

Graphics processing units (GPUs) for LO matrix-element evaluations

[Talk @ CERN theory institute HO10]

Jan Winter ^a

– Fermilab –



- *Introduction*
- *Hardware structure and GPU memory*
- *Monte Carlo for LO leading-colour n -gluon MEs*
- *Results – timing, cross sections, distributions*

^a In collaboration with: W. Giele and G. Stavenga

Some words in the beginning

- Fast tree-level event generators are needed for multi-particle final states.
 - ⇒ evaluation time for event generation is crucial as one needs to average over many events to obtain good statistics for cross sections and observables
 - ⇒ not only @ LO, also @ NLO to calculate real-emission corrections ...
 - ⇒ ... and tree-level matrix elements when using generalized unitarity-cut methods to determine the virtual corrections
- Throw large computer farms/grids at the problem.
 - expensive; require certain infrastructure and maintenance
 - **What if problem could be handled on a single, affordable PC ?**
- Graphical Processor Units (GPUs) in addition to CPUs give an option. Explore capabilities.
 - ⇒ first applications within the framework of HELAS ME generator [[HAGIWARA, KANZAKI, OKAMURA, RAINWATER, STELZER, ARXIV:0909.5257, ARXIV:0908.4403](#)]
 - ⇒ can tame but not overcome factorial scaling of Feynman diagrammatic approach
- Define the project: LO LC n-gluon scattering cross sections. ⇒ Tools needed ...
 - unit-weight phase-space generator ... implementation of RAMBO [[KLEISS, STIRLING, ELLIS](#)]
 - strong-coupling evaluation, PDFs using LHAPDF and observables
 - $gg \rightarrow 2, \dots, 10$ g MEs ... Berends–Giele ordered recursions, use threading to tame n^4 -scaling

GPU hardware and programming principles

[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

- The C1060 Nvidia Tesla GPU is a plug-in card for your desktop. GPU has its own memory.
- The Tesla chip is designed for numerical applications and programmable in C/limited C++.
- The chip has 30 multi-processors (MPs), each comes with 1024 processors (threads). Each thread has an unique number (for I/O etc.). Threads essentially execute same processor instructions over different data (... can skip ahead and wait for others to catch up).
- Desirable: trivial parallelization (Monte Carlo algorithms: 1 event per thread). So, in principle we can run 30720 MC generators in parallel, each running N events ... a speed-up of 30000 !!

● Approach limited by amount of available fast-access memory.

- off-chip slow-access memory: 4 Gb; use for I/O only ... transfer to and bin results on CPU
- on-chip fast-access memory: only kbs; registers and shared memory
 - ⇒ 16384 32-bit registers per MP; once assigned only seen by specific thread; temporary storage for function evaluations
 - ⇒ 16384 bytes shared memory per MP; **accessible to all threads**

Memory layout

[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

- The n -gluon recursion relation needs n momenta and $n(n - 1)/2$ currents for a total of $n(n + 1)/2$ single precision 4-vectors.
 - Recursion relations are very suitable for GPU (memory efficient & algorithmically simple).
- We need $(4 \cdot 4) n(n + 1)/2$ bytes of fast accessible memory per event.
- This means $16384/(8 n(n + 1))$ events per MP.
- One constraint though: implementation needs **35 registers** per thread, i.e. $16384/35=468$ **threads** are useable **per MP**.

n	4	5	6	7	8	9	10	11	12
events per MP	102	68	48	36	28	22	18	15	13
avail. threads / evt	10	15	21	28	36	45	55	66	78
useable threads / evt	4	6	9	13	16	21	26	31	36

⇒ used threads per event = $n - 1$.

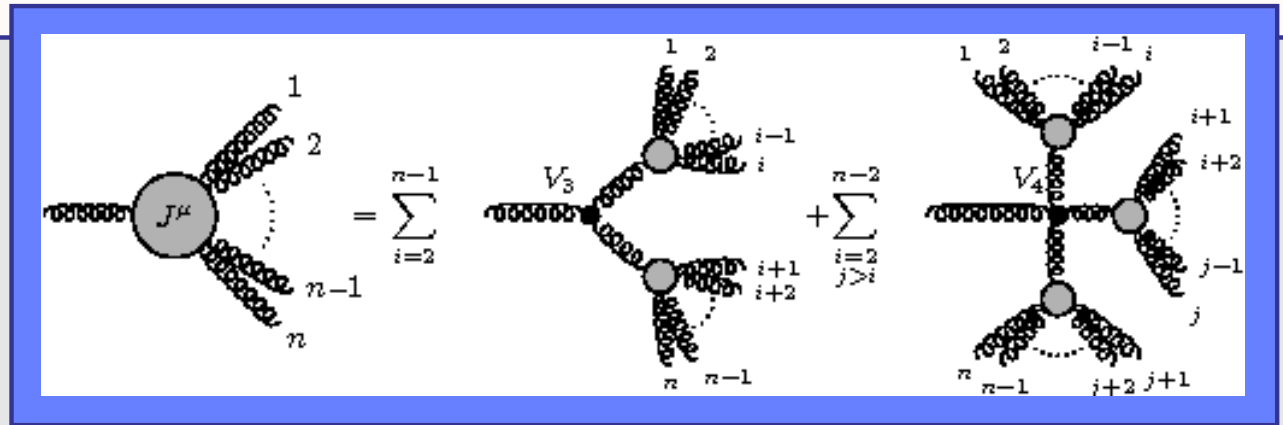
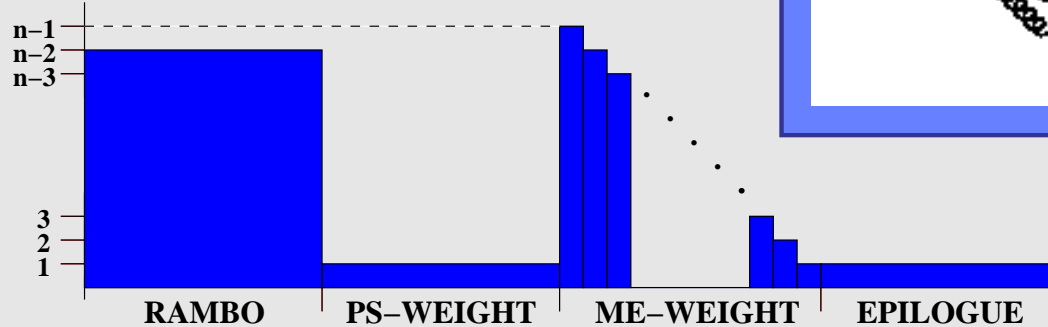
Processing events

[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

Tess MC program

(vircol.fnal.gov/TESS.html)

Thread usage per event



⇒ compute ME weight with BG recursion relation.

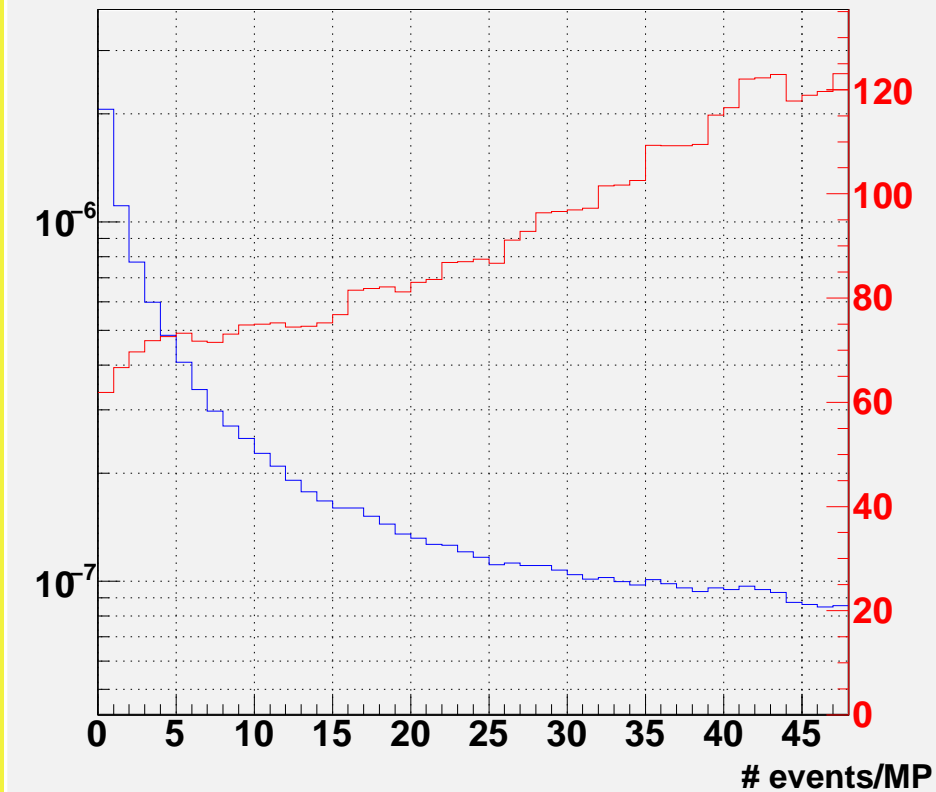
- Initialization phase: not shown.
- Rambo phase: $n - 2$ threads to construct outgoing momenta.
- Phase-space weight: off-chip texture memory to store $\alpha_s(\mu)$ 1D and PDF $f_g(x, \mu)$ 2D grids.
 - ⇒ hardware performs linear interpolation between grid points for non-integer values
- ME weight: computed in $n - 1$ steps instead of $n(n - 1)/2$.
 - ⇒ reduces computational effort from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$ complexity
 - ⇒ avoid complex multiplications; choose polarization vecs such that one has real-valued currents
 - ⇒ leading colour to avoid colour sum; use symmetry of FS (gluons only) to remove permutation sum over orderings

Computation times

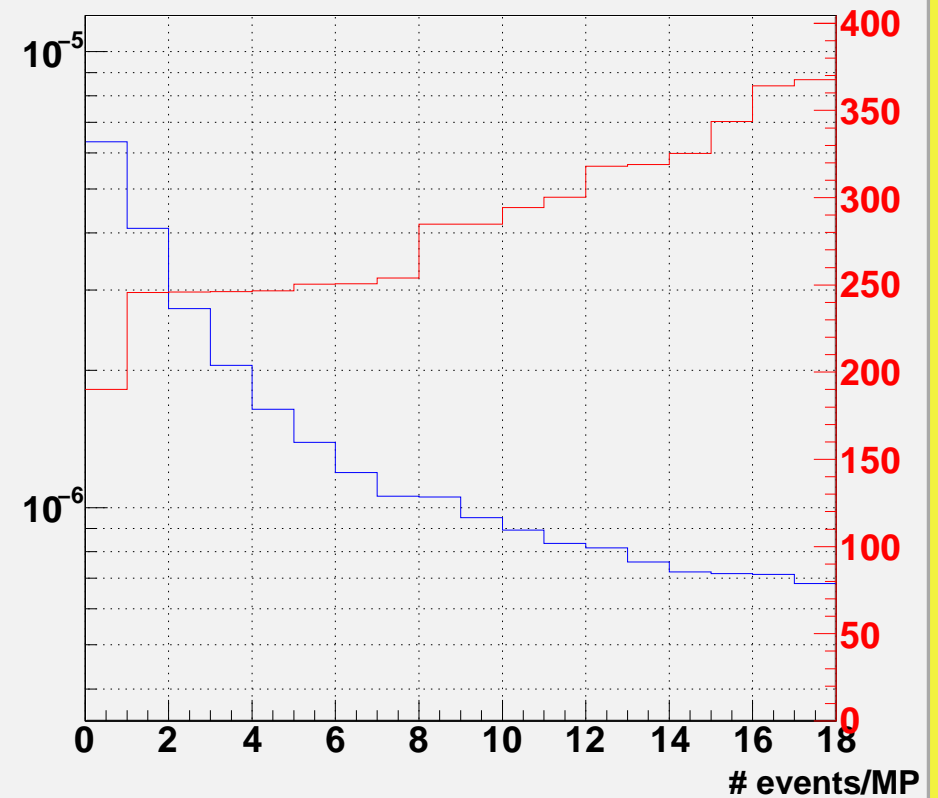
[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

- Red curves: total GPU time in seconds to evaluate 10^6 sweeps ($30 \times \text{events/MP}$).
- Sweep time should be independent of $\# \text{evts/MP}$, but queuing effects due to substantial amount of special-function calls.
- Blue curves: evaluation time per event = GPU time / total $\# \text{evts}$.

Timing profile in seconds for 6g



Timing profile in seconds for 10g



⇒ best performance if max $\# \text{evts}$ available per MP.

Timing

[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

- Compare evaluation time per event on GPU with that of running the same algorithm on CPU [AMD Phenom(tm) II X4 940 (3 GHz)]. $P_n(m) = [(n-1)/n]^m \sqrt[m]{T_n/T_{n-1}}$

- Speed-ups occur because events are evaluated in parallel.

n	T_n^{GPU} (seconds)	$P_n(3)$	T_n^{CPU} (seconds)	$P_n(4)$	G_n
4	2.975×10^{-8}		8.753×10^{-6}		294
5	4.438×10^{-8}	0.91	1.247×10^{-5}	0.87	281
6	8.551×10^{-8}	1.03	1.966×10^{-5}	0.93	230
7	2.304×10^{-7}	1.19	3.047×10^{-5}	0.96	132
8	3.546×10^{-7}	1.01	4.736×10^{-5}	0.98	133
9	4.274×10^{-7}	0.94	7.263×10^{-5}	0.99	170
10	6.817×10^{-7}	1.05	1.044×10^{-4}	0.99	153
11	9.750×10^{-7}	1.02	1.529×10^{-4}	1.00	157
12	1.356×10^{-6}	1.02	2.129×10^{-4}	1.00	158

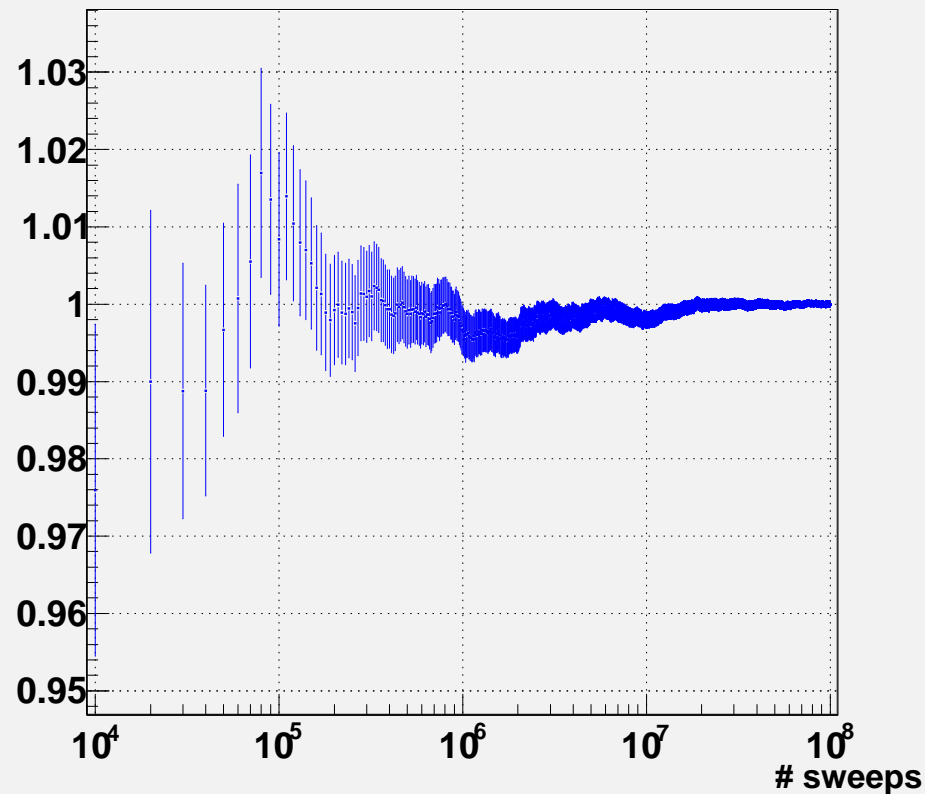
⇒ Gain.

LO LC multi-gluon cross sections

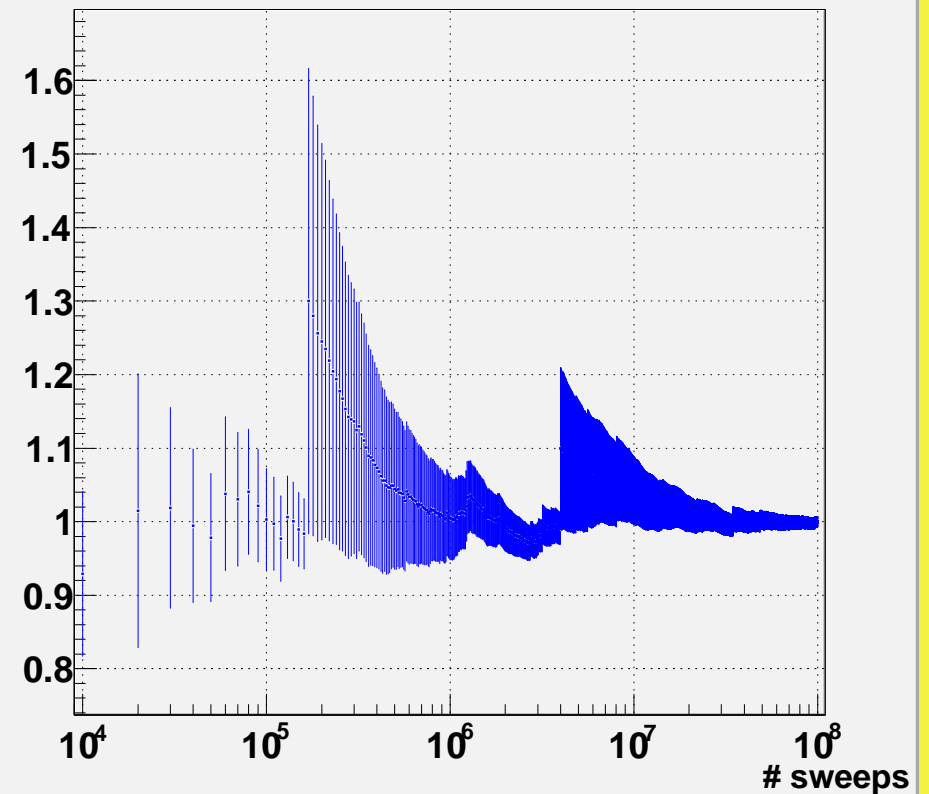
[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

- Convergence of cross section normalized to best xsec estimate.
- 14 TeV LHC, cteq6l1 PDFs, $\mu_F = \mu_R = M_Z$, $p_T^{jet} > 20$ GeV, $|\eta^{jet}| < 2.5$ and $\Delta R_{jet-jet} > 0.4$.
- Subtleties: random-number generator cycle $>$ total #evts; Kahan summation to avoid loss of precision (averaging $\mathcal{O}(10^{11})$ numbers).

The normalized 6g cross section



The normalized 8g cross section

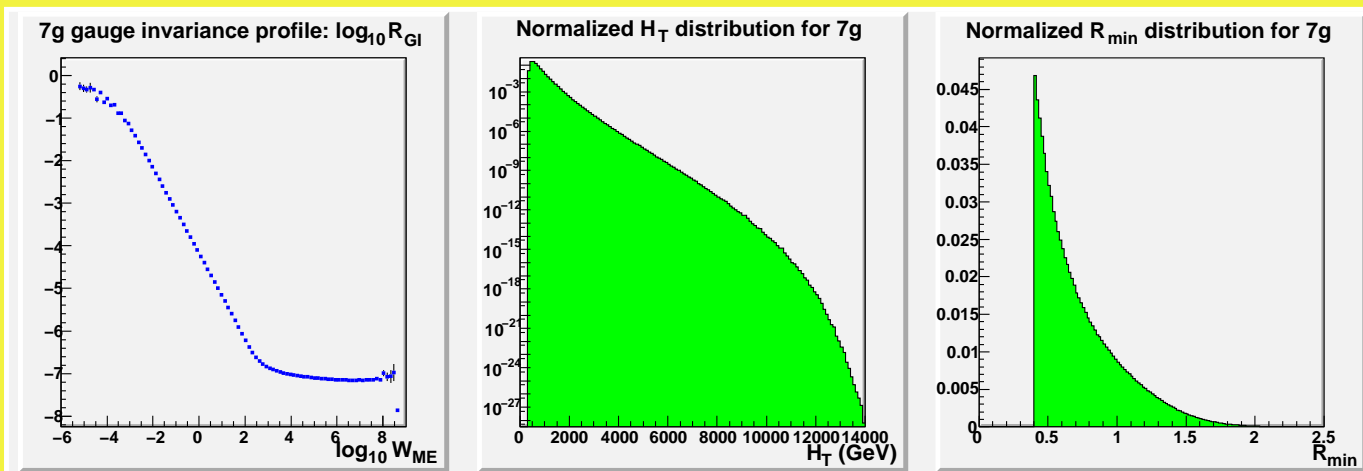


⇒ flat phase-space integration may under-sample regions of large weights → peaks.

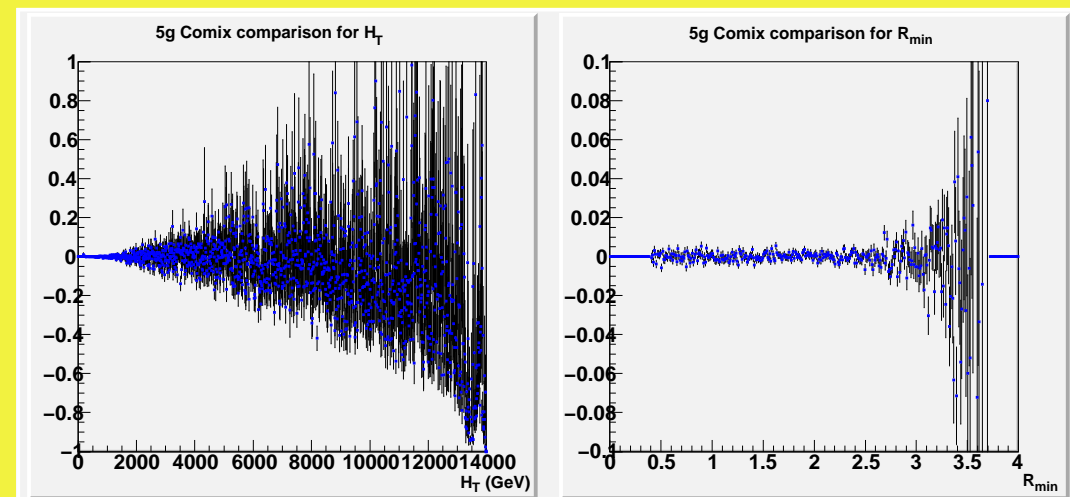
Distributions

[GIELE, STAVENGA, WINTER, ARXIV:1002.3446]

- Observables: H_T and minimum R-separation $R_{min} = \min\{R_{ij}\}$ normalized to total xsec.
- 14 TeV LHC, cteq6l1 PDFs, $\mu_F = \mu_R = H_T$ (upper), M_Z (lower), $p_T^{jet} > 60$ GeV, $|\eta^{jet}| < 2.0$ and $\Delta R_{jet-jet} > 0.4$.
- upper: example for $gg \rightarrow 5g$; lower: $gg \rightarrow 3g$ compared to COMIX. [GLEISBERG, HÖCHE]



COMIX uses importance sampling, i.e. tail less populated, i.e. larger uncertainties @ large H_T .



Summary

- We obtained encouraging results in our first exploration of the potential of using multi-threaded GPU based workstations for Monte Carlo programming.
- Testbed chosen: leading-order leading colour n -gluon scattering.
⇒ **Tess** Monte Carlo program.
- Wrt the CPU based implementation, we found speed-ups ranging from $\mathcal{O}(300)$ and $\mathcal{O}(150)$ for 4-gluon and 12-gluon scattering, respectively.
- Outlook: @ LO – include quarks, vector bosons, subleading colour contributions, replace Rambo by Sarge. [VAN HAMEREN, PAPADOPOULOS]
⇒ application to NLO
- GPU chips are still evolving rapidly ... next generation, Fermi chip (Fall 2010).