



南京邮电大学
Nanjing University of Posts and Telecommunications

MySQL数据库中软件缺陷的实证研究

答辩人：B21031803蒋平



南京邮电大学
Nanjing University of Posts and Telecommunications



目录

CONTENT



研究背景与内容

Research Background and Contents



缺陷的分布特征

Distribution characteristics of defects



缺陷的修复情况

Repair of defects



缺陷的根本原因

The root cause of the defect



总结与展望

Summary and Prospect



南京邮电大学
Nanjing University of Posts and Telecommunications



绪论

Introduction

SECTION ONE



研究背景



研究内容



背景知识

基于MySQL数据库的官方缺陷管理系统 (<https://bugs.mysql.com>) 中的缺陷报告数据，围绕以下三个问题展开实证分析。

1. 缺陷的分布特征

MySQL数据库中的历史缺陷与版本之间是否存在相关性？是如何分布在各个模块中的？以及不同严重程度的缺陷是如何分布的？缺陷所在模块与缺陷严重性之间是否存在相关性？

2. 缺陷的修复情况

MySQL缺陷的修复情况如何？缺陷的严重性是否会影响其修复时长？

3. 缺陷的根本原因


缺陷发生的根本原因是什么？缺陷的根本原因是如何分布的？

背景知识

Background Knowledge

缺陷报告

Showing 1-30 of 35822 (Edit , Save , CSV , Feed)										Show Next 30 Entries »
ID#	Date	Updated	Type	Status	Sev	Version	OS	CPU	Summary	



Search

Login / Register

Developer ZoneBugs HomeReport a bugAdvanced searchSaved searchesTags

Bug #111663

Submitted: 5 Jul 2023 7:38

Reporter: [v v](#)

Status: Closed

Category: MySQL Server: Optimizer

Version: 5.7.42

Assigned to:

Tags: [regression](#)

Same data, same SQL, return different results

Modified: 12 Oct 2023 16:07

Email Updates:

Subscribe

Impact on me:

NoneAffects Me

Severity: S2 (Serious)

OS: CentOS (3.10.0-957.21.2.el7.x86_64)

CPU Architecture: x86

View

Add Comment

Files

Developer

Edit Submission

View Progress Log

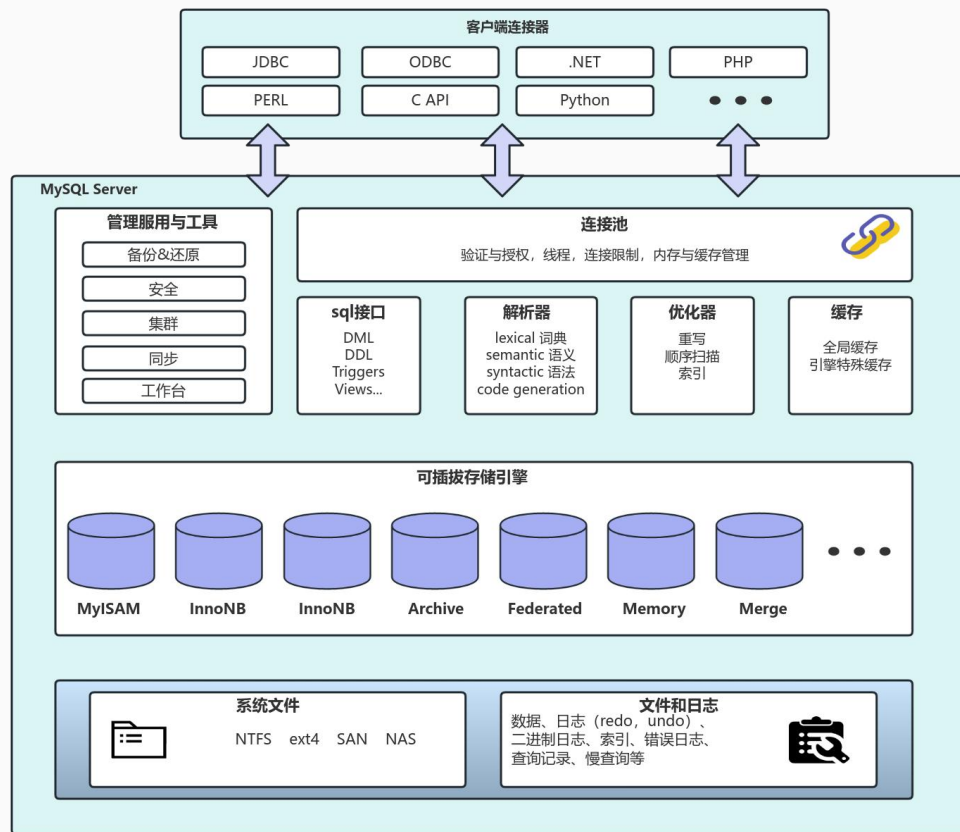
Contributions

[5 Jul 2023 7:38] v v

Description:

Same 4 rows, same SQL, different insert order, return different results!

模块定义



MySQL的缺陷报告中的Type标签标记了各缺陷发生的位置，以便于让对应负责的开发和测试维护人员及时了解并修复缺陷。为了更系统地研究和分析MySQL数据中的缺陷特征，根据MySQL的软件体系结构并结合Type标签中的实际内容，将MySQL系统的缺陷划分为以下九个模块类型。

- 连接模块
- 管理服务与工具模块
- SQL接口模块
- 解析器模块
- 优化器模块
- 缓冲池模块
- 存储引擎模块
- 文件系统模块
- 核心服务模块



缺陷的分布特征

Distribution characteristics of defects

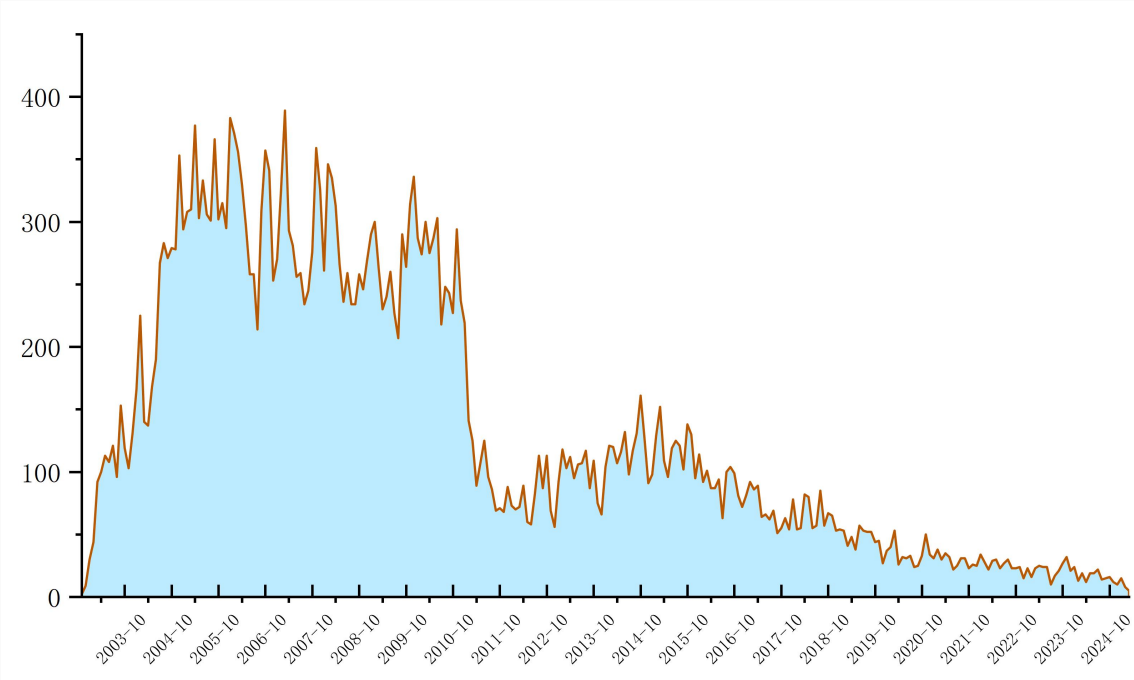
SECTION TWO

- 01 关于时间的演变
- 02 在版本中的分布
- 03 在模块中的分布
- 04 严重性分布
- 05 缺陷严重性与模块的关系

缺陷的分布特征

Distribution characteristics of defects

缺陷在时间上的演变



MySQL中缺陷随时间的演变情况
(以月为单位统计每月提交缺陷数量)

缺陷在版本中的分布

主要版本中的缺陷数量

主要版本号	缺陷数量	比例
1.y.z	3202	8.71%
3.y.z	1167	3.17%
4.y.z	4184	11.38%
5.y.z	25330	68.91%
8.y.z	2814	7.66%
9.y.z	61	0.17%

二级版本中的缺陷数量

版本号	缺陷数量/占比	版本号	缺陷数量/比例
5.1.z	8491/23.10%	4.0.z	954/2.60%
5.0.z	7158/19.47%	1.1.z	892/2.43%
4.1.z	3208/8.73%	1.2.z	681/1.85%
8.0.z	2670/7.26%	3.51.z	510/1.39%
5.2.z	2669/7.26%	5.4.z	247/0.67%
5.7.z	2414/6.57%	3.1.z	232/0.63%
5.6.z	2266/6.16%	1.3.z	208/0.57%
6.0.z	2112/5.75%	3.23.z	148/0.40%
5.5.z	1954/5.32%	3.0.z	123/0.33%
1.0.z	1123/3.06%	1.4.z	116/0.32%

缺陷的分布特征

Distribution characteristics of defects

缺陷的严重性分布

不同严重性的缺陷数量

严重性	缺陷数量/比例
S1	4480/12.52%
S2	8900/24.87%
S3	21883/61.14%
S5	529/1.48%

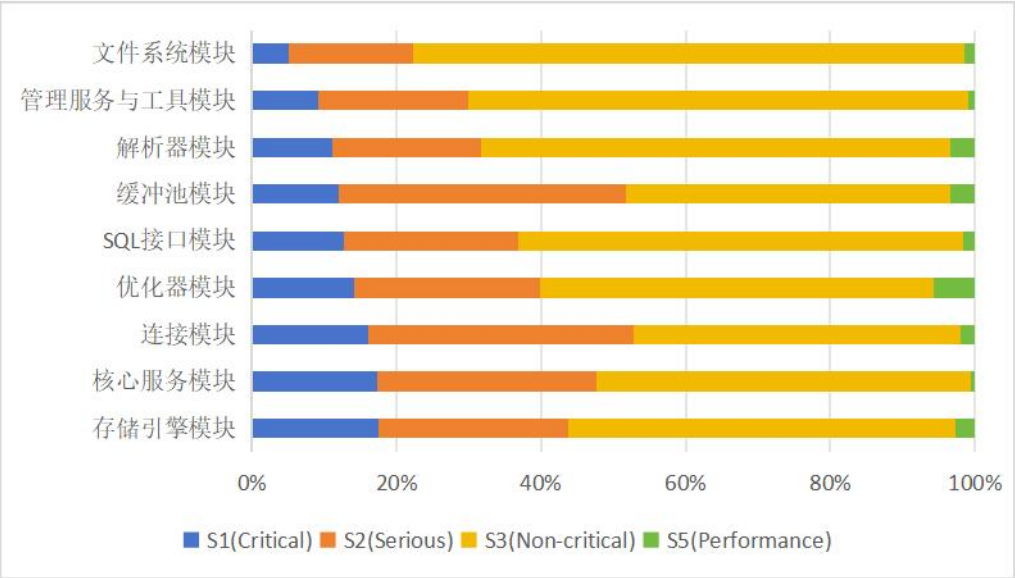
缺陷严重性与模块的关系

定义函数 $f(m,s)$ 为具有指定严重程度的缺陷在模块中所有缺陷中的占比。

$$f(m,s) = \frac{|r \in R|r \text{ 的严重性是 } s|}{|R|}$$

使用如下公式定义任意两个模块 m_1 和 m_2 和之间整体缺陷严重性的顺序。

$$\varphi(m_1,m_2,s) = \begin{cases} >, & f(m_1,s) > f(m_2,s) \\ <, & f(m_1,s) < f(m_2,s) \\ \varphi(m_1,m_2,s+1), & f(m_1,s) = f(m_2,s) \end{cases}$$





南京邮电大学
Nanjing University of Posts and Telecommunications



缺陷的修复情况

Repair of defects

SECTION THREE

缺陷的修复情况

Repair of defects

缺陷修复时长的分布

修复时长（天）	缺陷数量	比例
[0,1]	2351	6.57%
(1,7]	3222	9.00%
(7,30]	5194	14.51%
(30,180]	10649	29.75%
(180,360]	4361	12.18%
(360,∞)	10015	27.98%

缺陷修复时长的统计信息（单位：天）

平均值	中位数	标准差	最大值	最小值
368.04	103.83	672.21	7231.00	0

60%的缺陷能够在半年内得到修复，但修复时长超过一年的缺陷比例未27.98%。
存在因开发者对缺陷认识错误、不够重视、资源限制等原因导致修复时长过长的情况，如Bug#199；
存在少数缺陷已经修复但是未及时关闭，如Bug#6662.

缺陷的严重性与修复时长之间的相关性（总体）

	相关系数	P值
Pearson	0.0023	6.6819e-01
Spearman	-0.0295	2.3312e-08

缺陷的严重性与修复时长之间的相关性（半年内）

	相关系数	P值
Pearson	-0.0398	5.7662e-09
Spearman	-0.0335	9.2182e-07

MySQL数据库软件的缺陷修复时长与缺陷严重程度之间并不存在显著的相关性，这说明缺陷重要性的标记工作还有进一步的提高空间，同时缺陷仓库维护人员应该多关注一些严重性较高的缺陷。



总结与展望

Summary and Prospect

SECTION FIVE

1

分布特征

MySQL数据库历史缺陷主要集中于5.y.z版本中，其中5.0.z和5.1.z两个子版本中缺陷分布较多，这与上述版本中大幅引进新功能有极大关系；管理服务和工具模块中的缺陷最多，但高严重性缺陷比例较低，存储引擎模块缺陷占比和整体严重性都位于前列；MySQL缺陷中S3(Non-critical)级别的缺陷占比最高。

2

修复情况

MySQL数据库中60%的缺陷能够在半年内被修复，然而仍有近27.98%的缺陷修复时长超过一年，整体的平均修复时间偏长；缺陷的修复时长与其严重性之间不存在显著的关联。

3

根本原因

语义错误是MySQL数据库历史缺陷里最主要的根本原因，其中数据处理、控制流类型的错误最为常见；文档问题、环境与配置问题同样占据相当比例，需重点关注。

1 提高缺陷标注机制和自动化识别能力

目前缺陷的根本原因判定仍主要依赖人工判断，未来可引入更完善的缺陷标签体系，同时尝试利用大模型对全体缺陷进行根本原因的分析。对大模型标记根本原因的准确性和有效性进行深入研究，并尝试利用脚本去驱动大模型来实现对所有缺陷的根本原因进行分析，更全面地研究软件缺陷根本原因的分布。提升缺陷属性提取和分类相关工作的效率与客观水平。

2 引入动态运行数据及代码分析工具

本文基于缺陷报告展开相关分析，缺乏对实际代码路径与运行状态的深入探究，后续可搭配静态分析、模糊测试、动态追踪等手段来辅助挖掘深层缺陷逻辑以及修复行为模式。

3 对不同数据库系统缺陷特征做横向比较

将MySQL与其他主流数据库在缺陷分布、修复效率等方面进行比对研究，有利于更全面地评估系统演化的利弊，为高可靠数据库的设计与运维工作提供跨系统借鉴。



南京邮电大学
Nanjing University of Posts and Telecommunications

感谢您的聆听与指正

答辩人：B21031803蒋平