KIV/ZEP – 2011

Lehký úvod do paralelního výpočtu a GPGPU

Pojmy

- program = zápis algoritmu v nějakém programovacím jazyce resp. přeložen do nativního nebo bytecodu
- proces = popisovač běžícího programu; sestává z vláken
- vlákno = běžící kód, tj. to, co dostává CPU (GPU), aby se provedly nějaké operace nad nějakými daty

Paralelní zpracování

- paralelní zpracování = více HW jednotek spolupracuje na řešení úkolu
  - pracují souběžně
  - na každé jednotce jedno vlákno
- obvykle nějaká jednotka čas od času musí počkat na jinou jednotku → 4 jednotky neudělají práci za ¼ času
  - obvykle platí, že čím více jednotek použito, tím menší zisk, přidáme-li další jednotku

Současný typický HW

- počítače do kanceláře / domácnosti
  - základ (duben 2011) 7000 Kč

	Intel
procesor	Intel **Celeron Dual-Core E3400 BOX 2.6GHz** – 975 Kč
základní deska	ASUS P5G41TD-M PRO, **VGA Intel GMA 4500** – 1266 Kč
paměť	2× 2GB Kingston DIMM DDR III 1333MHz – 954 Kč
pevný disk	WD Caviar Blue WD5000AAKX - 500GB SATA III – 851 Kč
ostatní	DVD LG GH22NS, Cooler Master Elite 341 skříň, Seasonic Energy Knight SS 350W zdroj, MS Windows 7 Home Premium SP1 CZ 64bit OEM

Současný typický HW

- počítače do kanceláře / domácnosti
  - základ (duben 2011) 9700 Kč

	AMD platforma
procesor	AMD **Athlon II X3 450 3.2GHz** – 1594 Kč
základní deska	ASUS M4A88T-V EVO/USB3, **VGA ATI HD4250** – 2010 Kč
paměť	2× 2GB Kingston DIMM DDR III 1333MHz – 954 Kč
pevný disk	WD Caviar Blue WD5000AAKX - 500GB SATA III – 851 Kč
ostatní	DVD LG GH22NS, Cooler Master Elite 330 skříň, Seasonic Energy Knight SS 350W zdroj, MS Windows 7 Home Premium SP1 CZ 64bit OEM

Současný typický HW

- AMD Athlon II X3 450 3.2GHz
  - 3 jádra (3 paralelní jednotky)
- Intel Celeron Dual-Core E3400 2.6GHz
  - 2 jádra (2 paralelní jednotky)
- ATI HD4250
  - integrovaná grafická karta
  - 40 paralelních jednotek
    - streamové specializované jednotky
- Intel Graphics Media Accelerator X4500
  - integrovaná grafická karta
  - 10 paralelních jednotek
    - vektorové specializované jednotky

Současný typický HW

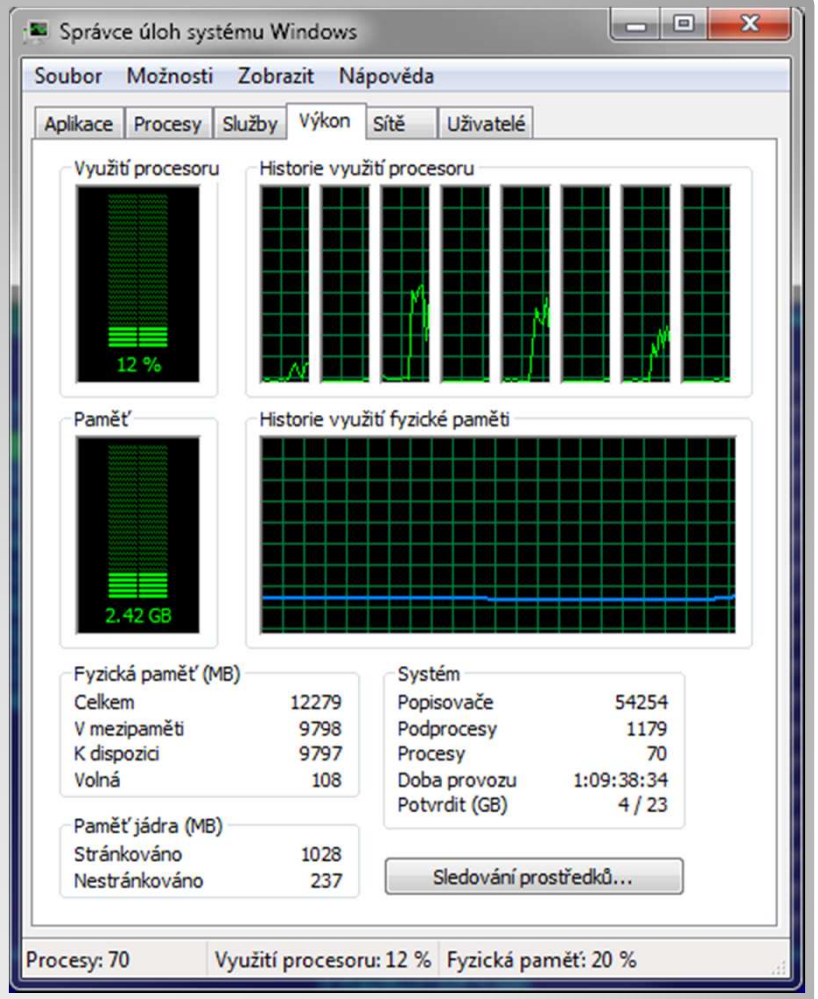
- levnější počítač pro multimédia
  - cena kolem 21 tisíc

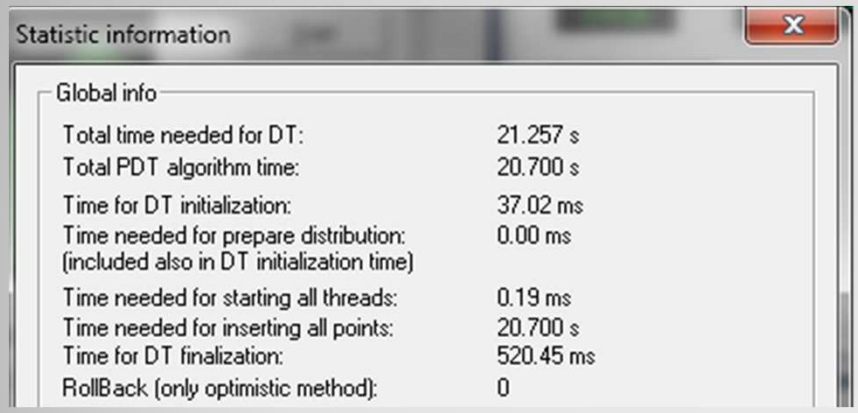
procesor	**AMD Phenom II X6 1075T 3GHz, Intel Core i5-750 2.66GHz**
grafická karta	**MSI N560GTX-Ti Twin Frozr II/OC, Sapphire Radeon HD 6870 1GB DDR5**
paměť	2×4GB Kingston DIMM DDR III 1333MHz
pevný disk	WD Caviar Black WD1002FAEX – Sata III, 1TB
ostatní	DVD LG GH22NS, CoolerMaster Centurion 534, Seasonic S12II-520 520W Bronze zdroj, MS Windows 7 Home Premium SP1 CZ 64bit OEM

Současný typický HW

- AMD Phenom II X6 1075T 3GHz
  - 6 jader (6 jednotek) pro paralelní výpočet
- Intel Core i5-2500 3.3GHz
  - 4 jádra (4 jednotky) pro paralelní výpočet
- Sapphire Radeon HD 6870 1GB DDR5
  - 1120 paralelních jednotek
    - streamové specializované jednotky
- MSI N560GTX-Ti Twin Frozr II/OC 1GB
  - GeForce GTX 560
  - 384 CUDA jader

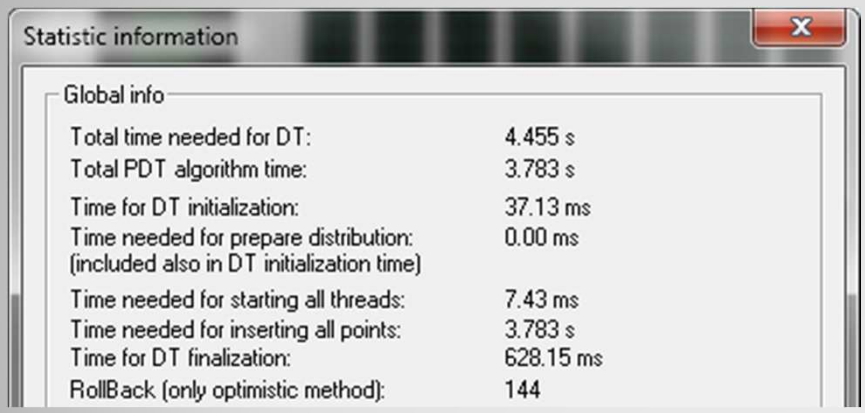
- většina aplikací zdaleka nevyužívá plný výkon CPU
  - sekvenční programy

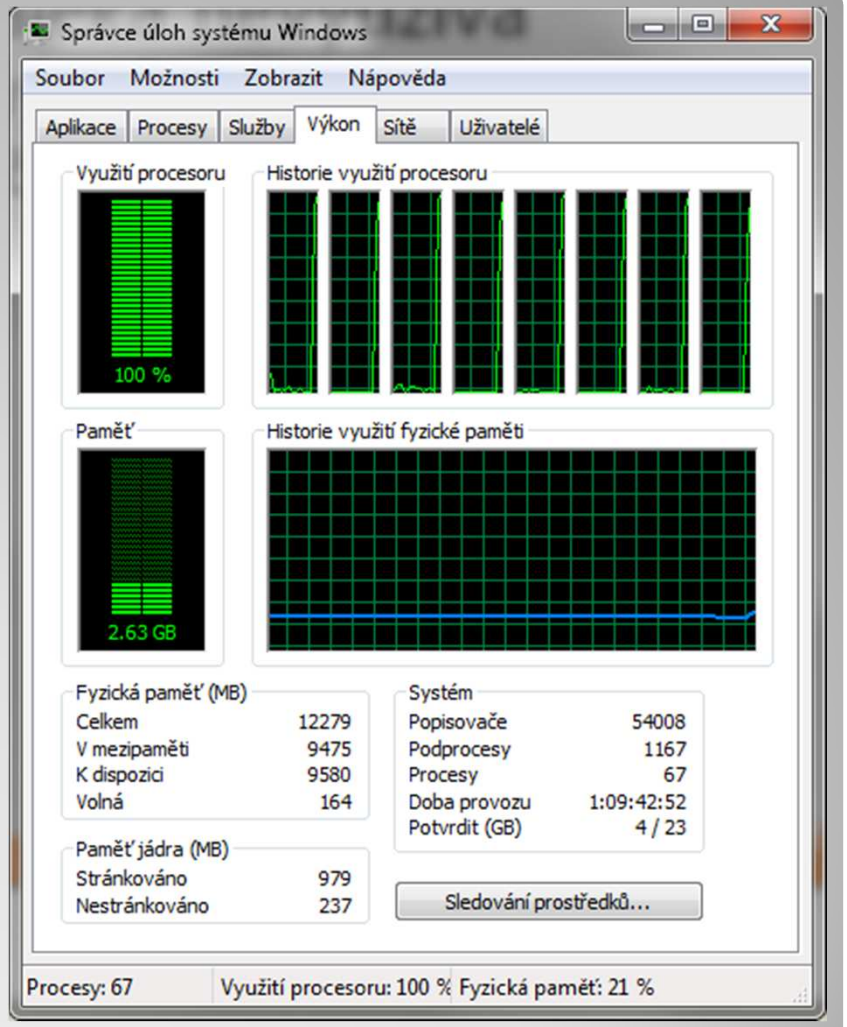




Současný typický HW

- urychlení může být významné
  - typicky menší než počet vláken





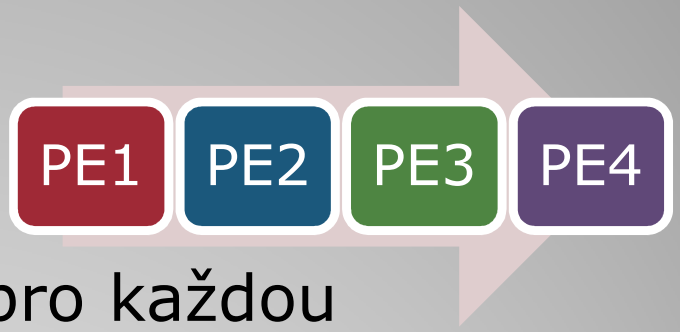
Současný typický HW

Současný typický HW

- většina aplikací buď GPU nevyužívá, nebo ho využívá prostřednictvím DirectX nebo OpenGL pro vykreslení 3D scény
  - hry, semestrálky ze ZPG
- Windows 7 využívá GPU pro veškeré vykreslování okének, totéž platí o IE9
  - do té doby se o kreslení staralo CPU

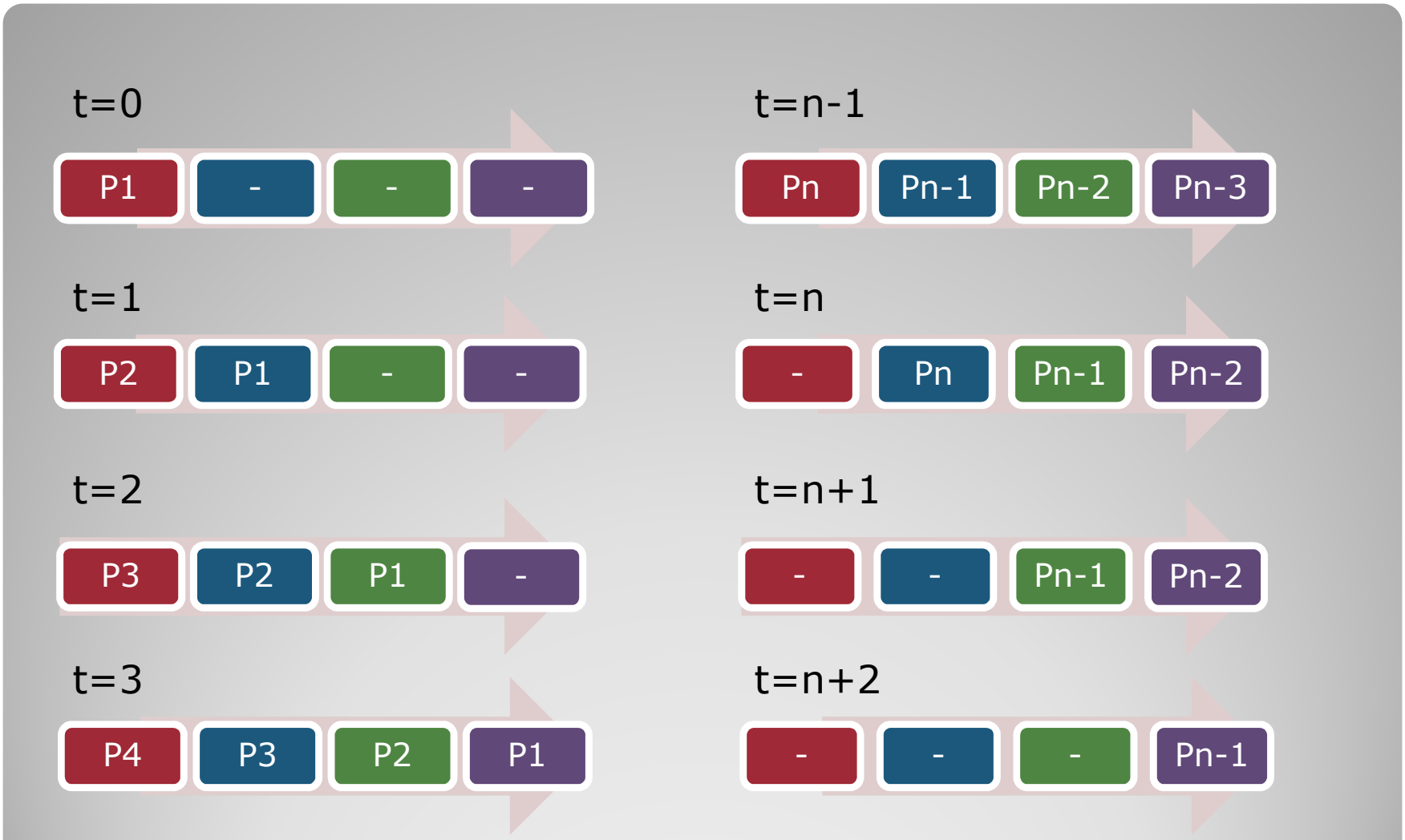
Paralelní zpracování

- funkční paralelismus
  - vhodné pro aplikace, které mají na vstupu množinu o N položkách a pro každou položku provedou algoritmus A1, nad jeho výsledkem pak algoritmus A2, nad výsledkem A3, ...
    - např. filtrace a konverze sady obrázků, komprese souborů (načtení souboru do paměti, komprese dat z paměti do paměti, uložení paměti na disk)



Paralelní zpracování

- každý algoritmus vykonáván jednou paralelní jednotkou (PE)
- když PE činnost dokončí předá výsledek dalšímu PE v procesní „rouře“
- v čase t = 0, pracuje PE1, PEx neaktivní
- v čase t = t1, PE1 zpracovává druhou položku, PE2 první položku, PEx neaktivní



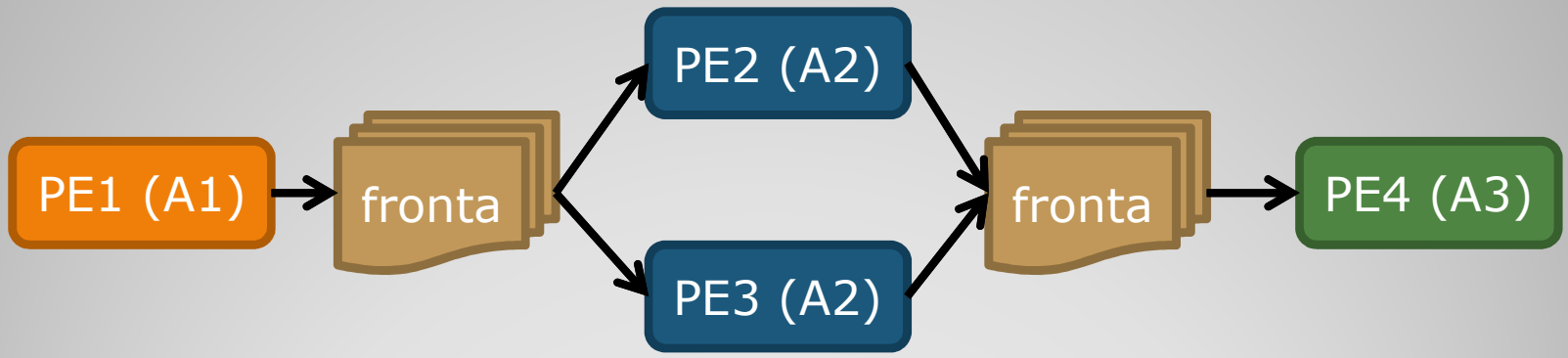
Paralelní zpracování

Paralelní zpracování

- za předpokladu, že žádné PE nemusí čekat, tak urychlení pro velká N je: ≈ k
  - k = počet PE
- problém: všechny PE musí pracovat stejně dlouho, jinak musí na sebe čekat
- možné řešení = producent – konzument
  - mezi PE existuje fronta (buffer), producent PE vloží data do fronty a pokračuje, konzument ho někdy vybere a zpracuje

Paralelní zpracování

- producent-konzument je obecná technika
  - producentů může být více (tj. např. trvá-li algoritmus A2 2x dlouho, než A3 a A1, pak A2 provozován dvěmi PE souběžně



Paralelní zpracování

- datový paralelismus
  - vhodné pro aplikace, které mají na vstupu množinu o N položkách, které mohou být „nezávisle“ zpracovávány nějakým algoritmem
  - např. nalezení extrémů, součet matic, hledání *m* položek v datové struktuře, zobrazení scény ray-tracingem
  - lepší škálovatelnost (větší urychlení)

Paralelní zpracování

- jsou-li položky zcela nezávislé, paralelizace jse velmi snadná
- závisí-li zpracování i-té položky na výsledku zpracování i-1-té položky, pak vůbec nelze paralelizovat
- obvykle něco mezi, tj. činnost PE se musí synchronizovat
  - musí se zajistit, aby závislé operace se vykonávaly ve stejném pořadí jako v případě sekvenčního procesu

Paralelní zpracování

- požaduje-li PE1 ke své činnosti přístup k datům zpracovávaným PE2, musí čekat až PE2 práci s daty dokončí
- deadlock = uváznutí – PE1 čeká na PE2 a PE2 čeká na PE1
  - cyklus může být i přes více PE
  - program přestane pracovat

- řešení deadlocku
  - program je napsán tak, aby k deadlocku buď vůbec nemohlo dojít
    - priority PE, PE s nižší prioritou nesmí čekat se zdroji uzamčenými pro sebe
  - včasná detekce
    - kdyby čekání mělo vést k deadlocku, tak nebudeme čekat a stáhneme se
      - datové zámky
      - transakční mechanismus

- řešení deadlocku
  - pozdní řešení = až to uvázne, tak se činnost nějakého vlákna (PE) násilně ukončí

Paralelní zpracování

Atomické operace

- x = 45
  - atomická operace, pokud x je int
  - je-li x double nebo long, pak už ne

```
private int[] _prvky = new int[50];
private int _index = 0;

public void AddNext(int hodnota)
{
    _prvky[_index++] = hodnota;
}
```

Atomické operace

- kód, který má běžet atomicky, je umístěn v tzv. kritické sekci
  - Java – klíčové slovo synchronized
- pro jednoduché operace podpora v CPU
  - např. InterlockedIncrement, InterlockedAdd, InterlockedCompareExchange, ...
  - vždy rychlejší než vstupovat a vystupovat z kritické sekce

Vlákna

- způsob založení vlákna (vykonává paralelní kód) v různých programovacích jazycích různý
- OS Windows – funkce CreateThread
  - parametrem je adresa funkce, která se má v novém vlákně spustit

Java – třída odvozena od Thread

```
private class CollectionAcceptor extends Thread {
… }
```

GPU

- více HW paralelních jednotek, ale jsou specializovány, nelze na nich vše
  - přesnost 32-bitový float
  - nelze pořádně pro závislá data
  - vhodné pro vektorové operace
    - maticový počet, vyhledávání, intepolace, filtrování obrázků, holografie ...
- speciální programovací jazyky
  - DirectX: HLSL, OpenGL: GLSL,
  - nVidia: Cg, CUDA
  - ATI (AMD), nVidia: OpenCL

- princip zcela odlišný:
  - typicky se specifikuje 2D rozlišení řešení
    - tj. počet vláken – bývá větší, než počet HW
  - data předávána texturou
  - načte se speciální program do gr. karty
  - provede se „vykreslení“
  - výsledek přečte do hlavní op. paměti

GPU

- checksums, kryptování

Příště