Understanding Customer Problem Troubleshooting from Storage System Logs

Weihang Jiang (wjiang3@uiuc.edu)

Weihang Jiang[*+], Chongfeng Hu[*+], Shankar Pasupathy[+], Arkady Kanevsky[+], Zhenmin Li[#], Yuanyuan Zhou[*]

University of Illinois[*] NetApp[+] Pattern Insight[#]

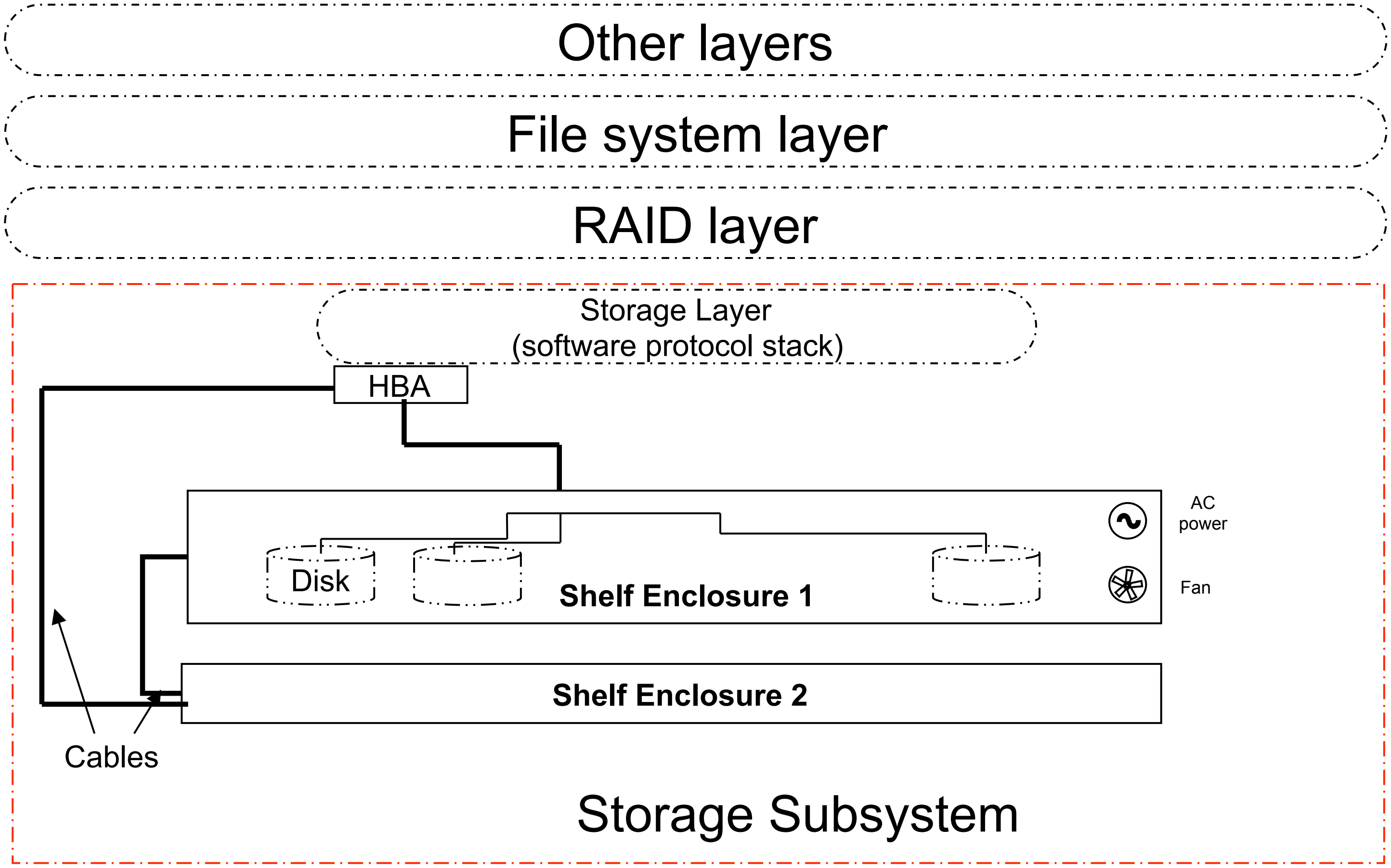
Customer problem troubleshooting is critical

- Customer problems result in costly downtime for customers
  - Cost a customer 18.35% of TCO [Crimson ’07].

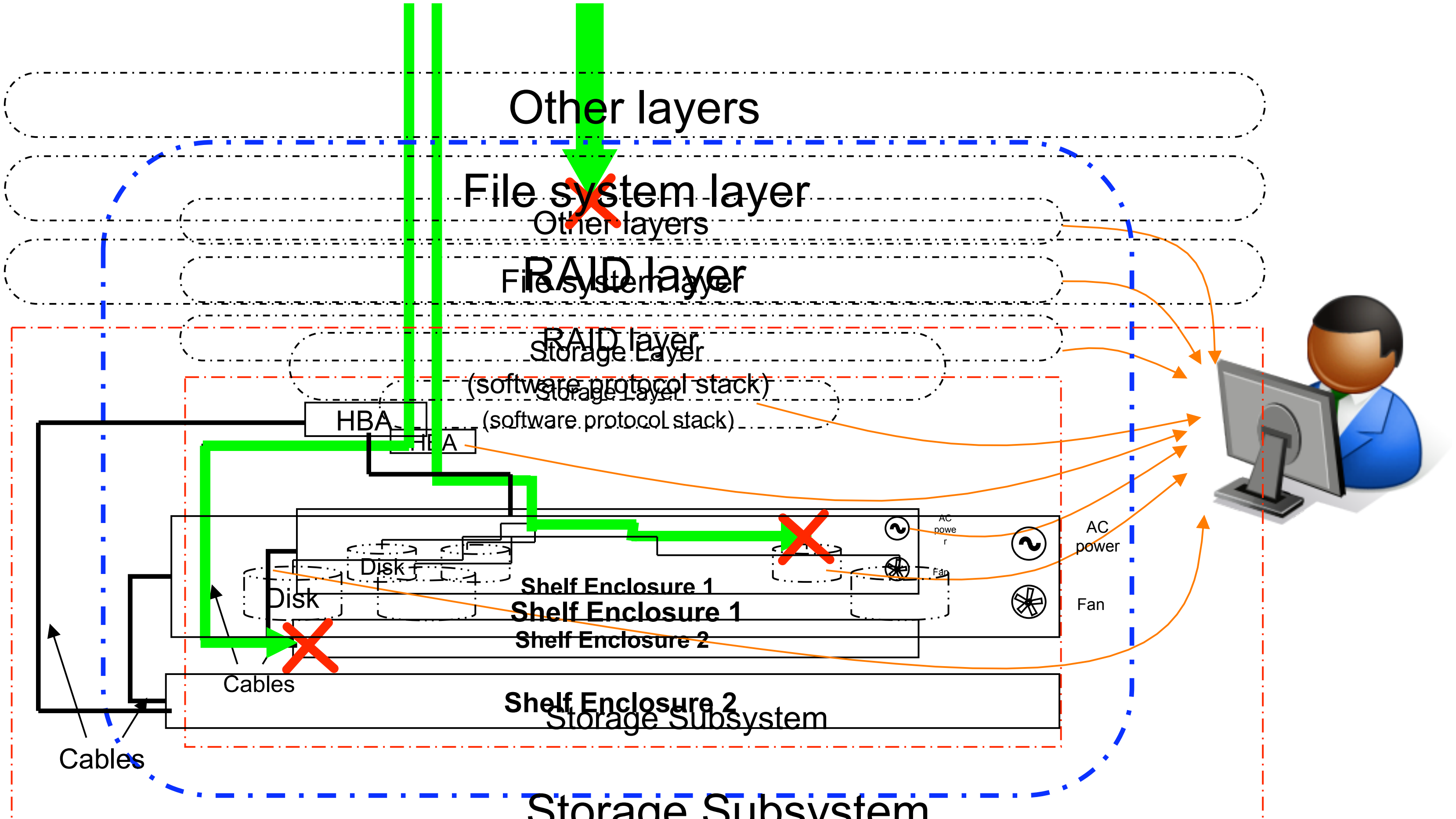
- Customer problems are expensive for system vendors
  - Vendors devote more than 8% of total revenue and 15% of total employee costs on customer problem support [ASP’08].

- Complex modern storage systems make problem troubleshooting more challenging

Storage system is complex

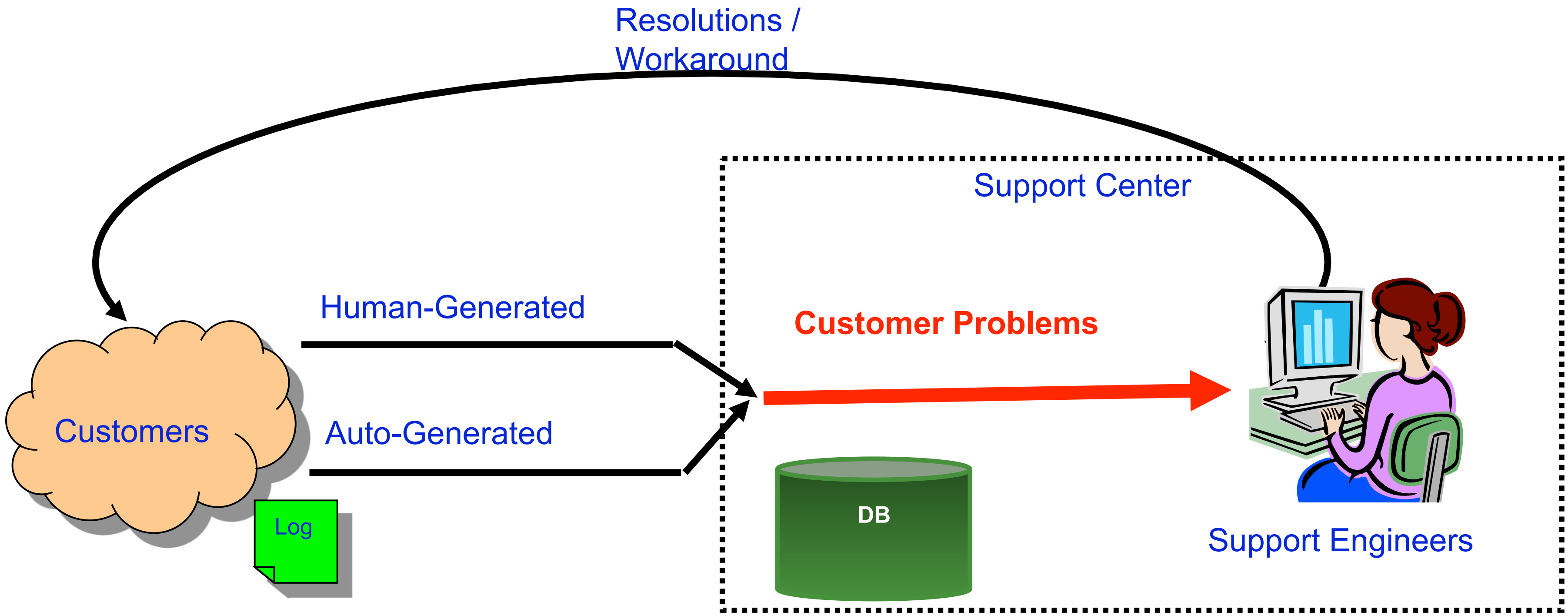


Customer problems occur in different ways



- Customer problems include storage failures, partial failures and any other system misbehaviors that users observe and do not expect from a healthy system.

Customer problem management workflow



Quantitatively understand problem troubleshooting

Can we systematically use system logs for troubleshooting?

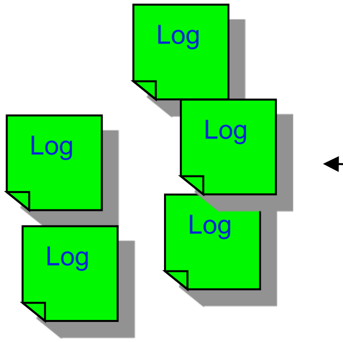
Outline

- Motivation
- Understanding customer problem troubleshooting
  - Problem troubleshooting time
  - Problem category
  - Problem impacts
- Use log information for problem troubleshooting
- Conclusions

Data source

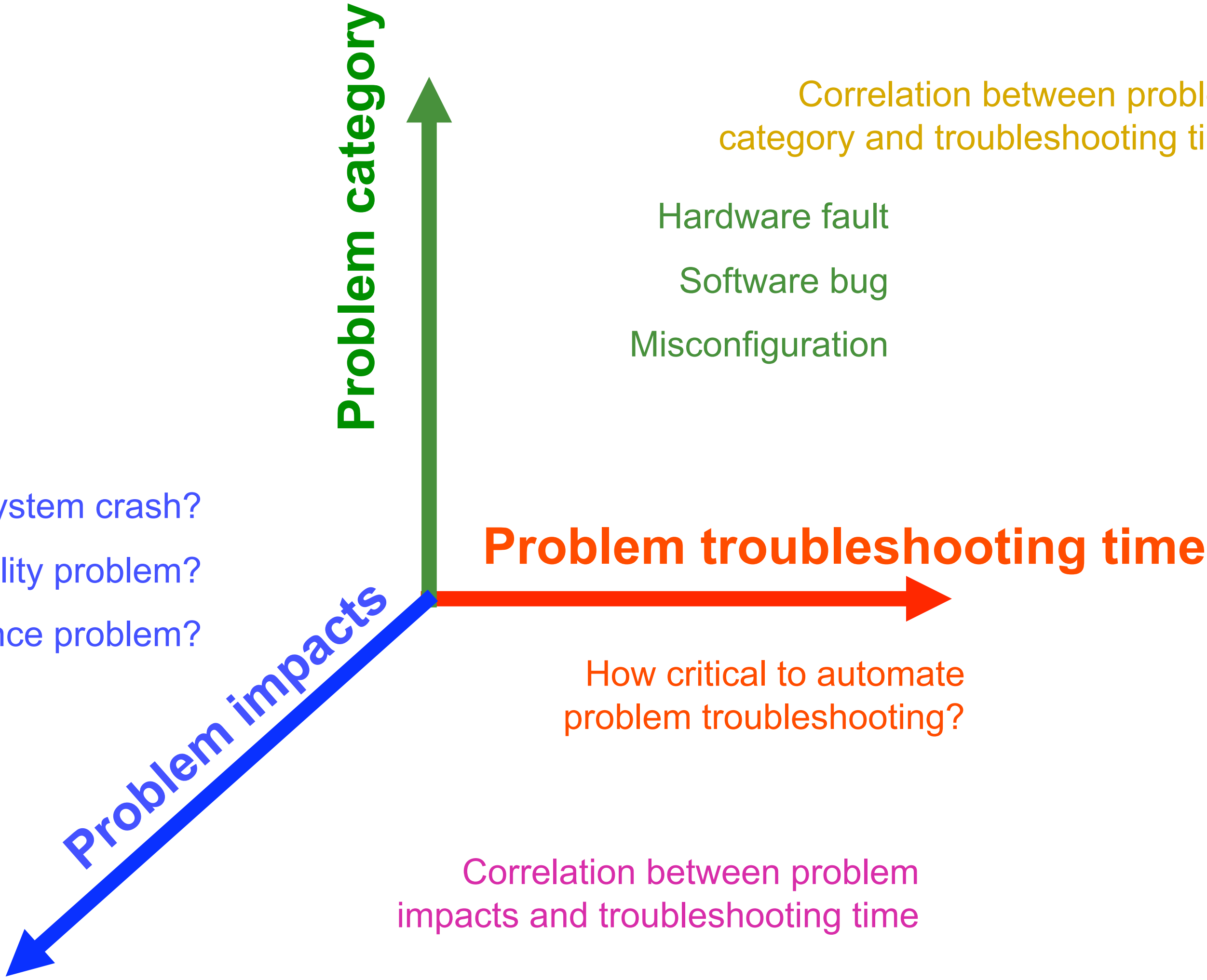
Customer problem case database (636,108)

Case ID	Report Date	Resolution/ Workaround Date	Problem cause		Auto-generated	Critical Event
			High-level	Module-level		
1	5/1/06 11:21	5/2/06 13:35	Software Bugs	File System	Y	Crash
2	5/2/06 11:02	5/7/06 9:01	Hardware Fault	SCSI	N	N/A
3	5/3/06 15:40	5/8/06 14:48	Misconfiguration	Shelf	N	N/A



Storage System Log Archive (306,624 logs)

Analysis dimensions



Problem category

Problem troubleshooting time

Problem impacts

Correlation between problem category and troubleshooting time

Hardware fault

Software bug

Misconfiguration

System crash?

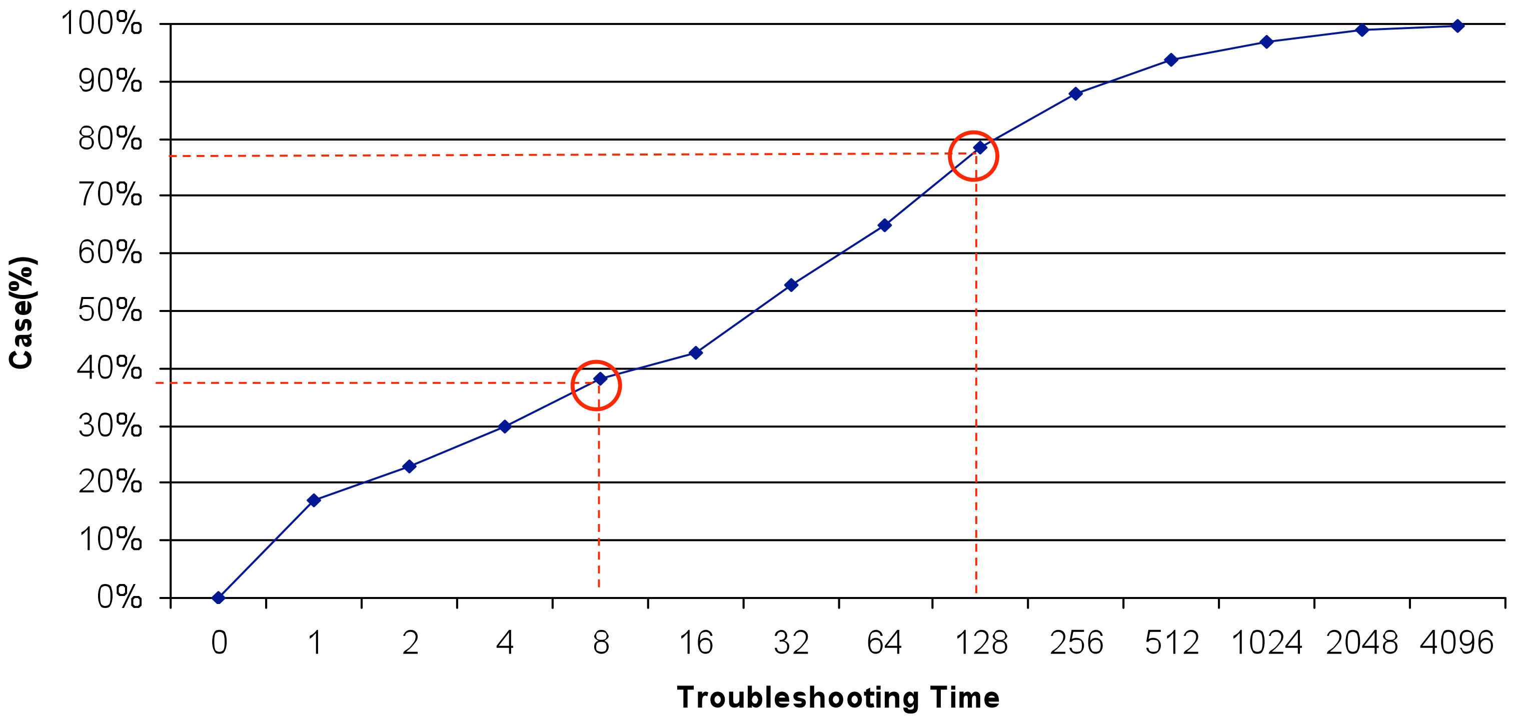
Usability problem?

Performance problem?

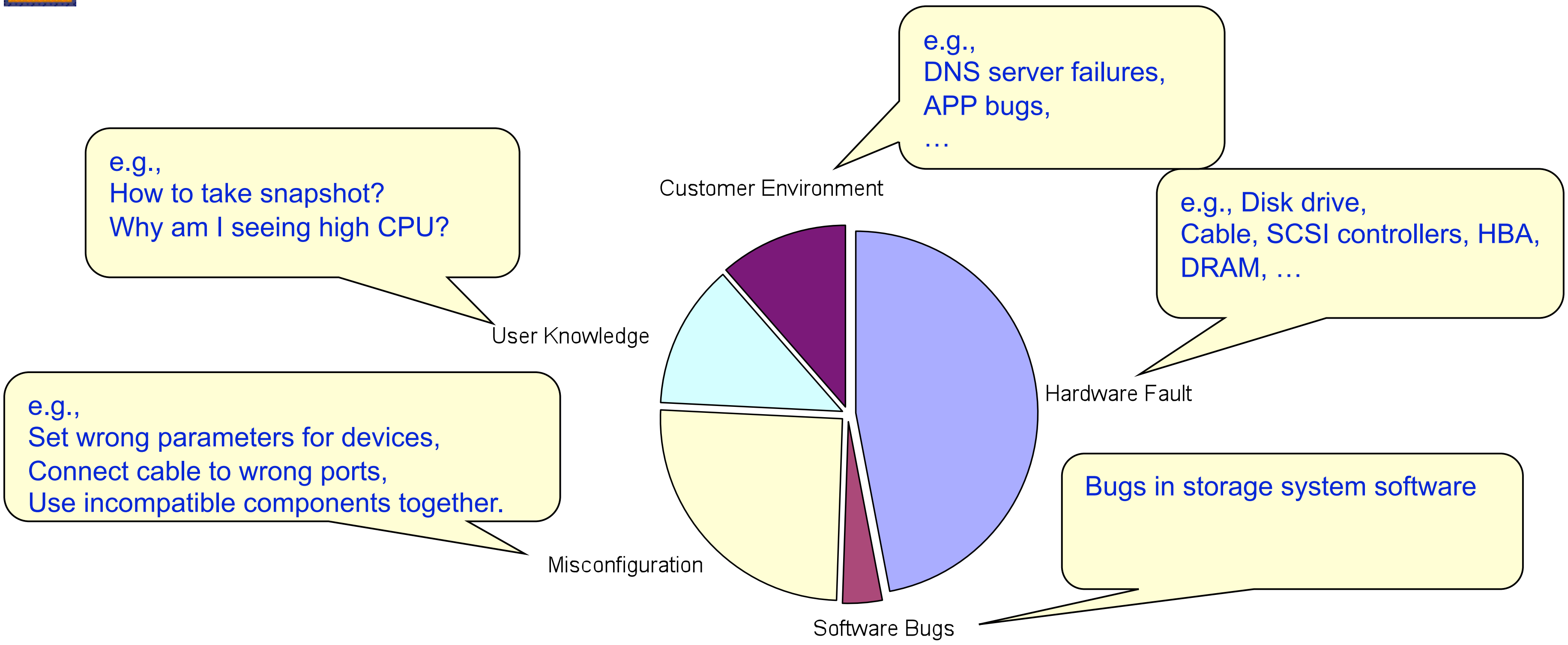
How critical to automate problem troubleshooting?

Correlation between problem impacts and troubleshooting time

Problem troubleshooting is time-consuming



Problem category distribution



e.g.,
How to take snapshot?
Why am I seeing high CPU?

e.g.,
DNS server failures,
APP bugs,
…

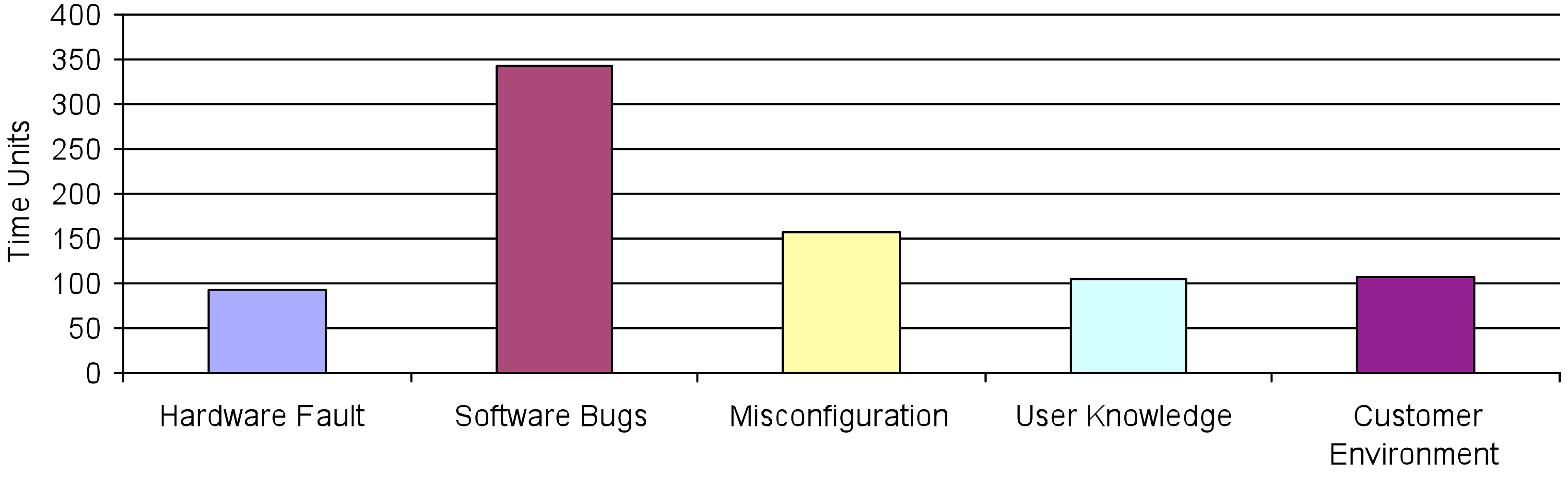
e.g., Disk drive,
Cable, SCSI controllers, HBA,
DRAM, …

e.g.,
Set wrong parameters for devices,
Connect cable to wrong ports,
Use incompatible components together.

Bugs in storage system software

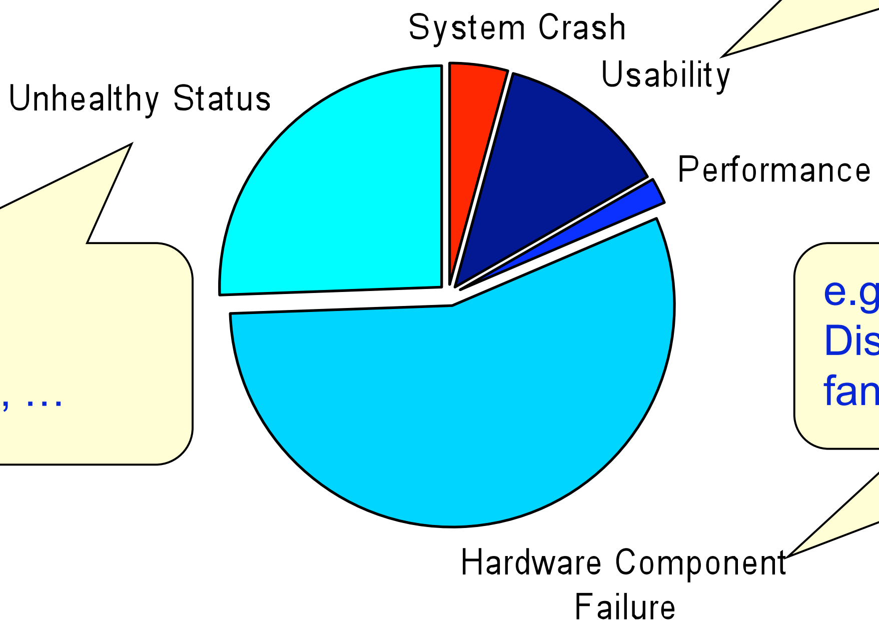
- Hardware fault (40%) and misconfiguration(21%) are the two most frequent categories, software bugs count for a small percentage(3%).
- User knowledge (11%) and customers’ own execution environment (9%).

Problem category and troubleshooting time



- Software bugs take longer time to troubleshoot.
- For all categories, troubleshooting is time-consuming.

Problem impact distribution



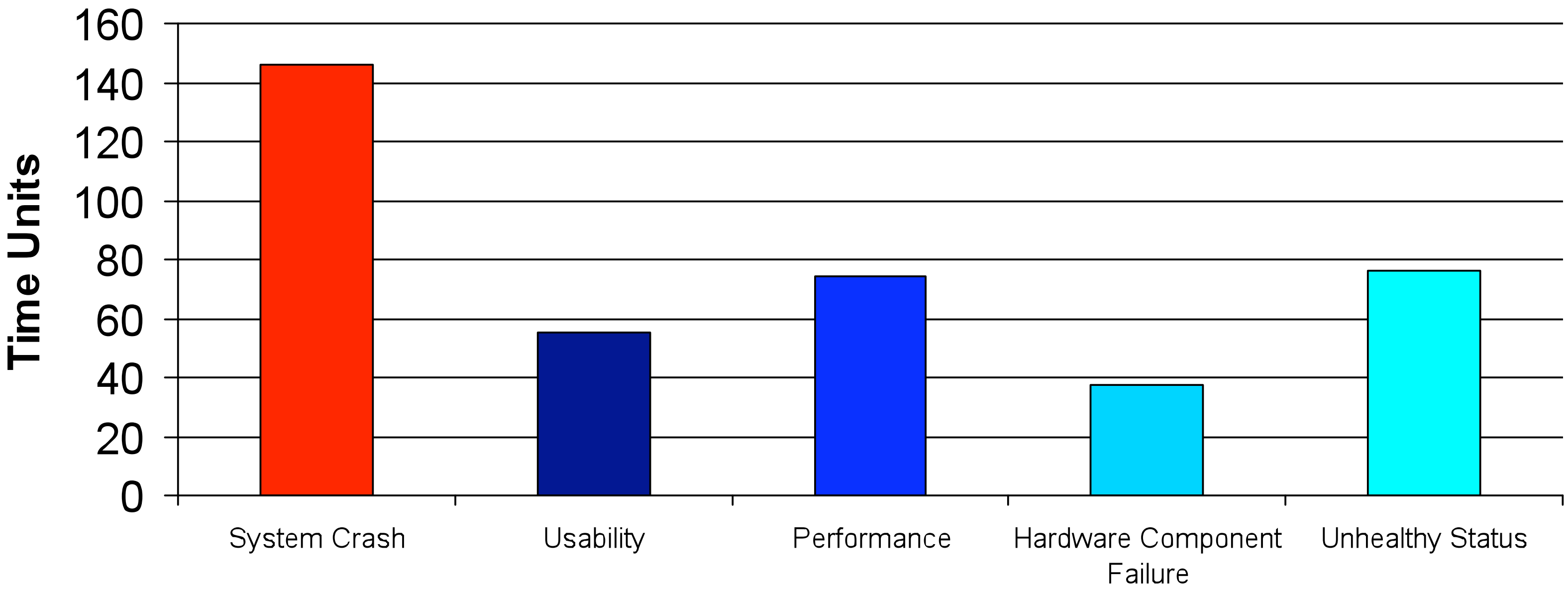
e.g.,
Can not access a disk volume,
Can not take snapshot,
…

e.g.,
Low spare disk count,
Instable interconnects, …

e.g.,
Disk, link, HBA, power, supply, fan.

- Problems are captured at early stages
  - System crash(3%)
  - Hardware component(44%), unhealthy status(20%)

Problem impact and troubleshooting time



- System crash takes longer time to troubleshoot.
- For all categories, troubleshooting is time-consuming.

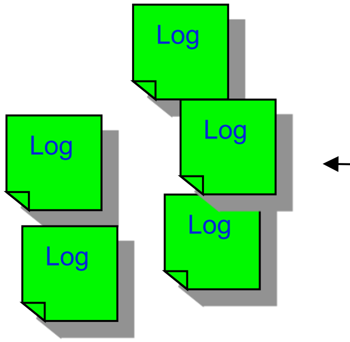
Outline

- Motivation
- → Understanding customer problem troubleshooting
  - Problem troubleshooting time
  - Problem category
  - Problem impacts
- Use log information for problem troubleshooting
- Conclusions

Use log information for problem diagnosis

Customer problem case database (636,108)

Case ID	Report Date	Resolution/ Workaround Date	Problem cause		Auto-generated	Critical Event
			High-level	Module-level		
1	5/1/06 11:21	5/1/06 13:35	Software Bugs	File System	Y	Crash
2	5/2/06 11:02	5/2/06 9:01	Hardware Fault	SCSI	N	N/A
3	5/3/06 15:40	5/8/06 14:48	Misconfiguration	Shelf	N	N/A



Storage System Log Archive (306,624 logs)

What log information to use?

ONE log event is enough?

Single Event revealing problem root cause

```
Sat Apr 15 05:58:15 EST [busError]: SCSI adapter encountered an unexpected bus phase. Issuing SCSI bus reset.
Sat Apr 15 05:59:10 EST [fs.warn]: volum        l1 is low on free space. 98% in use.
Sat Apr 15 06:01:10 EST [fs.warn]:
Sat Apr 15 06:02:14 EST [raidDisk
Sat Apr 15 06:02:14 EST [raidDisk
                                    …
Sat Apr 15 06:07:19 EST [timeoutE                                                             etried.
Sat Apr 15 06:07:19 EST [noPathsE
Sat Apr 15 06:07:19 EST [timeoutError]: de           not respond to requested I/O. I/O will be retried.
Sat Apr 15 06:07:19 EST [noPathsError]         re paths to device 9b. All retries have failed.
Sat Apr 15 06:08:23 EST [filerUp]: Fil      up and running.
                                    ……
Sat Apr 15 06:24:07 EST [crash:ALERT]: Crash String: File system hung in process idle_thread1
```

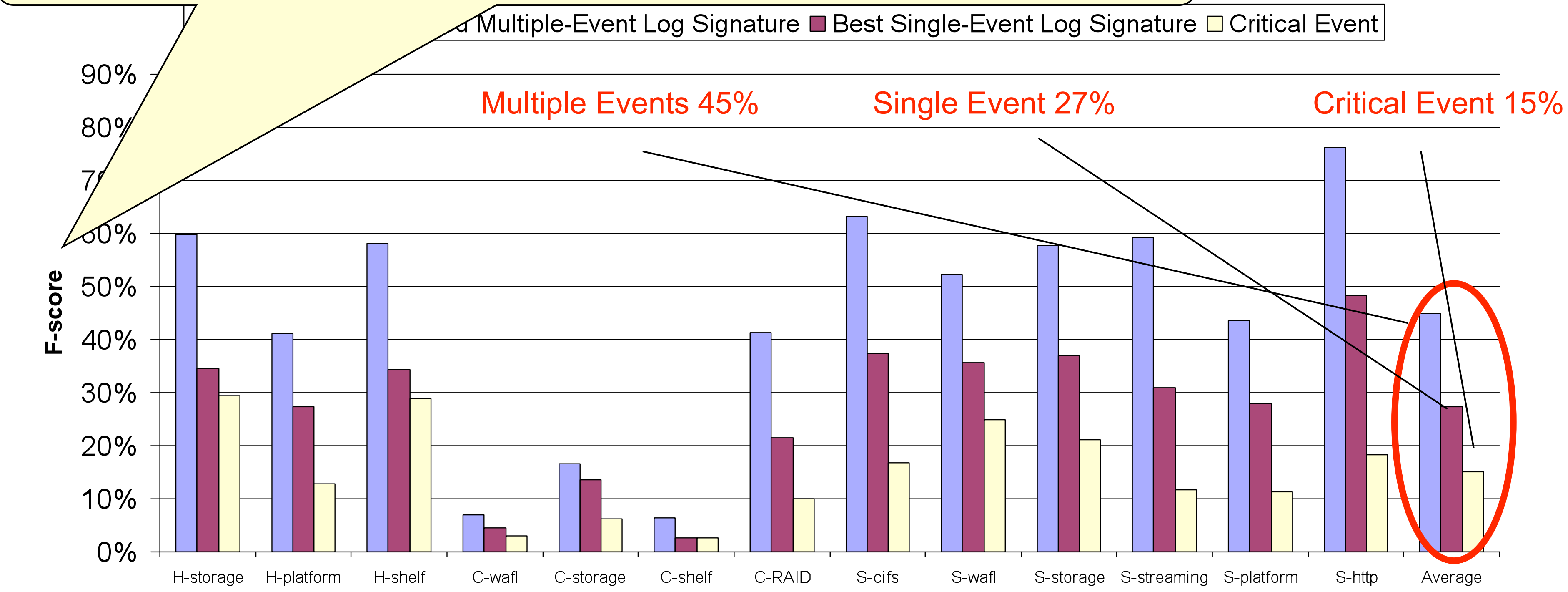
More events, better ?

→ Critical Event

Critical event is ready to use

More log events are more useful

How well the signature can uniquely identify cause?
F-score = 2 * Precision * Recall / (Precision + Recall)



- Critical event alone is not enough.
- Using more log events can bring better accuracy.

Challenges and opportunities

- 😐 Logs are noisy

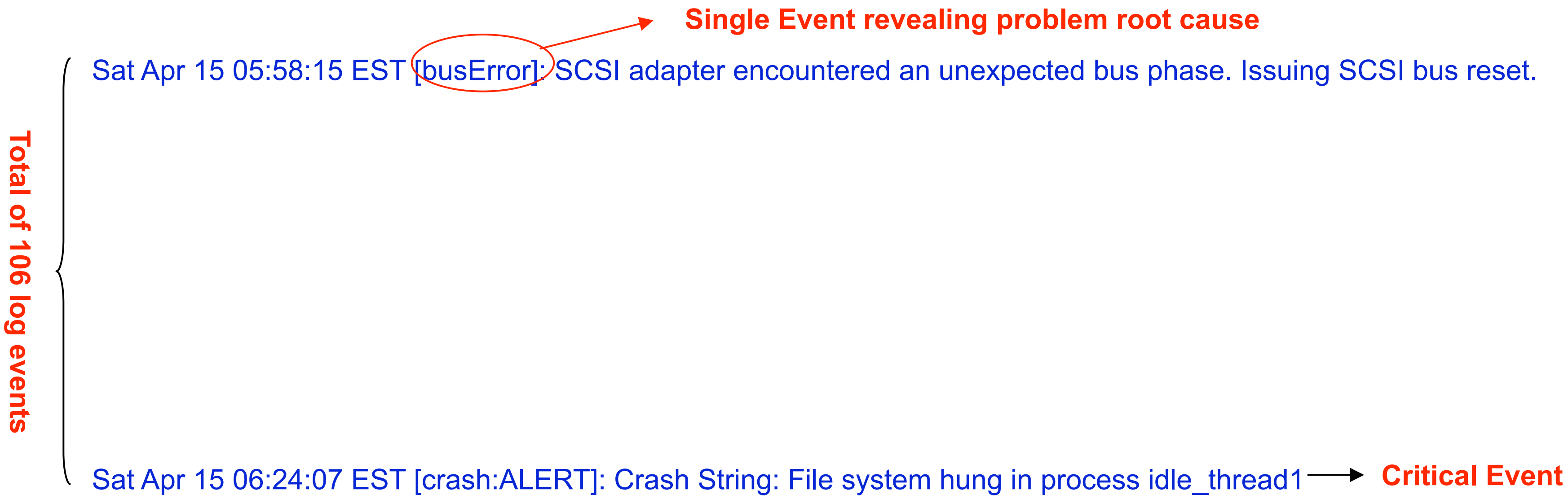
Single Event revealing problem root cause

```
Sat Apr 15 05:58:15 EST [busError]: SCSI adapter encountered an unexpected bus phase. Issuing SCSI bus reset.
Sat Apr 15 05:59:10 EST [fs.warn]: volume /vol/vol1 is low on free space. 98% in use.
Sat Apr 15 06:01:10 EST [fs.warn]: volume /vol/vol10 is low on free space. 99% in use.
Sat Apr 15 06:02:14 EST [raidDiskRecovering]: Attempting to bring device 9a back into service.
Sat Apr 15 06:02:14 EST [raidDiskRecovering]: Attempting to bring device 9b back into service.
                                          ……
Sat Apr 15 06:07:19 EST [timeoutError]: device 9a did not respond to requested I/O. I/O will be retried.
Sat Apr 15 06:07:19 EST [noPathsError]: No more paths to device 9a: All retries have failed.
Sat Apr 15 06:07:19 EST [timeoutError]: device 9b did not respond to requested I/O. I/O will be retried.
Sat Apr 15 06:07:19 EST [noPathsError]: No more paths to device 9b. All retries have failed.
Sat Apr 15 06:08:23 EST [filerUp]: Filer is up and running.
                                          ……
Sat Apr 15 06:24:07 EST [crash:ALERT]: Crash String: File system hung in process idle_thread1
```

Critical Event

Challenges and opportunities

- 😐 Logs are noisy
- 😐 Important log events are not easy to locate



Single Event revealing problem root cause

```
Sat Apr 15 05:58:15 EST [busError]: SCSI adapter encountered an unexpected bus phase. Issuing SCSI bus reset.

Sat Apr 15 06:24:07 EST [crash:ALERT]: Crash String: File system hung in process idle_thread1
```

Critical Event

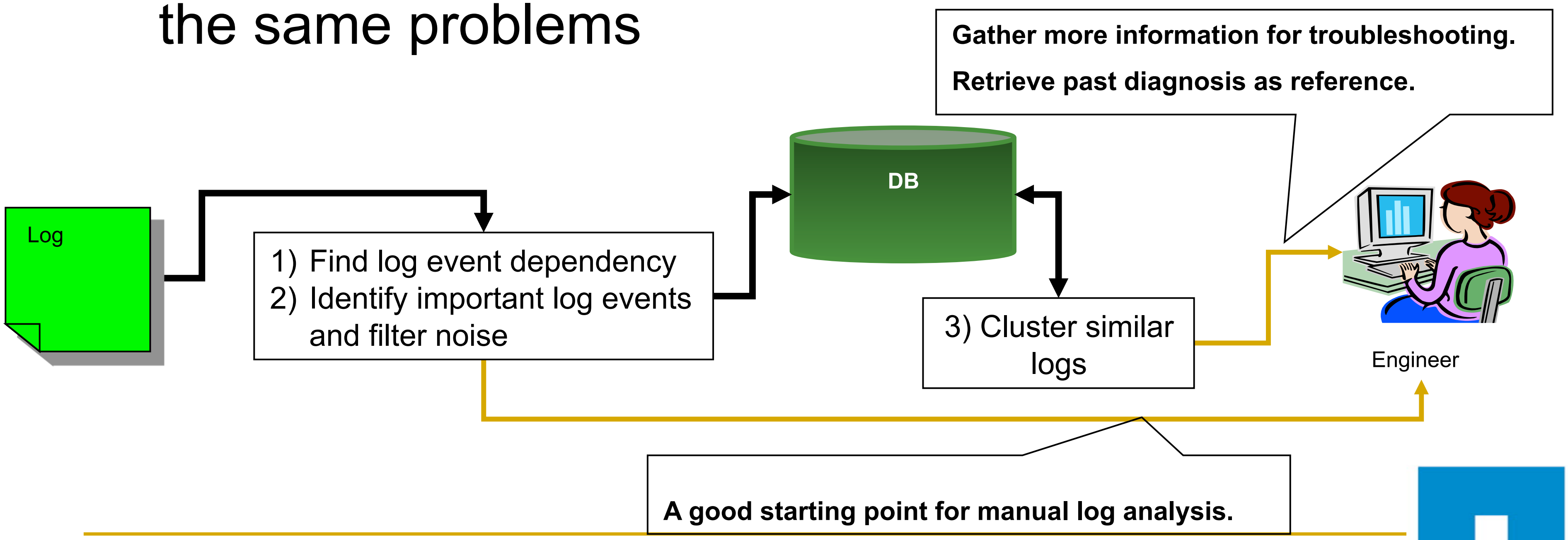
Total of 106 log events

Challenges and opportunities

- 😐 Logs are noisy
- 😐 Important log events are not easy to locate
- ☺ Similar log patterns appear on systems experience the same problems

Challenges and opportunities

- 😐 Logs are noisy
- 😐 Important log events are not easy to locate
- ☺ Similar log patterns appear on systems experience the same problems



Gather more information for troubleshooting.

Retrieve past diagnosis as reference.

Log

1) Find log event dependency
2) Identify important log events and filter noise

DB

3) Cluster similar logs

Engineer

A good starting point for manual log analysis.

Conclusions

- Problem troubleshooting is time-consuming.
  - Hardware fault and misconfiguration are common causes
  - Lack of sufficient user knowledge
  - Most problems have low impact, while high-impact problems are more difficult to troubleshoot
- Storage system logs contain useful information for problem troubleshooting
  - Critical event alone is not enough.
  - Log analysis tools that can filter noise and identify similar patterns are essential to improve troubleshooting.

Thanks

Questions?