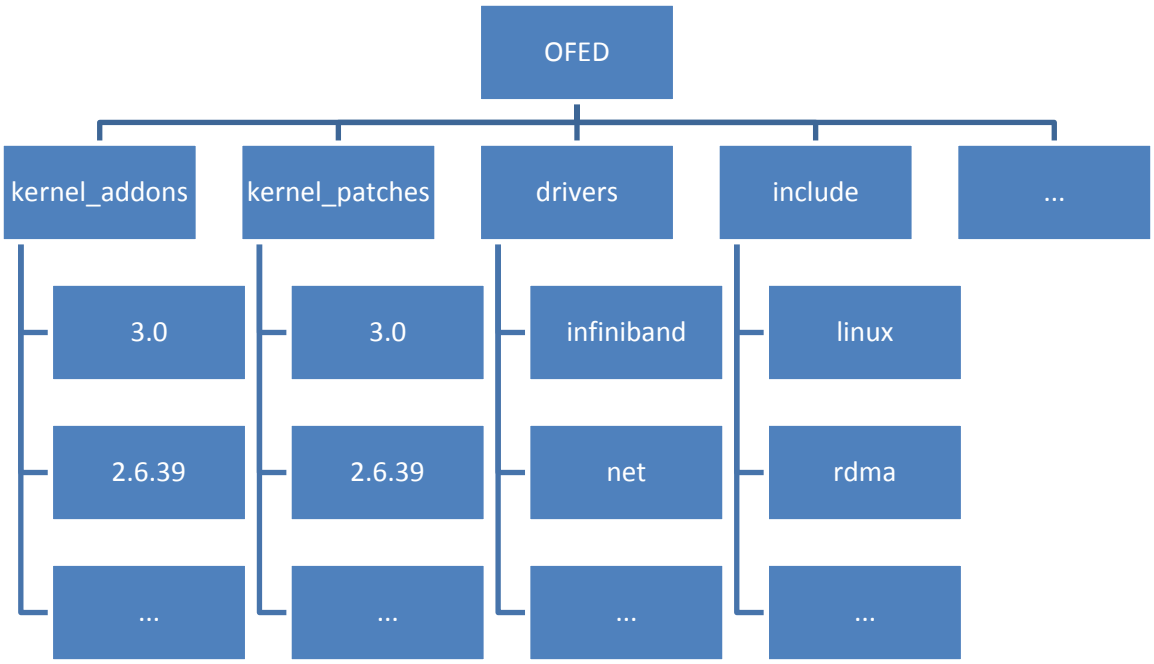
Proposal for Reorganization of the OFED Source Code Repository

At the Dec 13 EWG/OFED meeting there was consensus to use a new approach to base OFED on upstream kernel code and backports. This seems like a good time to re-organize the OFED source code repository to better fit within the kernel development environment and various Linux distributions. This document proposes a way to re-organize the source code repository in order to improve quality, maintainability, and speed of development.

Current Structure and Process

Structure

Currently, the OFED source repository contains one main branch, which in turn contains all files necessary to setup and patch the OFED source for any of the supported kernel versions. The tree is organized in the following manner:



The *"kernel_addons"* directory contains mostly header files which are needed for the individual backport branches. These header files are divided into their own directory named after the backport branch. The content of this directory is common for all backports – in other words, all header files for all backports are present regardless of which backport is being built.

The appropriate include path is given by a complier flags which lives in *"config.mk"*.

The *"kernel_patches"* directory contains all patch files (again, separated by backport) to be applied to the base OFED code in order to bring the base OFED code to the appropriate backport level. This is done by the "ofed_patch.sh" script before doing any work or builds with the tree.

Process

The current development process is largely based on the structure of the OFED tree. Each commit in the OFED tree does not contain changes to the actual source code but rather adds, deletes, or modifies one of more of the patch files stored in the *"kernel_patches"* directory. In addition, commits can add, delete, or modify header files in the *"kernel_addons"* directory.

An OFED developer goes roughly through the following process to make a change to the source tree:

- Clone the OFED tree
- Run *ofed_checkout.sh* to checkout all necessary files and directories
- Run *ofed_patch.sh* to apply all necessary backport patches
- Make the changes that he/she wants to make to the source code
- Generate a diff of the changes made above into a patch file(s)
- Add the patch file(s) to the appropriate backport directories
- Commit the new patch files

The process of modifying a change made by a previous patch file (effectively, modifying the patch file itself and generating a patch to a patch) seem to be even more complex.

Drawbacks of Current Process

The organizational structure and process described above is rather error-prone. This is evident by the complexity of the development process, complexity of maintenance of the tree, and most of all the fact that "broken" patches (patches which do not apply cleanly or break the code/build) have been committed to the repo on multiple occasions.

Developers do not have a good way to verify the quality of their patch submissions and, on occasion, broken patches have remained unfixed for several days.

Another noticeable drawback of the current methodology is that its main vehicles for change are patch files and not actual code changes. This makes following the changes in the repository much harder for the various development groups.

Proposed OFED Tree Organization

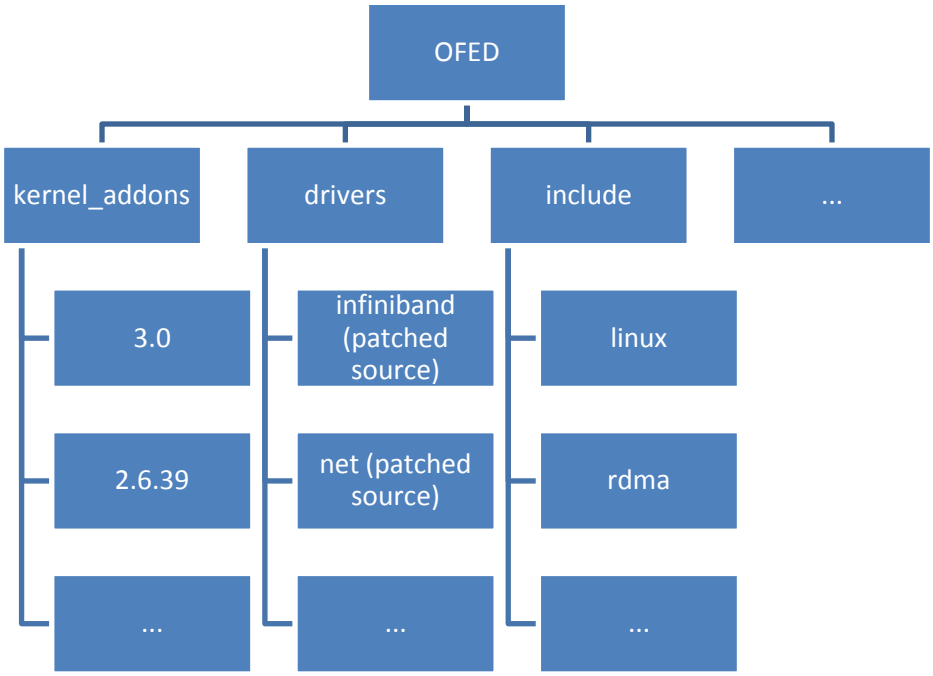
The goal of the proposed OFED source code organization is to limit the amount of maintenance work needed, make development in the source tree much easier, and eliminate some of the problems related to the broken patches and patches which only apply on a certain set of backport branches.

Structure

The structure of the tree would change to make extensive use of Git branches. In this new structure, each supported backport will have its own Git branch which will be a “flattened” version of the OFED tree. Effectively, each Git branch will contain the checked out and patched OFED source.

The master branch will contain only files needed for the building, packaging, and maintenance of the source and branches.

With this new structure scheme, the *“kernel_patches”* directory disappears as there is no need for it anymore. The individual backport subdirectories in *“kernel_addons”* could also be removed and only the appropriate subdirectory be present in the individual backport branches.



Development Method

Once restructured, the source tree should greatly improve ease of development. Due to the fact that each branch contains a “flattened” version of the OFED source, developers will go through the following process:

- Clone the OFED tree
- Checkout the appropriate backport branch
- Make the changes that the developer want to make to the code directly
- Commit the changes to the branch
- If the changes are needed in another backport branch, merge (resolving any conflicts) the two backport branches (scripts could be created to make this process easier)

The advantages in rapid development that this new structure and method allows are obvious. The code is already ready for development when the repository is cloned. The new organization also helps in ensuring that no broken patches are committed as there will be no patch files anymore, the repository commits will contain actual code changes.

Since each individual backport branch will be buildable (using Makefiles and scripts specific to the backport, as well as the general ones from the master branch), it would be easier for individual developers to build and perform unit tests on their changes, improving quality of commits.