VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

IVS – 2. projekt

Profiling: Výstupní zpráva

18. 4. 2021 R. Marek, J. Dědič, V. Dvořák, T. Dvořák

Obsah zprávy

Úvod

V následujícím textu je popsán profiling programu *math_prof,* který byl vytvořen v rámci vývoje kalkulačky *dumdumCalc*. Tento program načítá ze standartního vstupu hodnoty (celá nebo desetinná čísla) a následně z nich pomocí funkcí z matematické knihovny *math_lib* (taktéž využívané v kalkulačce) počítá tzv. výběrovou směrodatnou odchylku, kterou poté tiskne na standartní výstup.

Pro profiling byly využity, pro větší směrodatnost výsledků, hned dva programy – *callgrind* a *gprof*. Program byl profilován třikrát (s 10, 100 a 1000 hodnotami) na platformě Ubuntu 20.04 (64bit).

Gprof

Překlad programu

Program *math_prof* byl přeložen pro gprof následujícím způsobem:

```
g++ -pg -g -O2 -Werror -Wall -pedantic -std=c++11 -c -o
build/math_prof.o math_prof.cpp
```

```
g++ -pg -g -O2 build/math_prof.o build/math_lib.o  -lm -o
build/math_prof
```

Přepínač *-O2* byl přidán za účelem automatické optimalizace (a lepších výsledků).

Profiling

Samotné profiler byl spuštěn touto sekvencí příkazů:

```
./build/math_prof < profiling_sample
```

```
gprof -b ./build/math_prof
```

Kde soubor *profiling_sample* byl vzorek s náhodně vygenerovanými 10/100/1000 hodnotami oddělenými náhodným počtem a typem bílých znaků (mezery, tabulátory, nové řádky). Pro účely tvorby vzorků jsme vytvořili jednoduchý skript v jazyce Python (viz Příloha č. 1).

Výstup

Po profilování s 10, 100, 1000 byly výstupy programu gprof spojeny do souboru *vystup.txt* a odděleny oddělovači. Pro větší přehlednost zde uvádíme pouze prvních pět řádků Flat profile z každého běhu programu (kompletní výstup je dostupný ve výše zmíněném souboru).

10 hodnot:

```
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name

  0.00      0.00     0.00       20     0.00     0.00  add(double, double)

  0.00      0.00     0.00       11     0.00     0.00  f_pow(double, double)

  0.00      0.00     0.00       10     0.00     0.00  void
std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >::_M_construct<char*>(char*, char*,
std::forward_iterator_tag)
```

```
  0.00      0.00     0.00        5     0.00     0.00  void
std::vector<std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >, std::allocator<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > >
>::_M_realloc_insert<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> >
const&>(__gnu_cxx::__normal_iterator<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> >*,
std::vector<std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >, std::allocator<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > > > >,
std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> > const&)

  0.00      0.00     0.00        5     0.00     0.00  void
std::vector<double, std::allocator<double>
>::_M_realloc_insert<double>(__gnu_cxx::__normal_iterator<double*,
std::vector<double, std::allocator<double> > >, double&&)
```

100 hodnot:

```
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name

  0.00      0.00     0.00      200     0.00     0.00  add(double, double)

  0.00      0.00     0.00      101     0.00     0.00  f_pow(double, double)

  0.00      0.00     0.00      100     0.00     0.00  void
std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >::_M_construct<char*>(char*, char*,
std::forward_iterator_tag)

  0.00      0.00     0.00        8     0.00     0.00  void
std::vector<std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >, std::allocator<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > >
>::_M_realloc_insert<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> >
const&>(__gnu_cxx::__normal_iterator<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> >*,
std::vector<std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >, std::allocator<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > > > >,
std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> > const&)

  0.00      0.00     0.00        8     0.00     0.00  void
std::vector<double, std::allocator<double>
>::_M_realloc_insert<double>(__gnu_cxx::__normal_iterator<double*,
std::vector<double, std::allocator<double> > >, double&&)
```

1000 hodnot:

```
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name

  0.00      0.00     0.00     2000     0.00     0.00  add(double, double)

  0.00      0.00     0.00     1001     0.00     0.00  f_pow(double, double)
```

```
  0.00      0.00     0.00     1000     0.00     0.00  void
std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >::_M_construct<char*>(char*, char*,
std::forward_iterator_tag)

  0.00      0.00     0.00       11     0.00     0.00  void
std::vector<std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >, std::allocator<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > >
>::_M_realloc_insert<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> >
const&>(__gnu_cxx::__normal_iterator<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> >*,
std::vector<std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> >, std::allocator<std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > > > >,
std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> > const&)

  0.00      0.00     0.00       11     0.00     0.00  void
std::vector<double, std::allocator<double>
>::_M_realloc_insert<double>(__gnu_cxx::__normal_iterator<double*,
std::vector<double, std::allocator<double> > >, double&&)
```

Callgrind

Překlad programu

Program *math_prof* byl přeložen pro callgrind následujícím způsobem:

```
g++ -O2 -Werror -Wall -pedantic -std=c++11 -c -o build/math_prof.o
math_prof.cpp

g++ -O2 build/math_prof.o build/math_lib.o  -lm -o build/math_prof
```

Oproti překladu pro gprof byly odebrány přepínače *-g* a *-pg*, které pro profilování callgrindem nejsou potřeba a naopak byly příčinou občasných chybových hlášení při jeho běhu.

Profiling

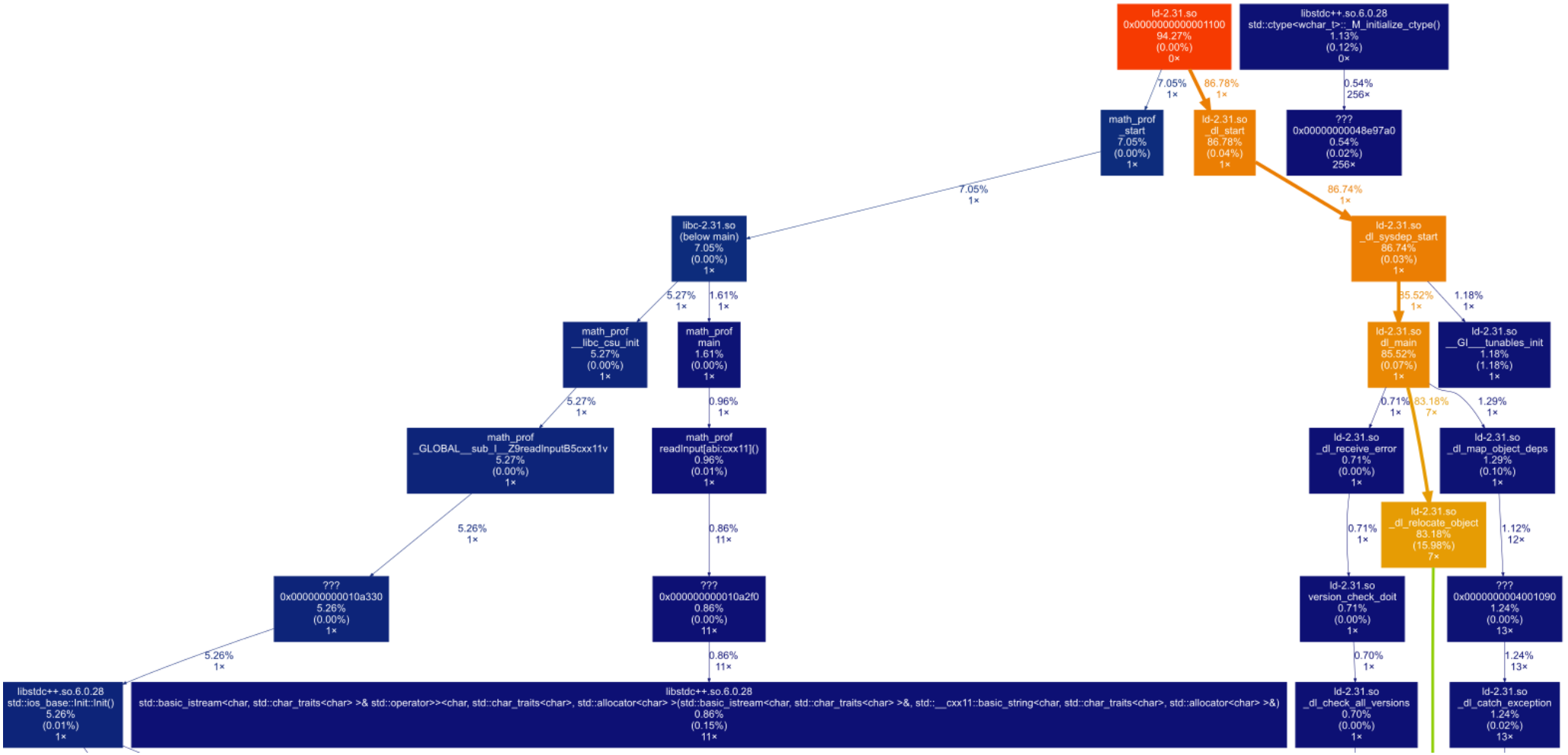
Nástroj callgrind byl po kompilaci spuštěn tímto příkazem:

```
valgrind --tool=callgrind --callgrind-out-file=callgrind-prof.out\
./build/math_prof < profiling_sample
```

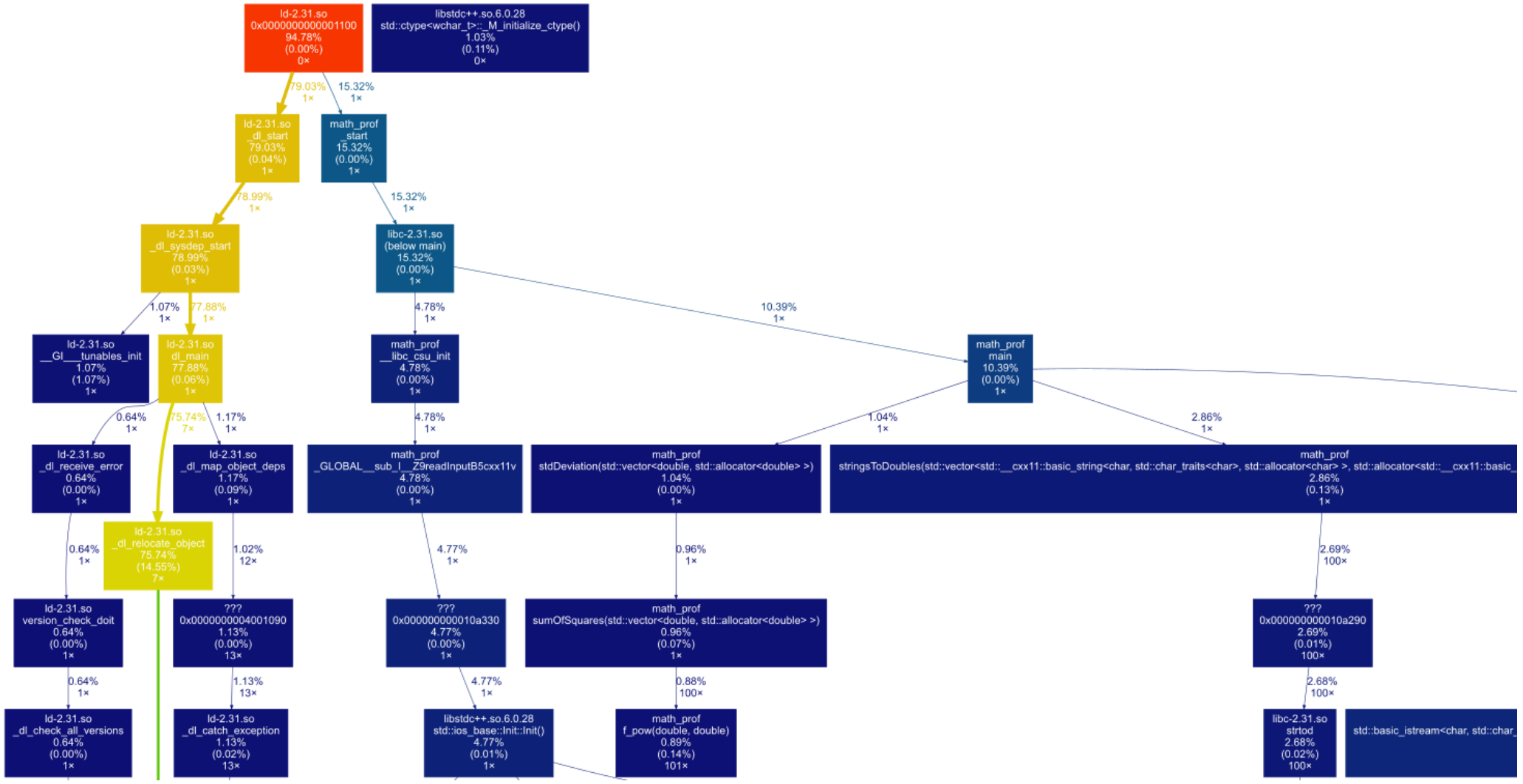
Kde soubor *profiling_sample* byl vzorek s náhodně vygenerovanými 10/100/1000 hodnotami oddělenými náhodným počtem a typem bílých znaků (mezery, tabulátory, nové řádky).

Výstup

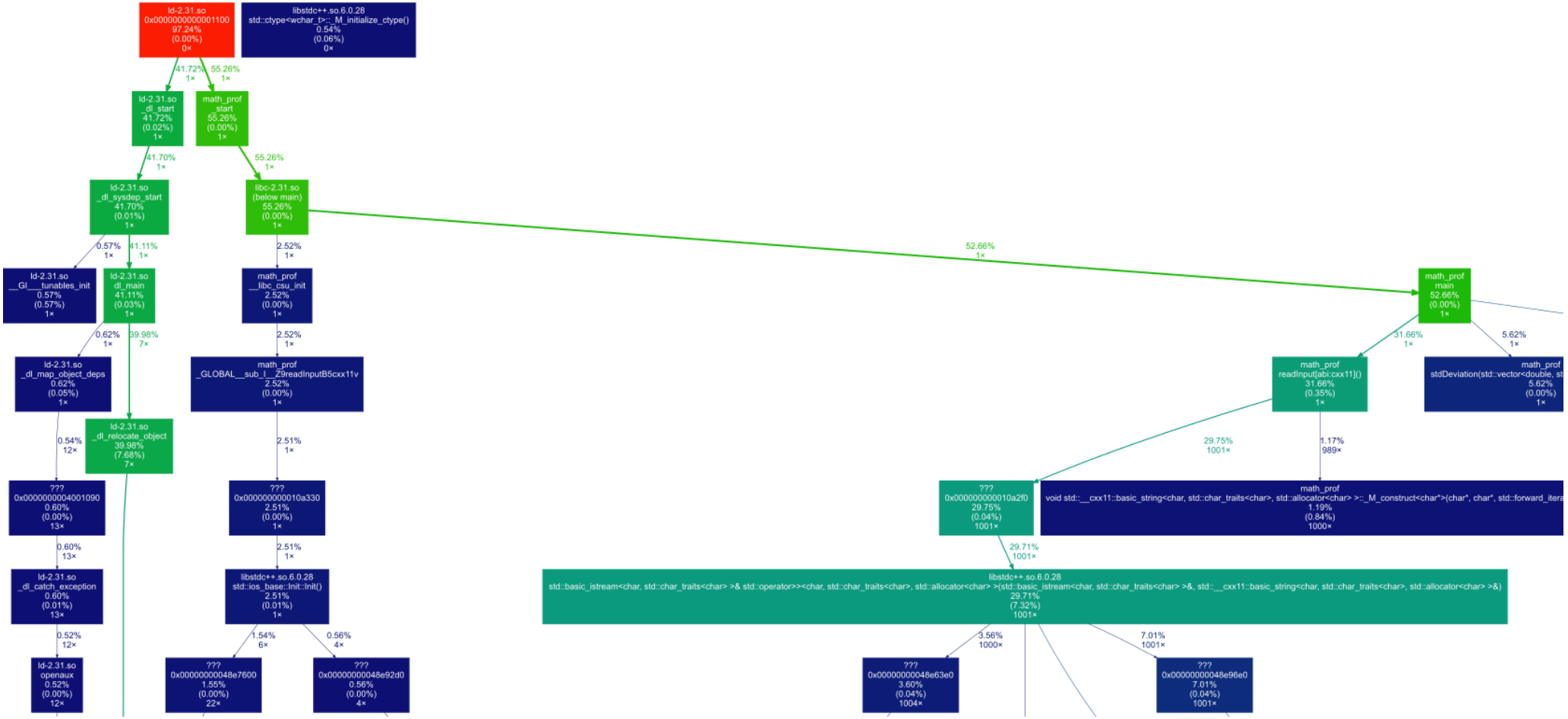
Výsledné soubory v callgrind formátu byly pro lepší čitelnost následně konvertovány nástrojem graph2dot.py do grafické podoby. Pro větší přehlednost uvádíme výstřižky z grafů s kořeny příp. částmi grafu, které zachycují volání matematických funkcí z knihovny (viz následující stránky). Celé grafy je možné si prohlédnout v příloze.



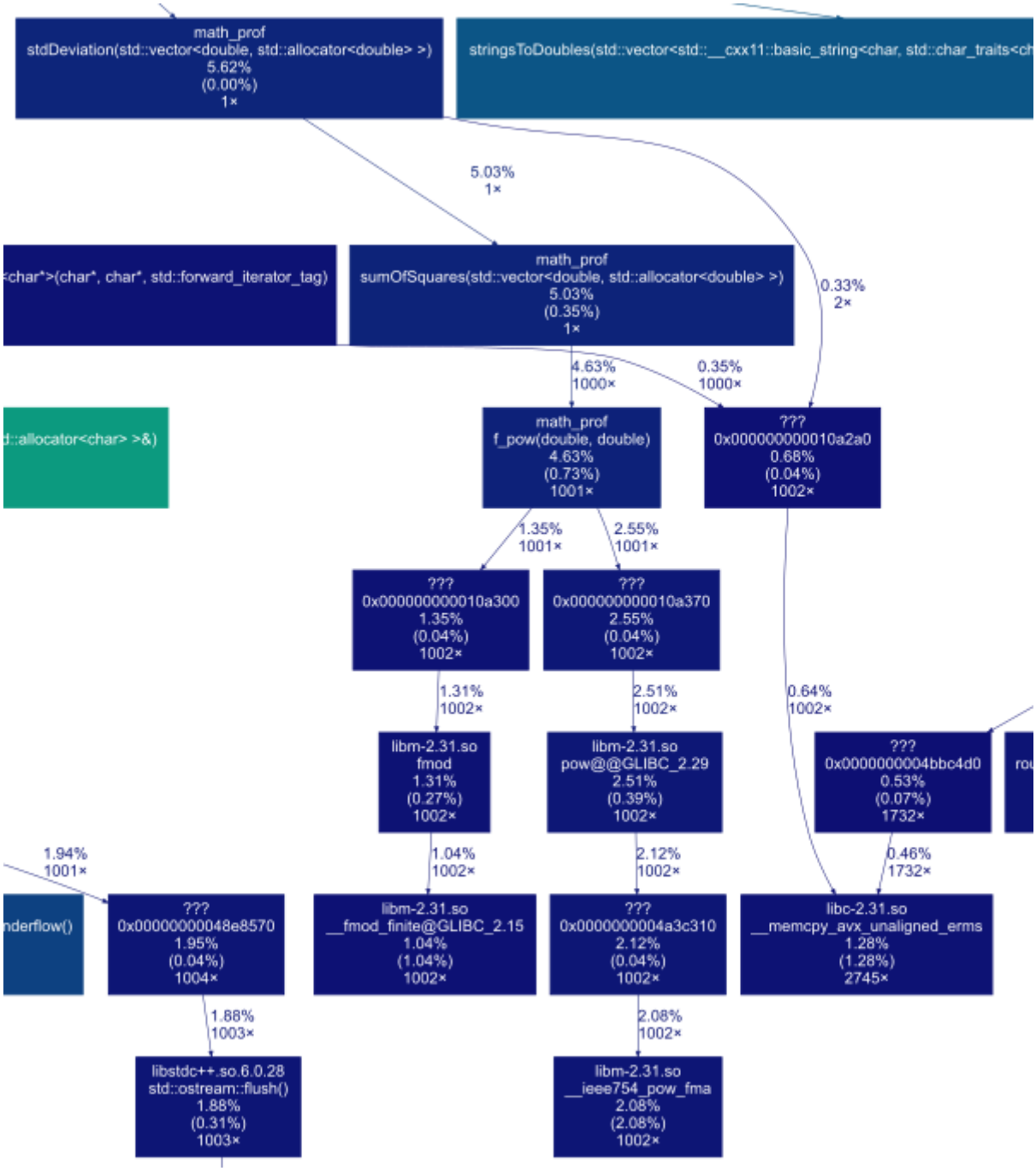
Obrázek 1 - Výsledek profilingu s 10 hodnotami (callgrind)



Obrázek 2 - Výsledek profilingu se 100 hodnotami (callgrind)



Obrázek 3 - Výsledek profilingu s 1000 hodnotami (callgrind)



Obrázek 4 - Výsledek profilingu s 1000 hodnotami (callgrind) - volání matematických funkcí

Interpretace výsledků

Vzhledem ke krátkému běhu programu, program gprof nenaměřil žádný čas. Z jeho výstupů však poměrně přehledně vidíme nejčastěji volané funkce.

Nejčastěji byly podle gprofu volány funkce *add* (tj. sčítání dvou reálných čísel) a *f_pow* (umocňování s přirozeným exponentem) z matematické knihovny. S přibývajícím počtem hodnot roste pochopitelně také počet jejich volání, protože jsou tyto funkce volány pro každou načtenou hodnotu ze stdin. Dále jsou pak často volány funkce, které souvisejí s režií tříd *string* a *vector* (realokace, konstruktory), jež slouží pro uložení načtených hodnot.

Absence časových údajů způsobila nedůvěru v gprof a proto jsme se rozhodli získat podrobnější data prostřednictvím callgrindu. Výsledné grafy ukázaly, že se program velkou část doby svého běhu zabývá vyhledáváním funkcí ve sdílených knihovnách a voláním funkcí s tím spojených (např. `check_match`). S rostoucím počtem hodnot ve vzorku se však podíl této operace na celkové době běhu snižuje (resp. počet volání zůstává stejný jak pro 10, tak i pro 1000 hodnot), protože se nejspíše jedná o jednorázovou akci.

Závěr

Při optimalizaci se doporučujeme zaměřit na umocňování, které v tomto případě (pouze umocňování druhou mocninou), lze snadno realizovat např. násobením, jež bude zřejmě ke zdrojům šetrnější.

Dále by bylo vhodné se při optimalizaci zabývat třídami sloužícími pro uchování a čtení hodnot (`vector` a `string`). Navrhujeme předávat hodnoty funkcím zásadně prostřednictvím ukazatele (nikoliv skrze hodnoty) nebo např. implementovat vlastní jednodušší struktury. Tyto úpravy by mohly vést ke snížení počtu volaných funkcí (a ke snížení nároků na zásobník).

Nakonec navrhujeme se ujistit, že všechna volání funkcí ze sdílených knihoven (byť standartních) jsou opravdu nutná, popř. počet těchto funkcí zredukovat na minimum. Mohl by se pak snížit čas potřebný k jejich vyhledání pomocí *ld-2.31.so*.

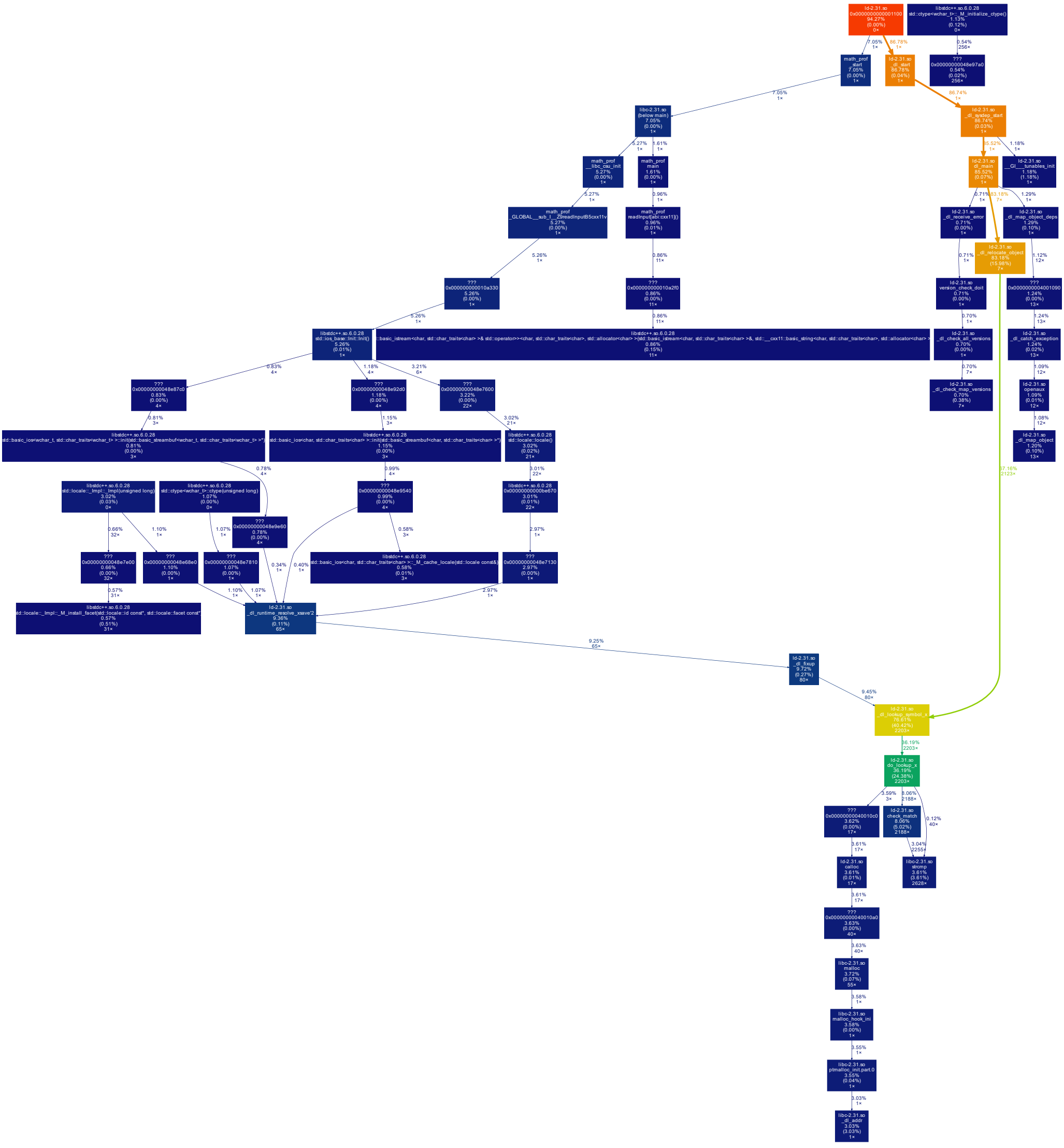
Příloha č. 1 – Skript pro generování vzorků

Pro generování vzorků využitých při profilování byl využit následující skript v jazyce Python. Přijímá právě jeden argument, tj. počet náhodných hodnot, které mají být vygenerovány, oddělených náhodným počtem a typem bílých znaků.

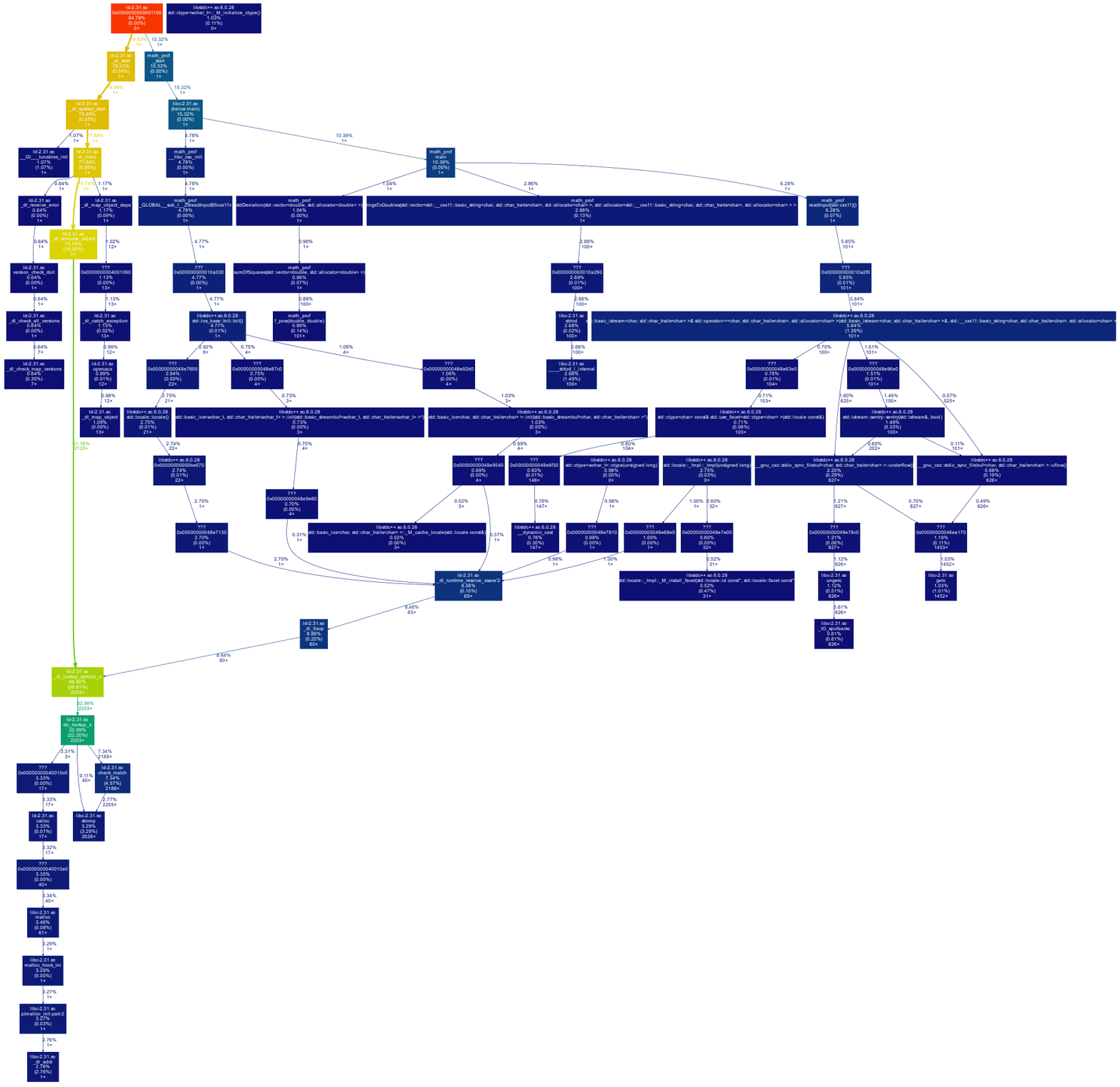
```
#!/usr/bin/python


import sys

import random


separators = ["\n", " ", "\t"]


if len(sys.argv) == 1:

    print("Missing argument!")

    quit(1)


for x in range(int(sys.argv[1])):

    number = random.random()*pow(10, random.randint(0,3))

    number = round(number, random.randint(0,5))

    print(number, end=random.choice(separators))


print()
```

Příloha č. 2 – Úplné grafy vygenerované nástroji callgrind a graph2dot

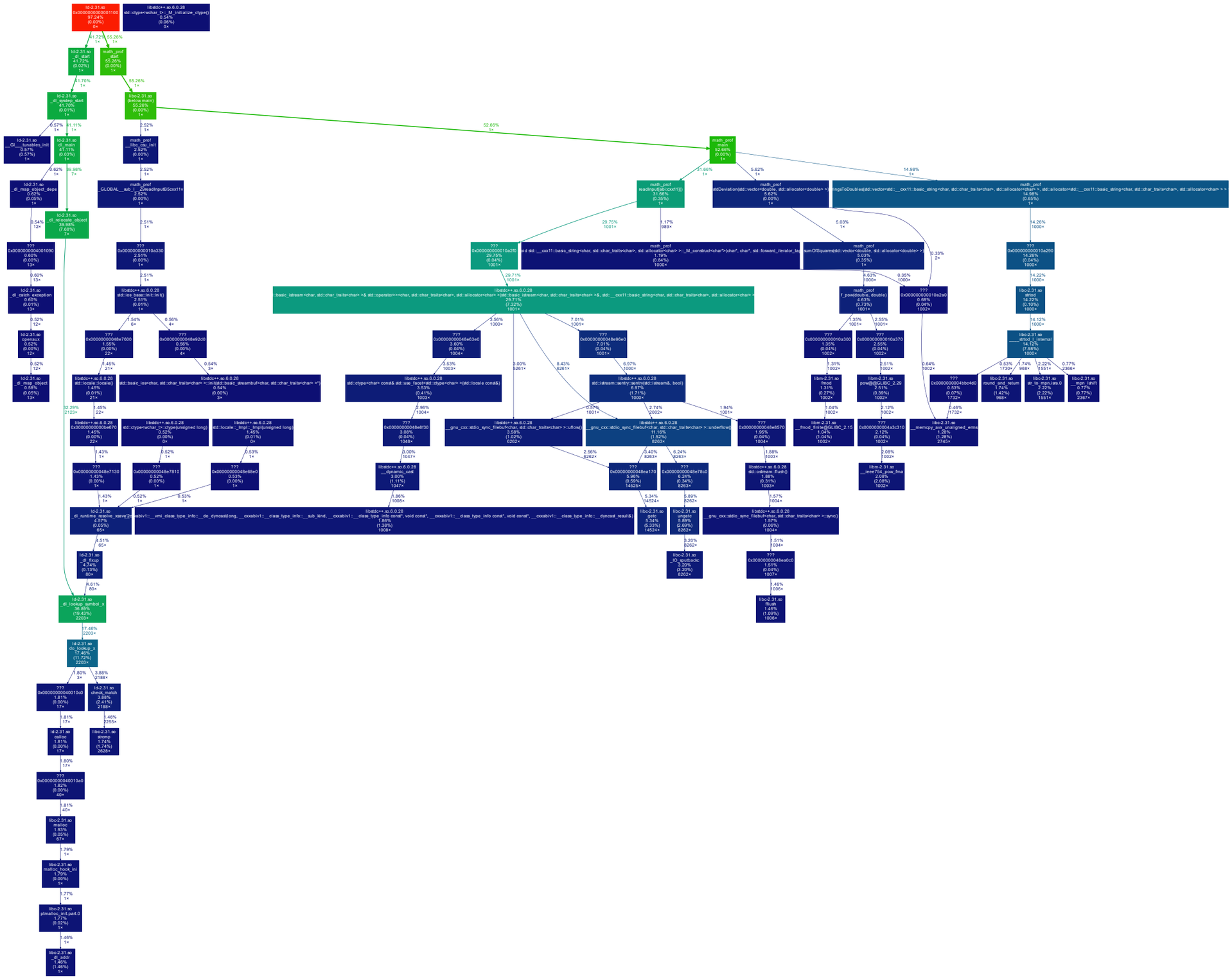
10 hodnot



100 hodnot


ld-2.31.so
0x0000000000001100
94.78%
(0.00%)
0×
libstdc++.so.6.0.28
std::ctype<wchar_t>::_M_initialize_ctype()
1.03%
(0.11%)
0×
9.03%
1×
15.32%
1×
ld-2.31.so
_dl_start
79.03%
(0.24%)
1×
math_prof
start
15.32%
(0.00%)
1×
6.99%
1×
15.32%
1×
ld-2.31.so
_dl_sysdep_start
78.99%
(0.03%)
1×
libc-2.31.so
(below main)
15.32%
(0.00%)
1×
4.07%
1×
7.88%
1×
4.78%
1×
10.39%
1×
ld-2.31.so
__GI___tunables_init
1.07%
(1.07%)
1×
ld-2.31.so
_dl_main
77.84%
(0.06%)
1×
math_prof
libc_csu_init
4.76%
(0.00%)
1×
math_prof
main
10.39%
(0.00%)
1×
0.64%
1×
5.74%
7×
1.17%
1×
4.78%
1×
1.04%
1×
2.86%
1×
6.28%
1×
ld-2.31.so
_dl_receive_error
0.64%
(0.00%)
1×
ld-2.31.so
_dl_map_object_deps
1.17%
(0.09%)
1×
math_prof
_GLOBAL__sub_I__ZReadInputIBScxx11v
4.78%
(0.00%)
1×
math_prof
stdDeviation(std::vector<double, std::allocator<double> >)
1.04%
(0.00%)
1×
math_prof
readInputI(std::vector<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, std::allocator<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > >
2.86%
(0.13%)
1×
math_prof
readInputI(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >)
6.28%
(0.07%)
1×
0.64%
1×
1.02%
12×
4.77%
1×
0.96%
1×
2.69%
100×
5.85%
101×
ld-2.31.so
_dl_relocate_object
75.74%
(14.55%)
7×
ld-2.31.so
version_check_doit
0.64%
(0.00%)
1×
???
0x0000000004001090
1.13%
(0.00%)
13×
???
0x00000000010a330
4.77%
(0.00%)
1×
math_prof
sumOfSquares(std::vector<double, std::allocator<double> >)
0.96%
(0.07%)
1×
???
0x00000000010a290
2.69%
(0.01%)
100×
???
0x00000000010a2f0
5.85%
(0.01%)
101×
0.64%
1×
1.13%
13×
4.77%
1×
0.88%
100×
2.68%
100×
5.84%
101×
ld-2.31.so
_dl_check_all_versions
0.84%
(0.00%)
1×
ld-2.31.so
_dl_catch_exception
1.13%
(0.02%)
13×
libstdc++.so.6.0.28
std::ios_base::Init::Init()
4.77%
(0.01%)
1×
math_prof
f_pow(double, double)
0.89%
(0.14%)
101×
libc-2.31.so
__strtod_l
2.68%
(0.02%)
100×
libstdc++.so.6.0.28
std::basic_istream<char, std::char_traits<char> >& std::operator>><char, std::char_traits<char>, std::allocator<char> >(std::basic_istream<char, std::char_traits<char> >&, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >
5.84%
(1.39%)
101×
0.64%
7×
0.99%
12×
2.92%
6×
0.75%
4×
1.06%
4×
2.66%
100×
0.70%
100×
1.51%
101×
ld-2.31.so
_dl_check_map_versions
0.64%
(0.35%)
7×
ld-2.31.so
_dl_map_object
1.09%
(0.09%)
13×
???
0x00000000048e7600
2.94%
(0.00%)
22×
???
0x00000000048e87c0
0.75%
(0.00%)
4×
???
0x00000000048e92d0
1.06%
(0.00%)
4×
libc-2.31.so
__strtod_l_internal
2.66%
(1.49%)
100×
???
0x00000000048e63e0
0.76%
(0.01%)
104×
???
0x00000000048e96e0
1.51%
(0.01%)
101×
2.75%
21×
1.03%
3×
0.71%
103×
1.49%
100×
1.60%
625×
0.57%
525×
libstdc++.so.6.0.28
std::locale::locale()
2.75%
(0.01%)
21×
libstdc++.so.6.0.28
std::basic_ios<wchar_t, std::char_traits<wchar_t> >::init(std::basic_streambuf<wchar_t, std::char_traits<wchar_t> >*)
0.73%
(0.00%)
3×
libstdc++.so.6.0.28
std::basic_ios<char, std::char_traits<char> >::init(std::basic_streambuf<char, std::char_traits<char> >*)
1.03%
(0.00%)
3×
libstdc++.so.6.0.28
std::ctype<char> const& std::use_facet<std::ctype<char> >(std::locale const&)
0.71%
(0.08%)
103×
libstdc++.so.6.0.28
std::istream::sentry::sentry(std::istream&, bool)
1.49%
(0.33%)
100×
31.16%
2123×
2.74%
22×
0.70%
4×
0.89%
4×
0.60%
104×
1.49%
100×
libstdc++.so.6.0.28
0x00000000000be670
2.74%
(0.01%)
22×
???
0x00000000048e9540
0.89%
(0.00%)
4×
???
0x00000000048e8ff30
0.83%
(0.01%)
148×
libstdc++.so.6.0.28
std::ctype<wchar_t>::ctype(unsigned long)
0.98%
(0.00%)
0×
libstdc++.so.6.0.28
std::locale::__imp::~impl(unsigned long)
2.75%
(0.03%)
0×
libstdc++.so.6.0.28
__gnu_cxx::stdio_sync_filebuf<char, std::char_traits<char> >::underflow()
2.20%
(0.29%)
827×
libstdc++.so.6.0.28
__gnu_cxx::stdio_sync_filebuf<char, std::char_traits<char> >::uflow()
0.88%
(0.19%)
626×
0.60%
202×
0.11%
101×
2.70%
1×
???
0x00000000048e9e60
0.70%
(0.00%)
4×
0.52%
3×
0.76%
147×
0.98%
1×
1.00%
1×
0.60%
32×
1.21%
827×
0.70%
827×
0.49%
626×
???
0x00000000048e7130
2.70%
(0.00%)
1×
libstdc++.so.6.0.28
std::basic_ios<char, std::char_traits<char> >::_M_cache_locale(std::locale const&)
0.52%
(0.00%)
3×
libstdc++.so.6.0.28
__dynamic_cast
0.76%
(0.33%)
147×
???
0x00000000048e7810
0.98%
(0.00%)
1×
???
0x00000000048e68e0
1.00%
(0.00%)
1×
???
0x00000000048e7e00
0.60%
(0.00%)
32×
???
0x00000000048e78c0
1.21%
(0.08%)
827×
???
0x00000000048ea170
1.19%
(0.11%)
1453×
0.31%
1×
0.37%
1×
2.70%
1×
0.98%
1×
1.00%
1×
0.52%
31×
1.12%
826×
1.03%
1452×
ld-2.31.so
_dl_runtime_resolve_xsave
8.56%
(0.10%)
65×
libstdc++.so.6.0.28
std::locale::_Impl::_M_install_facet(std::locale::id const*, std::locale::facet const*)
0.52%
(0.47%)
31×
libc-2.31.so
__GI___uflow
1.12%
(0.51%)
826×
libc-2.31.so
_IO_getc
1.03%
(1.01%)
1452×
8.46%
65×
ld-2.31.so
_dl_fixup
8.89%
(0.25%)
80×
0.61%
826×
libc-2.31.so
_IO_sputbackc
0.61%
(0.61%)
826×
8.64%
80×
ld-2.31.so
_dl_lookup_symbol_x
69.80%
(36.81%)
2203×
2.99%
2203×
ld-2.31.so
do_lookup_x
52.99%
(22.20%)
2203×
5.31%
3×
7.34%
2188×
???
0x00000000040010c0
3.33%
(0.00%)
17×
ld-2.31.so
check_match
7.34%
(4.57%)
2188×
0.11%
40×
2.77%
2255×
libc-2.31.so
strcmp
3.29%
(3.29%)
2628×
3.33%
17×
ld-2.31.so
calloc
3.33%
(0.01%)
17×
3.32%
17×
???
0x00000000040010a0
3.35%
(0.00%)
40×
3.34%
40×
libc-2.31.so
malloc
3.46%
(0.08%)
61×
3.29%
1×
libc-2.31.so
malloc_hook_ini
3.29%
(0.00%)
1×
3.27%
1×
libc-2.31.so
ptmalloc_init.part.0
3.27%
(0.03%)
1×
2.76%
1×
libc-2.31.so
_dl_addr
2.76%
(2.76%)
1×

1000 hodnot


ld-2.31.so
0x0000000000001100
97.24%
(0.00%)
0×
libstdc++.so.6.0.28
std::ctype<wchar_t>::_M_initialize_ctype()
0.54%
(0.06%)
0×
97.24%
1×
41.72%
1×
55.26%
1×
ld-2.31.so
_dl_start
41.72%
(0.02%)
1×
math_prof
start
55.26%
(0.00%)
1×
41.70%
1×
55.26%
1×
ld-2.31.so
_dl_sysdep_start
41.70%
(0.01%)
1×
libc-2.31.so
(below main)
55.26%
(0.00%)
1×
0.57%
1×
41.11%
1×
2.52%
1×
52.66%
1×
ld-2.31.so
__GI___tunables_init
0.57%
(0.57%)
1×
ld-2.31.so
_dl_main
41.11%
(0.03%)
1×
math_prof
__libc_csu_init
2.52%
(0.00%)
1×
math_prof
main
52.66%
(0.00%)
1×
0.62%
1×
39.98%
7×
2.52%
1×
2.52%
1×
31.66%
1×
5.62%
1×
14.98%
1×
ld-2.31.so
_dl_map_object_deps
0.62%
(0.05%)
1×
ld-2.31.so
_dl_relocate_object
39.98%
(7.65%)
7×
math_prof
_GLOBAL__sub_I__ZN4math8readInputB5cxx11v
2.52%
(0.00%)
1×
math_prof
readInput(std::istream&)
31.66%
(0.35%)
1×
math_prof
stdDeviation(std::vector<double, std::allocator<double> > const&)
5.62%
(0.00%)
1×
math_prof
splitStringsToDoubles(std::vector<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, std::allocator<std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > >
14.98%
(0.65%)
1×
0.54%
12×
2.51%
1×
29.75%
1001×
1.17%
989×
5.03%
1×
14.26%
1000×
??? 0x0000000004001090
0.60%
(0.00%)
13×
??? 0x00000000010a330
2.51%
(0.00%)
1×
??? 0x00000000010a2f0
29.75%
(0.04%)
1001×
std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_construct<char*>(char*, char*, std::forward_iterator_tag)
1.19%
(0.84%)
1000×
math_prof
sumOfSquares(std::vector<double, std::allocator<double> > const&)
5.03%
(0.35%)
1×
??? 0x00000000010a290
14.26%
(0.04%)
1000×
0.60%
13×
2.51%
1×
29.71%
1001×
4.63%
1000×
0.35%
1000×
0.33%
2×
4.26%
1000×
ld-2.31.so
_dl_catch_exception
0.60%
(0.01%)
13×
libstdc++.so.6.0.28
std::ios_base::Init::Init()
2.51%
(0.01%)
1×
libstdc++.so.6.0.28
std::basic_istream<char, std::char_traits<char> >& std::operator>><char, std::char_traits<char>, std::allocator<char> >(std::basic_istream<char, std::char_traits<char> >&, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >&)
29.71%
(7.32%)
1001×
math_prof
f_pow(double, double)
4.63%
(0.73%)
1001×
??? 0x00000000010a2a0
0.68%
(0.04%)
1002×
libc-2.31.so
strtod
14.22%
(0.10%)
1000×
0.52%
12×
1.54%
6×
0.56%
4×
5.56%
1000×
7.01%
1001×
4.35%
1001×
2.55%
1001×
4.12%
1000×
ld-2.31.so
openaux
0.52%
(0.00%)
12×
??? 0x00000000048e7600
1.55%
(0.00%)
22×
??? 0x00000000048e92d0
0.56%
(0.00%)
4×
??? 0x00000000048e63e0
3.80%
(0.04%)
1004×
??? 0x00000000048e96e0
7.01%
(0.04%)
1001×
??? 0x00000000010a300
1.35%
(0.04%)
1002×
??? 0x00000000010a370
2.55%
(0.04%)
1002×
libc-2.31.so
__strtod_l_internal
14.12%
(7.98%)
1000×
0.52%
12×
1.45%
21×
3.53%
1003×
8.43%
8261×
6.97%
1000×
1.31%
1002×
2.51%
1002×
0.53%
1730×
1.74%
968×
2.22%
1551×
0.77%
2366×
ld-2.31.so
_dl_map_object
0.58%
(0.05%)
13×
libstdc++.so.6.0.28
std::locale::locale()
1.45%
(0.01%)
21×
libstdc++.so.6.0.28
std::basic_ios<char, std::char_traits<char> >::init(std::basic_streambuf<char, std::char_traits<char> >*)
0.54%
(0.00%)
3×
libstdc++.so.6.0.28
std::ctype<char> const& std::use_facet<std::ctype<char> >(std::locale const&)
3.53%
(0.41%)
1003×
libstdc++.so.6.0.28
std::istream::sentry(std::istream&, bool)
6.97%
(1.71%)
1000×
libm-2.31.so
fmod
1.31%
(0.27%)
1002×
libm-2.31.so
pow@@GLIBC_2.29
2.51%
(0.39%)
1002×
???
0x00000000048bc4d0
0.53%
(0.07%)
1732×
libc-2.31.so
round_and_return
1.74%
(1.42%)
968×
libc-2.31.so
str_to_mpn.isra.0
2.22%
(2.22%)
1551×
libc-2.31.so
__mpn_lshift
0.77%
(0.77%)
2367×
32.29%
2123×
1.45%
22×
2.96%
1004×
3.00%
5261×
2.74%
2002×
1.94%
1001×
0.64%
1002×
libstdc++.so.6.0.28
0x0000000000be670
1.45%
(0.00%)
22×
libstdc++.so.6.0.28
std::ctype<wchar_t>::ctype(unsigned long)
0.52%
(0.00%)
0×
libstdc++.so.6.0.28
std::locale::_Impl::_Impl(unsigned long)
1.46%
(0.01%)
0×
??? 0x00000000048e8f30
3.08%
(0.04%)
1048×
libstdc++.so.6.0.28
__gnu_cxx::stdio_sync_filebuf<char, std::char_traits<char> >::uflow()
3.58%
(1.02%)
8262×
libstdc++.so.6.0.28
__gnu_cxx::stdio_sync_filebuf<char, std::char_traits<char> >::underflow()
11.16%
(1.52%)
8263×
??? 0x00000000048e8570
1.95%
(0.04%)
1004×
libm-2.31.so
__fmod_finite@GLIBC_2.15
1.04%
(1.04%)
1002×
??? 0x00000000048a3c310
2.12%
(0.04%)
1002×
libc-2.31.so
__memcpy_avx_unaligned_erms
1.28%
(1.28%)
2745×
1.43%
1×
0.52%
1×
0.53%
1×
3.00%
1047×
2.56%
8262×
3.40%
8263×
6.24%
8263×
1.88%
1003×
2.08%
1002×
??? 0x00000000048e7130
1.43%
(0.00%)
1×
??? 0x00000000048e7810
0.52%
(0.00%)
1×
??? 0x00000000048e68e0
0.53%
(0.00%)
1×
libstdc++.so.6.0.28
__dynamic_cast
3.00%
(1.11%)
1047×
??? 0x00000000048ea170
5.96%
(0.59%)
14525×
??? 0x00000000048e78c0
6.24%
(0.34%)
8263×
libstdc++.so.6.0.28
std::ostream::flush()
1.88%
(0.31%)
1003×
libc-2.31.so
__ieee754_pow_fma
2.08%
(2.08%)
1002×
1.43%
1×
0.52%
1×
0.53%
1×
1.86%
1008×
5.34%
14524×
5.89%
8262×
1.57%
1004×
ld-2.31.so
_dl_runtime_resolve_xsavec
4.57%
(0.05%)
65×
libstdc++.so.6.0.28
__vmi_class_type_info::__do_dyncast(long, __cxxabiv1::__class_type_info::__sub_kind, __cxxabiv1::__class_type_info const*, void const*, __cxxabiv1::__class_type_info const*, void const*, __cxxabiv1::__class_type_info::__dyncast_result&)
1.86%
(1.38%)
1008×
libc-2.31.so
getc
5.34%
(5.33%)
14524×
libc-2.31.so
__ungetc
5.89%
(2.69%)
8262×
libstdc++.so.6.0.28
__gnu_cxx::stdio_sync_filebuf<char, std::char_traits<char> >::sync()
1.57%
(0.06%)
1004×
4.51%
65×
3.20%
8262×
1.51%
1004×
ld-2.31.so
_dl_fixup
4.74%
(0.13%)
80×
libc-2.31.so
_IO_switch_to_get_mode
3.20%
(3.20%)
8262×
??? 0x00000000048ea0c0
1.51%
(0.04%)
1007×
4.61%
80×
7.46%
1006×
ld-2.31.so
_dl_lookup_symbol_x
36.69%
(19.43%)
2203×
libc-2.31.so
ffush
1.46%
(1.09%)
1006×
17.46%
2203×
ld-2.31.so
do_lookup_x
17.46%
(11.72%)
2203×
1.80%
3×
3.88%
2188×
??? 0x00000000040010a0
1.81%
(0.00%)
17×
ld-2.31.so
check_match
3.88%
(2.41%)
2188×
1.81%
17×
1.46%
2255×
ld-2.31.so
calloc
1.81%
(0.00%)
17×
libc-2.31.so
strcmp
1.74%
(1.74%)
2628×
1.80%
17×
??? 0x00000000040010a0
1.82%
(0.00%)
40×
1.81%
40×
libc-2.31.so
malloc
1.93%
(0.05%)
67×
1.79%
1×
libc-2.31.so
malloc_hook_ini
1.79%
(0.00%)
1×
1.77%
1×
libc-2.31.so
ptmalloc_init.part.0
1.77%
(0.02%)
1×
1.46%
1×
libc-2.31.so
_dl_addr
1.46%
(1.46%)
1×