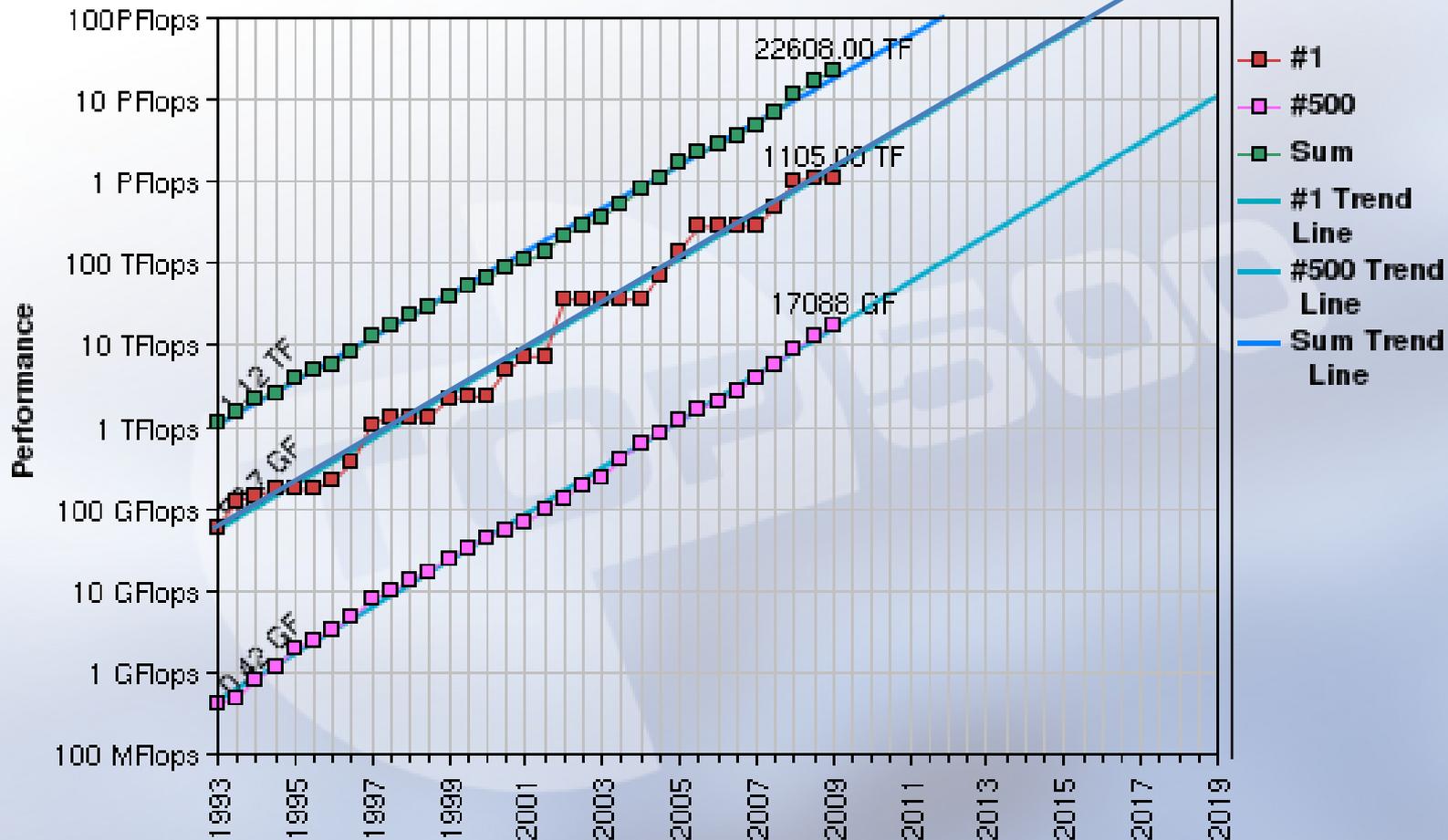


# GPGPU

西澤誠也

神戸大学

1 E(エクサ)Flops



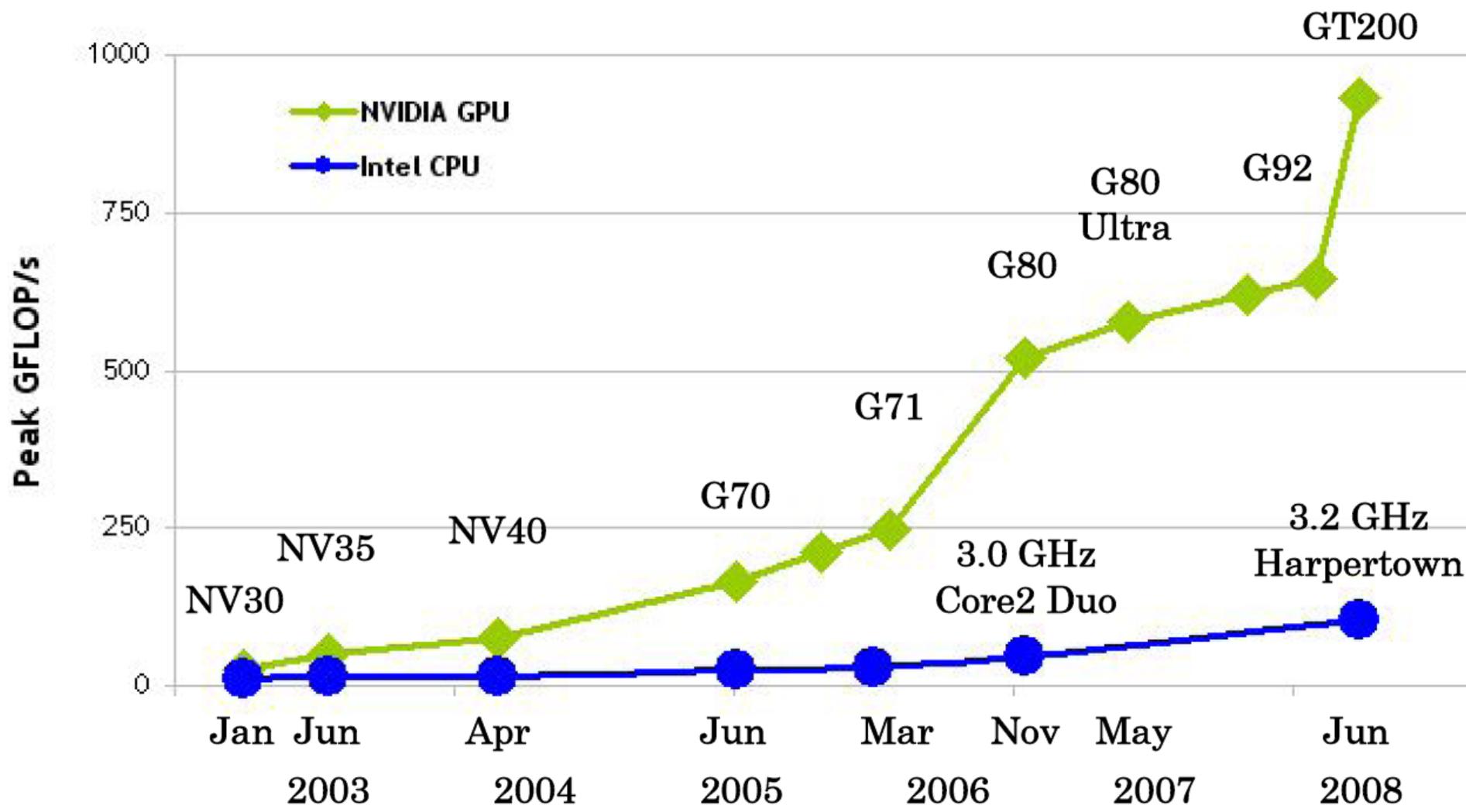
19/06/2009

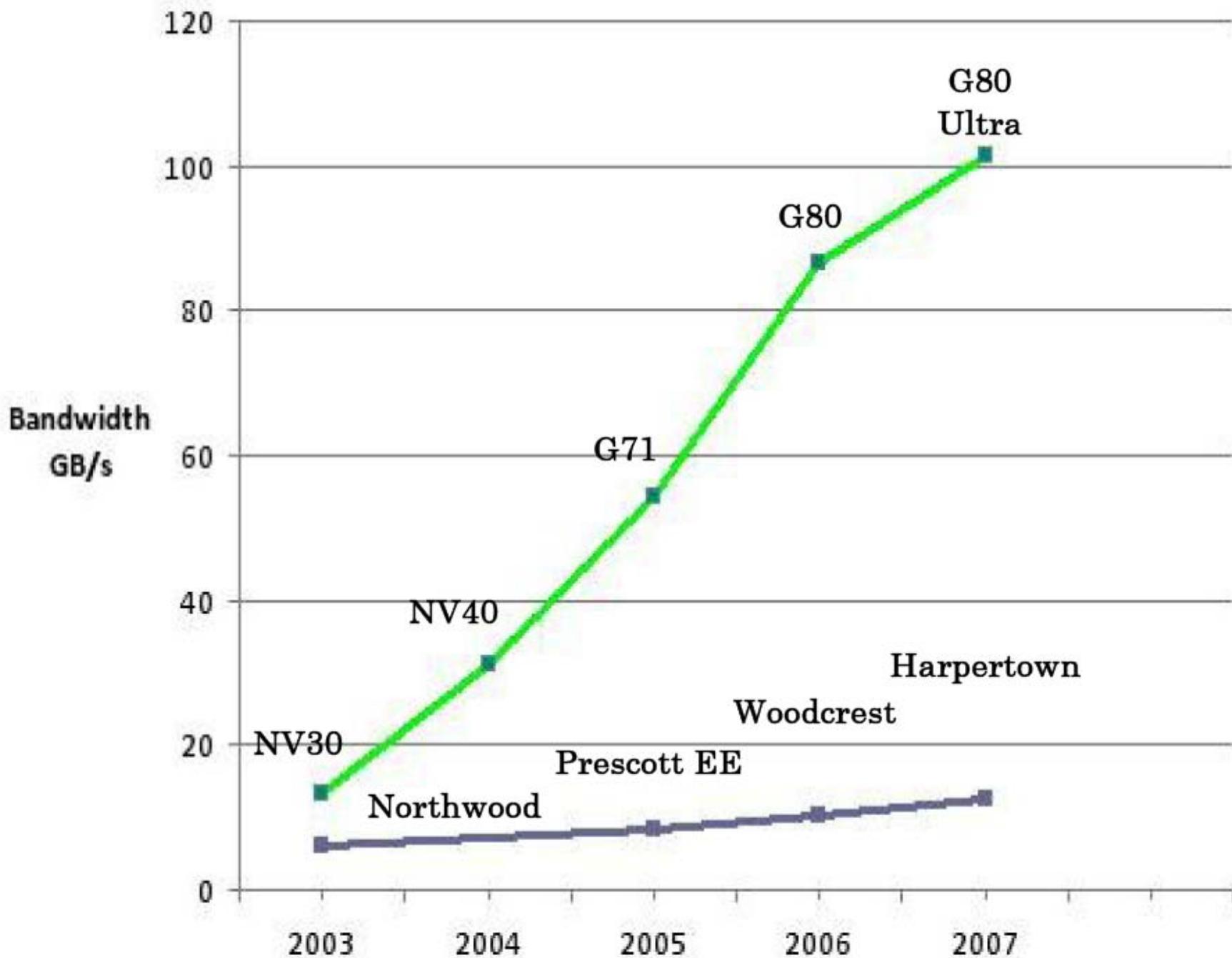
<http://www.top500.org/>

- このペースを保ったまま(or さらに加速して), 性能が伸び続けるか
  - 電力
  - メモリバンド幅
- 我らが(?)ベクトル機は....
  - 京速計算機もスカラー(富士通) 一本になったし
- そもそもスパコンは手軽ではない
  - 怪しい研究できないし

そこで

**加速器**





# What's GPGPU

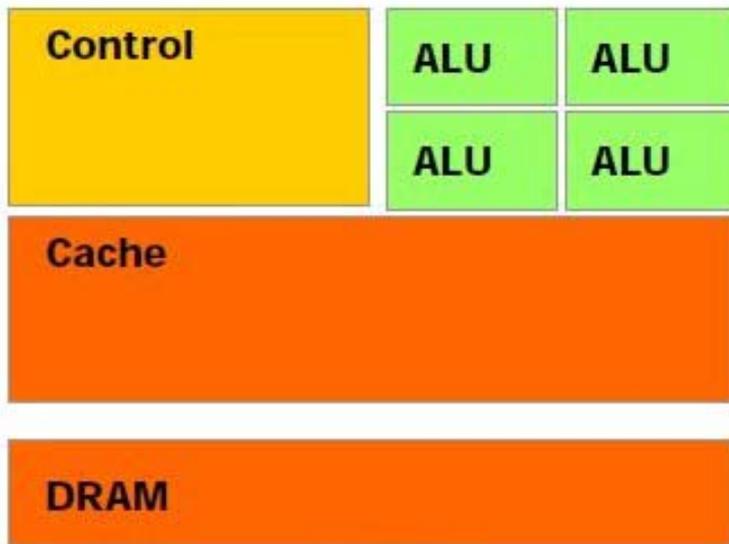
- General Purpose programming on GPUs
  - GPU: Graphical Processing Unit
    - 画像出力のためのデバイス
  - 画像処理以外の目的にGPUを利用する

# GPU の特徴

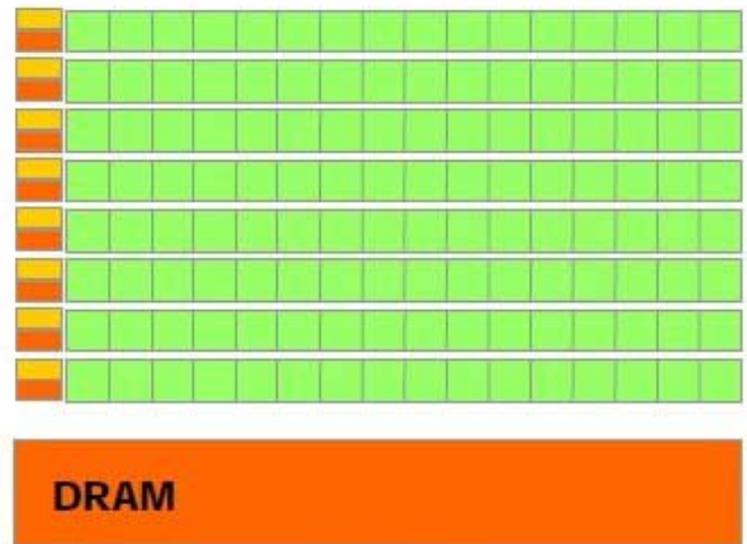
- 価格性能比が安い
  - 約10万円で 1TFlops
- 電力性能比が低い
  - 200W弱で1TFlops
- 超並列
  - メニーコア (数百)

# GPU vs CPU

- CPU
  - contorler, cache が大きく、汎用プログラム向き
- GPU
  - 演算器が大量にある



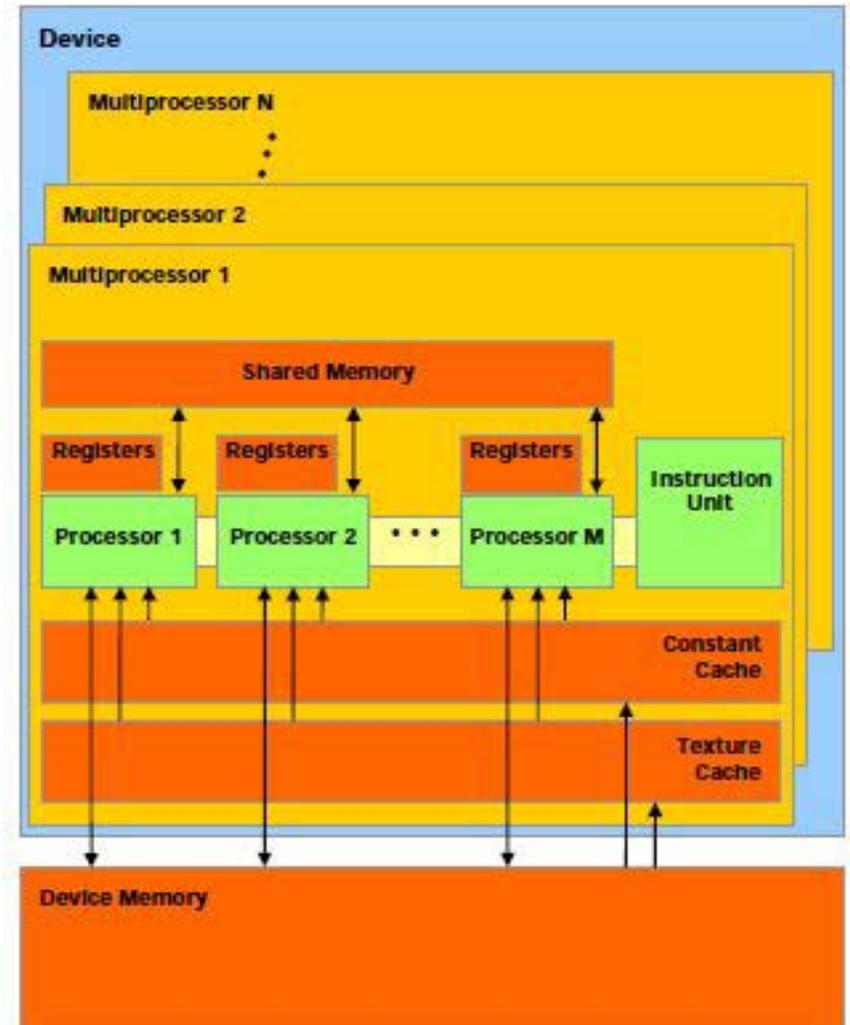
**CPU**



**GPU**

# GPU の構造

- 複数のメモリ階層
  - ホストとは別の独自のメモリをもつ
  - アクセスできる演算器の範囲が異なる



# 開発・実行環境

- CUDA
  - NVIDIA社が無償配布しているGPGPU開発/実行環境. 同社製のGPUカード専用.
- ATI Stream
  - AMD社が無償配布しているGPGPU開発/実行環境. 同社製のGPUカード専用.
- OpenCL
  - Apple社が提唱し, KHRONOSグループによりまとめられた標準API仕様. 対象として, GPU, CPU, Cell, 組み込み機器などを想定.
- DirectX Compute Shader
  - Microsoft社が開発中のGPGPU開発/実行環境. 同社製のOS上でのみ動作.

- 以上は C/C++ で実装する
  - 結構、大幅な変更が必要
  - 速度を追求するためにチューニングしようとする、ある程度ハードウェアの知識が必要
- 最近 PGI コンパイラーが、C/Fortran のコードに指示行をいれるだけで、CUDAのコードを内部で生成し、実行バイナリを作成してくれるものをリリースした (コンパイラのエクステンション)
  - ある程度速くなるが、最速のものはむずかしい
  - お手軽に、それなりの速度

# 紹介ムービー (by 爆発研究所)

[http://www.youtube.com/watch?v=WpunUewZ4p8&feature=player\\_embedded](http://www.youtube.com/watch?v=WpunUewZ4p8&feature=player_embedded)

# 注意点

- 現状では、倍精度計算は単精度に比べると遅い
  - 単精度の  $1/12$  (NVIDIA),  $1/5$  (ATI)
  - そもそも、グラフィック用演算で倍精度の精度は必要ない
  - 次世代GPU は、倍精度は単精度の半分(データ量が2倍なのでフルの性能)まで上がる

# 2次元乱流実験 with GPGPU

- $\beta$ 平面順圧渦度方程式
- 実験設定
  - 境界条件 : 2重周期境界
  - 空間差分 : スペクトル法 (切断波数 682)
  - 時間積分 : 4次のルンゲクッタ ( $\Delta t = 1.0e-4$ )
  - 初期エネルギー : 波数 49-51 にランダムに

# CPU計算 vs. GPU計算

## 100時間ステップ

- CPU (Fortran)
  - Intel Core2 Quad 9550
  - 417 sec
  - 倍精度
- GPU (CUDA)
  - NVIDIA Tesla C1060
  - 10 sec
  - 単精度

**40倍の実行速度!**

- 興味がある方は GPGPU勉強会に是非参加してください
  - [gpgpu@itpass.scitec.kobe-u.ac.jp](mailto:gpgpu@itpass.scitec.kobe-u.ac.jp)