BugBench: A Benchmark for Evaluating Bug Detection Tools

Shan Lu, Zhenmin Li, Feng Qin,
Lin Tan, Pin Zhou and Yuanyuan Zhou

University of Illinois, Urbana-Champaign

Content of This Talk

- Share our experience
- Bug/application characteristics analysis

- BugBench has been used by
  - Our previous work [Micro'04, ISCA'04, HPCA'05]
  - Other research groups: UCSD, Purdue, NCSU, etc.

Current Benchmark Suite

Name	Program	Source	LOC	Crash Latency	Bug Type
NCOM	ncompress-4.2.4	Red Hat	1.9K	N/A	Stack smash
POLY	polymorph-0.4.0	GNU	0.7K	9040K Inst	Stack smash & Global buffer overflow
GZIP	gzip-1.2.4	GNU	8.2K	15K Inst	Global buffer overflow
COMP	129.compress	SPEC95	2.0K	N/A	Global buffer overflow
GO	099.Go	SPEC95	29.6K	N/A	Global buffer overflow
MAN	man-1.5h1	Red Hat	4.7K	29.5M Inst	Global buffer overflow
BC	bc-1.06	GNU	17.0K	189K Inst	Global buffer overflow
SQUD	squid-2.3	squid	93.5K	0	Global buffer overflow
CALB	cachelib	UIUC	6.6K	N/A	Uninitialized read
CVS	cvs-1.11.4	GNU	114.5K	N/A	Double free
YPSV	ypserv-2.2	Linux NIS	11.4K	N/A	Memory leak
PFTP	proftpd-1.2.9	ProFTPD	68.9K	N/A	Memory leak
SQUD2	squid-2.4	squid	104.6K	N/A	Memory leak
HTPD	httpd-2.0.49	Apache	224K	N/A	Data race
MSQL1	msql-4.1.1	MySQL	1028K	N/A	Data race
MSQL2	msql-3.23.56	MySQL	514K	N/A	Atomicity
MSQL3	msql-4.1.1	MySQL	1028K	N/A	Atomicity
PSQL	postgresql-7.4.2	PostgreSQL	559K	N/A	Semantic
HTPD2	httpd-2.0.49	Apache	224K	N/A	Semantic

memory related

multi-thread related

semantic

Other type of bugs: *In searching …*

Functionality

Name	Catch Bug?			Related Memory Object Type
	Valgrind	Purify	CCured	
NCOM	No	No	Yes	Stack
POLY	Vary	Yes	Yes	Stack & global buffer
GZIP	Yes	Yes	Yes	Global buffer
COMP	No	No	Yes	
GO	No	Yes	Yes	
MAN	Yes	Yes	Yes	
BC	Yes	Yes	Yes	Heap buffer
SQUD	Yes	Yes	N/A	

Valgrind miss stack buffer overflow

miss moderate global-buffer overflow

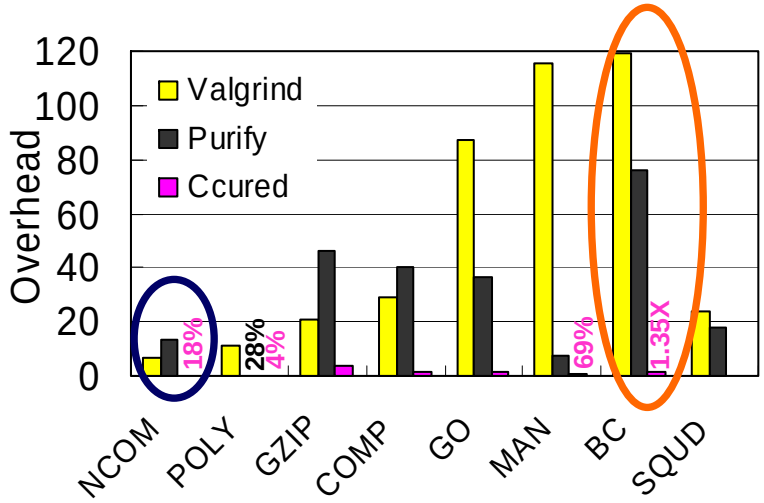
Purify miss stack buffer overflow

miss 1 Byte global-buffer overflow

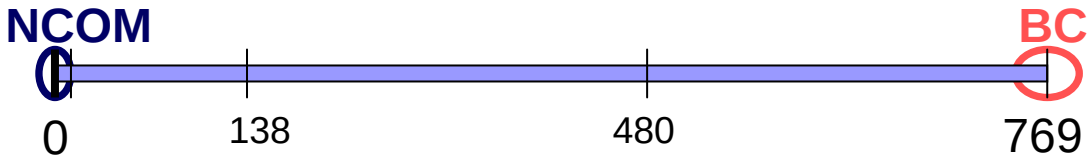
CCured Failed to apply

Overhead

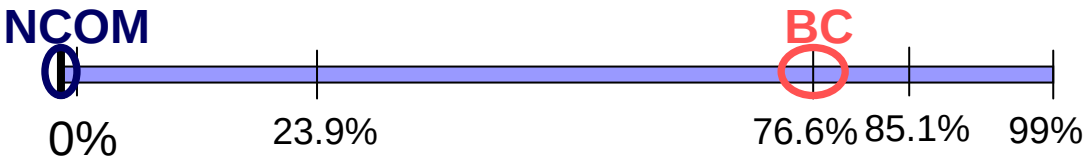
- Valgrind: 6.4X (NCOM) ~ 119X (BC)
- Purify: 28% (POLY) ~ 76X (BC)
- CCured: 4% (POLY) ~ 3.7X (GZIP)



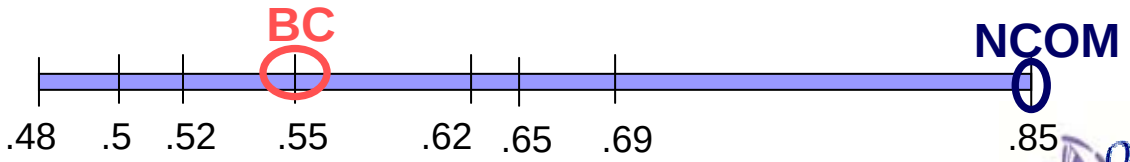
Memory Alloc Freq.
(# per MInst)



Heap Usage Ratio
[Heap/(Heap+Stack)]



Mem. Access Freq.
(# per Instruction)



Experience Summary

- Building benchmark is a time-consuming and long-term work
  - Motivate automatic tools to extract bugs
- Bug/application characteristics are important for selecting applications
- Need cooperation from entire community