Revize 4

C2115 Praktický úvod do superpočítání

12. lekce / Modul 5

Petr Kulhánek

kulhanek@chemi.muni.cz

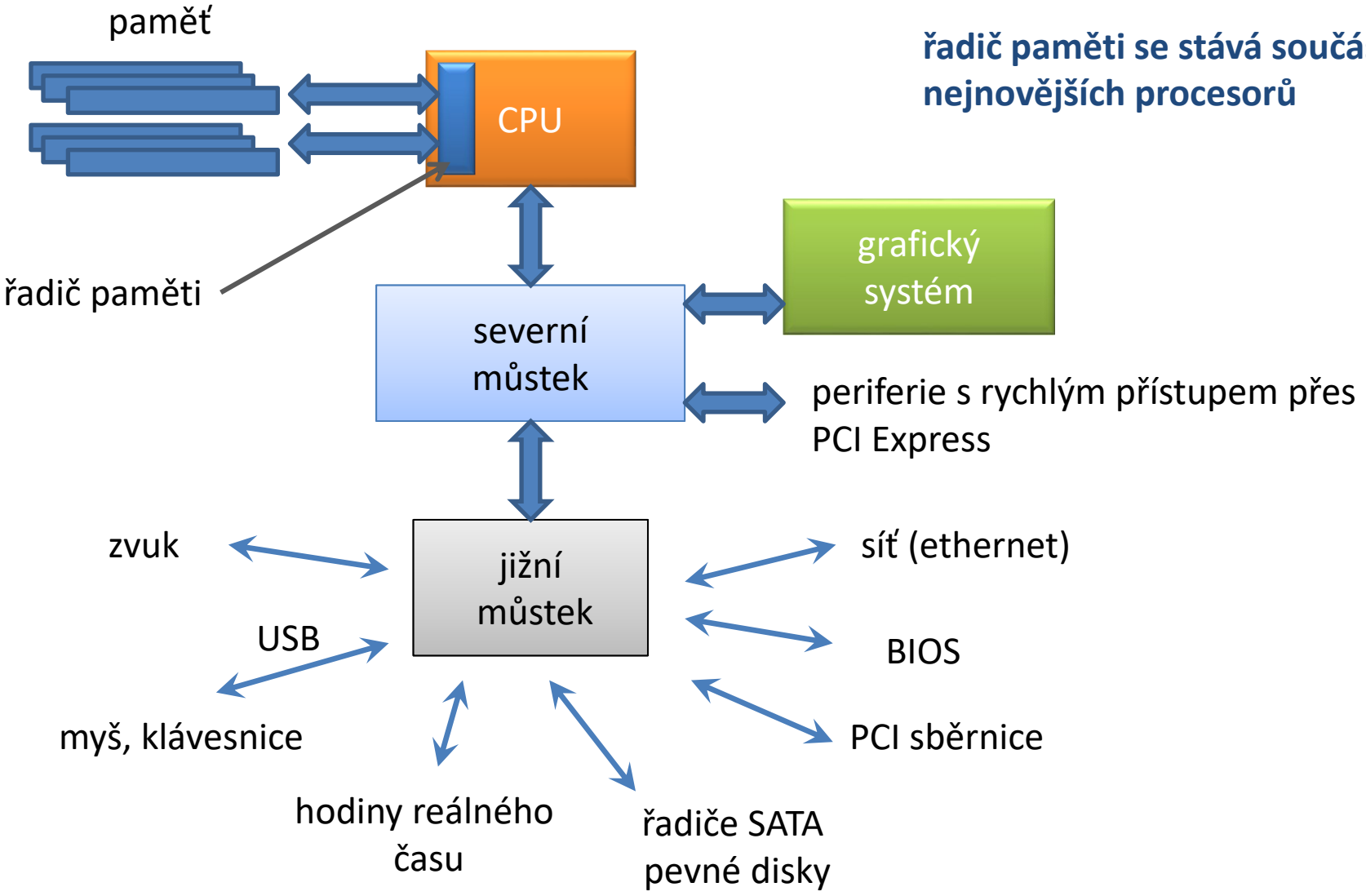
Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta,
Masarykova univerzita, Kotlářská 2, CZ-61137 Brno

Násobení matic

vs

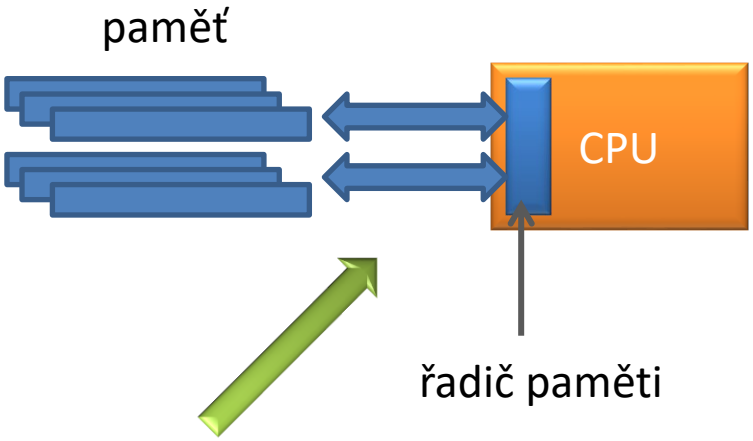
Architektura počítače

Architektura, celkový pohled



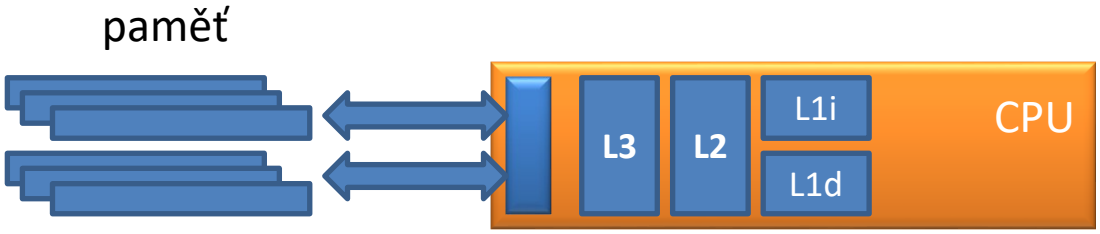
řadič paměti se stává součástí nejnovějších procesorů

Architektura, úzké hrdlo



úzké hrdlo: rychlost přenosu dat mezi pamětí a CPU je pomalejší než rychlost s jakou je CPU data schopno zpracovávat

Hierarchický model paměti

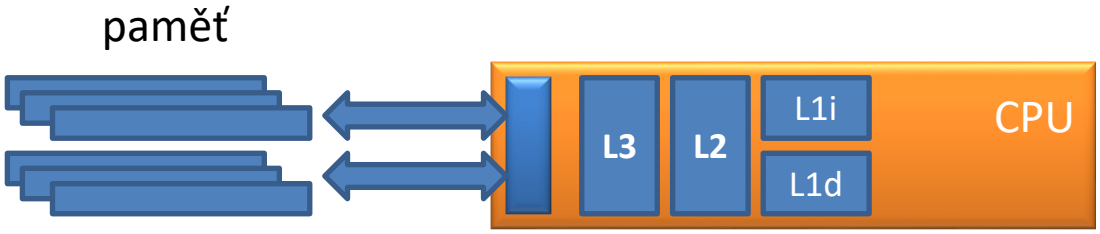


rychlá mezipaměť (cache), různé úrovně s různými rychlostmi

wolf21 – přenosové rychlosti (memtest86+, http://www.memtest.org/)

Typ	Velikost	Rychlost
L1	32kB	89 GB/s
L2	256 kB	35 GB/s
L3	8192 kB	24 GB/s
paměť	8192 MB	12 GB/s

Hierarchický model paměti



rychlá mezipaměť (cache), různé úrovně s různými rychlostmi

wolf21 – přenosové rychlosti (memtest86+, http://www.memtest.org/)

Typ	Velikost	Rychlost
L1	32kB	89 GB/s
L2	256 kB	35 GB/s
L3	8192 kB	24 GB/s
paměť	8192 MB	12 GB/s

Jakmile velikost problému přesáhne velikost mezi paměti CPU, **rychlost určujícím krokem** se stává rychlost přenosu dat mezi fyzickou pamětí a CPU.

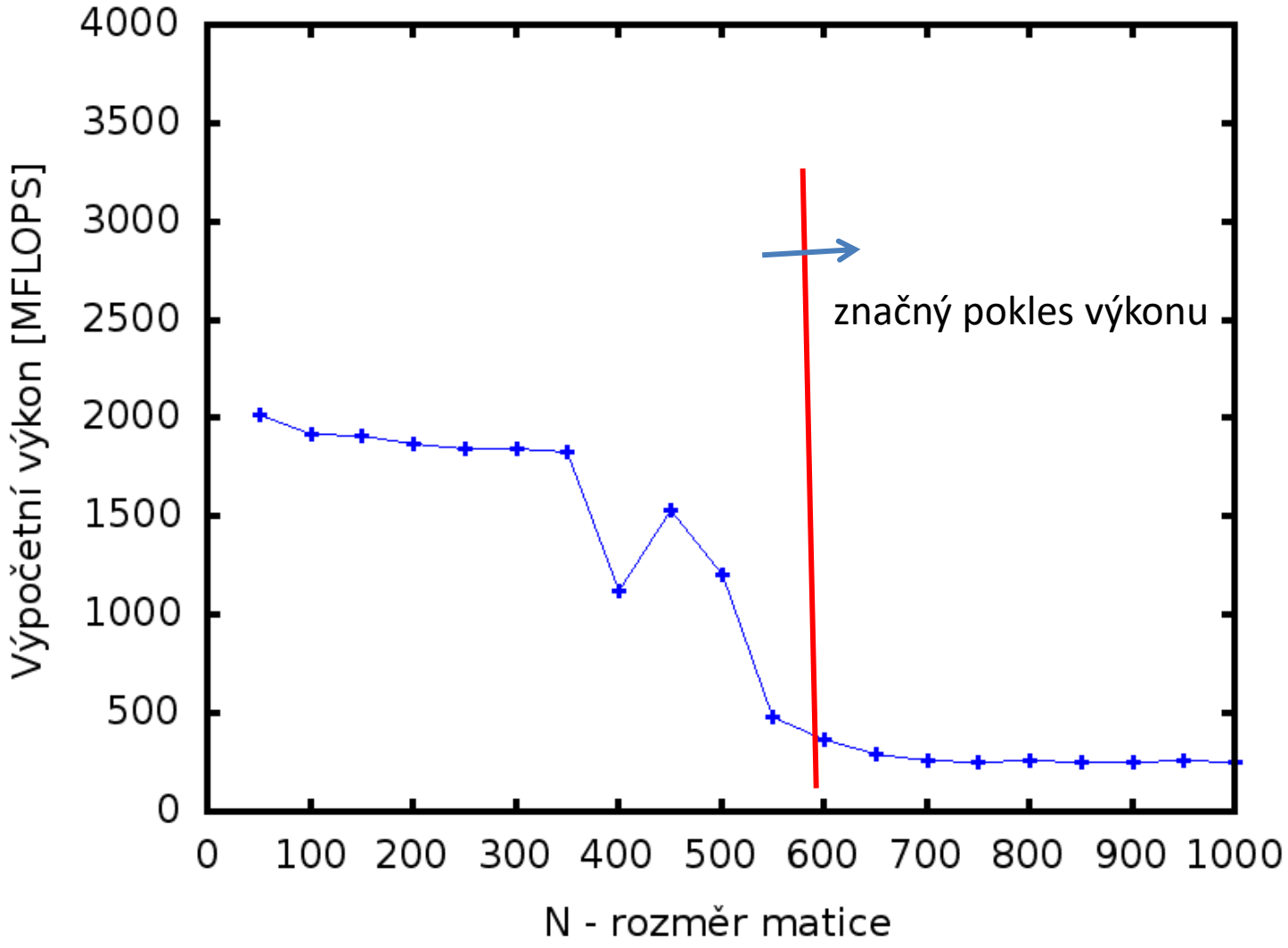
N=600

600x600x3x8 = 8437 kB

A,B,C double precision

Výsledky

wolf21



Knihovny pro lineární algebru

BLAS

The BLAS (**Basic Linear Algebra Subprograms**) are routines that provide standard building blocks for performing basic vector and matrix operations. The Level 1 BLAS perform scalar, vector and vector-vector operations, the Level 2 BLAS perform matrix-vector operations, and the Level 3 BLAS perform matrix-matrix operations. Because the BLAS are efficient, portable, and widely available, they are commonly used in the development of high quality linear algebra software, LAPACK for example.

LAPACK

LAPACK is written in Fortran 90 and provides routines for solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. The associated matrix factorizations (LU, Cholesky, QR, SVD, Schur, generalized Schur) are also provided, as are related computations such as reordering of the Schur factorizations and estimating condition numbers. Dense and banded matrices are handled, but not general sparse matrices. In all areas, similar functionality is provided for real and complex matrices, in both single and double precision.

http://netlib.org

Optimalizované knihovny

Optimalizované knihovny BLAS a LAPACK

- optimalizované dodavatelem hardware
- ATLAS http://math-atlas.sourceforge.net/
- MKL http://software.**intel.**com/en-us/intel-mkl
- ACML http://developer.**amd**.com/tools/cpu-development/amd-core-math-library-acml/
- cuBLAS https://developer.nvidia.com/cublas

Optimalizované knihovny FFT (Fast Fourier Transform)

- optimalizované dodavatelem hardware
- MKL http://software.**intel.**com/en-us/intel-mkl
- ACML http://developer.**amd**.com/tools/cpu-development/amd-core-math-library-acml/
- FFTW http://www.fftw.org/
- cuFFT https://developer.nvidia.com/cufft

Násobení matic pomocí BLAS - dp

```
subroutine mult_matrices_blas(A,B,C)

  implicit none
  double precision    :: A(:,:)
  double precision    :: B(:,:)
  double precision    :: C(:,:)
!-----------------------------------------------------

  if( size(A,2) .ne. size(B,1) ) then
    stop 'Error: Illegal shape of A and B matrices!'
  end if

  call dgemm('N','N',size(A,1),size(B,2),size(A,2),1.0d0, &
             A,size(A,1),B,size(B,1),0.0d0,C,size(C,1))

end subroutine mult_matrices_blas
```

F77 rozhraní BLAS knihovny neobsahuje informace o typech argumentů. Programátor musí zadat všechny argumenty ve správném pořadí a typu!!!!

Kompilace:

```
$ gfortran -O3 mult_mat_blas_dp.f90 -o mult_mat_blas_dp -lblas
```

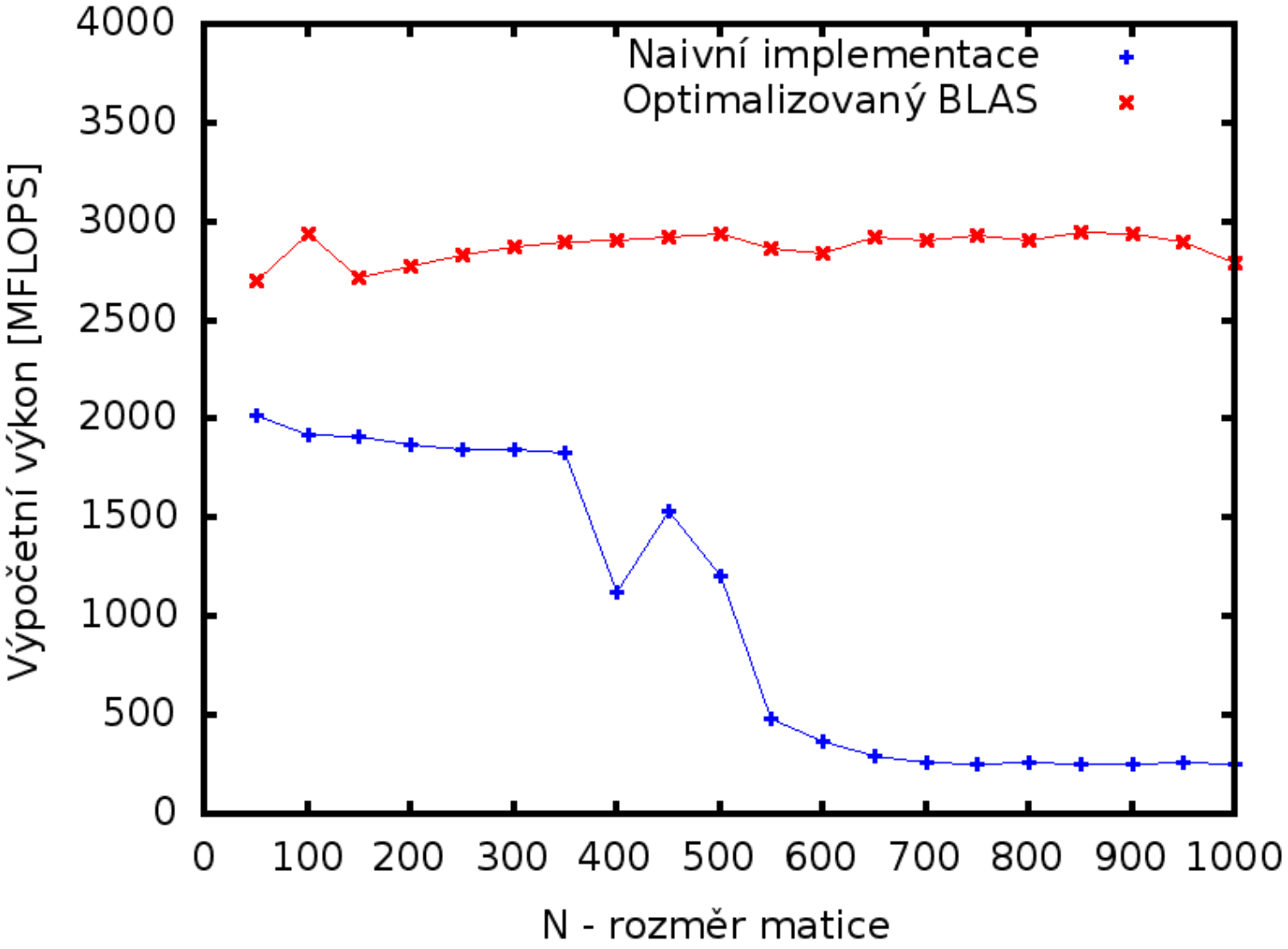
Násobení matic pomocí BLAS - sp

```
subroutine mult_matrices_blas(A,B,C)

  implicit none
  real(4)     :: A(:,:)
  real(4)     :: B(:,:)
  real(4)     :: C(:,:)
!-----------------------------------------------------

  if( size(A,2) .ne. size(B,1) ) then
    stop 'Error: Illegal shape of A and B matrices!'
  end if

  call sgemm('N','N',size(A,1),size(B,2),size(A,2),1.0, &
             A,size(A,1),B,size(B,1),0.0,C,size(C,1))

end subroutine mult_matrices_blas
```

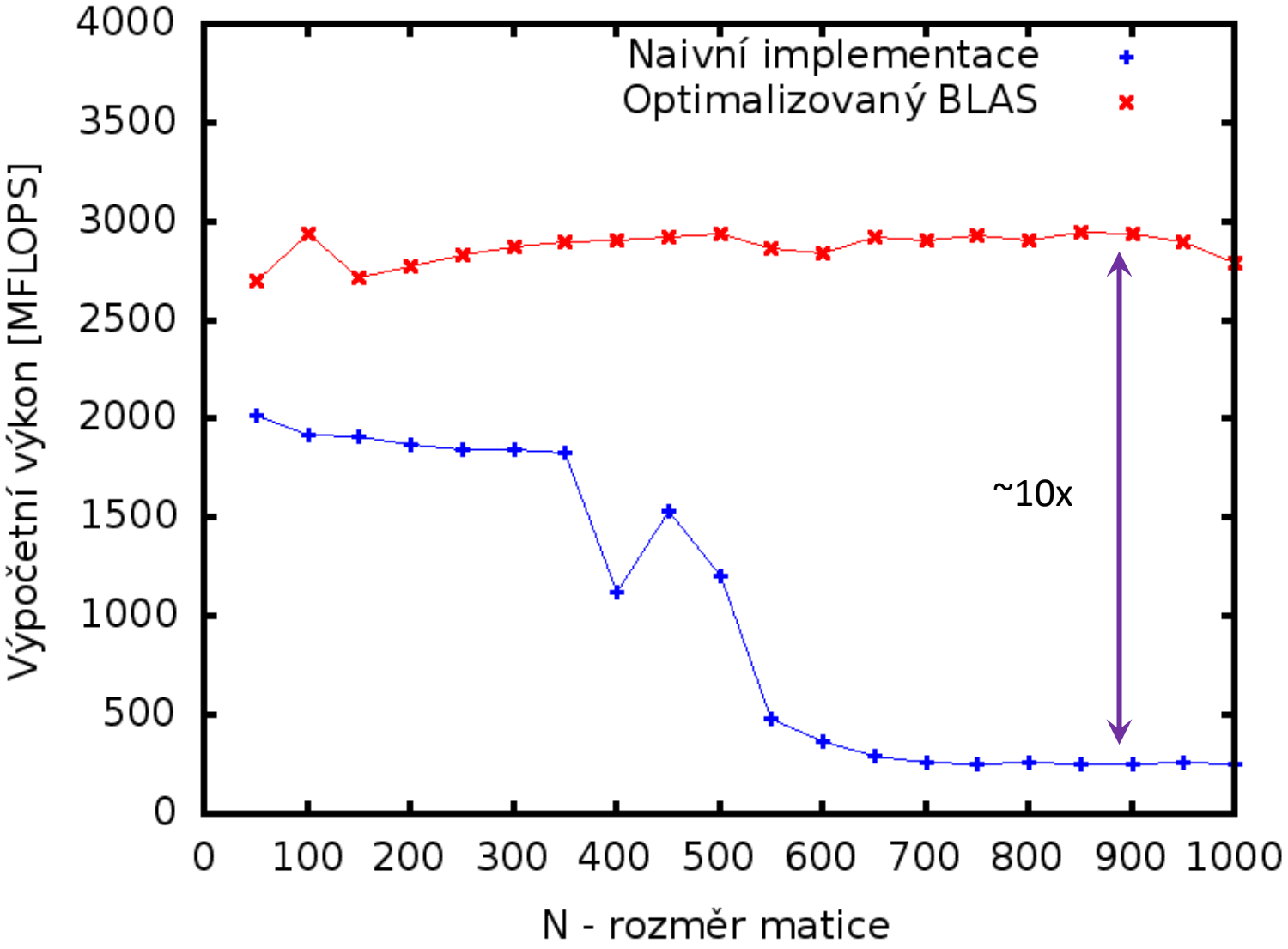
Kompilace:

```
$ gfortran -O3 mult_mat_blas_sp.f90 -o mult_mat_blas_sp -lblas
```

Naivní *vs* optimalizované řešení



Naivní *vs* optimalizované řešení



Cvičení M5.1

Zdrojové kódy:
/home/kulhanek/Documents/C2115/code/matrix

1. Zkompilujte program **mult_mat_blas_dp.f90** kompilátorem **gfortran**, použijte **-O3** optimalizaci.
2. Program spusťte a získanou závislost výpočetního výkonu v závislosti na rozměru matice zobrazte ve formě grafu (použijte interaktivní režim programu gnuplot).
3. Určete výpočetní výkon pro optimalizační úrovně **-O3** a **-O0**. Získané závislosti zobrazte v jednom grafu. Graf vložte do protokolu. Nezapomeňte uvést typ CPU (příkaz lscpu).
4. Srovnejte výpočetní výkon pro přístup **naive** a **blas** v optimalizované verzi (-O3). Získané závislosti zobrazte v jednom grafu. Graf vložte do protokolu. Nezapomeňte uvést typ CPU (příkaz lscpu).
5. Diskutujte získané výsledky.

Kompilace:

```
$ gfortran -O3 mult_mat_blas_dp.f90 -o mult_mat_blas_dp -lblas
```