How To Contribute to PostgreSQL

Hari Babu Kommi

Your speaker

+61 2 9452 9087

haribabuk@fast.au.fujitsu.com

postgresql.fastware.com

Contents

- Need of contribution to the PostgreSQL community
- Advantages of contributing to community
- Contribution methods

Need of contribution

- PostgreSQL is a "world's most advanced open source database" that powers many mission critical systems around the world.

- It needs support from individuals and companies to continue it's development further to compete against major commercial databases.

Advantages of contribution

■ Contributing to community will benefit both individuals and companies.

■ All the individuals that contributes to PostgreSQL echo system are listed as contributors.
  - ■ https://www.postgresql.org/community/contributors/

■ All the companies that contributes to PostgreSQL echo system are listed as sponsors.
  - ■ https://www.postgresql.org/about/sponsors/

Contribution methods

- There are plenty of ways to contribute to PostgreSQL (echo system) and many people are already doing the same.

- This topic is to list out the some of the possible ways to contribute to PostgreSQL database by individuals and companies.

Contribution methods

- Donation
- Participation in Survey
- Hardware/Infrastructure support
- Web updates/support
- Events/Meetup groups
- Documentation
- Translation support
- Answering questions
- Build farm support
- Bugs
- Writing tools/extensions
- Writing feature/bug fix patches
- Reviewing of Patches

Donation

- The Simple way to support PostgreSQL NFO is by contributing some donation.

- The funds that are collected by the PostgreSQL groups are used for various activities such as education, user groups and advocacy.

- https://www.postgresql.org/about/donate/

Participation in Survey

- PostgreSQL usually conducts surveys for every 4 to 6 months.
  - https://www.postgresql.org/community/

- Why is it so important to participate in survey?

- Where the survey results are used and how?

User Survey

What PostgreSQL 10 Feature are you most excited about?

- ◯ Logical replication
- ◯ Native table partitioning
- ◯ More query parallelism
- ◯ Quorum sync replication
- ◯ SCRAM authentication
- ◯ More FDW push-down
- ◯ libpq connection "failover"
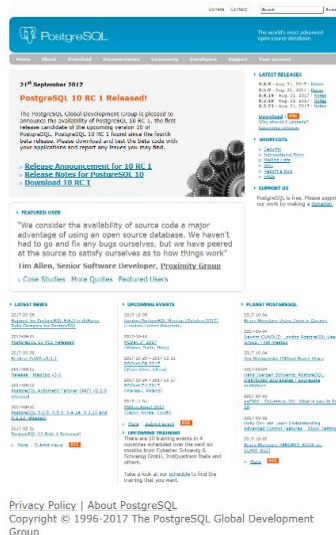- ◯ Other

Vote

Results

Participation in Survey

- The results of the latest survey are available in the following link:
  - https://www.postgresql.org/community/survey/93-what-postgresql-10-feature-are-you-most-excited-about/

- As you observe that, there are less number of people responded to the survey.

- Currently the open survey is not reaching many users, needs to identify other approaches by the infrastructure team.

Answer	Responses	Percentage
Logical replication	122	30%
Native table partitioning	144	36%
More query parallelism	66	16%
Quorum sync replication	16	4%
SCRAM authentication	7	1%
More FDW push-down	16	4%
libpq connection "failover"	11	2%
Other	17	4%
Total	399	

Hardware/Infrastructure support

- The servers that power the services of postgresql.org are provided by different companies and organisations around the world.
  - https://www.postgresql.org/about/servers/

- Hardware support can be a test machine or a performance machine

- I listed a Power2 machine sponsored by IBM to PostgreSQL for performance testing.

- Power2 configuration
  - Architecture: ppc64le
    Byte Order: Little Endian
    CPU(s): 192
    On-line CPU(s) list: 0-191
    Thread(s) per core: 8
    Core(s) per socket: 1
    Socket(s): 24
    NUMA node(s): 4
    Model: IBM,8286-42A
    L1d cache: 64K
    L1i cache: 32K
    L2 cache: 512K
    L3 cache: 8192K
    NUMA node0 CPU(s): 0-47
    NUMA node1 CPU(s): 48-95
    NUMA node2 CPU(s): 96-143
    NUMA node3 CPU(s): 144-191

Web updates/support

■ The infrastructure team itself takes care of web updates/support. Interested people can register with the following.

  ■ https://lists.postgresql.org/manage/ (pgsql-www)

■ Currently PostgreSQL website doesn't scale well on mobile devices.

■ There is no proper details/issues that are available anywhere in the website, so that people can participate in the development. It needs an improvement from community to increase the contribution.
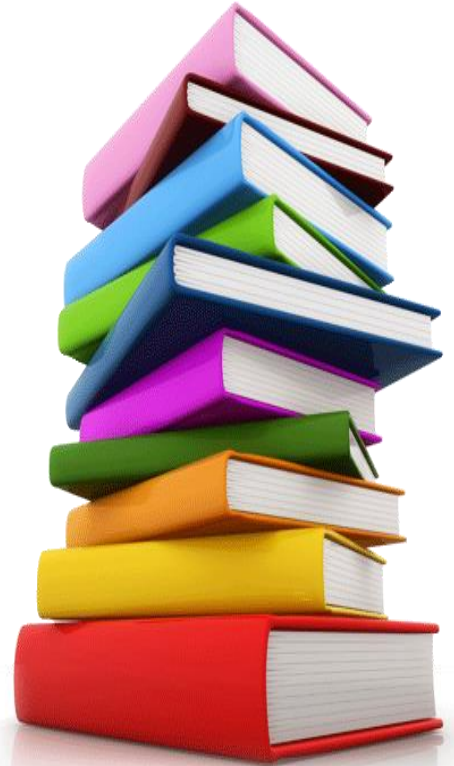
Events/Meetup groups

- Thanks for the Organizers and Sponsors of the PG Day Australia event.

- Conducting events/local meetups will help in growing the PostgreSQL community.

- Try to conduct monthly/quarterly meetups.

- Following are some of the local meetup groups in Australia.
    - https://www.meetup.com/Sydney-PostgreSQL-User-Group/
    - https://www.meetup.com/melpug/
    - https://www.meetup.com/Brisbane-PostgreSQL-User-Group/

Documentation

- PostgreSQL community provides a very high quality documentation, but there are cases that it may not handle everything and some information may misleading and etc.

- Register to the following mailing list and provide your updates including the review of the changes suggested by others.
  - https://lists.postgresql.org/manage/ (pgsql-docs)

- If you're a document developer or knows the tools that are used for the documentation, you valuable information is welcomed.

Translations support

- PostgreSQL programs (server and client) can issue their messages in your favourite language.

- Creating and maintaining translated message sets needs the help of people who speak their own language well and want to contribute to the PostgreSQL effort.

- All the translation issues are tracked in redmine.
  - https://redmine.postgresql.org/projects/pgtranslation

- There is a dedicated mailing list to discuss the translations updates/issues.
  - https://www.postgresql.org/list/pgsql-translators/

Answering questions

- Try to answer the simple to complex questions that are raised by the users not only related to the following,
  - Performance
  - General discussions
  - Administration
  - Etc.

- PostgreSQL community mailing list is very supportive, you will receive all the information that you needed.

Build farm support

- Build farm is something like PostgreSQL continuous integration environment.

- Currently there are extensive set of animals that are present in the build farm, but still there is a need of some more machines with different set of configurations that are used by users in the production environment.

- If you have some specific hardware/software where you want to PostgreSQL to be running without fail, it is better add that configuration as a build farm animal if that doesn't exist.

Legend	= cassert	= debug	= git	= integer-datetimes	= krb5	= nls	= openssl
	= pam	= perl	= python	[tcl] = tcl		= thread-safety	V = vpath	X = xml

		Branch: HEAD		
Alias	**System**	**Status**		**Flags**
baiji	Vista Ultimate 6.0.6000 MSVC 2005 Pro 8.0.50727.867 x86	08:11:29 ago Make **Details**		
bowerbird	Windows 8 8.1 Pro Visual Studio 2012 x86_64	08:22:54 ago Make **Details**		
currawong	Windows XP-PRO SP3 MSVC++ 2008 Express i686	08:38:24 ago Make **Details**		

Bugs

- PostgreSQL community review and it's extensive build farm support catches most of the problems, but still there may be problems that can occur only with production data.

- Need your support in validating new features or performance enhancements with respect to quality and functionality.

- Many people will face problems in sharing the bug details to the community. Steps to follow for quicker bug fix,
  - A reproducible test steps
  - If not, try to provide a call stack with debug symbols.
  - Or provide a core dump in a shared location.

Bugs

- Once all details are captured, raise the bug with the following bug reporting form.
  - https://www.postgresql.org/account/submitbug/

- Try to register to the following mailing list to check the progress of the bug and also the bug patterns, that can help in further testing.
  - https://lists.postgresql.org/manage/ (pgsql-bugs)

Writing tools/extensions

■ PostgreSQL have many variety of tools and extensions that are useful for many applications.

■ Before writing any tool/extension by your own for your needs, please make sure that the existing are supported the same.
  ■ https://www.postgresql.org/download/product-categories/

■ Writing an extension is easier than writing a core feature.

Writing feature/bug fix patches

- If you found a bug or interested in fixing bugs raised by others, generate a patch and post it to the mailing list.

- Community acts very quickly in fixing any bugs that arise.

- Try to clarify the users when they raised a bug, but actually that is an expected behavior with details.

Writing feature/bug fix patches

- If you found some interesting feature and thought of to be present in PostgreSQL.
    - Find out the use case scenario
    - Check for any older discussion on the mailing list
    - Post your idea to community.

- Once everyone agrees and come to a common approach, generate a POC patch.

- Every feature that is submitted to PostgreSQL, has to follow the review process.
    - https://commitfest.postgresql.org/

- Definitely there may be complete rewrite of the patch based on the feedback.

Reviewing Patches

- PostgreSQL is a community database, it needs support from you not just only features, but also from reviewing the submitted patches.

- Please make sure that if you submit a patch, make sure that you reviewed another patch.

- Review can be anything from source code, documentation, test and etc.

- PostgreSQL community started recognizing the contribution from the reviewers also by listing their names in the release notes from version 10.

- In the past, many patches are not received the much feedback from reviewers especially from actual users of those features.