

Quicksort is a highly efficient sorting algorithm that follows a divide-and-conquer approach in order to recursively sort large arrays of elements according to an order relation.

This report wants to explore the possibility of parallelizing this well known algorithm proposing different implementations both in distributed and shared memory paradigms. The objective is to compare both complexity and performance of these algorithm in order to understand the trade-offs and advantages of each of them.

With the purpose of avoiding ambiguity, the different parallel versions are labelled as follows.

- **Task Quicksort:** a shared memory parallel quicksort algorithm that takes advantage of OpenMP's `#pragma omp task` directive.
- **Simple Parallel Quicksort:** basic parallelization of the quicksort algorithm in which the initial array is split among processes.
- **Hyperquicksort:** improved version of the parallel quicksort where pivot selection is optimized to improve load balancing.
- **Parallel Sort by Regular Sampling (PSRS):** regular sampling is used in this parallel quicksort to optimize load balancing and minimize communications.

In the following sections, each of these algorithm will be first described from a theoretical point of view and the core ideas behind the practical implementation in distributed memory will be presented in the form of pseudocodes<sup>1</sup>. A proposed shared memory counterpart will also be then presented for each of the algorithms.

Performance analysis and scalability results will be then presented and discussed in the final sections of this report, together with some final considerations.

All the implementations have been written in C and have been tested on the *ORFEO* cluster at the AREA Science Park in Trieste [?]. With the intention of maintaining clarity and conciseness in this report, detailed code implementations have been omitted, but are available in the author's GitHub repository [?].

---

<sup>1</sup>With the exception of the serial version and the **Task Quicksort** algorithm, only implemented in OpenMP.