Hpc 2 parallel bubble sort

```
#include <iostream>
#include <omp.h>

using namespace std;

void bubbleSort(int *a, int n) {
   int swapped;
   for (int i = 0; i < n; i++) {
       swapped = 0;
       #pragma omp parallel for shared(a)
       for (int j = 0; j < n - 1; j++) {
           if (a[j] > a[j + 1]) {
               swap(a[j], a[j + 1]);
               swapped = 1;
           }
       }
       if (!swapped)
           break;
   }
}

int main() {
   int *a, n;
   cout << "Enter total number of elements: ";
   cin >> n;
   a = new int[n];
   cout << "Enter elements: ";
   for (int i = 0; i < n; i++) {
       cin >> a[i];
   }

   double start_time = omp_get_wtime();
   bubbleSort(a, n);
   double end_time = omp_get_wtime();

   cout << "Sorted array: ";
   for (int i = 0; i < n; i++) {
       cout << a[i] << " ";
   }
   cout << "\nTime taken: " << end_time - start_time << " seconds" << endl;

   delete[] a;
   return 0;
}
```