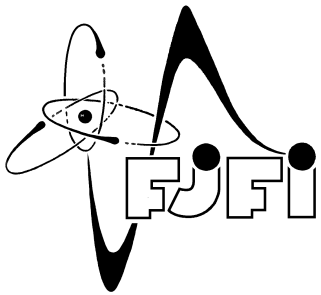
GPGPU

Obecné výpočty na grafických procesorech

Jan Vacata



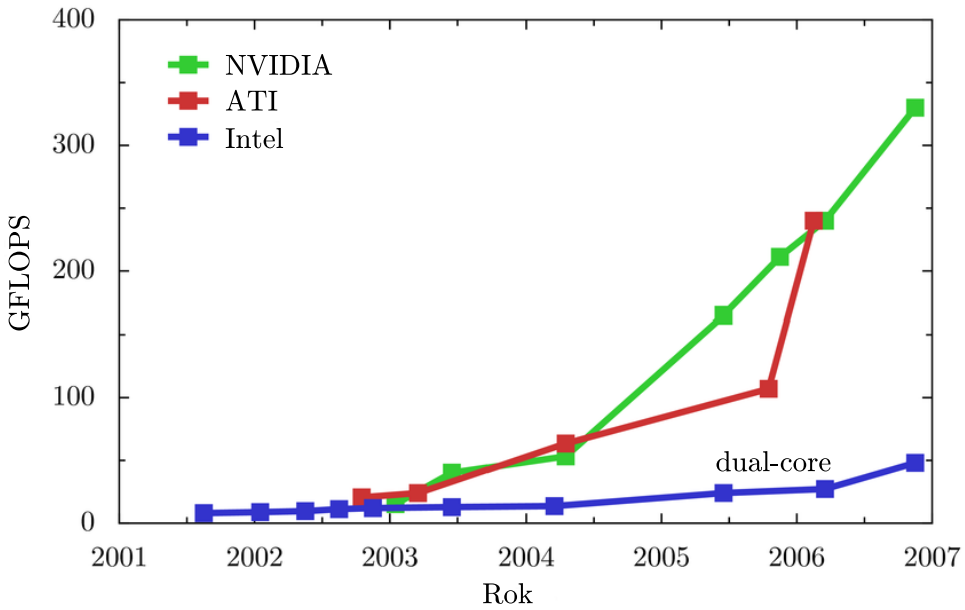
Motivace

3 GHz Intel Core 2 Extreme QX9650

- Výkon: 96 GFLOPS
- Propustnost paměti: 21 GB/s
- Orientační cena: 1300 USD

NVIDIA GeForce 9800 GX2

- Výkon: 768 GFLOPS
- Propustnost paměti: 128 GB/s
- Orientační cena: 600 USD



Obrázek: Sledování změn výkonu

Motivace - pokračování

Proč jsou GPU tak rychlé?

1. Technologický důvod: Specializace GPU pro rychlé provádění nezávislých paralelních výpočtů
2. Ekonomický důvod: Obrovský trh s počítačovými hrami

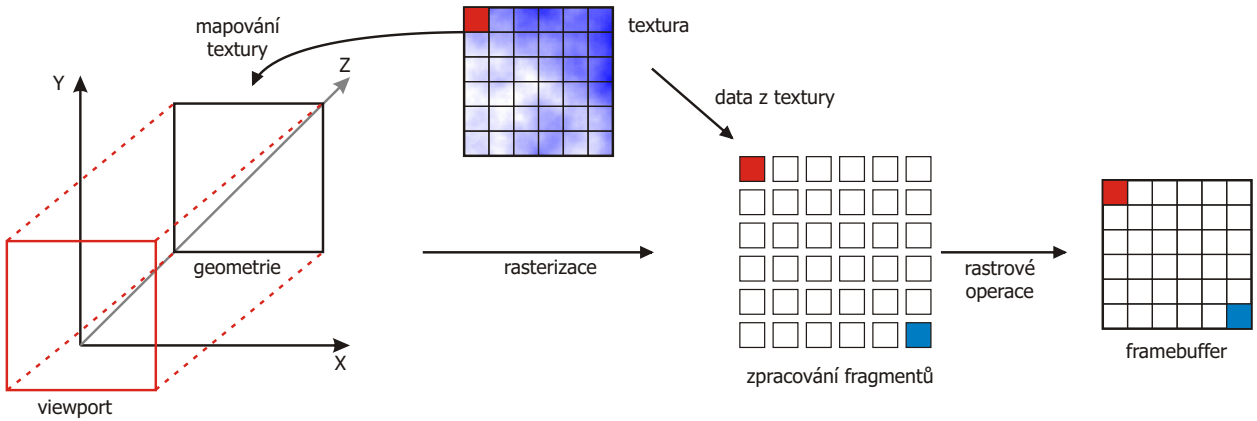
Podrobněji:

- Propustnost paměti
  - Široká paměťová směrnice pro rychlé čtení z textur (384 bitů u G80)
- Vícejádrové procesory - budoucnost paralelních výpočtů
  - Nejnovější CPU (AMD, Intel) $\longrightarrow$ 8 jader
  - GPU NVidia G92 $\longrightarrow$ 128 ($\times 2$) jader
  - GPU AMD RV670 $\longrightarrow$ 320 ($\times 2$) jader

General Purpose Computation on GPUs (GPGPU)

Nevýhody tradičního GPGPU:

- Nepřirozený programovací model, obtížné pro vývojáře neznalé principů zpracování 3D grafiky
- Další nevýhody spojené s využitím grafického API



Obrázek: Tradiční GPGPU

NVIDIA CUDA

NVIDIA CUDA (z angl. Compute Unified Device Architecture)

- Kombinace SW/HW řešení (Změny v HW, ovladač grafické karty, runtime knihovna)
- GPU jako jistá forma architektury se sdílenou pamětí
- GPU jako koprocesor pro provádění nezávislých, datově paralelních výpočtů
- Snadno uchopitelný programovací model, naprosté odstínění od grafických pojmů
- Očekávaná podpora aritmetiky *fp64* (double)

Metody Krylovových podprostorů

Řešení soustav lineárních algebraických rovnic

$$Ax = b, \quad A \in \mathbb{R}^{n,n} \text{ regulární}, x \in \mathbb{R}^n, b \in \mathbb{R}^n$$

- Nejznámější zástupci nestacionárních iteračních metod
- Hledáme aproximaci přesného řešení x_* v prostoru

$$K_m(A, r_0) = [r_0, Ar_0, A^2 r_0, \ldots, A^{m-1} r_0]_\lambda, \quad r_0 = b - Ax_0$$

- Metodám společné
  1. Nalezení ortonornální báze

$$K_m(A, r_0) = [v_1, \ldots, v_m]_\lambda, \quad V_m^T V_m = I$$

  2. Hledání aproximace ve tvaru

$$x_m = x_0 + V_m y_m$$

Metody Krylovových podprostorů - pokračování

Konkrétní metody:

- Metoda Konjugovaných gradientů (CG)

$$r_m = b - Ax_m \perp K_m(A, r_0)$$

Minimalizuje

$$\|x_* - x_m\|_A = (A(x_* - x_m), x_* - x_m)^{\frac{1}{2}}$$

- Metoda Zobecněných minimálních reziduí (GMRES)

$$r_m = b - Ax_m \perp AK_m(A, r_0)$$

Minimalizuje

$$\|b - Ax_m\|_2$$

Typy a struktura operací (CG, GMRES)

- Operace

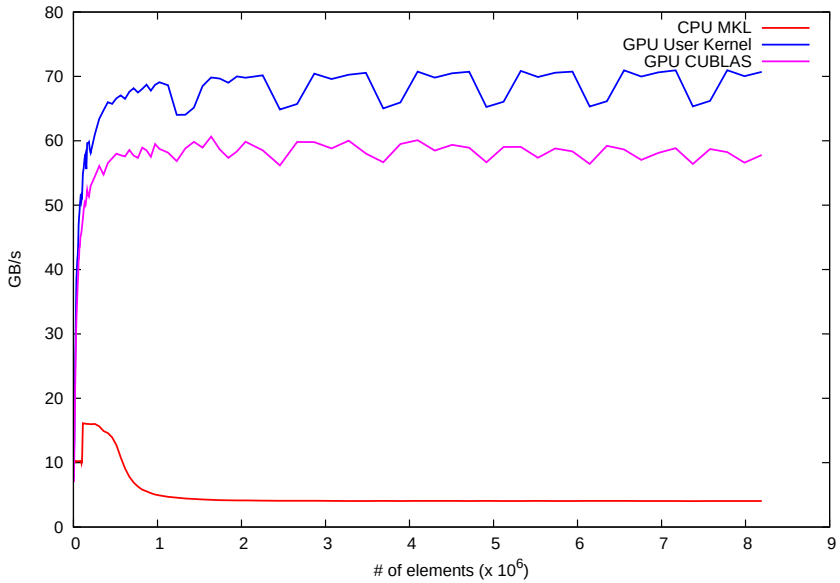
Saxpy	$y = \alpha x + y$	$x, y \in \mathbb{R}^n, \alpha \in \mathbb{R}$
Sdot	$\alpha = (x, y)$	$x, y \in \mathbb{R}^n, \alpha \in \mathbb{R}$
SpMV	$y = Ax$	$x, y \in \mathbb{R}^n, A \in \mathbb{R}^{n,n}$

- Struktura operací

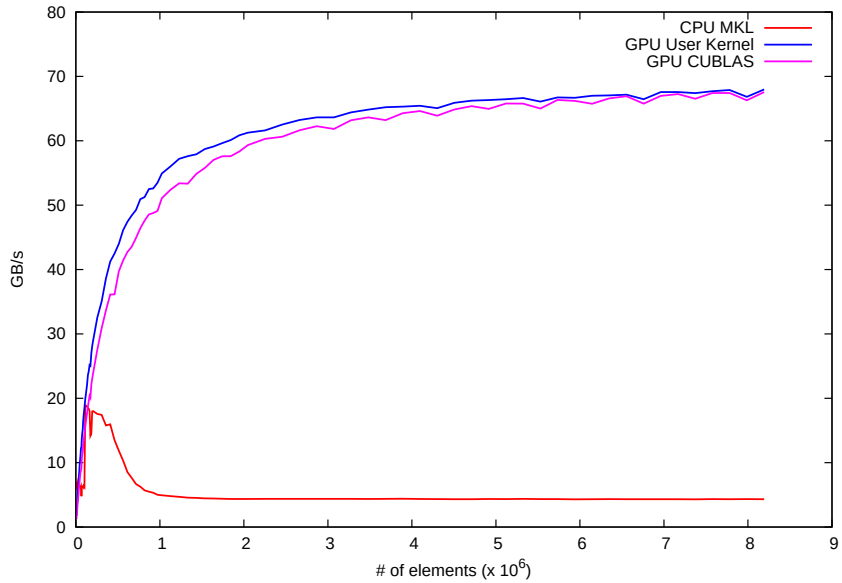
Metoda	`Sdot`	`Saxpy`	`SpMV`	Předpodmínění
CG	2	3	1	1
GMRES	$i+1$	$i+1$	1	1

Měření operací Saxpy/Sdot

- Optimalizované funkce knihovny MKL na straně hostitele
- Optimalizované funkce knihovny CUBLAS
- Uživatelská implementace (CUDA)



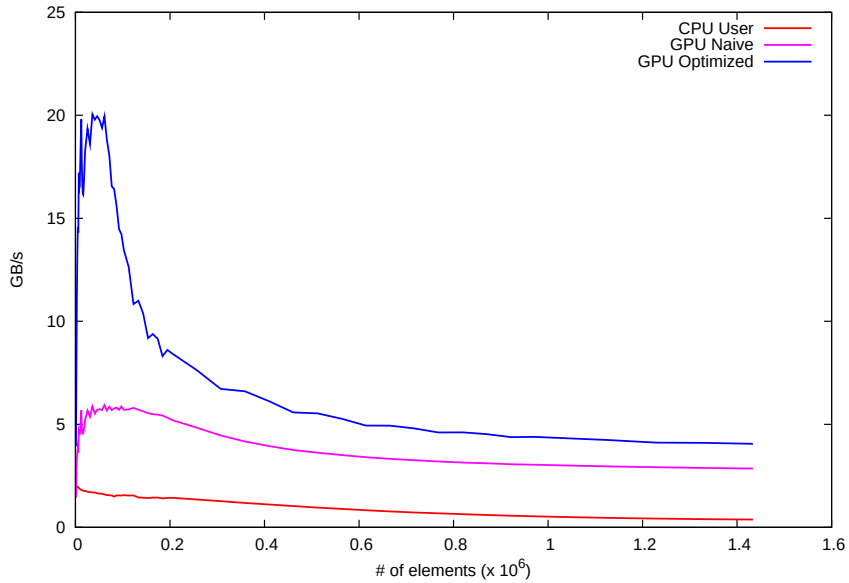
Obrázek: Operace Saxpy



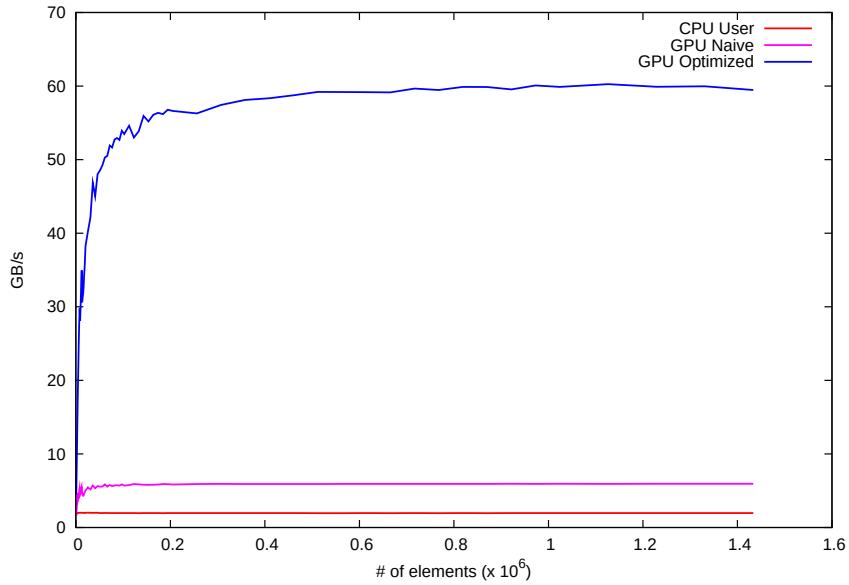
Obrázek: Operace Sdot

Měření operace SpMV

- OpenMP paralelizovaná implementace na straně hostitele
- Naivní CUDA implementace
- Optimalizovaná CUDA implementace

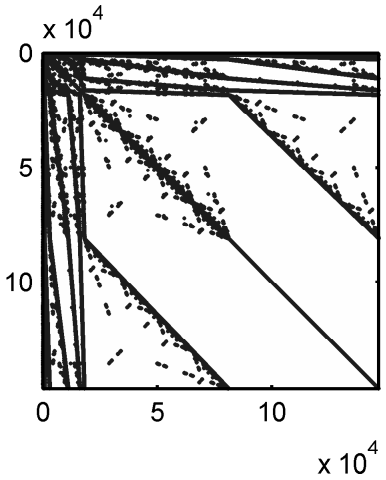


Obrázek: Operace SpMV - náhodná nestrukturovaná matice

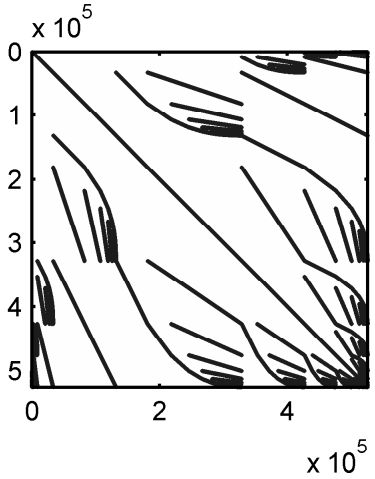


Obrázek: Operace SpMV - matice s několika diagonálami

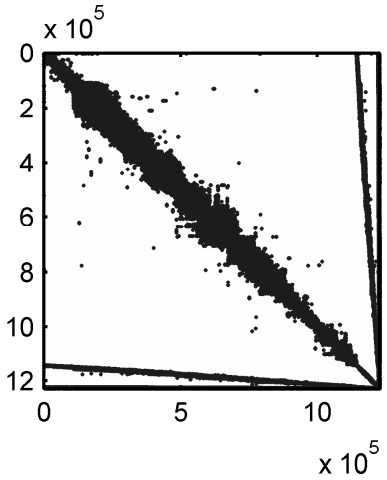
Testované matice



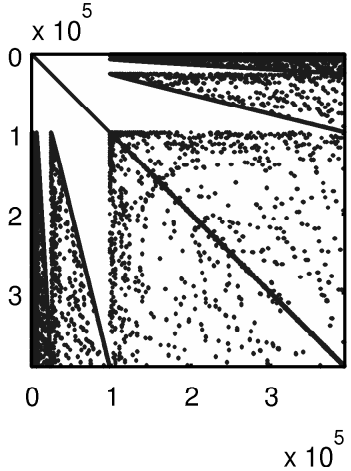
Matice *Dubcova3*



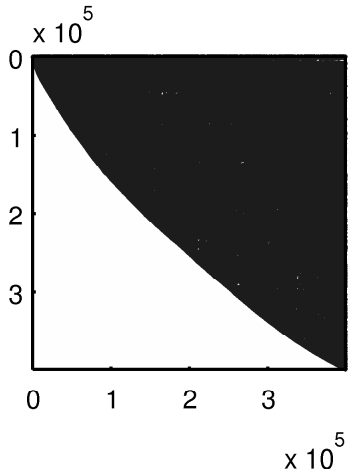
Matice *parabolic_fem*



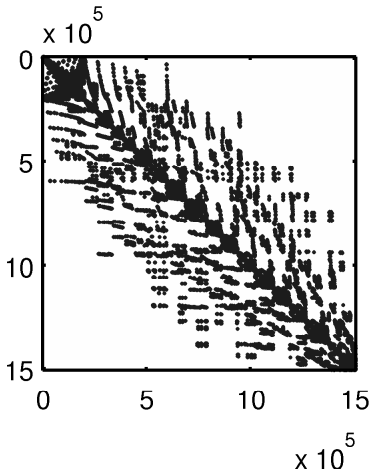
Matice *thermal2*



Matice *helm2do3*



Matice *language*



Matice *cage14*

Výsledky měření metody CG

Matice	Dubcova3	parabolic_fem	thermal2
Rozměr	146689	525825	1228045
Počet nenul	3636643	3674625	8580313
Počet Iterací	200	1000	3000
CPU			
Čas výpočtu (s)	3.97	24.7	196
CSR uloženo hodnot	7419988	7875076	18388672
Norma rezidua	2.23×10^{-4}	2.2×10^{-4}	3.4×10^{-3}
Chyba	5.18×10^{-3}	3.54×10^{-1}	4.107
GPU			
Čas výpočtu (s)	0.189	1.24	10.4
PCSR uloženo hodnot	7471914	7895026	22440374
Norma rezidua	2.18×10^{-4}	2×10^{-4}	3.2×10^{-3}
Chyba	4.75×10^{-3}	8.69×10^{-1}	2.739
Relativní urychlení	21	19.9	18.8

Výskedky měření metody GMRES

Matice	helm2d03	language	cage14
Rozměr	392257	399130	1505785
Počet nenul	2741935	1216334	27130349
Restartování	40	10	10
Počet Iterací	1000	1000	500
CPU			
Čas výpočtu (s)	40.5	66.5	96.5
CSR uloženo hodnot	5876128	2831799	55766484
Norma rezidua	1.2×10^{-1}	2.5×10^{-5}	2.2×10^{-5}
Chyba	37.4	8.55×10^{-3}	1.4×10^{-4}
GPU			
Čas výpočtu (s)	4	10.6	4.4
PCSR uloženo hodnot	5897508	10174163	58400009
Norma rezidua	1.2×10^{-1}	1.9×10^{-5}	2.3×10^{-5}
Chyba	37.3	2×10^{-4}	8.6×10^{-5}
Relativní urychlení	10.1	6.27	21.9

Literatura


Jan Vacata. *GPGPU: Obecné výpočty na grafických procesorech* diplomová práce, 2007.


Owens, J. D.; Houston, M.; Luebke, D.; Green, S.; Stone, J. E.; Phillips, J. C. *GPU Computing*. Proceedings of the IEEE 96(5):879-899, 2008.


`http://www.gpgpu.org`


`http://www.nvidia.com/object/cuda_home.html`