

# Repeatability and Workability for the Software Community: challenges, experiences and the future.

Dennis Shasha, [shasha@cs.nyu.edu](mailto:shasha@cs.nyu.edu)

# Background

- Sigmod – major research database conference, theory, algorithms, and experiments – how to make data-intensive systems faster.
- I was program chair in 2008
- The problem: experiments under-described.

# Repeatability and Workability

- Why not ask authors to submit code and data for testing?
- Didn't mean we didn't believe them, but portability is an issue.
- Workability – change data a little to determine how sensitive results are to the data.

# Agreeing to the Repeatability Test was Optional

- Had to be because of corporate confidentiality.
- Would have been cool to test inside a firewall. So a great tool would permit remote testing and timing.

# Statistics

- Most supported the effort: “If you had time, would you participate in a repeatability effort on any system you built?” Vote was something like 100 yes to 2 no.
- Over the last two conferences, more than half the authors tried to achieve repeatability.

# Excuses Revealing

- “We cannot distribute code and data because the author has moved, making the retrieval of code and data infeasible at this point.”
- “The subsets were chosen randomly from a large dataset, and unfortunately no trace about the identity of the used documents has been kept.”

# Positive Comments – people get it

- “This wasn't too hard and I think it was definitely worth it. We even found a mistake (thankfully a minor one, not affecting our conclusions)...”
- “I think this is a noble effort and costs almost nothing for authors if they set up experiments with repeatability in mind. The focus on repeatability will lead to better science in our community.”

# Recommendations

- Test only accepted papers
- Ask authors how long the tests should run – helps to determine whether non-response after 20 minutes is to be expected.
- One-way anonymous communication between authors and repeatability reviewers (reviewer is anonymous). Allows reviewer to clear up questions.

# Recommendations/Tools

- Utility that runs through source code checking for machine dependency (e.g. assembly language), operating system dependencies (e.g. system calls), shell dependencies, path dependencies or browser dependencies.
- Computational resource where code purged of dependencies can be tested by author, e.g. Wisconsin Metronome system
- Testing harness to collect results and parameters.

# Incentives/Benefits

- Authors: results are more meaningful if they can be built upon. Algorithm more believed if widely available.
- Reviewers: should be as prestigious as being on any program committee.
- Community: a new source of validated archived code.