

Elusive Bugs: Can you catch them all?

Hong Zhu

Department of Computing and Communications technology

Oxford Brookes University,

Oxford OX33 1HX, UK

Email: hzh@brookes.ac.uk

Program Debugging

- *Current state of art?*
 - **Automation:** Much work have been reported in the literature on automated debugging
 - Input:
 - positive and negative test executions (traces)
 - Output:
 - suggested location of bug(s)
 - Suggested correction of the bugs(s)
 - **Empirical:** Effectiveness of automated debugging tools are demonstrated by experiments and case studies
 - Measurements:
 - percentage of bugs correctly located over all know bugs
 - Percentage of bugs correctly corrected

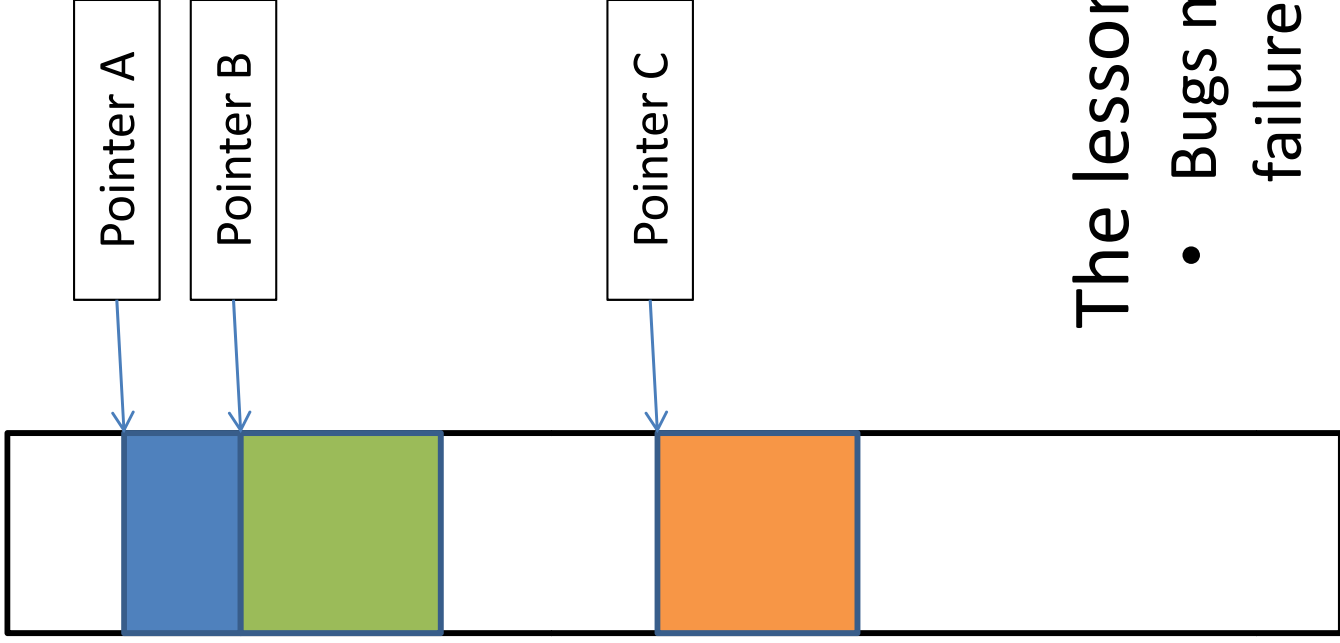
Experiences of debugging: Story 1

- Year: 1996
- Project: NDRASS – An Automated Tools for Requirements Analysis Support System
- What happened:
 - Phenomena:
 - A procedure executes correctly for the first two calls, but does not produce the correct output even on the same parameters as the first call.
 - The bug:
 - A local variable is declared as static.
- Lesson learned:
 - Bugs may be not in the statements.

Experiences of debugging: Story 2

- Year: 1987
- Project:
 - Meta-Programming for transformational software development
- Development environment:
 - Turbo-Pascal on PC 286
 - Program can declare variable to store data up to 64K;
 - DOS OS/hardware have 640K memory;
 - No graphic user interface.
- Phenomena:
 - The program has random errors:
 - cannot find where the faults are!
 - A piece of code is correct sometime, but not always correct!
- The bug:
 - Need one more slide to explain.

Heap



The pieces of code where failure occurred:

```
...  
B^ := ...  
...  
x1 := Exp1(B^);  
...
```

```
...  
B^ := ...  
...  
x2 := Exp2(B^);  
...
```

... ..

The defect in code:

```
...  
A^ := C^;  
...
```

The lesson learned:

- Bugs may be not in the piece of code where failure is observed.

Experiences of debugging: Story 3

- Year: 1981
- Project:
 - Checking the independency of the axioms of propositional logic using Hilbert method
- Development environment:
 - Modular-II on PDP-11 micro-computer
- Phenomena:
 - A piece of code is rather simple, and looks correct (I read it for so many times, and cannot find any problem.)
 - The output is not what I expected
 - Change the code to an equivalent one, the problem disappeared
- Lesson learned:
 - Bug may be not in your code!

In this case, the bug is in the compiler!

See AST 2011 proceedings for testing compilers and the findings of such bugs!

Where will your bugs be in the future?

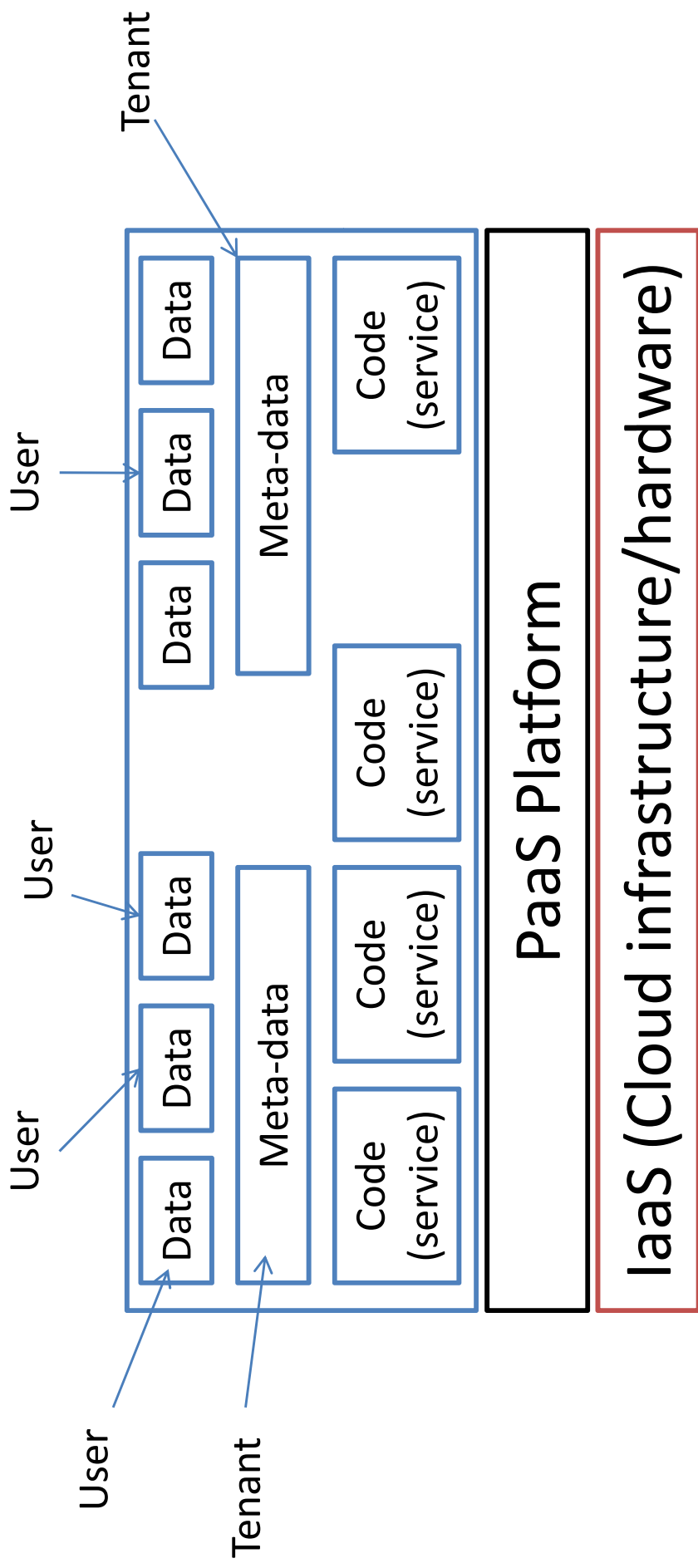


Illustration of Cloud computing architecture

Watch out!

New breeds of bugs are coming from the clouds!