Yocto Project 'AutoBuilder'
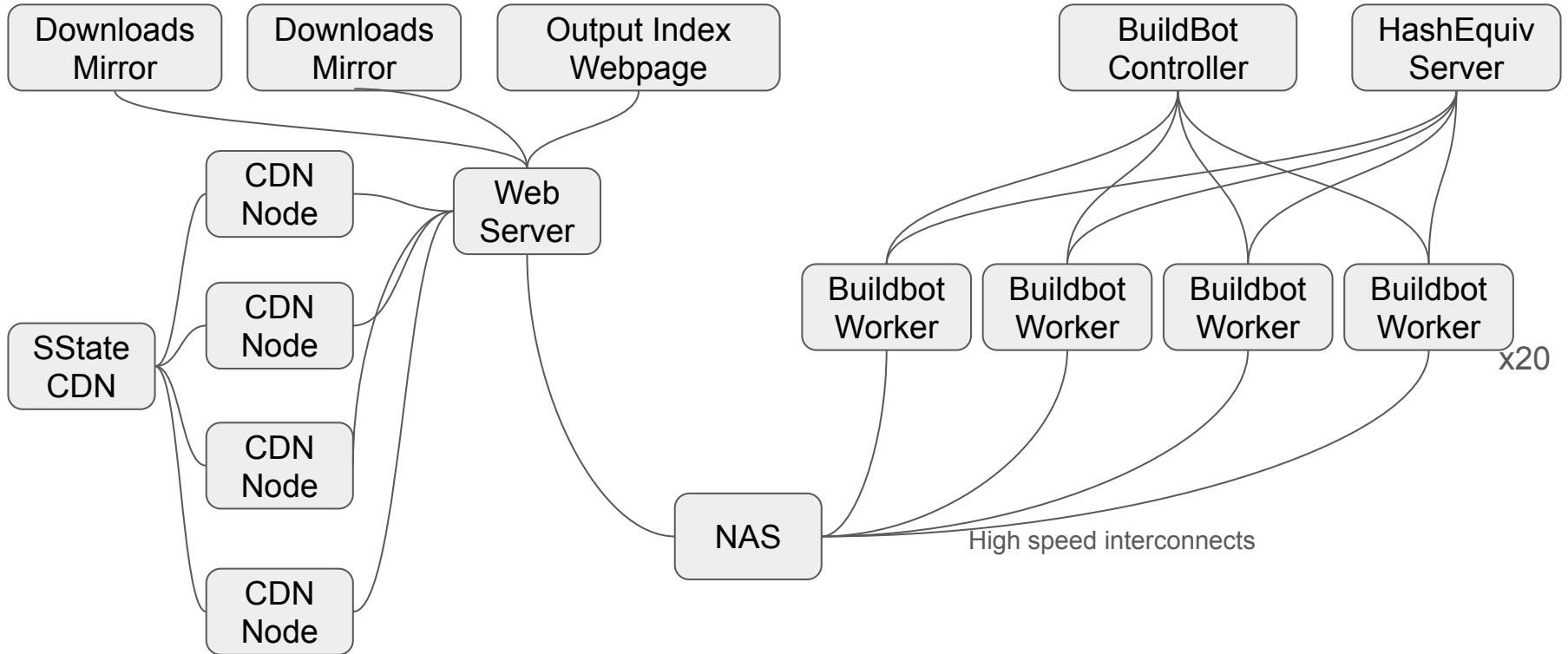
A lightning overview

Richard Purdie

Yocto Project 'AutoBuilder'

- Buildbot based (python), tests all patches before merging
- Builds/tests and generates all releases
- 20 x86 workers E5-2697 56 core Xeon 128GB memory, 2TB NVMe
- 3 aarch64 workers (eMAG to Altra)
- Each worker runs 3 build jobs in parallel
- NAS (20Gbps link) with dedicated network to workers (10Gbps each)
- SState and downloads on NAS over NFS
- Test results/output saved to NAS
- Result data also pushed to git repos (buildhistory, performance, test results)
- Shares and populates a hash equivalence server

AutoBuilder Block Diagram

AutoBuilder - Key URLs

Main Console: https://autobuilder.yoctoproject.org/typhoon/#/console

Output Index: https://autobuilder.yocto.io/pub/non-release/

Swatbot: https://swatbot.yoctoproject.org/

Buildhistory Test Results: https://git.yoctoproject.org/poky-buildhistory/

Test Results Archive: https://git.yoctoproject.org/yocto-testresults/

Performance Testing Results: https://git.yoctoproject.org/yocto-buildstats/

Documentation on many aspects in test manual: https://docs.yoctoproject.org/test-manual/index.html

AutoBuilder configuration - key repos

Controller code: https://git.yoctoproject.org/yocto-autobuilder2

Worker code: https://git.yoctoproject.org/yocto-autobuilder-helper

SwatBot code: https://git.yoctoproject.org/swatbot/

Key configuration file:
https://git.yoctoproject.org/yocto-autobuilder-helper/tree/config.json

AutoBuilder configuration

- SState+DL_DIR shared between all workers over NFS from NAS
- Official downloads mirror populated from server job copying DL_DIR to read-only share
- SState has files not accessed for X days removed
- NFS mounted with default options on all supported distros in 2024.
- We use option x-systemd.automount to allow easy NAS reboots
- Results/release output are saved to NAS directory
- SState and hashequivlance ideally have to match

Performance config tweaks (from config.json)

BB_NUMBER_THREADS = '16'
BB_NUMBER_PARSE_THREADS = '16'
PARALLEL_MAKE = '-j 16 -l 75'
BB_PRESSURE_MAX_CPU = '20000'
BB_PRESSURE_MAX_IO = '20000'
BB_LOADFACTOR_MAX = '1.5'
XZ_MEMLIMIT = '5%'
XZ_THREADS = '8'
ZSTD_THREADS = '8'
BB_TASK_NICE_LEVEL = '5'
BB_TASK_NICE_LEVEL:task-testimage = '0'
BB_TASK_IONICE_LEVEL = '2.7'
BB_TASK_IONICE_LEVEL:task-testimage = '2.1'
TEST_QEMUBOOT_TIMEOUT = '1500'
RUNQEMU_TMPFS_DIR = '/home/pokybuild/tmp'
BB_SERVER_TIMEOUT = '60'

AutoBuilder optimisations

- Local git repo caches to base clones off, only pull delta
- 'clobberdir' deferred deletion of build data with "ionice -c 3 rm"
- Jobs stop scheduling for workers low on disk space
- Results repos only shallow cloned
- Dedicated ext4 build filesystem with "tune2fs -o journal_data_writeback,^discard "
- Solid state drive is reformatted and a blocks discarded weekly to maintain optimal performance

Autobuilder test coverage

- Image, SDK and eSDK builds of all supported QEMU+HW Machines
- Runtime QEMU testing of images, SDKs and eSDKs (QEMU machines)
- poky and poky-altcfg - sysvinit, kernel versions, systemd
- Tests for reproducibility, wic, package managers, ptests, toolchain tests, musl, specific configs (no-x11, multilib, docs, much more), toaster
- oe-selftest and bitbake-selftest
- Layer Yocto Project Compatibility status checking
- Testing of member layers (member benefit)
- CVE and metric generation/testing
- Documentation building/publishing
- Runs build performance tests (on dedicated workers)

Autobuilder - The next generation

- No longer maintaining our own racks in a data centre
- Paying for dedicated servers at Hetzner (Helsinki)
- Able to have dedicated high speed network between NAS and workers
- Planned to have ~40 x86 workers and 6 aarch64 workers
- Tests successfully completed with small scale test setup
- Aim to have operational during August
- Also plan to have geo local hash equivalence servers