ISSRE 2017
Toulouse, France

Experience Report: Fault Triggers in Linux Operating System: From Evolution Perspective

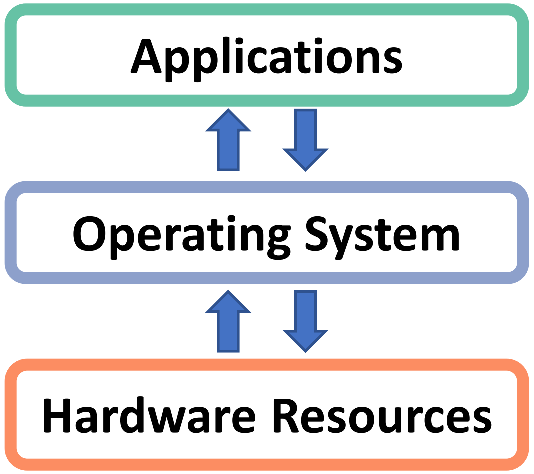
Guanping Xiao, Zheng Zheng*, Beibei Yin, Kishor S. Trivedi,
Xiaoting Du, Kaiyuan Cai

BEIHANG UNIVERSITY

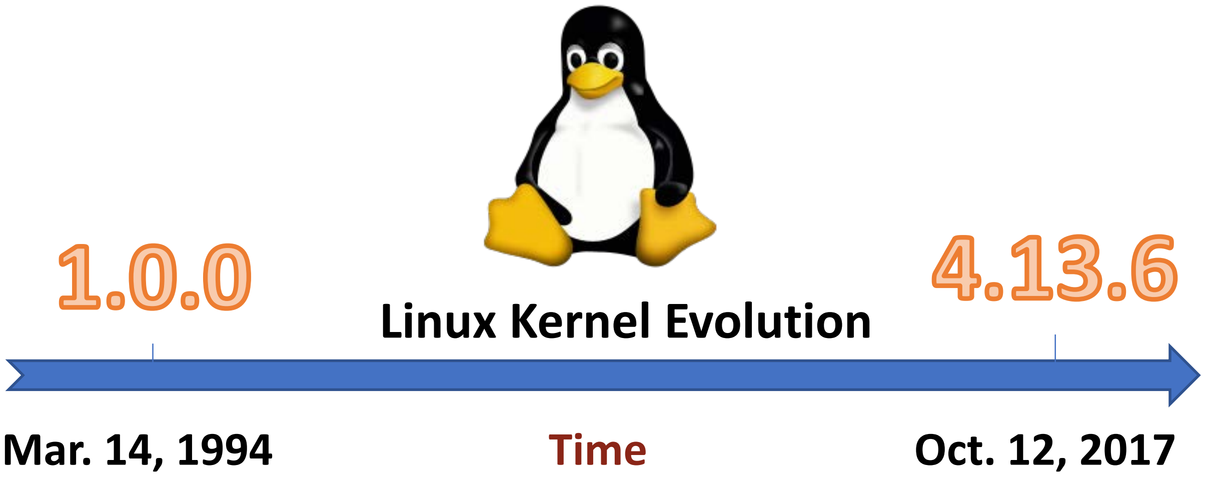
DUKE UNIVERSITY

Oct. 24, 2017 · Toulouse, France

Motivations



- An operating system (OS) provides operating environments for software. Thus, the assurance of high reliability of an OS is critical.



- **Linux** has put out **more than 1300 releases** ranging from **versions 1.0 to 4.1** over the past **20 years**

Motivations

- It is necessary to explore the characteristics of Linux bug data from the evolution perspective. For instance,

Evolution of bug type proportions

Evolution of regression bug proportion

Linux Bug Data

Difference of bug types' time to fix

What metrics can reflect bug evolutions

Outline

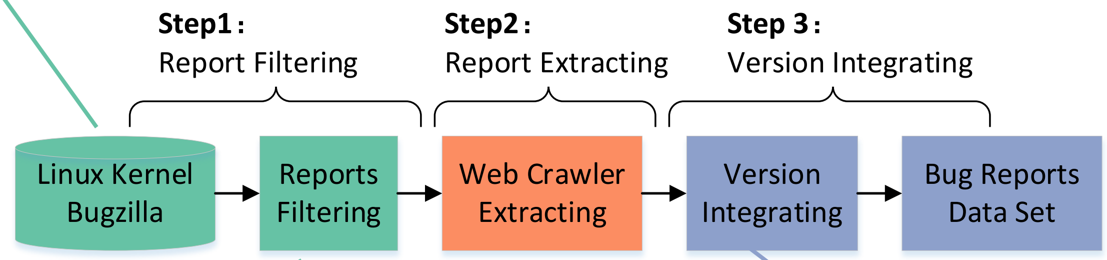
1. Bug Data Collection and Classification
2. Evolution of Bug Type Proportions
3. Evolution of Regression Bugs
4. Bug Types’ Time to Fix
5. Bug Types’ Network Characteristics

Outline

Bug Data Collection

Data source:
https://bugzilla.kernel.org/

Reports were downloaded by a python designed web crawler.



Details of data set

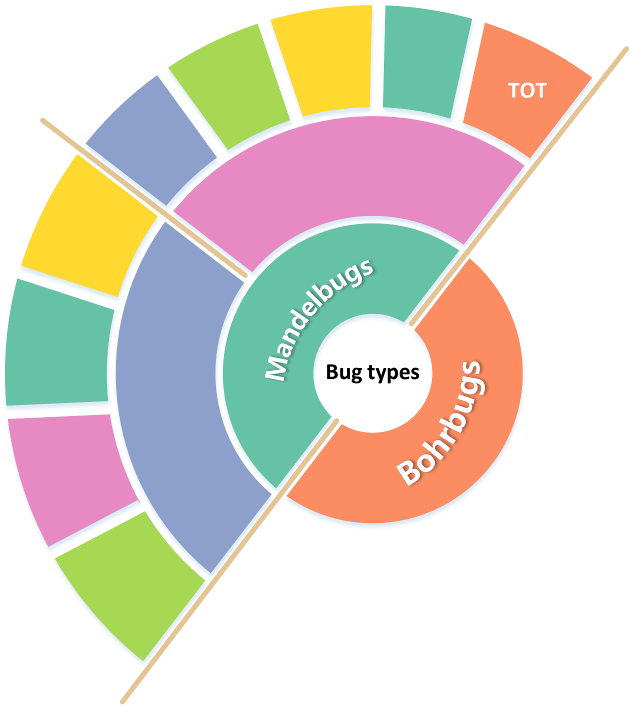
Versions	Products	Reports	Time frame
2.4-4.9	All	5741	Nov 2002 - Nov 2016

Status: **CLOSED**
Resolution: **CODE_FIX**

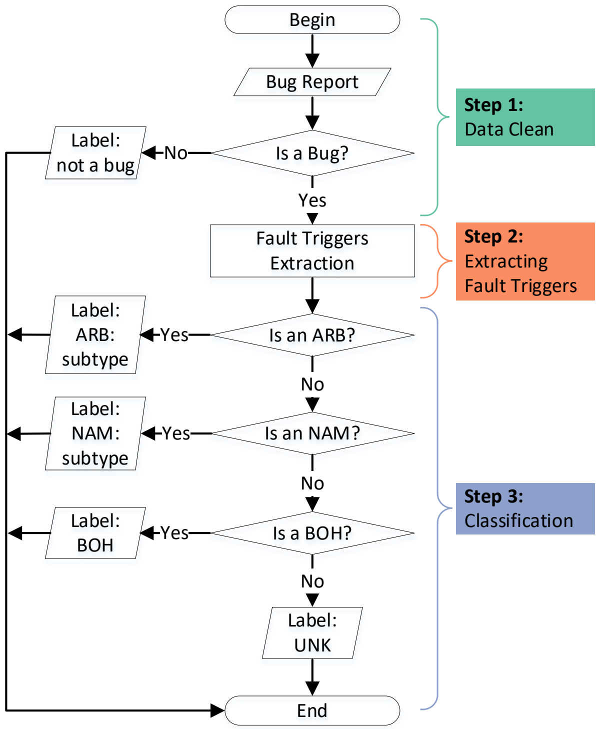
Version integrating, e.g.,
1) Ubuntu 2.6.17-10.34-generic -> 2.6.17
2) 2.6.33.4 -> 2.6.33

Bug Classification

➢ Bug types: **Bohrbug** (**BOH**) and **Mandelbug** (**MAN**).



Bug types: **Bohrbugs** and **Mandelbugs: subtypes**

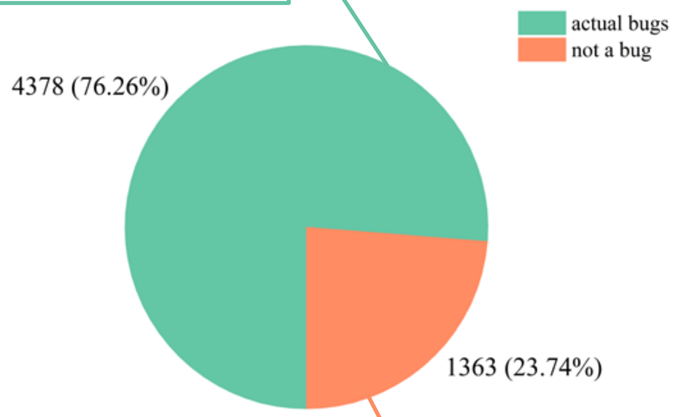


Manual classification procedure

Bug Classification

Total numbers and percentages for actual bugs and not a bug

actual bugs 76.26%



not a bug 23.74%

ID	Type	Description
6045	NAM/TIM	"Using the aic94xx/sas_class driver...,intermittent panic/hang on boot.. due to a race condition between device discovery of the root disk and an attempt to mount the root file system"
7968	BOH	"After booting (and during booting) the keyboard LEDs (NumLock, CapsLock and ScrollLock) don't work (they're always off)."
11805	NAM/ENV	"mounting XFS produces a segfault...When there is no memory left in the system, xfs_buf_get_noaddr() can fail."
12684	NAM/LAG	"After a suspend/resume, and a second suspend, the machine refuses to resume... this could be rectified by forcibly saving and restoring the ACPI non-volatile state"
50181	ARB/MEM	"After 20 hours of uptime, memory usage starts going up..."

Some examples of classified bugs

Outline

1 Bug Data Collection and Classification

2 Evolution of Bug Type Proportions

3 Evolution of Regression Bugs

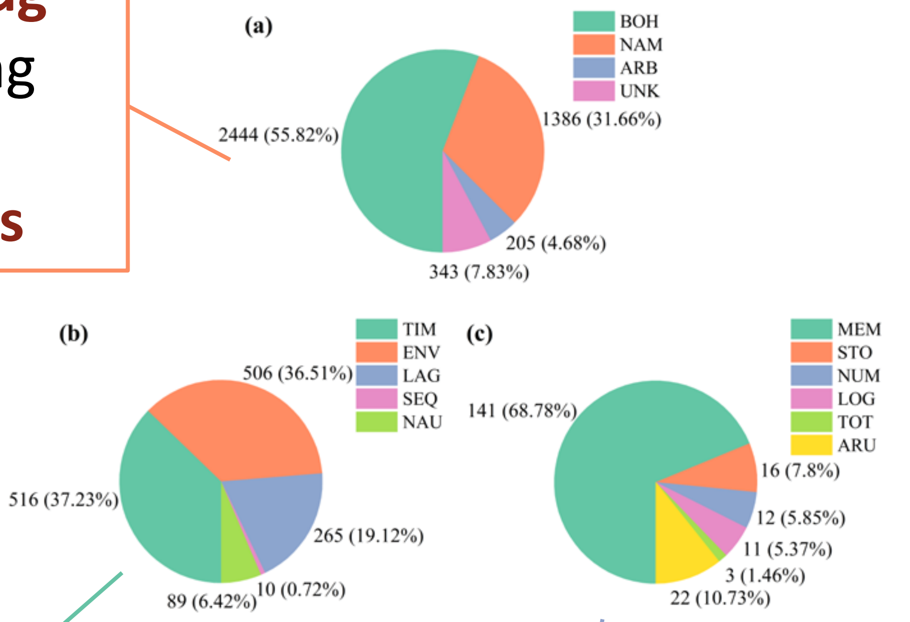
4 Bug Types' Time to Fix

5 Bug Types' Network Characteristics

Evolution of Bug Type Proportions

Overview of Bug Type Proportions

(a) **each bug type** among the 4378 **actual bugs**



(b) **NAM subtypes**

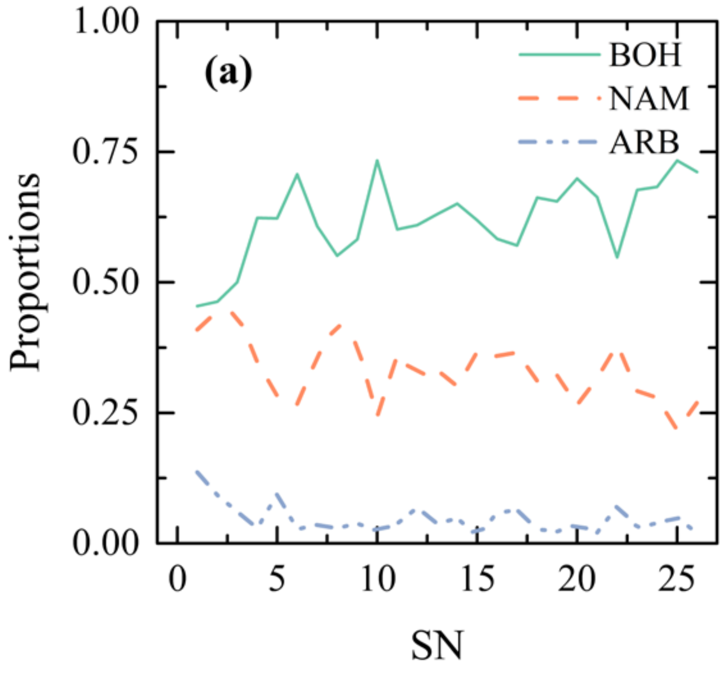
(c) **ARB subtypes**

- **More than 50%** of actual bugs are **BOHs**
- **NAMs**' major subtypes: **TIM**, **ENV** and **LAG**
- **ARBs**' major subtype: **MEM**

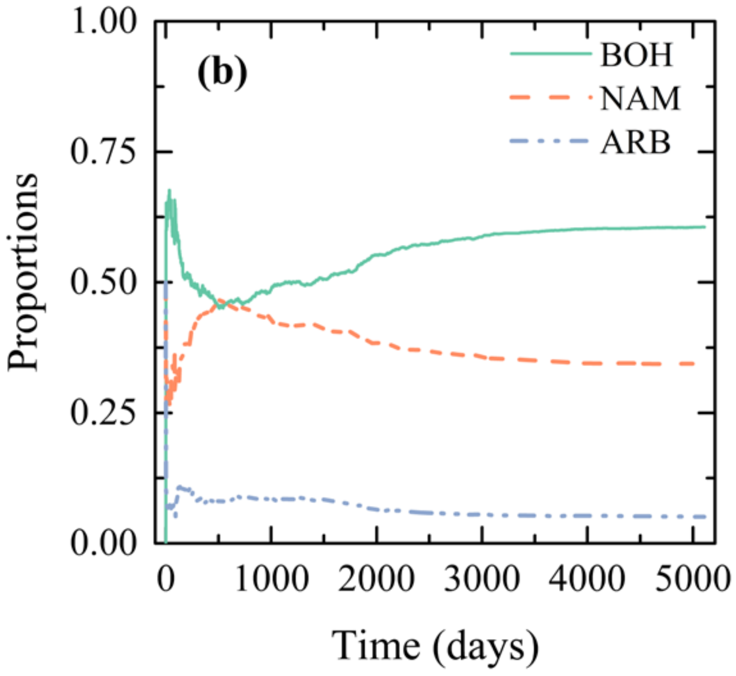
How do proportions of bug types in Linux evolve over versions or time?

Evolution of Bug Type Proportions

Evolution of Bug Type Proportions

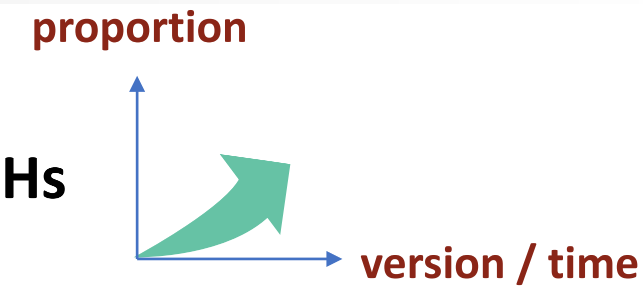


Evolution with versions

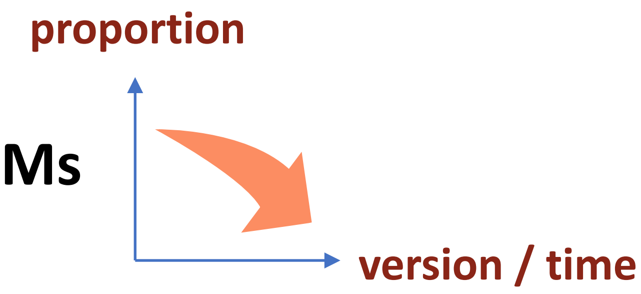


Evolution with time

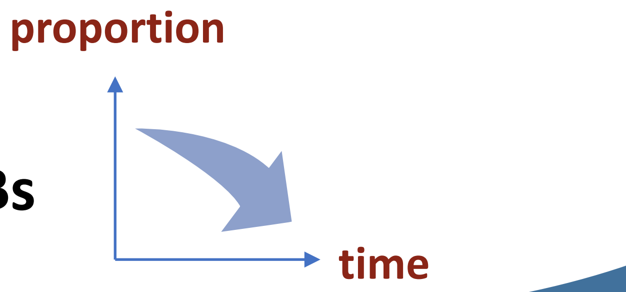
BOHs



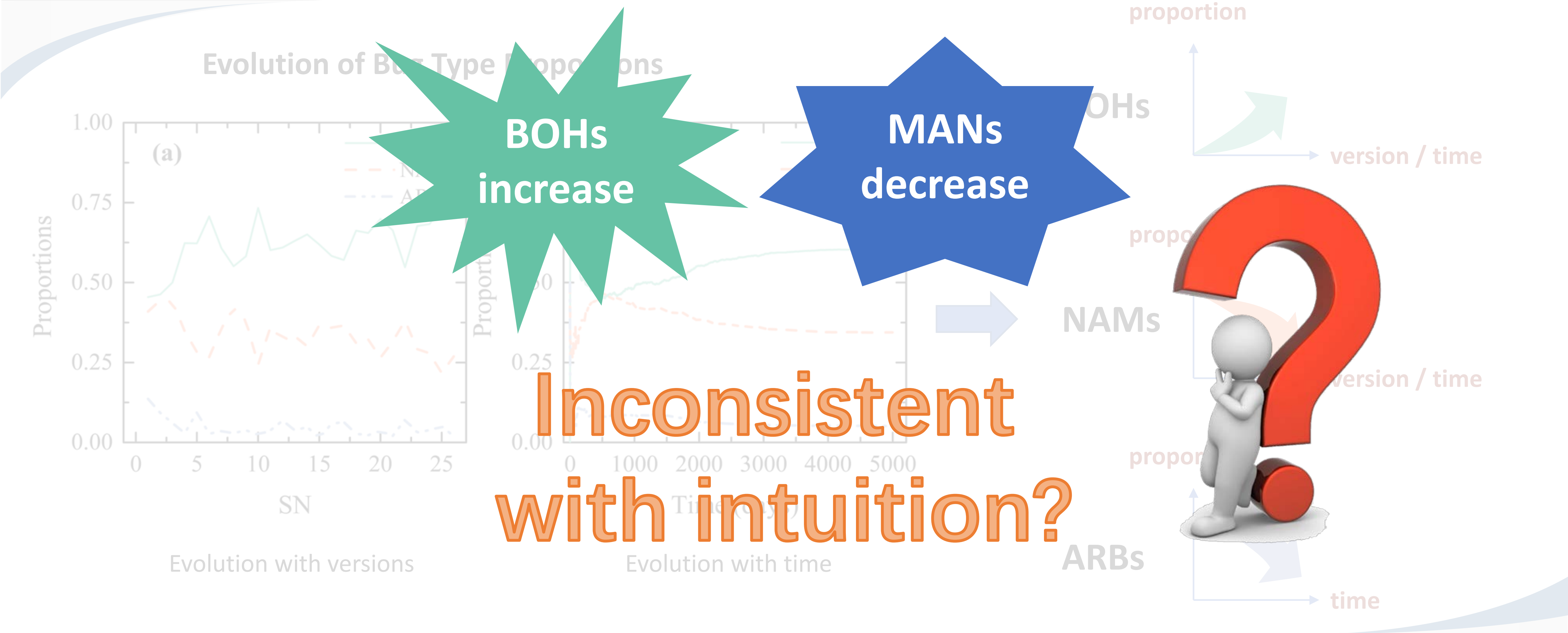
NAMs



ARBs

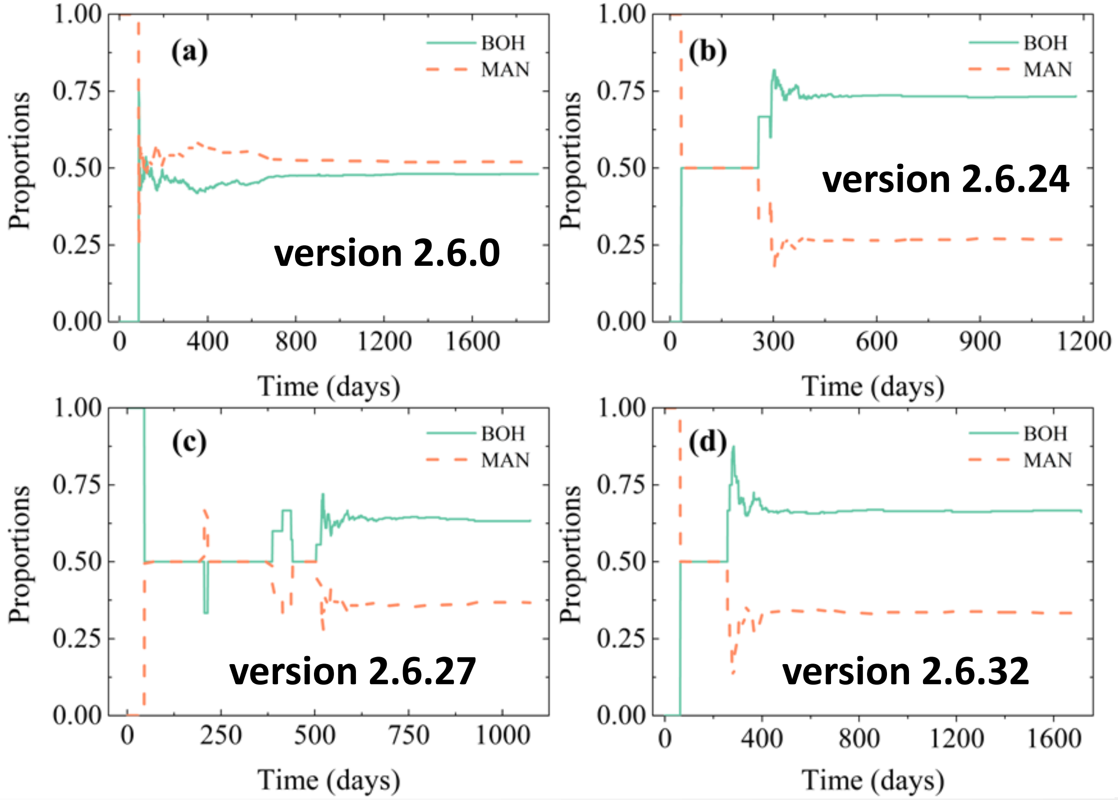


Evolution of Bug Type Proportions



Evolution of Bug Type Proportions

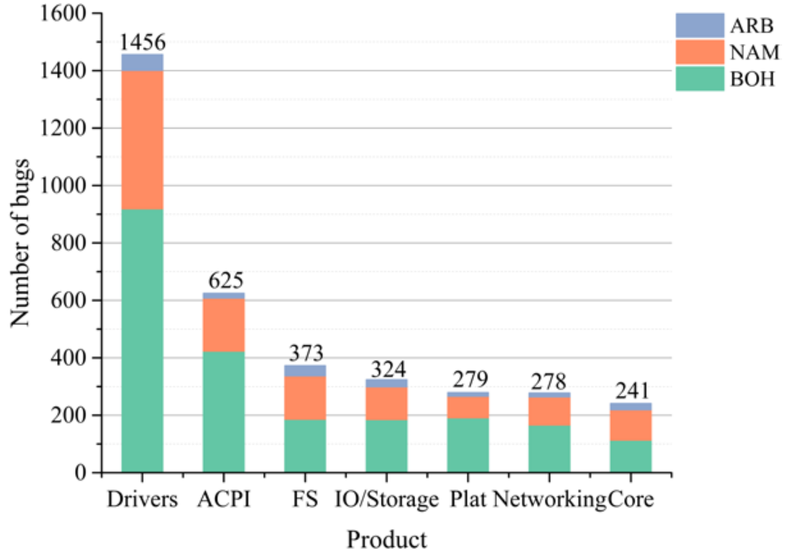
Comparison across four selected versions



- The proportions of **BOHs** and **MANs** and their **evolution trends** are **different** among **different versions**.

Evolution of Bug Type Proportions

*Note: **core** represents products **Memory, Process Management** and **Timers***



Distributions of classified bugs

	BOH	NAM	ARB
Drivers	**1.03**	0.98	0.74
ACPI	**1.11**	0.87	0.53
FS	0.82	**1.20**	**1.90**
IO/Storage	0.94	**1.69**	**1.52**
Plat	**1.12**	0.79	0.92
Networking	0.98	**1.04**	0.99
Core	0.77	**1.30**	**1.79**

Correlation *lift* between **bug types** and **products**

Drivers ACPI Plat → **BOH**

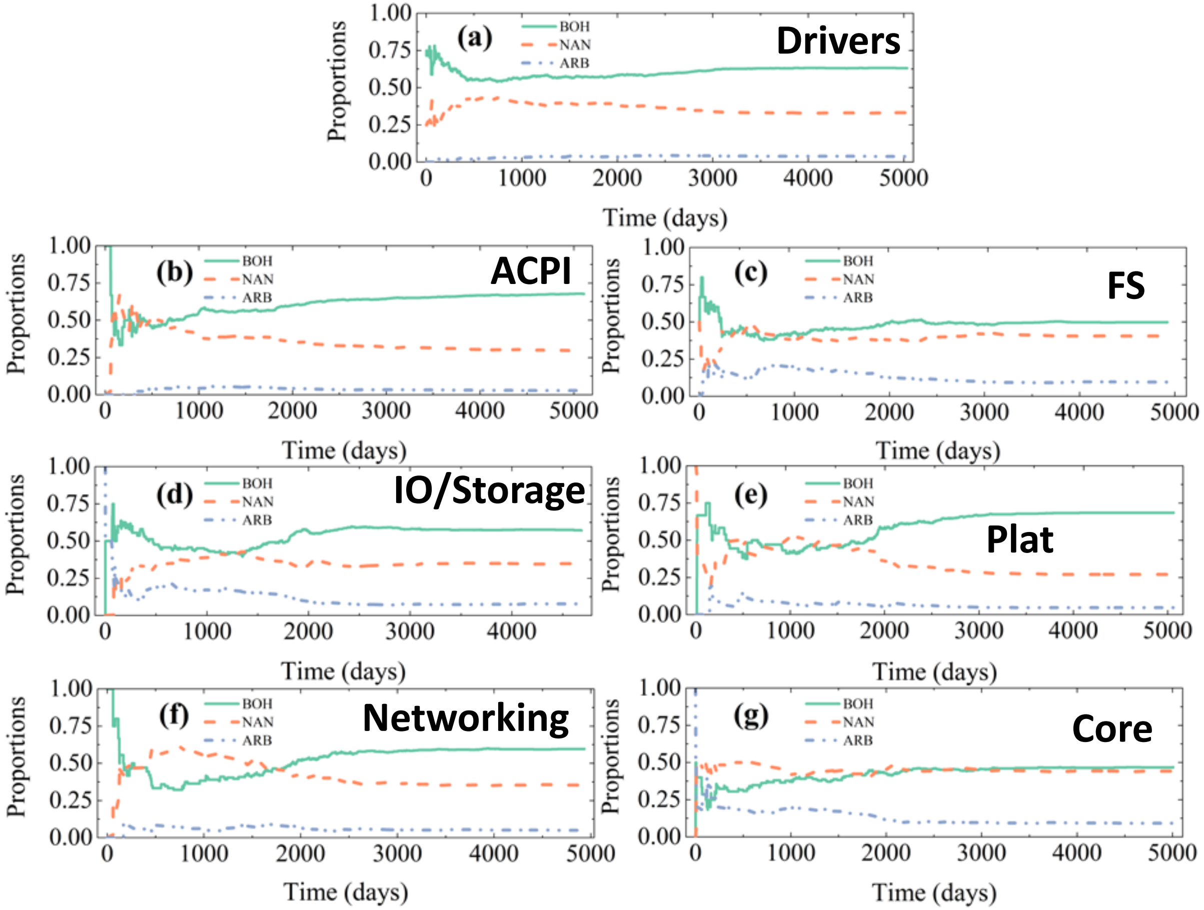
FS IO/Storage Core → **NAM & ARB**

Net → **NAM**

- **More than 50%** of classified bugs are related to **device drivers**

Evolution of Bug Type Proportions

Evolution of bug type proportions of selected products



- The proportions of **BOHs**, **NAMs**, **ARBs** and their **evolution trends** are **different** among **different products**.

Outline

Evolution of Regression Bugs

- **Regression bug**: a bug that causes a **normal feature** to **stop working after a certain event** (e.g., bug fixes, new feature work, etc.).

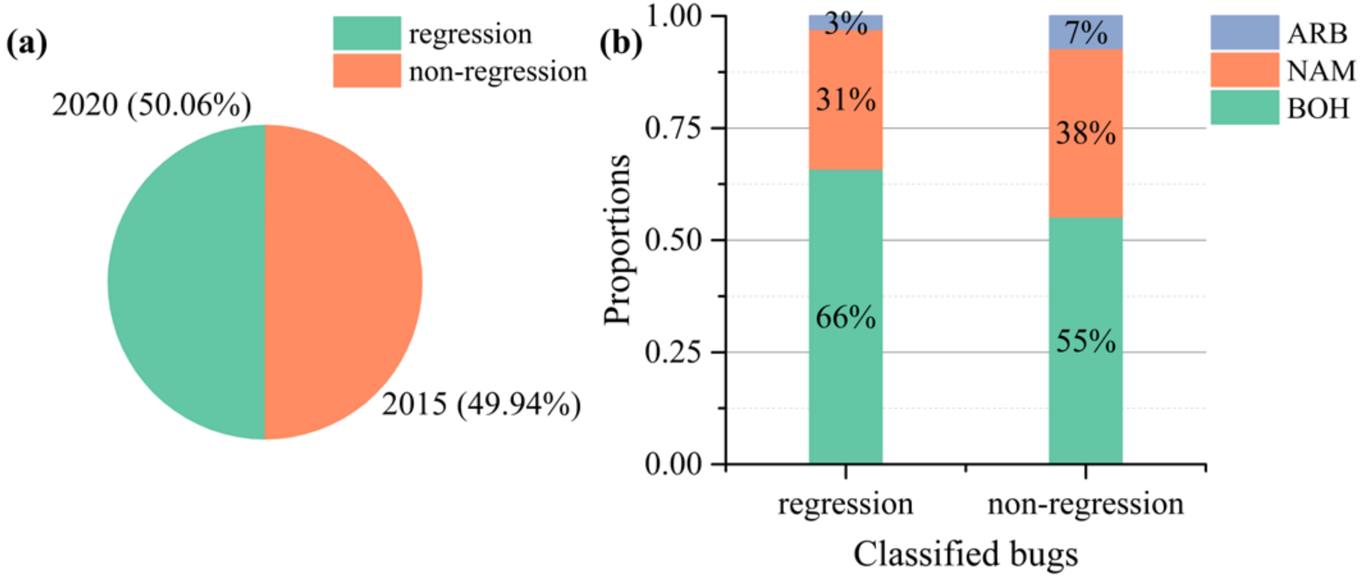
ID	Type	Description
8736	NAM/TIM	"Here is another scenario I bumped onto - qdisc_watchdog_cancel() and qdisc_restart() deadlock ...Please try reverting commit 1936502d0. This one is a regression in 2.6.22"
11329	BOH	"in git1 and previous, cpu0_vid is reported as 1475 (which is correct). Since git2, it is reported, as 725"

Examples of regression bugs

How does the proportion of regression bugs in Linux evolve over versions or time?

Evolution of Regression Bugs

Overview of The Proportion of Regression Bugs



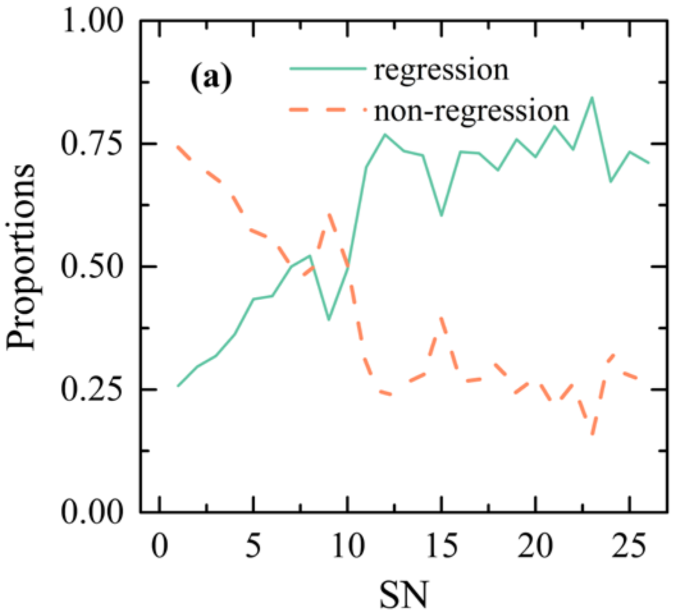
Correlation *lift* between bug types and regressions

	BOH	NAM	ARB
regression	**1.09**	0.91	0.58
non-regression	0.91	**1.09**	**2.00**

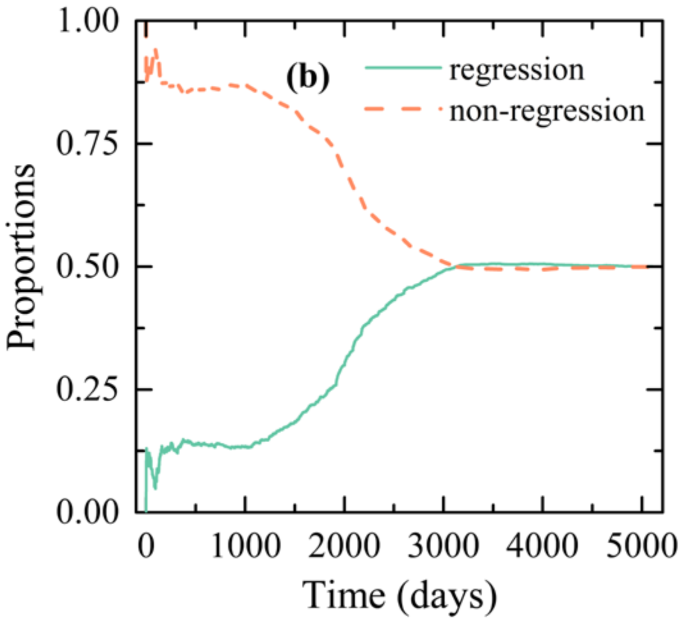
- **More than 50%** of the classified bugs are **regression bugs**.
- **Regression bugs** are more likely **BOHs**, while **non-regression bugs** are more likely **MANs**.

Evolution of Regression Bugs

Evolution of proportions of regression bugs

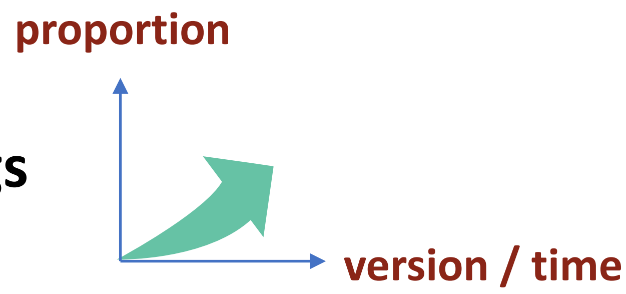


Evolution with versions

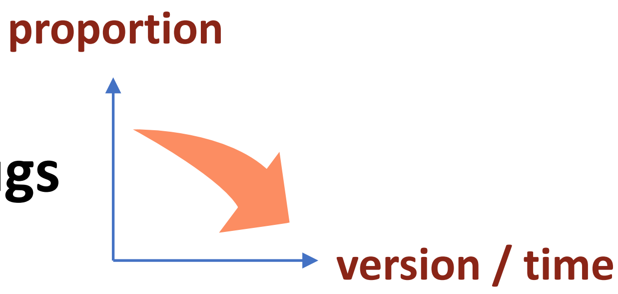


Evolution with time

Regression bugs



Non-regression bugs



Possible reasons: **introduction of new features** **and** **bug fixes**.

Outline

1 Bug Data Collection and Classification

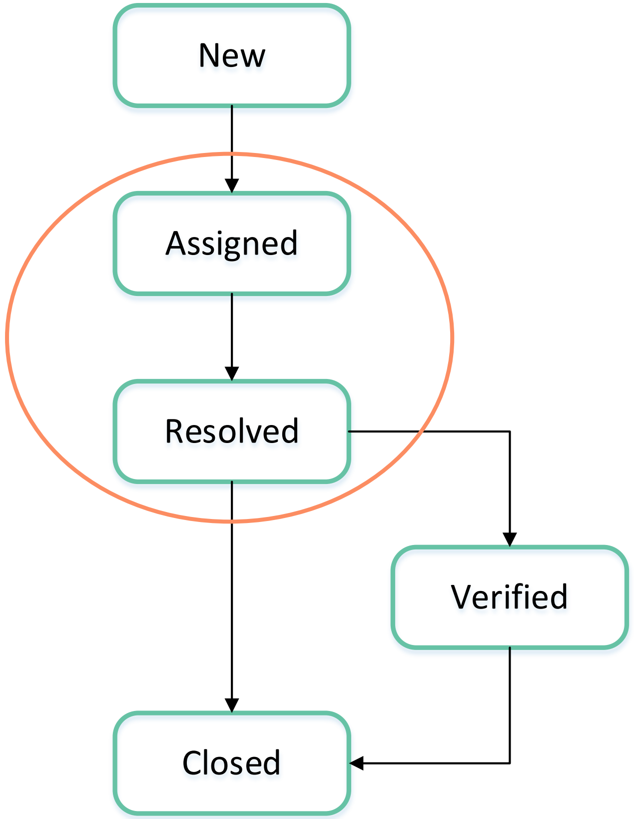
2 Evolution of Bug Type Proportions

3 Evolution of Regression Bugs

4 Bug Types’ Time to Fix

5 Bug Types’ Network Characteristics

Bug Types’ Time to Fix



A simplify life cycle depiction of a “CLOSED” and “CODE_FIX” bug report.

Average time to fix

MANs > BOHs

Bug type	Average time to fix (days)	Standard deviation
BOH	218.63	360.72
MAN	254.22	374.22

Comparison of time to fix for bug types

Outline

1 Bug Data Collection and Classification

2 Evolution of Bug Type Proportions

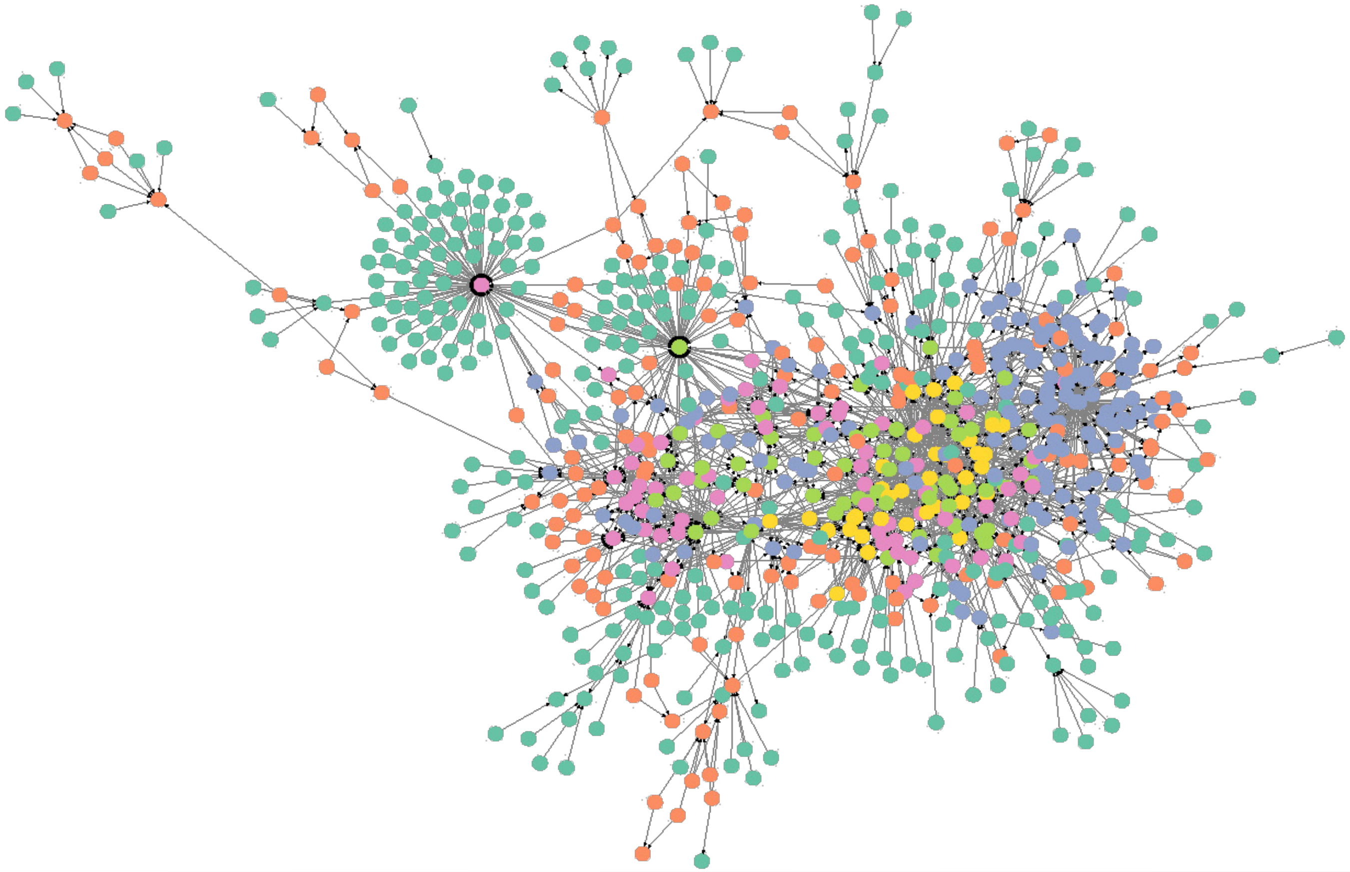
3 Evolution of Regression Bugs

4 Bug Types’ Time to Fix

5 Bug Types’ Network Characteristics

Bug Types' Network Characteristics

Complex Networks



"Scale-free" [*Nature*, *98'*]

"Small-world" [*Science*, *99'*]

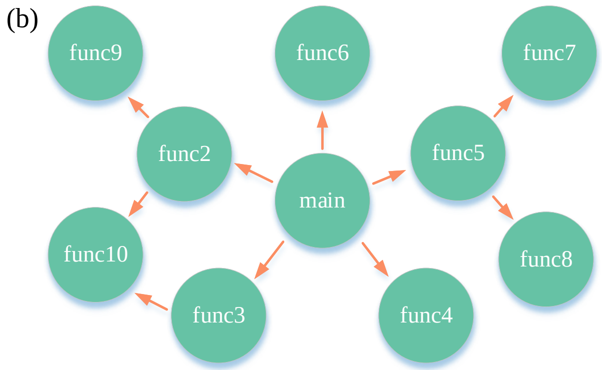
Bug Types' Network Characteristics

- LOS Network

(a)

```
#include<stdio.h>
//a_C_language_program.c
void func2()
{   func9(); func10(); }
void func3()
{   func10(); }
void func5()
{   func7(); func8(); }
int main()
{  func2(); func3(); func4();
   func5(); func6();
   return 0;
}
```

(b)



Nodes: functions, **Edges**: function calls

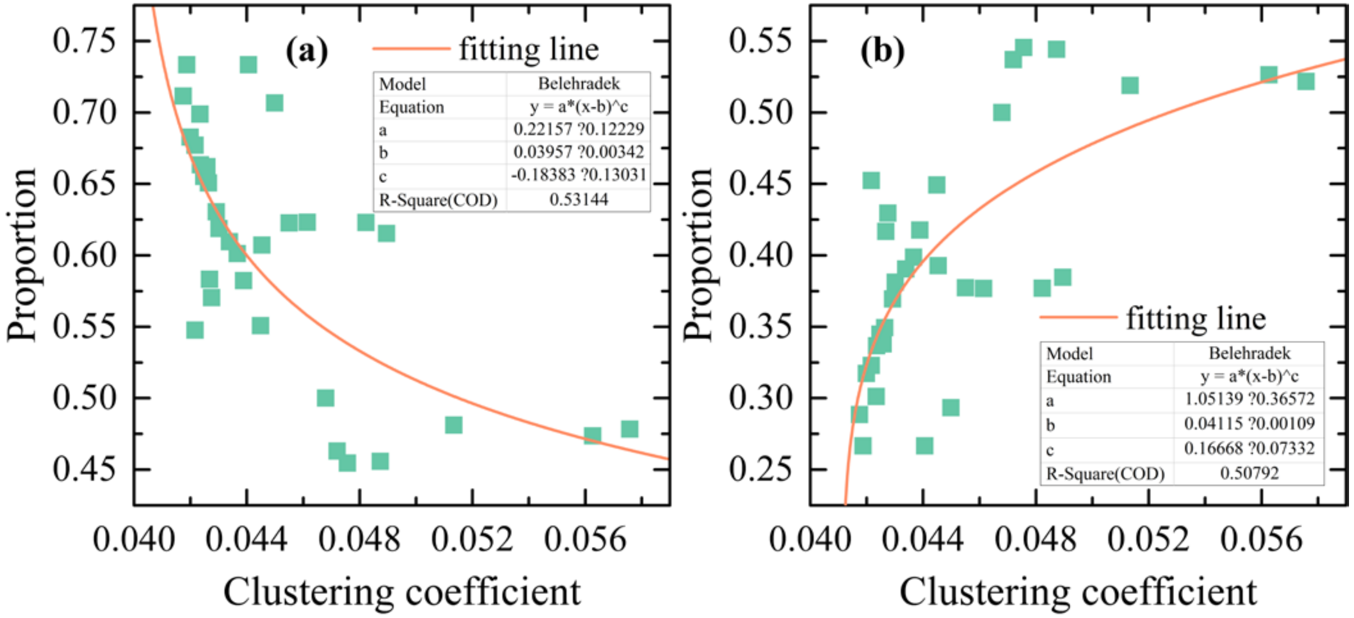
Xiao, G., Zheng, Z., & Wang, H. (2017). **Evolution of Linux operating system network**. *Physica A: Statistical Mechanics and its Applications*, *466*, 249-258.

In one of our previous works, we explored the **evolution of LOS networks**, including **62 major versions** from **versions 1.0 to 4.1**.

Is there a network metric that can reflect the evolution of bug type proportions?

Bug Types’ Network Characteristics

Large clustering coefficient → **Low proportion of BOHs** **High proportion of MANs.**



- **Clustering coefficient** is used to measure the **tendency of a network** to form **tightly connected neighborhoods**.

The **relationship** between **bug type proportions** and **clustering coefficient**

Bug Types’ Network Characteristics

Large clustering coefficient → **Low proportion of BOHs**
High proportion of MANs.

Pearson correlation analysis between **bug type proportions** and **clustering coefficient**

	Clustering coefficient	p value
BOH	-0.68	0.000022
MAN	0.68	0.000022

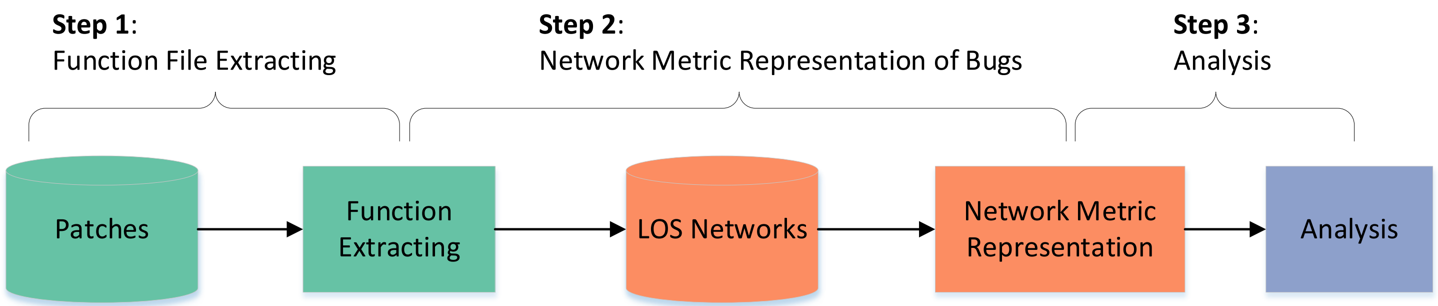
- **Clustering coefficient** is used to measure the **tendency of a network** to form **tightly connected neighborhoods**.

Extension and Future Work

- **RQ1**: What network metrics can be used to represent bug types' characteristics?
- **RQ2**: What's the difference of bug types' characteristics based on network metrics?

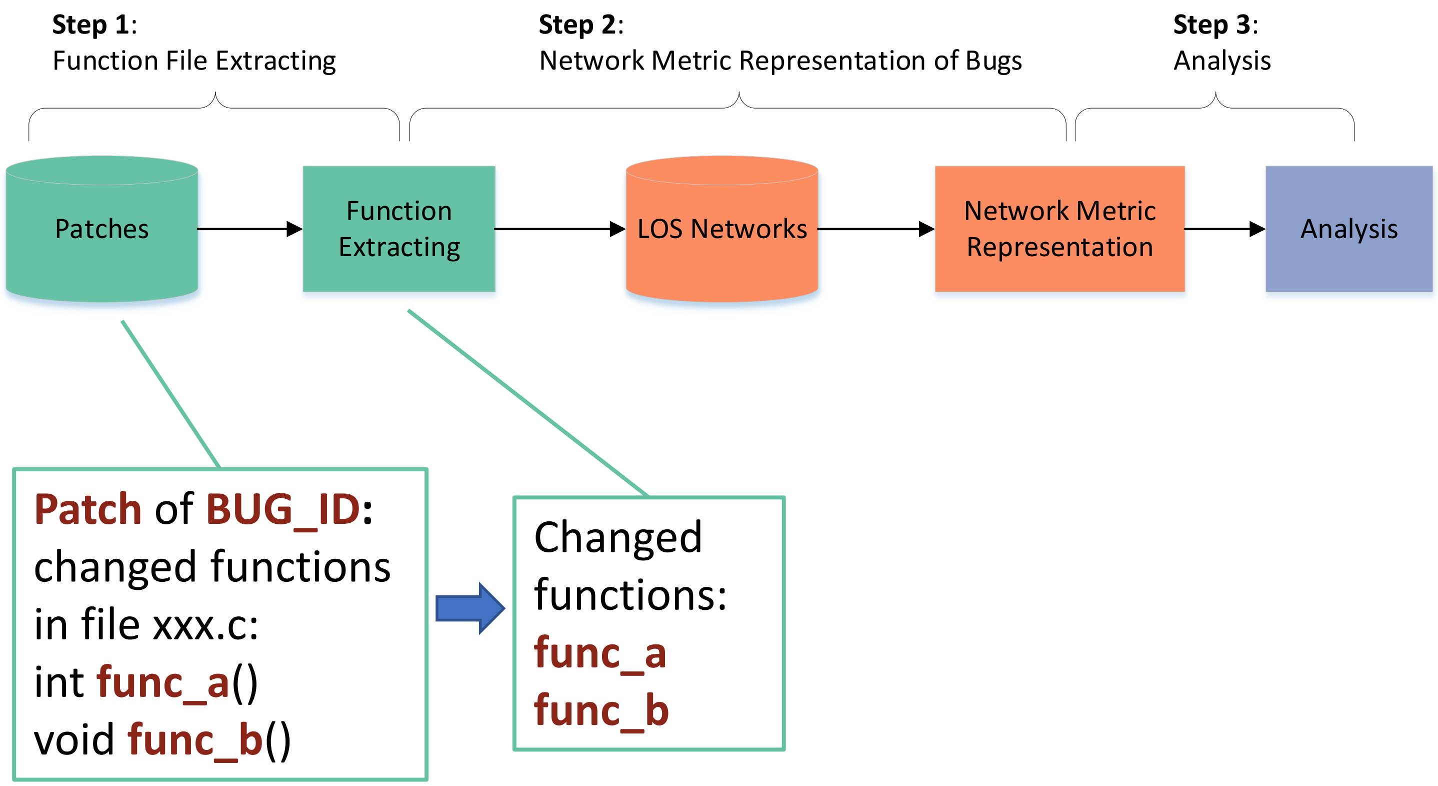
Extension and Future Work

➢ The Analysis Procedure



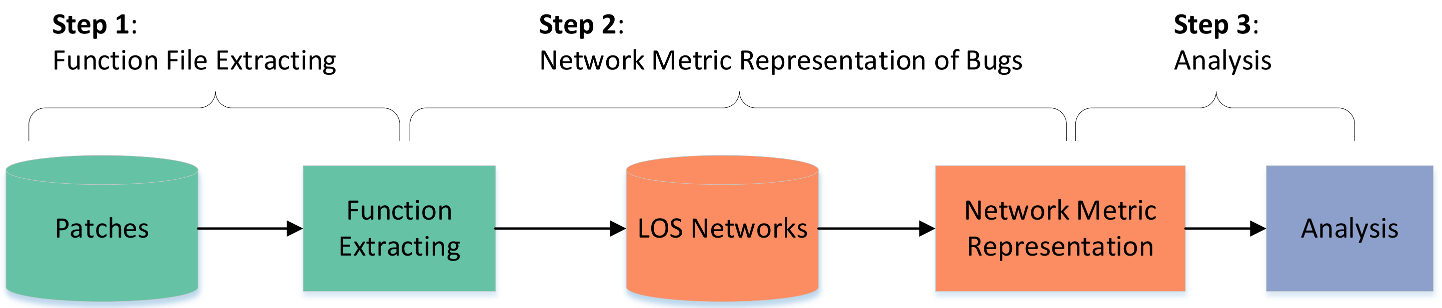
Extension and Future Work

- The Analysis Procedure



Extension and Future Work

➢ The Analysis Procedure



Patch of **BUG_ID**:
changed functions
in file xxx.c:
int **func_a**()
void **func_b**()

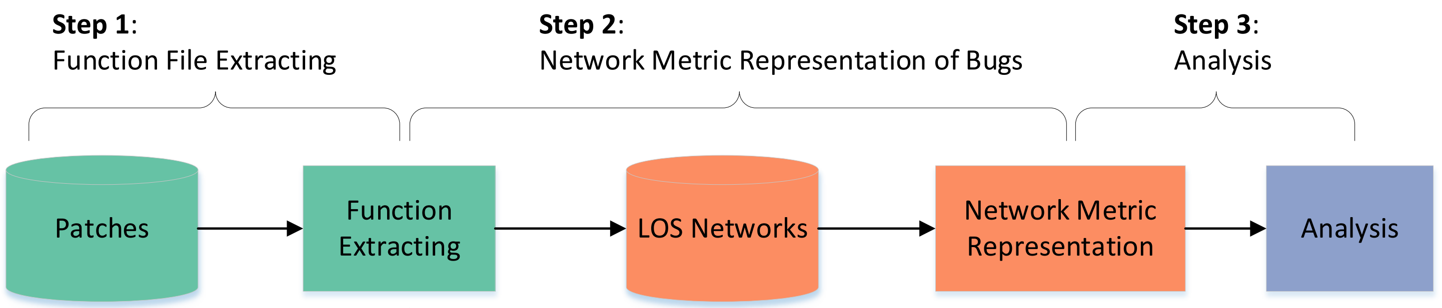
Changed
functions:
func_a
func_b

Network metrics:
e.g., **Degree**
$\boldsymbol{k}$(**func_a**)
$\boldsymbol{k}$(**func_b**)
or **Clustering Coefficient**
$\boldsymbol{C}$(**func_a**)
$\boldsymbol{C}$(**func_b**)

Network metric representation of
BUG_ID:
e.g., **Degree**
$\boldsymbol{k_{sum}}$ (**BUG_ID**) = $\boldsymbol{k}$(**func_a**) + $\boldsymbol{k}$(**func_b**)
or **Clustering Coefficient**
$\boldsymbol{C_{sum}}$ (**BUG_ID**) = $\boldsymbol{C}$(**func_a**) + $\boldsymbol{C}$(**func_b**)

Extension and Future Work

➢ The Analysis Procedure



Difference of the **bug types' characteristics based on network metrics**:
e.g., **Degree**
k_{sum}(**BOHs**) and k_{sum}(**MANs**)
or **Clustering Coefficient**
C_{sum}(**BOHs**) and C_{sum}(**MANs**)

Patch of **BUG_ID:**
changed functions
in file xxx.c:
int **func_a**()
void **func_b**()

Changed
functions:
func_a
func_b

Network metrics:
e.g., **Degree**
k(**func_a**)
k(**func_b**)
or **Clustering Coefficient**
C(**func_a**)
C(**func_b**)

Network metric representation of
BUG_ID:
e.g., **Degree**
k_{sum} (**BUG_ID**) = k(**func_a**) + k(**func_b**)
or **Clustering Coefficient**
C_{sum} (**BUG_ID**) = C(**func_a**) + C(**func_b**)

Extension and Future Work

- Findings:

1. **Degree: k**

   k_{sum}(BOHs) and k_{sum}(MANs) are significant different: k_{sum}(BOHs) < k_{sum}(MANs)
   k_{ave}(BOHs) and k_{ave}(MANs) are significant different: k_{ave}(BOHs) < k_{ave}(MANs)
   k_{max}(BOHs) and k_{max}(MANs) are significant different: k_{max}(BOHs) < k_{max}(MANs)

2. **Clustering Coefficient: C**

   C_{min}(BOHs) and C_{min}(MANs) are significant different: C_{min}(BOHs) > C_{min}(MANs)

3. **Closeness: C_c**

   $C_{c_{ave}}$(BOHs) and $C_{c_{ave}}$(MANs) are significant different: $C_{c_{ave}}$(BOHs) > $C_{c_{ave}}$(MANs)
   $C_{c_{min}}$(BOHs) and $C_{c_{min}}$(MANs) are significant different: $C_{c_{min}}$(BOHs) > $C_{c_{min}}$(MANs)

Relevant Published Papers

- [1] Gao, Y., Zheng, Z., & Qin, F. (2014). **Analysis of Linux kernel as a complex network**. *Chaos, Solitons & Fractals*, *69*, 246-252.

  Analyzed the core component of LOS as a complex network and showed that the large in-degree nodes providing basic services would do more damage on the whole system at the time of intentional attacks.

- [2] Wang, H., Chen, Z., **Xiao, G.**, & Zheng, Z. (2016). **Network of networks in Linux operating system**. *Physica A: Statistical Mechanics and its Applications*, *447*, 520-526.

  Explored the coupling correlations among Linux components and analyzed the impact of system failures on networks

- [3] **Xiao, G.**, Zheng, Z., & Wang, H. (2017). **Evolution of Linux operating system network**. *Physica A: Statistical Mechanics and its Applications*, *466*, 249-258.

  Studied the evolution of 62 major releases of Linux kernel ranging from versions 1.0 to 4.1 on network perspective and revealed the characteristics of the structure and functionality evolution of Linux network.

ISSRE 2017 Toulouse, France

Thank You!

Guanping Xiao: gpxiao@buaa.edu.cn

Zheng Zheng: zhengz@buaa.edu.cn

BEIHANG UNIVERSITY

DUKE UNIVERSITY