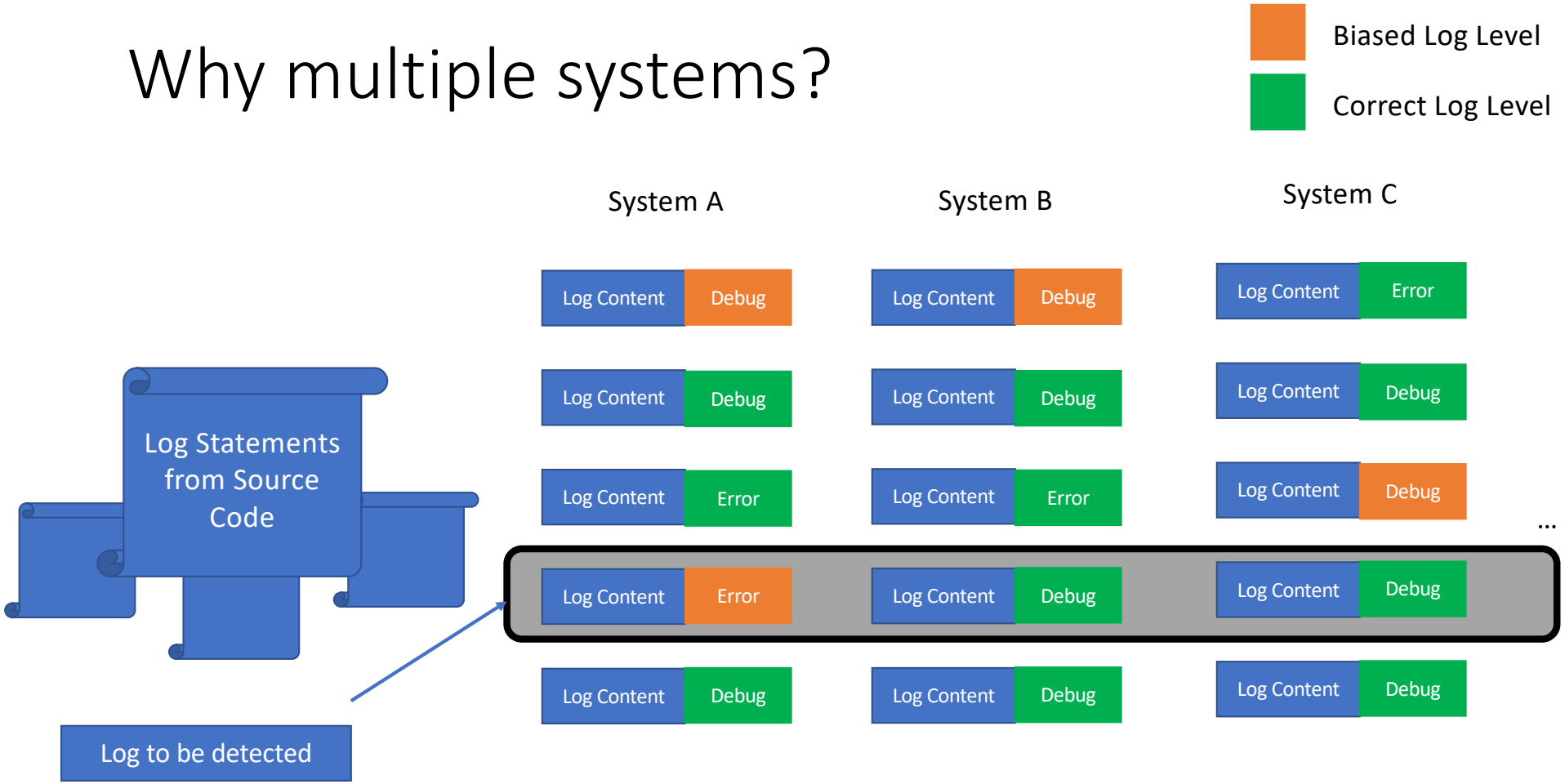


Dec14 22:54:24 Main [App] » Unable to create subdir: buddymir/inodes/C/60
Dec16 15:39:34 Main [MgmtdTargetStateStore.cpp:446] » Could not read states. node-Type: beegfs-meta; Error: Path does not exist

Appropriate Training Data



Why multiple systems?



Outline

1 Background

2 Challenge

3 SentiLog Design

4 Evaluation

5 Conclusion

Evaluation Setup

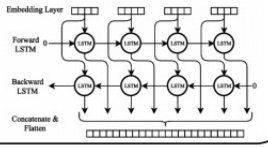
Log Source	Log Level		Log Mechanism
	Debug	Error	
OrangeFS [10]	1058	1202	gossip_debug, gossip_err,...
Ceph [7]	15459	2726	dout, derr,...
DAOS [9]	1549	3444	D_DEBUG, D_ERROR,...
GlusterFS [8]	2460	5260	gf_msg, gf_log,...

Pfault, ICS'18

Fault Model
Whole Device Failure (a-DevFail)
Global Inconsistency (b-Inconsist)
Network Partitioning (c-Network)

Train

Sentiment Model Trainer



Test

Fault Injection

Anomaly Detection

PFS Cluster

Lustre BeeGFS

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$$\text{Precision} = \frac{tp}{tp + fp}$$

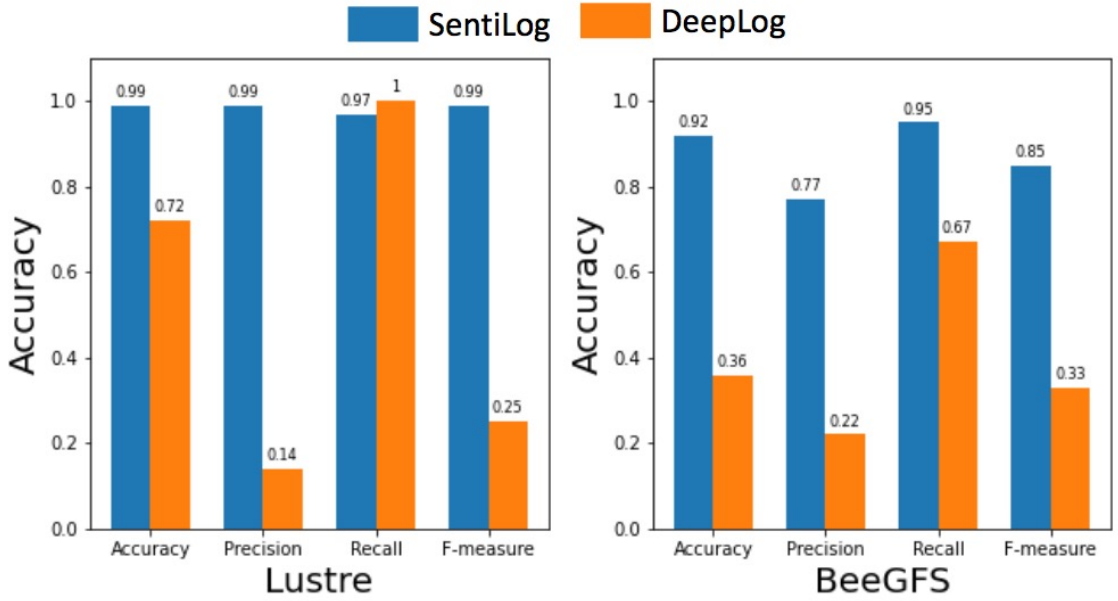
$$\text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

tp: true positive
tn: true negative
fp: false positive
fn: false negetive

Comparing with Existing Solutions

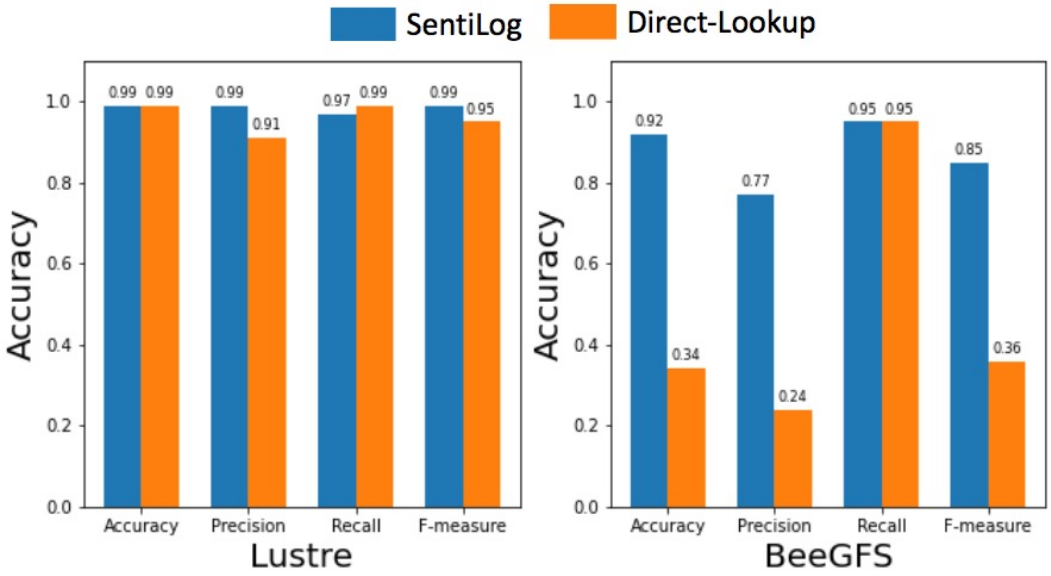
Log Content
vs.
Log Index



- Lack of sequence info in PFSes logs makes DeepLog not suitable.
- DeepLog has too many false positives.

Comparing with Direct-Lookup

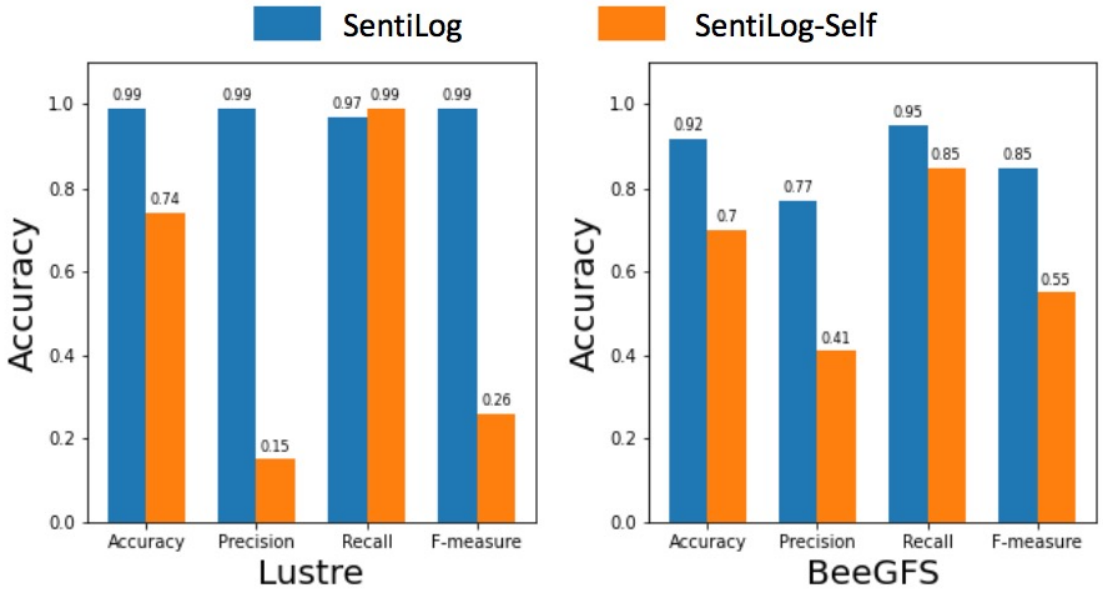
Log Content
vs.
Log Level



- Direct-Lookup: simply look up its corresponding logging statement in the source code and use its logging level to decide whether it is anomaly or not

Generality Evaluation

Multi source codes
vs.
Single source code



- SentiLog-Self: trained SentiLog using only the target PFS

Outline

1. Background
2. Challenge
3. SentiLog Design
4. Evaluation
5. Conclusion

Conclusion and Future Work

- Conclusion:
  - We propose to use sentimental analysis to understand log contents and detect the anomaly and show its effectiveness
  - We propose to train sentimental model using source code from multiple systems to solve the issue of lack of training data and to avoid bias of each system.
- Future Work:
  - Explore the possibility to consider more features besides the log statement description.
  - Conduct more experiments to validate and quantify the generic sentiment across different software.


usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION


HotStorage
2021
The 13th ACM Workshop on Hot Topics
In Storage and File Systems
July 27-28
Virtual

Q&A
Thank you!


UNC CHARLOTTE
College of Computing and Informatics


STATE