Текст программы:

```cpp
#include <vector>
#include <iostream>
#include <future>
#include <numeric>

using namespace std;

double find_max(double* beg, double* end) {
    return *max_element(beg, end);
}

double comp2(vector<double>& v)
{
    using Task_type = double(double*,double*);

    packaged_task<Task_type> ptO {find_max};
    packaged_task<Task_type> pt1 {find_max};

    future<double> f0 {ptO.get_future()};
    future<double> f1 {pt1.get_future()};

    double* first = &v[0];

    thread t1 {std::move(ptO), first, first+v.size()/2};

    thread t2{std::move(pt1), first + v.size() / 2, first + v.size()};

    t1.join();
    t2.join();

    return max(f0.get(), f1.get());
}

double find_max_linear(const vector<double>& vec) {
    return *max_element(vec.begin(), vec.end());
}

vector<double> generate_vector(int n) {
    vector<double> res;
    for (int i = 0; i < n; ++i) {
        res.push_back(rand() % 25);
    }
    return res;
}

int main() {
    int n = 100000000;
    vector<double> vec = generate_vector(n);

    // packaged_task
    cout << "Packaged_task program:" << endl;
    auto start = chrono::system_clock::now();
    double res = comp2(vec);
    auto stop = chrono::system_clock::now();
    cout << "Result: " << res << endl;
    cout << "Time elapsed: " << chrono::duration<double>(stop-start).count() << endl;

    // linear
```

```
    cout << "\nLinear program:" << endl;
    auto start1 = chrono::system_clock::now();
    cout << "Result: " << find_max_linear(vec) << endl;
    auto stop1 = chrono::system_clock::now();
    cout << "Time elapsed: " << chrono::duration<double>(stop1-start1).count() << endl;


    return 0;
}
```

Результаты экспериментов:

Размерность задачи	Последовательная програма	Параллельная программа		
		Время выполнения	Ускорение	Эффективность
10000	0,000227923	0,000196815	1,158057059	0,5790285293
1000000	0,0154988	0,0021571	7,185016921	3,59250846
100000000	1,63252	0,262562	6,217655259	3,108827629