LogReducer: Identify and Reduce Log Hotspots in Kernel on the Fly

Guangba Yu[*†] Pengfei Chen[*] Pairui Li[†] Tianjun Weng[†]
Haibing Zheng[†] Yuetang Deng[†] Zibin Zheng[*]

* Sun Yat-Sen University

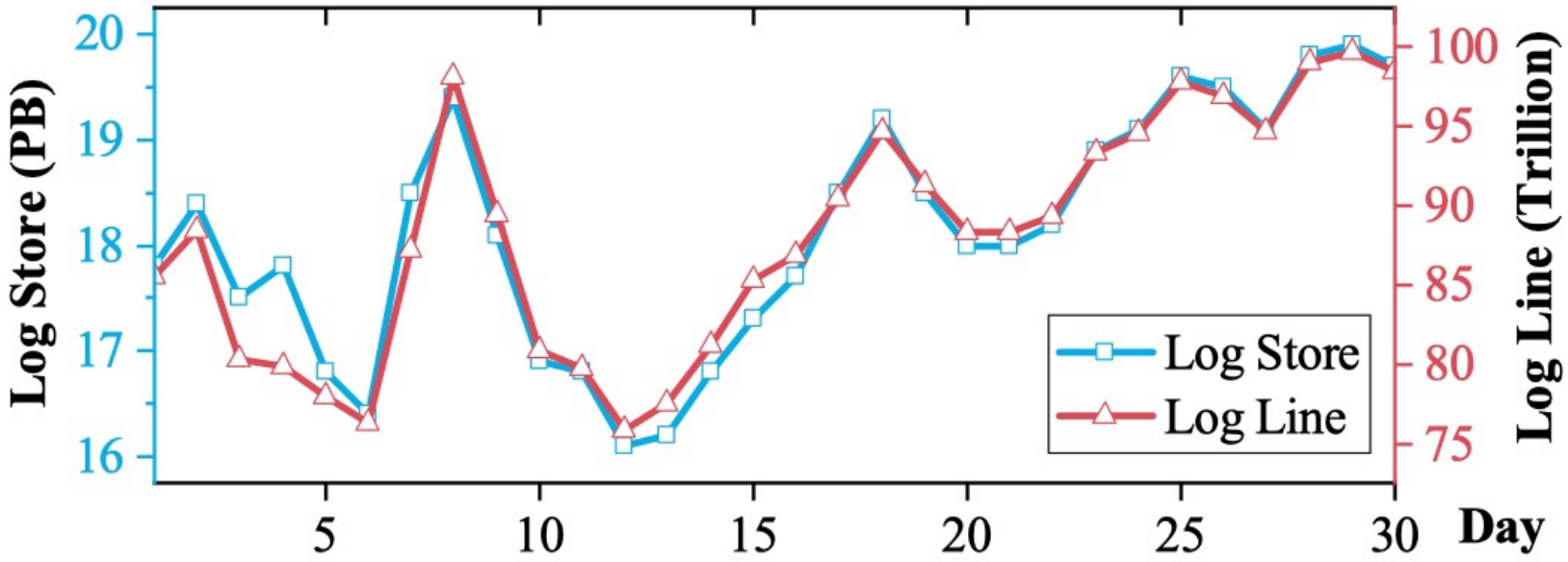
† Tencent

Introduction

- Logs is one of the “Three Pillars of Observability”
  - Check system status
  - Detect anomaly
  - Diagnose root cause

Introduction

- Logs is one of the “Three Pillars of Observability”
- Excessive logging is a double-edged sword
  - **High storage cost**
  - **Huge performance overhead**
  - **Difficult to troubleshoot**



Introduction

- How to reduce massive log amount?
  - Most of the logging overhead is due to **a very small number of log templates**
  - Log hotspot: **a small number of log templates occupy a lot of space**
  - Definition: S is the total storage of a service, and S_i is the storage of a log template

$$\frac{S_i}{S} > \xi$$

Log Format	< level> <service (pid,tid,cid,traceid)> time [code location] log info
Log Statement	MMERR("REQ %s Failed ", id);
Log Message	< 2> <Test (6,6,6,6666)> 11:48:43 66 [test.cpp:Test:6] REQ 6 Failed
Log Template	Error Test test.cpp:Test:6 REQ <ID> Failed

Introduction

- How to reduce massive log amount?
  - Most of the logging overhead is due to **a very small number of log templates**
  - Log hotspot: **A small number of log templates occupy a large amount of space**
  - Definition: S is the total storage of a service, and S_i is the storage of a log template

$$\frac{S_i}{S} > \xi$$

Reducing log hotspots is a cost-effective approach!

Empirical Study

- Research Object
  - WeChat is a large real-world instant messaging system serving billions of users
  - **Top 20 services** with the highest log storage in WeChat
  - **57 log hotspots** and interviewed 19 corresponding service owners
- Research Questions
  - RQ1: How do log hotspots impact **application storage**?
  - RQ2: How do log hotspots impact **application runtime**?
  - RQ3: What are the **root causes** of log hotspots?
  - RQ4: What are the **fixing solutions** of log hotspots?
  - RQ5: **How long** do developers take to fix log hotspots?

Empirical Study

- RQ1: How do log hotspots impact **application storage**?

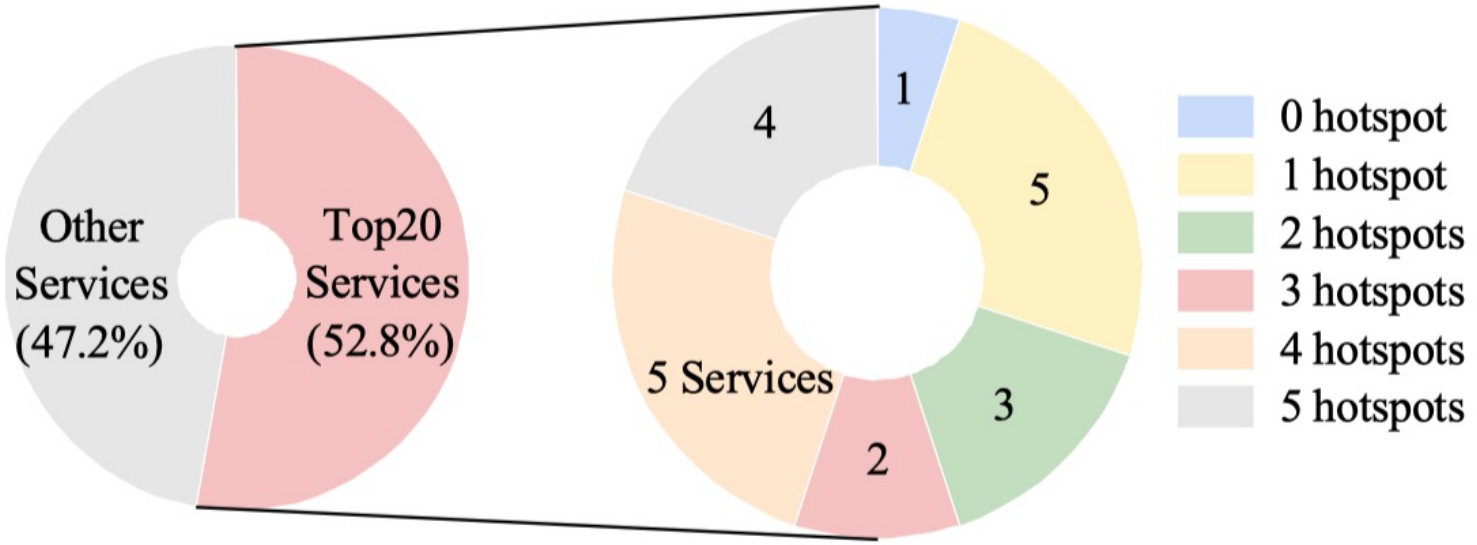


Fig. 5. The top 20 services with the highest storage in WeChat account for 52.8% of the total storage. 19 of the 20 services contain at least one log hotspots.

- ✓ **Top 20 services accounting for 52.8% of the total storage in Wechat**
- ✓ **19 services (19 out of 20) contain at least one log hotspots**

Empirical Study

➢ RQ1: How do log hotspots impact **application storage**?

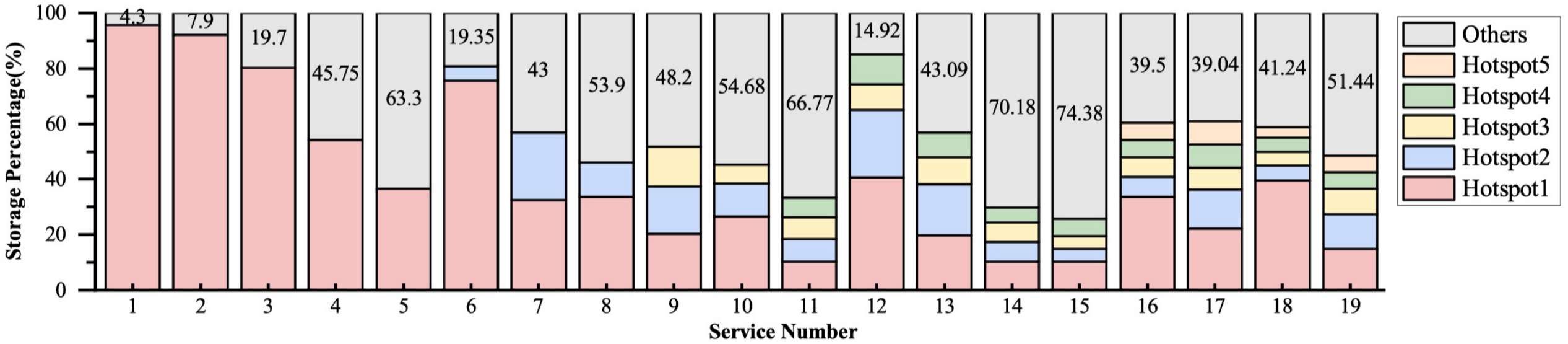


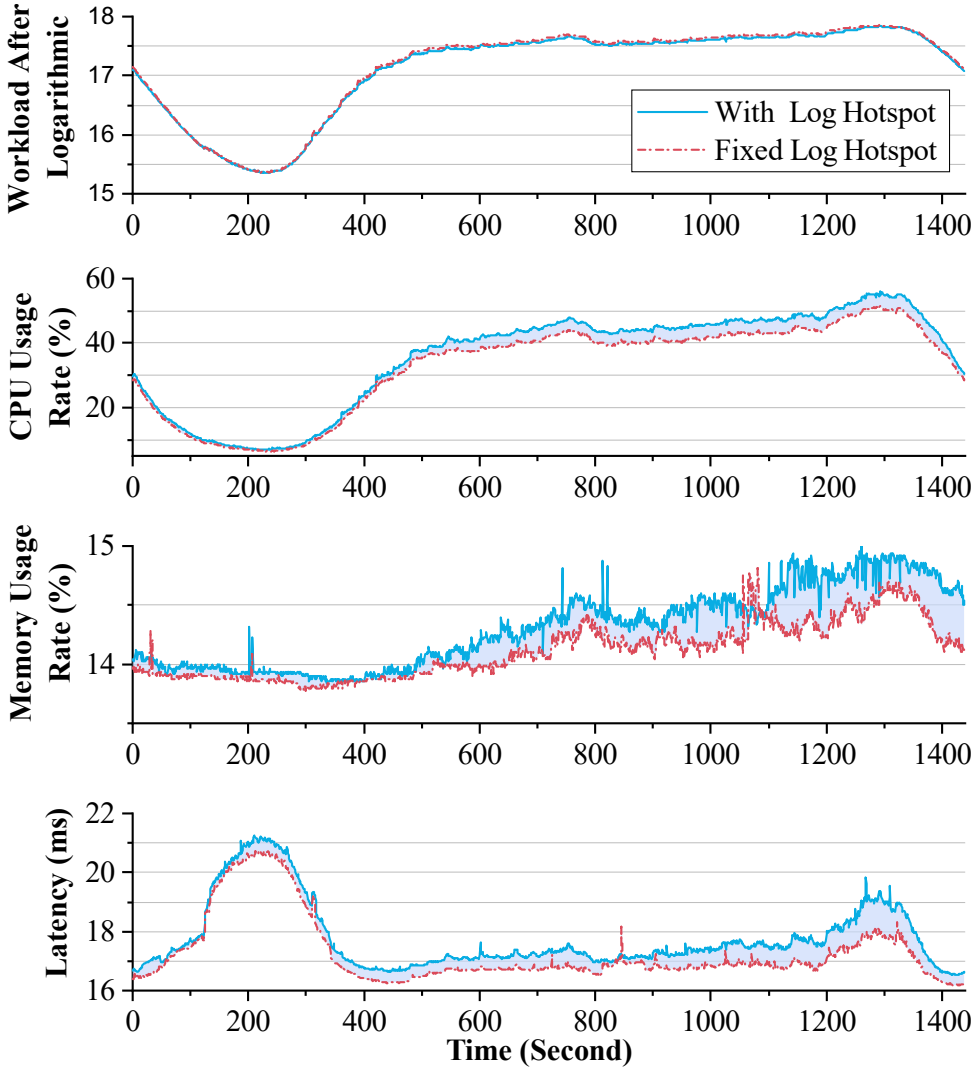
Fig. 6. For services containing at least one log hotspot in Figure 5, log hotspots occupy an average of 57.86% of the corresponding service storage.

✓ **Log hotspots occupy an average of 57.86% of corresponding storage**

✓ **One log hotspot occupied 95.7% of the storage for the service**

Empirical Study

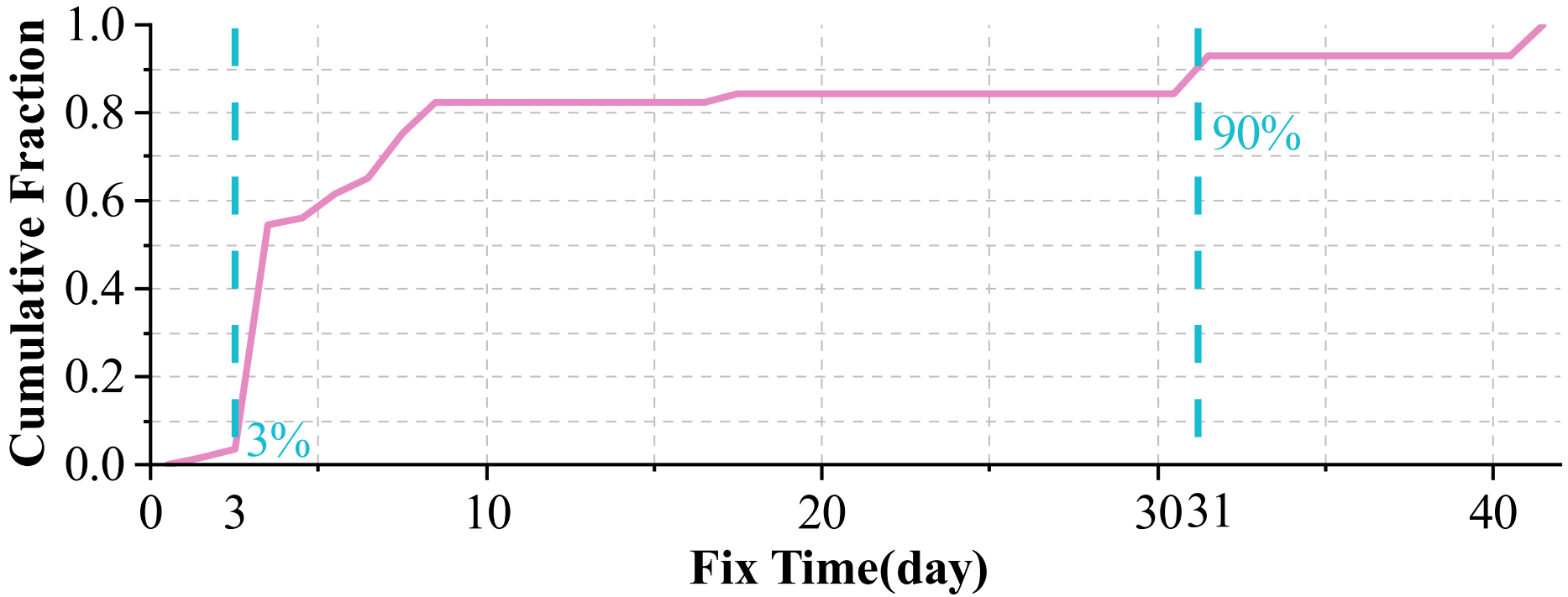
- RQ2: How do log hotspots impact **application runtime**?
  - New version only modifies the logging statements compared to the old version



- ✓ **Log volume dropped from 162 TB to 3.66 TB**
- ✓ **Service with log hotspots consumed up to 5.18% more CPU (58 cores in total)**
- ✓ **Service with log hotspots suffered up to 3% more response latency**

Empirical Study

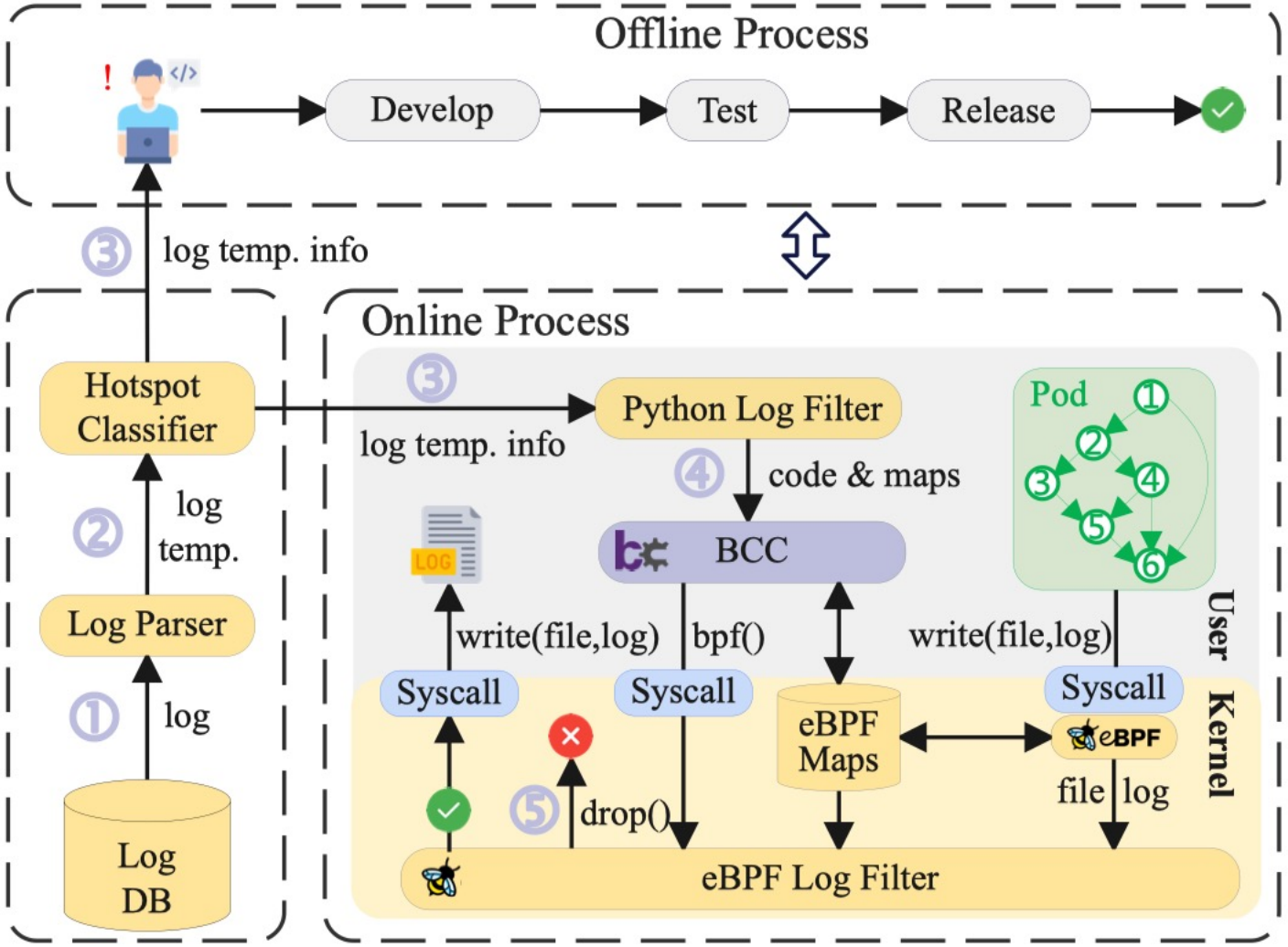
➢ RQ5: **How long** do developers take to fix log hotspots?



✓ **For 97% of log hotspots, developers need at least 3 days and on average 9 days to fix them**

LogReducer Approach

- LogReducer is a **non-intrusive** and **language-independent** framework for filtering log hotspots
  - **Offline Process**: alert developers to fix log hotspots
  - **Online Process**: filter log hotspots in the Linux kernel



LogReducer Offline Process

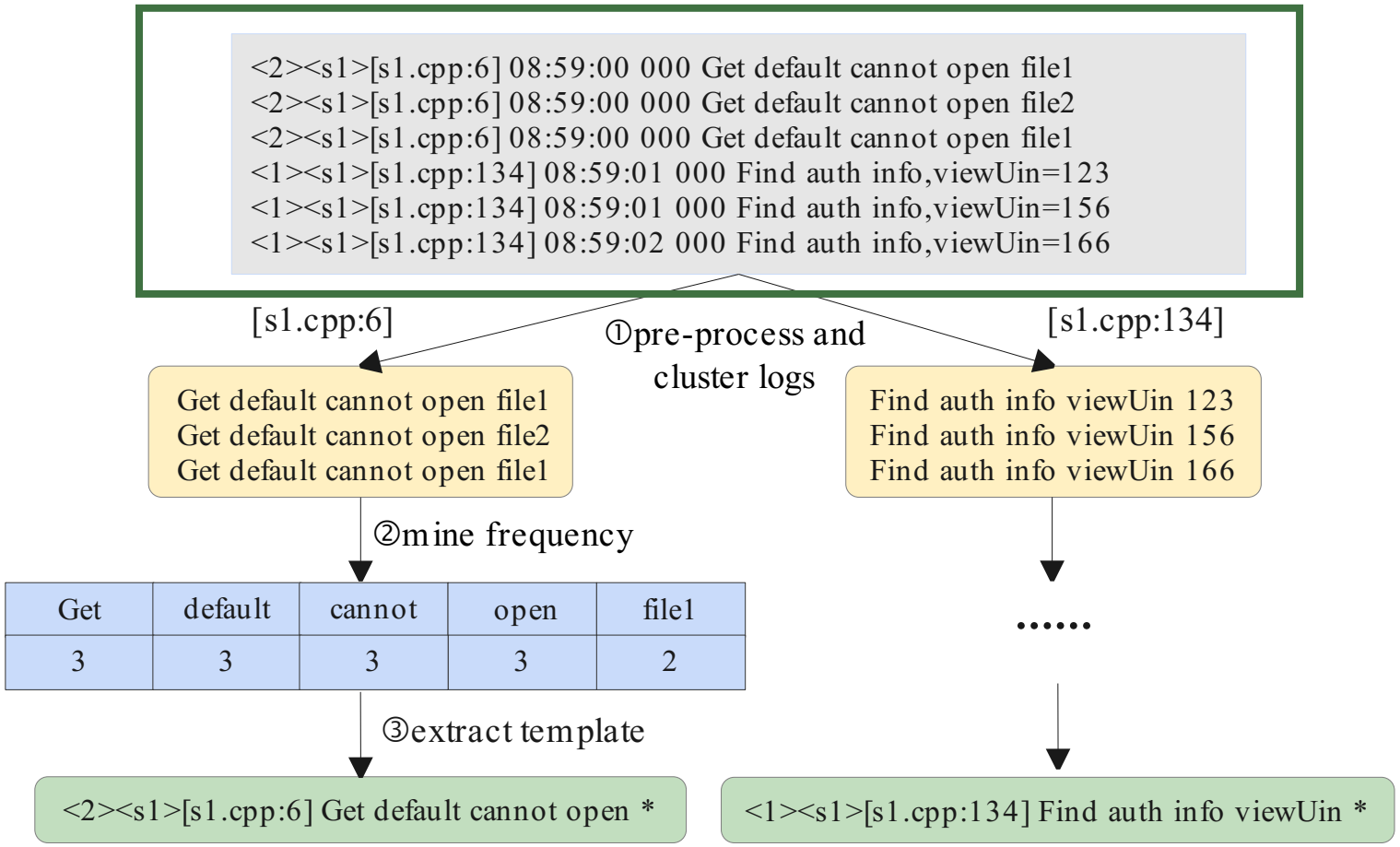
1. Log Parser: combine **log signature matching and frequency analysis** to parse logs
   - Log signature: strings that can uniquely identify a logging statement
   - We use **location of log statement** as log signature
     - ✓ Location of log statement **precisely bound** to log statement
     - ✓ **Insert location** into the log message based on log frameworks is **easy**

```
# Java Log 4j
<PatternLayout pattern="%d %-5p %c [%F:%L] - %m%n"/>

# C++ log4cplus
log4cplus::tstring pattern = LOG4CPLUS_TEXT("%D{%Y/%m/%d %H:%M:%S,%Q} %-5p
- [%l] %m %n");

# Golang Zap
logger := zap.New(core, zap.AddCaller())
```

LogReducer Offline Process

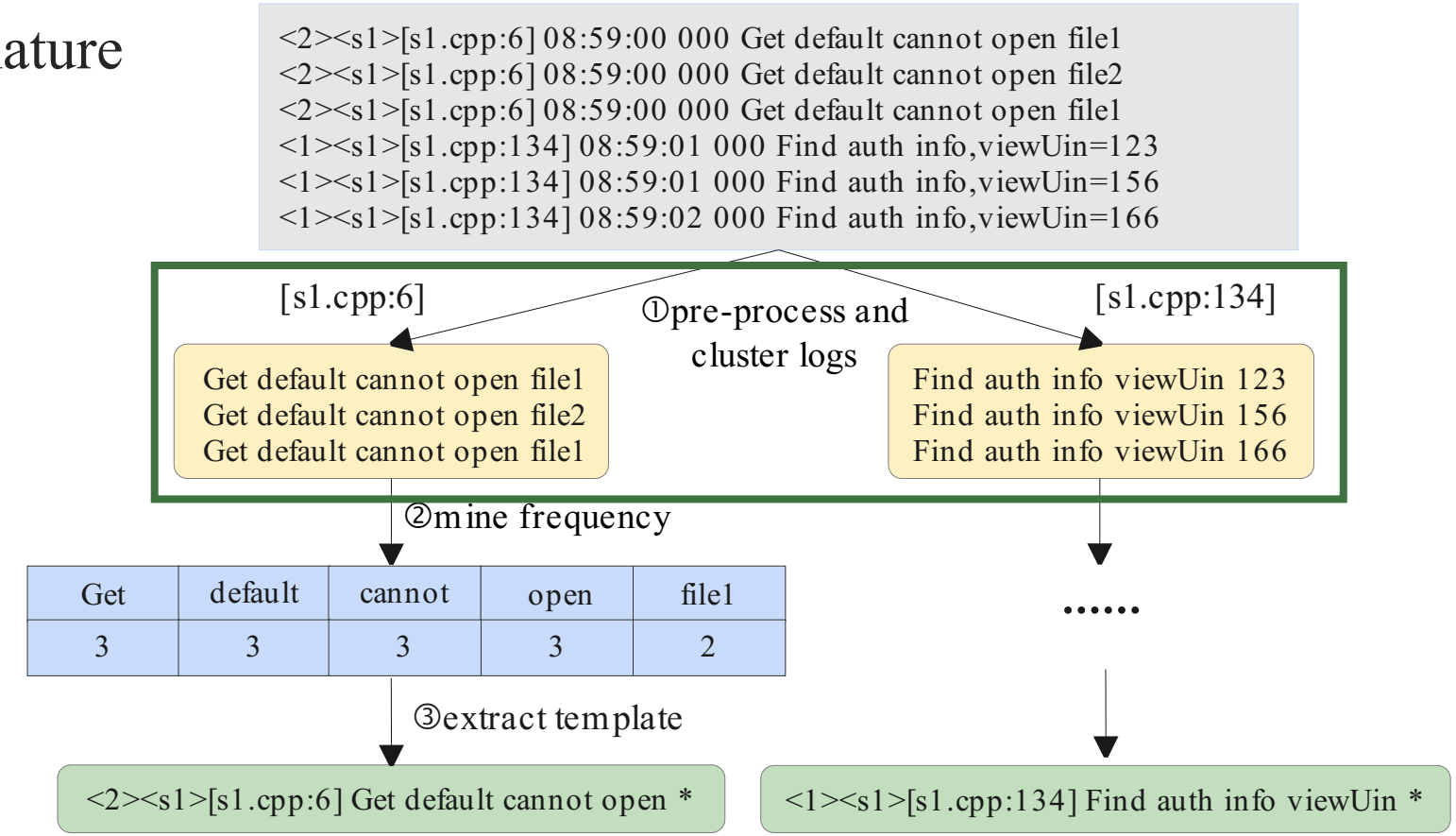
1. Log Parser: combine **log signature matching and frequency analysis** to parse logs

   ① Extract the **log level, service name, and log signature**



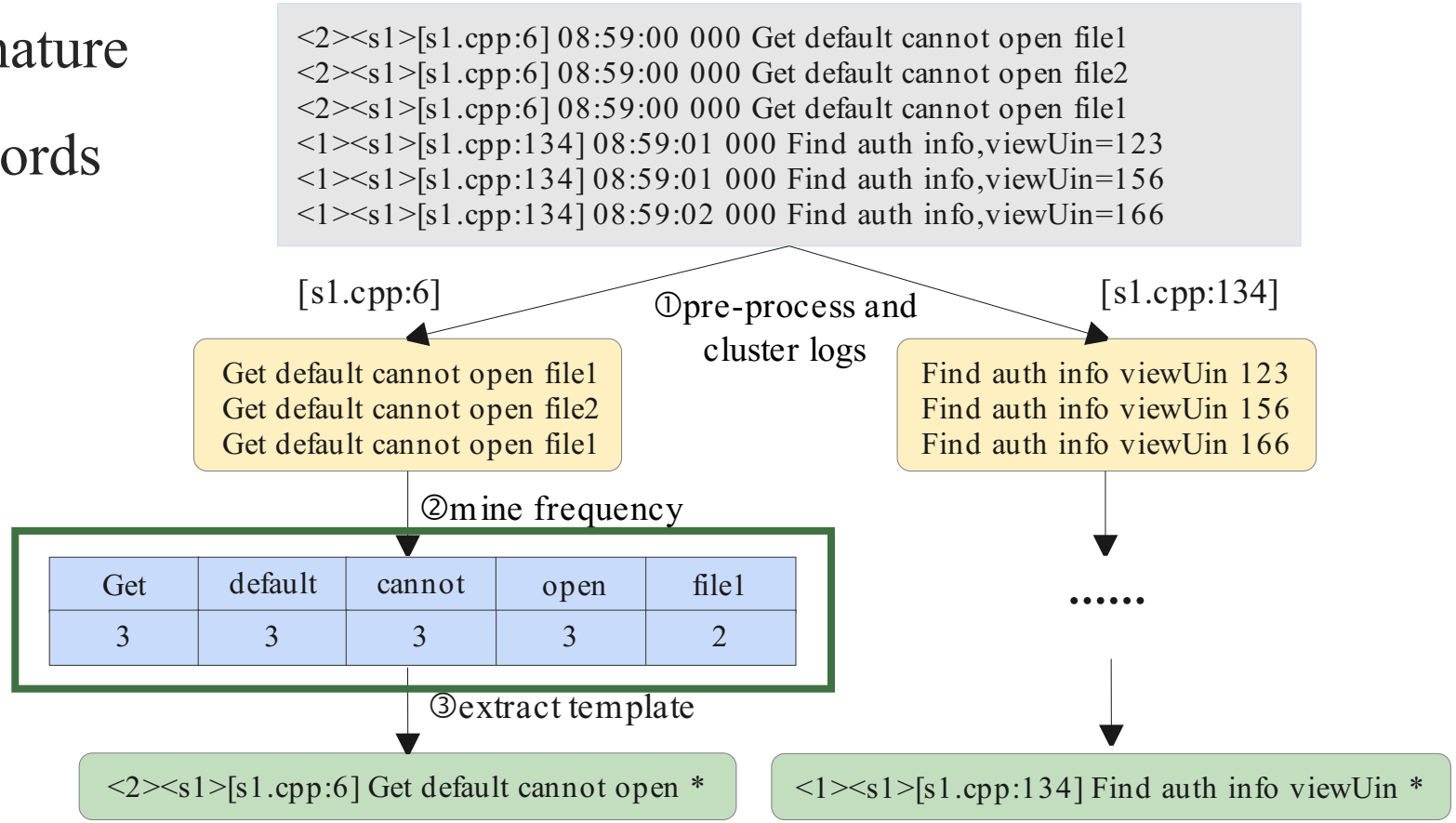
LogReducer Offline Process

1. Log Parser: combine log signature matching and frequency analysis to parse logs
   - ① Extract log level, service name, and log signature
   - ② **Cluster** logs based on log signature



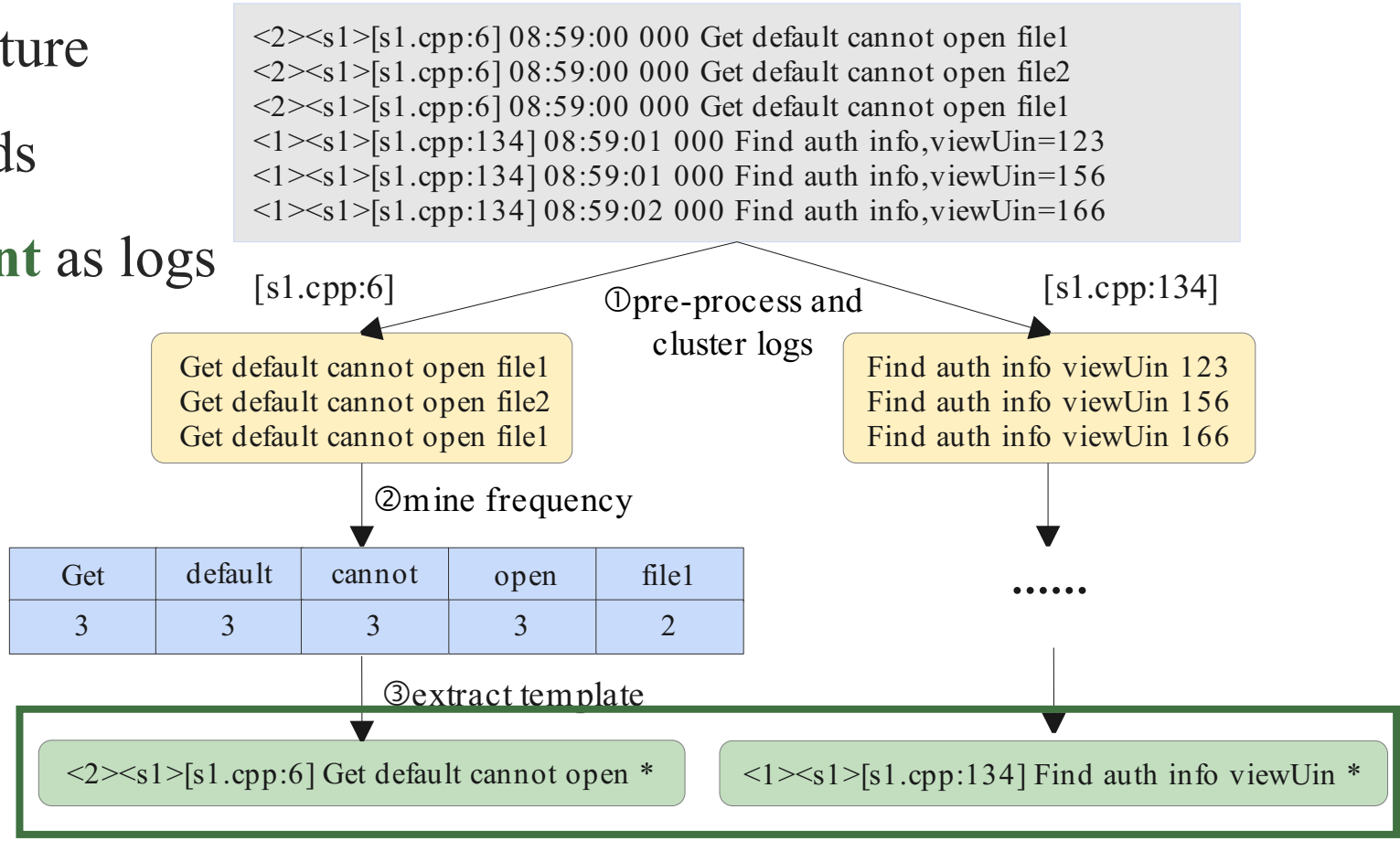
LogReducer Offline Process

1. Log Parser: combine log signature matching and frequency analysis to parse logs
   - ① Extract the log level, service name, and log signature
   - ② Cluster logs based on log signature
   - ③ Build a **frequency table** of words



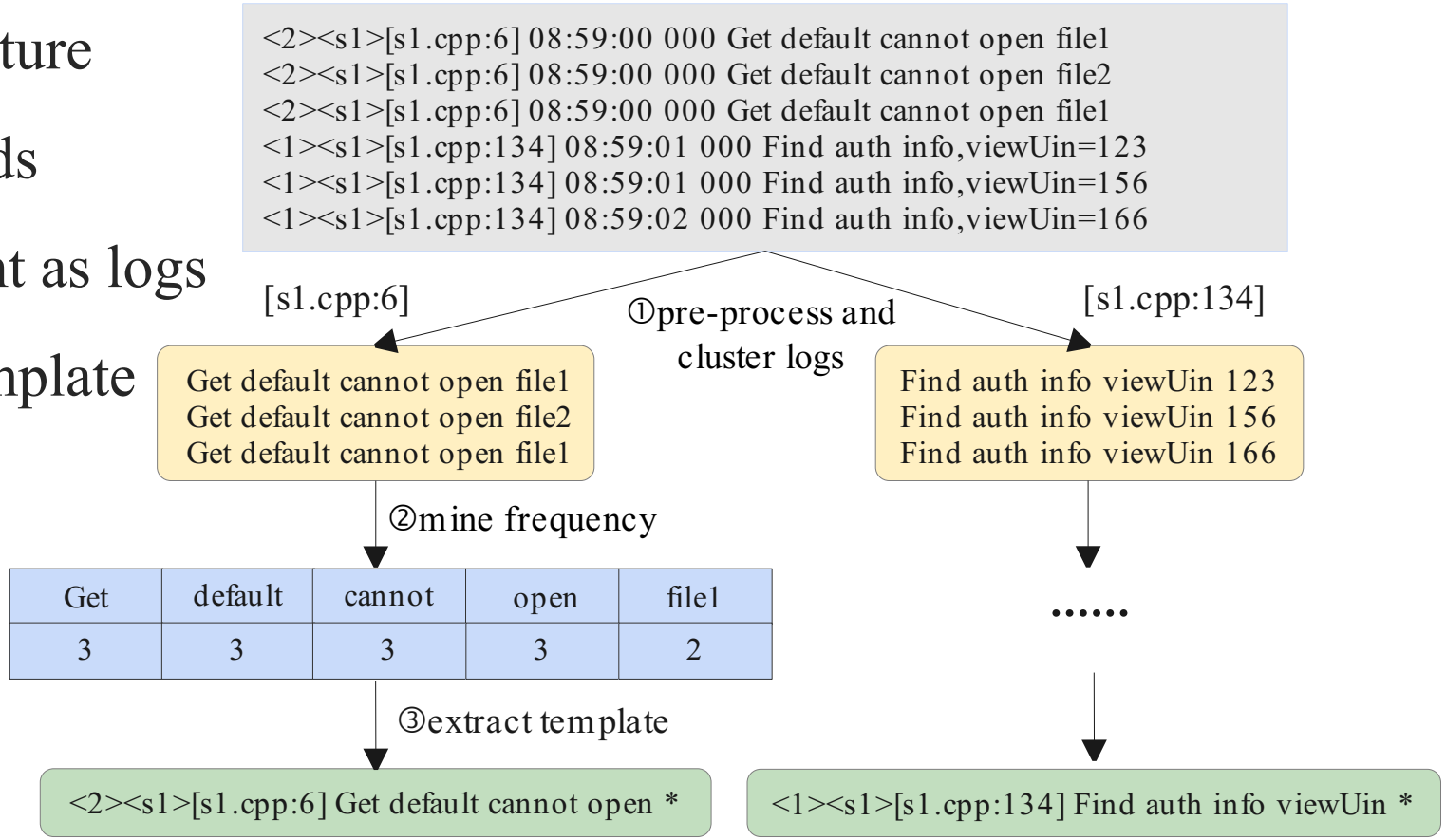
LogReducer Offline Process

1. Log Parser: combine log signature matching and frequency analysis to parse logs
   - ① Extract the log level, service name, and log signature
   - ② Cluster logs based on log signature
   - ③ Build a frequency table of words
   - ④ Seek words with the **same count** as logs



LogReducer Offline Process

1. Log Parser: combine log signature matching and frequency analysis to parse logs
   - ① Extract the log level, service name, and log signature
   - ② Cluster logs based on log signature
   - ③ Build a frequency table of words
   - ④ Seek words with the same count as logs
   - ⑤ **Obtain storage** of each log template



LogReducer Offline Process

1. Log Parser: combine log signature matching and frequency analysis to parse logs
2. Hotspot Classifier:

   ① Identify log hotspots from all log templates

$$\frac{S_i}{S} > \xi \qquad \xi = 0.05$$

   ② Trigger log reduction process if log hotspots exist

LogReducer Offline Process

1. Log Parser: combine log signature matching and frequency analysis to parse logs
2. Hotspot Classifier: Identify log hotspots from all log templates
3. **Alert developers** with log hotspots

Log Reducer

Date	OssID	Level	Template	Location	Storage	Storage Ratic	Rows	Rows Ratio	Log Length	Log detail
2023-04-10	19118	ERROR	new S[redacted]cene ...	[redacted]essionBuffer:3594	45.7 TiB	4.43%	14.3 Bil	0.389%	3672	detail
2023-04-10	19118	IMPT	ret_fla[redacted]ast_s...	[redacted]ess:3316	41.2 TiB	3.99%	13.9 Bil	0.377%	3411	detail
2023-04-10	19118	ERROR	CALL [redacted]ionKe...	[redacted]277	38.6 TiB	3.73%	10.1 Bil	0.274%	4391	detail
2023-04-10	19118	ERROR	DBG r[redacted]ot_rev...	[redacted]kingReqCtxAdapterV2:1141	34.0 TiB	3.29%	10.9 Bil	0.296%	3581	detail
2023-04-10	19118	IMPT	ERR T[redacted]	[redacted]	29.8 TiB	2.88%	190 Bil	5.17%	179	detail
2023-04-10	19118	IMPT	CALL [redacted]	[redacted]s:38	29.1 TiB	2.82%	8.58 Bil	0.233%	3890	detail
2023-04-10	19118	ERROR	resp B[redacted]	[redacted]Process:97	25.5 TiB	2.47%	8.60 Bil	0.234%	3397	detail
2023-04-10	19118	ERROR	resp B[redacted]	[redacted]s:44	25.5 TiB	2.47%	8.61 Bil	0.234%	3395	detail
2023-04-10	19118	ERROR	DBG r[redacted]ss_sa...	[redacted]kingReqCtxAdapterV2:1107	22.8 TiB	2.21%	10.9 Bil	0.297%	2402	detail
2023-04-10	19118	IMPT	MMFI[redacted]clipo...	[redacted]Log:7223	18.7 TiB	1.81%	13.9 Bil	0.379%	1535	detail

LogReducer Offline Process

1. Log Parser: combine log signature matching and frequency analysis to parse logs
2. Hotspot Classifier: Identify log hotspots from all log templates
3. Alert developers with log hotspots
4. Developers release new version

Log Reducer

Date	OssID	Level	Template	Location	Storage	Storage Ratic	Rows	Rows Ratio	Log Length	Log detail
2023-04-10	19118	ERROR	new S[redacted]cene ...	[redacted]essionBuffer:3594	45.7 TiB	4.43%	14.3 Bil	0.389%	3672	detail
2023-04-10	19118	IMPT	ret_fla[redacted]ast_s...	[redacted]ess:3316	41.2 TiB	3.99%	13.9 Bil	0.377%	3411	detail
2023-04-10	19118	ERROR	CALL [redacted]ionKe...	[redacted]277	38.6 TiB	3.73%	10.1 Bil	0.274%	4391	detail
2023-04-10	19118	ERROR	DBG r[redacted]t_rev...	[redacted]kingReqCtxAdapterV2:1141	34.0 TiB	3.29%	10.9 Bil	0.296%	3581	detail
2023-04-10	19118	IMPT	ERR T[redacted]	[redacted]	29.8 TiB	2.88%	190 Bil	5.17%	179	detail
2023-04-10	19118	IMPT	CALL [redacted]	[redacted]s:38	29.1 TiB	2.82%	8.58 Bil	0.233%	3890	detail
2023-04-10	19118	ERROR	resp B[redacted]	[redacted]Process:97	25.5 TiB	2.47%	8.60 Bil	0.234%	3397	detail
2023-04-10	19118	ERROR	resp B[redacted]	[redacted]s:44	25.5 TiB	2.47%	8.61 Bil	0.234%	3395	detail
2023-04-10	19118	ERROR	DBG r[redacted]ss_sa...	[redacted]kingReqCtxAdapterV2:1107	22.8 TiB	2.21%	10.9 Bil	0.297%	2402	detail
2023-04-10	19118	IMPT	MMFI[redacted]clipo...	[redacted]Log:7223	18.7 TiB	1.81%	13.9 Bil	0.379%	1535	detail

LogReducer Offline Process Result

- We applied LogReducer offline process in WeChat since April 14, 2022
- Log storage in WeChat dropped from **19.7 PB to 12.0 PB** per day

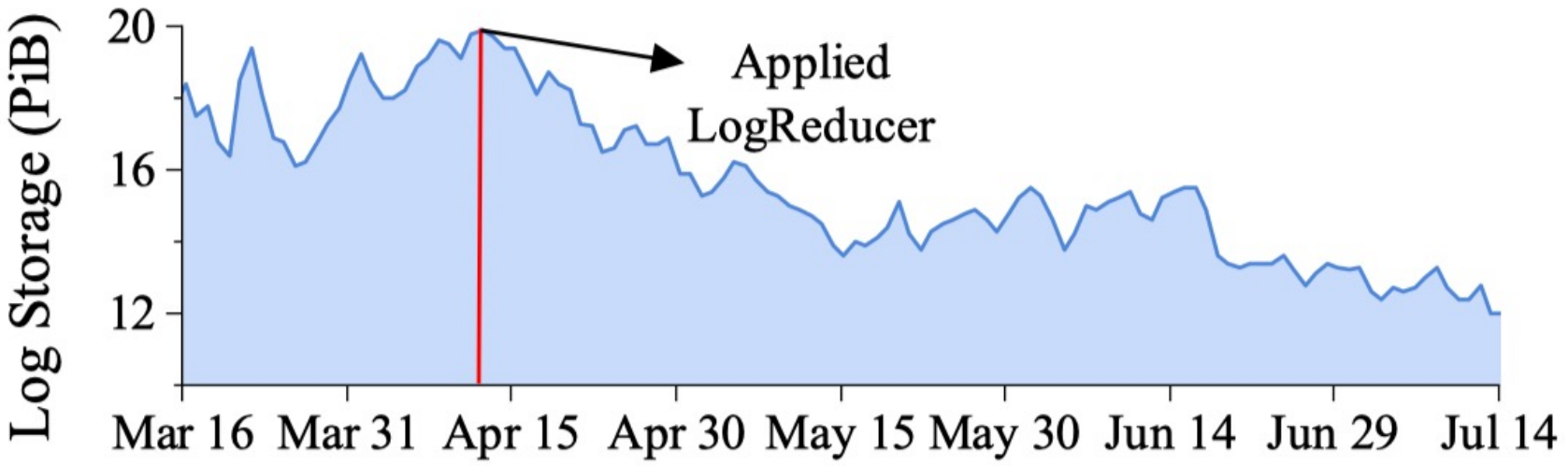
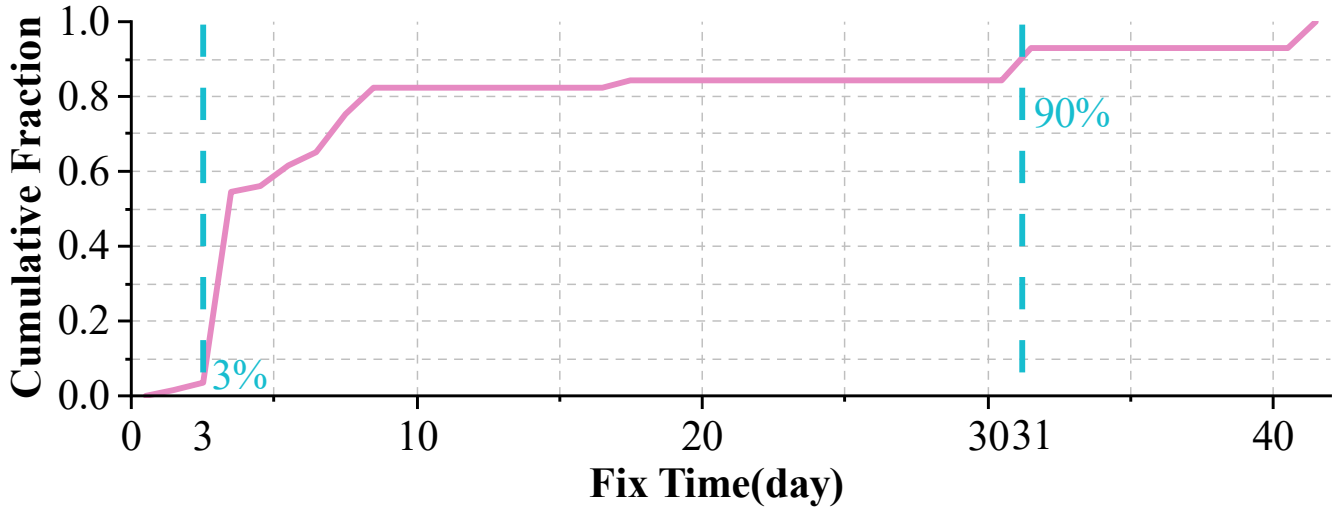


Fig. 17. Changes in the log storage of WeChat from Mar 16 to Jul 14 in 2022.

39.08% decrease

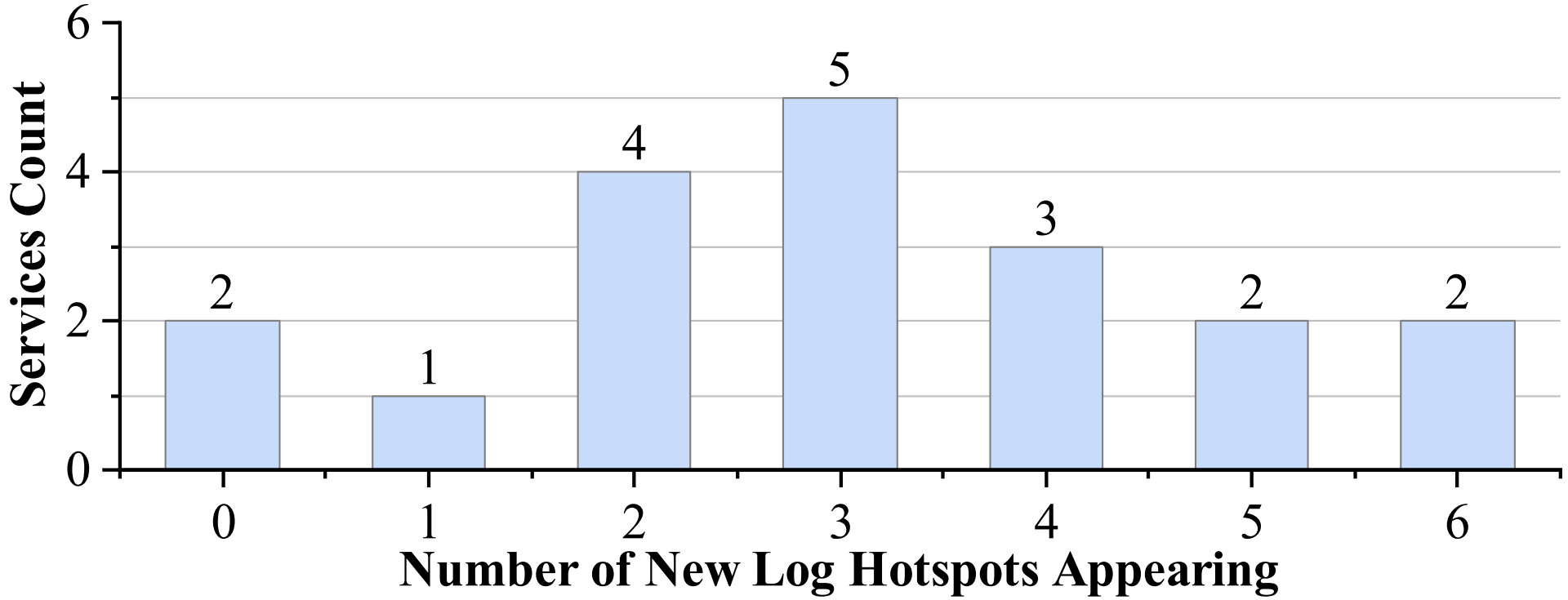
LogReducer Online Process

- Limitation of offline process
  - Fixing log hotspots **takes a long time**
    - ✓ Developers need **an average of 9 days** to fix log hotspots
    - ✓ Release new version until next new feature release
    - ✓ Canary change takes a long time
    - ✓ Holiday effect



LogReducer Online Process

- Limitation of offline process
  - Fixing log hotspots takes a long time
  - **New log hotspots appear frequently**



LogReducer Online Process

- Services are affected by log hotspots **until new version is released**
- Can we **avoid influence** of log hotspots before fixing them?

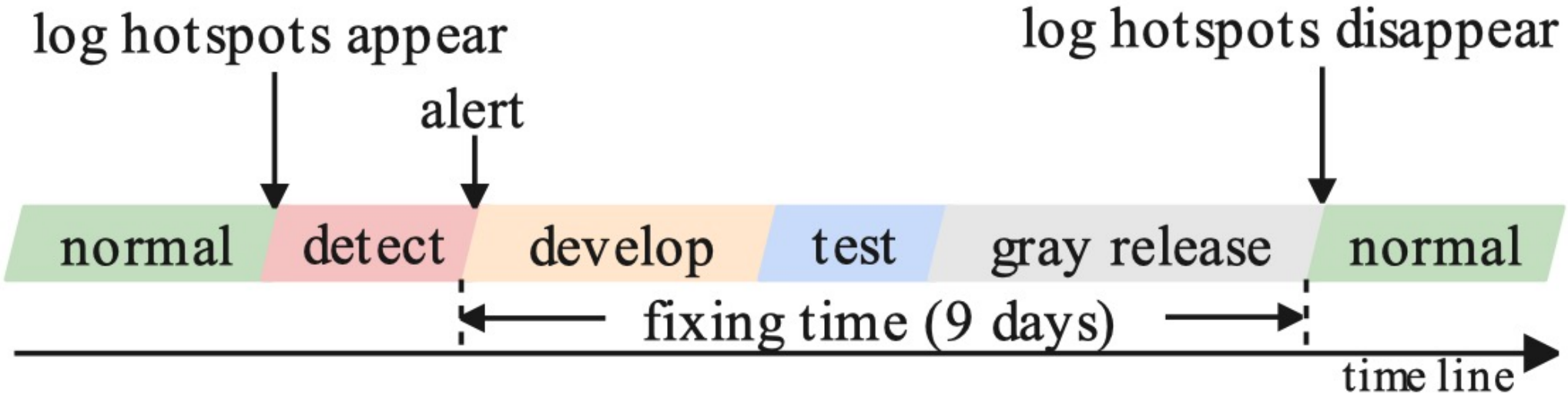
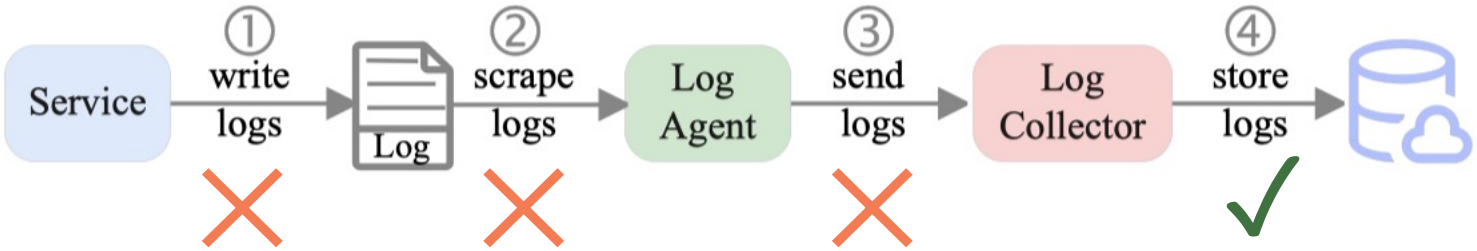


Fig. 2. Process of log hotspot identification and reduction.

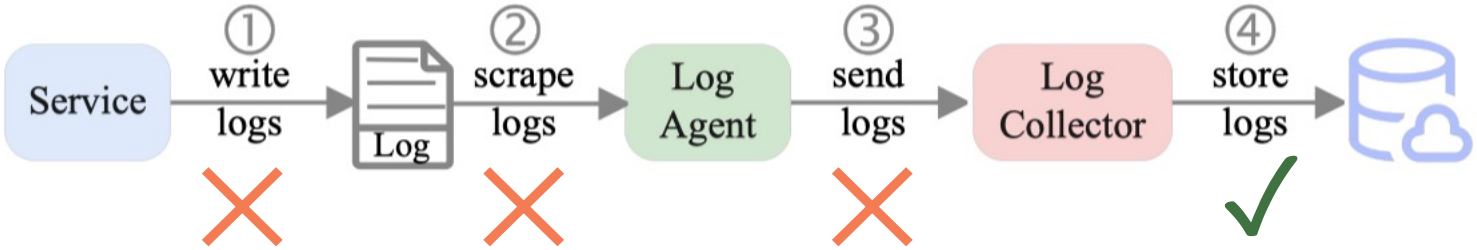
LogReducer Online Process

- Can we avoid the impact of log hotspots before fixing them?
  - Filter log hotspots in **log collector**
    - Reduce storage costs
    - **Cannot reduce printing overhead**
    - **Cannot reduce network transmission overhead**



LogReducer Online Process

- Can we avoid the impact of log hotspots before fixing them?
  - Filter log hotspots in **log collector**
    - Reduce storage costs
    - **Cannot reduce printing overhead**
    - **Cannot reduce network transmission overhead**

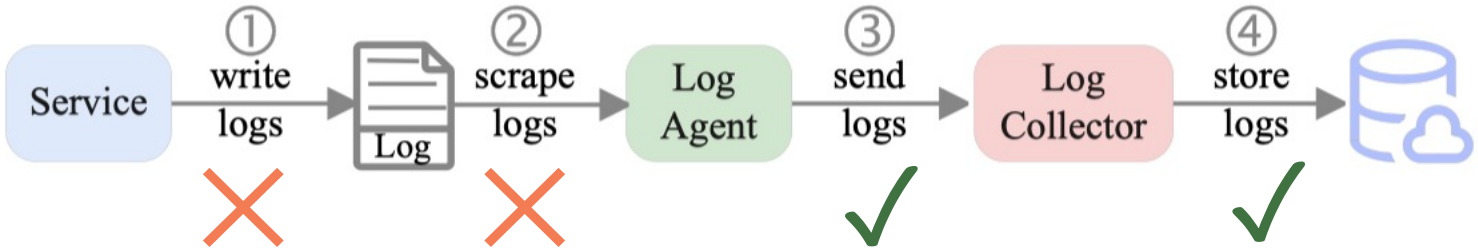


LogReducer Online Process

- Can we avoid the impact of log hotspots before fixing them?
  - Filter log hotspots in log collector
  - Filter log hotspots in **log agent**
    - Filter logs by configuring log agent
    - Reduce storage costs
    - Reduce network transmission overhead
    - **Cannot reduce printing overhead**
    - **Inefficient matching logs in user space**

```
output.elasticsearch:
  hosts: ['host:port']
  username: user
  password: pwd
  indices:
    - index: "filebeat-log-%{+yyyyMMdd}"
      when.regexp:
          msg: "foo.*"
```

Regular expression of log in Filebeat

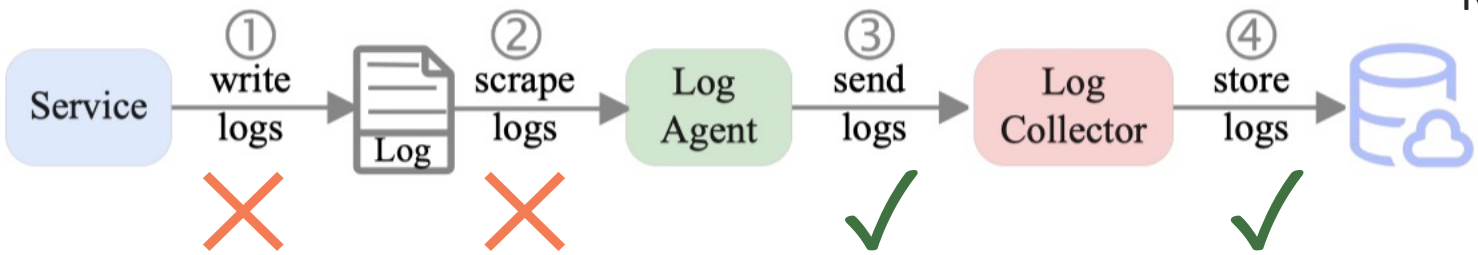


LogReducer Online Process

- Can we avoid the impact of log hotspots before fixing them?
  - Filter log hotspots in log collector
  - Filter log hotspots in **log agent**
    - Filter logs by configuring log agent
    - Reduce storage costs
    - Reduce network transmission overhead
    - **Cannot reduce printing overhead**
    - **Inefficient matching logs in user space**

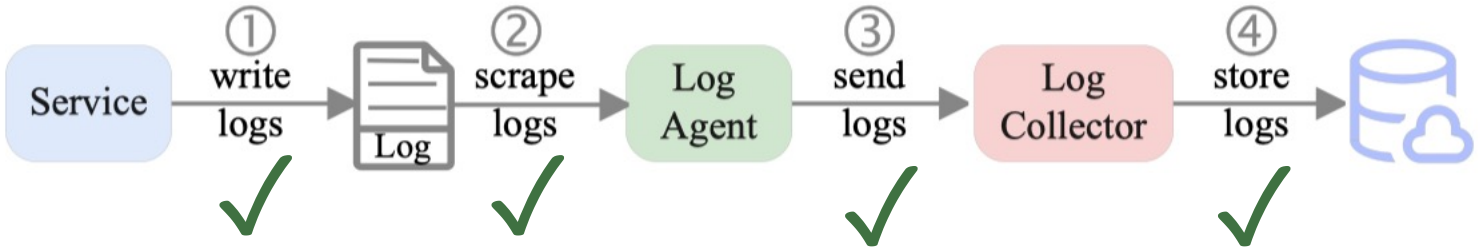
```
output.elasticsearch:
  hosts: ['host:port']
  username: user
  password: pwd
  indices:
    - index: "filebeat-log-%{+yyyyMMdd}"
      when.regexp:
          msg: "foo.*"
```

Regular expression of log in Filebeat



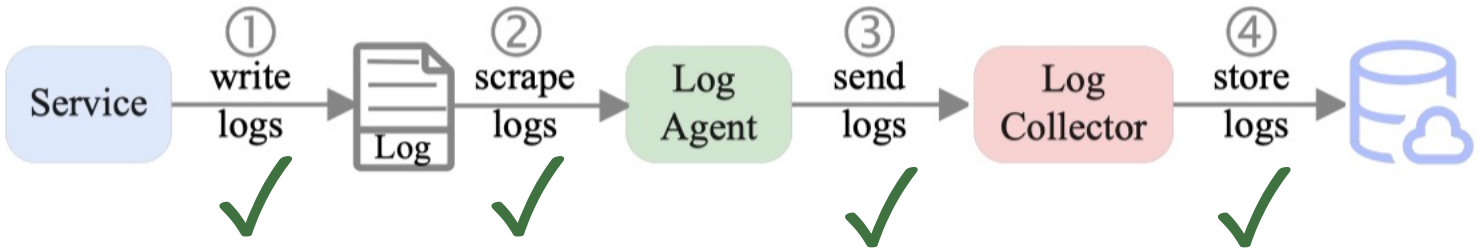
LogReducer Online Process

- Can we reduce the impact of log hotspots before fixing them?
  - Filter log hotspots in log collector
  - Filter log hotspots in log agent
  - Filter log hotspots **in the Linux kernel**
    - Reduce **storage costs and network transmission overhead**
    - Reduce **printing overhead**
    - **Reduce user and kernel switching, efficient kernel matching of logs**
    - **Non-intrusive, programming language independent**



LogReducer Online Process

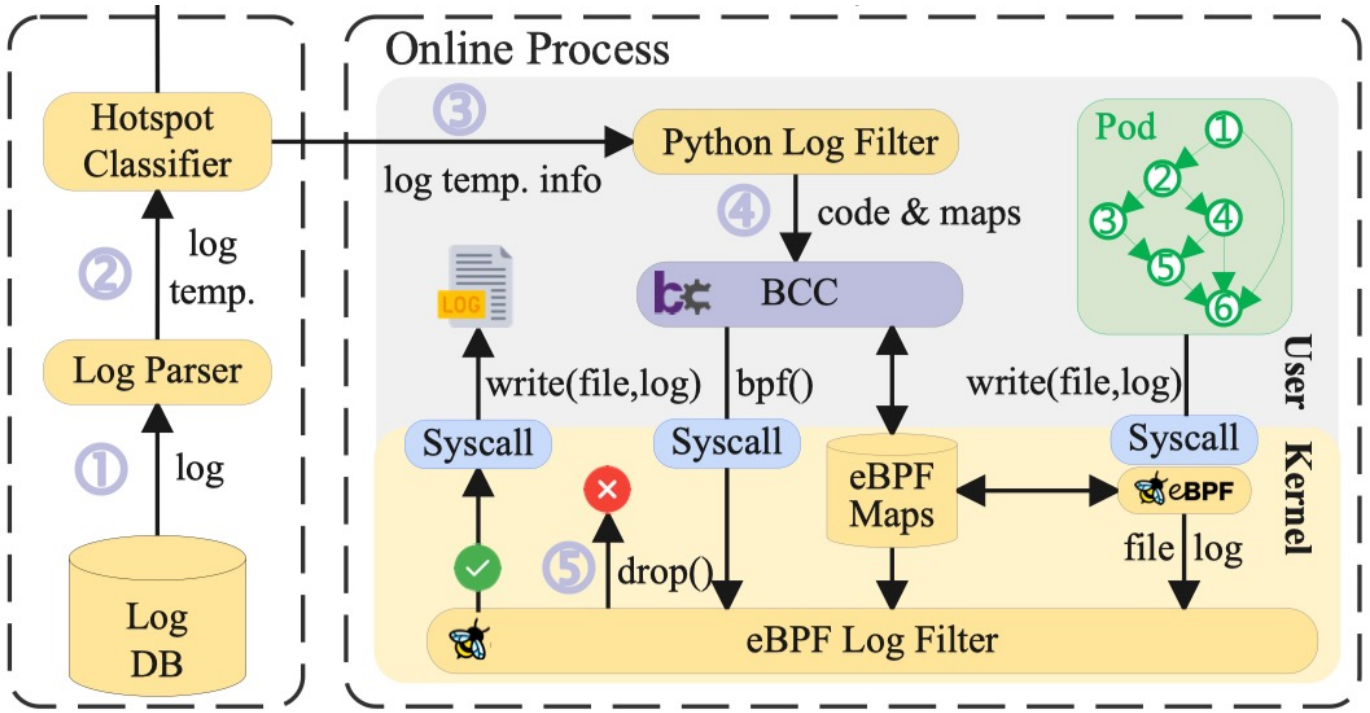
- Can we reduce the impact of log hotspots before fixing them?
  - Filter log hotspots in log collector
  - Filter log hotspots in log agent
  - Filter log hotspots **in the Linux kernel**
    - Reduce **storage costs and network transmission overhead**
    - Reduce **printing overhead**
    - **Reduce user and kernel switching, efficient kernel matching of logs**
    - **Non-intrusive, programming language independent**



LogReducer Online Process

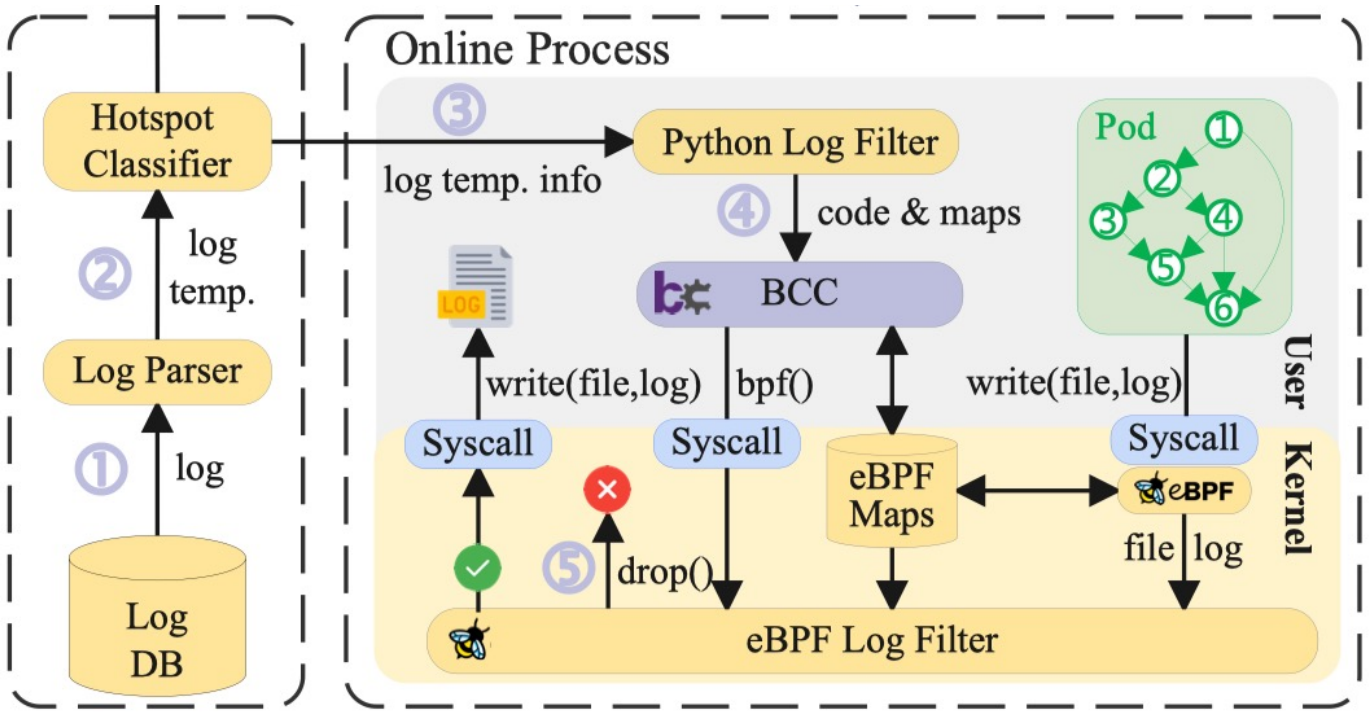
- Online Log Filter: **filter log hotspots in the Linux kernel with with extended Berkeley Packet Filter (eBPF)**

  ③ *Hotspot Classifier* triggers Online Log Filter



LogReducer Online Process

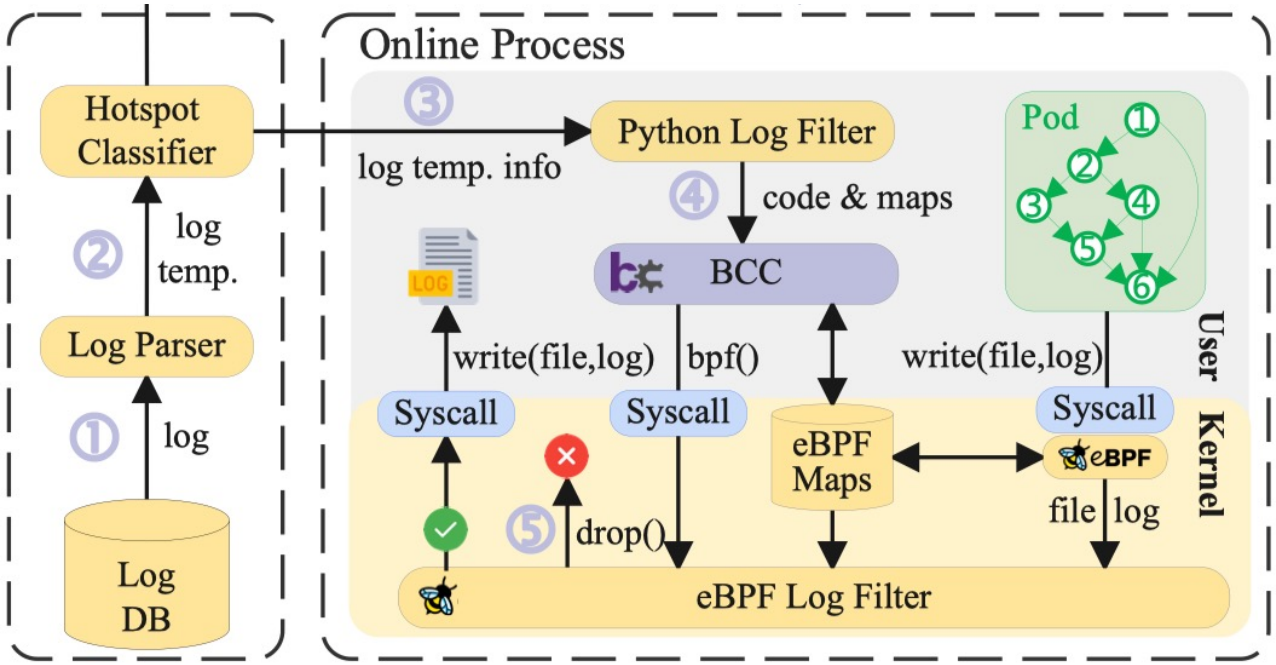
- Online Log Filter: **filter log hotspots in the Linux kernel with eBPF**
  - ③ *Hotspot Classifier* triggers Online Log Filter
  - ④ *Python Log Filter* loads **eBPF code and log signature** into kernel space



LogReducer Online Process

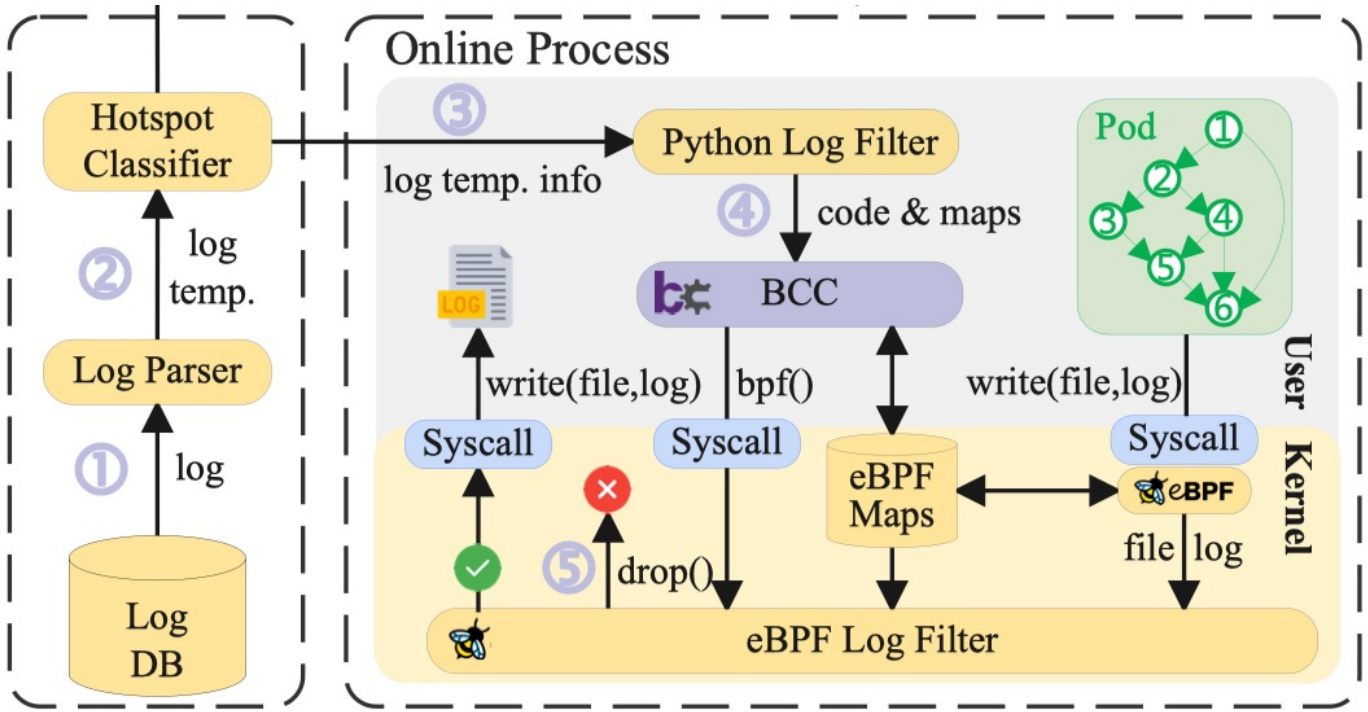
- Online Log Filter: **filter log hotspots in the Linux kernel with eBPF**
  - ③ *Hotspot Classifier* triggers Online Log Filter
  - ④ *Python Log Filter* loads eBPF code and log hotspots into kernel space
  - ⑤ *eBPF Log Filter* **intercepts sys_write() syscall, matches log signature** and **drops log hotspots** in the kernel

Log signature: [s1.cpp:6]	
11:45:00 [s1.cpp:6] Hello world	**drop**
11:45:00 [s1.cpp:2] Hello Melbourne	**write**
11:45:00 [s1.cpp:6] Hello world	**drop**



LogReducer Online Process

- Online Log Filter implementation [1]
  - Python 3.6 and BCC 0.24
  - Linux Kernel v5.4
  - CONFIG_BPF_KPROBE_OVERRIDE = y
- Benchmark design
  - Golang (go.uber.org/zap)
  - Python (nb_log)
  - Java (log4j2)
  - C++ (log4cplus)
- Overhead Measure
  - bpftool-prog



1 https://github.com/IntelligentDDS/LogReducer

LogReducer Online Process Result

- The overhead for Log Filter under different log volumes

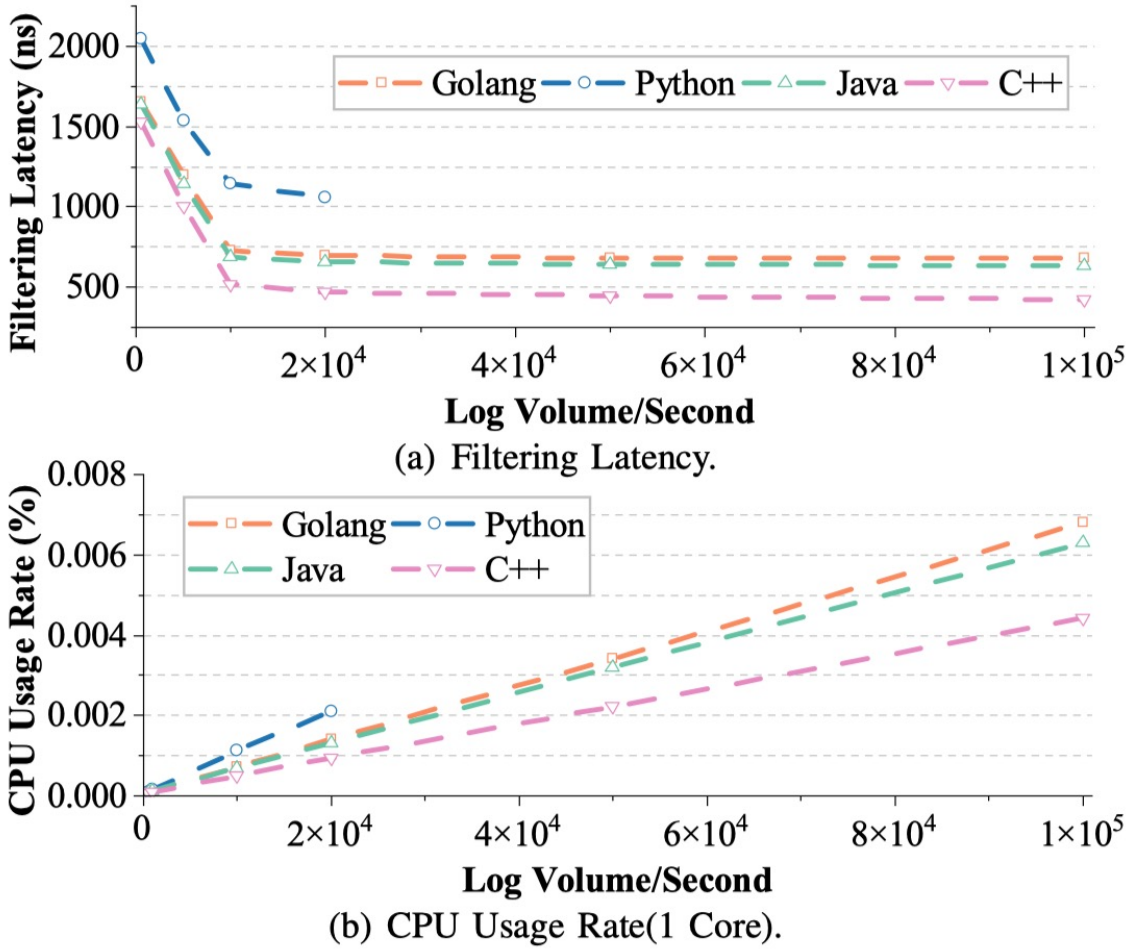
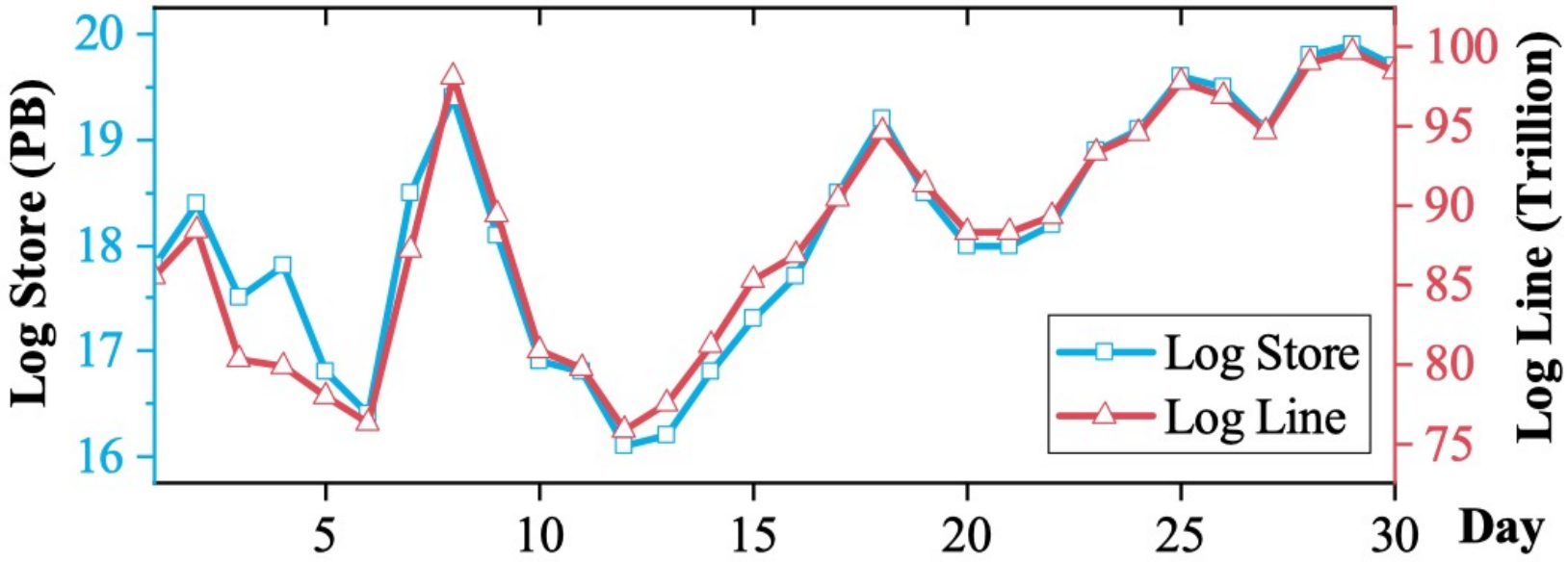


Fig. 14. The overhead for Log Filter in *LogReducer* when filtering log hotspots in kernel under different log volumes of 20 chars per log length.

- ✓ **Increase latency by about 500 nanoseconds when filtering 100,000 logs per second**
- ✓ **Consume 0.008% additional CPU utilization of 1 CPU core**

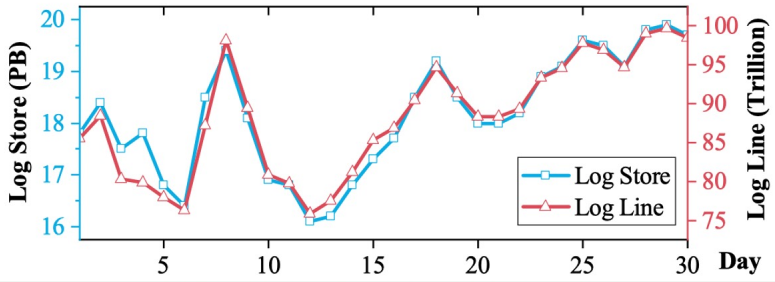
Conclusion

- Logs is one of the “Three Pillars of Observability”
- Excessive logging is a double-edged sword
  - **High storage cost**
  - **Huge performance overhead**
  - **Difficult to troubleshoot**



Conclusion

- Logs is one of the "Three Pillars of Observability"
- Excessive logging is a double-edged sword
  - **High storage cost**
  - **Huge performance overhead**
  - **Difficult to troubleshoot**

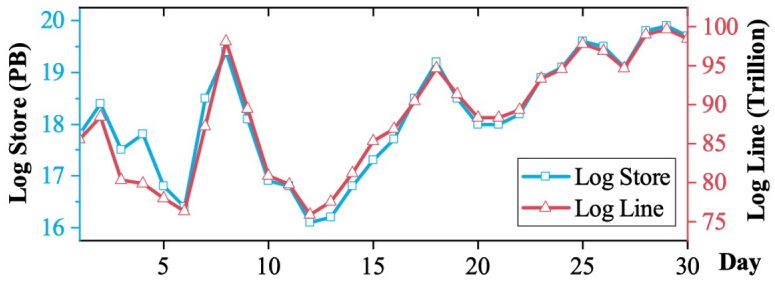


Conclusion

- Research Object
  - WeChat is a large real-world instant messaging system serving billions of users
  - **Top 20 services** with the highest log storage in WeChat
  - **57 log hotspots** and interviewed 19 corresponding service owners
- Research Questions
  - RQ1: How do log hotspots impact **application storage**?
  - RQ2: How do log hotspots impact **application runtime**?
  - RQ3: What are the **root causes** of log hotspots?
  - RQ4: What are the **fixing solutions** of log hotspots?
  - RQ5: **How long** do developers take to fix log hotspots?

Conclusion

- Logs is one of the "Three Pillars of Observability"
- Excessive logging is a double-edged sword
  - **High storage cost**
  - **Huge performance overhead**
  - **Difficult to troubleshoot**



- Research Object
  - WeChat is a large real-world instant messaging system serving billions of users
  - **Top 20 services** with the highest log storage in WeChat
  - **57 log hotspots** and interviewed 19 corresponding service owners
- Research Questions
  - RQ1: How do log hotspots impact **application storage**?
  - RQ2: How do log hotspots impact **application runtime**?
  - RQ3: What are the **root causes** of log hotspots?
  - RQ4: What are the **fixing solutions** of log hotspots?
  - RQ5: **How long** do developers take to fix log hotspots?

Conclusion

- We applied LogReducer offline process in WeChat since April 14, 2022
- Log storage in WeChat dropped from **19.7 PB to 12.0 PB** per day

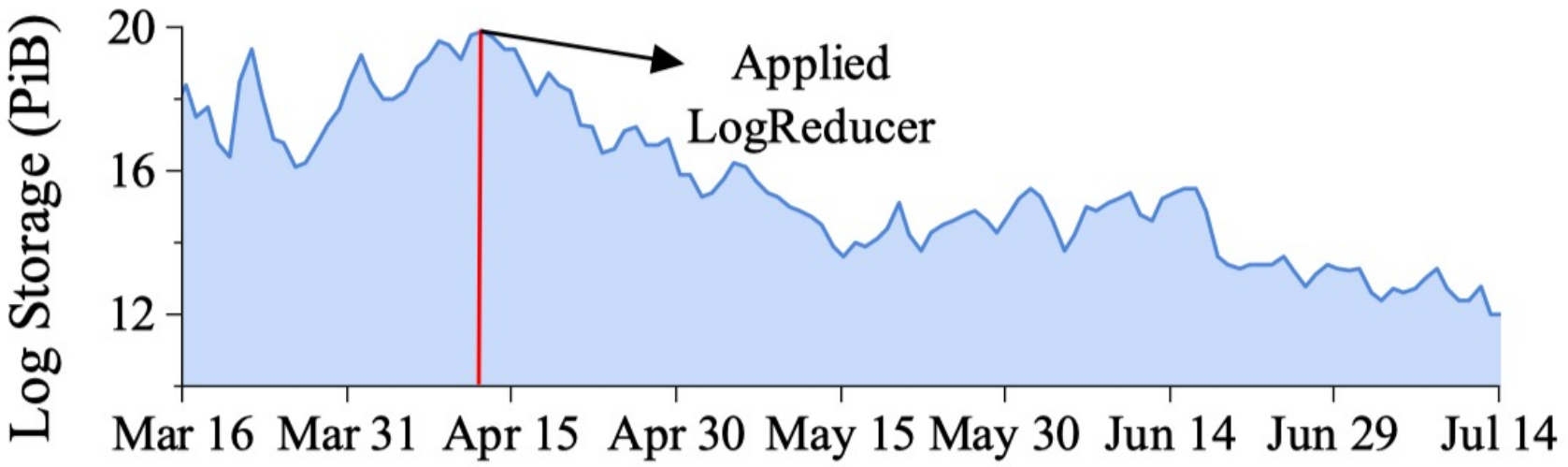
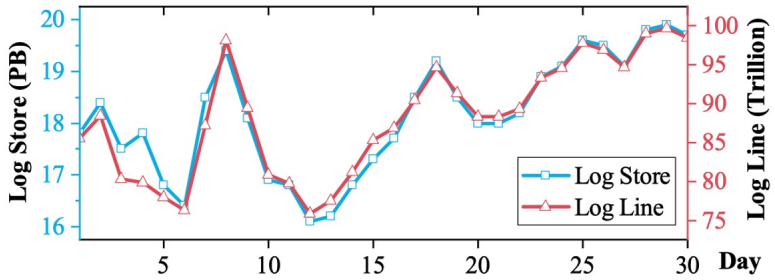


Fig. 17. Changes in the log storage of WeChat from Mar 16 to Jul 14 in 2022.

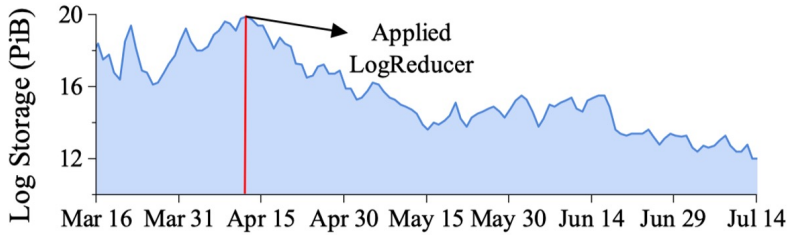
39.08% decrease

Conclusion

- Logs is one of the "Three Pillars of Observability"
- Excessive logging is a double-edged sword
  - **High storage cost**
  - **Huge performance overhead**
  - **Difficult to troubleshoot**



- We applied LogReducer offline process in WeChat since April 14, 2022
- Log storage in WeChat dropped from **19.7 PB to 12.0 PB** per day



39.08% decrease

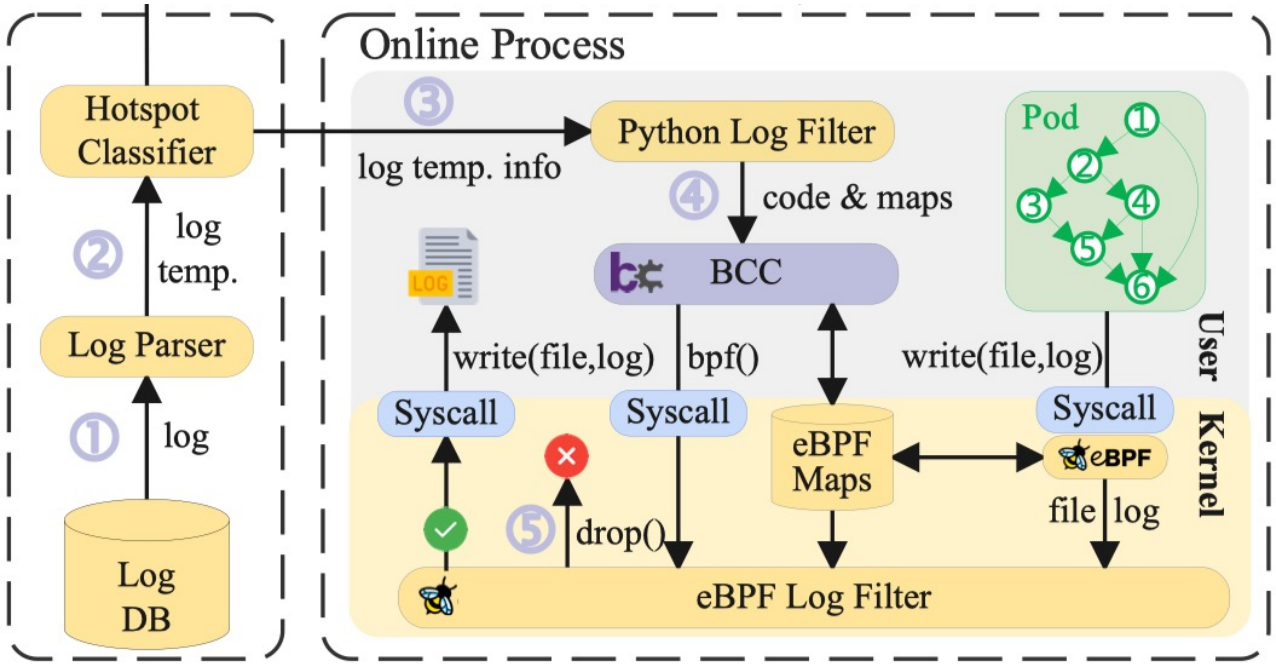
Fig. 17. Changes in the log storage of WeChat from Mar 16 to Jul 14 in 2022.

- Research Object
  - WeChat is a large real-world instant messaging system serving billions of users
  - **Top 20 services** with the highest log storage in WeChat
  - **57 log hotspots** and interviewed 19 corresponding service owners
- Research Questions
  - RQ1: How do log hotspots impact **application storage**?
  - RQ2: How do log hotspots impact **application runtime**?
  - RQ3: What are the **root causes** of log hotspots?
  - RQ4: What are the **fixing solutions** of log hotspots?
  - RQ5: **How long** do developers take to fix log hotspots?

Conclusion

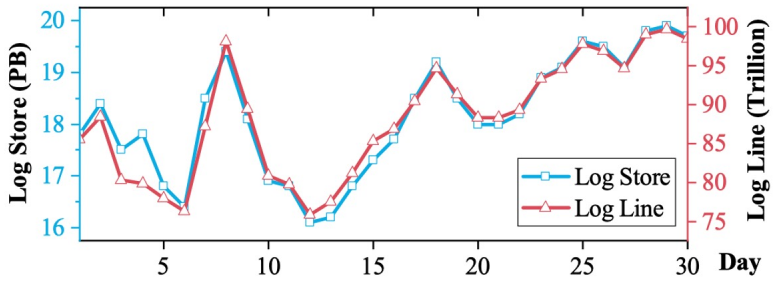
- Online Log Filter: **filter log hotspots in the Linux kernel with eBPF**
  - ③ *Hotspot Classifier* triggers Online Log Filter
  - ④ *Python Log Filter* loads eBPF code and log hotspots into kernel space
  - ⑤ *eBPF Log Filter* **intercepts sys_write() syscall, matches log signature** and **drops log hotspots** in the kernel

Log signature: [s1.cpp:6]	
11:45:00 [s1.cpp:6] Hello world	**drop**
11:45:00 [s1.cpp:2] Hello Melbourne	**write**
11:45:00 [s1.cpp:6] Hello world	**drop**



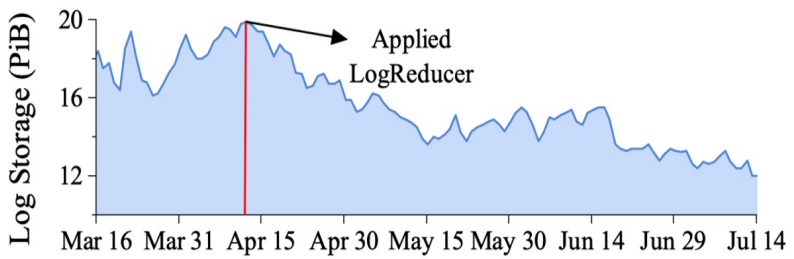
Conclusion

- Logs is one of the "Three Pillars of Observability"
- Excessive logging is a double-edged sword
  - **High storage cost**
  - **Huge performance overhead**
  - **Difficult to troubleshoot**



- Research Object
  - WeChat is a large real-world instant messaging system serving billions of users
  - **Top 20 services** with the highest log storage in WeChat
  - **57 log hotspots** and interviewed 19 corresponding service owners
- Research Questions
  - RQ1: How do log hotspots impact **application storage**?
  - RQ2: How do log hotspots impact **application runtime**?
  - RQ3: What are the **root causes** of log hotspots?
  - RQ4: What are the **fixing solutions** of log hotspots?
  - RQ5: **How long** do developers take to fix log hotspots?

- We applied LogReducer offline process in WeChat since April 14, 2022
- Log storage in WeChat dropped from **19.7 PB to 12.0 PB** per day

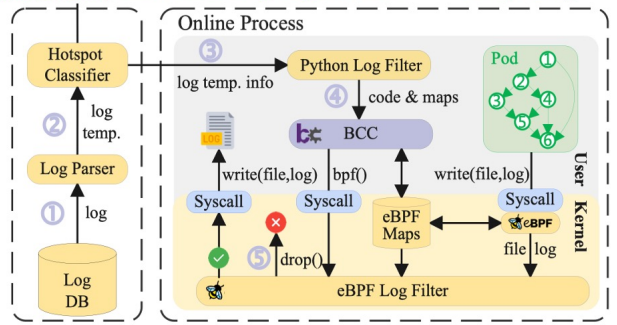


39.08% decrease

Fig. 17. Changes in the log storage of WeChat from Mar 16 to Jul 14 in 2022.

- Online Log Filter: **filter log hotspots in the Linux kernel with eBPF**
  - ③ *Hotspot Classifier* triggers Online Log Filter
  - ④ *Python Log Filter* loads eBPF code and log hotspots into kernel space
  - ⑤ *eBPF Log Filter* **intercepts sys_write() syscall, matches log signature** and **drops log hotspots** in the kernel

Log signature: [s1.cpp:6]	
11:45:00 [s1.cpp:6] Hello world	**drop**
11:45:00 [s1.cpp:2] Hello Melbourne	**write**
11:45:00 [s1.cpp:6] Hello world	**drop**



中山大學 SUN YAT-SEN UNIVERSITY

WeChat

ICSE2023

Thank you !

Q & A

Homepage

WeChat Official Accounts

https://yuxiaoba.github.io
yugb5@mail2.sysu.edu.cn
https://github.com/IntelligentDDS/LogReducer

Supplementary

- RQ3: What are the **root causes** of log hotspots?
  - We interview developers to obtain **root causes** and summarized their type

TABLE I
STATISTICS OF ROOT CAUSE FOR LOG HOTSPOTS.

Root Cause \ Metric	Distribution		Fixing Time(day)		
	Count	%	Mean	Std	Med
Incorrect Log Level	**23**	**40.35**	10.91	**11.28**	5
Forgotten Test Log	13	22.8	4.76	1.87	4
Dependent Module Fault	6	10.5	2.83	0.408	3
Dependent Package Log	5	8.77	3	0	3
Incorrect Log Dye	4	7.01	**41**	0	**41**
Reasonable Hotspot	3	5.26	5	2.0	5
Self-Module Fault	2	3.5	2	1.41	2
Others	1	1.75	3	0	3
Total	57	100	9.31	11.83	3

- ✓ **Incorrect Log Level (40.3%)**
- ✓ **Forgotten Test Log (22.8%)**
- ✓ **Dependent Fault (10.5%)**
- ✓ **Dependent Package Log (8%)**

Supplementary

- RQ4: What are the **fixing solutions** of log hotspots?
  - We interview developers to obtain **fixing solutions** and summarized their type

TABLE II
STATISTICS OF ROOT CAUSE FOR LOG HOTSPOTS.

Root Cause \ Metric	Distribution		Fixing Time(day)		
	Count	%	Mean	Std	Med
Correct Log Level	**18**	**31.57**	11.77	**12.68**	3
Delete Log Statement	17	29.82	5.76	2.07	7
Fix Module Fault	9	15.78	2.66	0.7	3
Turn on Log Dye	6	10.52	3	0	3
Correct Log Dye	4	7.01	**41**	0	**41**
Merge Log Statement	2	3.5	4	1.41	4
Reduce Log Length	1	1.75	7	0	7
Total	57	100	9.31	11.83	3

- ✓ **Correct Log Level (31.5%)**
- ✓ **Delete Log Statement (29.8%)**
- ✓ **Fix Module Fault (15.7%)**
- ✓ **Turn on Log Dye (10.5%)**

Supplementary

- eBPF can run sandboxed programs in Linux kernel
  - eBPF programs are event-driven and are run when the kernel passes a certain hook point