MG5 on GPU

Junichi Kanzaki (KEK)

MadGraph5_aMC@NLO Femto

March 27, 2015, Kavli IPMU, the University of Tokyo

Motivation

- Increase of the amount of LHC data (raw & simulated)
  - Run1: 5+20fb^{-1} up to 2012
  - Run2: 100fb^{-1} / 3 years from 2015
  - And more: 300fb^{-1} until 2022, 3000fb^{-1} until 2035
  - Also requires huge amount of simulation data for physics analysis.
- GRID: uses CPU and storage resources around the world
  - already takes several weeks to reprocess accumulated real data
- Need technical innovations of HPC (High Performance Computing) in large scale data processing and also in personal analysis environments.
  → Multi-core & Many-Core CPU, PC Farms, "GPGPU",...

Bibliography

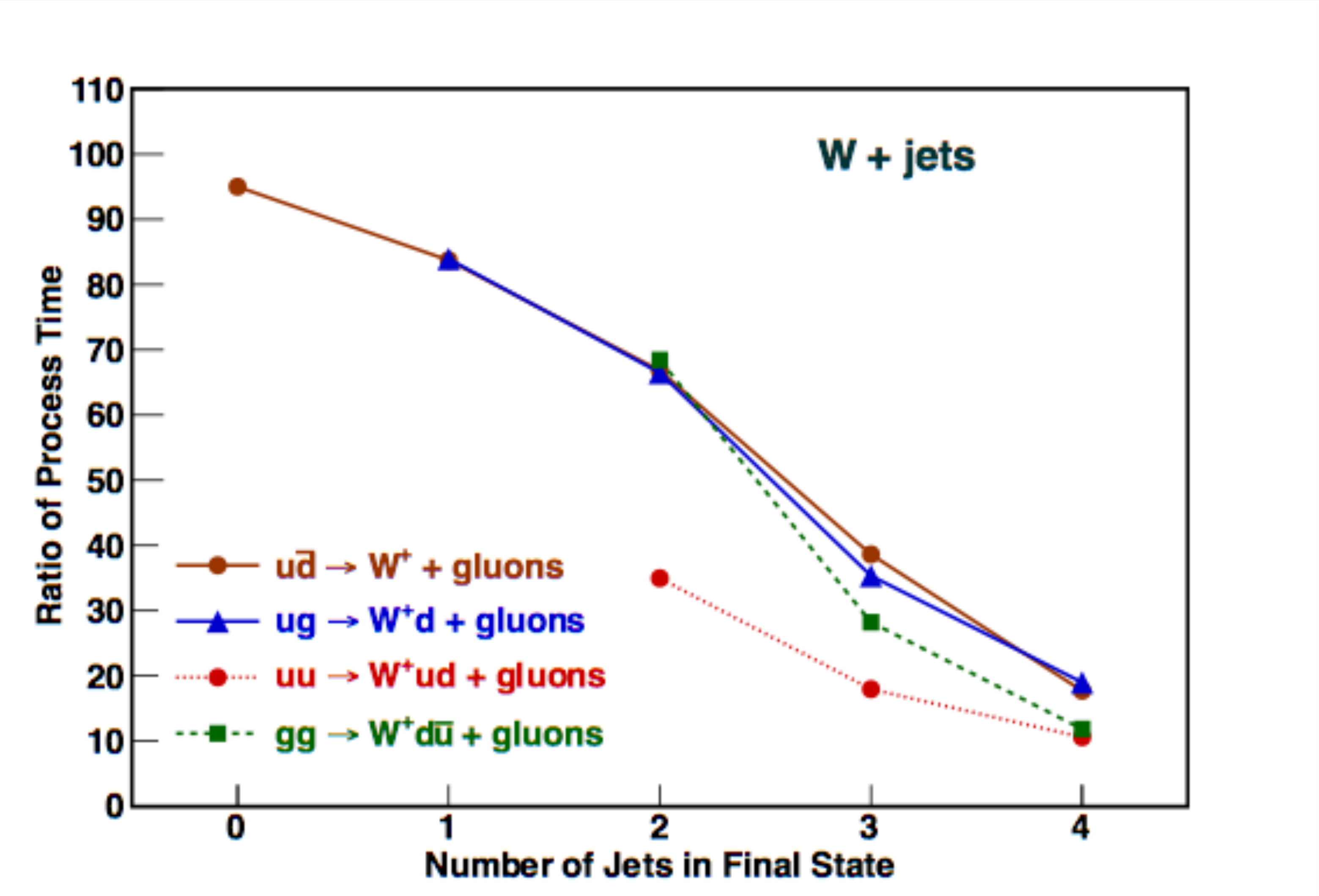
- QED: K. Hagiwara, J. Kanzaki, N. Okamura, D. Rainwater and T. Stelzer, Eur. Phys. J. C66 (2010) 477, e-print arXiv:0908.4403.
- QCD: K. Hagiwara, J. Kanzaki, N. Okamura, D. Rainwater and T. Stelzer, Eur. Phys. J. C70 (2010) 513, e-print arXiv:0909.5257.
- MC integration (VEGAS & BASES): J. Kanzaki, Eur. Phys. J. C71 (2011) 1559, e-print arXiv:1010.2107.
- SM: K. Hagiwara, J. Kanzaki, Q. Li, N. Okamura, T. Stelzer, Eur.Phys.J. C73 (2013) 2608 (2013), e-print arXiv:1305.0708v2.

Our GPU Environment

	C2075	GTX580	GTX285	GTX280	9800GTX
Streaming Processors	448	512	240	←	128
Global Memory	5.4GB	1.5GB	2GB	1GB	500MB
Constant Memory	64KB	64KB	64KB	←	64KB
Shared Memory/block	48KB	48KB	16KB	←	16KB
Registers/block	32768	32768	16384	←	8192
Warp Size	32	32	32	←	32
Clock Rate	1.15GHz	1.54GHz	1.30GHz	←	1.67GHz

- NVDIA GPUs + CUDA
- C2075@Illinois: Peak floating point performance 1.03 TFLops (single), 515 GFlops (double)

Ratio of Total Integration Time



- Comparison of total execution time in double precision.

Software development

- Inspired by Dave's proposal we have been working on the development of "CUDA" code to compute amplitude/cross sections of physics processes on GPU since early 2008.
- We converted the FORTRAN HELAS code into the CUDA code (HEGET) which can be executed on the NVIDIA's GPU.
- GPU versions of Monte-Carlo integration and event generation packages, BASES/SPRING (VEGAS), were developed and their performances were tested.

→ Installation to MG5:

- CUDA code generation with ALOHA (by O. Mattelaer)
- Installation of GPU ver. of MC integration code

Physics event simulation on GPU

- Hard collision processes
  - Efficient calculation and generation on GPU is possible.
  - Working on the installation to MG5.
  - Multi-parton amplitude with recursive relations (Y. Takaesu)
- In order to simulate physics processes at colliders we still need further components.
  - Parton shower (+hadronization) and their matching
- Fast detector simulation

  PGS in MG → already running on GPU

Parton shower (in progress)

- Based on the coherent branching algorithm by G. Kirilin and K. Hagiwara, we are developing (FSR & ISR) parton shower code.
  - Core part of parton shower evolution was already coded (S.-F. Ge & J.K.).
  - Attaching hadronization code.
  - Test with real data.
  - Include γ radiations (started by K. Mawatari)
  - Porting them on to GPU.
- Memory requirement and total performance should be carefully checked on GPU.

Summary & Prospect

- The application of GPU to event simulation/generation is on going a la MG way.
- Improvement factor of performance of computations of cross sections and generation of events becomes ~(10-100) (with our GPU device).
- Installation of hard process generation code to MG5 is in progress.
- For complete event generation for hadron colliders further software components should be developed aiming for the execution on GPU:
  - Parton shower (+hadronization)
  - PS matching (CKKW-L merging by J. Nakamura)
- We still have a long way to go, but it should be achieved in near future for the LHC HL physics.