Katedra
aplikované matematiky

IT4Innovations
national01$#&0
supercomputing
center@#01%101

Od lineárních rovnic k extrémním superpočítačovým úlohám

Vít Vondrák

Od lineárních rovnic k extrémním superpočítačovým úlohám

Vít Vondrák

+

David Horák a jeho **Permoník©®™**

Osnova přednášky

- *Zobrazování čísel v počítači*
- *Základy lineární algebry*
  - Vektory, matice a operace s nimi
  - Výpočetní náročnost operací
- *Basic Linear Algebra Subprograms*
- *Blokové matice a paralelizace operací*
  - Efektivita a škálovatelnost operací
- *Řešení soustav lineárních rovnic*
  - Linpack test a výkon počítačů
- *Závěr*

Zobrazení čísel v počítači

- *1 bit = hodnota 0 nebo 1*
- *1Byte (1B) = 8 bitů*
- *Dvojková číselná soustava*

$$100101_2 = (-1)^1 \cdot (0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0) = -5_{10}$$

Pozice	7	6	5	4	3	2	1	0
Hodnota	1	0	0	0	0	1	0	1

- *1B zobrazí -128 .. 127, 0 .. 255 bez znaménka*
- *2B zobrazí -32 768 ..32 767, 0..65 565 bez znaménka*
- *32bit architektura: integer = 4B (až 4 294 967 295)*
- *64bit architektura: long = 8B (až 18 446 744 073 709 551 615)*

Jak se zobrazí racionální nebo reálné číslo?

Systém s pohyblivou řádovou čárkou

$$25{,}167 = 0{,}25167 \cdot 10^{2} =$$

$$= (-1)^{0} \cdot (2 \cdot 10^{-1} + 5 \cdot 10^{-2} + 1 \cdot 10^{-3} + 6 \cdot 10^{-4} + 7 \cdot 10^{-5}) \cdot 10^{2}$$

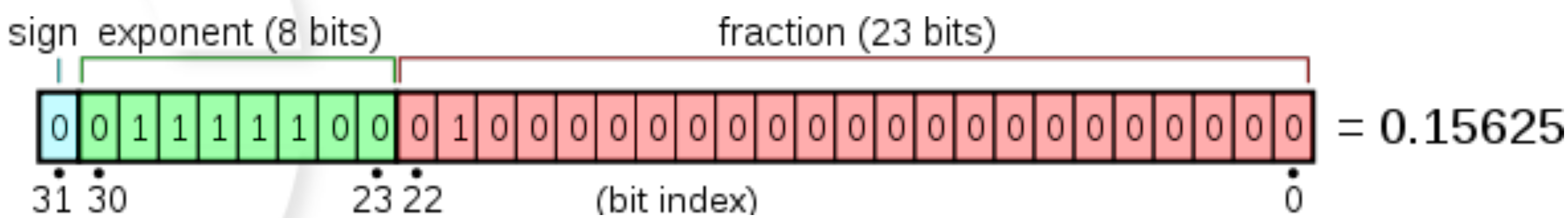
- *b=10: základ systému (2,10,16)*
- *a=[2,5,1,6,7]: mantisa (0..9)*
- *e=2: exponent*
- *+ znaménko*

- *Pohyblivá řádová čárka = Floating point (eng.)*
- *Čísla uložená v tomto systému jsou racionální*
  - Ryze reálná čísla nelze přesně reprezentovat
  - Dokonce ani některá racionální čísla (⅓) nelze přesně reprezentovat

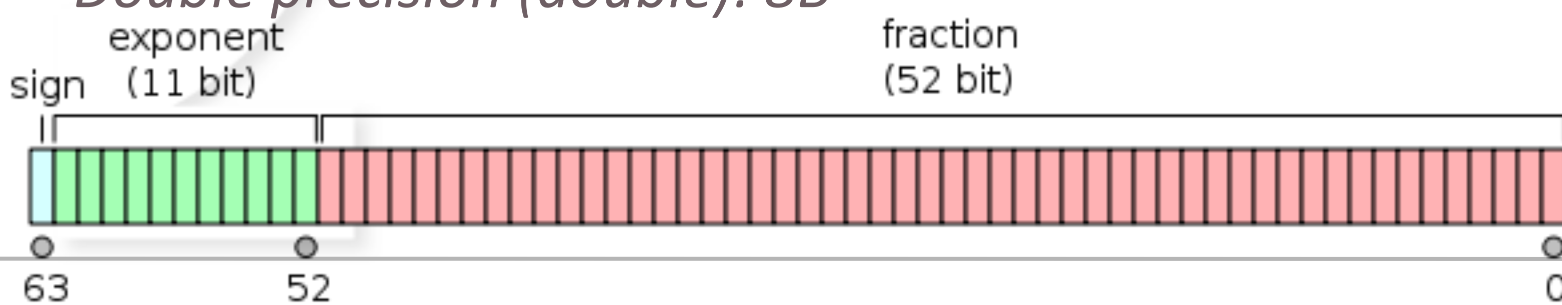
Zobrazení v počítači

$$\left(1+\sum_{k=1}^{p-1} a_k \cdot 2^{-k}\right)\cdot 2^e = (1+1\cdot 2^{-1}+0\cdot 2^{-2}+1\cdot 2^{-3}+1\cdot 2^{-4}+0\cdot 2^{-5})\cdot 2^1 = 3{,}375$$

- *a=[1,0,1,1,0] mantisa (p=6), e=1 exponent*
- *Single precision (float): 4B*



- *Double precision (double): 8B*



Floating point operation - FLOP

- *Výpočetní náročnost– počet operací v pohyblivé řádové čárce FLOPs*
- *Výpočetní výkon počítače – počet operací v pohyblivé řádové čárce za sekundu*
  - FLOPS = Floating point Operations Per Second (FLOP/s)
- *Teoretický výkon:*

$$FLOPS = \# cores \times frequency \times \frac{FLOPs}{cycle}$$

- *#cores … počet jader procesoru, např. 2*
- *Frequency … takt procesoru, např. 1.8GHz*
- *FLOPs/cycle … počet instrukcí za sekundu, typicky 4 až 8.*

Aritmetické vektory

- *Aritmetický vektor = uspořádaná n-tice reálných čísel*

$$\mathbf{v} = \left[v_1, v_2, \ldots, v_n \right]$$

- *n se nazývá dimenze vektoru, n=dim* **v**
- $[\mathbf{v}]_i$ *značí i-tou složku vektoru* **v**

- Příklady:

$$\mathbf{x} = [1, 3, -2.5, 300], \quad \dim \mathbf{x} = 4$$

$$\mathbf{o} = \left[0, 0, \ldots, 0\right], \quad \mathbf{1} = \left[1, 1, \ldots, 1\right], \quad \mathbf{e}_i = \left[0, \ldots, 1, \ldots, 0\right]$$

Operace s aritmetickými vektory

- *Násobení aritmetického vektoru skalárem*

$$[\alpha \mathbf{v}]_i = \alpha [\mathbf{v}]_i, \qquad \alpha \mathbf{v} = \left[\alpha v_1, \alpha v_2, \ldots, \alpha v_n\right]$$

- *Výpočetní náročnost: f(n)=n*

- *Sčítání aritmetických vektorů stejné dimenze*

$$[\mathbf{u} + \mathbf{v}]_i = [\mathbf{u}]_i + [\mathbf{v}]_i, \qquad \mathbf{u} + \mathbf{v} = \left[u_1 + v_1, u_2 + v_2, \ldots, u_n + v_n\right]$$

- *Výpočetní náročnost: f(n)=n*

- **Asymptotická složitost:** *O(N) … f(n)=A+B.n*

Operace s aritmetickými vektory

- *Skalární součin dvou vektorů stejné dimenze*

$$\mathbf{u} \cdot \mathbf{v} = u_1 \cdot v_1 + u_2 \cdot v_2 + \ldots + u_n \cdot v_n$$

- *Výpočetní náročnost: f(n)=2n-1, tj. O(n)*
- *Norma vektoru*

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2} = \sqrt{v \cdot v}$$

- *Výpočetní náročnost: f(n)=2n, tj. O(n)*

Výkonnostní test

- *Operace násobení skalárem + sčítání vektorů*
- Implementace: **y** <- α**x** + **y**
- *Výpočetní náročnost: f(n)=2n, tj. O(n)*

```
void daxpy(int n,double alpha, double *x, double *y) {

    for(int i=0;i<n;i++)
        y[i] = alpha*x[i] + y[i];
}
```

Testováno na MacBook Air, Intel i7, 1.8GHz

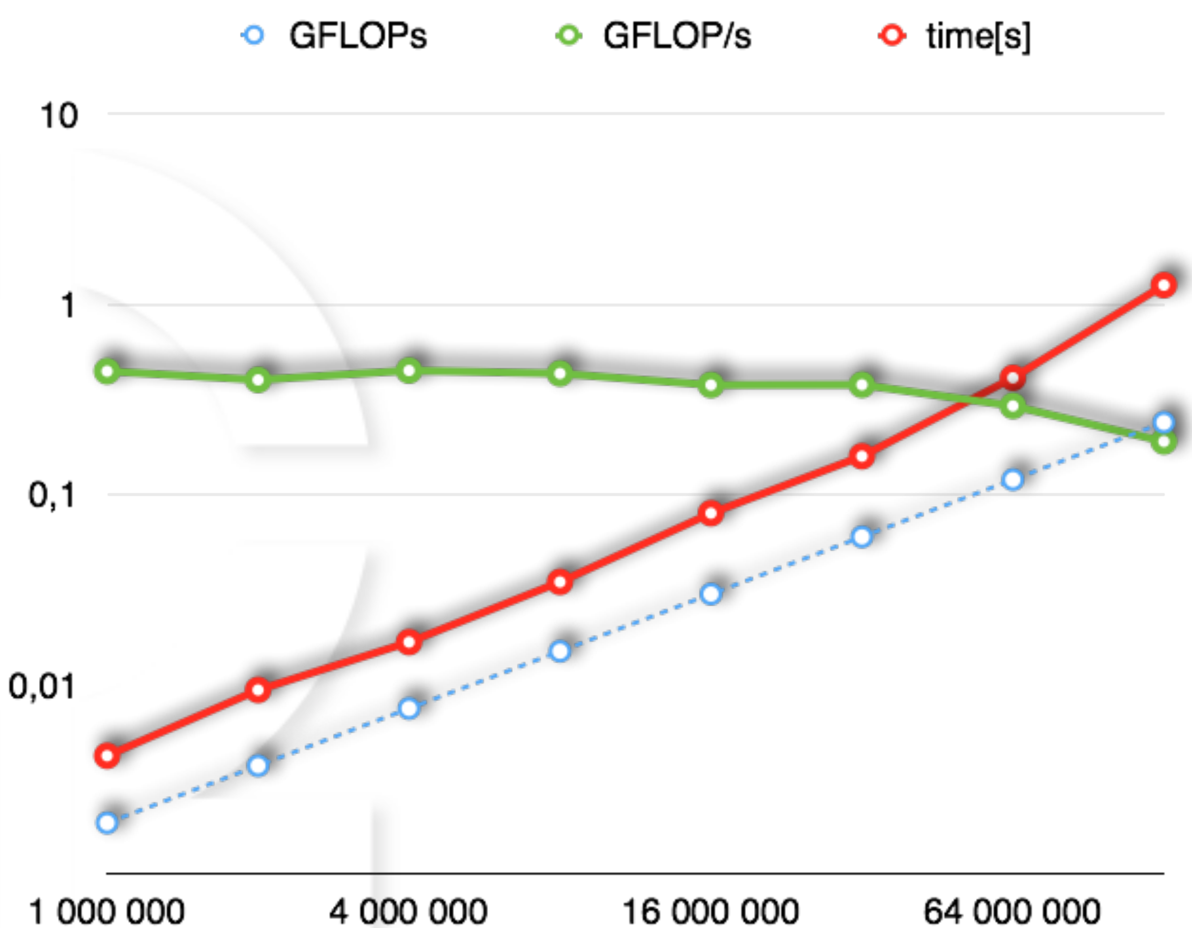
Výsledky testu procedury DAXPY

VV_DAXPY	GFLOPs	GFLOP/s	time[s]
1 000 000	0,00186265	0,443569	0,00419922
2 000 000	0,00372529	0,399445	0,00932617
4 000 000	0,00745058	0,447472	0,0166504
8 000 000	0,0149012	0,431649	0,0345215
16 000 000	0,0298023	0,37537	0,0793945
32 000 000	0,0596046	0,376412	0,15835
64 000 000	0,119209	0,291303	0,409229
128 000 000	0,238419	0,189491	1,2582

Předpona	FLOPS
yotta	10^{24}
zetta	10^{21}
exa	10^{18}
peta	10^{15}
tera	10^{12}
giga	10^{9}
mega	10^{6}
kilo	10^{3}

$$\frac{f(2n)}{f(n)} = \frac{2(2n)}{2n} = 2$$

Výsledky testu procedury DAXPY



Předpona	FLOPS
yotta	10^{24}
zetta	10^{21}
exa	10^{18}
peta	10^{15}
tera	10^{12}
giga	10^{9}
mega	10^{6}
kilo	10^{3}

Matice

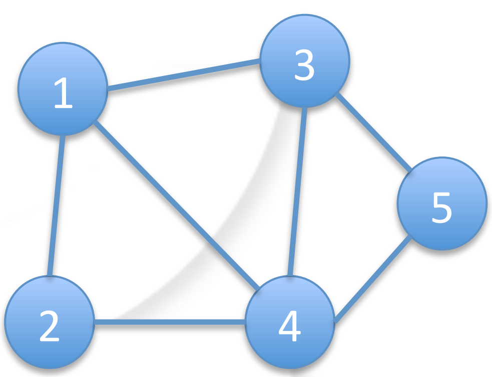
- *Matice = množina m x n reálných čísel uspořádaných do m řádků a n sloupců*

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^A \\ \mathbf{r}_2^A \\ \vdots \\ \mathbf{r}_m^A \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1^A & \mathbf{s}_2^A & \ldots & \mathbf{s}_n^A \end{bmatrix}$$

- $[\mathbf{A}]_{i,j}$ *značí prvek v i-tém řádku a j-tém sloupci*
- $\mathbf{r}_i^A = [a_{i1}, a_{i2}, \ldots, a_{in}]$ označuje *i*-tý řádek
- $\mathbf{s}_j^A$ značí *j*-tý sloupec

Příklady matic

$$\mathbf{A} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & 1 & -1 & 2 \\ 0 & 1 & 1 & 2 \\ \frac{1}{3} & \pi & \sqrt{2} & 5 \end{bmatrix}, \quad \mathbf{O} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$



$$\Longleftrightarrow \quad \mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Operace s maticemi

- *Násobení matice skalárem*

$$[\alpha\mathbf{A}]_{ij} = \alpha[\mathbf{A}]_{ij}$$

$$\alpha\mathbf{A} = \begin{bmatrix} \alpha a_{11} & \alpha a_{12} & \cdots & \alpha a_{1n} \\ \alpha a_{21} & \alpha a_{22} & \cdots & \alpha a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha a_{m1} & \alpha a_{m2} & \cdots & \alpha a_{mn} \end{bmatrix} = \begin{bmatrix} \alpha\mathbf{r}_1^A \\ \alpha\mathbf{r}_2^A \\ \vdots \\ \alpha\mathbf{r}_m^A \end{bmatrix} = \begin{bmatrix} \alpha\mathbf{s}_1^A, & \alpha\mathbf{s}_2^A, & \ldots, & \alpha\mathbf{s}_n^A \end{bmatrix}$$

- *Výpočetní náročnost při m=n: $f(n)=n^2$, tj. $O(n^2)$*

Operace s maticemi

- *Sčítání matic stejného typu*

$$[\mathbf{A}+\mathbf{B}]_{ij} = [\mathbf{A}]_{ij} + [\mathbf{B}]_{ij}$$

$$\mathbf{A}+\mathbf{B} = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \cdots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22} & \cdots & a_{2n}+b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & \cdots & a_{mn}+b_{mn} \end{bmatrix}$$

- *Výpočetní náročnost při m=n: $f(n)=n^2$, tj. $O(n^2)$*

Operace s maticemi

- *Transponování matic*

$$[\mathbf{A}^T]_{ij} = [\mathbf{A}]_{ji}$$

$$\mathbf{A}^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} \left(\mathbf{r}_1^{\mathbf{A}}\right)^T, & \left(\mathbf{r}_2^{\mathbf{A}}\right)^T, & \ldots, & \left(\mathbf{r}_m^{\mathbf{A}}\right)^T \end{bmatrix} = \begin{bmatrix} \left(\mathbf{s}_1^{\mathbf{A}}\right)^T \\ \left(\mathbf{s}_2^{\mathbf{A}}\right)^T \\ \vdots \\ \left(\mathbf{s}_n^{\mathbf{A}}\right)^T \end{bmatrix}$$

- *Zamění se počet řádků a počet sloupců*
- *Výpočetní náročnost: f(n)=0, tj. O(1)*

Operace s maticemi

- *Násobení matice m x n a vektoru dimenze n*

$$[\mathbf{Av}]_i = [\mathbf{A}]_{i1}[\mathbf{v}]_1 + [\mathbf{A}]_{i2}[\mathbf{v}]_2 + \ldots + [\mathbf{A}]_{in}[\mathbf{v}]_n = \sum_{k=1}^{n} [\mathbf{A}]_{ij}[\mathbf{v}]_j = \mathbf{r}_i^{\mathbf{A}} \cdot \mathbf{v}$$

$$\mathbf{Av} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} a_{11}v_1 + a_{12}v_2 + \ldots + a_{1n}v_n \\ a_{21}v_1 + a_{22}v_2 + \ldots + a_{2n}v_n \\ \vdots \\ a_{m1}v_1 + a_{m2}v_2 + \ldots + a_{mn}v_n \end{bmatrix}$$

- *Výpočetní náročnost při m=n: $f(n)=n(n+n-1)=2n^2-n$, tj. $O(n^2)$*

Výkonnostní test

- *Operace násobení skalárem + násobení vektorem + sčítání vektorů*
- Implementace: **y** <- α**Ax** + β**y**
- *Výpočetní náročnost: $f(n)=3n^2+n$, tj. $O(n^2)$*

```
void dgemv(int m, int n, double alpha, double *A, double *x,
                double beta, double *y) {
    int i,j;
     for(i=0;i<m;i++) {
          y[i] *= beta;
           for(j=0;j<n;j++)
                y[i] += alpha*A[m*i+j]*x[j];
     }
}
```

Výkonnostní test

- *Operace násobení skalárem + násobení vektorem + sčítání vektorů*
- Implementace: $\mathbf{y} := \alpha(\mathbf{Ax}) + \beta\mathbf{y}$
- *Výpočetní náročnost:* $f(n)=2n^2+3n$, *tj.* $O(n^2)$

```
void dgemv(int m, int n, double alpha, double *A, double *x,
                double beta, double *y) {
   int i,j; double temp;
    for(i=0;i<m;i++) {
        y[i] *= beta; temp=0;
         for(j=0;j<n;j++)
            temp += A[m*i+j]*x[j];
        y[i] += temp*alpha;
    }
}
```

Výsledky testu procedury DGEMV

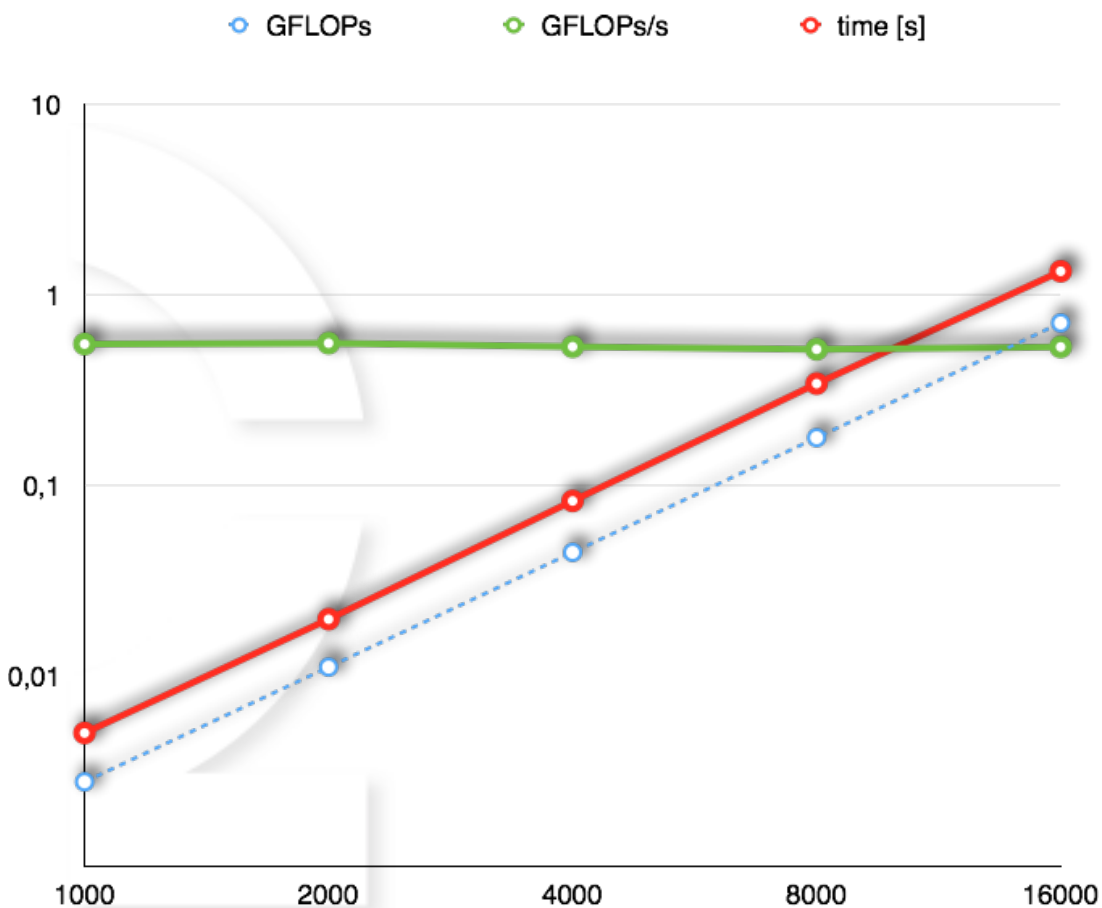
MY_DGEMV	GFLOPs	GFLOPs/s	time [s]
1000	0,0027949	0,554378	0,0050415
2000	0,0111777	0,560392	0,0199463
4000	0,0447072	0,536932	0,0832642
8000	0,178821	0,520346	0,343658
16000	0,715271	0,535532	1,33563

Předpona	FLOPS
yotta	10^{24}
zetta	10^{21}
exa	10^{18}
peta	10^{15}
tera	10^{12}
giga	10^{9}
mega	10^{6}
kilo	10^{3}

$$\frac{f(2n)}{f(n)} = \frac{3(2n)^2 + 2n}{3n^2 + n} = \frac{12n^2 + 4n - 2n}{3n^2 + n} =$$

$$= \frac{4(3n^2 + n) - 2n}{n(3n+1} = 4 - \frac{2n}{n(3n+1)} = 4 - \frac{2}{3n+1} \approx 4$$

Výsledky testu procedury DGEMV



Předpona	FLOPS
yotta	10^{24}
zetta	10^{21}
exa	10^{18}
peta	10^{15}
tera	10^{12}
giga	10^{9}
mega	10^{6}
kilo	10^{3}

Operace s maticemi

- *Násobení matice m x p s maticí p x n*

$$\mathbf{AB} = \mathbf{A}\left[\mathbf{s}_1^{\mathbf{B}}, \ldots, \mathbf{s}_n^{\mathbf{B}}\right] = \left[\mathbf{A}\mathbf{s}_1^{\mathbf{B}}, \ldots, \mathbf{A}\mathbf{s}_n^{\mathbf{B}}\right]$$

$$[\mathbf{AB}]_{ij} = [\mathbf{A}]_{i1}[\mathbf{s}_j^{\mathbf{B}}]_1 + [\mathbf{A}]_{i2}[\mathbf{s}_j^{\mathbf{B}}]_2 + \ldots + [\mathbf{A}]_{ip}[\mathbf{s}_j^{\mathbf{B}}]_p = \sum_{k=1}^{p}[\mathbf{A}]_{ik}[B]_{kj} = \mathbf{r}_i^{\mathbf{A}} \cdot \mathbf{s}_j^{\mathbf{B}}$$

$$\mathbf{AB} = \begin{bmatrix} \\ a_{i1} & \cdots & a_{ip} \\ \\ \end{bmatrix} \begin{bmatrix} & b_{1j} & \\ & \vdots & \\ & b_{pj} & \end{bmatrix} = \begin{bmatrix} \\ & \boxed{\mathbf{r}_i^{\mathbf{A}} \cdot \mathbf{s}_j^{\mathbf{B}}} & \\ \\ \end{bmatrix}$$

- *Výpočetní náročnost při m=n=p: $f(n)=n^2(2n-1)=2n^3-n^2$, tj. $O(n^3)$*

Výkonnostní test

- *Operace násobení skalárem + násobení vektorem + sčítání vektorů*
- Implementace: **C** := α**AB** + β**C**
- *Výpočetní náročnost: $f(n)=3n^3+n^2$, tj. $O(n^3)$*

```
void dgemm(int m, int n, int k, double alpha, double *A, double *B,
            double beta, double *C) {
    int i,j,c;
    for(i=0;i<m;i++) {
        for(j=0;j<n;j++) {
            C[m*i+j] *= beta;
             for(c=0;c<n;c++)
                C[m*i+j] += alpha*A[m*i+c]*B[k*c+j];
        }
    }
}
```

Výsledky testu procedury DGEMM

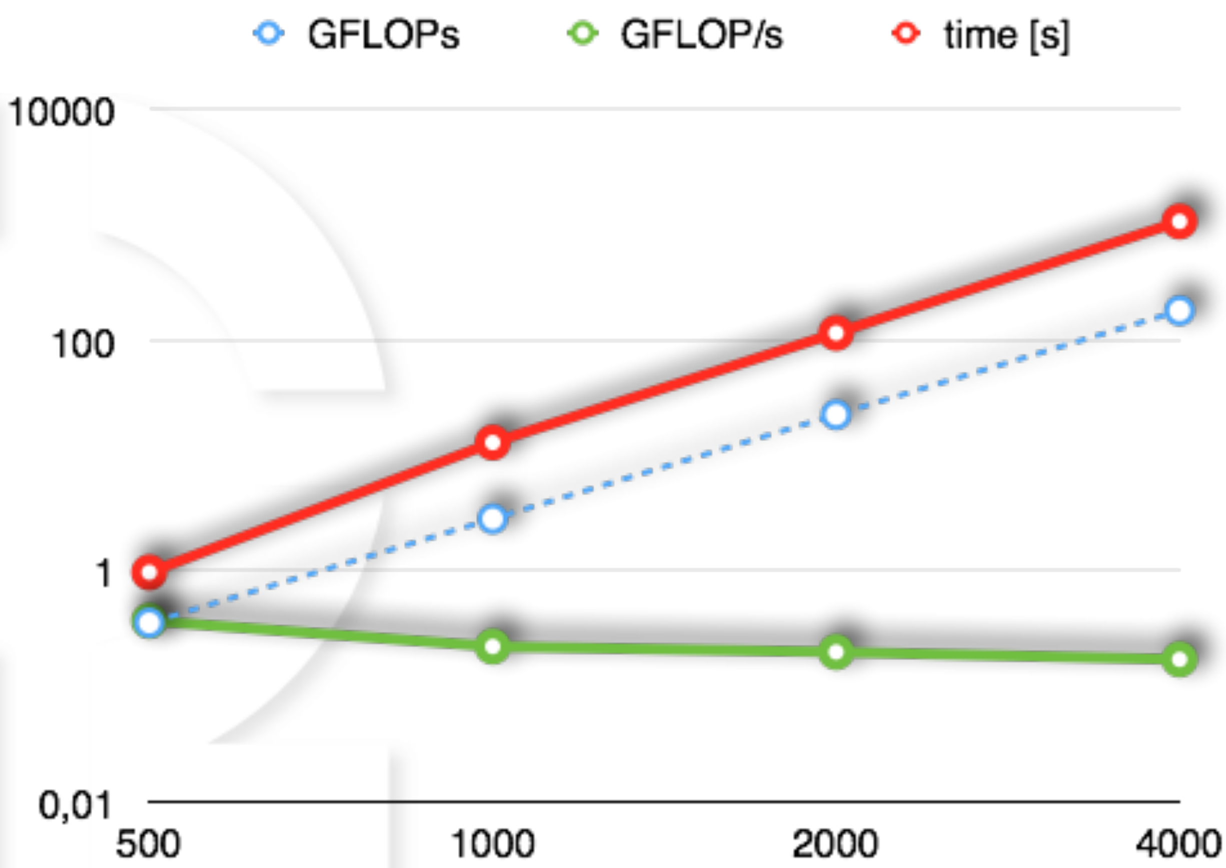
VV_DGEMM	GFLOPs	GFLOP/s	time [s]
500	0,349479	0,364412	0,959021
1000	2,7949	0,218035	12,8186
2000	22,3555	0,195408	114,404
4000	178,829	0,168494	1061,34

Předpona	FLOPS
yotta	10^{24}
zetta	10^{21}
exa	10^{18}
peta	10^{15}
tera	10^{12}
giga	10^{9}
mega	10^{6}
kilo	10^{3}

$$\frac{f(2n)}{f(n)} = \frac{3(2n)^3 + 2(2n)^2}{3n^3 + 2n^2} = \frac{24n^3 + 8n^2 + 8n^2 - 8n^2}{3n^3 + 2n^2} =$$

$$= \frac{8(3n^3 + 2n^2) - 8n^2}{3n^3 + 2n^2} = 8 - \frac{8n^2}{n^2(3n+1)} = 8 - \frac{8}{3n+1} \approx 8$$

Výsledky testu procedury DGEMM



Předpona	FLOPS
yotta	10^{24}
zetta	10^{21}
exa	10^{18}
peta	10^{15}
tera	10^{12}
giga	10^{9}
mega	10^{6}
kilo	10^{3}

Basic Linear Algebra Subprograms - BLAS

- *Standardizace nejčastěji používaných operací v lineární algebře, tj. operací s vektory a maticemi*
- *Poprvé zmíněno Lawsonem, Hansonem, Kincaidem a Kroghem v roce 1979.*
- *Dnes existuje celá řada knihoven implementující velmi efektivně BLAS*
  - **ACML -** AMD Core Math Library
  - **ATLAS -** Automatically Tuned Linear Algebra Software
  - **cuBLAS –** BLAS pro NVIDIA GPU karty
  - **Intel MKL -** Intel Math Kernel Library
  - **Netlib BLAS ….**

Standardizace názvů subroutin

- *Dle datových typů*
  - S ... Single precision
  - D ... Double precision
  - C ... Komplexní čísla
- *Dle typů matic:*
  - GE – GEneral – obecné matice
  - GB – General Banded – obecná pásová matice
  - SY – Symmetric – symetrická matice
  - TR – Triangular – trojúhelníková matice
- *Příklad: DGEMM = Double precision, GEneral matrix, Matrix-Matrix operation*

BLAS úrovně 1

- *Operace s vektory a s asymptotickou náročností O(n)*

Název	Operace	Předpony
_SWAP	$x \leftrightarrow y$	S, D, C, Z
_SCAL	$x \longleftarrow \alpha x$	S, D, C, Z, CS, ZD
_COPY	$y \longleftarrow x$	S, D, C, Z
_AXPY	$y \longleftarrow \alpha x + y$	S, D, C, Z
_DOT	$dot \longleftarrow x^T y$	S, D, DS
_NRM2	$nrm2 \longleftarrow \|\|x\|\|_2$	S, D, SC, DZ
_ASUM	$asum \longleftarrow \|\|Re(x)\|\|_1 + \|\|Im(x)\|\|_1$	S, D, SC, DZ

BLAS úrovně 2

- *Operace s vektory a maticemi a s asymptotickou náročností $O(n^2)$*

Název	Operace	Předpony
_GEMV	$y \longleftarrow \alpha Ax + \beta y$, $y \longleftarrow \alpha A^T x + \beta y$	S, D, C, Z
_GBMV	$y \longleftarrow \alpha Ax + \beta y$, $y \longleftarrow \alpha A^T x + \beta y$	S, D, C, Z
_SYMV	$y \longleftarrow \alpha Ax + \beta y$	S, D
_SBMV	$y \longleftarrow \alpha Ax + \beta y$	S, D
_TRMV	$x \longleftarrow Ax$, $x \longleftarrow A^T x$, $x \longleftarrow A^H x$	S, D, C, Z
_TRSV	$x \longleftarrow A^{-1}x$, $x \longleftarrow A^{-T}x$, $x \longleftarrow A^{-H}x$	S, D, C, Z
_GER	$A \longleftarrow \alpha x^T y + A$	S, D
_SYR	$A \longleftarrow \alpha x^T x + A$	S, D

BLAS úrovně 3

- *Operace s maticemi a s asymptotickou náročností $O(n^3)$*

Název	Operace	Předpony
_GEMM	$C \longleftarrow \alpha op(A)op(B) + \beta C$, $op(X)=X$, $op(X)=X^T$	S, D, C, Z
_SYMM	$C \longleftarrow \alpha AB + \beta C$, $C \longleftarrow \alpha AB + \beta C$	S, D, C, Z
_SYRK	$C \longleftarrow \alpha AA^T + \beta C$, $C \longleftarrow \alpha A^TA + \beta C$	S, D
_TRMM	$B \longleftarrow \alpha AB$, $B \longleftarrow \alpha A^TB$, $B \longleftarrow \alpha BA$, $B \longleftarrow \alpha BA^T$	S, D, C, Z
_TRSM	$B \longleftarrow \alpha A^{-1}B$, $B \longleftarrow \alpha A^{-T}B$, $B \longleftarrow \alpha BA^{-1}$, $B \longleftarrow \alpha BA^{-T}$	S, D, C, Z

Výkonnostní test DGEMM

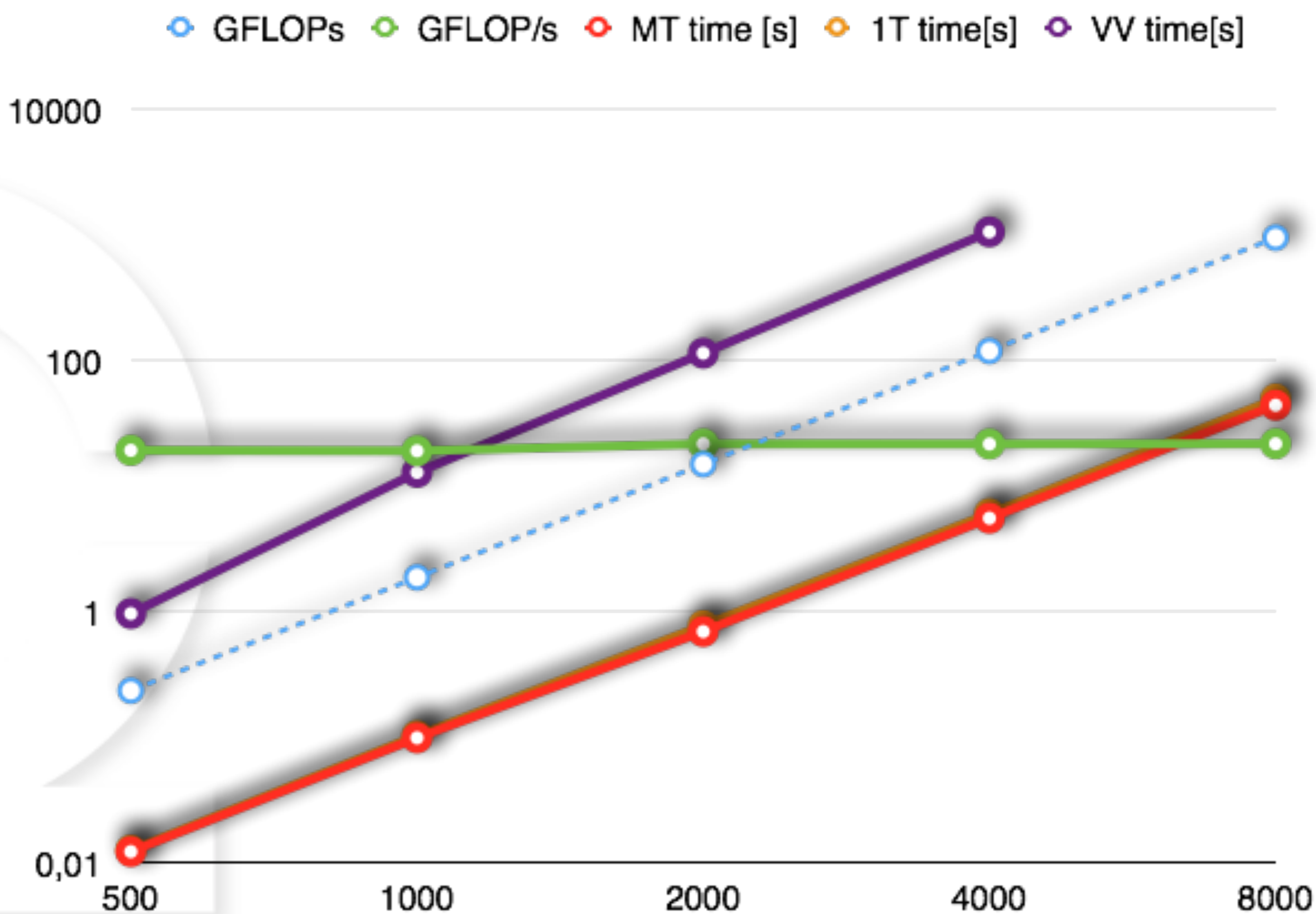
- DGEMM: **C** := α**AB** + β**C** implementace **Intel Math Kernel Library**
- *Výpočetní náročnost: $O(n^3)$, paměťové nároky: cca 1,5GB*

MKL DGEMM	GFLOPs	GFLOP/s	MT time [s]	1T time[s]	VV time[s]
500	0,233063	19,2271	0,0121216	0,0124512	0,959021
1000	1,86358	19,0366	0,0978943	0,101953	12,8186
2000	14,9049	21,6213	0,689362	0,769043	114,404
4000	119,224	21,577	5,52551	5,99673	1061,34
8000	953,734	21,6729	44,0057	49,2425	

Testováno na MacBook Air, Intel i7, 1.8GHz

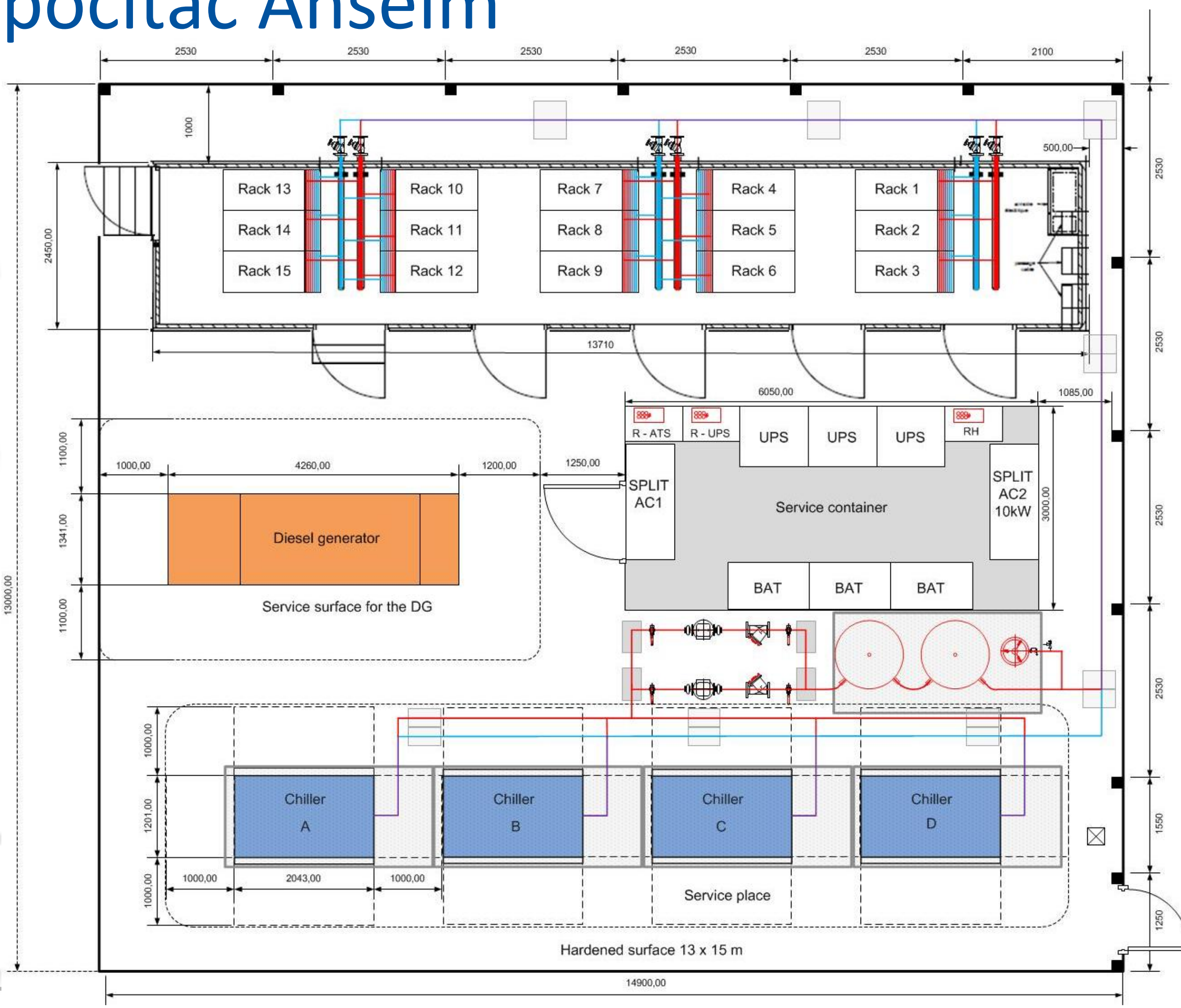
Teoretický výkon: 2 x 1,8GHz x 8 = 28,8 GFLOP/s

Výsledky testu procedury DGEMM



Superpočítač Anselm

Superpočítač Anselm



Superpočítač Anselm

X86_64 cluster Bull Extreme Computing bullx technology

- *Výpočetní uzly bez akcelerace*
  - 180 uzlů
  - 2880 CPU jader
  - 2x Intel Sandy Bridge E5-2665, 8-core, 2.4GHz procesorů na jádro
  - 64 GB fyzické paměti na uzel
  - 1x 500GB SATA 2,5” 7,2 krpm HDD na uzel
  - Teoretický výkon:: 19.2 Gflop/s na jádro
- *Výpočetní uzly s akcelerací*
  - 27 uzlů
  - 2x Intel Sandy Bridge E5-2470, 8-core, 2.3GHz procesorů na uzel
  - 96 GB fyzické paměti na uzel
  - 1x 500GB SATA 2,5” 7,2 krpm na uzel
  - 23x GPU accelerator NVIDIA Tesla Kepler K20 1x per node
  - 4x MIC accelerator 1x Intel Phi 5110P per node

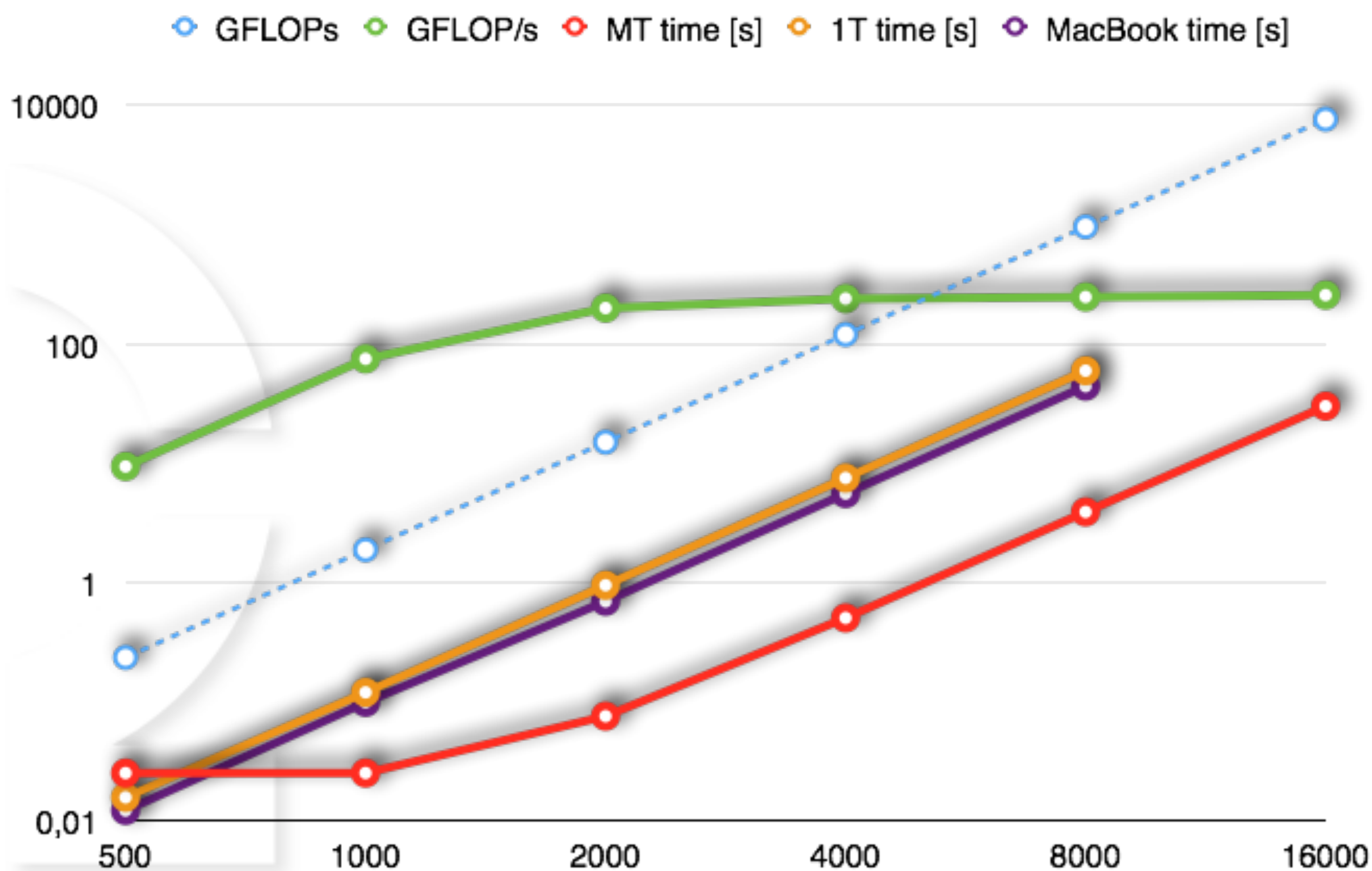
Výkonnostní test DGEMM na Anselmovi

- DGEMM: **C** := α**AB** + β**C** implementace **Intel Math Kernel Library**
- *Výpočetní náročnost: $O(n^3)$, Paměťová náročnost: cca 6GB*

MKL DGEMM	GFLOPs	GFLOP/s	MT time [s]	1T time [s]	MacBook time [s]
500	0,233063	9,32254	0,025	0,015625	0,0121216
1000	1,86358	74,5431	0,025	0,11875	0,0978943
2000	14,9049	198,732	0,075	0,940625	0,689362
4000	119,224	238,448	0,5	7,46562	5,52551
8000	953,734	246,125	3,875	59,3	44,0057
16000	7629,63	255,385	29,875		

Teoretický výkon 1 uzlu: 16 x 2,4GHz x 8 = 307,2 GFLOP/s

Výsledky testu DGEMM na Anselmovi



Blokové vektory a matice

- *Rozdělení vektoru do nezávislých bloků*

$$\mathbf{v} = \left[\begin{array}{ccc|ccc|ccc} 1 & 2 & -1 & 0 & 1 & 3 & -2 & 5 & 2 \end{array}\right]$$

$$\mathbf{v} = \left[\begin{array}{ccc} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{array}\right]$$

Příklad:

Blokový výpočet skalárního součinu

$$\left.\begin{array}{l} \mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3] \\ \\ \mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] \end{array}\right\} \mathbf{u} \cdot \mathbf{v} = \left\{\begin{array}{l} \alpha_1 = \mathbf{u}_1 \cdot \mathbf{v}_1 \\ \alpha_2 = \mathbf{u}_2 \cdot \mathbf{v}_2 \\ \alpha_3 = \mathbf{u}_3 \cdot \mathbf{v}_3 \end{array}\right\} = \alpha_1 + \alpha_2 + \alpha_3$$

Blokové vektory a matice

- *Rozdělení matic do nezávislých bloků*

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 & 0 \\ -1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 1 & -1 \end{bmatrix} \rightarrow \mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

Blokový součin matice vektor:

$$\mathbf{A}\mathbf{v} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \left\{ \begin{array}{l} \mathbf{u}_1 = \mathbf{A}_{11}\mathbf{v}_1 \\ \mathbf{u}_2 = \mathbf{A}_{12}\mathbf{v}_2 \\ \mathbf{u}_3 = \mathbf{A}_{21}\mathbf{v}_1 \\ \mathbf{u}_4 = \mathbf{A}_{22}\mathbf{v}_2 \end{array} \right\} = \begin{bmatrix} \mathbf{u}_1 + \mathbf{u}_2 \\ \mathbf{u}_3 + \mathbf{u}_4 \end{bmatrix}$$

Jiný příklad distribuce maticových bloků

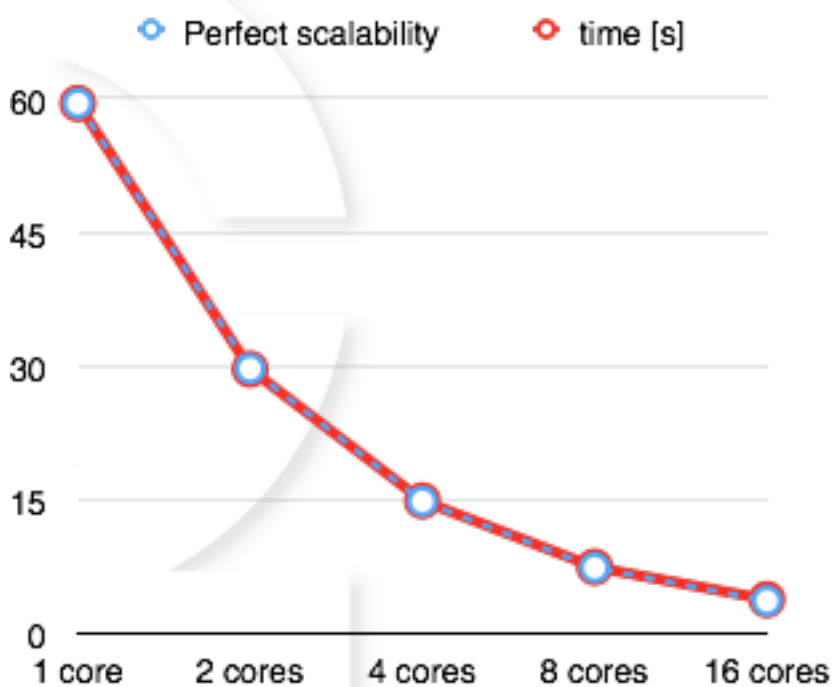
$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 & 0 \\ -1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 1 & -1 \end{bmatrix} \rightarrow \mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} \\ \mathbf{A}_{21} \\ \mathbf{A}_{31} \\ \mathbf{A}_{41} \end{bmatrix}$$

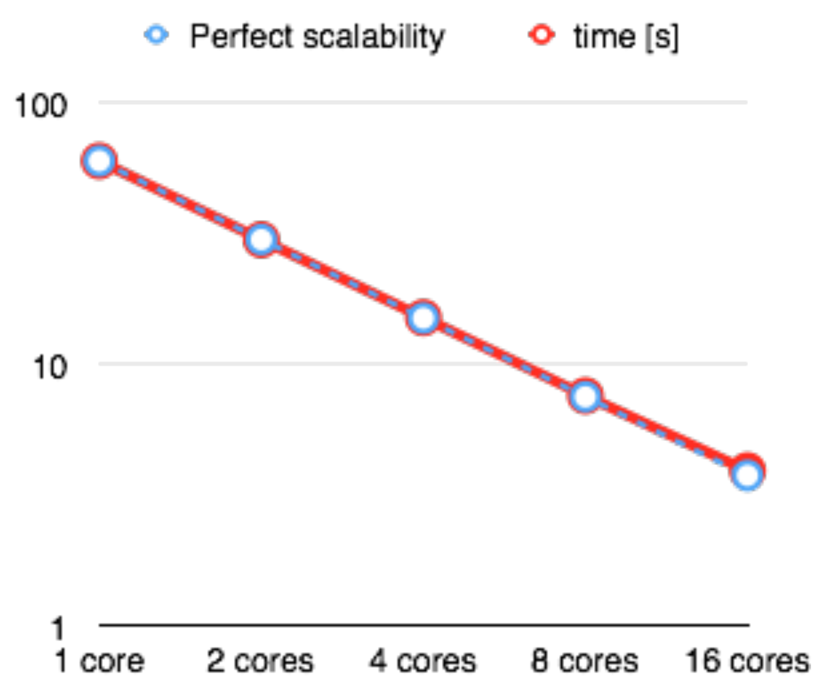
Blokový součin matice vektor:

$$\mathbf{Av} = \begin{bmatrix} \mathbf{A}_{11} \\ \mathbf{A}_{21} \\ \mathbf{A}_{31} \\ \mathbf{A}_{41} \end{bmatrix} \mathbf{v} = \left\{ \begin{bmatrix} \mathbf{u}_1 = \mathbf{A}_{11}\mathbf{v} \\ \mathbf{u}_2 = \mathbf{A}_{21}\mathbf{v} \\ \mathbf{u}_3 = \mathbf{A}_{31}\mathbf{v} \\ \mathbf{u}_4 = \mathbf{A}_{41}\mathbf{v} \end{bmatrix} \right.$$

Škálovatelnost DGEMM na Anselmovi

- *Silná paralelní škálovatelnost: čas potřebný pro řešení úlohy na N procesorech je T(N) = T(1)/N*

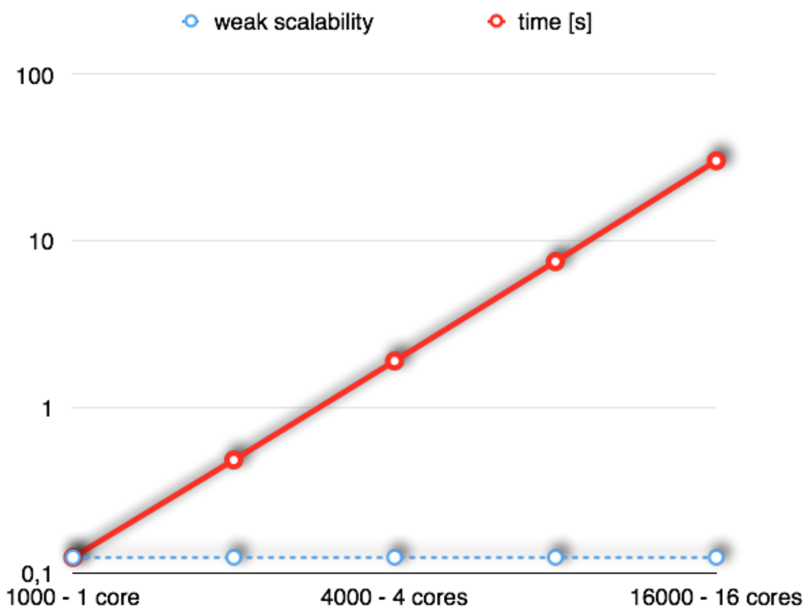




Škálovatelnost DGEMM na Anselmovi

- *Slabá paralelní škálovatelnost: čas potřebný pro řešení úlohy se nemění při zachování velikosti úlohy na jeden procesor*

DGEMM weak	weak scalability	time [s]
1000 - 1 core	0,125	0,125
2000 - 2 cores	0,125	0,48125
4000 - 4 cores	0,125	1,89375
8000 - 8 cores	0,125	7,49375
16000 - 16 cores	0,125	30,2188

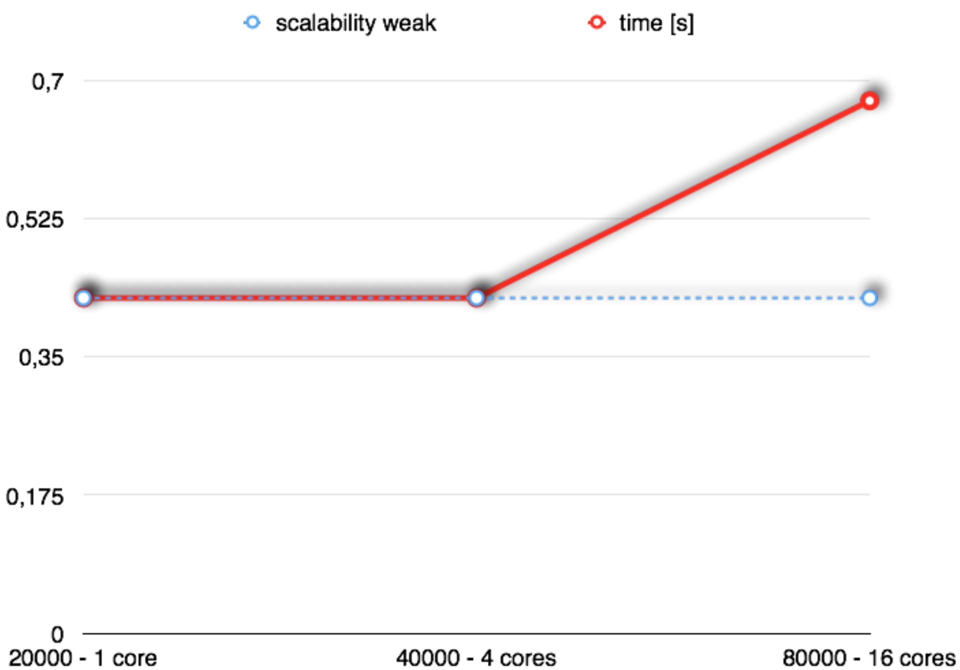


Chyba!!! – Výpočetní náročnost 2x větší úlohy je osminásobná!
2000 musíme spustit na 8 jádrech
4000 musíme spustit na 64 jádrech!!!!

Škálovatelnost DGEMV na Anselmovi

- *Slabá paralelní škálovatelnost: čas potřebný pro řešení úlohy se nemění při zachování velikosti úlohy na jeden procesor*

DGEMV weak	scalability weak	time [s]
20000 - 1 core	0,425	0,425
40000 - 4 cores	0,425	0,425
80000 - 16 cores	0,425	0,675



Soustavy lineárních rovnic

- *Jedna lineární rovnice o jedné neznámé:*

$$ax = b \Leftrightarrow x = \frac{b}{a} = a^{-1}b \quad \text{pokud } a \neq 0$$

- *2 lineární rovnice o 2 neznámých*

$$\begin{array}{ccccc} ax & + & by & = & e \\ cx & + & dy & = & f \end{array}$$

  – Řešení substitucí nebo sečtením rovnic

- *Jak se řeší soustavy o tisících, miliónech až miliardách neznámých?*

Soustavy lineárních rovnic

- *Soustava m lineárních rovnic o n neznámých*

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & a_{mn}x_n & = & b_m \end{array}$$

- *Maticový zápis*

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \mathbf{b}$$

Trojúhelníkový (LU) rozklad

- *Dále budeme uvažovat m=n, tj. n rovnic o n neznámých*
- *A=LU, kde L resp. U je tzv. dolní resp. horní trojúhelníková matice*

$$\mathbf{A}=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}=\mathbf{LU}$$

$$a_{11}=\mathbf{r}_1^{\mathbf{L}}\cdot\mathbf{s}_1^{\mathbf{U}}=1\cdot u_{11}+0\cdot 0+0\cdot 0=u_{11}$$
$$a_{12}=\mathbf{r}_1^{\mathbf{L}}\cdot\mathbf{s}_2^{\mathbf{U}}=1\cdot u_{12}+0\cdot u_{22}+0\cdot 0=u_{12}$$
$$a_{13}=\mathbf{r}_1^{\mathbf{L}}\cdot\mathbf{s}_3^{\mathbf{U}}=1\cdot u_{13}+0\cdot u_{23}+0\cdot u_{23}=u_{13}$$

$$a_{21}=\mathbf{r}_2^{\mathbf{L}}\cdot\mathbf{s}_1^{\mathbf{U}}=l_{21}\cdot u_{11}+1\cdot 0+0\cdot 0=l_{21}\cdot u_{11}\Rightarrow l_{21}=\frac{a_{21}}{u_{11}}$$
$$a_{22}=\mathbf{r}_2^{\mathbf{L}}\cdot\mathbf{s}_2^{\mathbf{U}}=l_{21}\cdot u_{12}+1\cdot u_{22}+0\cdot 0=l_{21}\cdot u_{12}+u_{22}\Rightarrow u_{22}=a_{22}-l_{21}\cdot u_{12}$$
$$a_{23}=\mathbf{r}_2^{\mathbf{L}}\cdot\mathbf{s}_3^{\mathbf{U}}=l_{21}\cdot u_{13}+1\cdot u_{23}+0\cdot u_{33}=l_{21}\cdot u_{13}+u_{23}\Rightarrow u_{23}=a_{23}-l_{21}\cdot u_{13}$$

$$a_{31}=\mathbf{r}_3^{\mathbf{L}}\cdot\mathbf{s}_1^{\mathbf{U}}=l_{31}\cdot u_{11}+l_{32}\cdot 0+1\cdot 0=l_{31}\cdot u_{11}\Rightarrow l_{31}=\frac{a_{31}}{u_{11}}$$
$$a_{32}=\mathbf{r}_3^{\mathbf{L}}\cdot\mathbf{s}_2^{\mathbf{U}}=l_{31}\cdot u_{12}+l_{32}\cdot u_{22}+1\cdot 0=l_{31}\cdot u_{12}+l_{32}\cdot u_{22}\Rightarrow l_{32}=\frac{a_{32}-l_{31}\cdot u_{12}}{u_{22}}$$
$$a_{33}=\mathbf{r}_3^{\mathbf{L}}\cdot\mathbf{s}_3^{\mathbf{U}}=l_{31}\cdot u_{13}+l_{32}\cdot u_{23}+1\cdot u_{33}\Rightarrow u_{33}=a_{33}-l_{31}\cdot u_{13}-l_{32}\cdot u_{23}$$

```
U=A; L=I;
for k = 1 to n-1
    for i = k+1 to n
        L[i,k]=U[i,k]/U[k,k]
        for j = k to n
            U[i,j]=U[i,j]-L[i,k]*U[k,j]
        endfor
    endfor
endfor
```

Výpočetní náročnost:

f(n)=⅔n³, tj.O(n³)

Řešení soustav rovnic LU rozkladem

$$\mathbf{Ax}=\mathbf{b} \Leftrightarrow (\mathbf{LU})\mathbf{x}=\mathbf{b} \Leftrightarrow \mathbf{L}(\mathbf{Ux})=\mathbf{b} \left\langle \begin{array}{l} \mathbf{Ly}=\mathbf{b} \\ \mathbf{Ux}=\mathbf{y} \end{array} \right.$$

- *Dopředná substituce:* **Ly=b**

$$\begin{array}{ccccccccc} l_{11}y_1 & & & & & & & = & b_1 \Rightarrow y_1 \\ l_{21}y_1 & + & y_2 & & & & & = & b_2 \Rightarrow y_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ l_{n1}y_1 & + & l_{n2}y_2 & + & \cdots & & y_n & = & b_n \Rightarrow y_n \end{array}$$

- *Zpětná substituce:* **Ux=y**

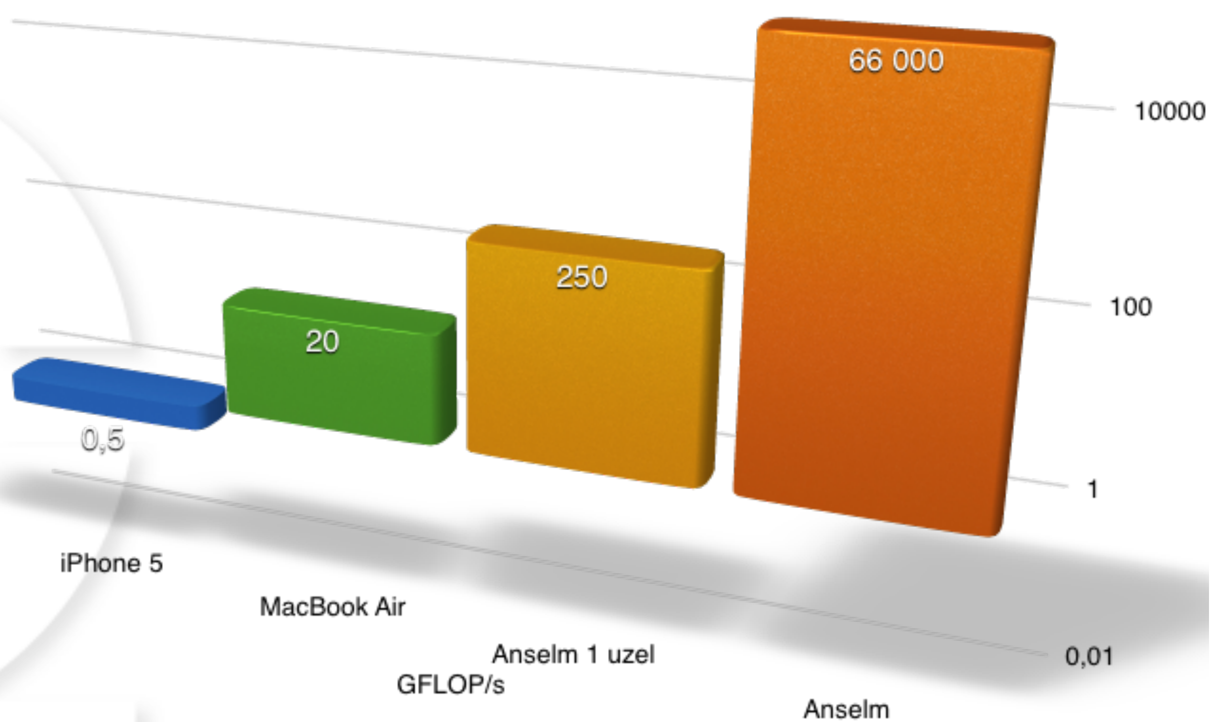
$$\begin{array}{ccccccccc} u_{11}x_1 & + & u_{12}x_2 & + & \cdots & & u_{1n}x_n & = & y_1 \Rightarrow x_1 \\ & & & & & & \vdots & & \vdots \\ & & & & u_{n-1,n}x_{n-1} & + & u_{n-1,n}x_n & = & y_{n-1} \Rightarrow x_{n-1} \\ & & & & & & u_{nn}x_n & = & y_n \Rightarrow x_n \end{array}$$

Výpočetní náročnost: $O(n^2)$

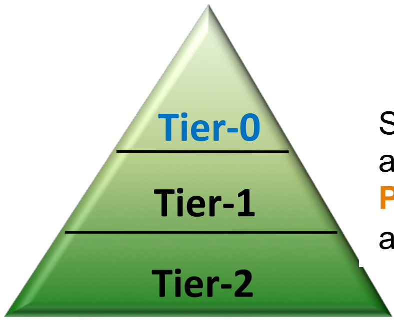
Knihovny pro řešení soustav lineárních rovnic

- *Snaha o standardizaci stejně jako v případě BLAS*
  - LINPACK, LAPACK, ScaLAPACK
  - PLASMA, MAGMA, Intel MKL, …
- *LINPACK benchmark*
  - J. Dongarra
  - Měření výkonu počítačů www.top500.org
  - HPL – High Performance Linpack
  - Výpočetní náročnost $2/3n^3+2n^2$

Test výkonu


66 000
250
20
0,5
10000
100
1
0,01
iPhone 5
MacBook Air
Anselm 1 uzel
GFLOP/s
Anselm

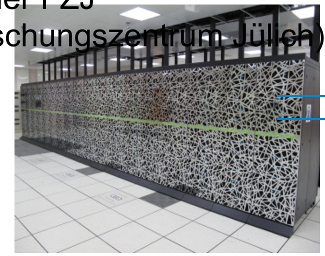
Superpočítače PRACE

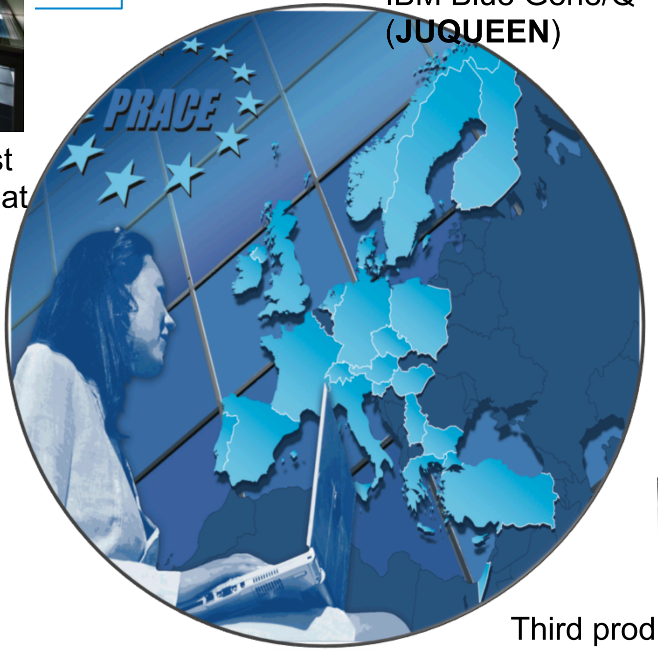


Sixth production system available by January 2013: **1 Petaflop/s** IBM (**MareNostrum**) at BSC.

Upgrade: **5.9 Petaflop/s** IBM Blue Gene/Q (**JUQUEEN**)

First production system available:
1 Petaflop/s IBM BlueGene/P (**JUGENE**) at GCS (Gauss Centre for Supercomputing) partner FZJ (Forschungszentrum Jülich)





Fifth production system available by August 2011: **2.1 Petaflop/s** IIBM BG/Q (**FERMI**) at CINEC.

Second production system available: Bull Bullx **CURIE** at GENCI partner CEA. Full capacity of **1.7 Petaflop/s** reached by late 2011.

Third production system available by the end of 2011:
1 Petaflop/s Cray (**HERMIT**) at GCS partner HLRS (High Performance Computing Center Stuttgart).

Fourth production system available by mid 2012: **3.2 Petaflop/s** IBM (**SuperMUC**) at GCS partner LRZ (Leibniz-Rechenzentrum).

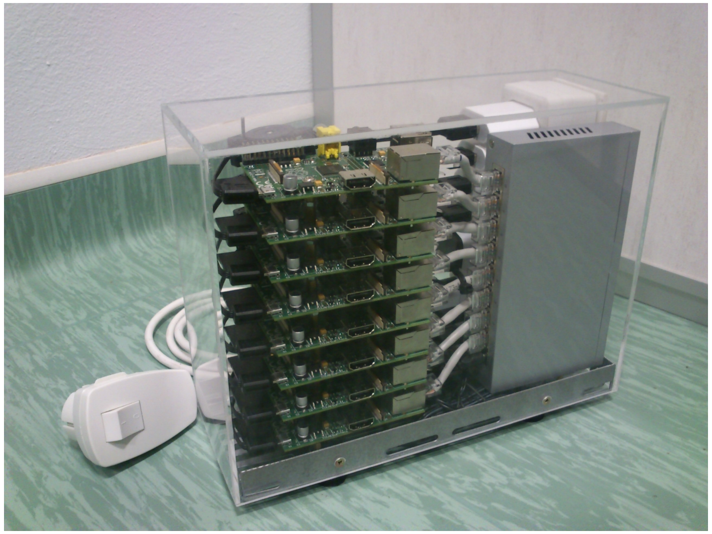
www.top500.org

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945

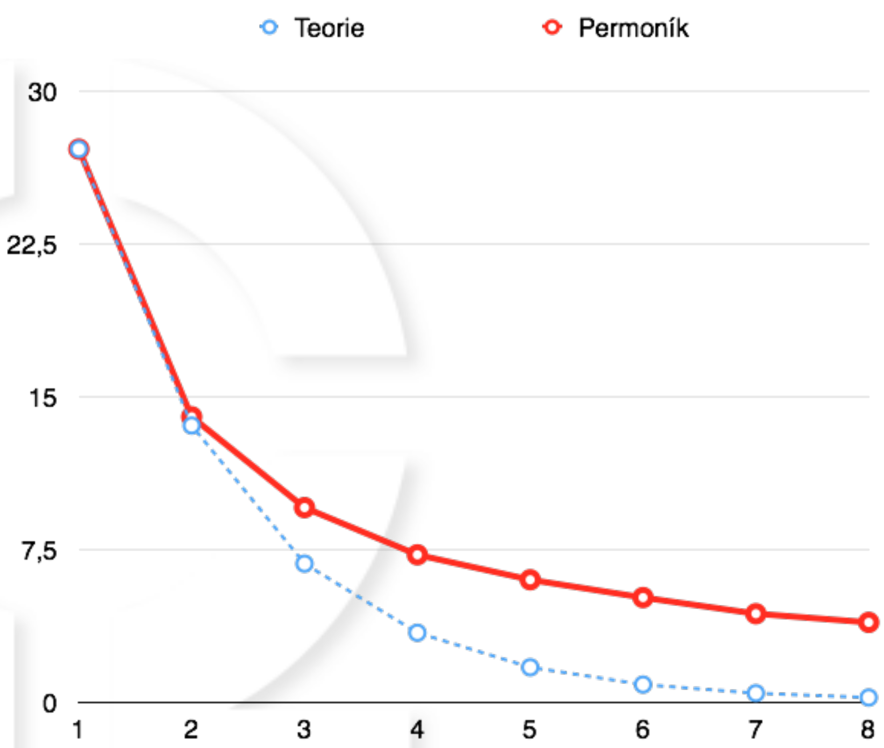
Náš Minisuperpočítač PERMONÍK (1GFLOP/s)

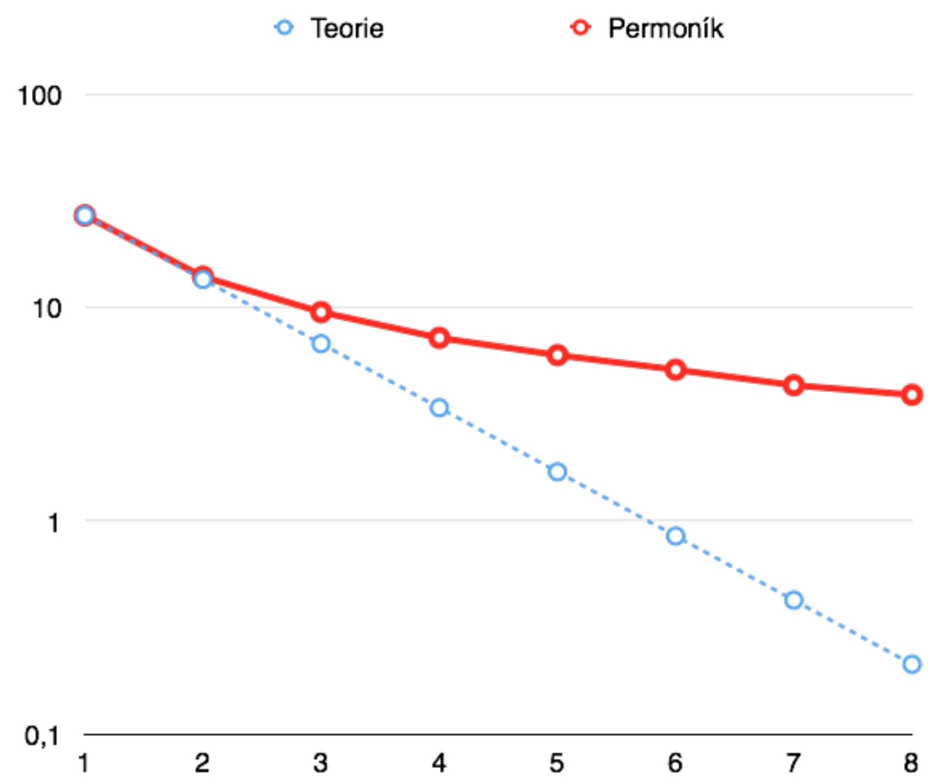
8 ks (900MHz,512MB RAM) + 1 ks + 1 ks + 2 ks + 8 ks + 8 ks

+ cca týden intenzivní práce D. & M. Horáka & L. Říhy =



Náš Minisuperpočítač PERMONÍK (1GFLOP/s)





LINPACK benchmark

http://software.intel.com/en-us/articles/intel-math-kernel-library-linpack-download

LINPACK benchmark

Available on the App Store

LINPACK benchmark

ANDROID APP ON Google play