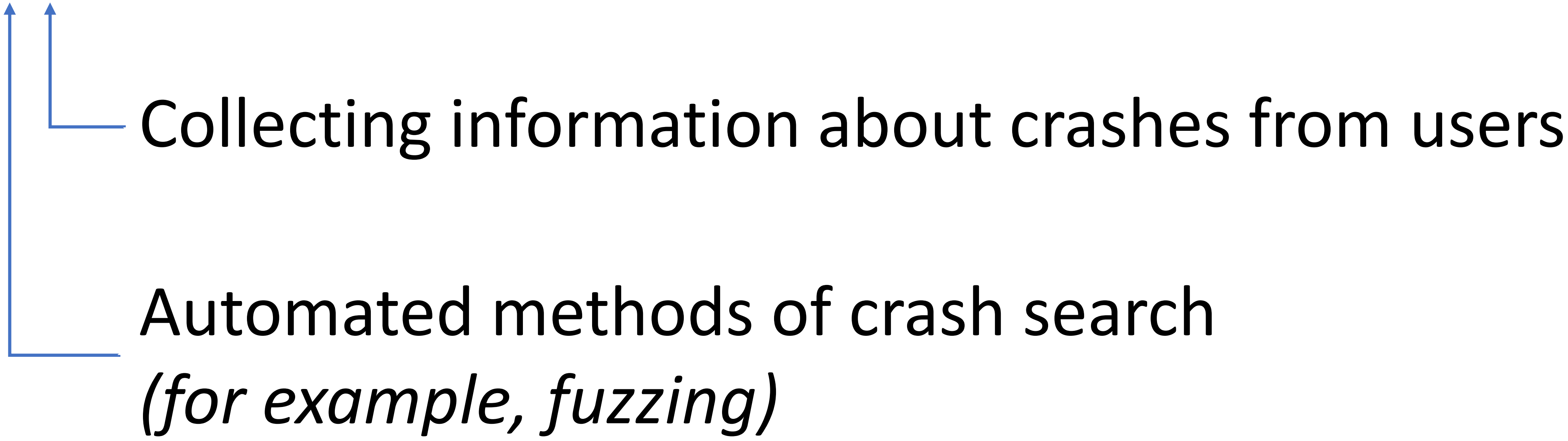
Crash processing for selection of unique defects

F.V. Niskov, A.N. Fedotov, Sh.F. Kurmangaleev
{fedor.niskov, fedotoff, kursh}@ispras.ru

Problem: to analyze manually a huge amount of program crashes



Collecting information about crashes from users

Automated methods of crash search *(for example, fuzzing)*

⇒ **An automated filtration is needed**

(selection of unique crashes, which are not similar to each other)

The AFL filtration method

- The AFL fuzzer (American Fuzzy Lop) filters discovered crashes.
- AFL uses instrumentation to collect information about CFG (control flow graph) – what blocks and jumps have been executed.
- New crash is considered unique, if one of two conditions is true:

1) Its CFG has an edge
   which each CFG of previous crashes did not have.
2) Its CFG does not have an edge
   which each CFG of previous crashes had.

Disadvantages of the AFL method

In some situations, the AFL filter fails and leaves too many crashes:

- Programs working with complex data formats
- Crashes are caused by a jump to address which had been overwritten with input data because of a bug

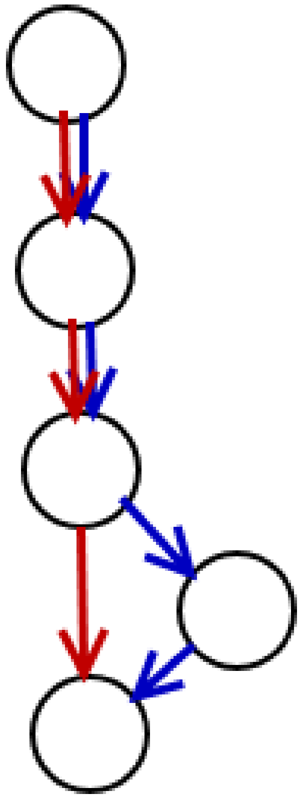
The new filtration method

- The suggested method uses comparison of CFGs, obtained by AFL-style instrumentation
- For comparison of graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ the following metric is introduced:

$$\rho(G_1, G_2) = \frac{|E_1 \Delta E_2|}{|E_1 \cup E_2|}$$

Example:

$$\rho = \frac{3}{5} = 60\%$$



The new filtration method

- Two crashes are considered similar, if:

1) Their CFGs are similar
   (the metric is not greater than given threshold)
2) They have the same crash point
   (machine instruction address)

- A crash is added to the set of unique crashes (which is empty initially), if it is not similar to any of them.

The algorithm of fixing bugs

1) C is a set of inputs which cause program P to crash.
2) K is a number of crashes to analyze (parameter, set by developers).
3) Set such metric threshold that the filter selects approximately K crashes.
4) Analyze the selected crashes and fix bugs in program P.
5) Run fixed program P on all inputs of C, removing inputs which don’t cause crashes.
6) If C is not empty, go to (2).
7) The end.

Testing

Set of programs for x86-64/Linux:

- `swfdump` (package `swftools`)
- `h5dump` (package `hdf5-tools`)
- `pdfinfo` (package `poppler-utils`)
- `jbig2dec`
- `goblin` (library)
- `faad`

The testing has shown that the filter can successfully reduce number of crashes.

The results of testing

Metric threshold	Number of crashes for programs					
	swfdump	h5dump	pdfinfo	jbig2dec	goblin	faad
None (AFL)	158	156	225	86	25	12
10%	124	28	19	22	2	4
20%	64	22	5	11	1	3
30%	25	22	2	7	1	3
40%	11	22	2	7	1	2
50%	6	22	1	6	1	2
60%	3	22	1	6	1	1
70%	3	22	1	6	1	1
80%	3	22	1	6	1	1
90%	3	22	1	6	1	1

Thank you!