```python
def partition(data, pivot):
    less = [x for x in data if x < pivot]
    equal = [x for x in data if x == pivot]
    greater = [x for x in data if x > pivot]
    return [less, equal, greater]

def merge(sorted_parts):
    return sum(sorted_parts, [])

if __name__ == '__main__':
    data_sizes = [100000, 500000, 1000000, 5000000, 10000000, 50000000, 100000000]
    times_parallel = []

    for size in data_sizes:
        data = np.random.randint(1, 1000000, size)
        start_time = time.time()
        sorted_data = parallel_sort(data)
        end_time = time.time()
        times_parallel.append(end_time - start_time)

    plt.plot(data_sizes, times_parallel, marker='o')
    plt.xlabel('Data Size')
    plt.ylabel('Time (s)')
    plt.title('Parallel Quicksort Performance')
    plt.grid(True)
    plt.show()
```

✓ 4m 21.3s