At first, the program randomly generates N/p of non-negative numbers in each processor, where N is the total of numbers, and p is the number of processors. The parallelization of the quicksort algorithm begins by randomly selecting a number within each processor. The selected numbers for each processor are gathered in the root processor; then, the root processor selects the median of these randomly selected numbers. The median is then used as a pivot, which is broadcast to all processors. Each processor, denoted as P_i , then partitions its own set of numbers into two subarrays: S_i for elements less than or equal to the pivot and L_i for elements greater than the pivot.

After partitioning, the root processor gathers the size of each subarray from each processor and calculates the next root based on the total number of S and L, where $S = \bigcup_i S_i$ and $L = \bigcup_i L_i$. The next root is the second root when dividing the processors into two groups. The root processor divides the available processors into two groups, one group to handle S, and another group to handle L . Then the root processor calculate how many elements should each processor holds by calculating $P_S / |S|$ for the first group of processors, and $P_L / |L|$ for the second group. Then, the root processor determines which subarray is sent to which processor using the gathered sizes of each subarray, then form a matrix for the all-to-all personalized communication later. In the matrix, the row vector, row_i is a receive count vector of processor i, and the j-th element in the row vector is number of elements to be received from the processor j; whereas the column vector col_i is a send count vector of processor i, and the j-th element in the column vector is number of elements to be sent to processor j. The row and column vector are then sent to the corresponding processors. Each processor computes the send and receive displayments vector using the send and receive count vector by prefix sum. Finally, the processors perform the all-to-all personalized communication such that all the first group of processors hold S whereas the other group of processors hold L.

The processors are divided into two groups, with each processor assigned a color, determined by the next root. The communicator is split into two. Then the processors repeat recursively until there is only one processor within the group. A serial quick sort algorithm is performed within the single processor.

To ensure each processor has N/p numbers at the end, the root processor after the above process, collects the current size of number holds in each processor. Then, it determines which element is sent to which processor by using a similar way from the above all-to-all personalized communication, which forms the matrix, sends the send and receive counts vector, forms the send and receive displayments vector, performs the all to all personalized communication.

	N = 1,000,000	N = 10,000,000	N = 100,000,000
P = 1	0.8512s	9.7612s	110.4674s
P = 2	0.4975s	5.3487s	80.5708s
P = 4	0.3502s	4.3638s	56.0480s
P = 8	0.2821s	2.6306s	36.4166s
P = 16	0.1654s	1.1515s	13.6363s