Protokol z profilování

1. Úvod

Podle zadání byl vytvořen program pro výpočet směrodatné odchylky. Program načte libovolné množství čísel ze STDIN a podle zadaného vzorce postupně spočítá směrodatnou odchylku, kterou vrátí na STDOUT. Program je vytvořen v jazyce C++ a využívá k výpočtům knihovnu *matlib.h*.

2. Použité nástroje

Profilování bylo provedeno pomocí nástroje *gprof*, kdy byl program přeložen s přepínačem *-pg* a následně se pomocí *gprof* analyzoval výstup *gmon.out*. *Gprof* bylo spouštěno s přepínači *-- brief* a *-- no-graph* pro větší přehlednost.

3. Výsledky

3.1. Pro 10 vstupních hodnot

```
Flat profile:

Each sample counts as 0.01 seconds.
  no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00       45     0.00     0.00  std::isinf(double)
  0.00      0.00     0.00       42     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::base() const
  0.00      0.00     0.00       36     0.00     0.00  std::numeric_limits<int>::max()
  0.00      0.00     0.00       30     0.00     0.00  double* std::__niter_base<double*>(double*)
  0.00      0.00     0.00       25     0.00     0.00  double const& std::forward<double const&>(std::remove_reference<double const&>::type&)
  0.00      0.00     0.00       22     0.00     0.00  std::numeric_limits<double>::max()
  0.00      0.00     0.00       22     0.00     0.00  std::numeric_limits<double>::lowest()
  0.00      0.00     0.00       21     0.00     0.00  add(double, double)
  0.00      0.00     0.00       21     0.00     0.00  std::vector<double, std::allocator<double> >::size() const
  0.00      0.00     0.00       12     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::__normal_iterator(double* const&)
  0.00      0.00     0.00       11     0.00     0.00  power(double, double)
  0.00      0.00     0.00       11     0.00     0.00  bool __gnu_cxx::operator!=<double*, std::vector<double, std::allocator<double> > >(__gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&, __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&)
  0.00      0.00     0.00       11     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_get_Tp_allocator()
  0.00      0.00     0.00       11     0.00     0.00  std::isnan(double)
  0.00      0.00     0.00       10     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::operator++()
  0.00      0.00     0.00       10     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::operator*() const
  0.00      0.00     0.00       10     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_get_Tp_allocator() const
  0.00      0.00     0.00       10     0.00     0.00  std::vector<double, std::allocator<double> >::max_size() const
  0.00      0.00     0.00       10     0.00     0.00  std::vector<double, std::allocator<double> >::_S_max_size(std::allocator<double> const&)
  0.00      0.00     0.00       10     0.00     0.00  std::vector<double, std::allocator<double> >::_S_relocate(double*, double*, double*, std::allocator<double>&)
  0.00      0.00     0.00       10     0.00     0.00  std::vector<double, std::allocator<double> >::push_back(double const&)
  0.00      0.00     0.00       10     0.00     0.00  double* std::__relocate_a<double*, double*, std::allocator<double> >(double*, double*, double*, std::allocator<double>&)
  0.00      0.00     0.00       10     0.00     0.00  std::enable_if<std::__is_bitwise_relocatable<double, void>::value, double*>::type std::__relocate_a_1<double, double>(double*, double*, double*, std::allocator<double>&)
  0.00      0.00     0.00       10     0.00     0.00  unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&)
  0.00      0.00     0.00       10     0.00     0.00  operator new(unsigned long, void*)
  0.00      0.00     0.00        6     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_deallocate(double*, unsigned long)
  0.00      0.00     0.00        6     0.00     0.00  std::vector<double, std::allocator<double> >::end()
  0.00      0.00     0.00        6     0.00     0.00  std::vector<double, std::allocator<double> >::begin()
  0.00      0.00     0.00        5     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::difference_type __gnu_cxx::operator-<double*, std::vector<double, std::allocator<double> > >(__gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&, __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&)
  0.00      0.00     0.00        5     0.00     0.00  std::vector<double, std::allocator<double> >::_M_check_len(unsigned long, char const*) const
  0.00      0.00     0.00        5     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_allocate(unsigned long)
  0.00      0.00     0.00        5     0.00     0.00  std::__new_allocator<double>::deallocate(double*, unsigned long)
  0.00      0.00     0.00        5     0.00     0.00  std::__new_allocator<double>::allocate(unsigned long, void const*)
  0.00      0.00     0.00        5     0.00     0.00  void std::vector<double, std::allocator<double> >::_M_realloc_insert<double const&>(__gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >, double const&)
  0.00      0.00     0.00        5     0.00     0.00  unsigned long const& std::max<unsigned long>(unsigned long const&, unsigned long const&)
  0.00      0.00     0.00        2     0.00     0.00  divide(double, double)
  0.00      0.00     0.00        1     0.00     0.00  root(double, int)
  0.00      0.00     0.00        1     0.00     0.00  multiply(double, double)
  0.00      0.00     0.00        1     0.00     0.00  substract(double, double)
  0.00      0.00     0.00        1     0.00     0.00  void std::_Destroy_aux<true>::__destroy<double*>(double*, double*)
  0.00      0.00     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_impl::_Vector_impl()
  0.00      0.00     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_impl::~_Vector_impl()
  0.00      0.00     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_impl_data::_Vector_impl_data()
  0.00      0.00     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_base()
  0.00      0.00     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::~_Vector_base()
  0.00      0.00     0.00        1     0.00     0.00  std::__new_allocator<double>::~__new_allocator()
  0.00      0.00     0.00        1     0.00     0.00  std::vector<double, std::allocator<double> >::vector()
  0.00      0.00     0.00        1     0.00     0.00  std::vector<double, std::allocator<double> >::~vector()
  0.00      0.00     0.00        1     0.00     0.00  void std::_Destroy<double*>(double*, double*)
```

3.2. Pro 1 000 vstupních hodnot

```
Flat profile:

Each sample counts as 0.01 seconds.
  no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00     4005     0.00     0.00  std::isinf(double)
  0.00      0.00     0.00     2046     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::base() const
  0.00      0.00     0.00     2011     0.00     0.00  double const& std::forward<double const&>(std::remove_reference<double const&>::type&)
  0.00      0.00     0.00     2002     0.00     0.00  std::numeric_limits<double>::max()
  0.00      0.00     0.00     2002     0.00     0.00  std::numeric_limits<double>::lowest()
  0.00      0.00     0.00     2001     0.00     0.00  add(double, double)
  0.00      0.00     0.00     1001     0.00     0.00  power(double, double)
  0.00      0.00     0.00     1001     0.00     0.00  bool __gnu_cxx::operator!=<double*, std::vector<double, std::allocator<double> > >(__gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&, __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&)
  0.00      0.00     0.00     1001     0.00     0.00  std::isnan(double)
  0.00      0.00     0.00     1000     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::operator++()
  0.00      0.00     0.00     1000     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::operator*() const
  0.00      0.00     0.00     1000     0.00     0.00  std::vector<double, std::allocator<double> >::push_back(double const&)
  0.00      0.00     0.00     1000     0.00     0.00  operator new(unsigned long, void*)
  0.00      0.00     0.00       66     0.00     0.00  double* std::__niter_base<double*>(double*)
  0.00      0.00     0.00       45     0.00     0.00  std::vector<double, std::allocator<double> >::size() const
  0.00      0.00     0.00       36     0.00     0.00  std::numeric_limits<int>::max()
  0.00      0.00     0.00       24     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::__normal_iterator(double* const&)
  0.00      0.00     0.00       23     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_get_Tp_allocator()
  0.00      0.00     0.00       22     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_get_Tp_allocator() const
  0.00      0.00     0.00       22     0.00     0.00  std::vector<double, std::allocator<double> >::max_size() const
  0.00      0.00     0.00       22     0.00     0.00  std::vector<double, std::allocator<double> >::_S_max_size(std::allocator<double> const&)
  0.00      0.00     0.00       22     0.00     0.00  std::vector<double, std::allocator<double> >::_S_relocate(double*, double*, double*, std::allocator<double>&)
  0.00      0.00     0.00       22     0.00     0.00  double* std::__relocate_a<double*, double*, std::allocator<double> >(double*, double*, double*, std::allocator<double>&)
  0.00      0.00     0.00       22     0.00     0.00  std::enable_if<std::__is_bitwise_relocatable<double, void>::value, double*>::type std::__relocate_a_1<double, double>(double*, double*, double*, std::allocator<double>&)
  0.00      0.00     0.00       22     0.00     0.00  unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&)
  0.00      0.00     0.00       12     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_deallocate(double*, unsigned long)
  0.00      0.00     0.00       12     0.00     0.00  std::vector<double, std::allocator<double> >::end()
  0.00      0.00     0.00       12     0.00     0.00  std::vector<double, std::allocator<double> >::begin()
  0.00      0.00     0.00       11     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::difference_type __gnu_cxx::operator-<double*, std::vector<double, std::allocator<double> > >(__gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&, __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&)
  0.00      0.00     0.00       11     0.00     0.00  std::vector<double, std::allocator<double> >::_M_check_len(unsigned long, char const*) const
  0.00      0.00     0.00       11     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_allocate(unsigned long)
  0.00      0.00     0.00       11     0.00     0.00  std::__new_allocator<double>::deallocate(double*, unsigned long)
  0.00      0.00     0.00       11     0.00     0.00  std::__new_allocator<double>::allocate(unsigned long, void const*)
  0.00      0.00     0.00       11     0.00     0.00  void std::vector<double, std::allocator<double> >::_M_realloc_insert<double const&>(__gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >, double const&)
  0.00      0.00     0.00       11     0.00     0.00  unsigned long const& std::max<unsigned long>(unsigned long const&, unsigned long const&)
  0.00      0.00     0.00        2     0.00     0.00  divide(double, double)
  0.00      0.00     0.00        1     0.00     0.00  root(double, int)
  0.00      0.00     0.00        1     0.00     0.00  multiply(double, double)
  0.00      0.00     0.00        1     0.00     0.00  substract(double, double)
  0.00      0.00     0.00        1     0.00     0.00  void std::_Destroy_aux<true>::__destroy<double*>(double*, double*)
  0.00      0.00     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_impl::_Vector_impl()
  0.00      0.00     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_impl::~_Vector_impl()
  0.00      0.00     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_impl_data::_Vector_impl_data()
  0.00      0.00     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_base()
  0.00      0.00     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::~_Vector_base()
  0.00      0.00     0.00        1     0.00     0.00  std::__new_allocator<double>::~__new_allocator()
  0.00      0.00     0.00        1     0.00     0.00  std::vector<double, std::allocator<double> >::vector()
  0.00      0.00     0.00        1     0.00     0.00  std::vector<double, std::allocator<double> >::~vector()
  0.00      0.00     0.00        1     0.00     0.00  void std::_Destroy<double*>(double*, double*)
```

3.3. Pro 1 000 000 vstupních hodnot

```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  ns/call  ns/call  name
33.33      0.02     0.02  2000002    10.00    10.00  std::numeric_limits<double>::max()
33.33      0.04     0.02  2000001    10.00    22.50  add(double, double)
16.67      0.05     0.01  4000005     2.50     2.50  std::isinf(double)
16.67      0.06     0.01                             main
 0.00      0.06     0.00  2000006     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::base() const
 0.00      0.06     0.00  2000021     0.00     0.00  double const& std::forward<double const&>(std::remove_reference<double const&>::type&)
 0.00      0.06     0.00  2000002     0.00     0.00  std::numeric_limits<double>::lowest()
 0.00      0.06     0.00  1000001     0.00     5.00  power(double, double)
 0.00      0.06     0.00  1000001     0.00     0.00  bool __gnu_cxx::operator!=<double*, std::vector<double, std::allocator<double> > >(__gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&, __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&)
 0.00      0.06     0.00  1000001     0.00     0.00  std::isnan(double)
 0.00      0.06     0.00  1000000     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::operator++()
 0.00      0.06     0.00  1000000     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::operator*() const
 0.00      0.06     0.00  1000000     0.00     0.00  std::vector<double, std::allocator<double> >::push_back(double const&)
 0.00      0.06     0.00  1000000     0.00     0.00  operator new(unsigned long, void*)
 0.00      0.06     0.00      126     0.00     0.00  double* std::__niter_base<double*>(double*)
 0.00      0.06     0.00       85     0.00     0.00  std::vector<double, std::allocator<double> >::size() const
 0.00      0.06     0.00       44     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::__normal_iterator(double* const&)
 0.00      0.06     0.00       43     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_get_Tp_allocator()
 0.00      0.06     0.00       42     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_get_Tp_allocator() const
 0.00      0.06     0.00       42     0.00     0.00  std::vector<double, std::allocator<double> >::max_size() const
 0.00      0.06     0.00       42     0.00     0.00  std::vector<double, std::allocator<double> >::_S_max_size(std::allocator<double> const&)
 0.00      0.06     0.00       42     0.00     0.00  std::vector<double, std::allocator<double> >::_S_relocate(double*, double*, double*, std::allocator<double>&)
 0.00      0.06     0.00       42     0.00     0.00  double* std::__relocate_a<double*, double*, std::allocator<double> >(double*, double*, double*, std::allocator<double>&)
 0.00      0.06     0.00       42     0.00     0.00  std::enable_if<std::__is_bitwise_relocatable<double, void>::value, double*>::type std::__relocate_a_1<double, double>(double*, double*, double*, std::allocator<double>&)
 0.00      0.06     0.00       42     0.00     0.00  unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&)
 0.00      0.06     0.00       36     0.00     0.00  std::numeric_limits<int>::max()
 0.00      0.06     0.00       22     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_deallocate(double*, unsigned long)
 0.00      0.06     0.00       22     0.00     0.00  std::vector<double, std::allocator<double> >::end()
 0.00      0.06     0.00       22     0.00     0.00  std::vector<double, std::allocator<double> >::begin()
 0.00      0.06     0.00       21     0.00     0.00  __gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >::difference_type __gnu_cxx::operator-<double*, std::vector<double, std::allocator<double> > >(__gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > > const&,
 0.00      0.06     0.00       21     0.00     0.00  std::vector<double, std::allocator<double> >::_M_check_len(unsigned long, char const*) const
 0.00      0.06     0.00       21     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_M_allocate(unsigned long)
 0.00      0.06     0.00       21     0.00     0.00  std::__new_allocator<double>::deallocate(double*, unsigned long)
 0.00      0.06     0.00       21     0.00     0.00  std::__new_allocator<double>::allocate(unsigned long, void const*)
 0.00      0.06     0.00       21     0.00     0.00  void std::vector<double, std::allocator<double> >::_M_realloc_insert<double const&>(__gnu_cxx::__normal_iterator<double*, std::vector<double, std::allocator<double> > >, double const&)
 0.00      0.06     0.00       21     0.00     0.00  unsigned long const& std::max<unsigned long>(unsigned long const&, unsigned long const&)
 0.00      0.06     0.00        2     0.00     2.50  divide(double, double)
 0.00      0.06     0.00        1     0.00     0.00  root(double, int)
 0.00      0.06     0.00        1     0.00    10.00  multiply(double, double)
 0.00      0.06     0.00        1     0.00    22.50  substract(double, double)
 0.00      0.06     0.00        1     0.00     0.00  void std::_Destroy_aux<true>::__destroy<double*>(double*, double*)
 0.00      0.06     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_impl::_Vector_impl()
 0.00      0.06     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_impl::~_Vector_impl()
 0.00      0.06     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_impl_data::_Vector_impl_data()
 0.00      0.06     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::_Vector_base()
 0.00      0.06     0.00        1     0.00     0.00  std::_Vector_base<double, std::allocator<double> >::~_Vector_base()
 0.00      0.06     0.00        1     0.00     0.00  std::__new_allocator<double>::~__new_allocator()
 0.00      0.06     0.00        1     0.00     0.00  std::vector<double, std::allocator<double> >::vector()
 0.00      0.06     0.00        1     0.00     0.00  std::vector<double, std::allocator<double> >::~vector()
 0.00      0.06     0.00        1     0.00     0.00  void std::_Destroy<double*>(double*, double*)
```

4. Závěr

Z výsledků jednotlivých běhů programu s různým počtem vstupních hodnot je vidět, že nejčastěji jsou volány funkce *add()* a *power()*. Volání funkce *add()* je způsobeno tím, že dle vzorce je potřeba spočítat součet všech hodnot a pak ještě jednou spočítat součet druhých mocnin vstupních hodnot. Funkce *power()* je volána pro každou vstupní hodnotu a jednou pro průměr. Z toho poznatku by tedy při optimalizaci bylo vhodné zaměřit se na tyto dvě funkce.