Item 294
git_comments:

git_commits:

1. **summary:** [MXNET-327] Fix docker disk-space leak by rearranging ubuntu_build_cuda #10581
   message: [MXNET-327] Fix docker disk-space leak by rearranging ubuntu_build_cuda #10581

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [MXNET-327] Fix docker disk-space leak by rearranging ubuntu_build_cuda
   body: ## Description ## This PR fixes the docker disk-space leak which was caused by our CI constantly re-creating docker layers and invalidating the cache. The reason for this was that the ```nvidia-install``` script was executed after the uncacheable ```add_user``` script. This result in every single run creating a new layer. While this is usally no problem since the following layers are very small and compressed into a single layer, this caused problems if a big layer like ```Nvidia-install``` is being re-created during every single run. Also, this speeds up the spin-up time of GPU build jobs. Considering the fact that the GPU build job is on our critical path, this will reduce the overall runtime of a CI job by about 5 minutes. ## Checklist ## ### Essentials ### Please feel free to remove inapplicable items for your PR. - [x] The PR title starts with [MXNET-$JIRA_ID], where $JIRA_ID refers to the relevant [JIRA issue](https://issues.apache.org/jira/projects/MXNET/issues) created (except PRs with tiny changes) - [x] Changes are complete (i.e. I finished coding on this PR) - [x] All changes have test coverage: - Unit tests are added for small changes to verify correctness (e.g. adding a new operator) - Nightly tests are added for complicated/long-running ones (e.g. changing distributed kvstore) - Build tests will be added for build configuration changes (e.g. adding a new build option with NCCL) - [x] Code is well-documented: - For user-facing API changes, API doc string has been updated. - For new C++ functions in header files, their functionalities and arguments are documented. - For new examples, README.md is added to explain the what the example does, the source of the dataset, expected performance on test set and reference to the original paper if applicable - Check the API doc at http://mxnet-ci-doc.s3-accelerate.dualstack.amazonaws.com/PR-$PR_ID/$BUILD_ID/index.html - [x] To the my best knowledge, examples are either not affected by this change, or have been fixed to be compatible with this change ### Changes ### - Reduced disk-space consumption on Ubuntu CPU slaves - Higher execution speed of all jobs that build with CUDA.

github_pulls_comments:

1. What do you mean by "creating a new layer"?
2. Everything up to (exclusively) ```RUN /work/ubuntu_adduser.sh``` is cached and everything afterwards is not cached. Usually, this is no problem since docker is able to fuse multiple layers together into a temporary one which is not being put into the docker cache. The problem here is that ```RUN /work/ubuntu_nvidia.sh``` creates a new unfusable permanent layer, which results in always creating a new docker cache entry and thus filling up the disk.

github_pulls_reviews:

jira_issues:

1. **summary:** Ubuntu CPU slaves run out of disk space
   description: At the moment, our Ubuntu CPU slaves run out of disk space every two weeks. This is caused by Docker constantly re-creating uncachable layers of Dockerfile.build.ubuntu_build_cuda.

jira_issues_comments: