Dynamic Program Analysis for a Better World

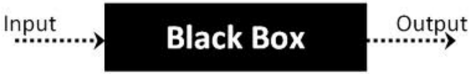
Yanyan Jiang

njujiangyy@gmail.com

Outline

Computers Changed Our World

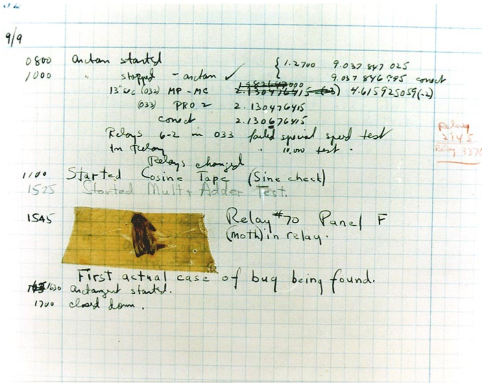
- We write programs to change the world



- The simple “blackbox” paradigm draws a paradise of applications
  - manufacturing, transportation, schedule, health care, education, entertainment, etc.
  - so be proud of what you are doing (though many times being called a “码农”)

But We are in Suffering of Bugs

- We changed the world, but with imperfections
  - endless time, money and human-hours spent on finding and fixing bugs
  - we are in a endless war between bugs



9/9

0800 Arctan started
1000 " stopped - arctan ✓ {1.2700 9.037 847 025
9.037 846 995 conct
13" uc (033) MP - MC ~~1.982147000~~ ~~2.130476415~~ (-3) 4.615925059(-2)
(033) PRO 2 2.130476415
conct 2.130676415
Relays 6-2 in 033 failed special speed test
In Relay " 11.000 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult + Adder Test.
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
1630 Antangent started.
1700 closed down.

Relay 2145
Relay 3376

To Be a Bug Fighter!

- Have you ever thinking to be a bug fighter
  - for salvation of the programmers,
  - *using knowledge learned in this course*?
- Write programs that play with programs!

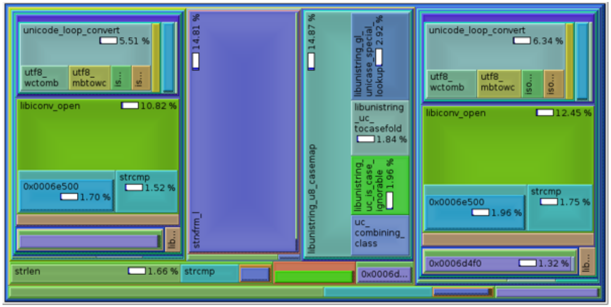
Outline

The Game Issue

- You wrote a fantastic game, but it runs only 15FPS on your i7 computer
  - but the game experience requires 30+ FPS
  - fortunately, you think the code can be tuned
  - but, to optimize what portion of your 100,000 lines of code?

The Profiler

- A magical tool that displays which function takes most of the time
  - run the program together with the profiler
- Rationales
  - usually, 10% of code takes up 90% of time
  - real case: string comparison bottleneck in git gc



Implement a Profiler

- By sampling
  - sample the “current running code” periodically
  - find EIP’s corresponding function at each time interrupt
- By instrumentation
  - change the source/binary to log function call timings
  - `f();`$\rightarrow$`t1=time();f();t2=time();cost_f+=t2-t1;`
- By cheating
  - use monitor (hardware/OS) provided tools (e.g., debugging interrupts)

Outline

Dynamic Program Analysis

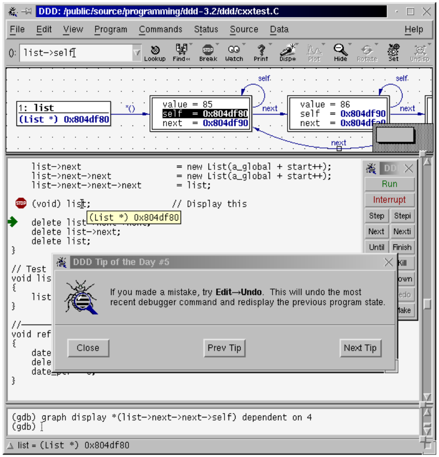
- The analysis of computer software that is performed by executing programs on a real or virtual processor
  - play with concrete execution of programs
- Two fundamental approaches
  - instrumentation / change the monitor
- Applications
  - anything you can imagine if you have a program at hand (software testing, debugging, maintenance, etc.)

Applications of Dynamic Program Analysis

- Anything related to a concrete execution
  - test input generation and coverage measurement
  - memory error (corruption/leak/overflow) detection
  - profiling and optimization hints
  - debugging aids
  - mobile application analysis (privilege/energy leaking)
  - multi-thread error (deadlock/data race/atomicity violation) detection

Example: Data Display Debugger

- Ever thinking of visualizing your program execution?
  - it is done 18 years ago[1]!



```
    list->next                  = new List(a_global + start++);
    list->next->next            = new List(a_global + start++);
    list->next->next->next      = list;

    (void) list;                // Display this
                   (List *) 0x804df80
    delete list
    delete list->next;
    delete list;
}

// Test
void lis
{
    list
}

//———
void ref
{
    date
    dele
    date
}
```

If you made a mistake, try **Edit→Undo**. This will undo the most recent debugger command and redisplay the previous program state.

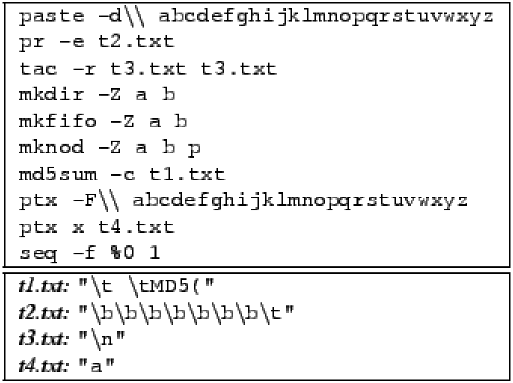
```
(gdb) graph display *(list->next->next->self) dependent on 4
(gdb)
```

```
list = (List *) 0x804df80
```

[1]A. Zeller, D. Lutkehaus. DDD: A free graphical front-end for UNIX debuggers. In *SIGPLAN Not.* 31(1), pp. 22–27, 1996.

Example: Dynamic Symbolic Execution

- The daily used coreutils contain bugs!
  - and these bugs are find by a dynamic program analysis tool[2]!



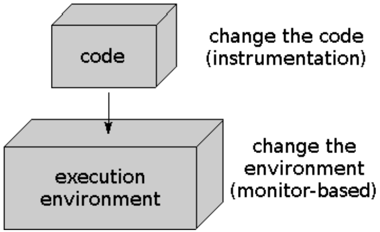
```
paste -d\\ abcdefghijklmnopqrstuvwxyz
pr -e t2.txt
tac -r t3.txt t3.txt
mkdir -Z a b
mkfifo -Z a b
mknod -Z a b p
md5sum -c t1.txt
ptx -F\\ abcdefghijklmnopqrstuvwxyz
ptx x t4.txt
seq -f %0 1
```

```
t1.txt: "\t \tMD5("
t2.txt: "\b\b\b\b\b\b\b\t"
t3.txt: "\n"
t4.txt: "a"
```

[2]C. Cadar, D. Dunbar, D. Englar. KLEE: Unassisted and automatic generation of high-coverage tests for complex systems programs. In *Proc. of OSDI*, 2008.

Two Ways to Implement a Dynamic Analysis Tool

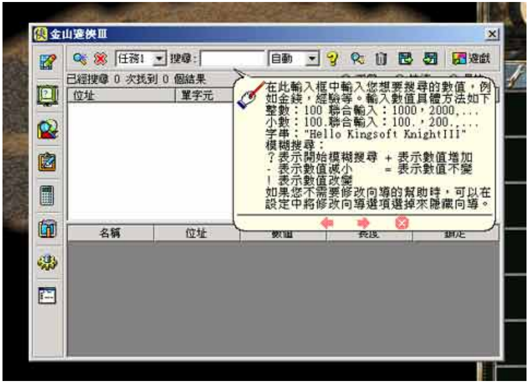
- Instrumentation
  - change the source code or binary to perform specific functions
  - example: gprof (insert code at function calls)
- Monitor-based
  - use the up-level function (operating system/virtual machine monitor) to achieve a specific goal
  - example: KLEE (a LLVM bitcode interpreter)



Outline

Program Analysis for Fun

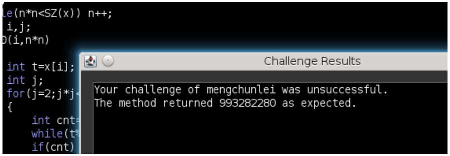
- Backing to the 2000s
  - hacking 大菠萝, 红警, 大航海 and, especially, HGAMES with “Kingsoft Knight”
  - have you ever wondered how to implement such a amazing tool?



在此輸入框中輸入您想要搜尋的數值，例如金錢，經驗等。輸入數值具體方法如下
整數：100 聯合輸入：1000，2000,...
小數：100.聯合輸入：100.，200.,...
字串："Hello Kingsoft KnightIII"
模糊搜尋：
？表示開始模糊搜尋 ＋ 表示數值增加
- 表示數值減小 ＝ 表示數值不變
！表示數值改變
如果您不需要修改向導的幫助時，可以在設定中將修改向導選項選掉來隱藏向導。

A Concrete Example

- Topcoder has a special challenge phase
  - you can only view codes of others, or “challenge” them with specified test cases
  - the Java client forces you can only view, but not copy-paste the code outside its client
  - succeed +50 pts, fail -25 pts
- We are sure that the codes are in the memory, and we want to dig them out



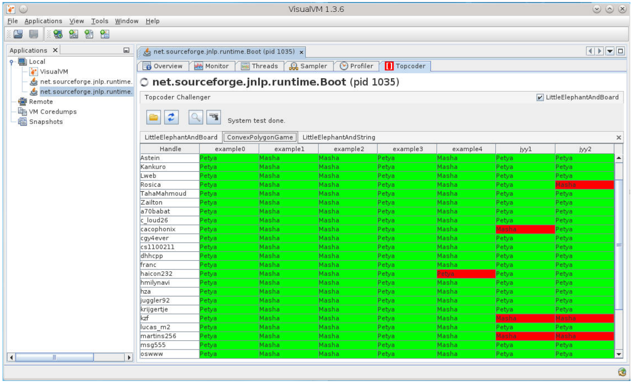
```
le(n*n<SZ(x)) n++;
 i,j;
)(i,n*n)

 int t=x[i];
 int j;
 for(j=2;j*j
 {
     int cnt=
     while(t%
     if(cnt)
```

```
Your challenge of mengchunlei was unsuccessful.
The method returned 993282280 as expected.
```

A Topcoder Challenger

- An insanely IMBA tool to achieve 100% challenge successful rate
  - scan the heap to find the code → compile → automatically test with pre-defined test cases



VisualVM 1.3.6
File Applications View Tools Window Help
Applications
Local
VisualVM
net.sourceforge.jnlp.runtime.
net.sourceforge.jnlp.runtime.
Remote
VM Coredumps
Snapshots
net.sourceforge.jnlp.runtime.Boot (pid 1035)
Overview Monitor Threads Sampler Profiler Topcoder

net.sourceforge.jnlp.runtime.Boot (pid 1035)

Topcoder Challenger ☑ LittleElephantAndBoard

System test done.

LittleElephantAndBoard | ConvexPolygonGame | LittleElephantAndString

Handle	example0	example1	example2	example3	example4	jyy1	jyy2
Astein	Petya	Masha	Masha	Petya	Masha	Petya	Petya
Kankuro	Petya	Masha	Masha	Petya	Masha	Petya	Petya
Lweb	Petya	Masha	Masha	Petya	Masha	Petya	Petya
Rosica	Petya	Masha	Masha	Petya	Masha	Petya	🟥 Masha
TahaMahmoud	Petya	Masha	Masha	Petya	Masha	Petya	Petya
Zailton	Petya	Masha	Masha	Petya	Masha	Petya	Petya
a70babat	Petya	Masha	Masha	Petya	Masha	Petya	Petya
c_loud26	Petya	Masha	Masha	Petya	Masha	Petya	Petya
cacophonix	Petya	Masha	Masha	Petya	Masha	🟥 Masha	Petya
cgy4ever	Petya	Masha	Masha	Petya	Masha	Petya	Petya
cs1100211	Petya	Masha	Masha	Petya	Masha	Petya	Petya
dhhcpp	Petya	Masha	Masha	Petya	Masha	Petya	Petya
franc	Petya	Masha	Masha	Petya	Masha	Petya	Petya
haicon232	Petya	Masha	Masha	Petya	🟥 Petya	Petya	Petya
hmilynavi	Petya	Masha	Masha	Petya	Masha	Petya	Petya
hza	Petya	Masha	Masha	Petya	Masha	Petya	Petya
juggler92	Petya	Masha	Masha	Petya	Masha	Petya	Petya
krijgertje	Petya	Masha	Masha	Petya	Masha	Petya	Petya
kzf	Petya	Masha	Masha	Petya	Masha	🟥 Masha	🟥 Masha
lucas_m2	Petya	Masha	Masha	Petya	Masha	Petya	Petya
martins256	Petya	Masha	Masha	Petya	Masha	🟥 Masha	🟥 Masha
msg555	Petya	Masha	Masha	Petya	Masha	Petya	Petya
oswww	Petya	Masha	Masha	Petya	Masha	Petya	Petya

A Closer Look at the State-of-the-art

- What is a researcher doing?
  - find things new, interesting and useful
- Why the term “research” is so far away from us?
  - not interested $\rightarrow$ bad teachers
  - do not have basic capability of doing research (e.g., implementation, reading and writing skills, etc.) $\rightarrow$ bad teachers
  - do not know what is the state-of-the-art $\rightarrow$ bad teachers
- But you still have good teachers!
  - papers, books and open courses are available online

Q & A Time

- Have fun!