pbSE:Phase-based Symbolic Execution

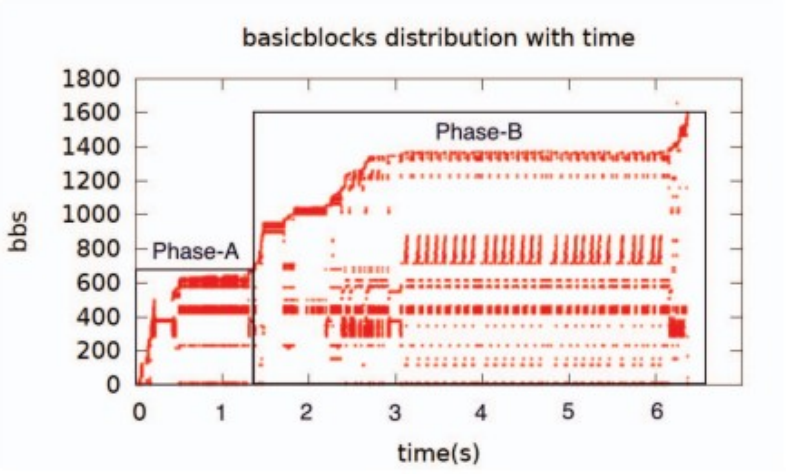
Qixue Xiao, Yu Chen, Chengang Wu, Kang Li, Junjie Mao

Background

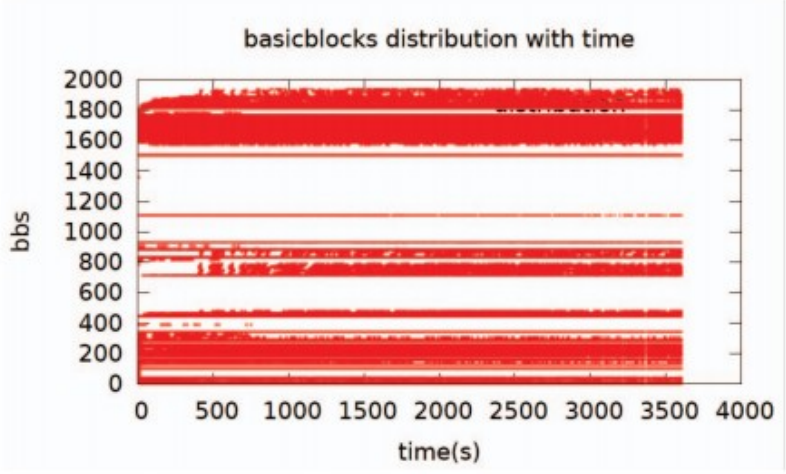
- Phase
  - A property that captures the time varying behavior of a program’s execution
  - A group of code that repeatedly being executed in a given time
  - Loops, recursive function calls
- Program often stuck in some specific phases of a program
- Trap phase
  - Phases that make symbolic execution difficult to move on
- Use phase information to guide dynamic symbolic execution

Program phase and symbolic execution

- Phase-A: file header data handling
- Phase-B: handle other data
- BB between 500 to 700
- Elf_header.e_phnum and elf_header.e_shnum



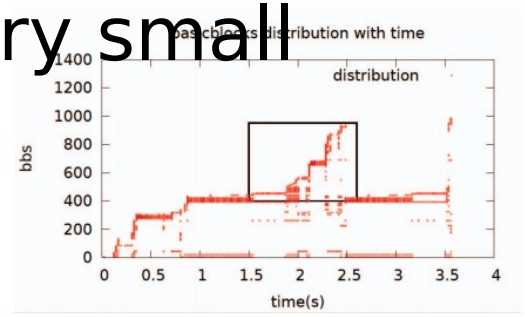
(a) BBs distribution of readelf performing concrete execution



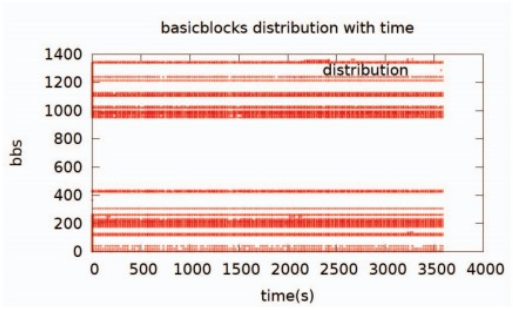
(b) BBs distribution of readelf performing symbolic execution

Program phase and symbolic execution

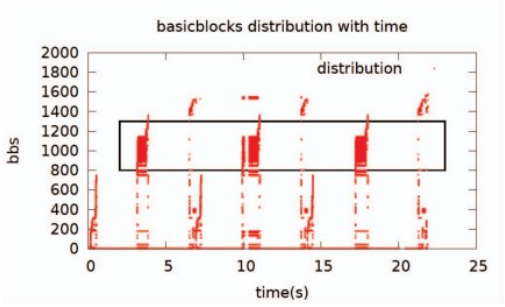
- Phase are executed in a continuous long time, caused by nested loops and deep recursions
- The chance of successfully reaching a deep phase is very small



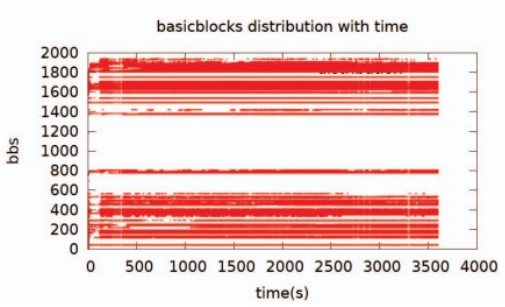
(c) BBs distribution of gif2tiff performing concrete execution



(d) BBs distribution of gif2tiff performing symbolic execution



(e) BBs distribution of pngtest performing concrete execution



(f) BBs distribution of pngtest performing symbolic execution

Phase-based Symbolic Execution

Algorithm 1 Phase-Based Symbolic Execution Testing

Require: $Seeds, Target$

1: $Seeds$: seeds selected using methods similar with fuzzing testing
2: $Target$: the tested program which compiled in llvm bitcode
3: **function** PBSYMBOLICEXETESTING($Seeds, Target$)
4: **while** $(!Seeds.empty())$ **or** $(!TIMEOUT)$ **do**
5: $seed = selectSeeds(seeds)$;
6: $BBVs =$ CONCOLICEXE(seed);
7: $phases = PhaseAnalysis(BBVs)$;
8: PBSYMBEXE($phases$)
9: **end while**
10: **end function**

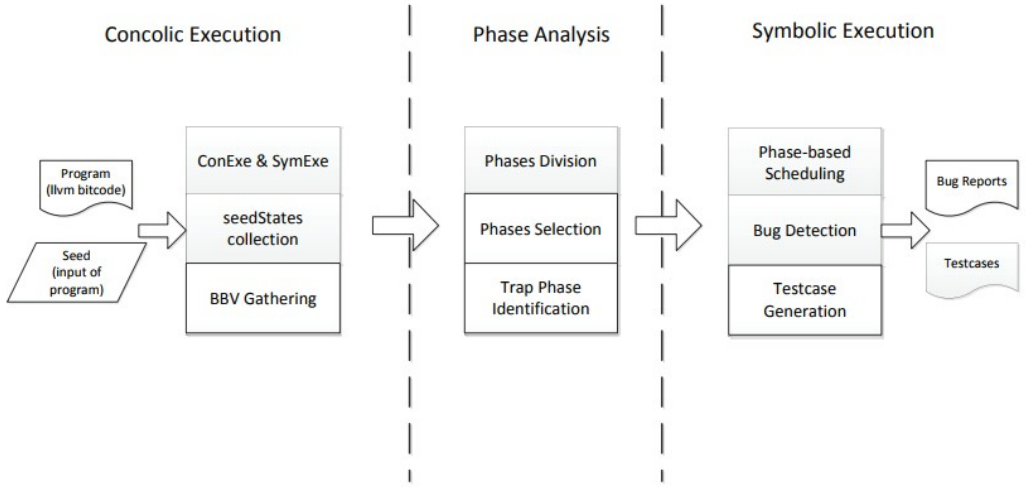


Fig. 3. Architecture of pbSE

Phase-based Symbolic Execution

- How we divide a program into phases using the concrete execution information
- How pbSE use the phase information to guide execution to new phases
- How to balance the execution time among different phases

Phase dividing

- BBV(basic block vector) gathering
  - Count of how many times a given BB has been entered during a certain interval
  - Code coverage at the moment
- Kmeans (from 1 to 20)
  - Cluster the greatest number of trap phases
- A phase consisting of N continuous BBVs is identified as a trap phase

Pass through a phase

- SeedStates
  - New states at each fork point
  - Map them to different phases based on the time
- Symbolic Execution performed to different phases can be transformed into one performed to the mapped seedStates
- When symbolic execution performed to one phase no longer covers any new code in a certain period of time, phSE begins to test the next phase

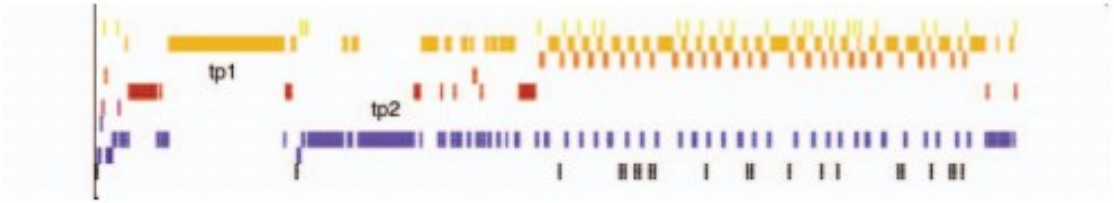
Evaluation

- KLEE and pbSE (readelf)

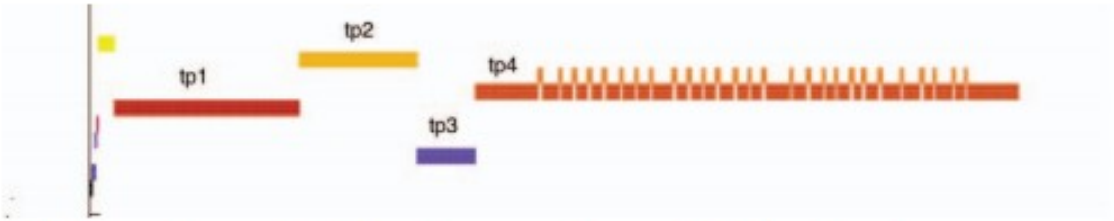
seacher	sym-10		sym-100		sym-1000		sym-10000	
	1h	10h	1h	10h	1h	10h	1h	10h
default	892	914	976	1079	996	1077	1013	1051
random-path	914	916	1162	1205	1163	1239	1141	1206
random-state	659	659	604	621	619	621	603	621
covnew	662	662	604	604	569	569	571	571
md2u	688	688	687	687	644	652	641	641
dfs	414	876	414	1231	413	1125	413	1103
bfs	757	853	687	687	644	652	634	662

pbSE	c-time	p-time	1h	10h
seed(576)	3.7s	0.2s	1687	2455
seed(7981)	407s	2.5s	1801	2597

Evaluation



(a) Trap phases identified by using BBVs without code coverage



(b) Trap phases identified by combining BBVs and code coverage

Evaluation

program	random-path								covnew								pbSE		
	sym-10		sym-100		sym-1000		sym-10000		sym-10		sym-100		sym-1000		sym-10000		1h	10h	inc
	1h	10h	1h	10h	1h	10h	1h	10h	1h	10h	1h	10h	1h	10h	1h	10h			
binutils-2.26 readelf	914	916	1162	1205	1163	1239	1141	1206	662	662	604	604	569	569	571	571	1801	2597	109%
libtiff-4.06 gif2tiff	541	541	573	573	567	567	567	567	541	541	558	561	566	567	544	544	1012	1457	134%
libpng-1.2.56 pngtest	606	606	843	843	846	848	846	848	606	606	825	843	843	843	825	835	1445	1878	121%
libdwarf-20150115 dwarfdump	460	465	462	495	465	492	453	482	460	470	455	488	453	506	457	502	932	1045	112%

Evaluation

TABLE III
BUGS FOUND BY PBSE

package	test-driver	s-size	t-p	b-p	CVEID
libpng	pngtest	8796	9	3	CVE-2015-7981
				5	CVE-2015-8540
libtiff	gif2tiff	407	4	3	N
				3	N
	tiff2rgba	243	4	3	N
		1428	5	3	N
	tiff2bw	1428	5	3	N
libdwarf	dwarfdump	9456	11	3	CVE-2015-8538
				2	N
		8336	10	5	CVE-2015-8750
				6	CVE-2016-2050
				3	N
		6908	7	5	N
				5	N
		8200	9	5	CVE-2016-2091
				4	N
		43600	9	3	CVE-2014-9482
binutils	readelf	7960	5	3	N
				4	N
		8336	7	5	N
				4	N