

Dear reviewers,

Thank you for your careful reading of the present work regarding XRootD. Upon reading your comments and then re-evaluating the structure of the current work (especially in terms of Introduction and Conclusions), we realized that a clearer and more concise description is needed for the objective of this paper, together with a strong motivation on why such a practical part was chosen. As a result, we modified the following:

1. Erasure Coding (EC) is introduced firstly in Section I, after a discussion on why data reliability is a crucial aspect in the storage facilities at CERN. (i.e., in the paragraph starting with *"In terms of its functionality..."*).
2. EC is therefore considered a great tool to have implemented into the XRootD framework. In the paragraph starting with *"Implementing an Erasure Coding..."* we mention that this is the aim of the practical part for the current work. Also here, we mention the fact that there is a standard approach in developing the plug-in, or a "better" alternative, by using the newly adopted Declarative API (an asynchronous implementation of the XRootD client).
3. In Section 4-B, we try to emphasize why the Declarative API is a great tool in developing EC plug-in, by firstly explaining that if one uses the standard C++ asynchronous API (pre-existing in older versions of the XRootD client), it will result in a much more complex codebase, with an unclear logic (in terms of operation workflow), and a lot of code-boilerplate. (Starting with the paragraph *"In order to understand why the Declarative API..."*)
4. We use Listing 9 as a codebase for the existing asynchronous API, to show that the entire operation workflow will be hidden right after the first operation, that is Open(). Explanations for the handler logic and parallel operations is also done throughout the following paragraphs. It is mentioned that for complexity reasons, we only show the code of the first operation, since the entire workflow would require a large number of listings which would be out of the scope of this work.
5. Finally, with Listing 10, we show the version of EC implementation using the Declarative API, where it should be noted that the entire workflow (including the parallel execution of the operations) is depicted.
6. In section 5 – Conclusions we adjusted the text such that importance of EC and the Declarative API are both put in light to the reader.

Taking all these points into consideration, we really believe that the current adjustments improve our work in such a way that all the requirements pointed out in your review are met.

Kind regards,  
The authors