

GPUの利用による高速事象生成

神前 純一 (KEK)

萩原 薫 (KEK)、岡村 直利 (山梨大学)

日本物理学会 第66回年次大会@新潟大学

2011年3月25日

これまでの成果

- 我々は、2008年以降、GPUを用いて物理プロセスの散乱振幅を計算し断面積を求めるプログラムを開発し、その性能テストを行ってきた。
- すでに物理現象の散乱振幅の計算、位相空間での変数の生成、MC積分計算などで GPU (Graphics Processing Unit) の高い並列度を生かし、非常に高い計算効率の向上が実現されている。
 - K. Hagiwara, J. Kanzaki, N. Okamura, D. Rainwater and T. Stelzer, Eur. Phys. J. C66 (2010) 477, e-print arXiv:0908.4403;
 - K. Hagiwara, J. Kanzaki, N. Okamura, D. Rainwater and T. Stelzer, Eur. Phys. J. C70 (2010) 513, e-print arXiv:0909.5257;
 - J. Kanzaki, Eur. Phys. J. C71 (2011) 1559, e-print arXiv:1010.2107.
- 条件により100倍を超える実行時間の短縮効果が得られる。

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.3.2 (V0.2.1221)
CUDA Driver	Ver.3.20
CUDA Runtime	Ver.3.20
gcc/gfortran	4.4.5 (Red Hat 4.4.5-2)

計算環境

- 新GPUの導入: NVIDIA GeForce GTX580

	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

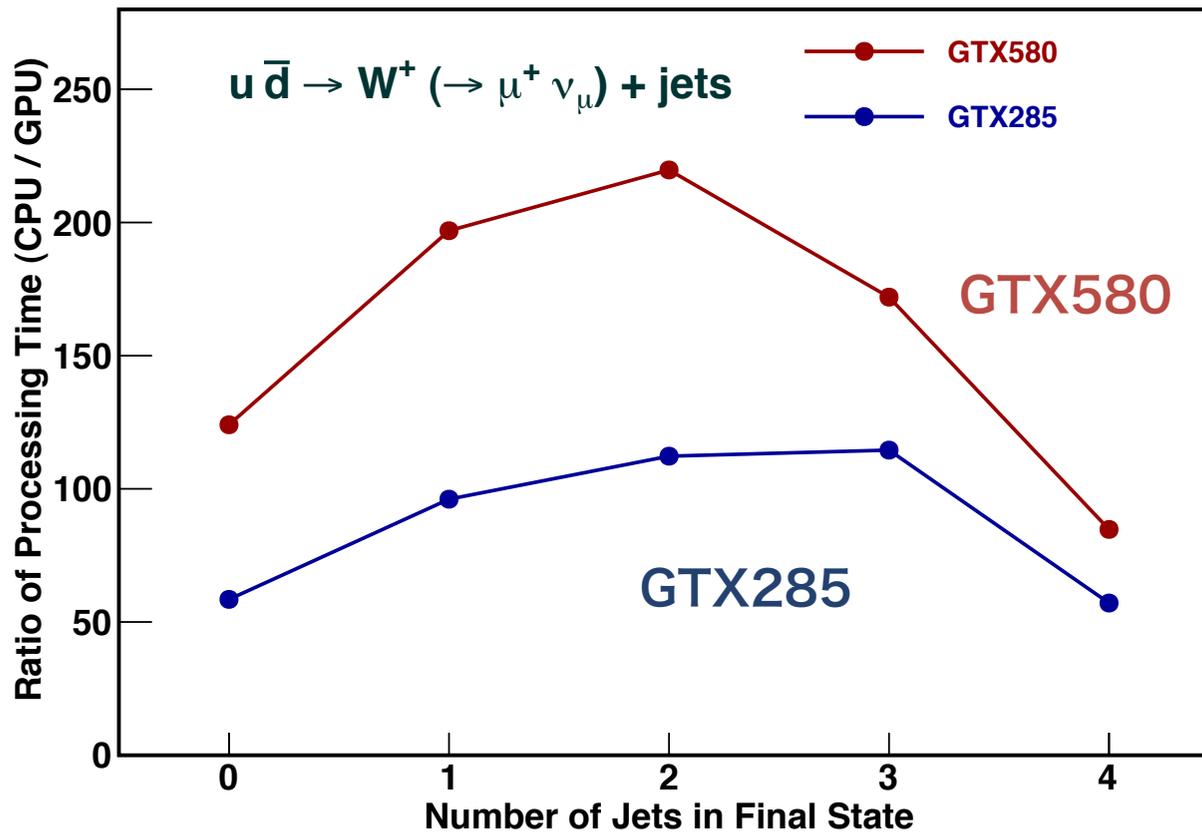
GPUの性能の比較

- SMの

$u \bar{d} \rightarrow W^+ (-\rightarrow \mu^+ \nu_\mu) + n\text{-gluons } (n=0\sim 4)$

の断面積の計算プログラムによる計算時間の比較

- 同構造の C と CUDA によるプログラムで CPU と GPU での実行時間を比較



- ほぼ processor の数の比に対応し、2倍の高速化が実現される

- GTX580ではCPUに対して200倍の加速効果がある

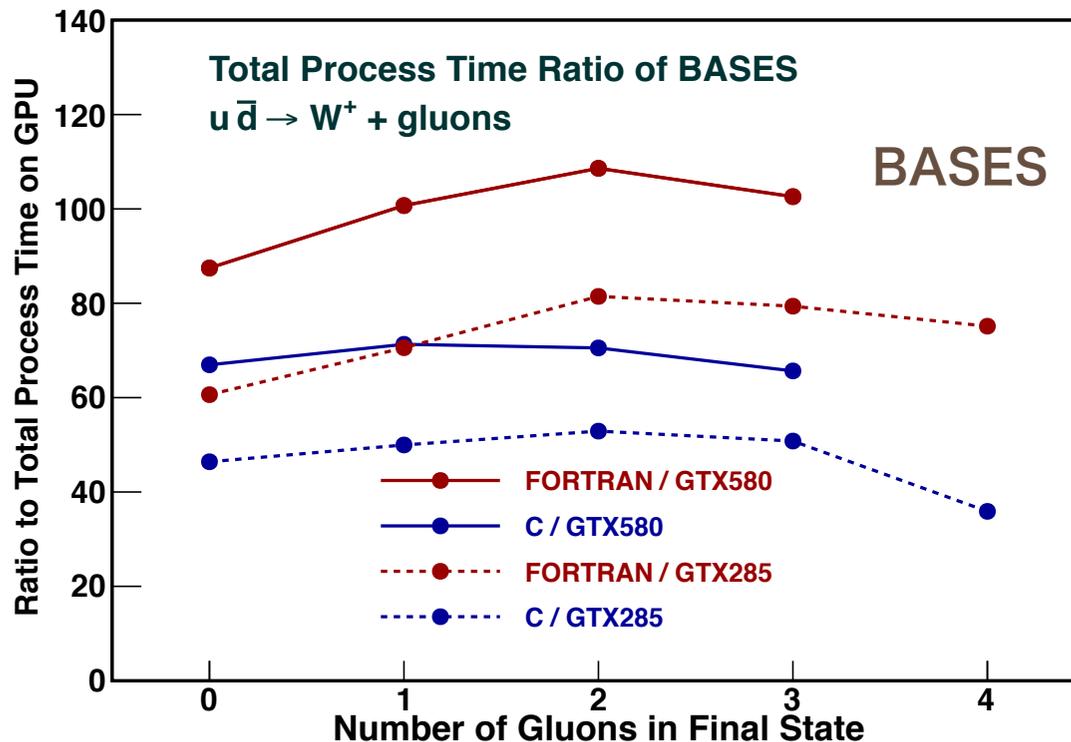
- GPUプログラムでのCPU部分の割合が少ないので、ほぼ両者の速度差が測定できる

BASES/SPRING

- KEKで開発されたMonte Carlo積分プログラムであるBASES/SPRINGの積分プログラムのBASESはすでにGPU上で稼働し、大きな性能の向上を得る事が出来た。

S. Kawabata, Comput. Phys. Commun. 41 (1986) 127; Comput. Phys. Commun. 88 (1995) 309

J. Kanzaki, Eur. Phys. J. C71 (2011) 1559, e-print arXiv:1010.2107.



- 新GPUの使用により従来のFORTRANプログラムと比較しておよそ100倍の実行速度の向上が見られる。
- CPUで実行される部分のoverheadのため加速効果が単純な断面積の計算の場合より小さくなる

SPRINGによる事象生成

- BASESプログラムの出力データを基にして事象生成を行う
SPRINGプログラムをGPUで実行可能にする。
- SPRINGでは積分空間を分割した hyper cube それぞれで積分値
に対応して event を割り振り、その hyper cube での event 生
成が成功するまで乱数発生と test を繰り返す。
- この algorithm をそのまま hyper cube での 1 event 生成に
1 processor に割り当て、それぞれが生成に成功するまで待つ様
に並列化すると、もっとも生成効率の悪い hyper cube により全
体の実行速度が決定されてしまう。
- 並列システムに合った新しい algorithm の開発の必要。

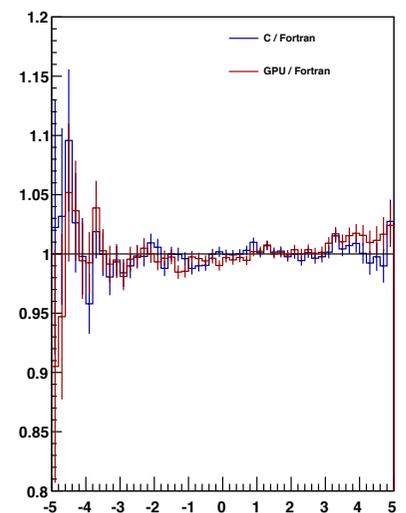
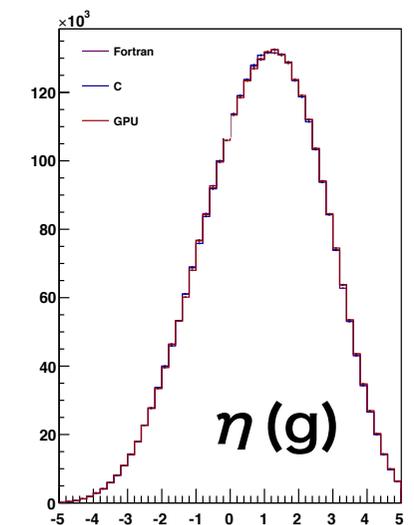
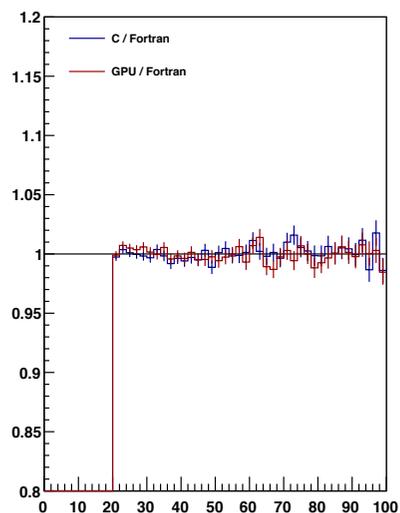
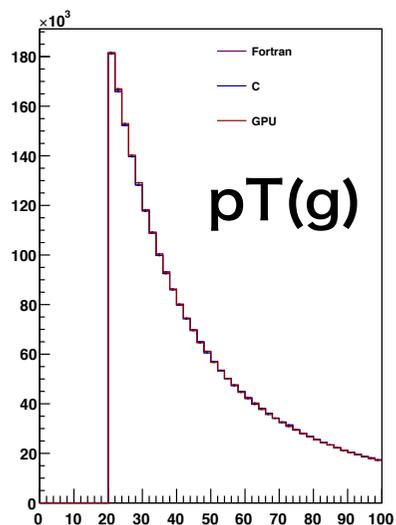
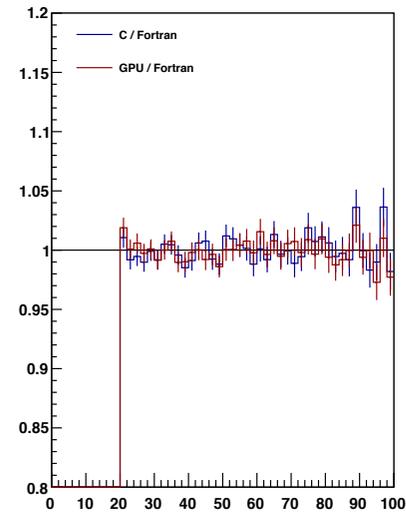
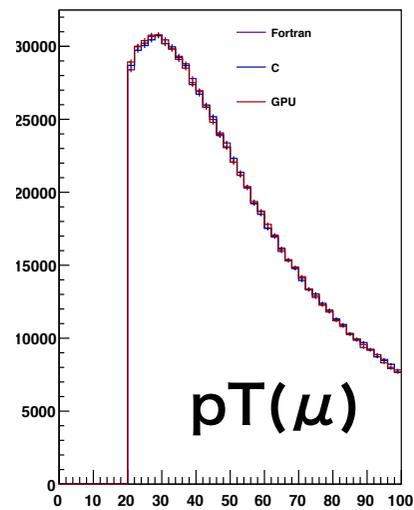
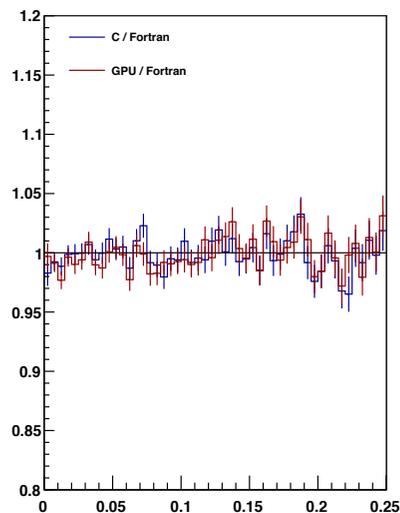
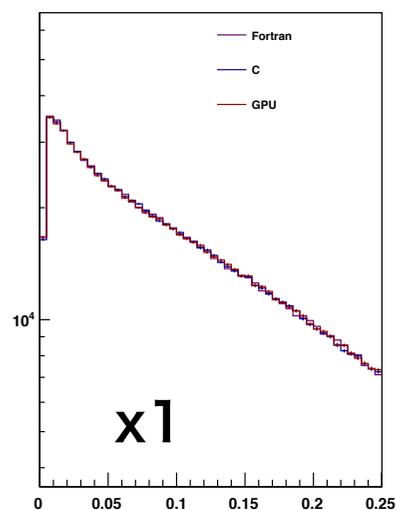
SPRINGによる事象生成

- まず通常のプログラムと同様に hyper cube での積分値に比例して生成 event を hyper cube に割り振る。GPUの各 thread が 1 event の生成を担う。
- 1回の kernel (GPUで実行されるプログラム) call で event 生成の試行を1回行う。
- その試行で生成できた event を除き、失敗した event のみで再試行を行う。その際、event 生成条件の複製を作り空いた thread に event 生成を割り当て、常に grid 内の全ての thread で event 生成が行われる様にする。
- 上の手順をすべての event 生成が成功するまで繰り返す。
- 生成効率の悪い hyper cube での event 生成の効率を、扱う processor の数を増やすことで上げることが出来る。試行が繰り返される度にその様な hyper cube に対し集中的に試行が行われる様になる。
- BASESのテストの際に用いた、LHCエネルギーでの
u d-bar \rightarrow W+ ($\rightarrow \mu + \nu \mu$) + n-gluons (n=0~4)
のプロセスについて SPRING による event 生成のテストを行った。

分布の比較

• $u \bar{d} \rightarrow W^+ (\rightarrow \mu + \nu_\mu) + 3\text{-gluons}$

FORTRAN、CそしてGPUでの結果を比較 (10^6 events を生成)



SPRINGの実行時間

- SPRING

10⁶ events を生成

BASESで全断面積が0.1%の精度まで積分した積分データを使って event 生成を行う。

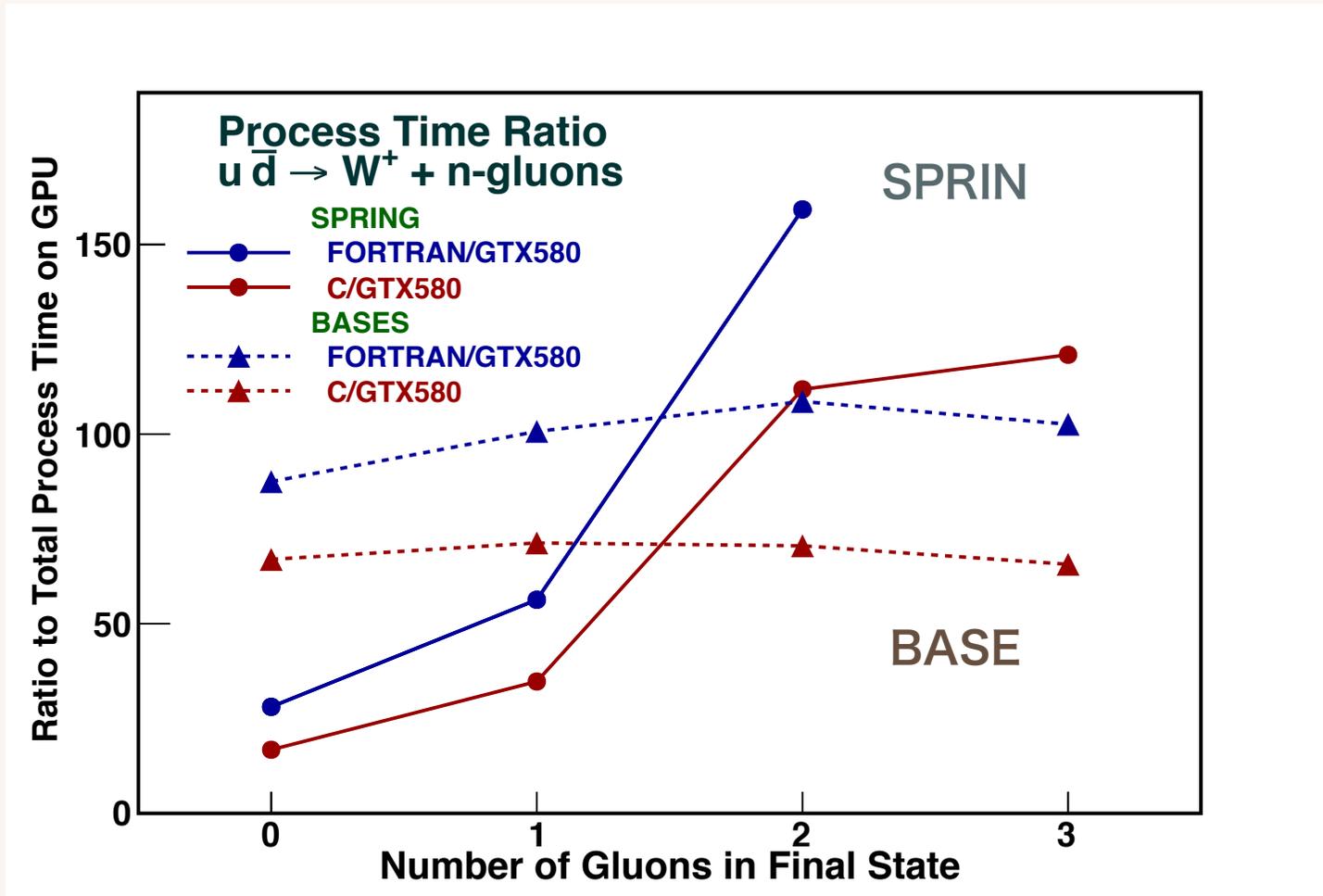
- 全実行時間 (sec)

No. of gluons	FORTRAN	C	GTX580	GTX285
0	9.72	5.80	0.346	0.411
1	43.2	26.7	0.768	0.994
2	4224.8	2966.7	26.53	42.58
3	***	32292	267.0	297.9

*** 測定中

実行時間の比

- 関数が小さい場合 ($n_{\text{gluons}}=0\sim 1$) には加速効果が小さい
- 関数が大きくなると ($n_{\text{gluons}}\geq 2$) には100倍程度の加速効果がある



まとめ

- 散乱振幅の計算への GPU の利用から始まり、BASESの稼働によりGPUが高エネルギー分野でのいろいろの計算に大きな効果の期待できることが分かってきた。
- 最新のGPUの利用によってさらなる加速効果が得られた。
- BASESに加え event 生成プログラムの SPRINGに対するGPUの応用を行った。
- 標準模型の $u \text{ dbar} \rightarrow W^+ (\rightarrow \mu^+ \nu_\mu) + \text{gluons}$ の物理過程の event の生成を行い、GPU を利用することにより、特に計算時間の必要な複雑な関数の場合 CPU で実行されるプログラムに対して100倍以上の実行速度の改善が見られた。

課題と展望

- 新しいアーキテクチャの GPU での倍精度計算の性能テスト
- BASESのヒストグラム・パッケージの GPU 化
- 素粒子・高エネルギー分野で利用されるより多くのソフトウェアへのGPUの応用（大きな効果が期待されるが開発体制の整備が必要）。
- 高速化により、単に量的な違いだけではなく、物理解析の内容、またその方法に質的な違いをもたらされる可能性
- 利用を促進し成果を上げることが、ハードウェアおよびソフトウェアの進歩を促すことが期待できる。
- そのためには情報と経験の共有が重要。