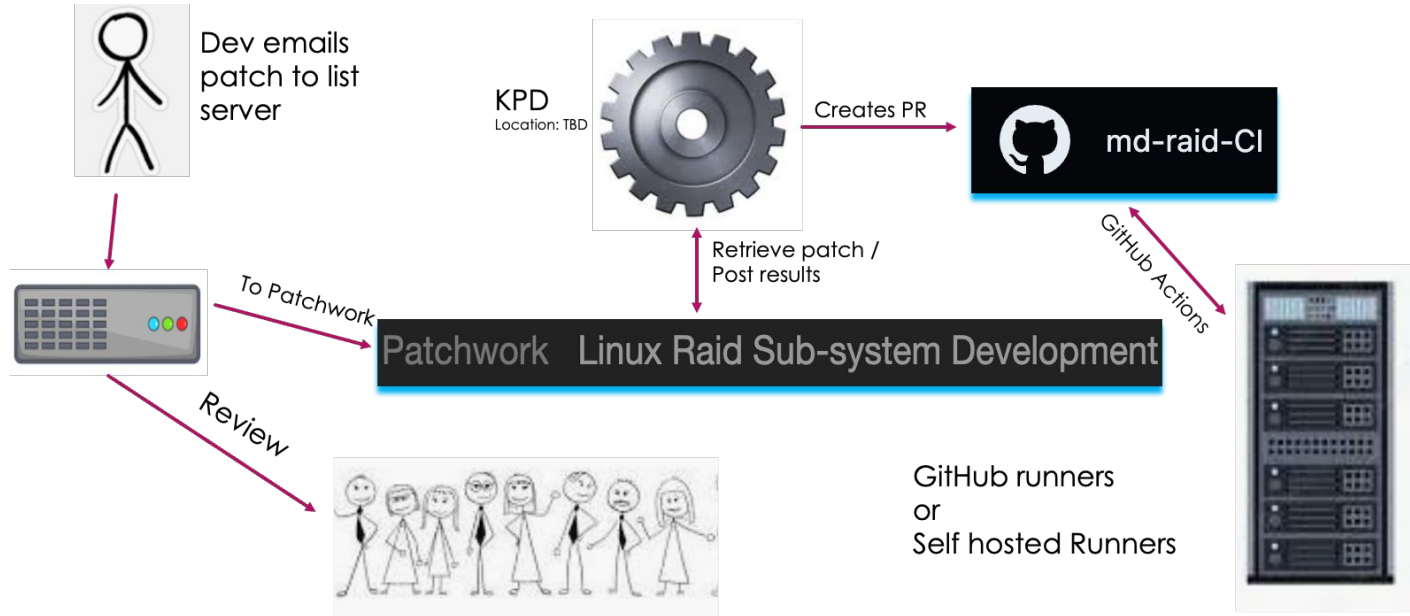


Get Started with KPD

Song Liu, Paul Luse, Manu Bretelle

Kernel Patches Daemon (KPD)

- <https://github.com/facebookincubator/kernel-patches-daemon>
- Connect patchwork and GitHub



Existing Users

- Netdev & bpf: <https://patchwork.kernel.org/project/netdevbpf/list/>
 - x86_64, arm64, s390
- RISC-V: <https://patchwork.kernel.org/project/linux-riscv/list/>
- Maybe more

Context	Check	Description
conchuod/vmtest-for-next-PR	success	PR summary
conchuod/patch-1-test-1	success	.github/scripts/patches/tests/build_rv32_defconfig.sh
conchuod/patch-1-test-2	success	.github/scripts/patches/tests/build_rv64_clang_allmodconfig.sh
conchuod/patch-1-test-3	success	.github/scripts/patches/tests/build_rv64_gcc_allmodconfig.sh
conchuod/patch-1-test-4	success	.github/scripts/patches/tests/build_rv64_nommu_k210_defconfig.sh
conchuod/patch-1-test-5	success	.github/scripts/patches/tests/build_rv64_nommu_virt_defconfig.sh
conchuod/patch-1-test-6	success	.github/scripts/patches/tests/checkpatch.sh
conchuod/patch-1-test-7	success	.github/scripts/patches/tests/dtb_warn_rv64.sh
conchuod/patch-1-test-8	success	.github/scripts/patches/tests/header_inline.sh
conchuod/patch-1-test-9	success	.github/scripts/patches/tests/kdoc.sh
conchuod/patch-1-test-10	success	.github/scripts/patches/tests/module_param.sh
conchuod/patch-1-test-11	success	.github/scripts/patches/tests/verify_fixes.sh
conchuod/patch-1-test-12	success	.github/scripts/patches/tests/verify_signedoff.sh

Context	Check	Description
qemu/vmtest-for-next-PR	success	PR summary
qemu/vmtest-for-next-VMX_Test-0	success	Logs for Lint
qemu/vmtest-for-next-VMX_Test-0	success	Logs for UnitTests
qemu/vmtest-for-next-VMX_Test-1	success	Logs for StaticCheck
qemu/vmtest-for-next-VMX_Test-2	success	Logs for Valgrind memcheck
qemu/vmtest-for-next-VMX_Test-3	success	Logs for asanbpf-gcc / build release
qemu/vmtest-for-next-VMX_Test-4	success	Logs for asanbpf-gcc / build / build for asanbpf with gcc
qemu/vmtest-for-next-VMX_Test-5	success	Logs for asanbpf-gcc / test (test_verify, false, 300) / test_verify on asanbpf with gcc
qemu/vmtest-for-next-VMX_Test-12	success	Logs for x390-gcc / build release
qemu/vmtest-for-next-VMX_Test-16	success	Logs for x390-gcc / test (test_verify, false, 300) / test_verify on x390s with gcc
qemu/vmtest-for-next-VMX_Test-10	success	Logs for asanbpf-gcc / verisat
qemu/vmtest-for-next-VMX_Test-11	success	Logs for x390-gcc / build / build for x390s with gcc
qemu/vmtest-for-next-VMX_Test-13	success	Logs for x390-gcc / test (test_map, false, 300) / test_map on x390s with gcc
qemu/vmtest-for-next-VMX_Test-20	success	Logs for x86_64-gcc / build release
qemu/vmtest-for-next-VMX_Test-17	success	Logs for x390-gcc / verisat
qemu/vmtest-for-next-VMX_Test-18	success	Logs for test-matrix
qemu/vmtest-for-next-VMX_Test-19	success	Logs for x86_64-gcc / build / build for x86_64 with gcc
qemu/vmtest-for-next-VMX_Test-21	success	Logs for x86_64-gcc / test (test_map, false, 300) / test_map on x86_64 with gcc
qemu/vmtest-for-next-VMX_Test-22	success	Logs for x86_64-llvm-17 / test (test_verify, false, 300) / test_verify on x86_64 with llvm-17
qemu/vmtest-for-next-VMX_Test-24	success	Logs for x86_64-llvm-17 / build release / build for x86_64 with llvm-18 and -O2 optimization
qemu/vmtest-for-next-VMX_Test-26	success	Logs for x86_64-llvm-17 / build / build for x86_64 with llvm-18
qemu/vmtest-for-next-VMX_Test-28	success	Logs for x86_64-gcc / test (test_verify, false, 300) / test_verify on x86_64 with gcc
qemu/vmtest-for-next-VMX_Test-29	success	Logs for x86_64-llvm-17 / build release / build for x86_64 with llvm-17 and -O2 optimization
qemu/vmtest-for-next-VMX_Test-26	success	Logs for x86_64-gcc / test (test_verify, false, 300) / test_verify on x86_64 with gcc
qemu/vmtest-for-next-VMX_Test-41	success	Logs for x86_64-llvm-18 / test (test_verify, false, 300) / test_verify on x86_64 with llvm-18
qemu/vmtest-for-next-VMX_Test-42	success	Logs for x86_64-llvm-18 / verisat
qemu/vmtest-for-next-VMX_Test-6	success	Logs for asanbpf-gcc / test (test_map, false, 300) / test_map on asanbpf with gcc
qemu/vmtest-for-next-VMX_Test-7	success	Logs for asanbpf-gcc / test (test_map, false, 300) / test_map on asanbpf with gcc
qemu/vmtest-for-next-VMX_Test-15	success	Logs for x390-gcc / test (test_map, no_asanbpf, false, 300) / test_map on x390s with gcc
qemu/vmtest-for-next-VMX_Test-14	success	Logs for x390-gcc / test (test_map, no_asanbpf, false, 300) / test_map on asanbpf with gcc
qemu/vmtest-for-next-VMX_Test-23	success	Logs for x86_64-gcc / test (test_map, no_asanbpf, false, 300) / test_map on x86_64 with gcc
qemu/vmtest-for-next-VMX_Test-31	success	Logs for x86_64-llvm-17 / test (test_map, false, 300) / test_map on x86_64 with llvm-17
qemu/vmtest-for-next-VMX_Test-22	success	Logs for x86_64-gcc / test (test_map, false, 300) / test_map on x86_64 with gcc
qemu/vmtest-for-next-VMX_Test-37	success	Logs for x86_64-llvm-17 / test (test_map, false, 300) / test_map on x86_64 with llvm-18
qemu/vmtest-for-next-VMX_Test-25	success	Logs for x86_64-gcc / test (test_map_parallel, true, 300) / test_map_parallel on x86_64 with gcc
qemu/vmtest-for-next-VMX_Test-30	success	Logs for x86_64-llvm-17 / test (test_map, false, 300) / test_map on x86_64 with llvm-17
qemu/vmtest-for-next-VMX_Test-07	success	Logs for x86_64-gcc / verisat / verisat on x86_64 with gcc
qemu/vmtest-for-next-VMX_Test-40	success	Logs for x86_64-llvm-18 / test (test_map, no_asanbpf, false, 300) / test_map, no_asanbpf on x86_64 with llvm-18
qemu/vmtest-for-next-VMX_Test-24	success	Logs for x86_64-gcc / test (test_map, no_asanbpf_parallel, true, 300) / test_map, no_asanbpf_parallel on x86_64 with gcc
qemu/vmtest-for-next-VMX_Test-32	success	Logs for x86_64-llvm-17 / test (test_map, no_asanbpf, false, 300) / test_map, no_asanbpf on x86_64 with llvm-17
qemu/vmtest-for-next-VMX_Test-39	success	Logs for x86_64-llvm-18 / test (test_map_parallel, false, 300) / test_map_parallel on x86_64 with llvm-18
qemu/vmtest-for-next-VMX_Test-38	success	Logs for x86_64-llvm-18 / test (test_map, false, 300) / test_map on x86_64 with llvm-18
netdev/headers_format	success	Printing correctly formatted
netdev/headers_selection	success	Clearly marked for test-net
netdev/pyl	success	Generated files up to date, no warnings/errors, no diff is generated.
netdev/flags_present	success	Flags tag not required for -netdev entries
netdev/headers_inline	success	No static functions without inline keyword in header files
netdev/build_32bit	success	Errors and warnings before: 0 this patch: 0
netdev/build_tools	success	No tools touched, skip
netdev/cv_maintainers	success	CCed 17 of 17 maintainers
netdev/build_clang	success	Errors and warnings before: 0 this patch: 0
netdev/verify_signedoff	success	Signed-off-by tag matches author and committer
netdev/deprecated_api	success	None detected
netdev/check_offset	success	No not selftest shell script
netdev/verify_fixes	success	No Fixes tag
netdev/build_allmodconfig_warn	success	Errors and warnings before: 0 this patch: 0
netdev/headerspatch	success	Total: 0 errors, 0 warnings, 0 checks, 0 lines checked
netdev/build_clang_test	success	No Rust files in patch, skipping build
netdev/hdoc	success	Errors and warnings before: 0 this patch: 0
netdev/source_inline	success	Wrote 0 new: 0

Get Started with KPD

- Use md/raid patchwork as example
- Run build test for the patches
- Run KPD on Oracle Cloud free VM (free)
- Use GitHub runner to build the kernel (free)
- Update results to patchwork (free)

GitHub Account and Kernel Code Repository

- Create a free organization
- Import kernel code to a repository
 - Do NOT fork from <https://github.com/torvalds/linux>
 - Instead do the import from <https://github.com/torvalds/linux> or `https://git.kernel.org/pub/scm/linux/kernel/git/<user>/<tree>.git`
 - This makes sure PR to this repository doesn't go to torvalds/linux by default
 - The import will take some time and maybe also some retries

GitHub Account and Kernel Code Repository

 **md-raid-utilities** / **md-ci-linux** Public

[Code](#) [Issues](#) [Pull requests](#) 9 [Actions](#) [Projects](#) [Security](#) [Watches](#)

[md-6.10_base](#) [14 Branches](#) [249 Tags](#)

 **Oracle Public Cloud User** adding ci files

 .github/workflows	adding ci files
 Documentation	Merge tag 'kbuild-fixes-v
 LICENSES	LICENSES: Add the copyl
 arch	Merge tag 'kbuild-fixes-v

Create a GitHub App

- Goto <https://github.com/organizations/<your org>/settings/apps>
- Create a new app
- Give the app read write permissions of “Content”, “Pull request”, and “Workflow”
- Create a private key for the app (KPD uses this)
- Install the app to your organization, and give it access to the kernel code repository we just created
- Get App ID and Installation ID (KPD uses these)

Create Repository for GitHub Actions

- A lot of things can be done with GitHub Actions
- We will keep it simple here
 - Only one file: `.github/workflows/kernel_build.yml`
- Manu will talk about more details in the BPF track

name: md-ci

on:

pull_request:

.github/workflows/kernel_build.yml
1

jobs:

build-kernel:

runs-on: ubuntu-latest

steps:

- name: Configure git
run: |
git config --global --add safe.directory '*'
- name: Checkout git
run: |
sudo apt-get install -y libelf-dev
mkdir -p linux
cd linux
git init
git remote add origin https://github.com/\${{ github.repository }}
git fetch origin --depth=5 \${{ github.event.pull_request.head.sha }}
git reset --hard \${{ github.event.pull_request.head.sha }}
git log -1
- name: Build kernel
run: |
cd linux
make defconfig
make -j 8

Inspired by BPF and RISC-V yml files

Patchwork Account

- Assume you already have a patchwork project
 - If not, send an email to helpdesk@kernel.org
- Create an patchwork account, and make it the maintainer of your project
 - Or use your own account
 - You may also need helpdesk@kernel.org
- Create an API token for the account
- Find your project ID from <https://patchwork.kernel.org/api/1.2/projects/>

Start Kernel Patches Daemon

- <https://github.com/facebookincubator/kernel-patches-daemon>
- Follow commands to install it
- To run it, use

```
poetry run python -m kernel_patches_daemon --config  
<config_path> --label-color configs/labels.json
```

The config file

```
{
  "version": 3,
  "patchwork": {
    "server": "patchwork.kernel.org",
    "project": "linux-raid",
    "search_patterns": [{"archived": false, "project": 411}],
    "api_username": "<patchwork_id>",
    "api_token": "<patchwork_token>",
    "lookback": 7
  },
  "tag_to_branch_mapping": {"__DEFAULT__": ["md-6.10"]},
  "branches": {
    "md-6.10": {
      "github_app_auth": {
        "app_id": <github_app_id>,
        "installation_id": <github_app_installation_id>,
        "private_key_path": "<path_to_github_private_key>"
      },
      "repo": "https://github.com/md-raid-utilities/md-ci-linux",
      "repo_branch": "md-6.10",
      "upstream": "https://git.kernel.org/pub/scm/linux/kernel/git/song/md.git",
      "upstream_branch": "md-6.10",
      "ci_repo": "https://github.com/md-raid-utilities/vmtest.git",
      "ci_branch": "main"
    }
  },
  "base_directory": "/tmp/repos"
}
```

Tip for Running KPD on Oracle free VM

- When KPD runs for the first time, it clones kernel code to `/tmp/repos/pw_sync_md-ci-linux_<sha>`
- This may take a long time on Oracle free VM, because the memory is too small, and git clone causes a lot of swapping
- One way to fix this issue is to clone the kernel code repository on a beefier machine, and scp the tarball to the Oracle free VM
- We only need this once. KPD reuses the local checkout on restart

The Result

Patch	Series	A/R/T	S/W/F	▲ Date	Submitter
[md-6.10] md: remove parameter check_seq for stop_sync_thread()	[md-6.10] md: remove parameter check_seq for stop_sync_thread()	---	2 - -	2024-05-11	Yu Kuai

Checks

Context	Check	Description
mdraidci/vmtest-md-6-10-PR	success	PR summary
mdraidci/vmtest-md-6-10-VM_Test-0	success	Logs for build-kernel

Commit Message

Yu Kuai

From: Yu Kuai <yukuai3@huawei.com>

Caller will always set MD_RECOVERY_FROZEN if check_seq is true, and always clear MD_RECOVERY_FROZEN if check_seq is false, hence replace the parameter with test_bit() to make code cleaner.

Signed-off-by: Yu Kuai <yukuai3@huawei.com>

← md-ci

✓ md: remove parameter check_seq for stop_sync_thread() #32

Summary

Jobs

✓ build-kernel

Run details

Usage

Workflow file

build-kernel

succeeded 7 hours ago in 10m 17s

> ✓ Set up job

> ✓ Configure git

> ✓ Checkout git

> ✓ Build kernel

> ✓ Complete job

Acknowledgement

- Thanks to folks who developed the KPD framework
- To name a few of them, in alphabetical order
 - Andrii Nakryiko, Daniel Müller, Daniel Xu, Manu Bretelle, Mykola Lysenko, Nikolay Yurin, Sergei Iudin, Yucong Sun, and more
- To contribute to KPD
 - <https://github.com/facebookincubator/kernel-patches-daemon/blob/main/CONTRIBUTING.md>

Thanks for your attention!