Fast Computation of MadGraph amplitudes on GPU

N.Okamura (Univ.Yamanashi)

Sept.09, 2011, MC-workshop@ YITP

Collaborate with K.Hagiwara, J.Kanzaki, Q.Li,
D.Rainwater, T.Stelzer,

Based on EPJ C66, 477-492 (2010)
EPJ C70, 513-524 (2010)
and now in progress

Motivation

- Many physicists, especially in this room, have a strong desire for MC computation,

"More Speed!!" "More Time!!"

- Because of more interaction types involved, more complicated event topologies, our time is consumed by the calculation time.
- It is important to accelerate the computation speed for the LHC data analysis.
- Materialize the acceleration.

Contents

- Motivation
- GPU?
  - What’s the GPU? / How to use? / Why so fast?
- Computation
  - Environment / Program flow / HEGET
- Results
  - Conditions / Processes / New GPU performance
- Conclusion
- Appendix (no time, no see)

GPU?

What's the GPU?

How to use?

Why so fast?

What's the GPU?

- **G**raphics **P**rocessing **U**nit.
    - Recently, most of the computers have the GPU.
    - Windows7 claims "GPU" for operation.
    - Games, MMORPG, require the "GPU".


GTX580

- Some GPU can be applied to the numerical calculations. We use "$500" GPU.
    - GPU connects HOST PC via PCI express x16 bus.

How to use?

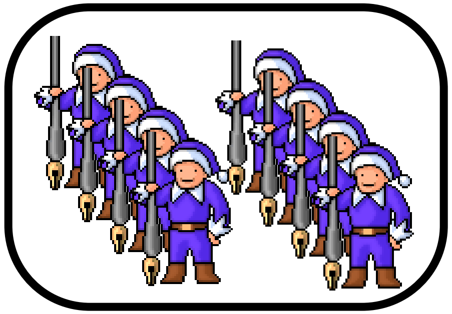
- GPGPU (**G**eneral **P**urpose computing on **GPU**)
  - two Environment / Language
    - CUDA for NVIDIA on linux/win./mac
    - OpenCL for NVIDIA, AMD, *etc*, on linux/win./mac
- Programing/Execution Model
  - Make programs for GPU written in CUDA/OpenCL.
  - Calling the GPU kernels from the CPU program.
  - CPU/GPU programs are compiled by gcc/nvcc.
  - GPU has own memories inside unit.
    - We need data transfer CPU ⇔ GPU.

GTX580

Why so fast?

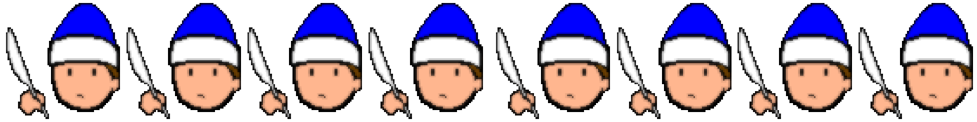
- Architecture of GPU
  - SIMD : Single Instruction Multiple Data

One core,
one instruction



All cores,
one instruction

  - Many cores in GPU execute same instruction for different data, 1 core ⇔ 1 phase space point.



  - #cores depends on the hardware, typically one unit has 100 cores. ➔ naively, x100 faster.

COMPUTATION

Machine

Program flow

HEGET

Environment 1

- Host PC
  - CPU : Core i7 920 2.67GHz (cache 8M)
  - Memory 6GB, Bus Speed 1.333GHz
  - OS : Fedora10 (64bit)
- Compilers
  - CUDA 3.2 (nvcc 3.2) and gcc 4.4.5
- GPUs
  - GTX580 ←New Architecture, “Fermi(GF100)”
  - GTX285 ←G200 series
  - 9800GTX ←G92 series

Environment 2

	GTX580(Fermi)	GTX285(G200)
Multi Processor	16SM	30SM
Total Core	512(16cores/SM)	240(8cores/SM)
Global Mem.	1.5GB	2.0GB
Const. Mem.	64KB(1page)	64KB(1page)
Shared Mem./Block	48KB	16KB
Registers/Block	32768	16384
Clock Rate	1.54GHz	1.48GHz

Multi Processor (SM) has many cores.
⇔ #cores/SM depends on the hardware generation.
There are many memory types.
“Block” is unit of the execution sequence.

Program flow

GPU

1. Initialization,
2. generate the random number,
3. generate the phase space point, $\boldsymbol{p}$ and hel.,
4. computes the parton level amplitudes,
5. summing up them multiplying the PDF of the initial state,
6. make a list of the external particles $\boldsymbol{p}$, hel., and the cross section, and GPU→CPU,
7. summing up all cross sections on CPU.

HEGET 1

- **HEGET :** HELAS Evaluation with GPU Enhanced Technology

  - HEGET is the subroutine package for computing the amplitude on GPU written in CUDA.
  - Naming scheme of HEGET functions follows that of HELAS subroutines.
  - Ready for all SM processes.

- Need “Phase Space Generator” on GPU.
- Need “control program” on CPU.

HEGET 2

- Procedure for using
  1. Generates the FORTRAN code by MadGraph.
  2. Translate generated “matrix.f” to CUDA automatically for the GPU computation.
  3. Gathering the “phase space generator”, “matrix.cu”, and so on written in CUDA.
  4. Compile by using gcc/nvcc.

     Wait a few minutes/hours, make a cup of coffee.
  5. Execute the program

     Wait a few moments, sipping a cup of coffee.
  6. Get results, have fun!!

RESULTS

Comparison

Physics Conditions

Results1, 2

Comparison

- Accuracy check
  - CPU: MG/ME4 and BASES (Fortran)
    with double precision.
  - GPU: HEGET (CUDA)
    with single precision,
    without any optimizations like BASES.

- Speed check
  - CPU: Prepare the same structure program (C).
  - GPU: Include the transfer time (CPU⇔GPU).

Physics Conditions

- 14TeV collision
- Final state heavy particles decay as follows
  - $W^+ \to l^+ \nu$, $Z \to l^+ l^-$, $t \to b l^+ \nu$, $H \to \tau^+ \tau^-$ (τ not decay)
  - "b-jets" as light jets, no isolation cut.
  - Higgs mass:120GeV, $Br(H \to \tau^+ \tau^-)$=0.0425
- Event cuts
  - Jets: $|\eta|<5.0$, $p_{Ti} > 20$GeV, $p_{Tij} > 20$GeV
    $p_{Tij} = \min(p_{Ti}, p_{Tj})\Delta R_{ij}$ (isolation cut)
  - leptons: $|\eta|<2.5$, $p_T > 20$GeV
- PDF:CTEQ6L1

Processes in paper

- $\boldsymbol{W^+/Z}$ + *n-jets* ($n \leqq 4$)
- $\boldsymbol{W^-W^+/\ W^+Z/\ ZZ}$ +*n-jets* ($n \leqq 3$)
- $\boldsymbol{t\bar{t}}$ + *n-jets* ($n \leqq 3$)
- $\boldsymbol{W^+/Z}$+Higgs +*n-jets* ($n \leqq 3$)
- $\boldsymbol{t\bar{t}}$+Higgs +*n-jets* ($n \leqq 2$)
- Higgs (*via* WBF) +*n-jets* ($n \leqq 4$)
- Multiple Higgs (*via* WBF) +*n-jets*
  ($n \leqq 3$ for 2Higgs, $n \leqq 2$ for 3Higgs)
- pure-QCD/pure-QED processes.

Results 1-1

α_s=0.13
Q=M_Z

- W^+ + *n-jets* ($n \leqq 4$)

n	subprocess	Cross section [fb]				Process time [μsec]	
		HEGET	Bases	MG-ME		GPU	CPU
0	$u\bar{d} \to W^+$	8.549 ± 0.000	8.558 ± 0.008	8.553 ± 0.007	$\times 10^6$	3.89×10^{-2}	2.28×10^0
1	$u\bar{d} \to W^+ + g$	7.147 ± 0.002	7.133 ± 0.007	7.092 ± 0.012	$\times 10^5$	4.741×10^{-2}	4.56×10^0
	$ug \to W^+ + d$	1.228 ± 0.000	1.227 ± 0.004	1.224 ± 0.002	$\times 10^6$	4.73×10^{-2}	4.54×10^0
2	$u\bar{d} \to W^+ + gg$	7.506 ± 0.006	7.498 ± 0.006	7.495 ± 0.009	$\times 10^4$	5.97×10^{-2}	6.71×10^0
	$ug \to W^+ + dg$	6.189 ± 0.001	6.194 ± 0.001	6.154 ± 0.001	$\times 10^6$	5.83×10^{-2}	6.74×10^0
	$uu \to W^+ + ud$	3.683 ± 0.001	3.683 ± 0.003	3.657 ± 0.006	$\times 10^4$	7.65×10^{-2}	7.95×10^0
	$gg \to W^+ + d\bar{u}$	4.110 ± 0.005	4.109 ± 0.004	4.104 ± 0.006	$\times 10^4$	5.95×10^{-2}	6.72×10^0
3	$u\bar{d} \to W^+ + ggg$	1.168 ± 0.008	1.150 ± 0.002	1.138 ± 0.002	$\times 10^4$	1.20×10^{-1}	1.38×10^1
	$ug \to W^+ + dgg$	2.682 ± 0.006	2.685 ± 0.002	2.646 ± 0.004	$\times 10^5$	1.15×10^{-1}	1.38×10^1
	$uu \to W^+ + udg$	3.266 ± 0.005	3.259 ± 0.007	3.206 ± 0.004	$\times 10^4$	2.47×10^{-1}	1.80×10^1
	$gg \to W^+ + d\bar{u}g$	2.438 ± 0.008	2.443 ± 0.002	2.413 ± 0.003	$\times 10^4$	1.21×10^{-1}	1.39×10^1
4	$u\bar{d} \to W^+ + gggg$	2.524 ± 0.012	2.494 ± 0.056	2.428 ± 0.004	$\times 10^3$	1.19×10^0	6.81×10^1
	$ug \to W^+ + dggg$	1.215 ± 0.010	1.203 ± 0.002	1.139 ± 0.002	$\times 10^5$	1.15×10^0	6.87×10^1
	$uu \to W^+ + udgg$	2.350 ± 0.009	2.324 ± 0.015	2.166 ± 0.004	$\times 10^4$	1.93×10^0	8.00×10^1
	$gg \to W^+ + d\bar{u}gg$	1.041 ± 0.008	1.028 ± 0.002	0.968 ± 0.002	$\times 10^4$	1.71×10^0	6.69×10^1

- Cross section: we generate 10^{10} events for high multiplicity.
- Process time : we generate 10^8 events, [μsec/event]

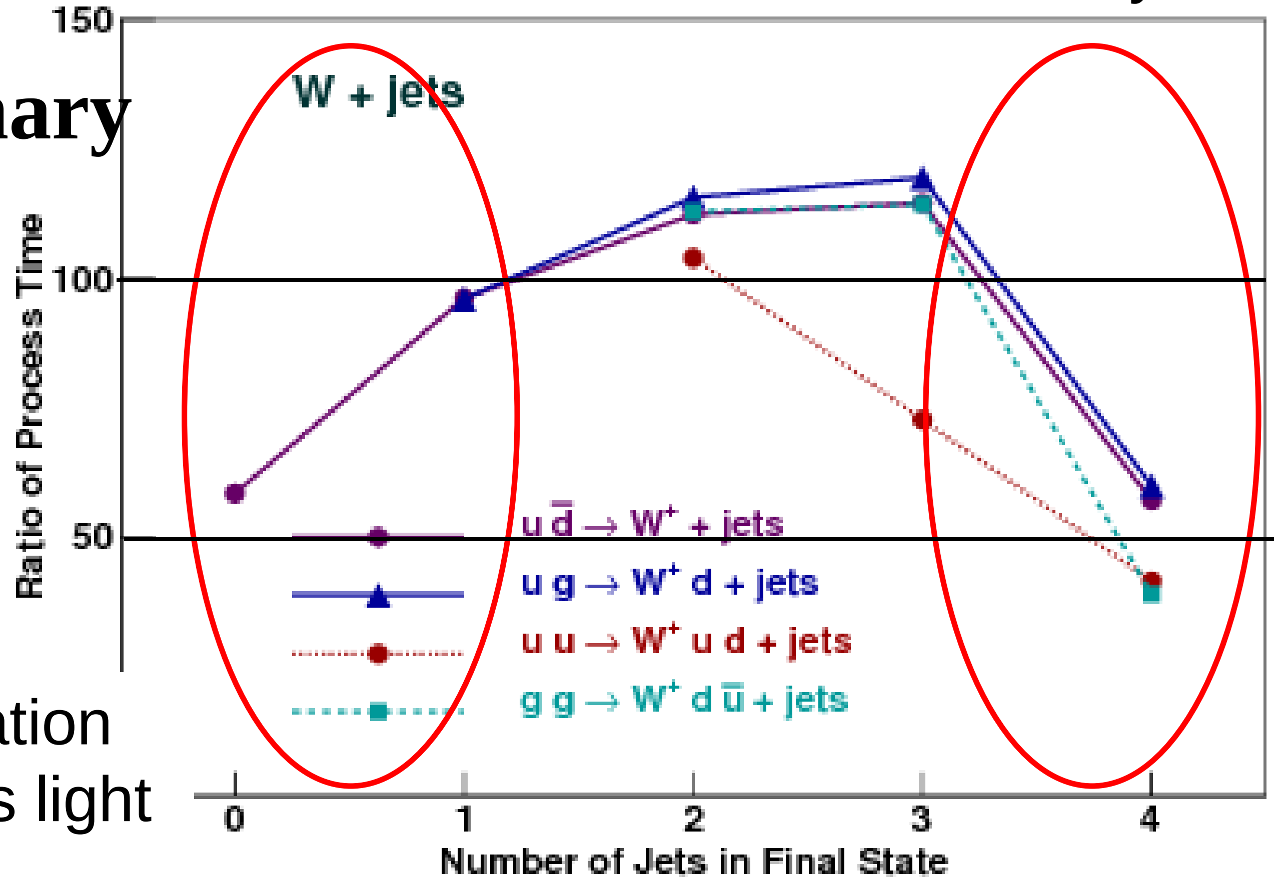
preliminary

Results 1-2

- W^+ + *n-jets* ($n \leqq 4$)

Memory cost is high

preliminary



fast

slow

Computation weight is light

Results 2-1

α_s=0.13
Q=M_Z

- ***HZ*** + *n-jets* ($n \leqq 3$)

n	subprocess	Cross section [fb]				Process time [μsec]	
		HEGET	Bases	MG-ME		GPU	CPU
0	$u\bar{u} \to HZ$	3.395 ± 0.005	3.393 ± 0.004	3.376 ± 0.007	$\times 10^{-1}$	5.80×10^{-2}	4.97×10^{0}
1	$u\bar{u} \to HZ + g$	1.378 ± 0.000	1.378 ± 0.002	1.373 ± 0.003	$\times 10^{-1}$	6.74×10^{-2}	5.88×10^{0}
	$ug \to HZ + u$	7.889 ± 0.003	7.885 ± 0.014	7.882 ± 0.012	$\times 10^{-2}$	6.70×10^{-2}	5.70×10^{0}
2	$u\bar{u} \to HZ + gg$	3.988 ± 0.004	3.922 ± 0.026	2.820 ± 0.019	$\times 10^{-2}$	9.14×10^{-2}	8.66×10^{0}
	$ug \to HZ + ug$	7.205 ± 0.038	7.097 ± 0.012	6.077 ± 0.008	$\times 10^{-2}$	8.94×10^{-2}	8.54×10^{0}
	$uu \to HZ + uu$	5.038 ± 0.056	4.427 ± 0.004	4.376 ± 0.006	$\times 10^{-3}$	1.55×10^{-1}	1.40×10^{1}
	$gg \to HZ + u\bar{u}$	2.566 ± 0.002	2.570 ± 0.002	2.562 ± 0.004	$\times 10^{-3}$	9.21×10^{-2}	8.67×10^{0}
3	$u\bar{u} \to HZ + ggg$	1.094 ± 0.004	1.075 ± 0.017	0.496 ± 0.004	$\times 10^{-2}$	2.63×10^{-1}	2.07×10^{1}
	$ug \to HZ + ugg$	4.518 ± 0.018	4.427 ± 0.008	2.199 ± 0.009	$\times 10^{-2}$	2.53×10^{-1}	2.07×10^{1}
	$uu \to HZ + uug$	7.651 ± 0.168	3.031 ± 0.008	1.592 ± 0.007	$\times 10^{-3}$	5.93×10^{-2}	4.14×10^{2}
	$gg \to HZ + u\bar{u}g$	2.330 ± 0.006	2.318 ± 0.003	1.784 ± 0.004	$\times 10^{-3}$	2.64×10^{-1}	2.08×10^{1}

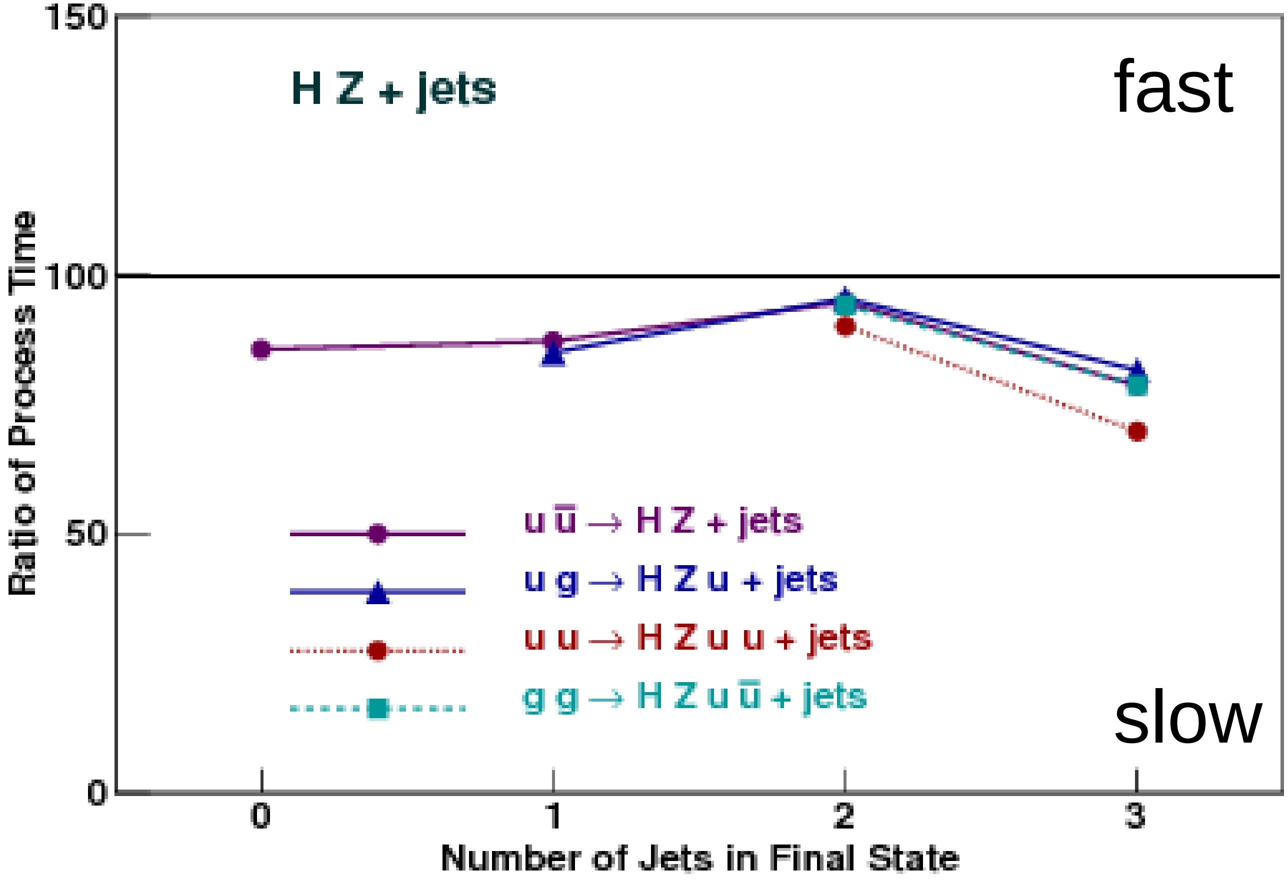
- Cross section: we generate 10^{10} events for high multiplicity.
- Process time : we generate 10^{8} events, [μsec/event]

preliminary

Results 2-2

- ***HZ*** + *n-jets* (*n*≦3)

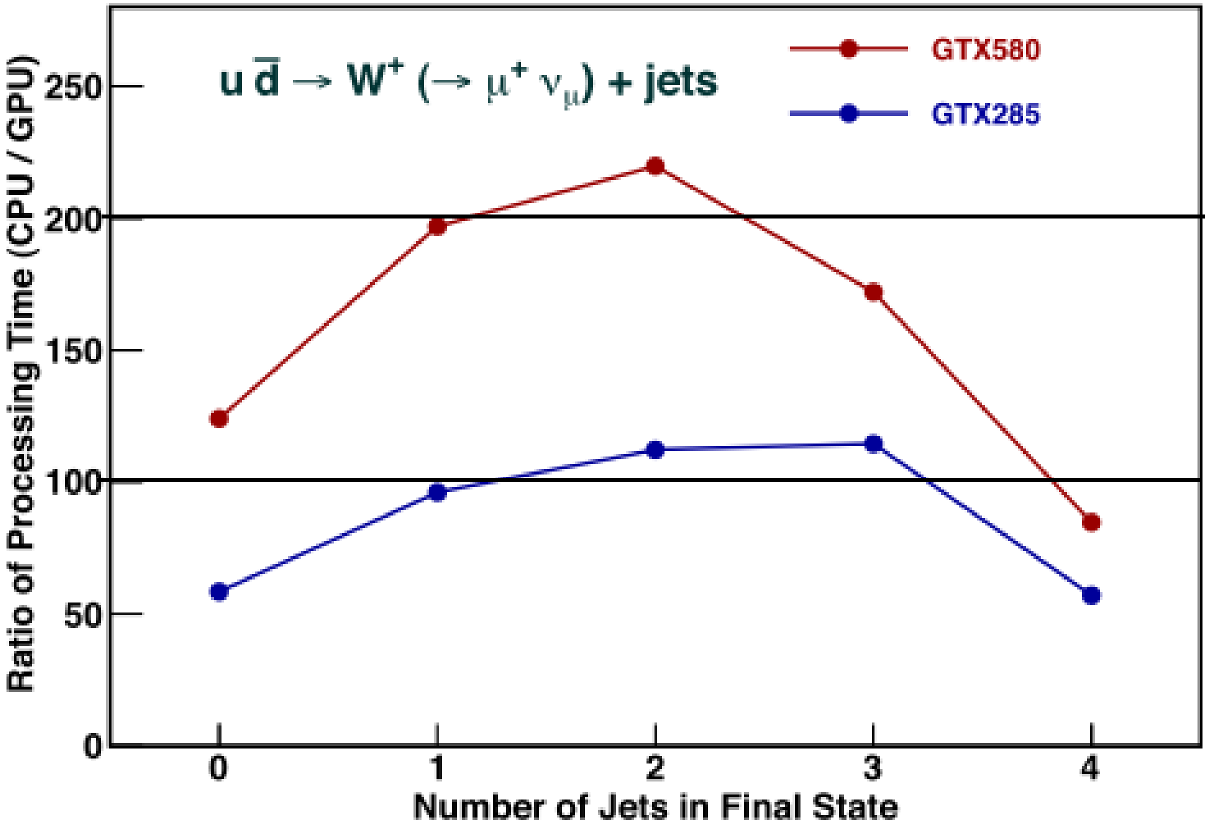
preliminary



Results of the new GPU

- New GPU, GTX580 has 512 cores.
- GTX285 has 240 cores.

preliminary



fast

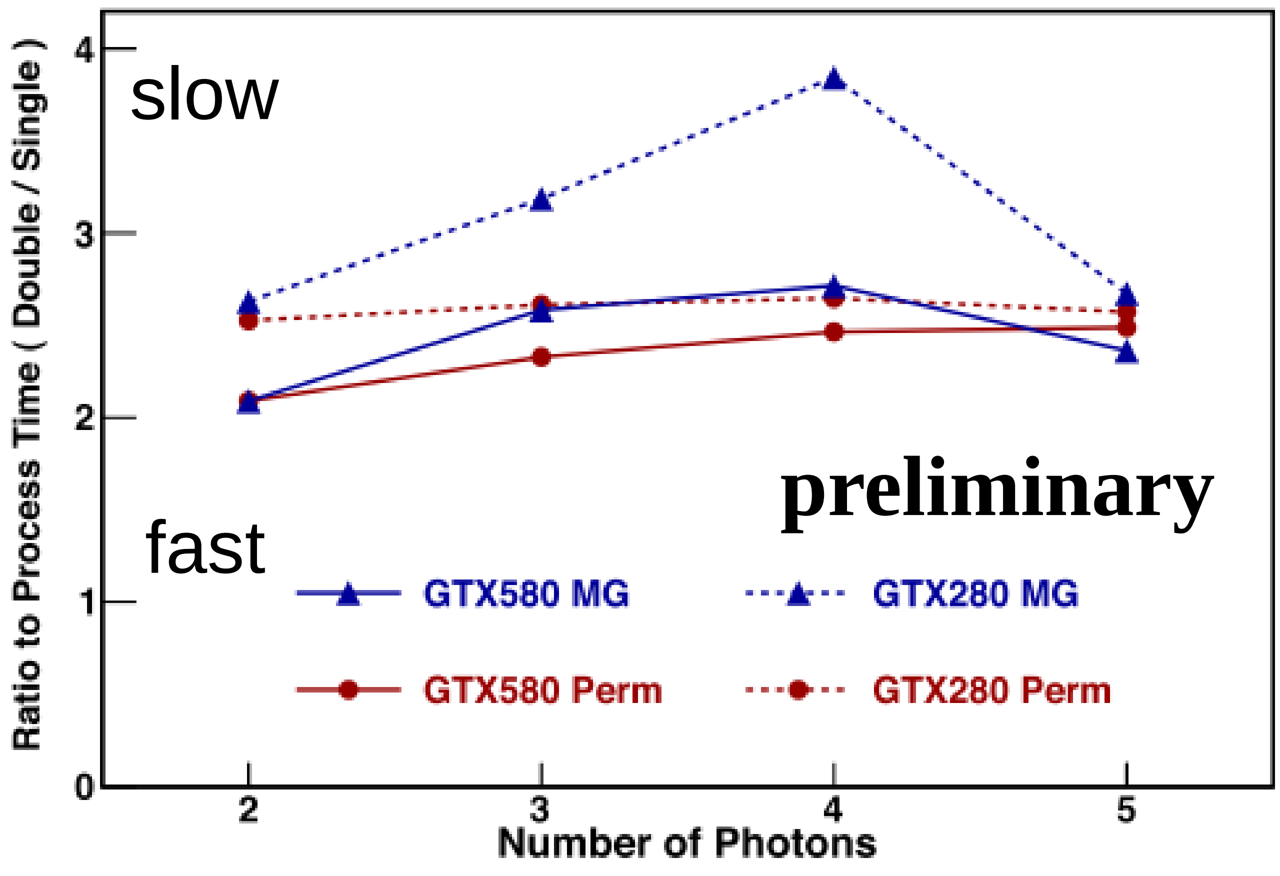
slow

Double / Single

- New GPU, GTX580 has many unit for “double precision” computation

uu→nγ

Double/Single



CONCLUSION

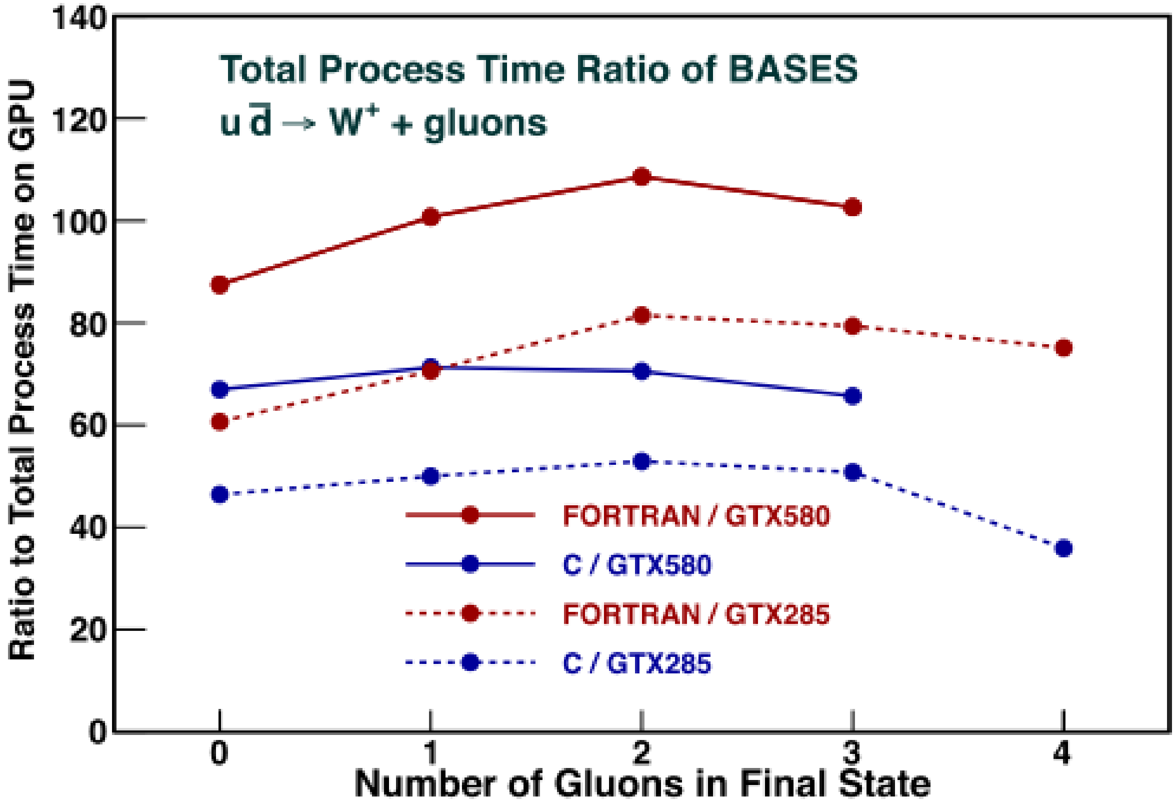
Conclusion

- GPU opens the “new world” for MC.
  - cheap (∼$500), fast (1Tflops/unit)
  - Not a super tool, but so powerful.
- HELAS(fortran) → HEGET(CUDA)
  - Now on stage, all SM processes.
- Acceleration
  - more than x50, x100 is not a dream.
  - New GPU is more powerful, x200 is not a fiction.

………Please wait OpenCL version, now under construction………

And…

- VEGAS/BASES: J.Kanzaki, EPJ C71,1559 (2011).



- SPRING : Now Ready
- PS, PGS : preparing

APPENDIX

No time

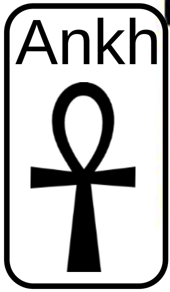
No see

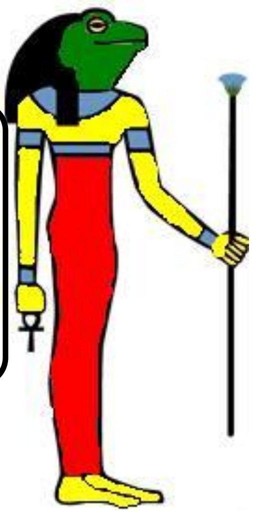
What's "HEGET"

- **H**ELAS
- **E**valuation with
- **G**PU
- **E**nhanced
- **T**echnology

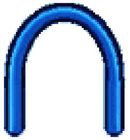
and.................

Heqet





- Heqet (Heket, Heget)
  - Goddess of Egyptian myth
  - Symbol of life, fertility.
- Hieroglyph of "frog" means "100,000"

1=  10=  100=  1,000= 

10,000=  100,000=  1,000,000= 

"Millions of frogs were born after the annual inundation of the Nile, which brought fertility to the otherwise barren lands"

(from "wikipedia")