The Repeatability Experiment of SIGMOD 2008

I. Manolescu[1] L. Afanasiev[2] A. Arion[1] J. Dittrich[3] S. Manegold[4]
N. Polyzotis[5] K. Schnaitter[5] P. Senellart[1] S. Zoupanos[1]
D. Shasha[6]
[1] INRIA Saclay–Île-de-France, France firstname.lastname@inria.fr
[2] University of Amsterdan, Netherlands lafanasi@science.uva.nl
[3] ETH Zurich, Switzerland jens.dittrich@inf.ethz.ch
[4] CWI, Netherlands stefan.manegold@cwi.nl
[5] U. California, Santa Cruz, USA (karlsch|alkis)@soe.ucsc.edu
[6] Courant Institute, New York, USA shasha@courant.nyu.edu

ABSTRACT

SIGMOD 2008 was the first database conference that offered to test submitters' programs against their data to verify the experiments published. This paper discusses the rationale for this effort, the community's reaction, our experiences, and advice for future similar efforts.

1. MOTIVATION

Repeatability has been a fundamental driver of progress in science since the time of Francis Bacon in the 16th century. In natural science, repeatability allows one scientist to verify the assertions of another, occasionally exposing fraud, but more often simply providing a check against exuberant claims.

Natural science papers conform to the repeatability requirement by providing a complete description of the protocol used in an experiment (reagents, equipment used down to the model number, times, temperatures etc.). The protocol must be described in sufficient detail for another lab to replicate the experiment. Computer science papers can't practically do this, because software is far more complex than laboratory procedures.

Fortunately for computer science, however, a computational paper could, in an ideal world, describe the core of its algorithms in the paper and then provide software and data to enable repeatability on another researcher's computer or cluster. The key benefit of this procedure to the community is that the full specification of algorithms, code, and data helps keep track of the factors that influence the experimental results. Repeatability is thus a way to ensure that there are no hidden factors that influence the results (e.g. compiler settings).

Also, fortunately for computer science, a repeatability tester can easily change data, thus testing software in new settings. This permits the field to go beyond repeatability to what one might call "workability" for a domain of application. Finally, and once more fortunately for computer science, preparing code and data for repeatability leads, without much additional work, to preparing the code for archiving and distribution, thus allowing future researchers to compare their implementations with previous ones.

Our world is not ideal, however, in at least two relevant ways:

1. Intellectual property rights may prevent some researchers from submitting code and/or data. For this reason, repeatability or workability should remain voluntary. SIGMOD 2008 chose to give an incentive to researchers to achieve repeatability by allowing them to mention their success (or partial success) in their papers. This was enough to convince roughly 2/3 of all submissions to attempt repeatability. That number constituted nearly all those who did not invoke an Intellectual Property exemption.

2. Assessing repeatability entails a lot of work. New tools and better specification of input formats will be required to make this manageable. These practices could be of much general use.

The rest of this paper describes the community feedback to the repeatability initiative both during (Section 2) and after (Section 4) the process, the results of the assessment (Section 3), the experiences of the members of the repeatability committee (Sections 4 and 5), as well as our recommendations for the implementation of this initiative in the future (Section 6).

2. EARLY FEEDBACK FROM THE COMMUNITY

Because repeatability was new, we received many questions and tried to clarify the specification on the website. We also received a variety of comments that underscore how useful repeatability could be for the integrity of our field:

> *We cannot distribute code and data because the authors have moved, making the retrieval of code and data infeasible at this point.*

We lost some old code. Due to the short notice, we could not reproduce our lost code for these parts.

The subsets were chosen randomly from a large dataset, and unfortunately no trace about the identity of the used documents has been kept. The experiments were performed long months ago, and it wasn't expected to send results to SIGMOD, that's why we didn't pay attention about keeping a trace.

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) in our submission, so I think it was very helpful. Thanks a lot for taking the time to do the repeatability eval!

Some comments hinted at some misunderstandings of the purpose of the repeatability assessment:

My experimentation is fully deterministic: if it is wrong, running again my own program would not detect it.

It was not our purpose to declare experiments right or wrong, but simply to establish that the code yields similar results to those claimed in a paper when run by another person.

Authors of several papers suggested the evaluation should focus on accepted papers only, to reduce the effort required:

Since most submissions are going to be rejected, this assessment should focus mainly on the accepted papers, to guarantee their quality. Thus, it would be good if this procedure can be performed again when the paper decisions come out, and then request and carefully evaluate again the results reported in those accepted papers.

Why not restrict this effort to accepted papers? If repeatability results have no bearing on paper acceptance, then the current scheme wastes time and resources on papers that are ultimately rejected.

Deploying our code and the large amount of data will require some days of work. We will postpone this until the notification. If our paper is accepted we will do an attempt to deploy our system.

Surprisingly perhaps, of the total submissions of 436 papers, a full 288 attempted repeatability (or about 66%).

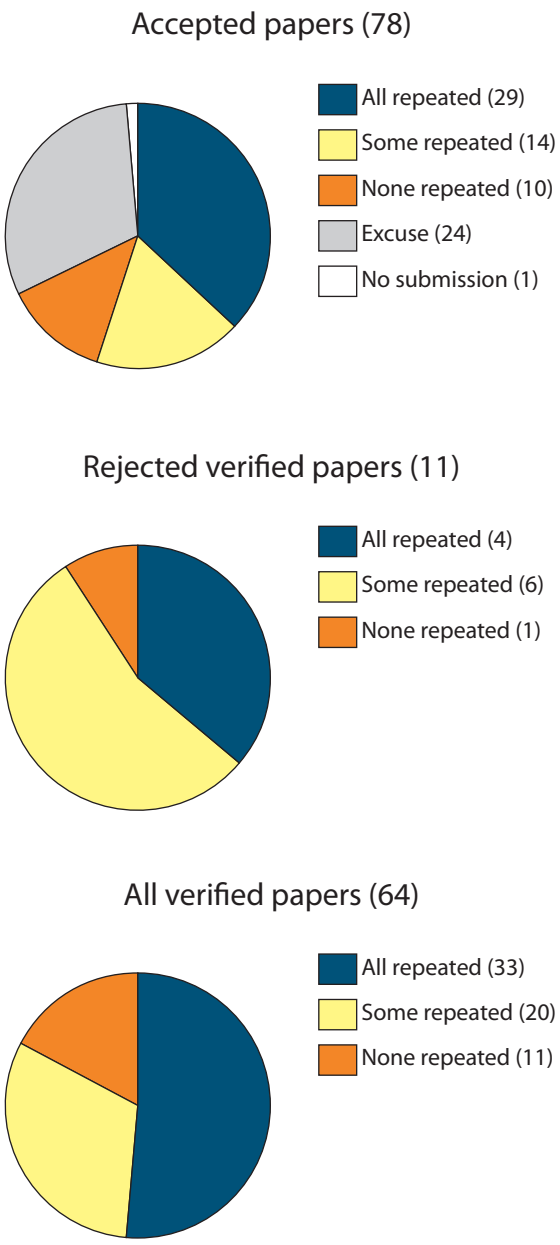

Figure 1: Repeatability assessment results.

3. RESULTS

Figure 1 presents the results of our evaluation process. The charts present the results for the 78 accepted papers as well as for the 11 submissions which were not accepted, but for which we were able to verify the code. In the first chart, "Excuse" stands for papers which presented a reason not to participate in the assessment, such as IP constraints that prevented giving away code, or confidentiality of the data used. Out of the 78 accepted papers, 54 (or about 70%) participated to the repeatability assessment, and 44 (or 56%) achieved at least some repeatability. We find these results very encouraging for a first-time effort. The second chart also shows that these ratios are reproduced almost exactly among the rejected papers with promising reviews. The third chart shows that more than half of the paper experiments were completely repeated, and only 17% of the papers did not achieve any repeatability.

Among the 11 papers for which no experiments could be repeated, three required hardware unavailable to the repeatability committee, two required unavailable software, the installation of the necessary software failed for one paper, and five papers had various runtime failures that prevented experiment completion. At the authors' request, we have continued the execution of two of these code batches beyond the SIGMOD CR deadline. One of them has since been completely repeated (initial problems included the authors' sending us "the wrong version out of CVS"). The

other one required more fixes and is still running at the time of this writing.

4. AUTHOR SURVEY

After the repeatability assessment process, a short survey was made by sending the following text to the authors of accepted SIGMOD research papers:

> *This is meant to be a sub-5 minute survey about experimental repeatability. In the case of multi-author papers, only one of you needs to answer (though we are happy to receive comments from more than one). We will strip your email headers from your responses programmatically, so please speak your mind.*
>
> 1. *Did your paper succeed on all/some/none of the repeatability tests? Or did you not submit for intellectual property reason?*
> 2. *If you submitted, was the repeatability experience helpful? If so, how? If not, how could it be improved?*
> 3. *Would you attempt repeatability in the future if it remained voluntary (i.e. had no effect on acceptance decision but you would be allowed to mention success in your paper) and you had no intellectual property constraints?*
> 4. *Do you think it would be useful to have a Wiki page for each paper so the community could comment on it, you could post code etc.?*
>
> *Warm Regards,*
> *Ioana (repeatability chair) and Dennis (program committee chair)*

The Wiki idea was suggested by Donald Kossmann, the SIGMOD 2009 program chair.

Survey results are summarized in Figure 2. The horizontal axis divides the respondents into those that did not participate in repeatability, those whose software passed all repeatability tests, those whose software passed some repeatability tests, and those whose software passed no repeatability tests. For each class of people, we give the percentage responding yes to each question, based on the color coding.

Most answers we received were very clear (yes/no), but some answers were ambiguous, in the style of "Yes and no; on one hand... but on the other hand..." We counted 0.5 points for such answers. They represented less than 20% of the answers.

It should be noted that a certain confusion occured concerning the Wiki site, as evidenced by their detailed comments. Some understood the Wiki to be an alternative to the CMT, i.e. an anonymous site where authors could interact with (code) reviewers *during the assessment*. This is not what was meant by the question; rather, Donald's idea was a permanent repository of information concerning a given paper, accessible to many, and persistent also *after* the conference. When authors simply said yes or no to a Wiki, we are not able to infer which interpretation they had chosen. Another potential confusion concerns whether to establish a Wiki for each *accepted* paper (author comments seem to show that they understood it this way) of for any *submitted* paper.

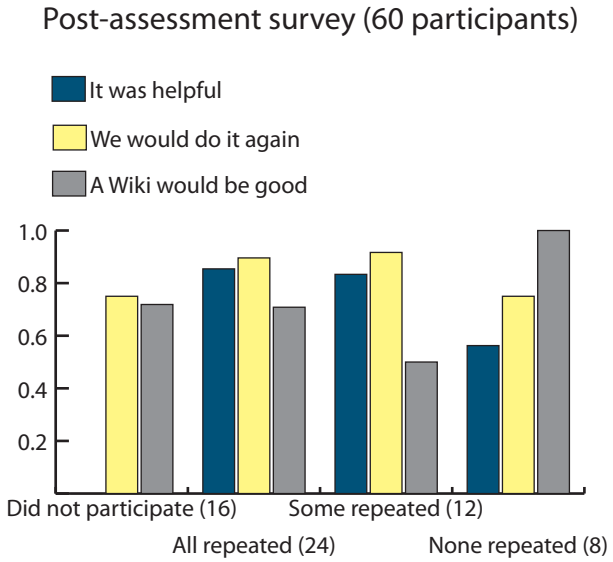

Figure 2: Survey results.

In the following, we present some representative comments, grouped by topics. After each comment, we specify the category in which the author falls, with respect to the repeatability assessment process.

4.1 On the process

> *I think this is the right direction for the community to move forward.* (Did not participate.)
>
> *This requirement is extremely important for us to improve the quality of the paper. Also, readers can trust SIGMOD papers more than before.* (All experiments repeated.)
>
> *We are happy to see that our algorithms show consistent results through machines with different hardware/software configuration.* (All experiments repeated.)
>
> *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.* (Some experiments repeated.)
>
> *Sharing experiments (code and data) benefits a lot researchers, especially small groups. I highly respect groups that publish code and/or data, such as the Heikki Mannila group. Public code and data support both repeatability tests and fair comparisons. I hesitate to study any paper without public data.* (Some experiments repeated.)
>
> *The point that 'experimental code has no effect on acceptance decision' is important, since there can be some trivial mistakes in packaging experiments. We are not professional in packaging softwares.* (All experiments repeated.)

We do not think that it is reasonable to have the authors be responsible for making the code of every tool they compare with portable and easily testable (and this would certainly discourage submission to the repeatability process!). (All experiments repeated.)

I think experimental repeatability is an important thing, and I support the motivation. But I believe that the current mechanism is just plain wrong. Way too much work for the benefit derived. (Not just for authors, but more importantly for the repeatability committee, who I am sure had to work incredibly hard). (No experiments repeated.)

An interesting comment suggested facilitating the process by means of suitable software tools:

What would really help with this situation is a SIGMOD-WORKBENCH. Workflow systems are becoming prevalent in computational biology for specifying a series of steps and then executing that series given an input. For example, to run a program that talks to a database and uses an input dataset A, you would declare that the input A and a database (with externally set) password/user are used by the program. When the repeatability committee comes along, they just have to set the appropriate database user/password, outside of any compiled code. Thus, each author doesn't need to build all the bat scripts, or even configure a database, just drag in a "Sigmod-db-standard-setup" object and have their program call it. Check out Taverna or VisTrails or Kepler. (No experiments repeated.)

4.2 On the helpfulness

Yes, it was helpful to organize the source code properly for future use. (All experiments repeated.)

It was helpful. It forced me to write documentation which I would otherwise have postponed indefinitely. (Some experiments repeated.)

It was helpful. It required us to further clean up my code and scripts and prepare documentation. (Some experiments repeated.)

Helpful? Greatly yes. Some scripts written for this test could be used to append additional experimental results immediately. To package experiments in a script form, at first, seemed bothersome, but we found out that it is good for ourselves, and improves our productivity. (All experiments repeated.)

It is a great thing for the community that this service is available, and I hope that it will have a very positive effect on both the trustworthiness of SIGMOD results and the quality of publicly-available research tools. (All experiments repeated.)

It's only helpful in the sense that it provides some extra credibility to the paper. It was not helpful to myself in any way. (Some experiments repeated.)

4.3 On Wikis

Wikis can be easily abused by those who make unfair comments on a paper since the comments are usually anonymous. (Some experiments repeated.)

A Wiki page for each paper sounds like a good idea, but I don't know how (or whether) these pages would be maintained after the conference. (Did not participate.)

A Wiki would be helpful, but it may also increase our workload for clarifying misunderstandings. I prefer private comments to public discussion. (All experiments repeated.)

It should be up to the authors to choose whether a Wiki is created or not, as there might be a maintenance overhead. A wiki may end up serving as an unfair/baseless defaming of published work by anonymous people of unknown credibility (rather than collecting constructive comments). As an author, one should either spend a lot of time rebutting against irresponsible comments or allow random people to anonymously defame their work. (Some experiments repeated.)

An anonymous (to deal with double-blind reviewing) Wiki might be a good idea, e.g., to post more detail about the experiments than will fit in the paper. (Did not participate.)

It might be interesting to have a centralized place for feedback from readers, but it would have to be carefully moderated and it might quickly become out-of-date unless there are clear expectations about author participation. (Some experiments repeated.)

The Wiki could have a possibility of degenerating into a shouting match. This would necessitate a moderator. The moderators would invariably come from the PC members of the conference where the paper was prevented. It is doubtful that PC members really want to make such a commitment. (All experiments repeated.)

As an author, I'd be glad to see people taking an interest in my paper, but a bit remiss about potentially having to spend a lot of time defending it. (Did not participate.)

A paper should be a snapshot of the research results at a certain point in time. Do we want to end up "maintaining" each individual paper ? (Did not participate.)

5. THE REPEATABILITY TESTING PROCESS

The repeatability evaluation process involved a lot of hard work, likely more so than it needed to be. The potential for simplification is available now that we have gained some experience with it. We explain the process below.

5.1 The timeline

Authors were required to upload, at most one month after the SIGMOD deadline, i.e. on December 16, 2007, on an INRIA-hosted FTP site, tarballs containing:

- the code and data needed to run the experiments subject to the repeatability test;
- an XML file describing the required hardware, software, instructions to install the code, to run experiments etc.;
- the PDF file containing the paper.

The latter was needed since the repeatability program committee (hereafter called the *rep PC*) did not have access to the conference management tool hosted by Microsoft [2], where the authors submitted papers. The converse was also true: neither the members of the SIGMOD regular PC nor the SIGMOD 2008 program chair had access to the FTP site. Care had been taken that the rep program committee be disjoint from the SIGMOD regular PC. This separation has been enforced (*i*) to preserve the anonymity of SIGMOD submission authors from the SIGMOD PC, as it was thought that code submission might leak the authors' identity to the rep PC; (*ii*) to prevent the result of repeatability assessment from influencing the SIGMOD acceptance decision.

We have used a second conference management tool, powered by MyReview [3], to manage *metadata* concerning the submissions, that is, their characterization according to the dimensions described in the XML file (OS, software, programming language, IP or other concerns preventing repeatability testing, etc.), and the repeatability reviews. To reduce authors' efforts, they had been asked only to access the FTP site. Therefore, the myReview site had to be filled in manually with 436 tuples extracted from the XML files. Unfortunately, most of the files were either not well-formed, or not valid according to the given DTD, which prevented the automation of this information gathering. For 41 papers we obtained no submission whatsoever. The myReview site was inaccessible to the regular SIGMOD PC, SIGMOD chair, and SIGMOD authors.

Around December 16, 2007, every paper should have had two reviews. On December 26, two ranked lists of paper IDs were sent by the SIGMOD proceedings chair (Denilson Barbosa, whom we thank for his many efforts!) to the rep PC. The first contained 34 papers with 3 positive reviews, sorted in descending order of their average overall. The second contained 48 papers with 2 positive reviews, similarly sorted. (The two lists were disjoint.)

On January 2, 2008, the 82 papers with good perspectives were evenly split among the repeatability reviewers. The rep PC was quite small. Therefore it focused on the (likely to be) accepted papers, and processed others only if there was extra time. (This did not happen.) Most papers were assigned just to one reviewer. Three papers, however, were assigned to 2 reviewers, to obtain some rough information on how much the repeatability result depends on the reviewer. (This variability is the topic of heated conversations in the context of regular SIGMOD reviewing.)

On February 22, 2008, we obtained from Denilson the list of IDs of accepted SIGMOD 2008 submissions, together with the contact information for each paper. Thus, the anonymity of SIGMOD 2008 accepted paper authors was breached to us, but only after the acceptance decision had been taken, and only concerning the accepted papers. Some of the papers initially assigned have not been accepted; and, some papers not previously assigned had been accepted. Accepted papers which had submitted code were immediately assigned.

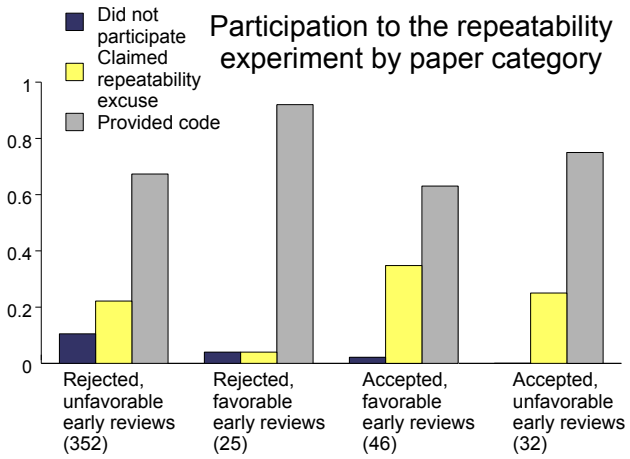

Figure 3: SIGMOD 2008 submissions and their participation to the repeatability assessment.

From February 22 to March 20, the rep PC interacted with the authors, in order to elicit from the authors missing information (the PDF file was frequently missing), and to get authors' help and feedback when their code did not function properly. Finally, inside the rep PC, several papers had to be co-assigned (given to a second rep PC member in parallel to the first one, for instance to parallelize execution of long-running experiments) or re-assigned (moved from one rep PC member to another). The goal of re-assignment was to improve the matching of the submitted papers with the hardware, software, technical know-how, and availabity of each rep PC member. The interactions took place via e-mail. All rep PC members shared an anonymous e-mail account to exchange messages with the authors. We are aware of at two papers for which the lack of time limited the interaction and potentially led to classifying some experiments as non-repeated.

Repeatability results were handed out until March 20, 2008 (the SIGMOD camera-ready deadline). Each paper's authors were given a snippet of text that they were invited to include in their camera-ready paper, explaining how much of their experiments had been repeated by our committee and in some cases, why this has not been possible (e.g. lack of time, special hardware etc.).

5.2 A quantitative view

Figure 3 presents a breakdown of all SIGMOD submission according to several dimensions. First, we distinguish accepted from rejected papers. Second, we distinguish those that had at least two favorable reviews by the end of December 2007 from the others. Figure 3 shows the number of papers in each of the three categories: authors who did not participate in the repeatability experiment but provided no excuse; provided an explanation (excuse) of why they did

not participate; and finally, provided code to be tested by the rep PC.

The numbers in Figure 3 lead to several observations. First, the percentage of papers claiming a repeatability excuse varies between 20% and 40% for various paper categories. The percentage of papers participating in the experiment lies between 60% and 90% across different categories. In particular, for many rejected papers the rep PC did receive a code submission, but did not have time to process it. Some authors of such authors have written to the rep PC complaining strongly about having made the effort of packing the code submission and not receiving any feedback.

Another interesting observation based on Figure 3 concerns the number of papers accepted in February which did not have good prospects in December: they represent almost half of all accepted papers. Assessing the code submissions corresponding to these papers in a short time interval was quite challenging.

6. LESSONS LEARNED

In this section we summarize operational lessons learned from the SIGMOD 2008 repeatability experiment.

6.1 Electronic Tools and Communications

Author feedback on the code. Similar to feedback for papers, the rep PC should be able to get code feedback from authors, when code installation fails, the results obtained by the rep PC raise some questions, or differ significantly from those in the paper. This year, one single round of messaging was never sufficient to get useful feedback. Therefore, we believe longer conversations should be supported by the CMT, in the style of paper discussions currently going on among the reviewers. Moreover, it may be very helpful to circulate files both ways, e.g., for reviewers to communicate their obtained output or for authors to send missing libraries, files etc. Thus, the support needed is similar to email with attachments.

Authors of at least four papers whose experiments were not all repeated have decided not to include a repeatability notice in the CR. These authors felt that the non-repeated stamp on some or all of their experiments does not do justice to their code. In all these cases, the code had portability or configuration errors which may have been fixed given more time. In one of these cases, the authors told us that they felt "awful for doing a sloppy job on the experiment submission". They prepared an independent open-source release of their code, as an alternative way of letting the community build on their results.

Avoiding conflicts of interest. Proper mechanisms need to be set in place to avoid conflicts of interest (CoI) between authors and the rep PC. Trying to best fit the hardware and software environment of the authors, with those of the rep PC, actually favors sending a batch of code to (close colleagues of) the paper authors for verification! Due to some missed CoIs, one paper was assigned to its own author, and another to close colleagues of the authors. (Both were re-assigned when this was noticed.)

Early notification. If the rep PC is to focus on the accepted papers, it needs to know which they are as soon as possible. Time is crucial for this process, in order to fit repeatability assessment tasks in the tight time frame available, as well as the possible interaction with the authors that it needs.

Single CMT. Paper and code submissions should be managed using a single CMT. This considerably facilitates management of paper metadata, paper discussion, and the early transmission of paper acceptance results to the rep PC. Observe that this does not imply that the regular and the rep PC should have the means to communicate or see each other's assessments. The CMT can be tuned to give the PC chair the option of enabling or not such communication.

A reviewing marketplace. One possible reviewing mechanism is to have authors of accepted papers who desire their results to be verified for repeatability to be required to review the results of two other accepted papers. This could lessen the burden needed between acceptance time and camera-ready submission time.

6.2 Code Submission Guidelines

We have used the SIGMOD conference Web site and, separately, emails to the authors of SIGMOD 2008 submissions. Authors were first instructed to provide text-based instructions in two files named INSTALL and HOWTO, but subsequently an XML file was solicited, which included more details about the hardware and software environment etc. In the end, authors provided one and/or the other. We have found the XML files much more informative and helpful in assigning papers to rep PC members. A future interface should ensure submitted XML files are well-formed and valid.

An important element missing from this year's XML file was the estimated time that it takes to run each experiment. This is a very useful piece of information, as it allows rep PC members to better allocate their time and the time of their available machines. The differences that may exist between submissions in this respect are much larger than when considering the regular reviewing process. Some code batches required 2-3 hours in all; others needed more than 20 days.

6.3 Code Assessment Guidelines

Rep PC members should inspect their assigned submissions when they are assigned to them, in order to establish which submissions concern long-running experiments, what extra software installation is needed, and to have sufficient time to reserve cycles on the machines available to them. This step is crucial: it can make or break the evaluation of a given code batch. Code should be installed very early on, in order to spot potential problems and leave sufficient time to contact the authors if needed and/or get extra help.

Rep PC members should initiate and conduct discussions with the authors concerning installation problems, unclear instructions, or unexpected experimental results. Such discussions should not reveal the identity of reviewers. Rep PC members should not be expected to do the authors' work, for instance automating their experiments or producing their graphs by cut and paste from number files in some graphic tool.

6.4 Repeatability

The most frequent obstacle to repeatability turned out to be the limited or non-existing code portability. Many of the submissions provided scripts and/or programs that contained hard-wired and "well hidden" configuration parameters—ranging from path names of both the submission itself and third-party software to access information and credentials

for database servers. In most cases, these parameters were not documented let alone obvious, and hence, could not be located and changed easily in the reviews environment. Moreover, even if documented, changing experimental parameters inside the source code by hand and recompiling the code for each parameter value is a vary tedious, time consuming and error-prone way to run experiments—not only for the reviewers but also for the authors.

Additionally, analyzing and patching failing experiments was often very complicated due to insufficient or completely missing error-detection, -reporting and/or -handling. A "segmentation fault" in case of absent input files or non-existing output directories does not help much to locate, understand and fix the problem. Scripts that go on running for days on an invalid input, produced by a failed experiment, made up a lot of the time spend on the repeatability evaluation.

Finally, many submissions produced raw performance results, sometimes hidden in up to 25 MB of (seemingly) unstructured result and log outputs. They produced neither the tables and graphs as shown in the paper nor did they extract the performance results supporting these tables and graphs in easy-to-find, documented, human readable files.

It appears very advisable to motivate authors to build more portable and parametrized experimental setups—not only for repeatability evaluations as done here, but also for the authors' own purpose, such as continuing research based on their prior work, experimenting with different parameters, using their code months or years after it has been initially written etc. Recommendations and guidelines on how to make experimental setups parametrized and hence portable and easily repeatable can be found in [1].

A pragmatic intermediate solution is to allow the authors to log in to the host machine after they have submitted their code in order to check that the code is working properly. Specific time slots could be allocated to specific authors to avoid possible overloading of the machines used for the submission.

7. CONCLUSION

The recognition of the value of repeatability is widespread. Here for example is the last call for the 2008 SIGKDD conference:

> *We need to take steps to ensure the long term viability of the research output of this community. A basic requirement is to enable the careful scrutiny and repeatability of evaluation results reported in a paper. The description of experimental results in submitted papers should be accompanied with all relevant implementation details and exact parameter specifications. Reviewers will be encouraged to downgrade ratings of papers that do not meet this guideline. Datasets used in the experiments should be made publicly available, whenever possible. When you must use proprietary datasets, please make every effort to supplement your results with those from closely matching synthetic datasets or other public datasets.*

Other efforts in the database research community to encourage good experimental practice and thorough experimental evaluation are reflected in the reviewing guidelines for the VLDB 2008 conference, as well as in the creation of a new Experimental track in VLDB 2008.

This paper by contrast reports on an explicit attempt at testing code and data, implemented as an optional step of the SIGMOD 2008 submission process. Our major findings can be summarized as follows:

1. Roughly 2/3 of submitters were willing to participate in the repeatability experiment, with most of the remaining 1/3 prevented to do so based on IP reasons. This 2/3 ratio applied almost equally to accepted and rejected papers.

2. The vast majority of those who participated found the process helpful to themselves and thought it raised the standards for the community.

3. This experiment required a lot of effort. Better workflow technology, better specification, and better interaction between authors and testers can mitigate this substantially.

We hope the results presented in this paper will contribute to the ongoing discussions concerning experimental repeatability in computer science systems research. Repeatability and archiving are easier in our field than in most. We can lead the way.

Acknowledgements

We thank all the authors who participated in the SIGMOD 2008 repeatability experiments. Without their good will and effort, this experience would not have been possible. We would also like to thank the SIGMOD executive committee who supported this experiment with remarkable cheerfulness. Finally, Jerome Simeon helped with several insightful comments.

8. REFERENCES

[1] I. Manolescu and S. Manegold. Performance Evaluation in Database Research: Principles and Experience. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, Cancun, Mexico, 2008. (Seminar/Tutorial). Slides are available from `http://www.icde2008.org/` or from the authors.

[2] The Microsoft Research Conference Management Tool. `https://cmt.research.microsoft.com`.

[3] The MyReview Conference Management System. `http://myreview.lri.fr`.