DPDK.org services, latest and future planned improvements

ALI ALNUBANI
MELLANOX
Overview of dpdk.org services (as of 2018)

- **Main websites:**
  - [www.dpdk.org](http://www.dpdk.org) - News and events, Managed by LF. Built with WordPress - [https://make.wordpress.org/](https://make.wordpress.org/)
  - [core.dpdk.org](http://core.dpdk.org) - Core components. Built with Hugo - [https://gohugo.io/contribute/](https://gohugo.io/contribute/)

- **Documentation:**
  - [doc.dpdk.org/api](http://doc.dpdk.org/api) Built with Doxygen [https://github.com/doxygen/doxygen/doxygen]

- **CDN:**
  {fast,static}.dpdk.org. Hosted by fastly. [fastly](https://www.fastly.com)
Overview of dpdk.org services (as of 2018) - Cont'd

• cgit - repository browser - git.dpdk.org
  Contributing to cgit: https://git.zx2c4.com/cgit/

• Patchwork - patch tracking system - patches.dpdk.org
  Contributing to Patchwork: https://github.com/getpatchwork/patchwork

• Mailman - mailing list manager - mails.dpdk.org
  Contributing to Mailman: https://wiki.list.org/DEV/MailmanOnLaunchpad

• Bugzilla - bug tracking system - bugzilla.dpdk.org
  Contributing to Bugzilla: https://github.com/bugzilla/bugzilla
The DPDK Mailing Lists

• DPDK.org uses GNU Mailman 2.
• Mailman 2 uses the internal archiver (Pipermail).
• Archives available at mails.dpdk.org/archives/.
Limitations/Issues of Mailman 2 archives

- Plaintext passwords.
- No support for text searching through threads.

Alternatives are:
- Searching external mirrors:
  - https://marc.info/?l=dpdk-*
  - https://www.mail-archive.com/*/dpdk.org/
  - inbox.dpdk.org
- Using web search engines (e.g., "some error message site:mails.dpdk.org").
- Searching with your mail client.
• Hard to reply to old threads. You need to either:
  • Import mails.dpdk.org/archives/dev/<YEAR>-<MONTH>.txt.gz into your mail client.
  • Look for the Message-ID in the page source and manually use it as the In-Reply-to.
    In both cases, you will need to manually set To, and Cc.
Public-Inbox Archives

- DPDK mailing lists were recently mirrored via a public-inbox instance.
- Public-Inbox provides multiple features:
  - Archives are searchable through the web interface.
  - mbox per thread.
  - Emails are stored as git objects:
    ➢ Archives are cloneable.
    ➢ Archives are searchable with git commands.
  - Atom feeds.
  - nntp accessible.
  - UI is customizable using userContent.css. Start with `inbox.dpdk.org/dev/userContent.css` as a template.
  - Message-ID-based URL format.

For more info, see [https://public-inbox.org](https://public-inbox.org)
• DPDK PUBLIC-INBOX ARCHIVES ARE AVAILABLE AT:
  • http://inbox.dpdk.org/
  • https://lore.kernel.org/dpdk-dev/
Contributing to core.dpdk.org

- The site core.dpdk.org is built with Hugo.
- Syntax is markdown. Hugo renders it to HTML.
- To setup a development environment:
  - Follow the installation guide [https://gohugo.io/getting-started/installing/](https://gohugo.io/getting-started/installing/).
  - `$ git clone git://dpdk.org/tools/dpdk-web --recurse-submodules`
  - `$ cd dpdk-web; hugo server`
- Follow the "Improve this page" link on the bottom of each page.
- Make your changes and send patches to web@dpdk.org.
Elixir – Source Code Cross-Referencer

- Web-based browsing of source code, with links to the definition and usage of any identifier across all releases.[1]
- Queries can be performed with CLI as well.
- Available at code.dpdk.org.

- Source code: https://github.com/bootlin/elixir
Elixir – Queries

$ ./query.py ident v18.11 rte_pci_get_sysfs_path

Defined in 2 files:
drivers/bus/pci/pci_common.c: 34 (function)
drivers/bus/pci/rte_bus_pci.h: 34 (prototype)

Referenced in 6 files:
drivers/bus/pci/linux/pci.c: 399, 472, 487, 594
drivers/bus/pci/linux/pci_vfio.c: 652, 811, 927, 964
drivers/bus/pci/pci_common.c: 34
drivers/bus/pci/rte_bus_pci.h: 34
drivers/net/ifc/ifcvf_vdpa.c: 122
Patchwork - git-pw

- Download/apply/query patches and series.
- It’s a Python package.
  - $ pip install --user git-pw.
  - Or install from source: https://github.com/getpatchwork/git-pw.
- Configuration:
  - $ git config pw.server 'https://patches.dpdk.org/api/1.0/'
  - $ git config pw.project 'dpdk'
  - $ git config pw.token 'TOKEN' # generated from https://patches.dpdk.org/user.
- Example usage:
  - $ git-pw series apply|download <SERIES_ID>
  - $ git-pw patch apply|download <PATCH_ID>
  - $ git-pw patch list --submitter <SUBMITTER>
Patchwork - Special email headers

- **X-Patchwork-Hint: ignore**
  Tells Patchwork to ignore the patch.

- **X-Patchwork-Delegate: DELEGATE**
  Sets the patch's delegate.

- **X-Patchwork-State: STATE**
  Sets the patch's state (Accepted, RFC, etc..)

- `git send-email -1 --add-header="<HEADER>" --to dev@dpdk.org`
Bugzilla - REST API

- Root endpoint: `/rest`
- Get/Create/Update/Search bugs/comments.
- Get information about products and components.
- and more.

Bugzilla - Creating a bug via the rest API

- Generate an API key from https://bugs.dpdk.org/userprefs.cgi?tab=apikey.

- To create a bug:

  ```json
  POST /rest/bug

  {
    "product" : "DPDK",
    "component" : "core",
    "version" : "19.08",
    "summary" : "Something isn't right",
    "description" : "Long description",
    "assigned_to" : "user@example.com",
    ...
  }
  ```
requests.post(
    'https://bugs.dpdk.org/rest/bug',
    data=json.dumps(data),
    params={'api_key': 'API_KEY'},
    headers={
        'Content-Type': 'application/json',
        'Accept': 'application/json'})

curl --data '{
    "api_key": "API_KEY",
    "product": "DPDK",
    "component": "core",
    "summary": "Something is not right",
    "version": "19.08"}'
--header "Content-Type: application/json"
--header "Accept: application/json"
--request POST https://bugs.dpdk.org/rest/bug
Bugzilla - Creating a bug via email

@product = DPDK | dpdk-burst-replay | lab
@component = core | cryptodev | doc | meson | ..
@version = 19.08 | 19.05 | ..
@severity = critical | major | normal | minor | enhancement
@cc = user1@example.com,user2@example.com
@assigned_to = user3@example.com
@platform = ARM | POWER | x86 | Other
@op_sys = Linux | FreeBSD | Other
@bug_id = <BUG_ID>

This is the bug description.
Future Plans

- IRC bot?
  - Bugzilla notifications.
  - Release notifications.
  - Documentation queries.
- Mailman 3?
  - First released in 2015.
  - More modern looking.
  - Single user account across all lists.
  - REST api.
  - Searchable archives.
  - Example: https://lists.fedoraproject.org
- Patchwork:
  - Add the DTS project.
  - Link patches/series to their mailing list threads.
  - Automatic delegation based on the maintainers file.
  - Whine about patches with no activity.
Thank you

Questions?

Suggestions?