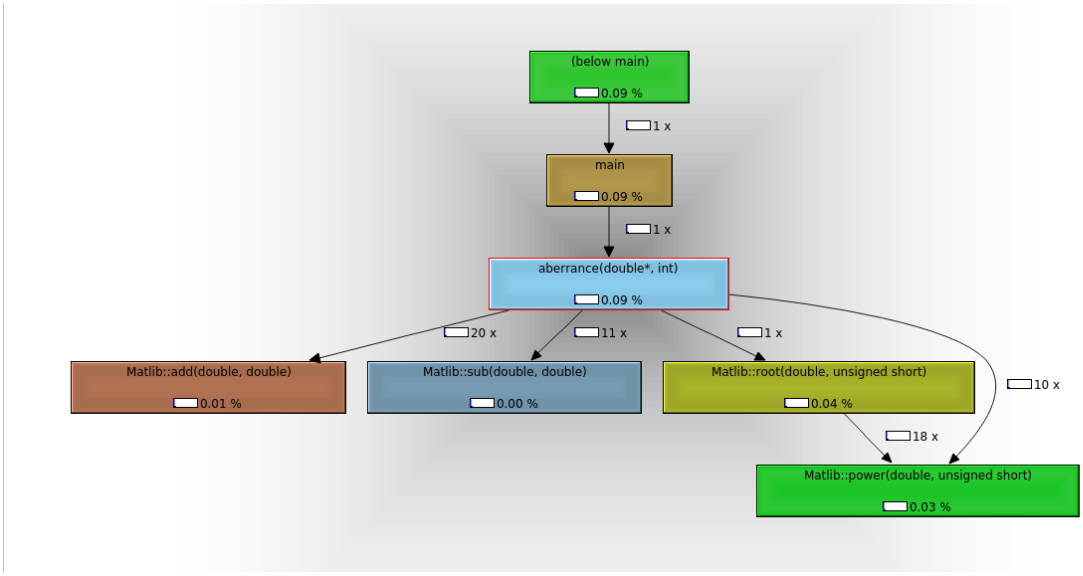
Výstup pro 10 čísel:



1 - graf volání funkcí z matlib.h

aberrance(double*, int)

Types | Callers | All Callers | Callee Map | Source Code

```
#    Ir       Source
  0           --- From '/home/jonys/CalculandumMachina-main/profiling/stddeviation.cpp' ---
 26
 27
 28           double aberrance( double numbers[], int numSize )
 29     0.00  {
 30               Matlib math; //creating a instation of class
 31     0.00      long double sum = 0;
 32     0.00      double tmp = 0;
 33
 34     0.00      for ( int i = 0; i < numSize; i++ )
 35               {
 36                   //count average
 37     0.00          tmp = numbers[i];
 38     0.00          sum = math.add(sum, tmp);
        0.00      10 call(s) to 'Matlib::add(double, double)' (stddeviation: matlib.cpp)
 39               }
 40     0.00      double average = math.div(sum,numSize);
        0.00      1 call(s) to 'Matlib::div(double, double)' (stddeviation: matlib.cpp)
 41
 42     0.00      long double sumOfNumbers = 0;
 43
 44     0.00      for ( int j = 0; j < numSize; j++ )
 45               {
 46                   // sumOfNumbers = sumOfNumbers + (( x - average)^2)
 47     0.01          sumOfNumbers =  math.add( ( math.power( math.sub( numbers[j], average ), 2) ), sumOfNumbers);
        0.00      10 call(s) to 'Matlib::add(double, double)' (stddeviation: matlib.cpp)
        0.00      10 call(s) to 'Matlib::sub(double, double)' (stddeviation: matlib.cpp)
        0.01      10 call(s) to 'Matlib::power(double, unsigned short)' (stddeviation: matlib.cpp)
 48               }
 49               // total = 1 / (N-1) * sumOfNumbers
 50     0.00      double total = 0;
 51     0.00      total = math.mul( sumOfNumbers, math.div( 1, math.sub( numSize, 1 ) ));
        0.00      1 call(s) to 'Matlib::mul(double, double)' (stddeviation: matlib.cpp)
        0.00      1 call(s) to 'Matlib::sub(double, double)' (stddeviation: matlib.cpp)
        0.00      1 call(s) to 'Matlib::div(double, double)' (stddeviation: matlib.cpp)
 52     0.00      double result = math.root(total, 2);
        0.04      1 call(s) to 'Matlib::root(double, unsigned short)' (stddeviation: matlib.cpp)
 53     0.00      return result;
 54     0.00  }
 55
 56           //main function - to read numbers from file
 57
```

2 - detail volání funkcí z matlib.h zobrazený v kódu

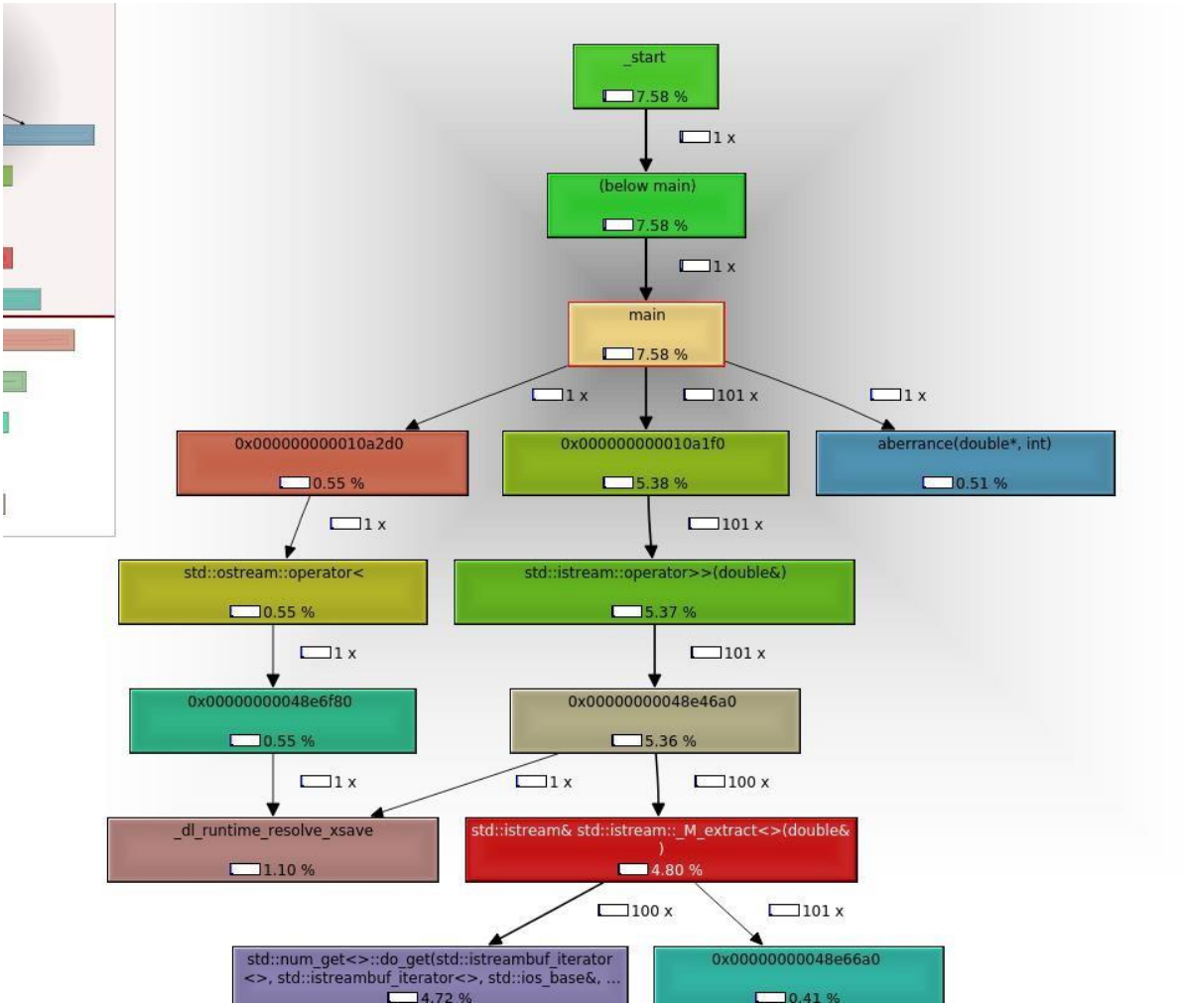
Incl.	Self	Called	Function	Location
100.00	0.00	(0)	0x0000000000001100	ld-2.31.so
88.95	0.04	1	_dl_start	ld-2.31.so: rtld.c, dl-machine.h, get-dynamic-info.h, do-rel.h
88.91	0.02	1	_dl_sysdep_start	ld-2.31.so: dl-sysdep.c, dl-sysdep.c, cpu-features.c, cpu-features.c
88.34	0.07	1	dl_main	ld-2.31.so: rtld.c, dl-prop.h, get-dynamic-info.h, setup-vdso.h, dl-vdso.h, dl-vdso-setup.h, dl-osinfo.h
86.13	15.95	7	_dl_relocate_object	ld-2.31.so: dl-reloc.c, dl-machine.h, do-rel.h, ldsodefs.h
74.08	41.03	2 244	_dl_lookup_symbol_x	ld-2.31.so: dl-lookup.c
33.05	24.61	2 244	do_lookup_x	ld-2.31.so: dl-lookup.c, ldsodefs.h
11.94	0.17	100	_dl_runtime_resolve_xsave'2	ld-2.31.so: dl-trampoline.h
9.78	0.04	21	_dl_runtime_resolve_xsave	ld-2.31.so: dl-trampoline.h
8.27	5.11	2 229	check_match	ld-2.31.so: dl-lookup.c
7.52	0.00	1	_start	stddeviation
7.52	0.00	1	(below main)	libc-2.31.so: libc-start.c
4.45	0.00	1	__libc_csu_init	stddeviation
4.45	0.00	1	_GLOBAL__sub_I__Z9aberra...	stddeviation: stddeviation.cpp
4.44	0.00	1	__static_initialization_and_d...	stddeviation: stddeviation.cpp, iostream
4.44	0.00	1	0x000000000010a280	(unknown)
4.44	0.01	1	std::ios_base::Init::Init()	libstdc++.so.6.0.28
4.34	0.41	121	_dl_fixup	ld-2.31.so: dl-runtime.c, dl-machine.h, dl-irel.h
3.63	3.63	2 635	strcmp	ld-2.31.so: strcmp.S
3.53	0.00	1	_dl_init	ld-2.31.so: dl-init.c
3.52	0.02	7	call_init.part.0	ld-2.31.so: dl-init.c
3.25	0.00	12	0x00000000048e3f10	(unknown)
3.22	0.00	1	0x00000000000a6a60	libstdc++.so.6.0.28
3.18	0.00	26	0x00000000048e45c0	(unknown)
3.18	0.00	1	malloc_hook_ini	libc-2.31.so: hooks.c, arena.c, malloc.c
3.15	0.04	1	ptmalloc_init.part.0	libc-2.31.so: arena.c, malloc.c
3.14	0.01	26	0x00000000000be5d0	libstdc++.so.6.0.28
3.10	0.00	1	0x00000000048e4100	(unknown)
3.07	3.03	1	_dl_addr	libc-2.31.so: dl-addr.c
3.07	0.03	(0)	std::locale::_Impl::_Impl(un...	libstdc++.so.6.0.28
2.81	0.02	1	main	stddeviation: stddeviation.cpp
1.23	0.10	1	_dl_map_object_deps	ld-2.31.so: dl-deps.c, scratch_buffer.h
1.18	0.00	12	0x0000000004001090	(unknown)
1.18	0.02	12	_dl_catch_exception	ld-2.31.so: dl-error-skeleton.c
1.14	0.00	1	0x00000000048e38c0	(unknown)
1.14	0.09	12	_dl_map_object	ld-2.31.so: dl-load.c
1.11	0.00	(0)	std::ctype<>::ctype(unsig...	libstdc++.so.6.0.28
1.11	0.00	1	0x00000000048e47d0	(unknown)
1.07	0.12	(0)	std::ctype<>::_M_initialize...	libstdc++.so.6.0.28
1.05	0.00	11	0x000000000010a1f0	(unknown)
1.05	0.00	11	std::istream::operator>>(d...	libstdc++.so.6.0.28
1.05	0.00	11	0x00000000048e46a0	(unknown)
1.03	0.01	11	openaux	ld-2.31.so: dl-deps.c
0.80	0.04	10	std::num_get<>::do_get(st...	libstdc++.so.6.0.28
0.77	0.00	6	0x00000000048e6290	(unknown)
0.64	0.00	1	_dl_receive_error	ld-2.31.so: dl-error-skeleton.c
0.64	0.00	1	version_check_doit	ld-2.31.so: rtld.c
0.64	0.00	1	_dl_check_all_versions	ld-2.31.so: dl-version.c
0.63	0.37	7	_dl_check_map_versions	ld-2.31.so: dl-version.c

main

Types | Callers | All Callers | Callee Map | Source Code

#	Ir	Source
0		--- From '/home/jonys/CalculandumMachina-main/profiling/stddeviation.cpp' ---
56		//main function - to read numbers from file
57		
58		int main(int argc, char *argv[])
59	0.00	{
60	0.00	ifstream file;
	0.25	1 call(s) to '0x000000000010a260'
	0.35	1 call(s) to '0x000000000010a1a0'
61		
62		
63	0.00	if (argc != 2)
64		{
65		cerr << "there must be 1 argument - file with array of numbers" << endl ;
66		return 0;
67		}
68		else
69		{
70	0.00	char *fileNum = argv[1];
71	0.00	file.open(fileNum, ifstream::in);
	0.29	1 call(s) to '0x000000000010a1d0'
72	0.00	if (!file.is_open()) //for case program can not open a file
	0.00	1 call(s) to '0x000000000010a290'
73		{
74		throw new std::runtime_error("Failed to open file");
75		return 0;
...		...
77		}
78		double num;
79		double array[10000]; // load numbers to this array
80	0.00	double numSize = 0;
81	0.06	for (int i = 0; file >> num ; i++)
	0.02	101 call(s) to '0x000000000010a270'
	5.38	101 call(s) to '0x000000000010a1f0'
82		{
83	0.02	array[i] = num;
84	0.02	numSize++;
85		}
86		
87	0.00	double deviation = aberrance(array, numSize);
	0.51	1 call(s) to 'aberrance(double*, int)' (stddeviation: stddeviation.cpp)
88	0.00	cout << deviation << endl;
	0.13	1 call(s) to '0x000000000010a240'
	0.55	1 call(s) to '0x000000000010a2d0'
89	0.00	return 0;
90	0.00	}
91		
92		
93		

3 - přehled všech běžících procesů



4 - přehled všech běžících procesů znázorněný v grafu