

GPUを利用した 高速事象生成システムの開発

神前 純一 (KEK)

萩原 薫 (KEK)、岡村 直利 (山梨大学)

日本物理学会 第67回年次大会@関西学院大学

2012年3月26日

これまでの成果

- 目的

Graphics Processing Unit (GPU) を物理プロセスの事象生成のソフトウェアに利用し大幅な性能改善を実現する

- これまで達成されたこと

- 散乱振幅の計算 (HELAS/Heget、位相空間生成)
- MC積分 (VEGAS、BASES)
- 事象生成 (SPRING)
- 簡易測定器シミュレーション・プログラム

MadGraph4 の生成するFORTRANプログラムを元にGPUプログラムを生成した

-> 20-100倍の性能改善効果

Publications

- QED: K. Hagiwara, J. Kanzaki, N. Okamura, D. Rainwater and T. Stelzer, Eur. Phys. J. C66 (2010) 477, e-print arXiv:0908.4403;
- QCD: K. Hagiwara, J. Kanzaki, N. Okamura, D. Rainwater and T. Stelzer, Eur. Phys. J. C70 (2010) 513, e-print arXiv:0909.5257;
- MC 積分: J. Kanzaki, Eur. Phys. J. C71 (2011) 1559, e-print arXiv:1010.2107.
- SM: K. Hagiwara, J. Kanzaki, Q. Li, N. Okamura and T. Stelzer, almost finished.

次の課題

- MadGraph5 への対応
- Single process から Multi subprocesses へ

ハドロンコライダーでの多ジェット生成過程では、多くの subprocess を同時に生成する必要があり、sub-process の混ざった事象が自動的に生成されなければならない

-> 多ジェット生成過程の事象生成の効率化は、QCDジェット生成過程のシミュレーションの効率的な改善につながり、QCD からの background の系統誤差の縮小が期待できる

MadGraph5

- Version 5 になり大きな変更がなされた
 - 散乱振幅計算プログラムが自動生成され、新しい物理への対応が柔軟になった (従来の HELAS)
C++の code は GPU 用の code に少しの変更で応用可
 - Sub-process の扱い方の変更
Equivalent な sub-processes がまとめられ、事象生成効率が非常に改善された

従来の MadGraph/MadEvent -> GPU の変換システムを対応させる必要

開発項目

- MG5 コードのGPUへの変換システムの開発
 - Feynrule+UFO interface -> ALOHA により自動的に散乱振幅計算パッケージが生成される
 - FORTRAN、C++、Python で出力可
 - C++用に出力した code を従来の heget (GPU用HELAS) の代わりに使う
-

- 全 sub-process 用の GPU code の一括自動生成
- 生成条件の MadEvent からの自動生成
- 全 system の自動実行

計算環境

- **ホストPC:**

CPU	Core i7 2.67GHz
L2 Cache	8MB
Memory	6GB
Bus Speed	1.333GHz
OS	Fedora 10 (64bit)

- **Software:**

nvcc	Rel.4.0 (V0.2.1221)
CUDA Driver	Ver.4.0
CUDA Runtime	Ver.4.0
gcc/gfortran	4.4.5 (Red Hat 4.4.5-2)

計算環境

- **NVIDIA GeForce GTX580**

従来使用していた GTX285 に比較し、ほぼ2倍の実行速度

	GTX580	GTX285
Multi Processor	16	30
Total SP	512	240
Global Memory	1.5GB	2GB
Constant Memory	64KB	64KB
Shared Memory/block	48KB	16KB
Registers/block	32768	16384
Warp Size	32	32
Clock Rate	1.54GHz	1.48GHz

MG5 amplitude の性能

- ALOHA により生成された C++ code をわずかな変更で新しい heget としてGPUで利用できた

ud~ \rightarrow $W^+(\mu+\nu)$ + n-gluons で性能を比較

phase space 生成 + amplitude 計算 の平均時間 (μ sec)

n-gluons	heget (old)	heget (ALOHA)
0	0.018	0.018
1	0.023	0.024
2	0.031	0.030
3	0.080	0.072

大きな amplitude の場合、わずかに高速になる可能性がある

計算時間の概算

- pp \rightarrow $W^+(\mu\nu)+3\text{-jets}$ を用いて計算時間を比較
 - lepton: $|\eta| < 2.5$, $p_T > 20\text{GeV}$
 - jet: $|\eta| < 5$, $p_T > 20\text{GeV}$
 - 10000 events の生成時間
- MadGraph/MadEvent V4
- MadGraph/MadEvent V5
 - Compile を含まない実行時間を比較
 - 4 cores を並列使用
- GPU (GTX580)
 - Single process の結果の積算

計算時間の概算

• MG5 & MG4 (sec)

n-gluons	MG5		MG4	
	subprocess	time	subprocess	time
0	1	12	2	19
1	2	19	6	78
2	4	60	41	620
3	8	450	73	4100

MG5 の効率改善の効果は大きい

計算時間の概算

• GPU (GTX580) (sec)

n-gluons	gBases	gSpring	Sum	x sub-process	MG5
0	0.2	0.1	0.3	0.3	12
1	0.3	0.1	0.4	0.8	19
2	0.4	0.2	0.6	4.8	60
3	1	3	4	32	450

全自動化により大きな効率の改善が期待できる

まとめと課題

- 新しい MadGraph/MadEvent (version 5) に対応させた GPU code 生成システムを開発した

MG5 が出力する新しい heget (helicity amplitude 計算用 function) を作成した -> MG5の高性能を確認した

わずかに高速化される可能性がある。

MG5 の amplitude 計算プログラムを GPU 用に変換した

- Sub-process の GPU code の一括生成

- 生成条件の MG5 からの変換

- 自動生成用の Python スクリプトの開発

課題をクリアすることで大きな性能改善の可能性