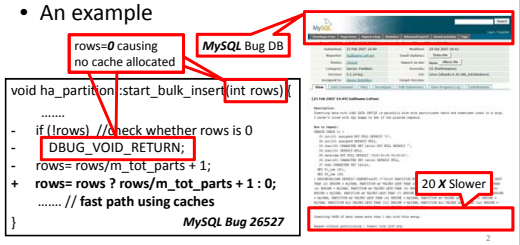
# Statistical Debugging for Real-World Performance Problems

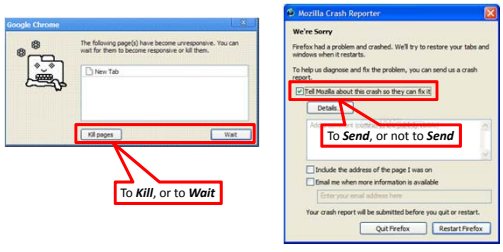
***Linhai Song***[1] and Shan Lu[2]
[1]University of Wisconsin-Madison
[2]University of Chicago

## What are Performance Problems?

- Definition of Performance Problems (PPs):
  - Implementation mistakes causing inefficiency
- An example

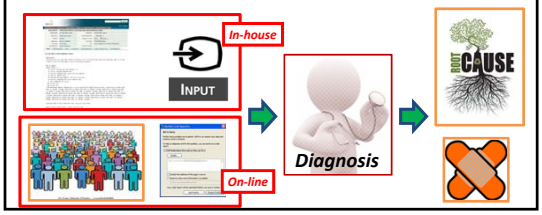



## Performance Diagnosis



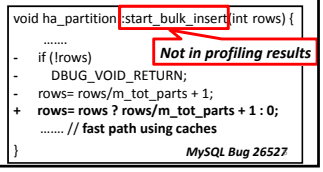

## Performance Diagnosis

- Identifying the causes of performance failures
  - In-house and on-line diagnosis




## Performance Diagnosis is Challenging

- The state of the art is ***preliminary***
  - Profilers
    - Only tools mentioned in bug reports we studied
    - Output time-consuming functions, not root causes
- More effective tools are necessary

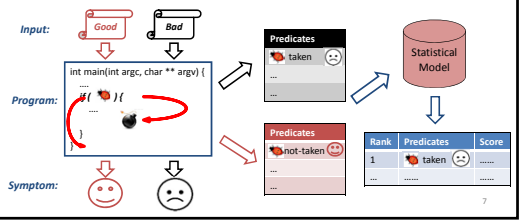



## How Can We Do Better?

***Can we learn from functional bug diagnosis?***

## How to Diagnose Functional Bugs

- The state of the art is ***mature***
- Statistical Debugging(SD)
  - Among the most effective




7

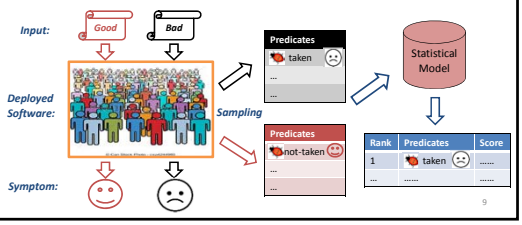
## Remember these examples?

```
1 // Print_tokens2 v7
2 if(ch == '\n')
3    return (TRUE);
4 else if(ch == ' ')
5 // Bug: should return FALSE
6    return (TRUE);
7 else
8    return (FALSE);
```

8

## How to Diagnose Functional Bugs

- The state of the art is ***mature***
- Statistical Debugging(SD)
  - Among the most effective




9

## Apply SD to Performance Diagnosis?

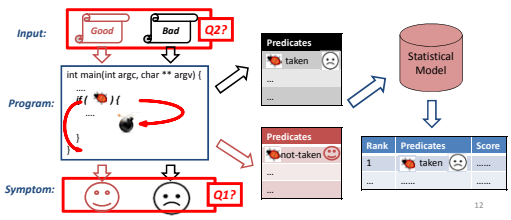
***Since statistical debugging is effective for functional diagnosis, maybe it will also be effective for performance diagnosis.***

## What are the challenges?

11

## Is it Feasible?

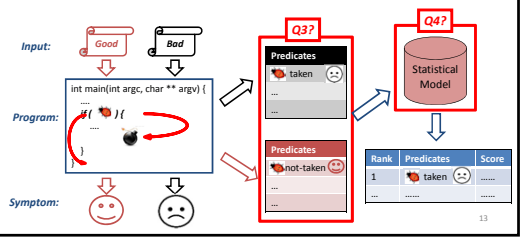
- Q1: How to tell success runs from failure runs?
- Q2: How to obtain good and bad inputs?




12

## How to Apply?

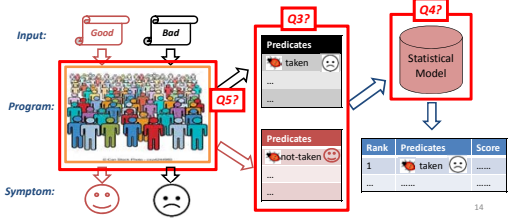
- Q3: What predicates to collect?
- Q4: What statistical model to use?




13

## How to Apply?

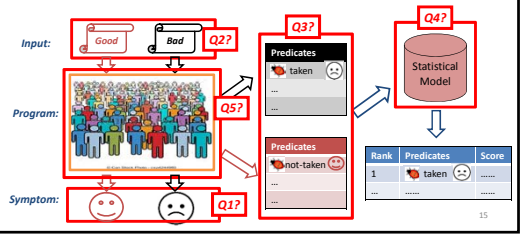
- Q3: What predicates to collect?
- Q4: What statistical model to use?
- Q5: How to do on-line performance diagnosis?




14

## Contributions (I)

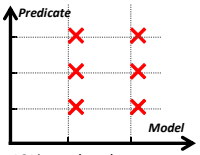
***We answer all these questions by studying 65 user-reported performance bugs.***




15

## Contributions (II)

- Is it feasible to apply SD to PPs?
  - Easy to tell failure runs based on users' reports (Q1)
  - Inputs are provided during reporting (Q2)
- How to apply SD to PPs?
  - In-house diagnosis
    - 3 predicates (Q3)
    - 2 statistical models (Q4)
  - On-line diagnosis (Q5)
    - Same diagnosis capability with <10% overhead
    - Not sacrifice diagnosis latency (**Unique**)




## Outline

- Overview
- Is it feasible to apply SD for PPs?
- How to conduct SD for PPs?
  - In-house diagnosis scenario
  - On-line diagnosis scenario
- Conclusion

17

## Methodology

- Application and Bug Source

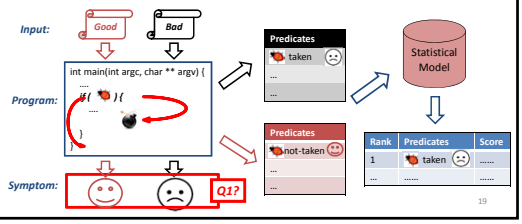
| App. | # Bugs[1] | # Bug User Perceived |
|---|---|---|
| ***Apache*** | 25 | 16 |
| ***Chrome*** | 10 | 5 |
| ***GCC*** | 11 | 9 |
| ***Mozilla*** | 36 | 19 |
| ***MySQL*** | 28 | 17 |
| Total: 110 | | 65 |

[1] Guoliang Jin, Linhai Song, Xiaoming Shi, Joel Scherpelz, and Shan Lu. Understanding and Detecting Real-World Performance Bugs. In PLDI'2012.

18

## Is It Feasible? (Part I)

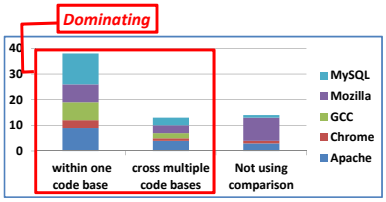
- **Q1: How to tell success runs from failure runs?**
  - A large workload? Or inefficient implementation?




19

## Q1: How to Identify Failure Runs?

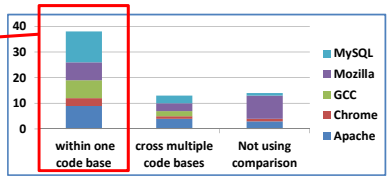
- The majority is observed through comparison




20

## Comparison within One Code Base

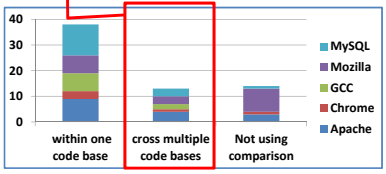
- the same input with different ***configuration***
- inputs with different ***sizes***
- inputs with slightly different ***functionality***




21

## Comparison across Multiple Code Bases

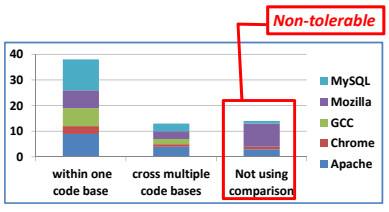
- same applications' different ***versions***
- different ***applications***




22

## Not Using Comparison

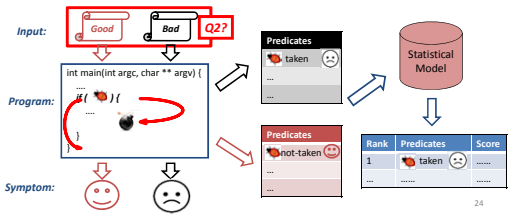
***Mozilla#299742***: "it frozen the GUI to crawl"
***Mozi*** ***Easy to tell failures from successes!*** h the page"
***MySQL46461***: "causing the test suit to fail due to timeout"




23

## Is It Feasible? (Part II)

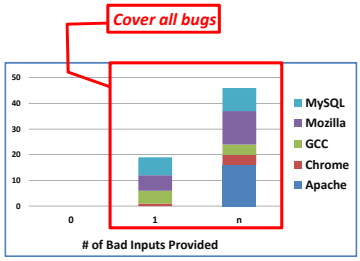
- Q1: How to tell success runs from failure runs?
- **Q2: How to obtain good and bad inputs?**




24

## How to Obtain Bad Inputs?

- Bad inputs are provided in all bug reports



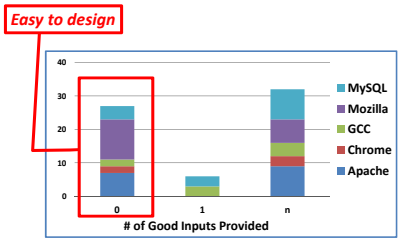

25

## How to Obtain Good Inputs? (I)

- 60% contain good inputs


26

## How to Obtain Good Inputs? (II)




27

## Other Findings and Implications

- Compared with functional bugs
  - More PPs observed through comparison
  - More PPs reported with good inputs
- Implications for SD
  - Easy to tell success runs from failure runs
  - Similar good inputs are provided
  - SD is a nature fit for PPs

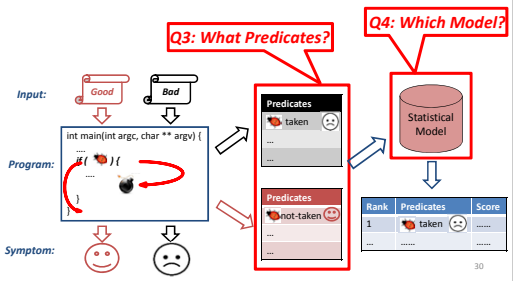
28

## Outline

- Overview
- Is it feasible to apply SD for PPs?
- How to conduct SD for PPs?
  - In-house diagnosis scenario
  - On-line diagnosis scenario
- Conclusion

29

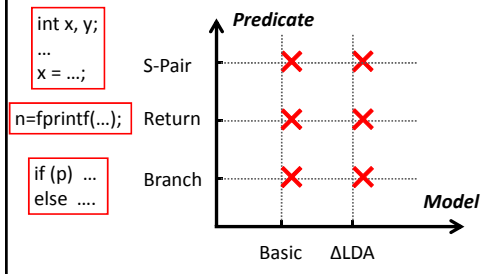
## In-house Statistical Debugging




30

## Design

- Study widely used predicates and models




## Experimental Methodology

- Experimental setting
  - 10 success runs vs. 10 failure runs
  - 20 unique inputs
- Techniques under comparison
  - CBI[2, 3] for C programs
  - Pin for C++ programs
  - Compared with profiling results from OProfile

[2] Ben Liblit, Alex Aiken, Alice X Zhen, and Michael I Jordan. Bug Isolation via Remote Program Sampling. In PLDI'2003.
[3] Ben Liblit, Mayur Naik, Alice X Zheng, Alex Aiken, and Michael I Jordan. Scalable Statistical Bug Isolation. In PLDI'2005.

## Experimental Results

| | Basic Model | | | ΔLDA | | | Profiler |
|---|---|---|---|---|---|---|---|
| BugID | Branch | Return | S-pair | Branch | Return | S-pair | |
| Mozilla258793 | √1 | - | / | - | - | / | - |
| Mozilla299742 | √1 | - | / | - | - | / | - |
| Mozilla347306 | - | - | - | √1 | √1 | √1 | √1 |
| Mozilla416628 | - | - | - | √1 | - | √1 | √1 |
| MySQL15811 | - | - | / | √1 | √1 | / | √1 |
| MySQL26527 | √1 | - | / | - | - | / | - |
| MySQL27287 | - | - | / | √1 | - | / | √1 |
| MySQL40337 | √1 | - | / | - | - | / | - |
| MySQL42649 | √1 | - | / | - | - | / | - |
| MySQL44723 | √1 | - | / | - | - | / | - |
| Apache3278 | √1 | √1 | √1 | - | - | - | - |
| Apache34464 | - | - | - | √3 | √1 | - | √5 |
| ... | ... | ... | ... | ... | ... | ... | ... |

## Experimental Results

| | Basic Model | | | ΔLDA | | | Profiler |
|---|---|---|---|---|---|---|---|
| BugID | Branch | Return | S-pair | Branch | Return | S-pair | |
| Mozilla258793 | √1 | - | / | - | - | / | - |
| Mozilla299742 | √1 | - | / | - | - | / | - |
| Mozilla347306 | - | - | - | √1 | √1 | √1 | √1 |
| Mozilla416628 | - | - | - | √1 | - | √1 | √1 |
| MySQL15811 | - | - | / | √1 | √1 | / | √1 |
| MySQL26527 | √1 | - | / | - | - | / | - |
| MySQL27287 | - | - | / | √1 | - | / | √1 |
| MySQL40337 | √1 | - | / | - | - | / | - |
| MySQL42649 | √1 | - | / | - | - | / | - |
| MySQL44723 | √1 | - | / | - | - | / | - |
| Apache3278 | √1 | √1 | √1 | - | - | - | - |
| Apache34464 | - | - | - | √3 | √1 | - | √5 |
| ... | ... | ... | ... | ... | ... | ... | ... |

*Most Useful*

## Experimental Results

| | Basic Model | | | ΔLDA | | | Profiler |
|---|---|---|---|---|---|---|---|
| BugID | Branch | Return | S-pair | Branch | Return | S-pair | |
| Mozilla258793 | √1 | - | / | - | - | / | - |
| Mozilla299742 | √1 | - | / | - | - | / | - |
| Mozilla347306 | - | - | - | √1 | √1 | √1 | √1 |
| Mozilla416628 | - | - | - | √1 | - | √1 | √1 |
| MySQL15811 | - | - | / | √1 | √1 | / | √1 |
| MySQL26527 | √1 | - | / | - | - | / | - |
| MySQL27287 | - | - | / | √1 | - | / | √1 |
| MySQL40337 | √1 | - | / | - | - | / | - |
| MySQL42649 | √1 | - | / | - | - | / | - |
| MySQL44723 | √1 | - | / | - | - | / | - |
| Apache3278 | √1 | √1 | √1 | - | - | - | - |
| Apache34464 | - | - | - | √3 | √1 | - | √5 |
| ... | ... | ... | ... | ... | ... | ... | ... |

## Experimental Results

| | Basic Model | | | ΔLDA | | | Profiler |
|---|---|---|---|---|---|---|---|
| **BugID** | Branch | Return | S-pair | Branch | Return | S-pair | |
| Mozilla258793 | √1 | - | / | - | - | / | - |
| Mozilla299742 | √1 | - | / | - | - | / | - |
| Mozilla347306 | - | - | - | √1 | √1 | √1 | √1 |
| Mozilla416628 | - | - | - | √1 | - | √1 | √1 |
| MySQL15811 | - | - | / | √1 | √1 | / | √1 |
| MySQL26527 | √1 | - | / | - | - | / | - |
| MySQL27287 | - | - | / | √1 | - | / | √1 |
| MySQL40337 | √1 | - | / | - | - | / | - |
| MySQL42649 | √1 | - | / | - | - | / | - |
| MySQL44723 | √1 | - | / | - | - | / | - |
| Apache3278 | √1 | √1 | √1 | - | - | - | - |
| Apache34464 | - | - | - | √3 | √1 | - | √5 |
| ... | ... | ... | ... | ... | ... | ... | ... |

37

## Experimental Results

| | Basic Model | | | ΔLDA | | | Profiler |
|---|---|---|---|---|---|---|---|
| **BugID** | **Branch** | Return | S-pair | Branch | Return | S-pair | |
| Mozilla258793 | √1 | - | / | - | - | / | - |
| Mozilla299742 | √1 | - | / | - | - | / | - |
| Mozilla347306 | - | - | - | √1 | √1 | √1 | √1 |
| Mozilla416628 | - | - | - | √1 | - | √1 | √1 |
| MySQL15811 | - | - | / | √1 | √1 | / | √1 |
| MySQL26527 | √1 | - | / | - | - | / | - |
| MySQL27287 | - | - | / | √1 | - | / | √1 |
| MySQL40337 | √1 | - | / | - | - | / | - |
| MySQL42649 | √1 | - | / | - | - | / | - |
| MySQL44723 | √1 | - | / | - | - | / | - |
| Apache3278 | √1 | √1 | √1 | - | - | - | - |
| Apache34464 | - | - | - | √3 | √1 | - | √5 |
| ... | ... | ... | ... | ... | ... | ... | ... |

38

## Experimental Results

| | Basic Model | | | ΔLDA | | | Profiler |
|---|---|---|---|---|---|---|---|
| **BugID** | Branch | Return | S-pair | **Branch** | Return | S-pair | |
| Mozilla258793 | √1 | - | / | - | - | / | - |
| Mozilla299742 | √1 | - | / | - | - | / | - |
| Mozilla347306 | - | - | - | √1 | √1 | √1 | √1 |
| Mozilla416628 | - | - | - | √1 | - | √1 | √1 |
| MySQL15811 | - | - | / | √1 | √1 | / | √1 |
| MySQL26527 | √1 | - | / | - | - | / | - |
| MySQL27287 | - | - | / | √1 | - | / | √1 |
| MySQL40337 | √1 | - | / | - | - | / | - |
| MySQL42649 | √1 | - | / | - | - | / | - |
| MySQL44723 | √1 | - | / | - | - | / | - |
| Apache3278 | √1 | √1 | √1 | - | - | - | - |
| Apache34464 | - | - | - | √3 | √1 | - | √5 |
| ... | ... | ... | ... | ... | ... | ... | ... |

39

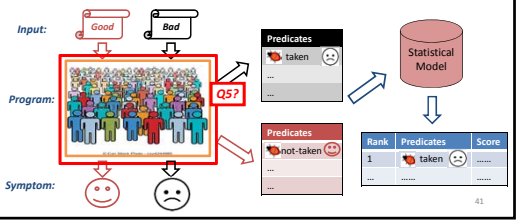
## Outline

- Overview
- Is it feasible to apply SD for PPs?
- How to conduct SD for PPs?
  - In-house diagnosis scenario
  - On-line diagnosis scenario
- Conclusion

40

## On-line Statistical Debugging

- Q5: How to do on-line performance diagnosis?
  - Less information from one single run
  - Diagnosis capability relies on multiple runs info.




41

## Experimental Methodology

- Tool design
  - CBI in sampling mode for C benchmarks
  - LBR for C++ benchmarks
- Benchmarks
  - Reuse benchmarks from in-house experiments
  - Most effective predicate and model
- Experiment design
  - Default sampling rate is roughly 1/10000
  - 1000 success runs and 1000 failure runs

42

## Experimental Results

| BugID | Diagnosis Capability | Run-time Overhead | Requested Failure Runs |
|---|---|---|---|
| Mozilla258793 | √1 | 2.39% | 100 |
| Mozilla299742 | √1 | 4.27% | 500 |
| Mozilla347306 | √1 | 1.42% | 10 |
| Mozilla416628 | √1 | 2.03% | 10 |
| MySQL15811 | √1 | 2.25% | 10 |
| MySQL26527 | √1 | 6.05% | 500 |
| MySQL27287 | √1 | 3.02% | 10 |
| MySQL40337 | √1 | 2.69% | 100 |
| MySQL42649 | √2 | 6.10% | 500 |
| MySQL44723 | √1 | 3.16% | 100 |
| Apache3278 | - | 0.23% | >1000 |
| Apache34464 | √3 | 0.18% | 10 |
| ... | ... | ... | ... |

43

## Experimental Results

| BugID | Diagnosis Capability | Run-time Overhead | Requested Failure Runs |
|---|---|---|---|
| Mozilla258793 | √1 | 2.39% | 100 |
| Mozilla299742 | √1 | 4.27% | 500 |
| Mozilla347306 | √1 | 1.42% | 10 |
| Mozilla416628 | √1 | 2.03% | 10 |
| MySQL15811 | √1 | 2.25% | 10 |
| MySQL26527 | √1 | 6.05% | 500 |
| MySQL27287 | √1 | 3.02% | 10 |
| MySQL40337 | √1 | 2.69% | 100 |
| MySQL42649 | √2 | 6.10% | 500 |
| MySQL44723 | √1 | 3.16% | 100 |
| Apache3278 | - | 0.23% | >1000 |
| Apache34464 | √3 | 0.18% | 10 |
| ... | ... | ... | ... |

*Same Diagnosis Capability*

*< 10%*

44

## Experimental Results

| BugID | Diagnosis Capability | Run-time Overhead | Requested Failure Runs |
|---|---|---|---|
| Mozilla258793 | √1 | 2.39% | |
| Mozilla299742 | √1 | 4.27% | |
| Mozilla347306 | √1 | 1.42% | |
| Mozilla416628 | √1 | 2.03% | |
| MySQL15811 | √1 | 2.25% | |
| MySQL26527 | √1 | 6.05% | |
| MySQL27287 | √1 | 3.02% | |
| MySQL40337 | √1 | 2.69% | |
| MySQL42649 | √2 | 6.10% | |
| MySQL44723 | √1 | 3.16% | |
| Apache3278 | - | 0.23% | |
| Apache34464 | √3 | 0.18% | |
| ... | ... | ... | ... |

*10 X 10000 ??*

45

## Experimental Results

| BugID | Diagnosis Capability | Run-time Overhead | Requested Failure Runs |
|---|---|---|---|
| Mozilla258793 | √1 | 2.39% | 100 |
| Mozilla299742 | √1 | 4.27% | 500 |
| Mozilla347306 | √1 | 1.42% | 10 |
| Mozilla416628 | √1 | 2.03% | 10 |
| MySQL15811 | √1 | 2.25% | 10 |
| MySQL26527 | √1 | 6.05% | 500 |
| MySQL27287 | √1 | 3.02% | 10 |
| MySQL40337 | √1 | 2.69% | 100 |
| MySQL42649 | √2 | 6.10% | 500 |
| MySQL44723 | √1 | 3.16% | 100 |
| Apache3278 | - | 0.23% | >1000 |
| Apache34464 | √3 | 0.18% | 10 |
| ... | ... | ... | ... |

46

## Experimental Results

| BugID | Diagnosis Capability | Run-time Overhead | Requested Failure Runs |
|---|---|---|---|
| Mozilla258793 | √1 | 2.39% | 100 |
| Mozilla299742 | √1 | 4.27% | 500 |
| Mozilla347306 | √1 | 1.42% | 10 |
| Mozilla416628 | √1 | 2.03% | 10 |
| MySQL15811 | √1 | 2.25% | 10 |
| MySQL26527 | √1 | 6.05% | 500 |
| MySQL27287 | √1 | 3.02% | 10 |
| MySQL40337 | √1 | 2.69% | 100 |
| MySQL42649 | √2 | 6.10% | 500 |
| MySQL44723 | √1 | 3.16% | 100 |
| Apache3278 | - | 0.23% | >1000 |
| Apache34464 | √3 | 0.18% | 10 |
| ... | ... | ... | ... |

**Why?**

47

## Conclusion and Future Works

- Study diagnosis process for PPs
  - Statistical debugging is a natural fit
- Study statistical debugging on PPs
  - Branch predicates + two statistical models
- Future works
  - Analyze inefficient loops
  - Provide detailed fix hints

48

# Break


49