

# Profiling

Na profilování byl původně vybrán nástroj *gprof*. Ten se však bohužel neosvědčil. Opakovaně jsme se střetli s problémem, kde výstup z *gprof* vrátil počet volání použitých funkcí, ale bez časů strávených v nich.

Proto jsme se rozhodli, použít profilovací nástroj *Callgrind*, který je součástí nástroje *Valgrind*. Následně jsme výstup z *Callgrindu* reprezentovali nástrojem *QCacheGrind*, který je windows obdobou nástroje *KCacheGrind*.

Měřicí jednotkou nástroje *KCacheGrind* je „Total Instruction Fetch Cost“. Po zadání 10 náhodných jedno ciferných hodnot do profilovací funkce *deviation()* měl „Total Instruction Fetch Cost“ hodnotu 3 673 691. Tato hodnota se dále dělí procentuálně na jednotlivé funkce, které se při běhu programu volali.

Incl.	Self	Called	Function	Location
100.00	0.00	(0)	0x0000000000001170	ld-2.17.so
84.20	0.02	1	_dl_start	ld-2.17.so
84.18	0.01	1	_dl_sysdep_start	ld-2.17.so
84.17	0.03	1	dl_main	ld-2.17.so
83.31	21.91	1 882	_dl_lookup_symbol_x	ld-2.17.so
82.79	7.58	7	_dl_relocate_object	ld-2.17.so
61.39	39.00	1 882	do_lookup_x	ld-2.17.so
26.02	0.17	154	_dl_runtime_resolve_xsave2	ld-2.17.so
21.98	13.00	15 246	check_match.9515	ld-2.17.so
15.70	0.01	9	_dl_runtime_resolve_xsave	ld-2.17.so
13.66	0.00	1	0x00000000004022a0	profiling
13.61	0.00	(0)	(below main)	libc-2.17.so
9.35	1.93	10 130	strcmp	ld-2.17.so
8.86	0.02	1	main	profiling: profilingApplication.cpp
8.42	0.31	163	_dl_fixup	ld-2.17.so
7.42	7.42	38 347	strcmp'2	ld-2.17.so
4.79	0.03	1	ProfilingApplication::convertStr...	profiling: profilingApplication.cpp
4.34	0.00	1	__libc_csu_init	profiling
4.32	0.00	1	_GLOBAL__sub_I_ZN20Profilin...	profiling: profilingApplication.cpp
4.27	0.00	1	__static_initialization_and_destr...	profiling: profilingApplication.cpp, iostream
2.84	0.01	77	std::locale::_S_initialize()	libstdc++.so.6.0.24: locale_init.cc

Obrázek č. 1: Inclusive cost

Na obrázku č. 1 můžete vidět funkce, seřazené podle procentuálního podílu času stráveného ve funkcích. Tato hodnota zahrnuje i čas, který se strávil ve funkcích volaných danými funkcemi. Proto zde vidíme hodnotu 100%. Funkce *main()*, která se stará o běh profilingu, má hodnotu 8.86 %.

Incl.	Self	Called	Function	Location
61.39	39.00	1 882	do_lookup_x	ld-2.17.so
83.31	21.91	1 882	_dl_lookup_symbol_x	ld-2.17.so
21.98	13.00	15 246	check_match.9515	ld-2.17.so
82.79	7.58	7	_dl_relocate_object	ld-2.17.so
7.42	7.42	38 347	strcmp'2	ld-2.17.so
9.35	1.93	10 130	strcmp	ld-2.17.so
1.53	1.53	1	_dl_addr	libc-2.17.so
0.93	0.93	1 081	_mcount_internal	libc-2.17.so
1.52	0.59	1 080	mcount	libc-2.17.so
0.64	0.32	246	__dynamic_cast	libstdc++.so.6.0.24: dyncast.cc, tinfo.h
0.61	0.31	416	_dl_name_match_p	ld-2.17.so
8.42	0.31	163	_dl_fixup	ld-2.17.so
0.30	0.30	29	std::locale::_Impl::_M_install_fac...	libstdc++.so.6.0.24: locale.cc, atomicity.h
0.25	0.22	255	btowc	libc-2.17.so
26.02	0.17	154	_dl_runtime_resolve_xsave'2	ld-2.17.so
0.15	0.15	172	__cxxabiv1::__si_class_type_info::...	libstdc++.so.6.0.24: si_class_type_info.cc, ty
0.16	0.14	47	_int_malloc	libc-2.17.so
0.21	0.12	75	__cxxabiv1::__vmi_class_type_inf...	libstdc++.so.6.0.24: vmi_class_type_info.cc,
0.12	0.12	48	_dl_cache_libcmp	ld-2.17.so
0.34	0.12	7	_dl_check_map_versions	ld-2.17.so
0.31	0.12	5	_dl_map_object_from_fd	ld-2.17.so

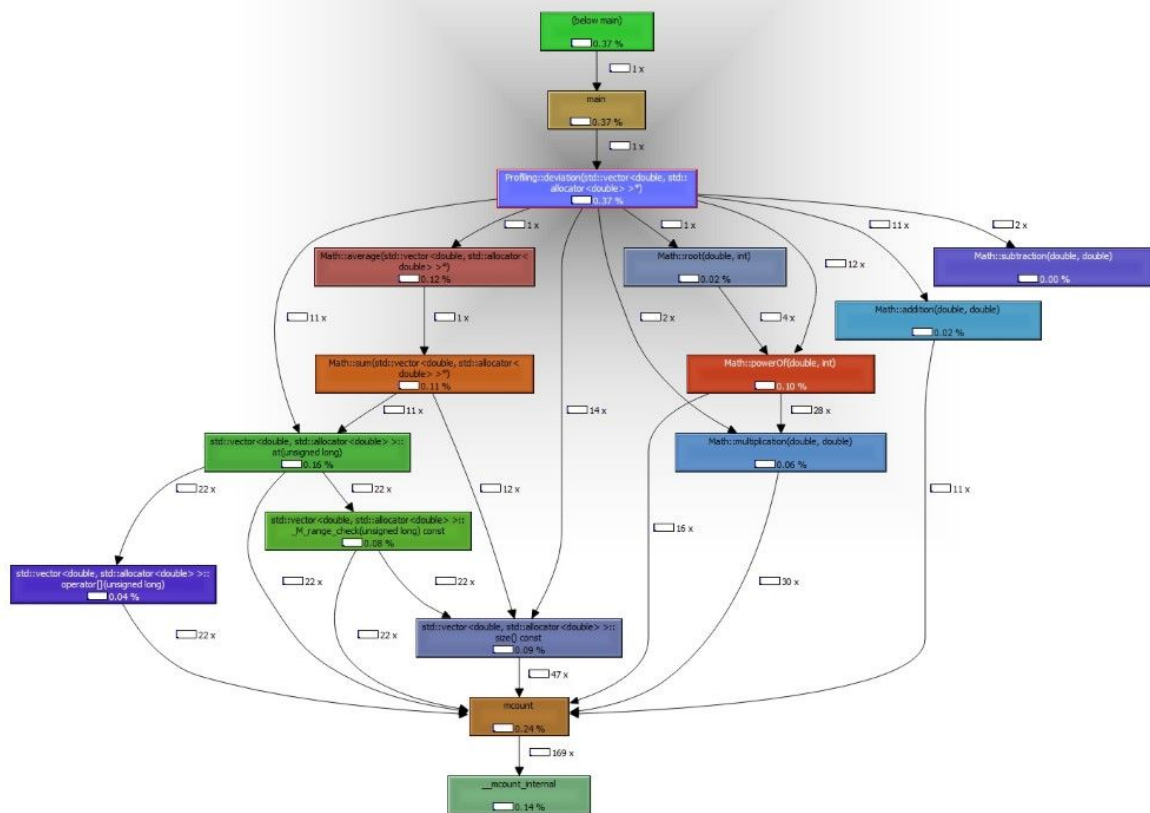
Obrázek č. 2: Self cost

Na obrázku č. 2 můžete vidět funkce seřazené podle procentuálního podílu času stráveného pouze v daných funkcích. Převažují zde funkce ze standardních knihoven jazyka C++.

Incl.	Self	Called	Function	Location
7.42	7.42	38 347	strcmp'2	ld-2.17.so
21.98	13.00	15 246	check_match.9515	ld-2.17.so
9.35	1.93	10 130	strcmp	ld-2.17.so
61.39	39.00	1 882	do_lookup_x	ld-2.17.so
83.31	21.91	1 882	_dl_lookup_symbol_x	ld-2.17.so
0.93	0.93	1 081	_mcount_internal	libc-2.17.so
1.52	0.59	1 080	mcount	libc-2.17.so
0.61	0.31	416	_dl_name_match_p	ld-2.17.so
0.05	0.05	317	std::locale::id::_M_id() const	libstdc++.so.6.0.24: locale.cc, atomicity.h
0.25	0.22	255	btowc	libc-2.17.so
0.64	0.32	246	__dynamic_cast	libstdc++.so.6.0.24: dyncast.cc, tinfo.h
0.15	0.15	172	__cxxabiv1::__si_class_type_info::...	libstdc++.so.6.0.24: si_class_type_info.cc,
8.42	0.31	163	_dl_fixup	ld-2.17.so
26.02	0.17	154	_dl_runtime_resolve_xsave'2	ld-2.17.so
0.02	0.02	128	_gconv_btowc_ascii	libc-2.17.so
0.01	0.01	128	_dl_mcount_wrapper_check	libc-2.17.so
0.07	0.07	127	wctob	libc-2.17.so
0.00	0.00	90	rtld_lock_default_lock_recursive	ld-2.17.so
0.00	0.00	90	rtld_lock_default_unlock_recur...	ld-2.17.so
0.05	0.01	86	std::__cxx11::basic_string<char, ...	libstdc++.so.6.0.24: basic_string.h, new_a
0.03	0.03	80	_GI_strlen	libc-2.17.so

Obrázek č. 3: Funkce seřazené podle počtu volání

Četné volání funkcí *strcmp'2* a *strcmp* je z důvodu časté konverze mezi „vektorem doublů“ a „stringů“ a naopak. Výrazné zpomalení jsme ale nepozorovali.



Obrázek č. 4: Call graph funkce *deviation()*

Na obrázku č. 4 můžete spatřit veškeré funkce z naší matematické knihovny, které byly použity na výpočet výběrové směrodatné odchylky. Graf taktéž ukazuje procentuální čas strávený ve funkcích (Self cost) a zároveň i počet volání jednotlivých funkcí.

Incl.	Self	Called	Function	Incl.	Self	Called	Function	Incl.	Self	Called	Function
61.39	39.00	1 882	do_lookup_x	46.64	29.63	1 882	do_lookup_x	10.26	10.26	62 761	__mcount_internal
83.31	21.91	1 882	_dl_lookup_symbol_x	63.29	16.65	1 882	_dl_lookup_symbol_x	10.09	9.60	7 973	_int_malloc
21.98	13.00	15 246	check_match.9515	16.70	9.87	15 246	check_match.9515	12.29	7.81	1 882	do_lookup_x
82.79	7.58	7	_dl_relocate_object	62.90	5.76	7	_dl_relocate_object	7.41	7.010	__memmove_sse3_back	
7.42	7.42	38 347	strcmp.2	5.63	5.63	38 347	strcmp.2	17.10	6.84	62 760	mcount
9.35	1.93	10 130	strcmp	4.26	4.26	6 811	__mcount_internal	9.61	4.72	18 066	_dynamic_cast
1.53	1.53	1	_dl_addr	7.08	2.82	6 810	mcount	4.47	4.47	7 971	_int_free
0.93	0.93	1 081	__mcount_internal	1.91	1.88	770	_int_malloc	16.67	4.39	1 882	_dl_lookup_symbol_x
1.52	0.59	1 080	mcount	3.76	1.85	1 866	_dynamic_cast	4.40	2.60	15 246	check_match.9515
0.64	0.32	246	_dynamic_cast	7.10	1.47	10 130	strcmp	2.17	2.17	12 052	__cxxabiv1::__si_class_type_info::...
0.61	0.31	416	_dl_name_match_p	1.36	1.36	769	_int_free	2.73	2.00	6 015	__cxxabiv1::__vmi_class_type_info::...
8.42	0.31	163	_dl_fixup	1.16	1.16	1	_dl_addr	11.83	1.74	7 972	malloc
0.30	0.30	29	std::locale::_Impl::_M_install_fac	0.85	0.85	1 252	__cxxabiv1::__si_class_type_info::...	21.50	1.72	5 002	void std::__cxx11::basic_string<c...
0.25	0.22	255	btowc	1.09	0.78	615	__cxxabiv1::__vmi_class_type_inf...	16.57	1.52	7	_dl_relocate_object
26.02	0.17	154	_dl_runtime_resolve_xsave2	5.90	0.65	502	void std::__cxx11::basic_string<c...	2.42	1.50	1 001	vfprintf
0.15	0.15	172	__cxxabiv1::__si_class_type_info::...	2.54	0.64	769	malloc	1.48	1.48	38 347	strcmp.2
0.16	0.14	47	_int_malloc	0.94	0.57	101	vfprintf	1.92	1.12	1 000	__strtod_l_internal
0.21	0.12	75	__cxxabiv1::__vmi_class_type_inf...	0.48	0.48	707	__memmove_sse3_back	5.24	0.77	9 972	free
0.12	0.12	48	_dl_cache_libcmp	0.74	0.43	100	__strtod_l_internal	0.72	0.72	6 014	_strcmp_sse42
0.34	0.12	7	_dl_check_map_versions	1.64	0.28	970	free	5.34	0.60	5 003	std::iterator_traits<char*>::diffe...
0.31	0.12	5	_dl_map_object_from_fd	0.28	0.28	614	_strcmp_sse42	12.84	0.59	3 006	std::basic_ios<char, std::char_tr...

Obrázek č. 5: Porovnání 10, 100 a 1000 vzorků na vstupu

Obrázek č. 5 zobrazuje porovnání výsledků při vstupu 10, 100 a 1000 (zleva) číselných hodnot. Můžete si povšimnout, že čím více hodnot je na vstupu, tím se čas strávený ve funkcích rozkládá mezi více funkcí.

## **Záver**

Jelikož rychlost výpočtu programu byla uspokojující, nerozhodli jsme se pro další optimalizace. Tento závěr připisujeme časté refaktORIZACI kódu v průběhu vývoje i kvalitnímu návrhu.