

Experimental Results

We have compared all the results with the respective serial algorithms. We have tested all the implementations on two machines. Description of machines is as follows :

1. Machine 1

- **Processor Name** : Intel(R) Core(TM) i7-4770
- **No of Cores** : 4
- **Threads per core** : 2
- **Cache**
 - L1 = 32KB
 - L2 = 256 KB
 - L3 = 8 MB

2. Machine 2

- **Processor Name** : Intel(R) Xeon(R) E5-2670
- **No of Cores** : 16
- **Threads per core** : 2
- **Cache**
 - L1 = 32KB
 - L2 = 256 KB
 - L3 = 20 MB

1 Breadth First Search (BFS)

We have have tested this algorithm on 4 graph datasets. For large graph cilk plus implementation shows the speedup over the serial implementation, but for smaller graph cilk plus implementation takes more time than serial implementation. In smaller graph, **Cilk Plus Reducer List** overhead is more than the parallel work. Reducer List takes more time if number of processors are less.

We have tested results on 4 dataset. Description of datasets are as follows.

1.1 List of Graph Dataset

1. rmat22

- Number of nodes = 4194304
- Number of edges = 33226138

2. rmat20

- Number of nodes = 1048576
- Number of edges = 8259994

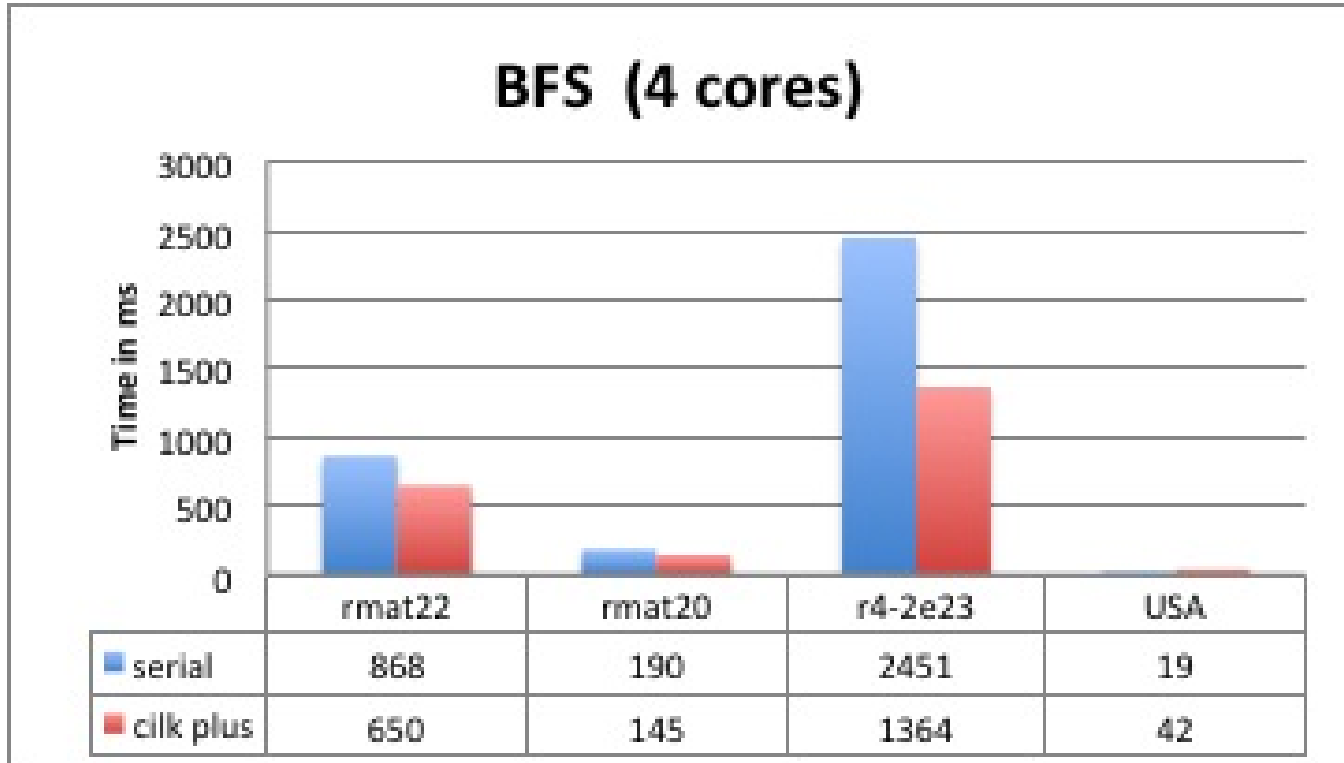


Figure 1: BFS results on 4 cores machine

3. r4-2e23

- Number of nodes = 8388608
- Number of edges = 33554432

4. USA

- Number of nodes = 264346
- Number of edges = 730100

1.2 Analysis

Figure 1 and figure 2 shows results on 4 core machine and 16 core machine respectively. Graph 1 and 2 are sparse graph. It shows 1.33x and 1.31x speedup on 4 cores, 5.65x and 4.16x on 16 cores machine for these graphs respectively. Graph 3 is dense graph, so it has more scope for parallelism. Cilk Plus implementation shows 1.8x speedup on 4 cores machine and 5.6x speedup on 16 cores machine. Graph 4 is a smaller graph. Because of reducer list overhead, cilk plus implementation performs slower than the serial implementation.

2 Single Source Shortest Path (SSSP)

We have tested this algorithm on 4 graph datasets as mentioned above. For large graph cilk plus implementation shows the speedup over the serial implementation, but for smaller graphs cilk plus implementation takes more time than serial implementation. In smaller graph, **Cilk Plus Reducer List** overhead is more than the parallel work. Reducer List takes more time if number of processors are less.

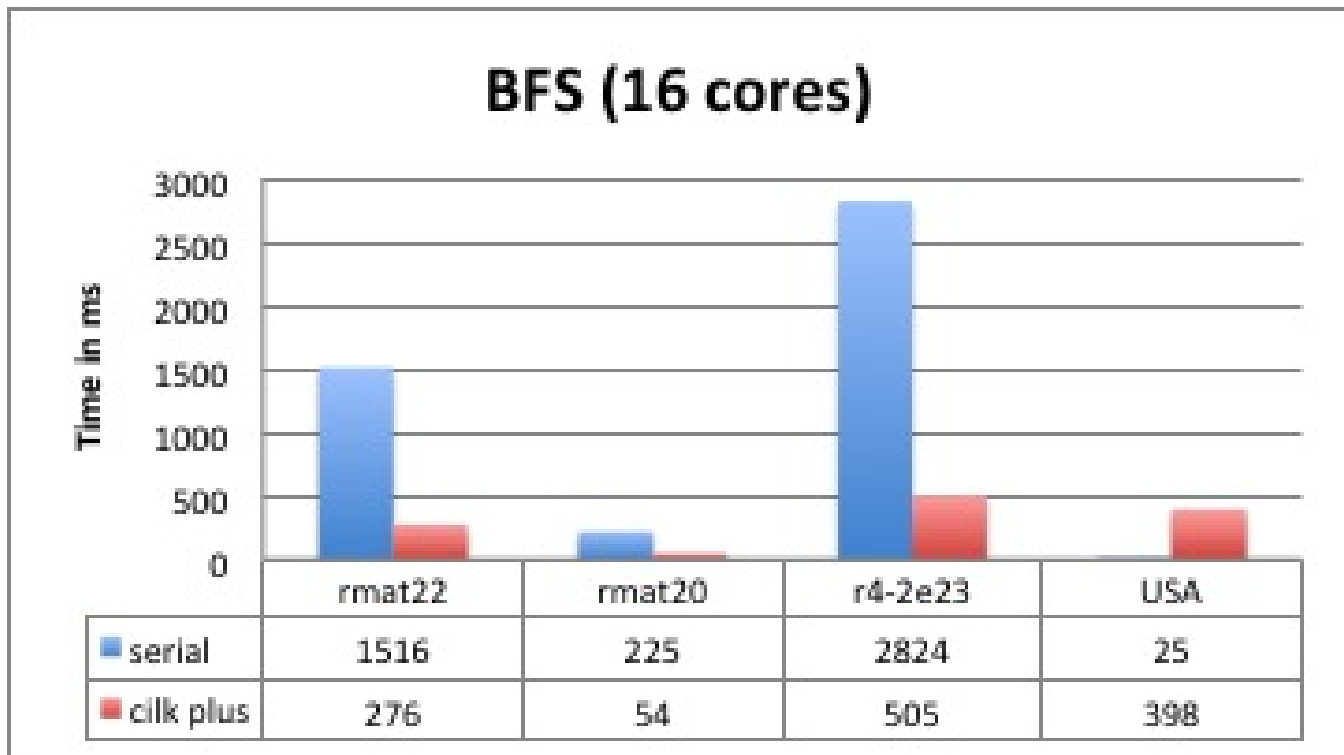


Figure 2: BFS results on 16 cores machine

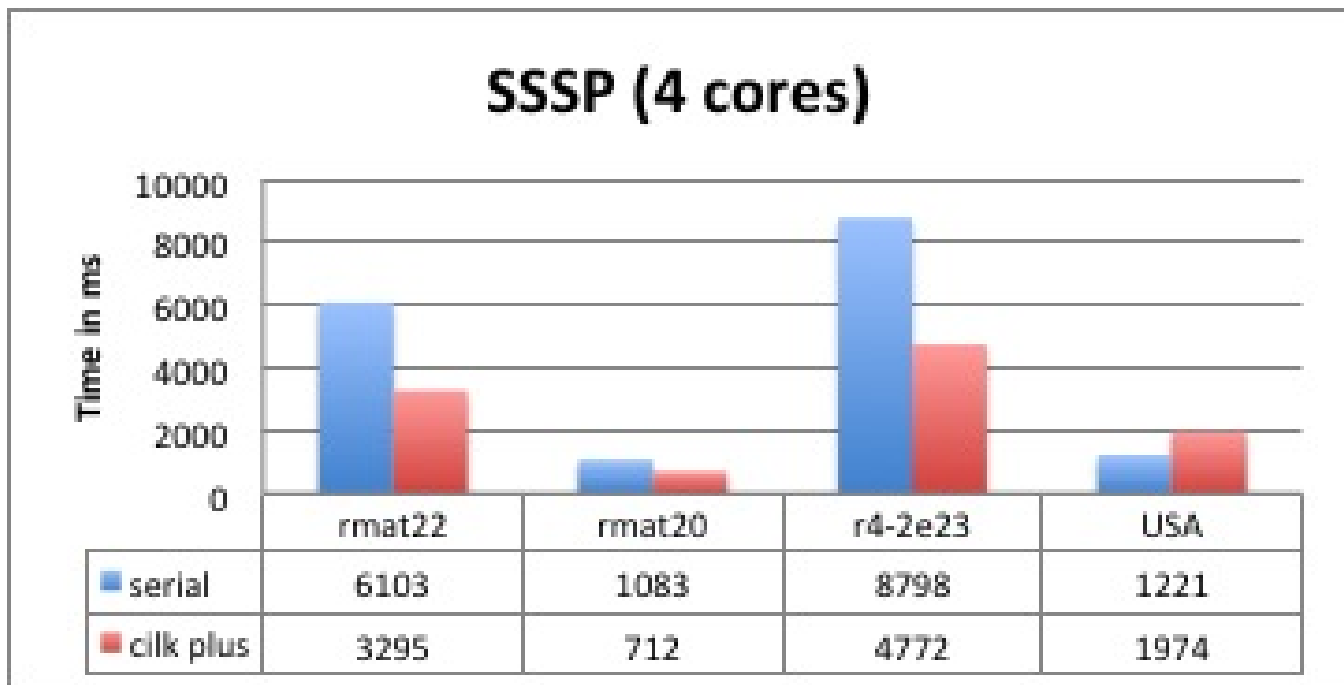


Figure 3: SSSP results on 4 cores machine

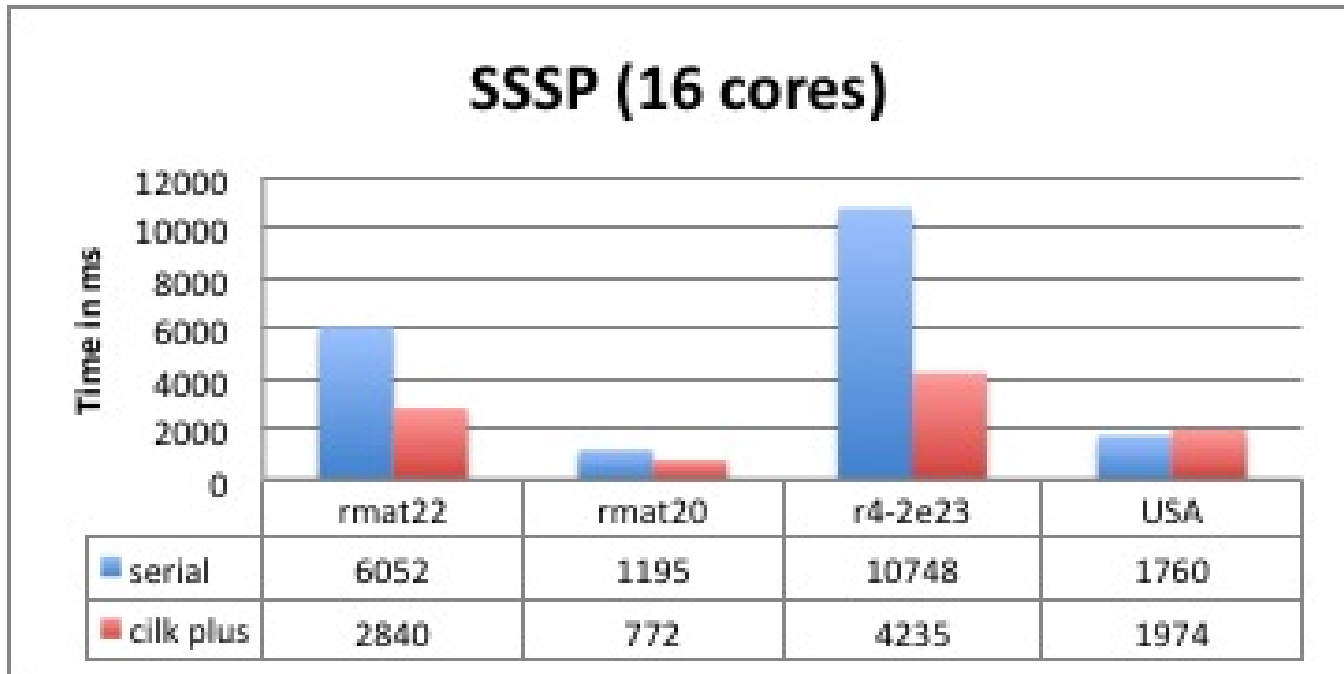


Figure 4: SSSP results on 16 cores machine

2.1 Analysis

Figure 3 and figure 4 shows results on 4 core machine and 16 core machine respectively. Graph 1 and 2 are sparse graph. It shows 1.8x and 1.5x speedup on 4 cores, 2.13x and 1.54x on 16 cores machine for these graphs respectively. Graph 3 is dense graph, so it has more scope for parallelism. Cilk Plus implementation shows 1.8x speedup on 4 cores machine and 2.5x speedup on 16 cores machine. Graph 4 is a smaller graph. Because of reducer list overhead, cilk plus implementation performs slower than the serial implementation.