Benchmarking Crimes

January 17, 2024

Mark Waschkeit

Friedrich-Alexander-Universität Erlangen-Nürnberg

Benchmarking Crimes: Categories

- A. Selective Benchmarking
  $\Rightarrow$ a single number is not representative of a system
- B. Improper Handling of Benchmark Results
  $\Rightarrow$ wrongly processing or interpreting benchmarks
- C. Using the Wrong Benchmarks
  $\Rightarrow$ not measuring what is intended to be measured
- D. Improper Comparison of Benchmarking Results
  $\Rightarrow$ results only become relevant when compared
- E. Benchmarking Omissions
  $\Rightarrow$ necessary measurements for evaluations that are not yet covered
- F. Missing Information
  $\Rightarrow$ important information has not been specified

A. Selective Benchmarking

- A1: Not evaluating potential performance degradation
  $\Rightarrow$ benchmark only shows improvements
- A2: Benchmark subsetting without proper justification
  $\Rightarrow$ making subbenchmarks and summarizing them to
  $\rightarrow$ one single number
- A3: Selective data set hiding deficiencies
  $\Rightarrow$ only testing a limited range of possible parameter
  $\Rightarrow$ values

B. Improper Handling of Benchmark Results

- B1: Microbenchmarks representing overall performance
  $\Rightarrow$ only single aspects have been tested, but not the
  $\Rightarrow$ system as a whole (e.g. individual functions)
- B2: Throughput degraded by x% $\Rightarrow$ overhead is x%
  $\Rightarrow$ throughput comparisons require comparisons of CPU
  $\Rightarrow$ load
- B3: Creative overhead accounting
  $\Rightarrow$ e.g. a runtime change from 5s to 20s being depicted as
  $\Rightarrow$ 75% slowdown instead of 300% slowdown
- B4: No indication of significance of data
  $\Rightarrow$ random variation due to measurement error has to be
  $\Rightarrow$ specified
- B5: Incorrect averaging across benchmark scores
  $\Rightarrow$ only the geometric mean is capable of properly
  $\Rightarrow$ averaging ratios ($\sqrt[n]{x_1 \cdot \ldots \cdot x_n}$)

C. Using the Wrong Benchmarks

- C1: Benchmarking of simplified simulated system
  $\Rightarrow$ emulated systems have different characteristics than
  $\Rightarrow$ real systems
- C2: Inappropriate and misleading benchmarks
  $\Rightarrow$ not measuring what is intended to be measured
- C3: Same dataset for calibration and validation
  $\Rightarrow$ train and test data sets intersecting

D. Improper Comparison of Benchmarking Results

- D1: No proper baseline
  $\Rightarrow$ e.g. changing the baseline for different tests
- D2: Only evaluate against yourself
  $\Rightarrow$ comparing to the state of the art is way more
  $\Rightarrow$ meaningful
- D3: Unfair benchmarking of competitors
  $\Rightarrow$ e.g. the competitor's system being tested on its worst
  $\Rightarrow$ settings

E. Benchmarking Omissions

- E1: Not all contributions evaluated
  $\Rightarrow$ no determination whether self-made claims are met or
  $\Rightarrow$ not
- E2: Only measure runtime overhead
  $\Rightarrow$ e.g. fails to measure memory overhead
- E3: False positives/negatives not tested
  $\Rightarrow$ missing information about the accuracy of the system's
  $\Rightarrow$ decisions (e.g. virus detection)
- E4: Elements of solution not tested incrementally
  $\Rightarrow$ optional optimizations (do not influence functonality)
  $\Rightarrow$ have not been tested individually

F. Missing Information

- F1: Missing platform specification
  ⇒ missing hardware information (CPU, cache architecture,
  ⇒ memory, etc.
- F2: Missing software versions
  ⇒ e.g. operating system, compiler, programs used (and
  ⇒ their versions)
- F3: Subbenchmarks not listed
  ⇒ benchmarking suites provide subbenchmark results
  ⇒ and should be listed
- F4: Relative numbers only
  ⇒ absolute numbers carry more information

Problems caused by Benchmarking Crimes

Table 1: Benchmarking Crimes' Influence on Completeness(C), Relevancy(R_1), Soundness(S) and Reproducibility(R_2). Bold = high impact [1]

	C	R_1	S	R_2
A1: Not evaluating potential performance degradation	**o**			
A2: Benchmark subsetting without proper justification	o	o		
A3: Selective data set hiding deficiencies	o			
B1: Microbenchmarks representing overall performance		o		
B2: Throughput degraded by x% $\Rightarrow$ overhead is x%			**o**	
B3: Creative overhead accounting			o	
B4: No indication of significance of data	o			
B5: Incorrect averaging across benchmark scores			o	
C1: Benchmarking of simplified simulated system			**o**	
C2: Inappropriate and misleading benchmarks		**o**		
C3: Same dataset for calibration and validation		o		

Problems caused by Benchmarking Crimes

Table 2: Benchmarking Crimes' Influence on Completeness(C), Relevancy(R_1), Soundness(S) and Reproducibility(R_2). Bold = high impact [1]

	C	R_1	S	R_2
D1: No proper baseline		**o**		
D2: Only evaluate against yourself		o		
D3: Unfair benchmarking of competitors		**o**		
E1: Not all contributions evaluated	**o**			
E2: Only measure runtime overhead	o			
E3: False positives/negatives not tested	o			
E4: Elements of solution not tested incrementally	o			
F1: Missing platform specification				o
F2: Missing software versions				o
F3: Subbenchmarks not listed	**o**			
F4: Relative numbers only	o			

Benchmarking Crimes' Presence

Table 3: Benchmarking Crimes' Presence in 2010 and 2015 [1]
c/p = crime/paper (c/p pair = element of the cross product of the set of crimes and set of papers)

	2010	2015
c/p pairs with crime(s)	27% (69/255)	27% (162/596)
c/p pairs being underspecified	3% (8/255)	3% (15/596)
c/p pairs for E1	38% (6/16)	0% (0/34)

(E1: Not all contributions evaluated, only significant change)

Conclusion

- some suggestions are easy to realize, e.g. with using benchmarking suites $\Rightarrow$ automation of benchmarks
- all suggestions are necessary to make useful benchmarks (completeness, relevancy, soundness, reproducibility)
- no real improvements in benchmarking over the years $\Rightarrow$ education on proper benchmarking is necessary

References

[1] Erik van der Kouwe, Dennis Andriesse, Herbert Bos, Cristiano Giuffrida, and Gernot Heiser. Benchmarking crimes: An emerging threat in systems security. 2018.