Event generation on GPU

Junichi Kanzaki (KEK)

MadGraph Spring 2011 @ Fermilab

May. 06, 2011

Contents

- Introduction
- New GPU
- BASES / SPRING (Event Generation)
- Summary & Prospects

Introduction

Overview

- Progress since last October:
  - Installation of new GPU
  - Event generation with SPRING
- New GPU: GTX580 (NVIDIA) was installed. Compare its performance with previous results.
- SPRING: event generation package based on BASES results.
- Publications:
  - QCD: Eur. Phys. J. C70 (2010) 513
  - MC integration: Eur. Phys. J. C71 (2011) 1559
  - SM: finalizing the draft
  - Event generation: in progress

Computing Environment

Host PC

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

Compilers

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)

New GPU

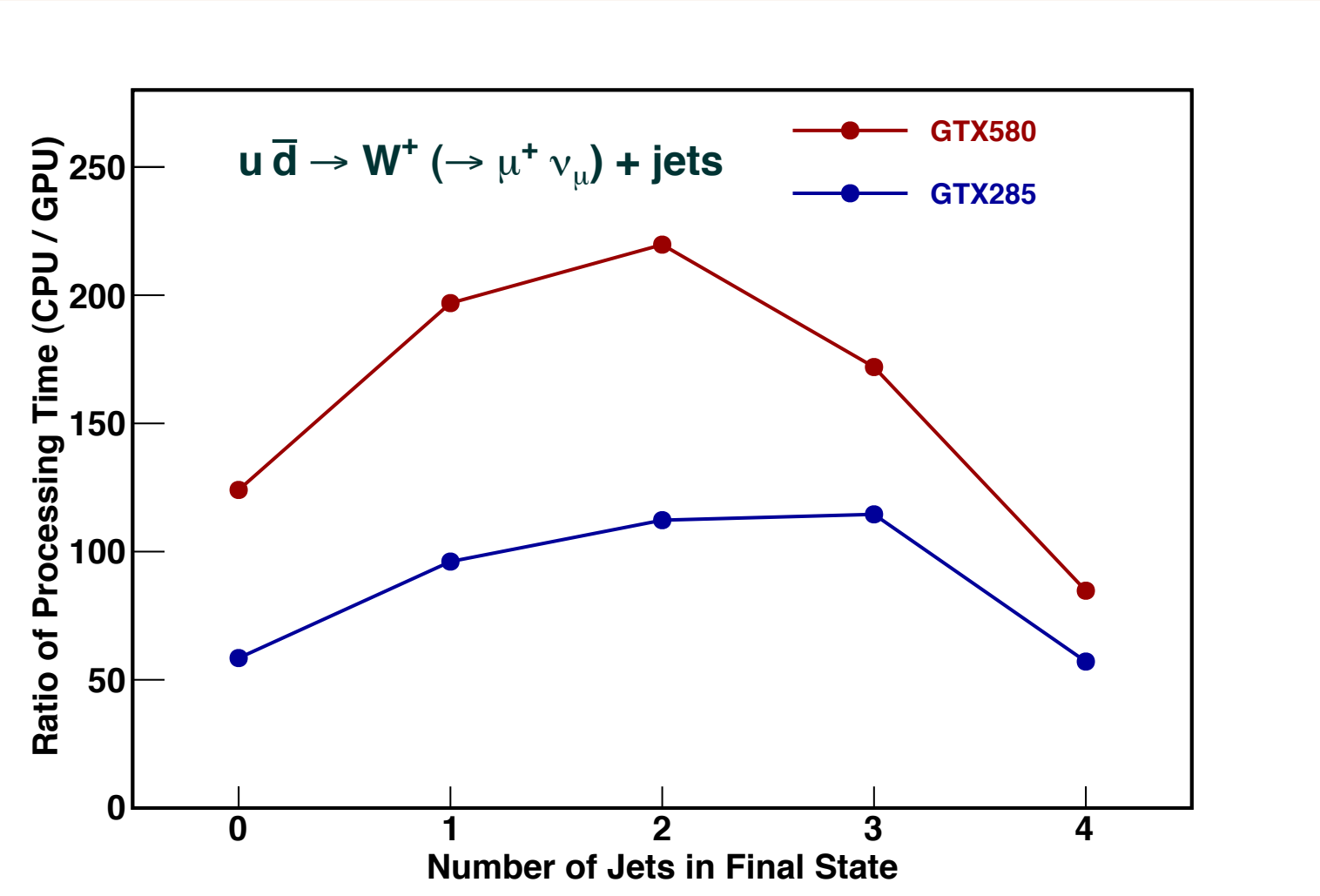
Our GPUs

- We installed a new GPU board, NVIDIA GeForce GTX580 with new architecture, to our system.

	GTX580	GTX285	GTX280	9800GTX
Multi Processor	16	30	←	16
Streaming Processors	512	240	←	128
Global Memory	1.5GB	2GB	1GB	500MB
Constant Memory	64KB	64KB	←	64KB
Shared Memory/block	48KB	16KB	←	16KB
Registers/block	32768	16384	←	8192
Warp Size	32	32	←	32
Clock Rate	1.54GHz	1.30GHz	←	1.67GHz

New GPU (Amplitude)

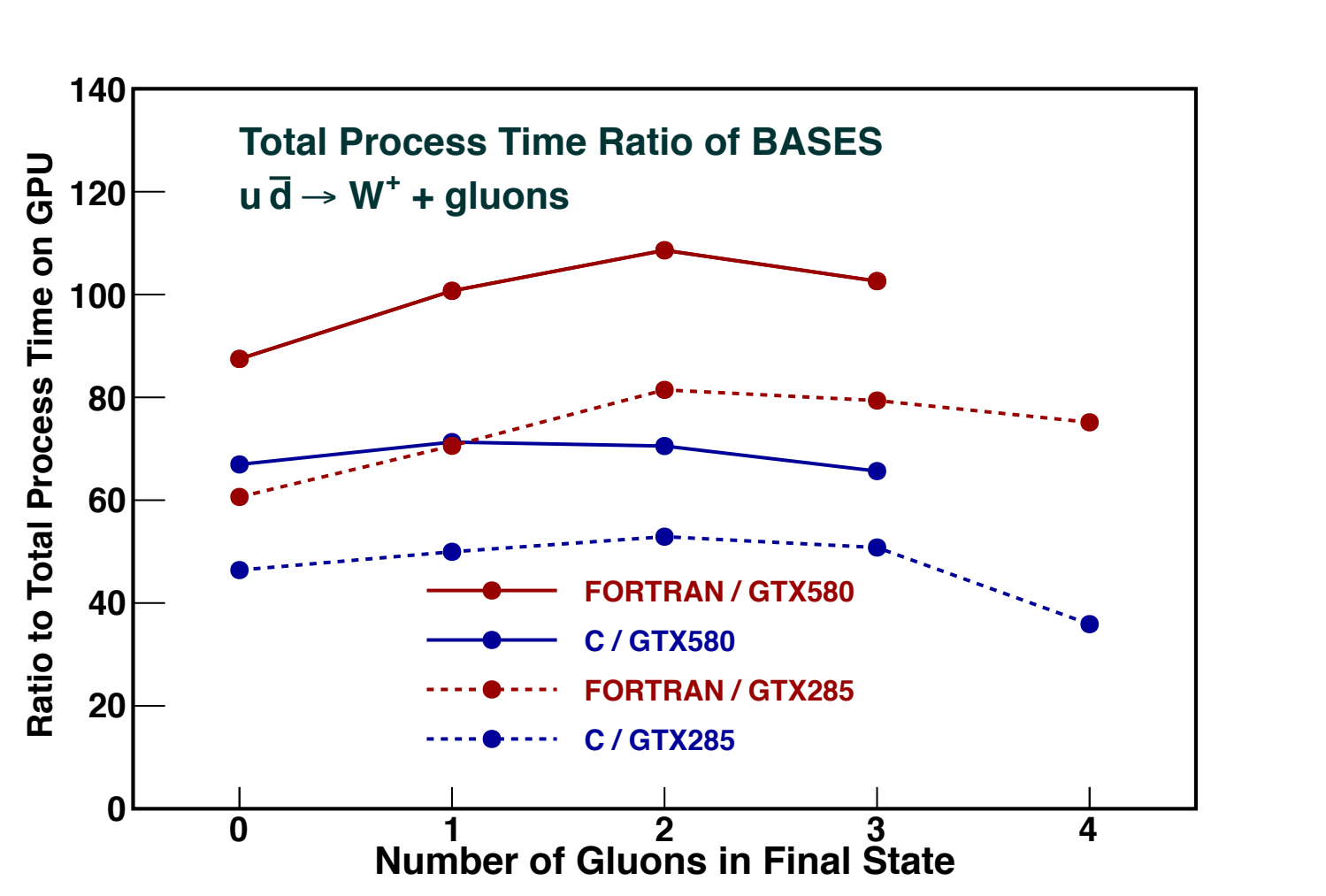
- Number of streaming processors is doubled. Hence the performance of programs on GPU is also roughly doubled.



Presented by J. Kanzaki: MadGraph Spring 2011 at Fermilab in May. 06, 2011

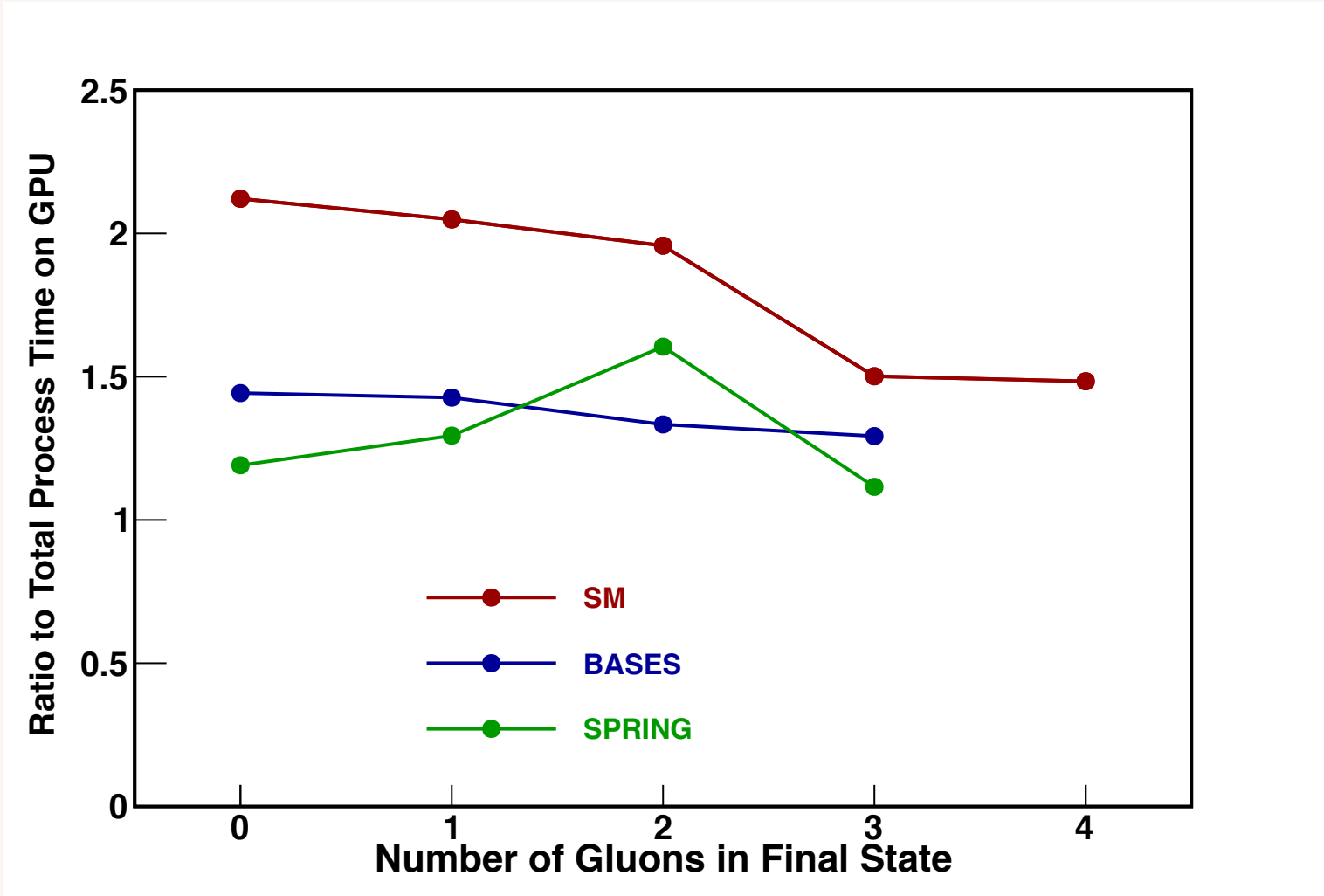
New GPU (BASES)

- BASES with high statistics. Performance improvement is smaller due to the CPU part of the program.



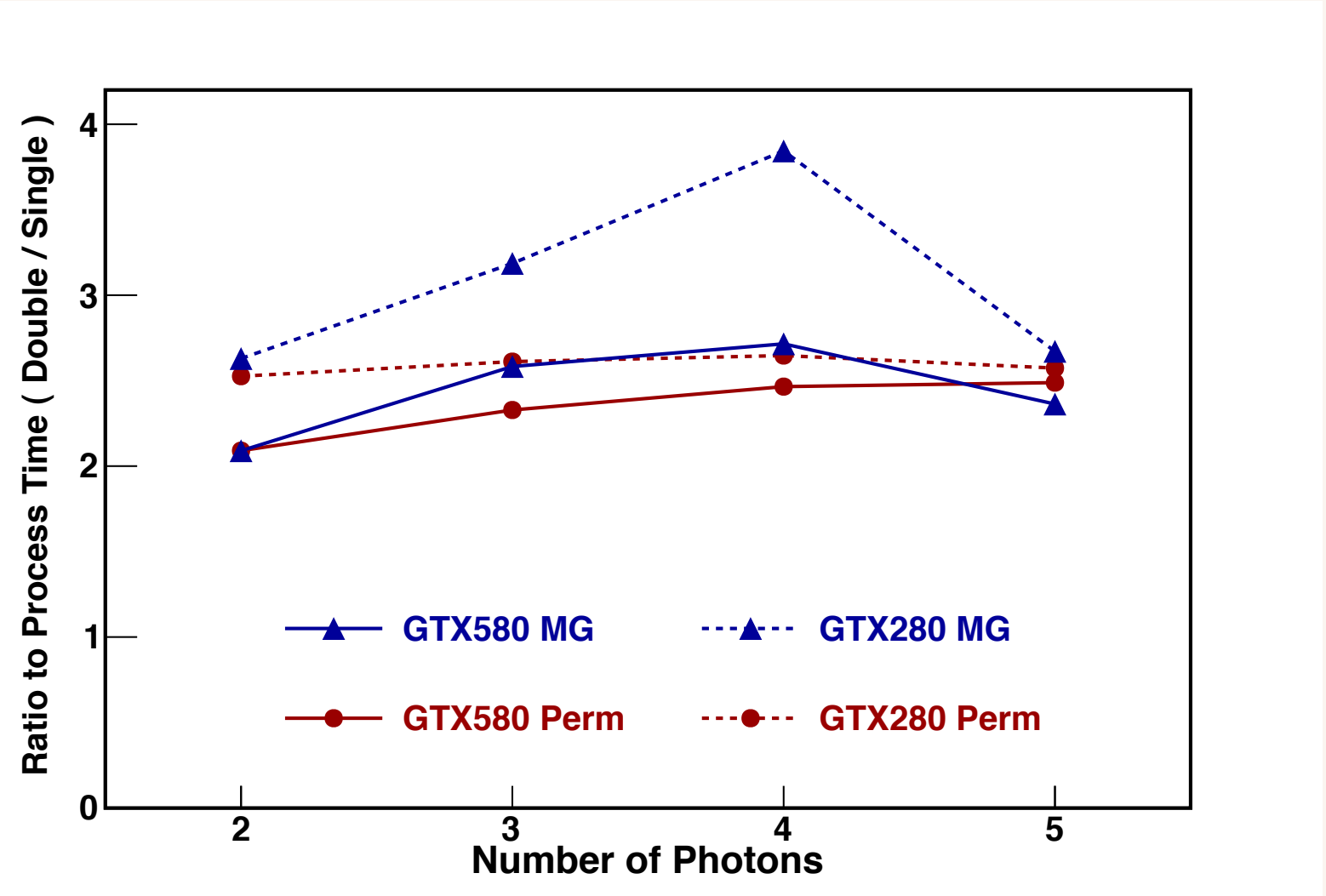
New GPU (Performance ratio)

- Improvement by new GPU itself ≈ 2.



New GPU (Double/Single)

- Double precision support?
  Not so good as I expected. -> TESLA: GPGPU specialized board.



Presented by J. Kanzaki: MadGraph Spring 2011 at Fermilab in May. 06, 2011

BASES / SPRING
(Event Generation)

BASES on GPU

- BASES: a program package for MC integration developed at KEK.
- We developed GPU version of BASES, "gBASES" (single precision). Sample processes:

udx -> W^+ (->mu^+ vm) + n-gluons (n=0~4).

Compute cross sections and measure process time.

- Parameters for the test of BASES:

No. of gluons	NCALL	ITMX	ITMX1	ITMX2
0	10^6	10	5	5
1	10^6	10	5	5
2	10^6	10	5	5
3	10^7	10	5	5
4	10^7	10	5	5

Results of BASES test

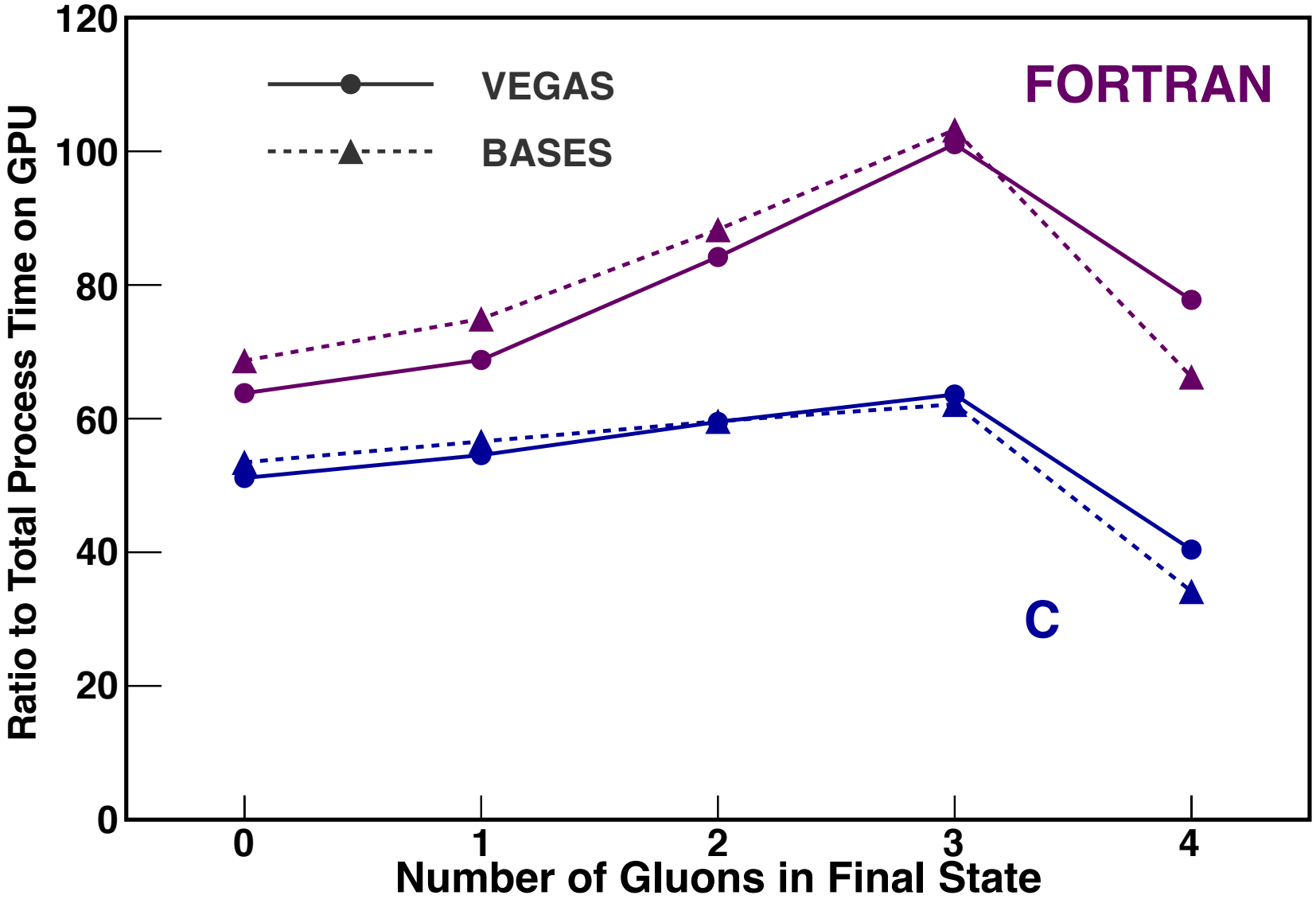
- Total cross section [fb]:

Ngluon	FORTRAN	C	GPU	[fb]
0	2.137±0.001	2.137±0.001	2.137±0.001	$\times 10^6$
1	1.785±0.001	1.784±0.001	1.782±0.001	$\times 10^5$
2	1.876±0.007	1.883±0.010	1.870±0.007	$\times 10^4$
3	2.860±0.010	2.855±0.014	2.907±0.012	$\times 10^3$
4	6.078±0.134	6.191±0.068	6.385±0.235	$\times 10^2$

- Total process time / (Ncall * # of Iterations) [micro sec]

No. of gluons	FORTRAN	C	GPU
0	1.78 (68.7)	1.39 (53.5)	0.0280
1	2.97 (75.0)	2.24 (56.6)	0.0396
2	4.97 (88.3)	3.35 (59.6)	0.0563
3	11.7 (103)	7.02 (62.2)	0.113
4	61.6 (66.2)	31.8 (34.2)	0.931

Ratio of process time



Event Generation by SPRING

- SPRING: accompanying software package of BASES
- Based on an output file generated by BASES, SPRING generate events (a set of variables) which follow the distribution of the integrand function.
- Given a desired number of events, SPRING allocates events to hyper cubes in the variable space in proportion to the value of integral in each hyper cube.
- For each event, generate a set of random variables within an assigned hyper cube, and compute the value of the integrand function and try accept/reject with another random number and the maximum function value within the hyper cube.

Event Generation by SPRING

- In CPU version of SPRING, requested number of events are generated sequentially. Generation efficiency of events depends on the property of the integrand function within the hyper cube.
- If this procedure is transferred to GPU as-is and each event generation is assigned to one processor, the total performance of SPRING must be determined by the most inefficient event generation at a certain hyper cube.
- In SPRING, there is a parameter, MAXTRY, which limits the number of trials for event generation. MAXTRY should be large to keep the rate of mis-generation zero or small enough.
- Need to develop a new algorithm suited to the parallel system.

SPRING on GPU (gSPRING)

- First allocates events to hyper cubes in the variable space in the same way as the CPU program, and one processor (one thread) takes care of one event generation.
- If the number of threads in a grid is larger than the number of events, duplicate hyper cube numbers until all threads of one kernel call (one execution of GPU program) are filled with events. If the number of threads is smaller, the kernel program is called multiple times.
- At one kernel call one trial of generation is executed. After the execution successfully generated events are removed and remaining unsuccessful events are duplicated to fill all threads. There should be no empty threads for a kernel call.

Event Generation by SPRING

- The kernel call is repeated until all events are successfully generated (or mis-generation rate becomes small enough).
- After some kernel calls, inefficient generation of events come to occupy all threads and generation efficiency for these events should be improved.

- For the test of SPRING use the same process as the BASES test:

udx -> W^+ (->mu^+ vm) + n-gluons (n=0~4).

Compare FORTRAN, C and GPU versions of SPRING program.

Test of SPRING on GPU

- In order to reproduce the shape of kinematic variables in a good accuracy, "NCALL" of BASES should be increased.
- BASES parameters for the test of SPRING:

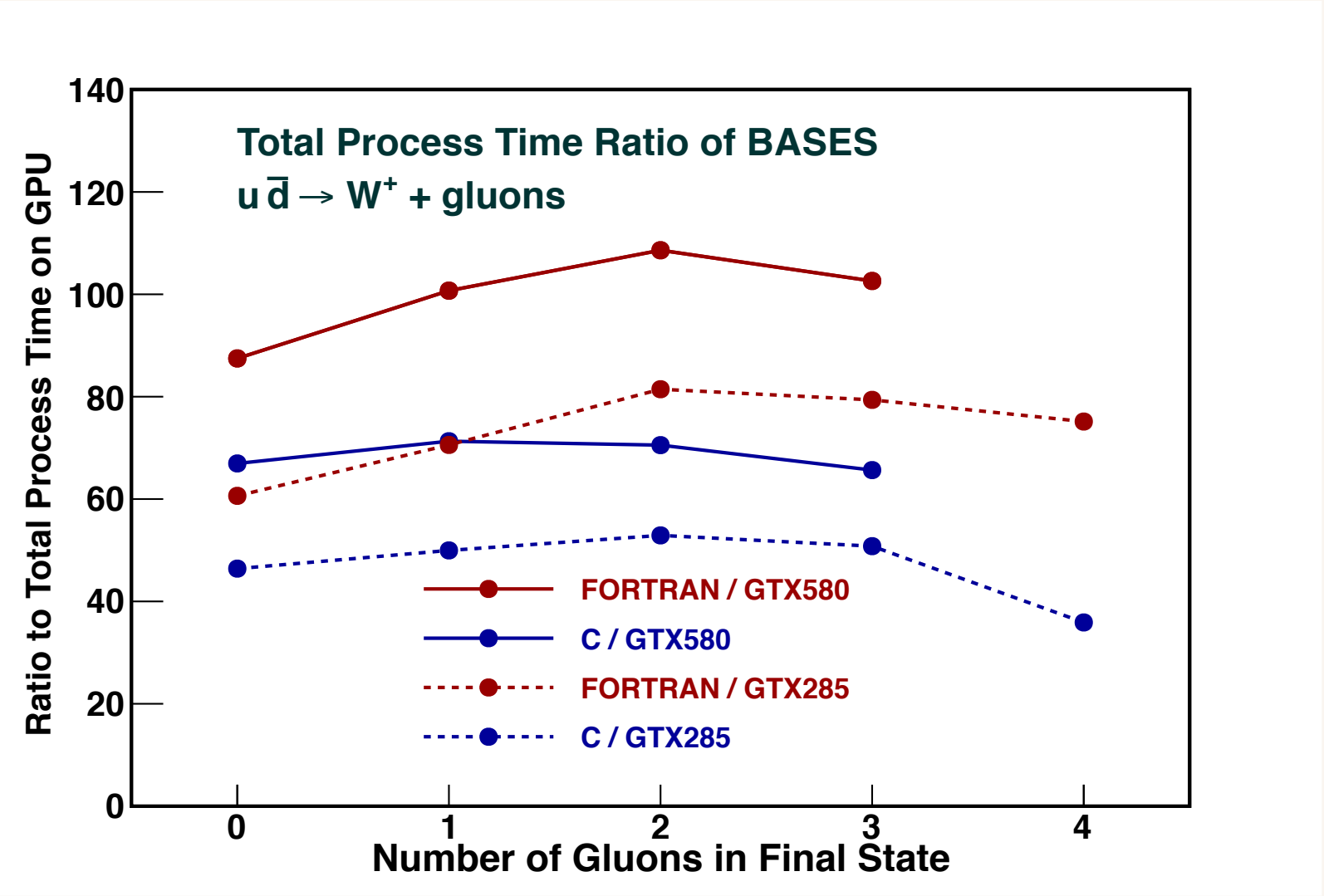
No. of gluons	NCALL	ITMX	ITMX1	ITMX2
0	10^6	10	5	5
1	10^7	10	5	5
2	10^8	10	5	5
3	10^9	10	5	5
4	10^9	10	5	5

With these parameters total cross sections are computed with 0.1% accuracy.

- Generate 10^6 events and compared the performance. Set MAXTRY and the number of kernel calls large enough so that there is no mis-generation for 10^6 events.

BASES with high statistics

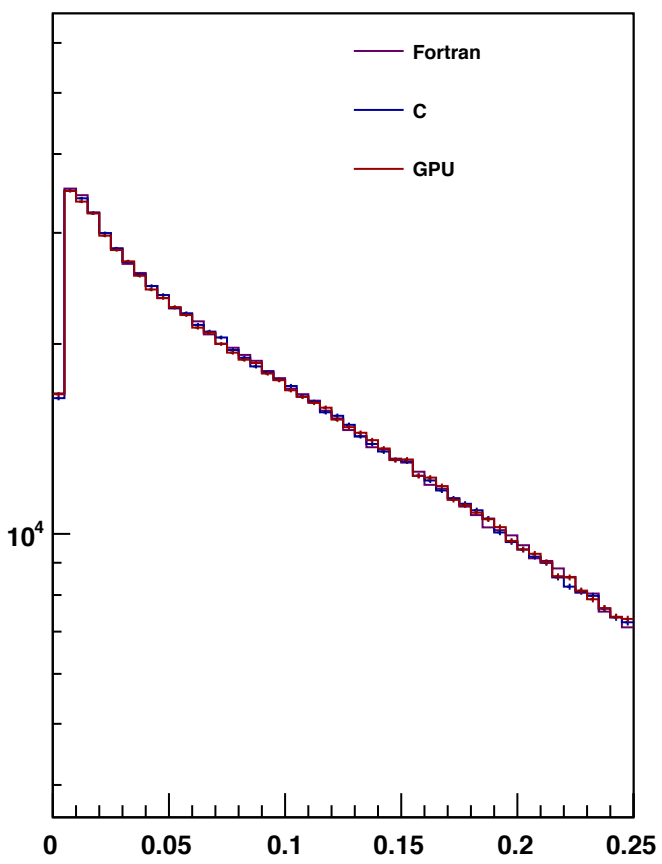
- Some results on GTX580 were not finished yet ...

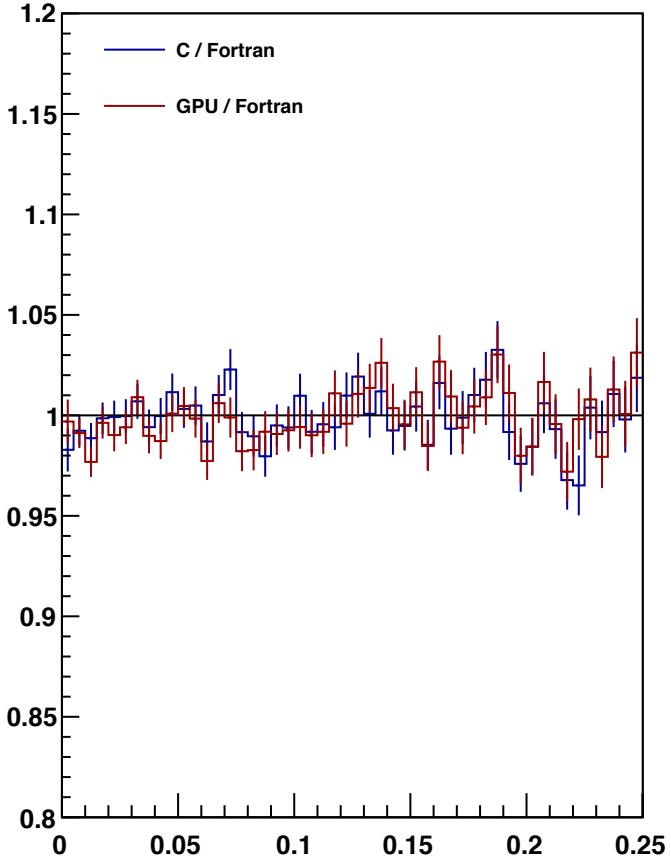


Presented by J. Kanzaki: MadGraph Spring 2011 at Fermilab in May. 06, 2011

Generated distributions

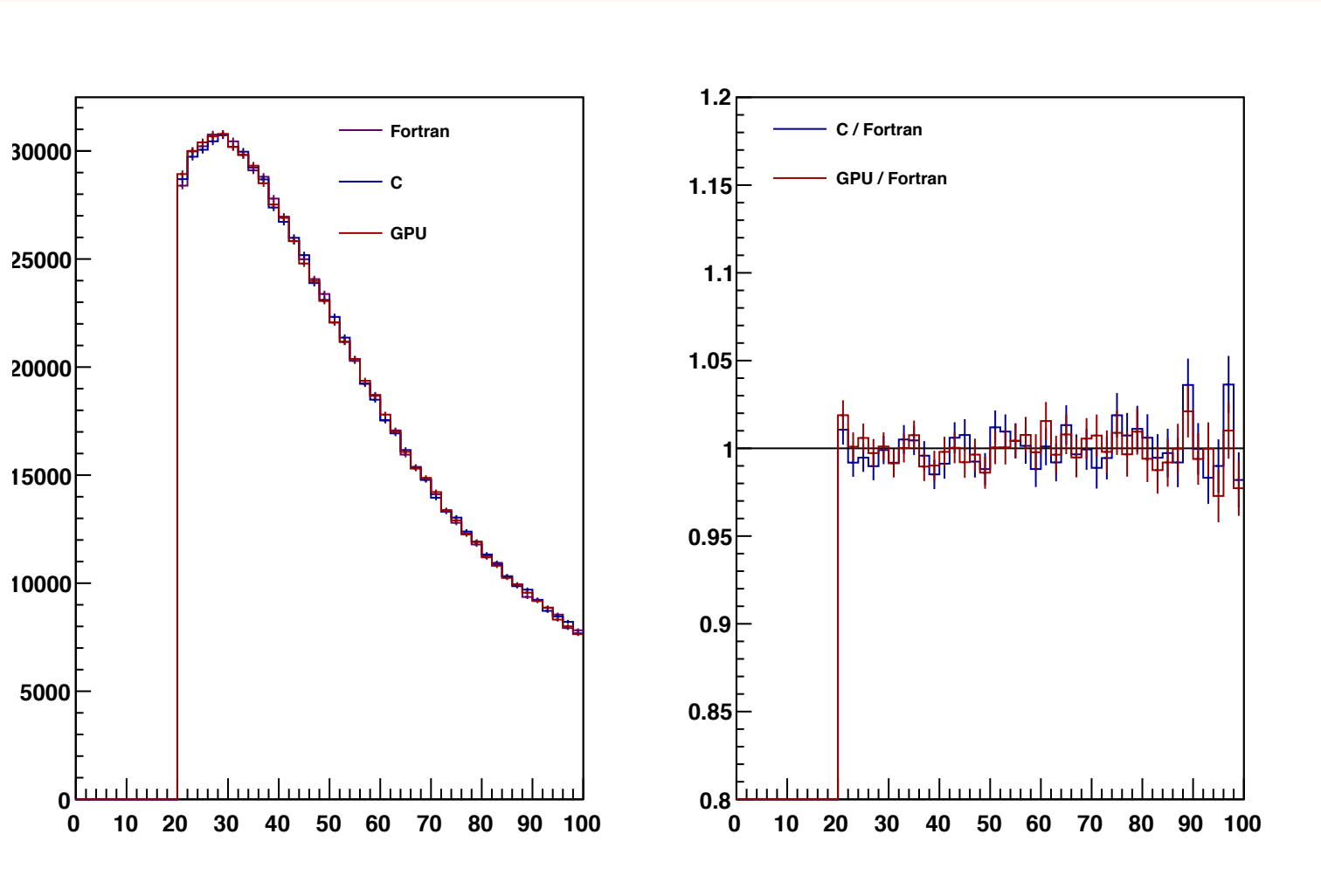
- udx -> W^+ (->mu^+ vm) + 3-gluons (10^6 events).
- x1 (energy fraction of u):





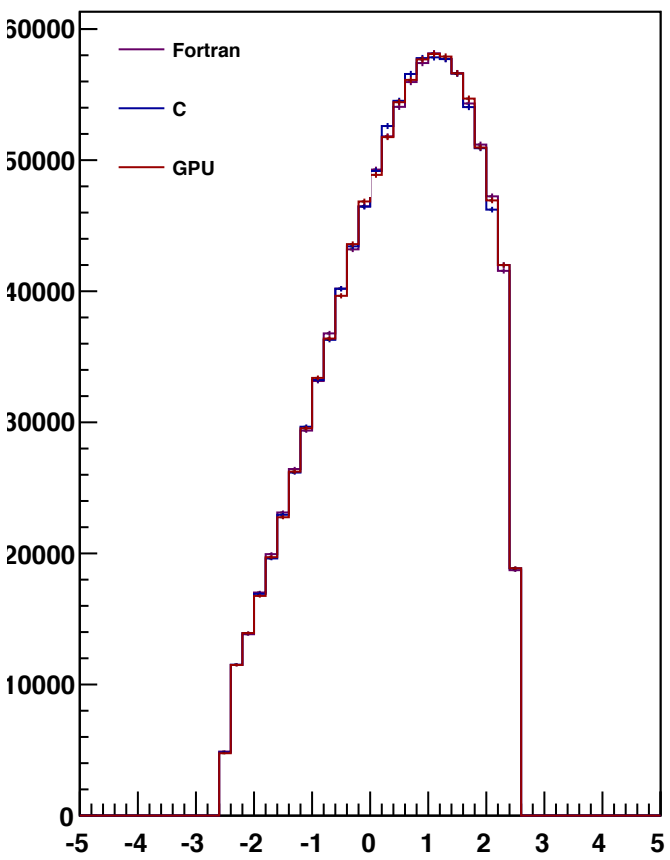
Generated distributions

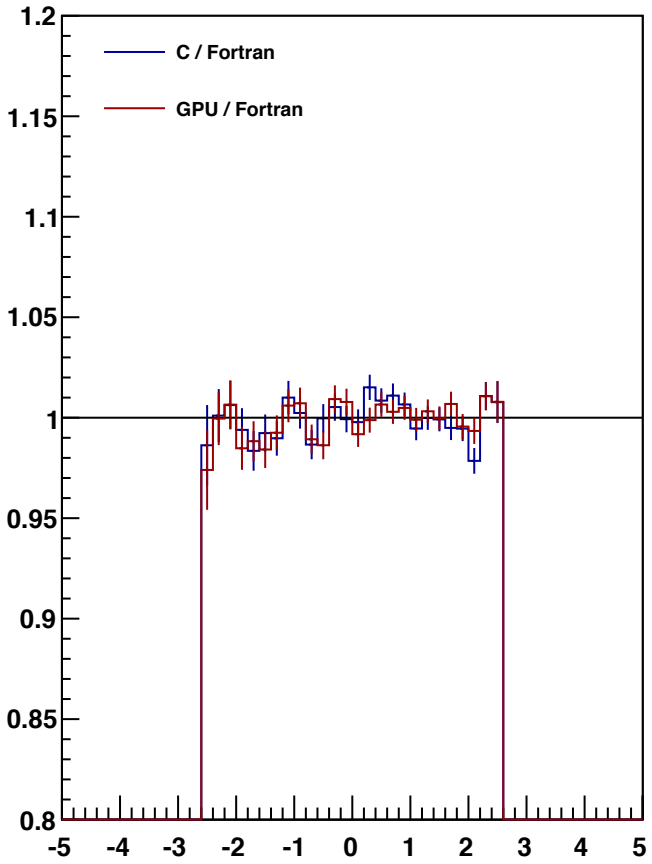
- p_T (mu^+):



Generated distributions

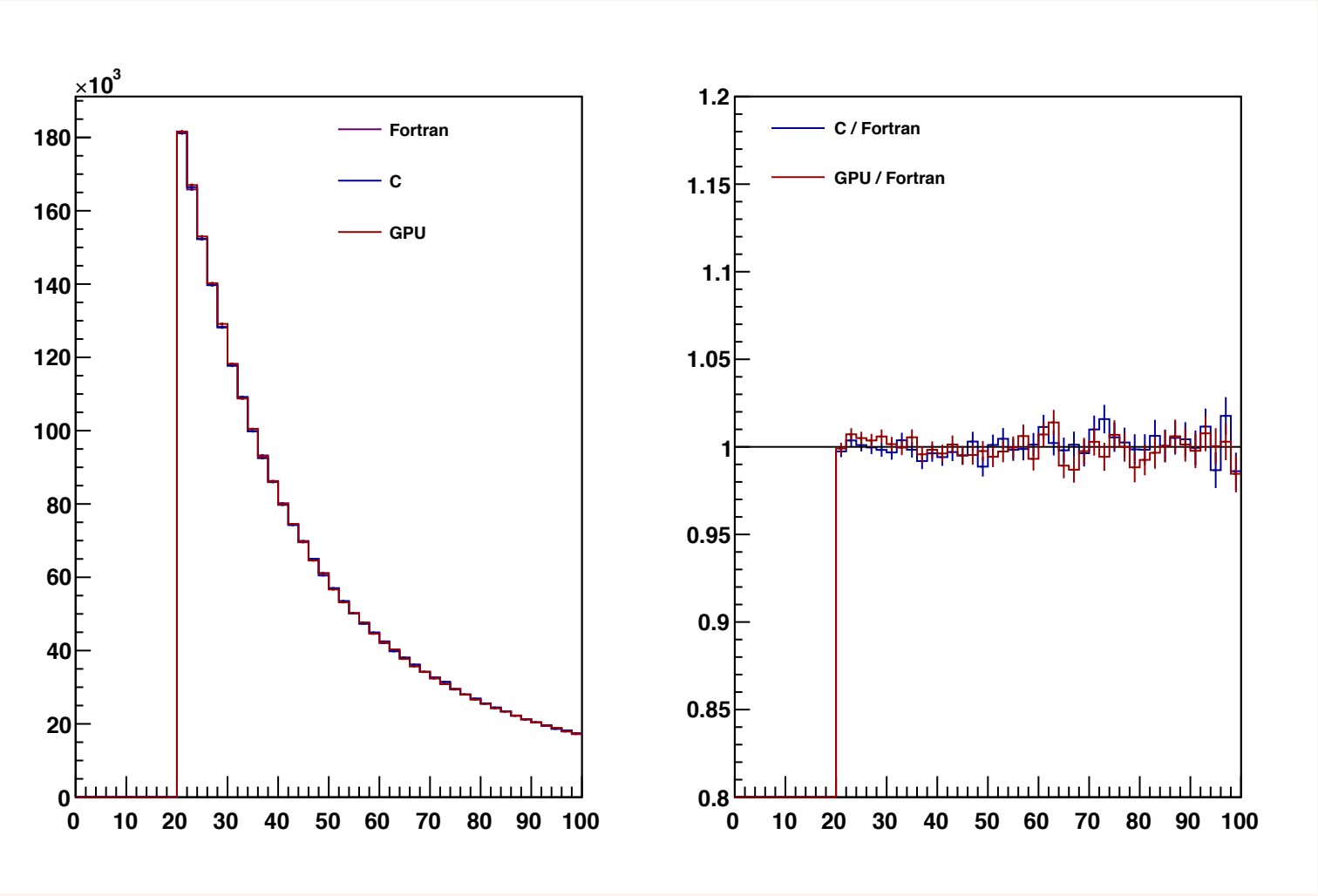
- eta (mu^+):





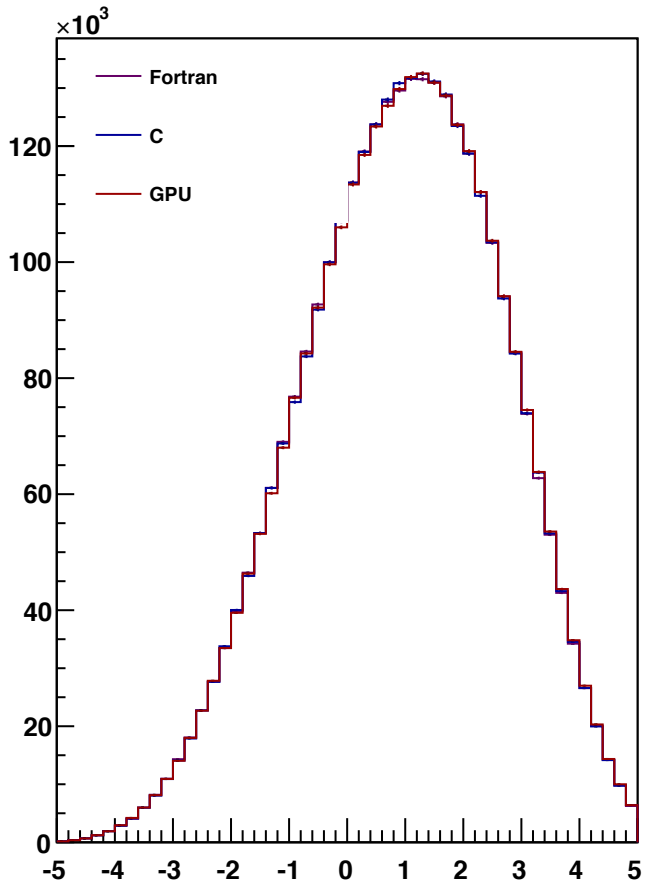
Generated distributions

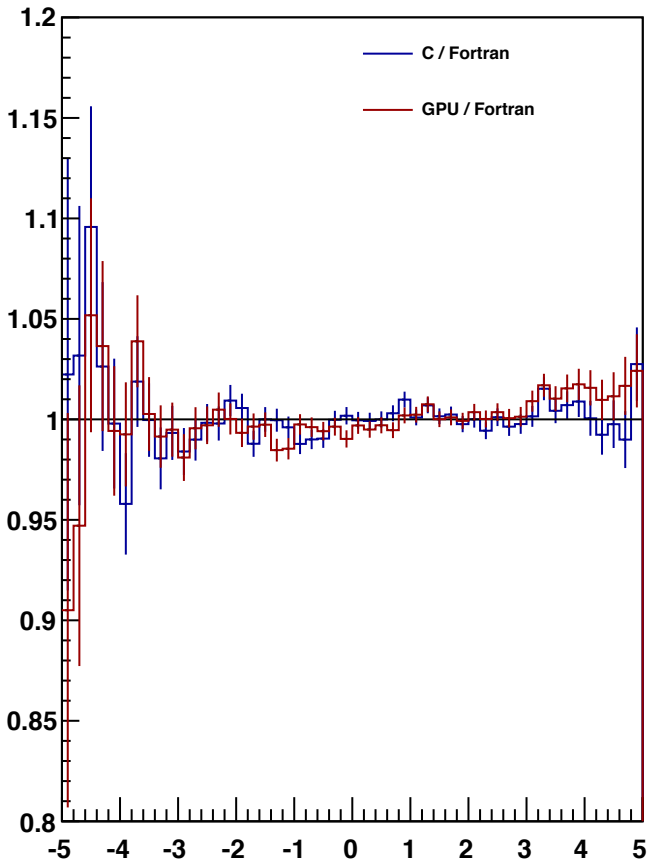
- p_T (gluon):



Generated distributions

- eta (gluon):





SPRING performance

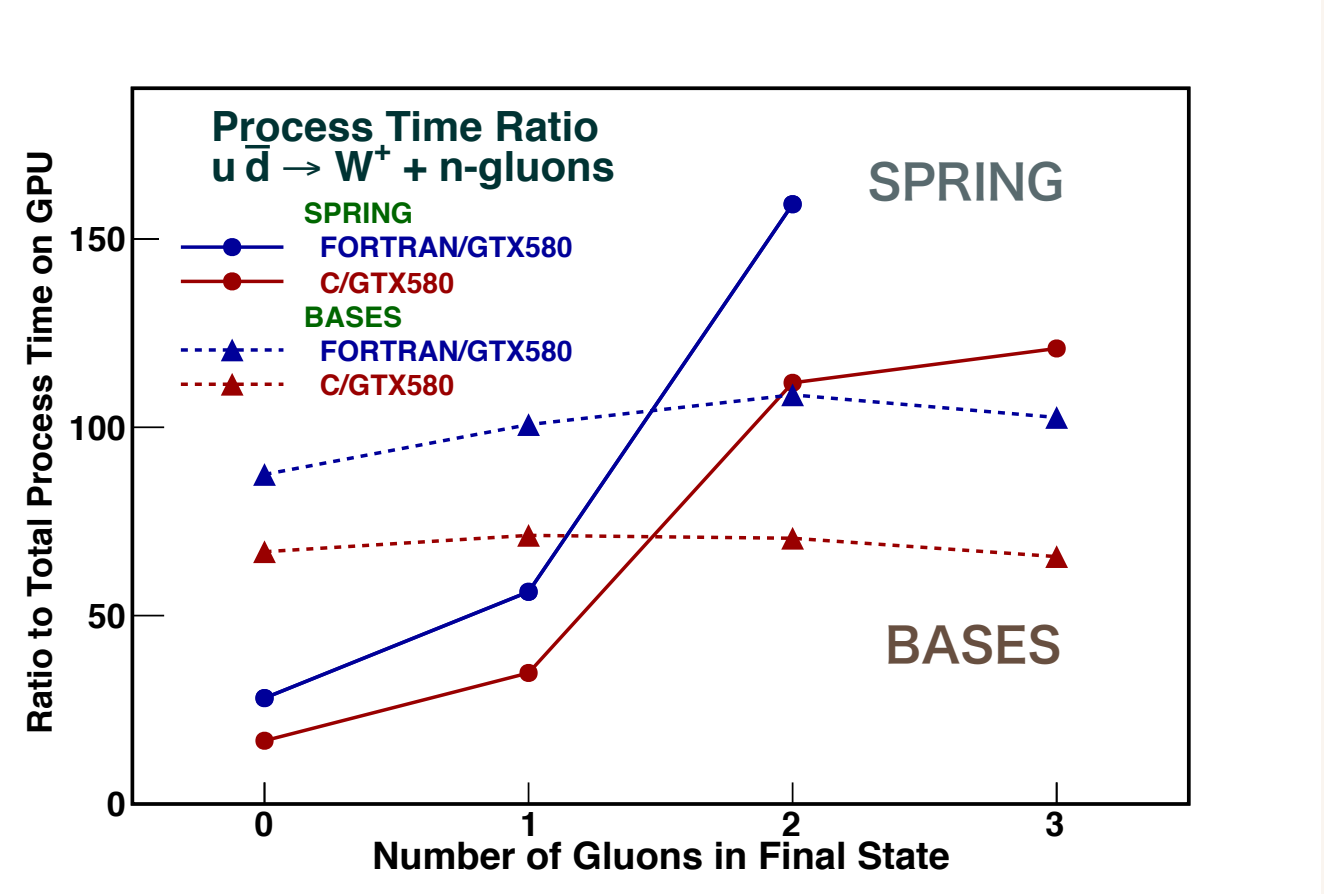
- Total execution time [sec]:

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

* Computations are not finished yet ...

Ratio of process time

- Results on GTX580:



- For ngluons=0~1, improvement is about 10-50, and for lager ngluons, it becomes larger, >100.

Summary & Prospects

Summary

- New GPU (NVIDIA GTX580) was installed. The performance of our GPU programs was roughly doubled. The total performance of BASES/SPRING on GPU was improved ≤50%.
- Double precision support was not enouogh as I expected. -> Probably more GPGPU specialized machine like TESLA is necessary.
- GPU version of the SPRING program for the event generation based on the BASES results was developed. Algorithm for event generation was modified for the parallel system.
- Improvement of performance becomes about 100 for the generation of events with complex final states which require a large amount of CPU resources.

Prospects

- Basic components for the amplitude computation and the event generation on GPU were prepared.
- More practical interface to the MG/ME system
- Use of multi-GPU system.

- Fast detector simulation on GPU.
- Parallel event loop in ROOT analysis on GPU.

✓ Presentation at ICCS2011 (Jun.01-03 at Singapore)