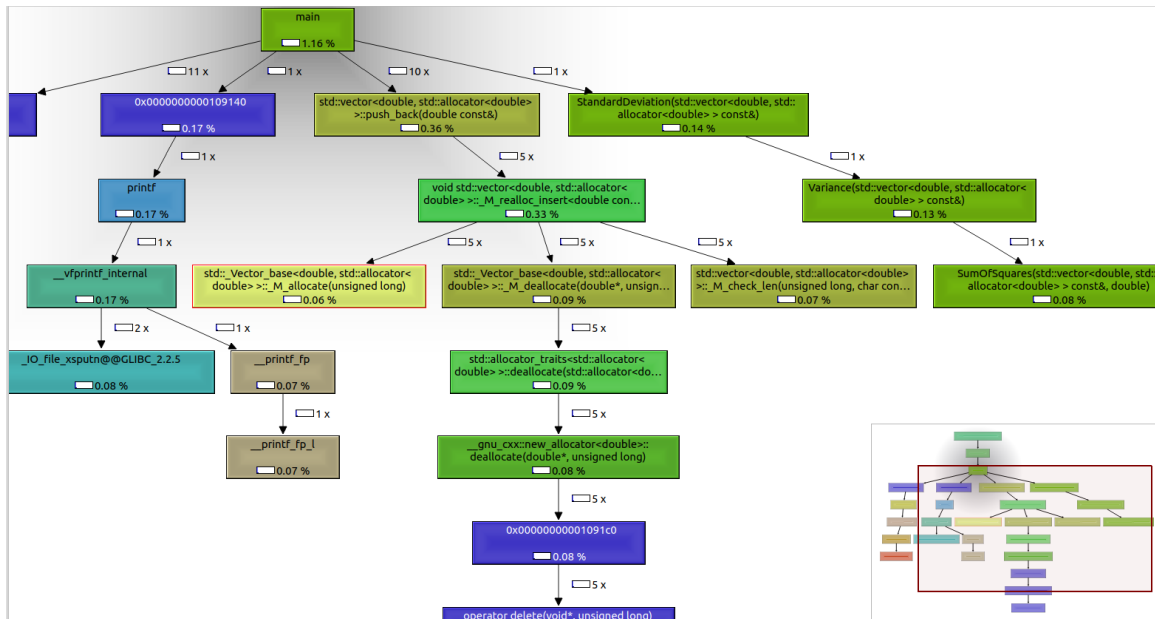
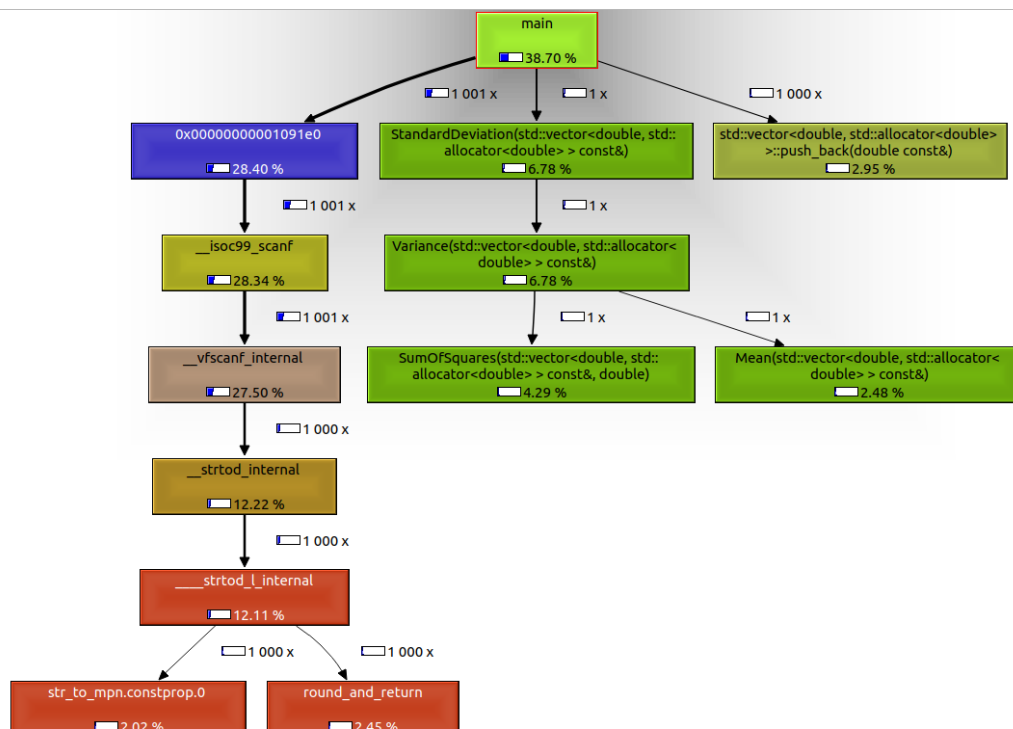


PROFILER REPORT

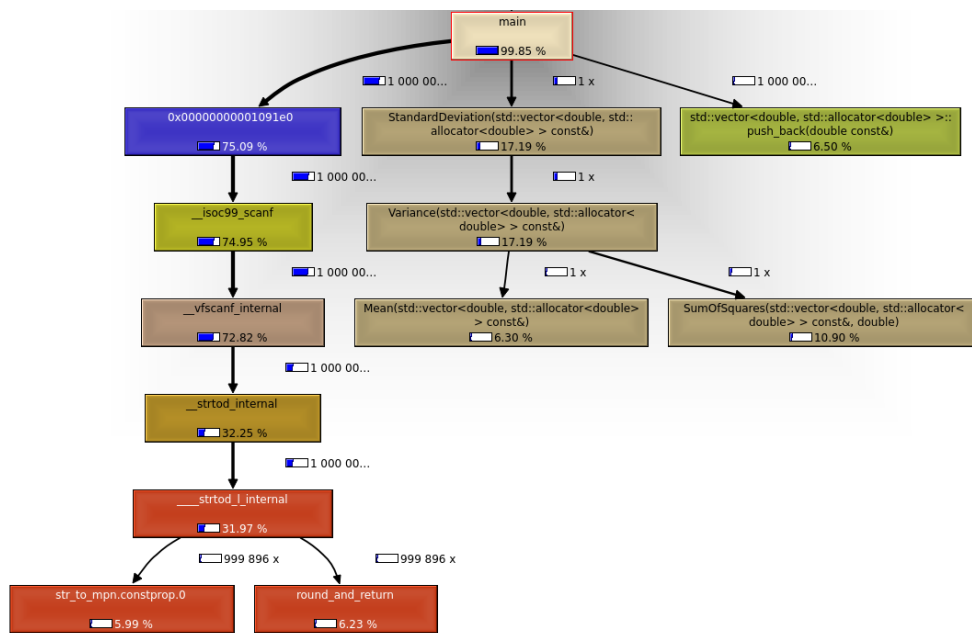
1st test – 10 input values



2nd test – 1000 input values



3rd test – 1000 000 input values



Incl.	Self	Called	Function	Location
2.69	2.69	1 000 000	Power(double, double)	standardDeviation: mathLib.cpp
1.84	1.84	2 000 000	Addition(double, double)	standardDeviation: mathLib.cpp
0.92	0.92	1 000 000	Subtraction(double, double)	standardDeviation: mathLib.cpp
0.00	0.00	1	Nthroot(double, double)	standardDeviation: mathLib.cpp
0.00	0.00	2	Division(double, double)	standardDeviation: mathLib.cpp

Incl.	Self	Called	Function	Location
99.85	1.06	1	main	standardDeviation: standartDeviation.cpp
17.19	0.00	1	StandardDeviation(std::vec...	standardDeviation: standartDeviation.cpp
17.19	0.00	1	Variance(std::vector<doubl...	standardDeviation: standartDeviation.cpp
10.90	2.55	1	SumOfSquares(std::vector...	standardDeviation: standartDeviation.cpp
6.30	1.56	1	Mean(std::vector<double, ...	standardDeviation: standartDeviation.cpp

```

69 int main()
70 {
71     std::vector<double> values;
72     double num;
73     while (scanf("%lf", &num) == 1)
74     {
75         values.push_back(num);
76     }
77     double sd = StandardDeviation(values);
78     printf("%lf\n", sd);
79     return 0;
80 }

```

(some additional photos were added to test 3:

-The usage of library functions – number of times they were called and how much time was spent in them

-StandardDeviation functions analysis

-Annotated code of the Main() function)

Conclusion: As we can see from the given graphs, the most machine time was spent in the section of code labeled Main. This amount of time is significantly increased with the size of the input file. The reason is probably the absence of assistant functions – during optimalization, it would be beneficial to add a separate function that handles loading input data. If we look at the library functions, the most machine time was spent in Power() function, so that part of the code is also fit for optimalization.