

# 生体分子シミュレーション におけるGPGPUの活用

※11 Mar 2010 修正版

株式会社クロスアビリティ  
代表取締役 古賀良太  
rkoga@x-ability.jp

- GPGPUの概要
- 生体分子シミュレーションにおけるGPGPUソフト
- GPGPUの特徴
- CPU-GPGPU Hybrid Execution
- 量子化学計算へのGPGPUの活用
- FMOの製薬分野での応用
- オリジナル研究の背景、アイデア、実装、結果、まとめ
- オリジナルGPGPUアプリケーション
  - GAMESS FMO
  - エネルギー表示法を用いた高速自由エネルギー計算(ER)
- GPGPU対応済の量子化学計算ソフト
- Fermi
- Fermi以外の有望アクセラレータ
- その他アクセラレータ
- GPGPU関連書籍
- 最後に会社紹介



# GPGPUの概要

- General Purpose GPUの略で、GPU(グラフィックカード)をHPC向けに用いること、およびそのカード
- 製品名はTesla(GPGPU), GTX285(GPGPUに使えるGPU)等
- 普通にPCのPCI ExpressバスにGPUを買ってきて挿し、GPGPUドライバを入れるだけで使える
  - 費用対効果が高い(一枚数万円程度)、GPGPUの唯一の存在意義ともいえる
- AMDとNVIDIAのカードがあるが、どちらもHPC向けにプログラミングできるBrook+とCUDAがある
  - CUDA環境の方が現時点では扱いやすく情報も多い(後述するオリジナル研究でもCUDAを採用)
  - OpenCLという統一プラットフォームがAMD GPGPU, NVIDIA GPGPU, およびCELLプロセッサなどに適用できる

# 生体分子シミュレーションにおけるGPGPUソフト

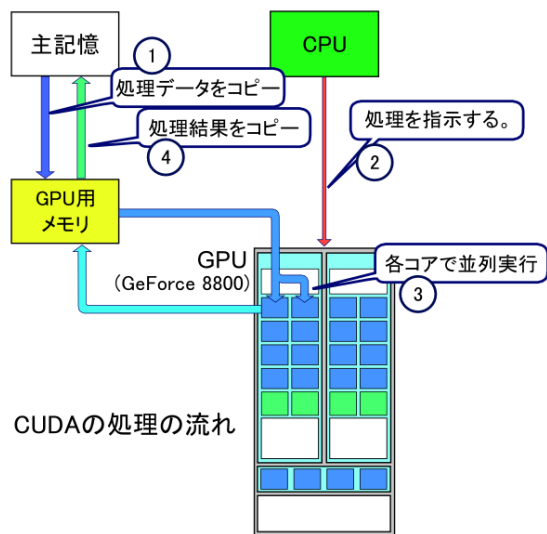
- Molecular Dynamics
  - GPGPU加速化が比較的容易なためいち早くGPGPU実装が行われた分野
  - 対応ソフト(NVIDIA社のウェブサイト参照)
    - Gromacs, NAMD, LAMMPS, etc.
- 量子化学計算(詳細は後述)
  - TeraChem
  - BigDFT

※本発表では今後の活用が期待されている量子化学計算を中心に扱う

# GPGPUの特徴1

- SIMT (Single Instruction Multi Thread)
- 高速だが容量が小さいon-chipメモリと低速だが容量が大きいexternal memoryによる構成
- 倍精度演算が単精度演算より12倍遅い

NVIDIAのGPGPUはCUDA環境で扱える(C言語に類似)。

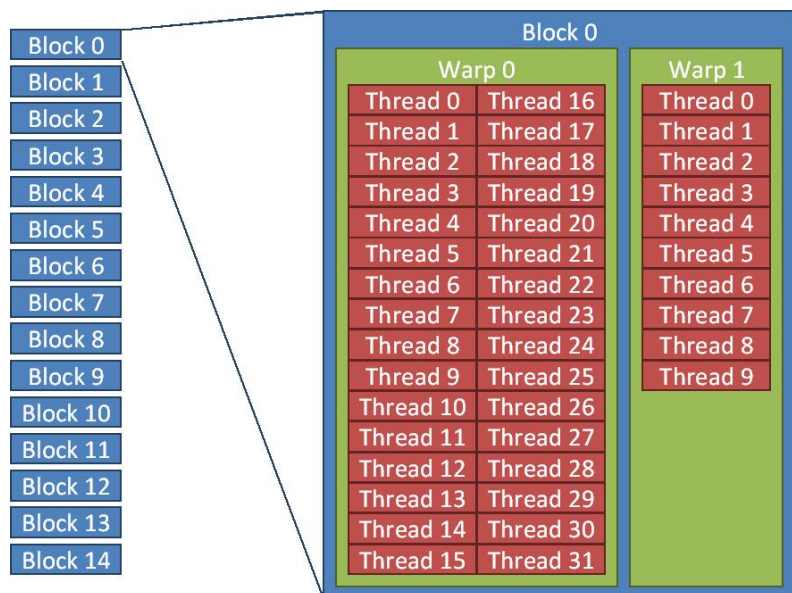


## 処理命令の例

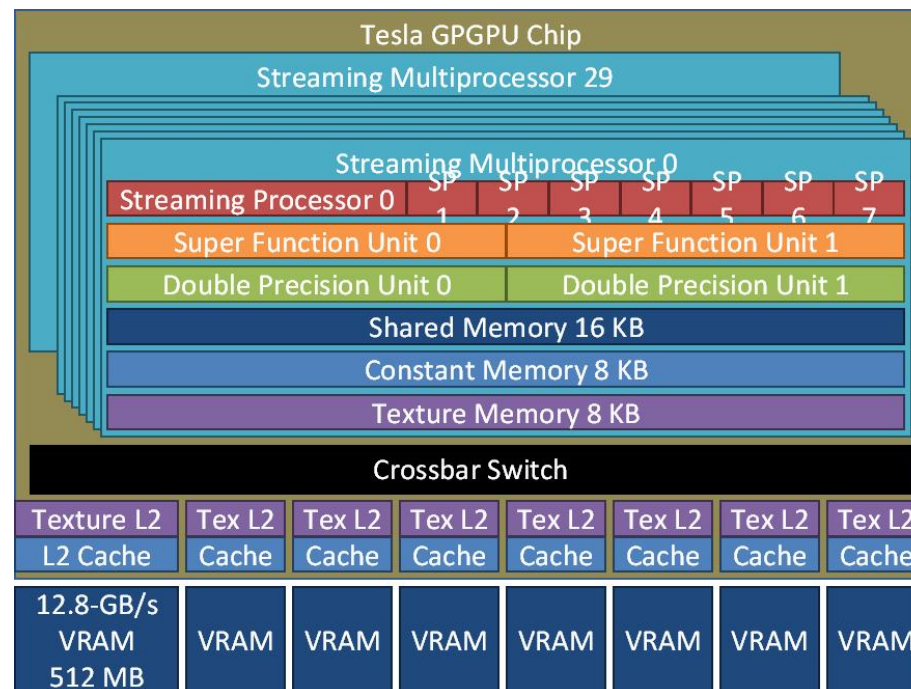
- ① `cudaMalloc, cudaMemcpy`
- ② `MatMulKernel<<<dimGrid, dimBlock>>>(d_A, d_B, d_C);` ※GPU側で実行されるプログラムがkernel
- ③ `__global__ void MatMulKernel(float* A, float* B, float* C){.....}`
- ④ `cudaMemcpy, cudaFree`

Wikipediaより引用

# GPGPUの特徴2



CPUスレッドから起動するkernel(grid, block)の中のblockの説明。GPGPU Threadはblockの中のWarpという単位でスケジューリングされ、Warp内のthreadは同じ命令を実行する(SIMT)。多くのWarpを使うことでメモリ遅延を隠蔽できる。32threads単位で動く。



GlobalメモリはOff-chipなので、Global memoryからchipに転送する(CPU-GPU転送コストよりはダイブ低い)。

※量子化学計算(二電子積分)においてはshared memoryのサイズが問題になる

# CPU-GPGPU Hybrid Execution

GPGPU(device)が処理を担当している間に遊んでしまっているCPU(host)を活用する。

- GPGPUカーネルを呼び出すCPUスレッドとCPUのみで処理するCPUスレッドを生成する(GPGPUで加速できる処理が多い並列化プログラム)。
  - オリジナル研究で用いた方法
  - GPUの方が速いためCPU処理の待ち時間が発生
- GPGPUカーネルを呼び出す関数を作っておいて、GPGPU処理を担当するCPUスレッドにジョブを投げる処理を行う。
  - GAMESS on GPGPUで用いている方法
  - 待ち時間の程度はCPUとGPGPUの処理性能比による

# 量子化学計算へのGPGPUの活用

- 座標データは全電子タンパクでも一括でGlobal memoryに転送可
- 分割した各フラグメント処理はフラグメント間の依存関係が少ないため、CPUとGPGPUに適切にスケジューリングして割り当てることで、無駄なくCPUおよびGPUを活用し、トータルのスループットを向上させられる可能性がある。
- 全電子計算をするのではなく、GPGPUによるFock行列対角化の加速が十分に得られる最も大きいサイズのフラグメントに分割したFMOが精度および加速率の両方を考慮した最適な計算である。
  - GPGPUによる加速率は分子サイズが大きい方が大きいですが、Fock行列の対角化のコストが上がる。
  - 対角化のメモリ問題が存在する(対称疎行列などの利用等で次元↓)
- 複数GPUスレッドで1つの二電子積分を計算するようにすれば、計算過程で出てくる変数を格納するshared memoryを節約でき高速化が図れる。

# FMOの製薬分野での応用

- 低分子化合物
  - FBDD screeningにPIEDAによる相互作用の構成解析, PIO解析(1フラグメント1残基)
  - 未来はVirtual screeningにFULL-QM screeningを導入
- 抗体医薬
  - バイシクルペプチド抗体など将来の化学合成の可能性 (c.f. 日経バイオテク)、低分子化合物と同様の計算
- 核酸医薬
  - アプタマーは化学合成可能
  - 核酸とタンパクのcomplex計算

# オリジナル研究：概要

## Acceleration of Fragment Molecular Orbital Method Using CPU-GPGPU Hybrid Execution and MPI

Ryohei Nishimura  
The University of Tokyo  
7-3-1 Hongo, Bunkyo, Tokyo,  
Japan  
nishihai@is.s.u-  
tokyo.ac.jp

Ryota Koga  
X-Ability Co., Ltd.  
801 Hongo Condominium,  
3-16-6 Hongo, Bunkyo, Tokyo,  
Japan  
rkoga@x-ability.jp

Yuki Furukawa  
X-Ability Co., Ltd.  
801 Hongo Condominium,  
3-16-6 Hongo, Bunkyo, Tokyo,  
Japan  
furukawa@x-ability.jp

Kei Hiraki  
The University of Tokyo  
7-3-1 Hongo, Bunkyo, Tokyo,  
Japan  
hiraki@is.s.u-tokyo.ac.jp

### ABSTRACT

The fragment molecular orbital (FMO) method is used for computing electronic states of protein complexes of receptors and ligands such as change transfer and polarization which are not able to be considered by molecular mechanics (MM) and molecular dynamics (MD). FMO consists of calculations of electronic state and perturbation. And the resolution of the identity second-order Møller-Plesset perturbation theory (RI-MP2) is a method to calculate perturbation. We implemented a GPGPU (General-Purpose computation on Graphics Processing Unit) application based on both FMO and RI-MP2 to solve perturbation using CUDA (Compute Unified Device Architecture) and accelerated the calculation of energy of complex 1FKF 45.2 times as fast as GAMESS. Electronic states

in-silico screening using FULL-QM calculation from millions of compounds to thousands of ones named "FULL-QM screening" will achieve with reasonable cost.

To develop low-molecular compounds, there are three main screening methods, which are High Throughput Screening (HTS), Fragment Based Drug Design Screening (FS) and Virtual Screening (VS). After these screenings, organic synthesis starts. HTS is the automatic mechanical screening without QM for target to million compounds. FS is the stepwise solution using X-ray, Surface Plasmon Resonance (SPR), Nuclear Magnetic Resonance (NMR) and various calculation including QM. VS is full QM calculation. In-silico screening is used in FS, VS or both. In-silico screening in

1/22にFMOによるエネルギー2次摂動の計算(RI-MP2)をCPU-GPGPU Hybrid Execution & MPIで試みた論文をACM International Conference Proceedings Seriesにsubmitした。

電子状態計算(Hartree-Fockなど)はGPUで現時点で十分に加速できていない(特にJ Matrix Engineを使えないK-matrixの実装部分)。

※2010/3/11追記

内容を修正し化学系雑誌に変更して再submit中

# オリジナル研究：アイデア

- FMOは計算対象を分割できるため、演算の独立性が高い。これより、CPUとGPGPUに各フラグメントの量子化学計算をランダムに割り当てることでCPUとGPUの同時計算 (CPU-GPGPU Hybrid Execution) がトータルスループットを向上させるのではないかな？
- 量子化学計算は二電子積分の演算にコストがかかるが、計算対象を分割してサイズを落とすことで、一度にGPUの各種メモリにおけるデータの広がりがあるのではないかな？

# オリジナル研究：実装1

- 対象はFK506結合タンパク(complex, receptor, ligand)
- Hartree-Fockで計算すべき電子状態計算(エネルギー & 軌道係数)はGPU向きの実装がうまくいっていないため、GAMESSの出力を用いた。
- Conventional MP2ではなくRI-MP2を用いた。
- CPU-GPGPU Hybrid Execution
  - 全体のGPU処理部分は50%だが、CPUスレッドからGPGPUカーネルを呼び出す処理となっているのは最適化不足(GPGPU処理を担当するCPUスレッドにジョブを投げる方式がよりベター)
- Full-CDも試みた
- CULA(CUDA LAPACK)も試みた

# オリジナル研究：実装2

RI-MP2に関する実装

$$\langle \mu\nu | \lambda\sigma \rangle = \sum_{PQ} \langle \mu\nu | P \rangle \langle P | Q \rangle^{-1} \langle Q | \lambda\sigma \rangle \quad \longrightarrow$$

行列

CPUで二電子二中心積分、  
GPGPUで二電子三中心積  
分を計算 ※二電子三中心  
積分が全てGlobal memory  
に乗ったのが高速化の肝

$$\langle P | Q \rangle^{-1} = L_{PQ}^T L_{PQ} \quad \longrightarrow$$

逆行列計算およびcholesky分解のGPGPUによる高  
速化を期待してCULA(CUDA LAPACK)で行ったが、  
精度が足りなかったのでCPUで計算

$$B_{\mu\nu Q} = \sum_Q \langle \mu\nu | P \rangle L_{PQ}^T \quad \longrightarrow$$

GPGPUでCUBLASによる行列積演算

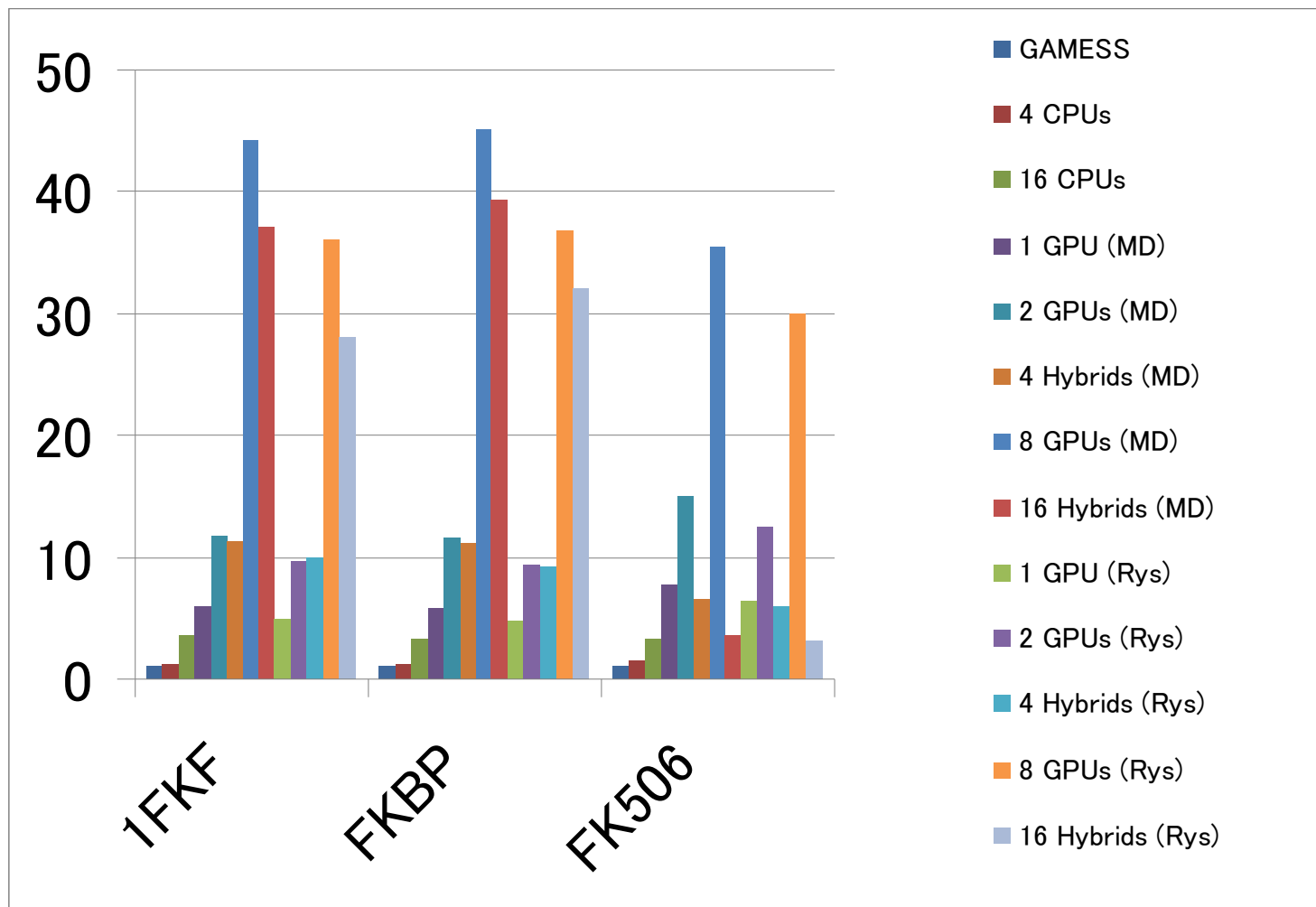
$$\sum_Q B_{\mu\nu Q} B_{\lambda\sigma Q} = \langle \mu\nu | \lambda\sigma \rangle \quad \longrightarrow$$

GPGPUでCUBLASによる行列積演算

$$\because C = A \times B \approx C_{ik} = \sum_j A_{ij} B_{jk}$$

※時系列としては、転送コストを抑えるため、CPUで  
計算できることを終えた後に一度だけCPU-GPGPU  
転送を行い、GPGPUで計算し、またCPUに戻す

# オリジナル研究：結果



摂動部のみ  
(RI-MP2)

GAMESSを  
1とした速度  
比の比較

MD, Rysは  
GPUで計算  
した二電子  
三中心積分  
のアルゴリ  
ズム

# オリジナル研究：まとめ

- HybridでないCPUとGPGPUの分担計算においては、16CPU-8GPGPU 4ノードの環境において、1FKF(complex)がCPU単独計算の45.2倍加速した
- 並列化効率が高くとも小さい問題サイズに対してはCPU-GPGPU Hybrid Executionは有効ではない
- 精度に関して、Rysが良く、MDは悪く、Hybridは低下させ、MDのHybridが一番悪い(単精度と倍精度が混ざる)

# GAMESS on GPGPU

- 名大・安田准教授とGPGPUモジュールを共同研究開発中
  - Koji Yasuda, *Journal of Computational Chemistry*, 29, 334, Published Online: 5 Jul 2007
  - Koji Yasuda, *Journal of Chemical Theory and Computation*, 4, 1230-1236, July 2008
- FMOの加速が目的
- GPGPU処理を担当するCPUスレッドにジョブを投げる方式でCPU-GPGPU Hybrid Executionを実行
- GPGPUモジュールをAdd-on (binary)にlink)
- ノードロックライセンスで提供予定
- GPGPUモジュールとGAMESSを結合するI/Fはクロスアビリティ製(フリー)
  - ※Gaussian用のI/Fは某社製を使用

# エネルギー表示法を用いた高速自由エネルギー計算

溶媒和自由エネルギー ※京大・松林  
准教授と共同  
研究開発中

ナノスケール不均一系での  
自由エネルギー解析

脂質膜・ミセルへの薬物の取り込み  
(結合量と結合位置の予測)

現実の使用  
に耐える  
正確さ

エネルギー表示  
によって  
高精度を達成

近似汎関数

分布関数(相関関数)

※エネルギー  
分布関数生成  
のための溶媒・  
溶質二体相互  
作用エネルギー  
計算がGPGPU  
で高速化可能

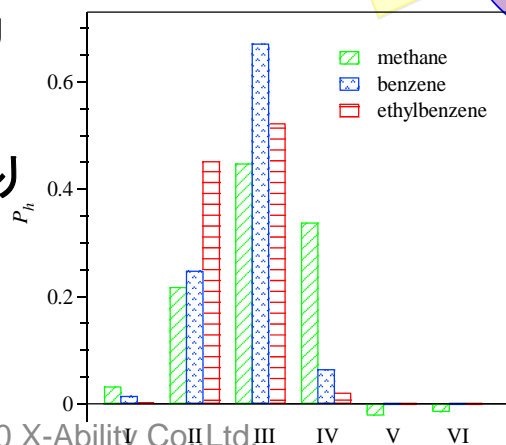
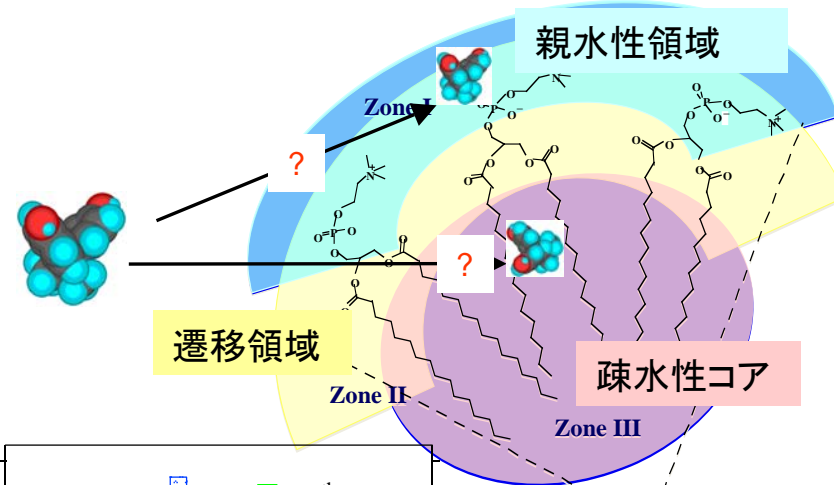
容易に  
計算可能

MD

(典型的に)

分子間相互作用

※FEPと異なり  
純溶媒系と溶  
液系のための  
MD計算



計算によるSDSミセルへの薬物結合位置の同定

# GPGPU対応済の量子化学計算ソフト

- TeraChem
  - StanfordのMartinez教授の論文をベースにしている  
<http://www.petachem.com/products.html>
  - PetaChem, LLCよりベータ版ダウンロード可能
  - CPUのコア数変えても速度があまり変わらないことから、ほとんどがGPUで実装されているようである ※CUDAで実装されている
  - タンパクの全電子状態計算をうたってるようだがGPGPUに不向き？
    - 大量のGPUクラスターを並列化してGPGPUカーネル並列化で対応？
  - 速度はX-Ability実装のGPGPUカーネルと同程度(現状でK-MatrixおよびXCのグリッド生成がGPGPU対応していると思われる点は進んでいる)
- BigDFT
  - Wavelet基底
    - 基底関数が大きい極限に近づき易い利点がありそうである
    - 対角化が大変そうな気がする、gridを使うので通信面でGauss基底より相当不利な気がする
  - GNU General Public License (GPL)

# Fermi

- 次回リリースされる予定のNVIDIA製GPU
- 既存GPGPUと比較して以下のような改善があるとされる
  - 倍精度計算が高速になる (Teslaの8倍、666GFLOPS)
    - 精度が必要な処理が改善
  - ECCが搭載される
    - 長時間計算に関する信頼性向上 (screeningなど)
  - Kernel実行が並列化される
    - 今のdevice (Teslaなど) は1つのkernelしか同時実行できないが、次々とKernel実行を試みたときに、余ったstreaming multi-processorのリソースが割り当てられる
  - Shared memoryサイズの向上
    - 二電子積分などの途中処理で生成する変数を格納するスペースが増えるため、高次元角運動量軌道やK-matrixの担当するスレッドを減らすことができる

# Fermi以外の有望アクセラレータ

- AMD GPU
  - ATI Radeon HD 5970の倍精度(理論性能)は928 GFLOPS
    - 品薄中
  - ATI Radeon HD 5870の倍精度(理論性能)は544 GFLOPS
  - Radeon 5000シリーズからOpenCLへの最適化がなされている
  - 値段がNVIDIAと同様に安い(数万円)
- GRAPE-DR ※GPGPUではない
  - 倍精度(理論性能)は 512GFLOPS
  - コンパイラが整備されていない
  - 量子化学計算(密度汎関数計算)は名大・安田准教授が実装  
<http://www.human.nagoya-u.ac.jp/~yasudak/QCGPU-3.html>
  - 値段が少々高い(数十万円)

# その他アクセラレータ

※以下、GPGPUではない

- CELL
  - OpenCLまたはCELL SDKで記述可能
  - GPGPUに勝てないが、プレイステーション3の並列化であれば比較的価格性能比が良い
  - 電力効率であればGPGPUよりも良いかもしれない
- FPGA
  - ゴードンベル賞の長崎大・濱田助教が取り組んでいたが、GPGPUの方が高速だとして乗り換えた経緯がある
    - 一部計算が二電子積分のssに類似しており量子化学計算もFPGAでは低速であると予想される(乗算・加算の演算がGPGPUの1/5)

# GPGPU関連書籍

- 『はじめてのCUDA』
  - 流体が専門の東工大・青木教授が執筆
  - 分かりやすい
  - GPUコンピューティング研究会でも同様のハンズオン講義が行われている
- 『OpenCL入門』
  - 株式会社フィックスターズが執筆
  - NVIDIAのOpenCLコンパイラが異常に遅いという結果が示されている
    - NVIDIA GPUに関してはCUDA以上の最適化は難しい
- 『GPU Gems 3』
  - 最初に日本語で出版されたGPGPU関連書籍

# 最後に会社紹介

- TSUBAMEユーザー(2010年産業利用・成果公開)
- 計算科学事業とセンサネットワーク事業
  - 計算科学事業は主に化学計算の高速化およびコンサルティング、サーバ販売協力など
  - センサネットワーク事業は、海外を中心に農地のセンサデータをインターネットにルーティングするフィールドルータの開発・販売・設置・保守
- 役員3名＋アルバイト・顧問4名
- 東大・赤門から南500mに自宅兼事務所
- 製薬会社研究員やベンチャーキャピタルと最近よく話しています

以上