Implement min, max, sum and avg operations using parallel reduction

```
import multiprocessing as mp

def parallel_reduction(array, op):
    num_processes = mp.cpu_count()
    chunk_size = len(array) // num_processes
    pool = mp.Pool(processes=num_processes)

    chunks = [array[i:i+chunk_size] for i in range(0, len(array), chunk_size)]
    results = pool.map(op, chunks)

    pool.close()
    pool.join()

    return op(results)

def reduce_sum(array):
    return sum(array)

def reduce_min(array):
    return min(array)

def reduce_max(array):
    return max(array)

def reduce_avg(array):
    return sum(array)/len(array)

if __name__ == '__main__':
    array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

    print(parallel_reduction(array, reduce_sum)) # 55
    print(parallel_reduction(array, reduce_min)) # 1
    print(parallel_reduction(array, reduce_max)) # 10
    print(parallel_reduction(array, reduce_avg)) # 5.5
```

```
55
1
10
5.5
```