MPD Root Continuous Integration

Experiments hardware:
- Microsoft Azure VM
- 20 Cores Intel Xeon CPU E5-2673 v3 @ 2.40GHz
- 120GB RAM
- 1TB SSD

Open-source Docker containerization technology wraps up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries.

Runner creates separate Docker container for each CI task. This guarantees that every build is running in the same environment.

Main software source code repository server. Contains main software source code.

It also contains CI scripts and container image configuration for building and testing purposes. This allows to alter the configuration of a build's dependencies without the developer having to directly access GitLab CI Runner service.

GitLab server software also provides web user interface for easy project management and code conflicts solving.

Remote servers with GitLab Runner software installed. Each can run multiple jobs concurrently.

Realtime Logs

CI Task

CI Task

docker

CI Task

Runner

GitLab Runners

MPD Root

MPD Root GitLab Repository Server

CODE CHANGES

REPORT

Full build log being pushed from runner to repository. Source code commit status changes to "passed" or "failed". Other build artefacts (such as binaries) can be also transferred to repository.

In-container execution flow

Cacheable stages

Get Ubuntu image from Docker repository

Install Ubuntu

Pull FairSoft from its repository

Build FairSoft

Build MPD Root

Test MPD Root

Every build can pull latest version of external software or stick to specified release.

Docker repo

Fair Soft

External dependencies

The most time consuming task is FairSoft installation. It takes up to 40 minutes on described runner hardware.

The optimization was ensured by means of caching the project's dependencies and environmental components. Caching was done by Docker container manager.

As a result, we substantially decreased the build time: from 45 to 2-3 minutes.

Ubuntu and system libraries installation
Boost compilation
Pythia, HepMC and XercesC compilation
Geant4 installation start
Geant4 installation end
Root installation
MPD Root compilation and test

100%
CPU
0%

32MB/s
NET
0MB/s

10GB
MEM
0GB

0min   45min