void f\_if(std::vector<int> &v)

{

const int rep = 100;

std::vector<int> bitset(v.size());

std::chrono::time\_point<std::chrono::system\_clock> start, end;

int elapsed\_mseconds = 0;

for (int i = 0; i < rep; i++)

{

start = std::chrono::system\_clock::now();

#pragma novector

for (int j = 1; j < v.size(); j++)

if (v[j] == 0)

bitset[j] = 1;

end = std::chrono::system\_clock::now();

elapsed\_mseconds += std::chrono::duration\_cast<std::chrono::milliseconds>

(end - start).count();

for (int j = 0; j < bitset.size(); j++)

bitset[j] = 0;

}

std::cout << "Time: " << elapsed\_mseconds << " ms\n";

}

void f\_eq(std::vector<int> &v)

{

const int rep = 100;

std::vector<int> bitset(v.size());

std::chrono::time\_point<std::chrono::system\_clock> start, end;

int elapsed\_mseconds = 0;

for (int i = 0; i < rep; i++)

{

start = std::chrono::system\_clock::now();

#pragma novector

for (int j = 1; j < v.size(); j++)

bitset[j] = (int)(v[j] == 0);

end = std::chrono::system\_clock::now();

elapsed\_mseconds += std::chrono::duration\_cast<std::chrono::milliseconds>

(end - start).count();

for (int j = 0; j < bitset.size(); j++)

bitset[j] = 0;

}

std::cout << "Time: " << elapsed\_mseconds << " ms\n";

}

int main()

{

const int size = 1000 \* 1000 \* 10;

const double p = 0.5;

std::vector<int> v(size);

std::iota(&v[0], &v[size \* p], 1);

std::random\_device rd;

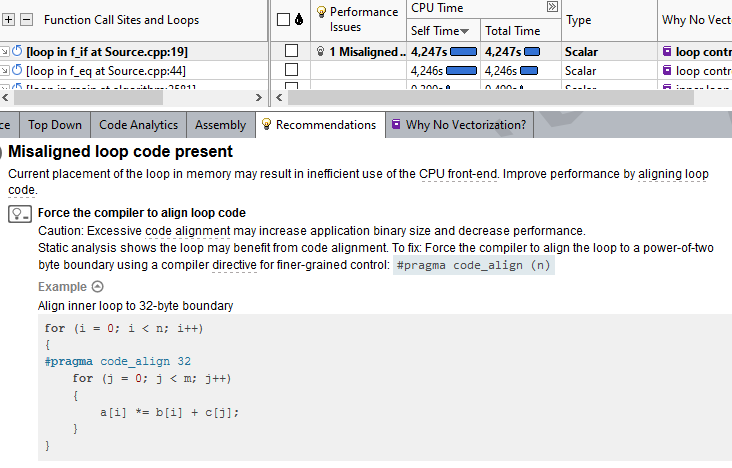
std::mt19937 g(rd());

std::shuffle(v.begin(), v.end(), g);

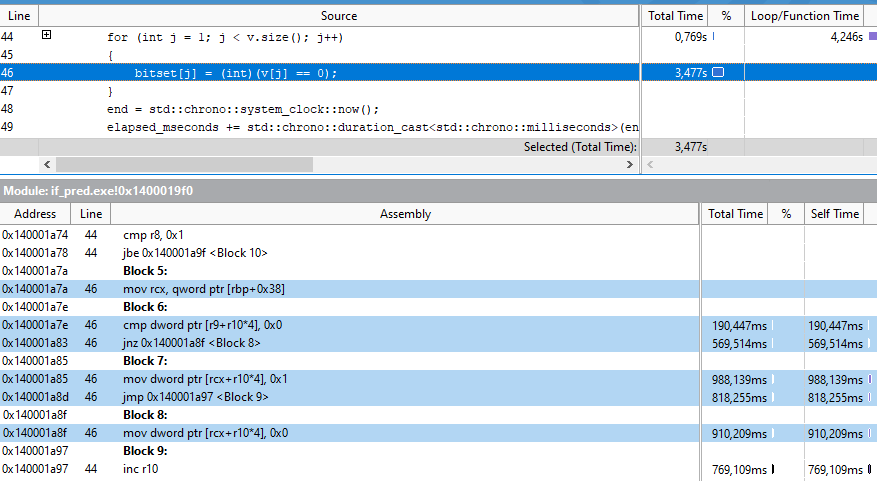
f\_if(v);

f\_eq(v);

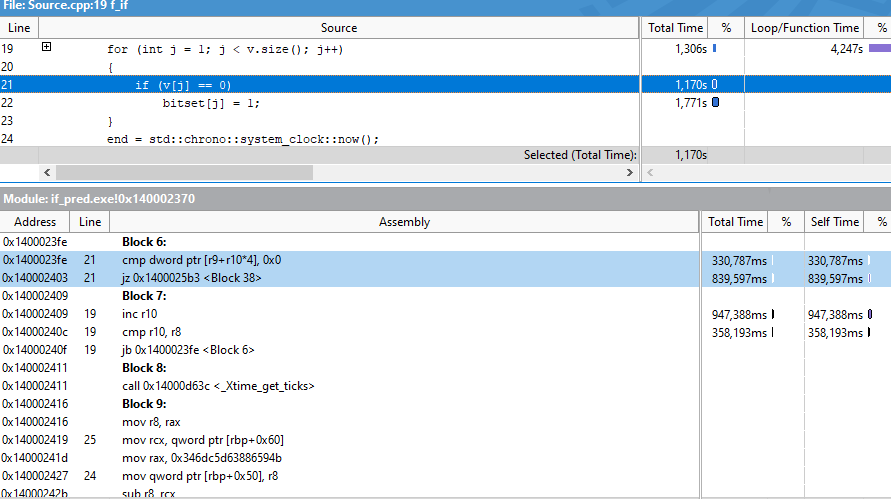
}

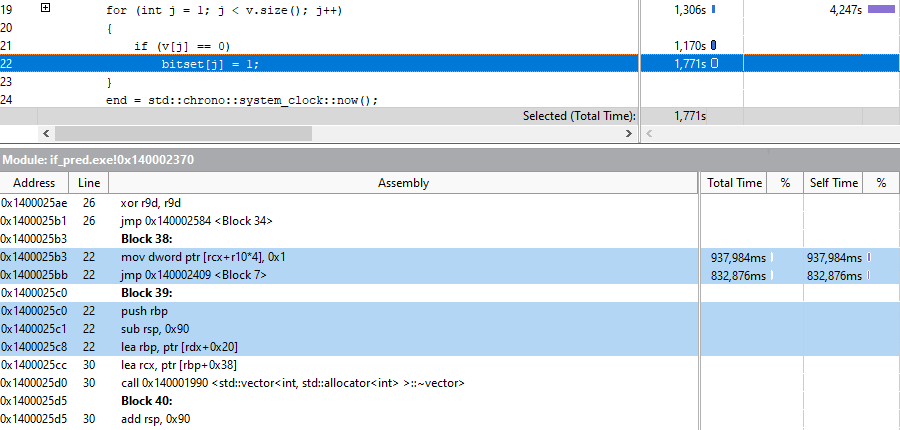


**f\_eq**



**f\_if**





|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Time\_if | Time\_eq | Time\_if | Time\_eq | Time\_if | Time\_eq |
| Доля p | 0.5 | 0.5 | 0.1 | 0.1 | 0.9 | 0.9 |
| Время | 4.2 | 4.2 | 1.7 | 1.8 | 1.5 | 1.5 |