ParallelSum.Java

```
import java.util.Arrays;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.TimeUnit;

public class ParallelSum {
    private static final int NUM_THREADS = Runtime.getRuntime().availableProcessors();

    public static void main(String[] args) {
        // Sample array
        int[] array = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

        // Divide the array into equal parts
        int partitionSize = array.length / NUM_THREADS;
        int[][] partitions = new int[NUM_THREADS][];
        for (int i = 0; i < NUM_THREADS; i++) {
            if (i == NUM_THREADS - 1) {
                partitions[i] = Arrays.copyOfRange(array, i * partitionSize, array.length);
            } else {
                partitions[i] = Arrays.copyOfRange(array, i * partitionSize, (i + 1) * partitionSize);
            }
        }

        // Create a thread pool
        ExecutorService executor = Executors.newFixedThreadPool(NUM_THREADS);

        // Submit tasks for each partition
        SumTask[] tasks = new SumTask[NUM_THREADS];
        for (int i = 0; i < NUM_THREADS; i++) {
            tasks[i] = new SumTask(partitions[i]);
            executor.execute(tasks[i]);
        }

        // Shutdown the executor and wait for all tasks to complete
        executor.shutdown();
        try {
            executor.awaitTermination(Long.MAX_VALUE, TimeUnit.NANOSECONDS);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        // Sum the partial results
        int sum = 0;
        for (SumTask task : tasks) {
            sum += task.getResult();
        }

        System.out.println("Sum: " + sum);
```

```
    }

    static class SumTask implements Runnable {
        private final int[] array;
        private int result;

        public SumTask(int[] array) {
            this.array = array;
        }

        public int getResult() {
            return result;
        }

        @Override
        public void run() {
            int sum = 0;
            for (int num : array) {
                sum += num;
            }
            result = sum;
        }
    }
}
```