Developing Open Source Software As A Significant Part Of Faculty-Student Collaborative Research Projects

Paul J. Wagner
Department of Computer Science
University of Wisconsin – Eau Claire
wagnerpj@uwec.edu

Justin Sabelko
jbirdjavi@yahoo.com

Abstract

We have developed an open-source software component as a major part of a faculty/student collaborative research project. Specifically, we have developed an open source database benchmark for the Oracle database management system that will soon be contributed to the Open Source Database Benchmark (OSDB) project. We found that the process of developing this software complemented the research we did, and provided a number of benefits to the overall project. We also found that this idea raised several issues that we had to deal with to maintain the focus of the project.

Introduction

In our department's efforts to improve the quality of faculty-student collaborative research projects, we have previously examined how such projects integrate with the world outside of academia. While there are many factors that contribute to the success of such collaborative research projects, we see providing some type of larger-world benefit as being a cornerstone of any good and useful research.

One possible approach, appropriate for some but not all research projects, is to make software generated by the research project be "open source" [9], and in fact to make a significant part of the research project to be the development of a contribution to an open source project. Open source software is founded on the idea that software should be generated by and tested by an open community of interested users and developers, and that such software should be freely available to all interested users. Significant open source projects include Linux [5], many Unix utilities (e.g. sendmail), the Apache web server project and related sub projects such as Tomcat and Jakarta [1], and a variety of other software projects. The perceived benefits of open source software have been discussed in many papers, e.g. [12]. While the use of open source software has been discussed in general academic environments [2] and in teaching in certain computer science areas [7], we found no previous discussion of the use of open source software specifically in faculty/student collaborative research projects.

Our work began with the need for benchmark software for the Oracle [10] database management system (DBMS) and our investigation of whether such benchmarks had already been created. Database benchmarking software tests the speed of a DBMS while executing certain tasks on a computer system, and often is set up to work with several variables, including single user vs. multi-user, the number of processors involved, the type of statements being executed (e.g. transaction processing vs. data warehousing queries), and others. While we were aware of commercial benchmarks for Oracle and enterprise software systems using Oracle such as versions of the TPC [15] and SPEC [14] benchmarks, implementations of such benchmarks are expensive, and are therefore unavailable to many smaller software developers. Upon further search, we found no freely available Oracle benchmark packages.

However, we did discover the Open Source Database Benchmark (OSDB) project [8], which had at that time developed open source benchmarks for open source DBMSs such as PostgreSQL [11] and MySQL [6], but did not have implementations for the major commercial DBMSs such as Oracle, IBM's DB2 [3], and Microsoft's SQL Server [13]. OSDB included a prototype implementation of the Informix [4] DBMS (a commercial DBMS now owned by IBM) present, but that implementation has since gone to unsupported status.

To consider our project to be research, we realized that we needed more than just a software development project. As we familiarized ourselves with the OSDB project, we realized that research issues were certainly present. We stated the following core research issues for our work:

1. How would the implementation of a benchmark system for a commercial DBMS such as Oracle be different than implementing a benchmark for an open source DBMS such as MySQL or PostgreSQL? Without access to the source code of Oracle, we realized that our ability to take advantage of DBMS internals would be somewhat limited.

2. How difficult would it be to implement a benchmark for Oracle using the preexisting structure of the OSDB system? OSDB prides itself on taking a generalized, multi-level approach to benchmarks, which MySQL and PostgreSQL fit into easily as the base systems. However, it was not clear whether the structure of a more complex commercial DBMS such as Oracle would fit into this same framework.

These issues were the foundation of the application for developing our project as a faculty/student collaborative research project. Our overall goals became:

1. To investigate the research issues above, primarily by working…
2. To develop a prototype Oracle benchmark for the OSDB project, and then…
3. To contribute that prototype to the OSDB project.

Project

Project Process

As previously mentioned, the project itself was a combination of intertwined research and software development. As with any project, we first worked to build understanding of our domain. We worked to understand

- the design of OSDB in general to in turn understand the framework the previous developers had constructed for the benchmark suite;
- the structure of Oracle command processing environment (including Data Manipulation Language (DML) commands such as select, insert, update and delete, as well as DDL commands such as the various create and alter statements) to find out how well they matched the OSDB structure;
- the structure of transaction processing in Oracle; and
- the low-level Oracle Call Interface (OCI) API that would allow us to work with the DBMS at the necessary level for the benchmark.

Second, we applied that understanding to the development of the Oracle benchmark prototype. As we began to develop the necessary components of the benchmark in Oracle (e.g. data loading, transaction control, query execution, other statement execution) we also began to gather insight on our research questions. As often happens, we often found ourselves blocked temporarily by the details of what should have been relatively smaller issues, such as makefile difficulties and the subtleties of the OCI function calls.

Third, we completed our prototype, evaluated the thoughts and data we had recorded in regard to our research questions, and generated our initial project conclusions.

Project Results

We have a prototype implementation of an Oracle benchmark for OSDB, which will be submitted to the OSDB CVS software repository in March 2004. While there is more work to do to make this fit seamlessly into the OSDB framework, we think that it is better (and have been encouraged) to submit our prototype early rather than later, as others may then be inspired to test it and contribute to it.

As to our research questions, we have an initial indication that the commercial nature of Oracle may have some impact on our ability to generate an Oracle benchmark, as the Oracle internals are to some degree a black box to us, even after review of available literature on their details. This may also affect future developers as they work to fine tune the benchmark and ensure it's operating as efficiently as possible. We also found that Oracle's structure does not match well to MySQL's and PostgreSQL's structure, which means that we've had to map some features of Oracle in somewhat non-direct ways to the OSDB benchmark architecture. We have tentatively concluded that both open-source and commercial DBMSs should be considered when developing an overall DBMS benchmark architecture.

Advantages of Combining Open Source Software Development with Research

Developing open source software in the context of a faculty/student research project has several advantages. We discuss each of these advantages in more detail in the paragraphs below.

First, the student knows that his/her work is considered important by others, as it will be added to an ongoing software project. We had announced on the OSDB project email distribution list that we were working on an Oracle benchmark for the project, and several people responded and asked when our contribution would be available. This knowledge is a significant motivator for completion of the overall project.

Second, the student (and the faculty member) has an opportunity (and often a need) to interact with other software developers, primarily through the open source project email discussion lists. We had several of our own questions answered this way, and saw the resolution of other related issues for developers working on other benchmark projects. Such interaction provides insights not only in the domain of the project but also in how real world software development is accomplished under the open source development paradigm. It shows the student that good software developers have communication skills as well as technical skills, and allows the student to act as part of a larger team for the overall open source project.

Third, the final result of the research project is freely used in the larger world beyond the walls of academia. This in turn has several sub-benefits:

- it is a useful reminder that part of the academic role is service to the outside world;
- it allows the student (and faculty) involved to make a contribution to the computer science profession and the domain involved in the project; and
- it generates a useful entry for a student's resume that involves project work as well as research work.

Each of these sub-benefits is significant. Encouraging contribution of one's work to the open source world not only promotes the ideal of academic service to the larger community, but provides a concrete methodology for this service. The students and faculty involved in the project have made a contribution that will generate immediate feedback and usage, thereby supporting the advancement of computer science as a whole and the advancement of a particular computer science domain (in our case, database benchmarking). Such work is very positive for the student, as it shows contributions to the academic discipline and the larger world at both research and practical levels.

Issues

Several issues must be addressed that, if not dealt with, can become disadvantages or at least problems with this approach. We feel these issues are relatively minor, and can be dealt with by proper preparation and consideration.

First, it is important to find true research issues in such work, to ensure that the project does not become just a programming or software development assignment. On our particular project, we discussed this at length, and looked for questions that went beyond the technical details of the project. While we came up with some, we would like to have found more. While project work can certainly be a part of research when it contributes to the research goals, investigation and/or evaluation, there is not necessarily an inverse relationship whereby all software development includes research issues. Finding a proper balance between the research and project development work is an aspect of this concept that requires further investigation.

Second, as with all limited term projects, it is important to accurately judge the scope of the project, in order to allow all advantages mentioned above to occur. As often happens, we found ourselves rushed close to the project completion date, and some tasks we had hoped to accomplish on the benchmark prototype were left as future project work. We realized that this impacted the amount of communication we had with others working on OSDB – i.e. we would have communicated more if we had better gauged our project time. Ensuring that the project is reasonably accomplished within the given time frame would lead to increased incidence of the advantages discussed above. However, given the rigid schedule of an academic calendar, finding projects that are an exact fit for the term is not always possible, and working through all possible issues for a given project may sometimes be impractical.

Third, the merger of research and open source software development is certainly not suitable for all projects. Research projects that don't involve software development, that involve development using commercial software components or that may otherwise be suitable but have too large or too small of scope are not suitable candidates for this approach.

Future Work

This work has not included a formal evaluation or scientific study of the issues relating to the interaction between research projects and open-source software project contributions. Such a study would be both interesting and useful for the computer science education community. For now, we view this approach as something to consider and try for other faculty/student collaborative research projects.

As mentioned above, we also see investigation of the proper balance and combination of research work and software development work as a topic for further investigation. Ensuring that the research components of a project are maintained while meeting the practical deadlines of a software development project is a challenging task.

Conclusion

We feel that the inclusion of the open source software development component was a strong addition to our research project. The advantages discussed above far outweigh the possible disadvantages stemming from the issues mentioned. We intend to use this approach on future projects where practical, and hopefully will find ways of better evaluating it as well as fine-tuning the process.

References

1. Apache Software Foundation, http://www.apache.org
2. Davis, M., O'Donovan, W., Fritz, J., Childress,C., "Linux and Open Source in the Academic Enterprise", Proceedings of the 28th Annual SIGUCCS Conference, Richmond, VA, 2000, pp. 65-69.
3. DB2 DBMS, http://www.ibm.com/db2
4. Informix DBMS, http://www.informix.com
5. Linux operating system, http://www.linux.org
6. MySQL DBMS, http://www.mysql.com
7. O'Hara, K.J., Kay, J.S., "Investigating Open Source Software and Educational Robotics", The Journal of Computing in Small Colleges, Vol. 18, No. 3, February 2003.
8. Open Source Database Benchmark (OSDB) project, http://osdb.sourceforge.net

9. Open Source Software organization, http://www.opensource.org
10. Oracle DBMS, http://www.oracle.com
11. PostgreSQL DBMS, http://www.postgresql.org
12. Raymond, Eric, “The Cathedral and the Bazaar”; http://www.catb.org/~esr/writings/cathedral-bazaar/
13. SQL Server DBMS, http://www.microsoft.com/sql
14. Standard Performance Evaluation Corporation (SPEC), http://www.spec.org
15. Transaction Processing Performance Council (TPC), http://www.tpc.org