

On The Evolution Of User Support Topics in Computational Science and Engineering Software

K. Rupp¹, S. Balay², J. Brown^{2,3}, M. Knepley^{4,5}, L. C. McInnes², and B. Smith²

¹ Institute for Microelectronics, TU Wien, Austria

² Mathematics and Computer Science Division, Argonne National Laboratory, USA

³ Laboratory for Advanced Numerical Simulation, University of Colorado Boulder, USA

⁴ Computation Institute, University of Chicago, USA

⁵ Computational and Applied Mathematics, Rice University, USA

Abstract

We investigate ten years of user support emails in the large-scale solver library PETSc in order to identify changes in user requests. For this purpose we assign each email thread to one or several categories describing the type of support request. We find that despite several changes in hardware architecture as well programming models, the relative share of emails for the individual categories does not show a notable change over time. This is particularly remarkable as the total communication volume has increased four-fold in the considered time frame, indicating a considerable growth of the user base. Our data also demonstrates that user support cannot be substituted with what is often referred to as *better documentation* and that the involvement of core developers in user support is essential.

Introduction. User support and community interaction is a crucial ingredient for the success software packages in computational science and engineering (CSE) [1, 2]. With several disruptive changes to computing such as the paradigm shift to multi-core processors or the use of graphics processing units for general purpose computations in recent years, one may expect a substantial change in the type of user support requests for software libraries tailored to the efficient use of computing resources in CSE. Similarly, one may expect that the type of user requests changes as the user base of a software package grows.

In the following we provide answers to these questions for the large-scale solver library PETSc¹ by systematically evaluating email archives for the years 2006 to 2014. Our study is based on the public *petsc-users* mailinglist², which is the primary list for general user support.

Email Categorization. All email discussions for the years 2006 to 2014 have been systematically assigned to one or more of the following categories:

- **Algorithms:** In-depth discussions and general advice on the best mathematical approaches for setting up a solver for a given problem.
- **Beginner:** Short questions related to documented functionality or on how to deal with basic operations for distributed parallel systems in general.
- **Bug:** Bug reports by users on either the release version of PETSc or new functionality which has not been officially released yet.
- **Features:** Discussions about whether PETSc provides a certain feature, about future directions, or about better explanation of existing features from a user's perspective.
- **Performance:** Email communication about potential performance enhancements for a given machine or hardware architecture. In contrast to the *Algorithmic* section, the user is primarily interested in reducing execution times for an existing solver setup rather than the design of new types of solvers for the problem at hand.
- **Runtime Errors:** Discussions triggered by runtime errors. These may be due to errors in user code such as memory corruption, invalid use of functionality provided by PETSc, or missing functionality in PETSc.

¹PETSc webpage: <http://www.mcs.anl.gov/petsc/>

²Archive available online: <http://lists.mcs.anl.gov/pipermail/petsc-users/>

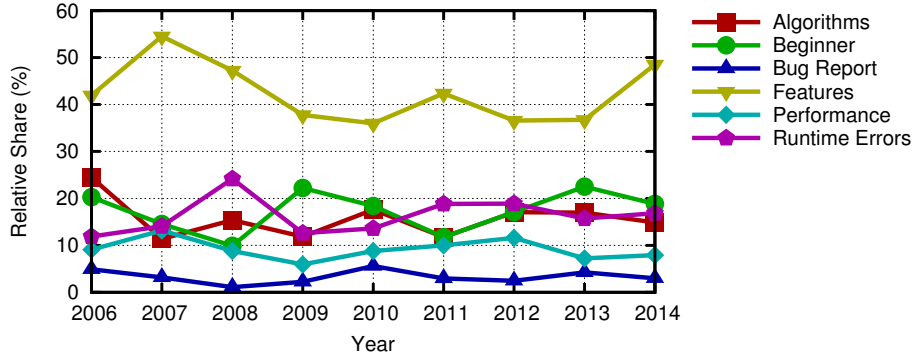


Figure 1: Categorization of email threads on the *petsc-users* mailing list.

We do not include configuration and installation issues here, because these are directed to the *petsc-maint* mailing list. Since the numbers of emails on *petsc-users* and *petsc-maint* are comparable, configuration and installation issues would be the dominating category if included in the list above. However, full data for the considered timeframe for *petsc-maint* is not available, yet empirical checks showed that the same trends hold true. For the sake of completeness we note that developer discussions on the *petsc-dev* mailinglist are about half of the email volume on *petsc-users* and *petsc-maint*.

Results and Discussion. The change in the relative share of email threads assigned to each category is depicted in Figure 1. Even though there is an inherent fluctuation in the relative share, none of the categories shows a significant net increase or reduction over the full time frame considered. Questions on features represent the largest category and amount to about 40 percent of the total email threads. The nature of these feature questions is such that these questions cannot be answered entirely through improved documentation. Thus, even if PETSc developers devoted all their time to documentation only, the total amount of support emails would not be reduced significantly.

Considering that the total number of emails to *petsc-users* was 964 in 2006 and 3947 in 2014, it is remarkable that the relative share of each category has changed by at most eight percent for each category when comparing 2006 with 2014. This suggests that the user base of PETSc has grown uniformly without preference for either highly experienced or less experienced users.

Overall, the results of our evaluations clearly underline that user support is an often underestimated ingredient for successful CSE software. Maintaining a sustainable software stack for CSE therefore not only requires funding for implementing new features from scratch, but also needs to account for the efforts required for preserving the health and integrity of existing packages. Also, our data shows that it is essential that core developers are involved in user support: All email threads in the category *Algorithms* and most in the category *Features* require in-depth knowledge of the library, hence core developers are essential for at least half of all email discussions.

References

- [1] W. Bangerth and T. Heister. What Makes Computational Open Source Software Libraries Successful? *Computational Science & Discovery*, 6:015010/1–18, 2013.
- [2] M. J. Turk. Scaling a Code in the Human Dimension. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*, XSEDE '13, pages 69:1–69:7, New York, NY, USA, 2013. ACM.