Summary: The Impact of Taskyield on the Design of Tasks Communicating Through MPI

`taskyield`: The taskyield construct specifies that the current task can be suspended in favor of execution of a different task. The benefit of this is the ability to hide latencies that may occur during the execution of a task.

Tasks allow OpenMP programs to be structure in such a way that one task can handle the MPI communication while other tasks handle data parallelism.

The paper can be summarized in three main components.

1. Explanation of different `taskyield` paradigms and how they can be used within MPI ranks to facilitate communicaiton. Also a explains how the different paradigmns can affect CORRECTNESS of a program!
2. Explain a series of blackbox tests that the researchers did to determine the `taskyield` paradigm being used by different implementations of OpenMP.
3. Performance Benchmarks of the Cholesky factorization on matrices using different communication/taskyield protocols. Results show that a correct hybridized implementation of OpenMP and MPI using OpenMP tasks and `taskyield`'s can have significant boosts in performance. However, being able to properly optimize the tasks and calls depends on a deep knowledge of the implementation. Furthermore the authors expose an issue and potential fix: without knowing the OpenMP implemenation being used (and by extension, the `taskyield` implementation) a program cannot know how to best handle the communication tasks. Certain implementations will even result in INCORRECT programs which deadlock due to the nature of the `taskyield` implementation. Further optimizations can be made depending on the result of the task yield implementation. The resulting fix for this problem is to expose some method in OpenMP that allows users to query the `taskyield` policy enable better optimizations.