

The **Hyperquicksort** algorithm tries to improve the load balancing among processes by having the P processes initially sorting their own chunks in $O(n/P \log n/P)$ time. The process that selects the pivot then chooses the median of its chunk and broadcasts it to all the other processes. The algorithm then proceeds in the same way as the **Simple Parallel Quicksort** described above.

Both the MPI and OpenMP implementations of this algorithm follow the same steps of the previous one with the only difference being the addition of the initial sorting of the chunks. Under the assumptions of uniform distribution of the elements in the processes' chunks and communication times dominated by transmission times (i.e. low latency), the overall complexity of the algorithm is $\tilde{O}(n/P \log(n + P))$. Overall, this algorithm provides a better alternative for load balancing but still relies on the assumption of finding uniformly distributed values for the partitions at each stage of the algorithm, which is not always guaranteed.