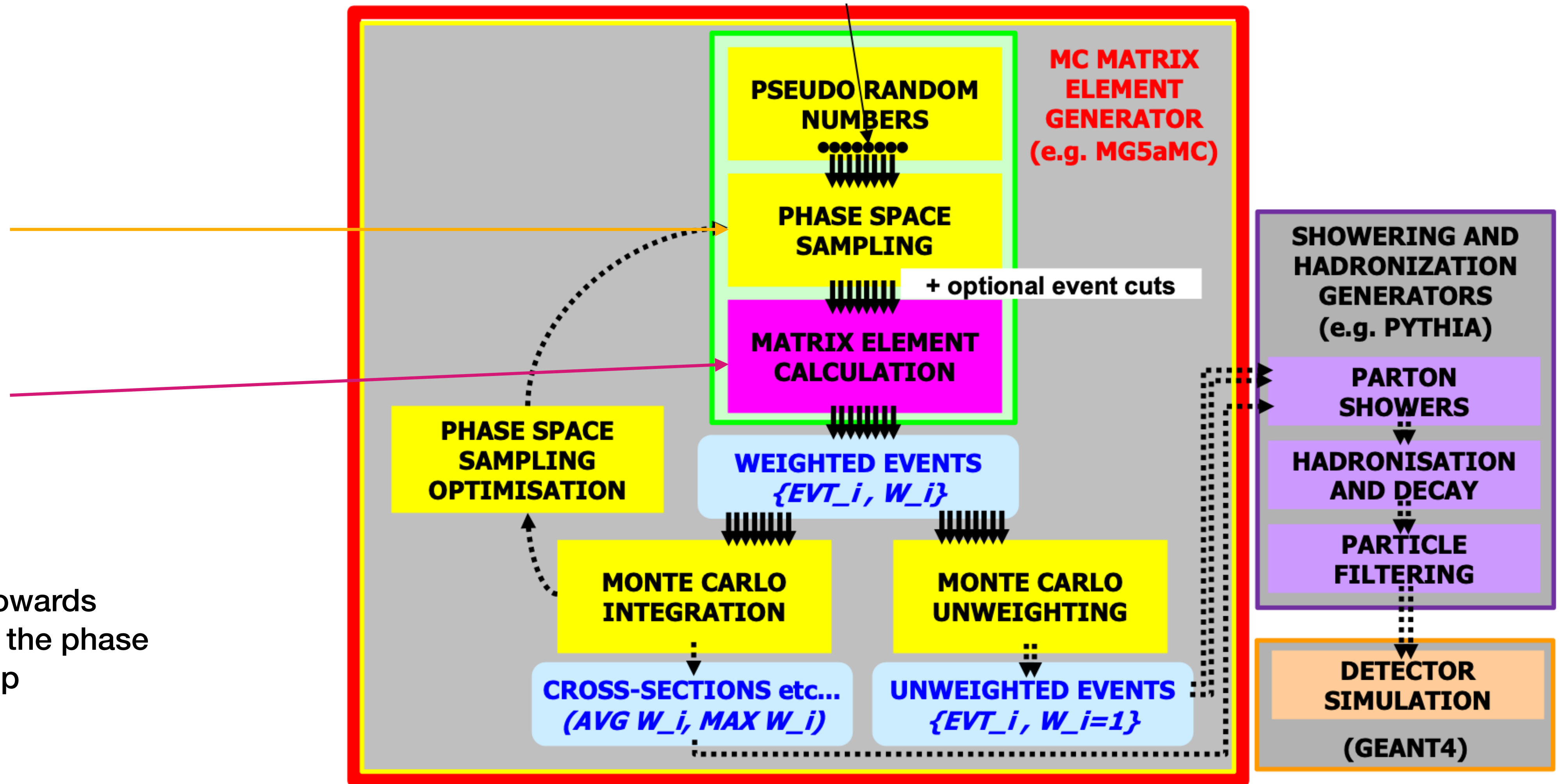


MadGraph5_aMC on GPUs

Vectorizing Phase Space Sampling

A. Thete & C. Vuosalo
University of Wisconsin-Madison

MadGraph5_aMC Computational Workflow

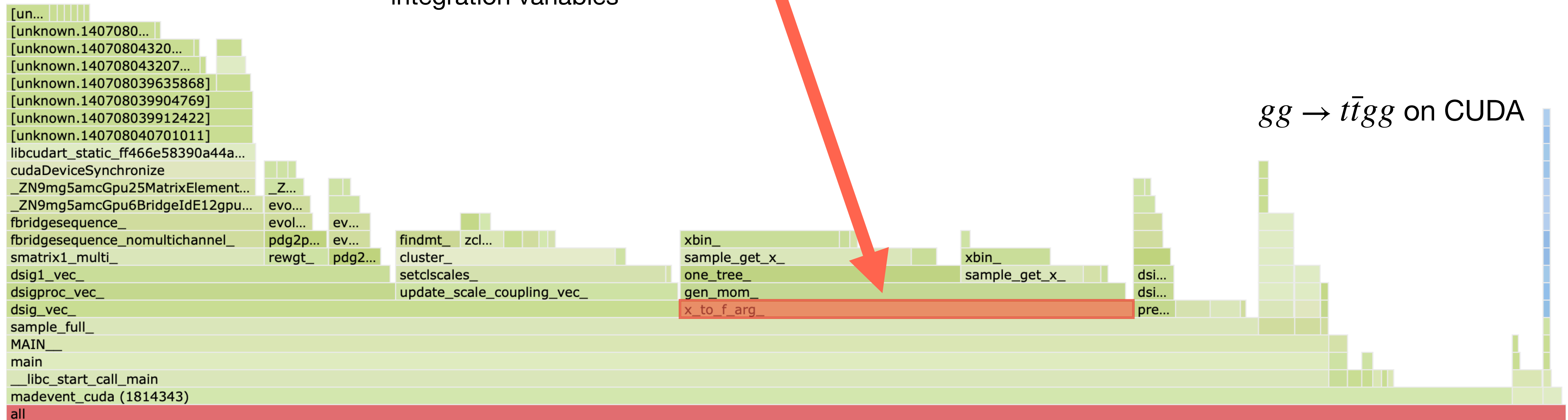


This work: Efforts towards vectorizing parts of the phase space sampling loop

A. Valassi (vCHEP 2021)

MadGraph5_aMC Phase Space Sampling

- Phase space sampling is the second-largest performance bottleneck [29%]
- Focus on `x_to_f_arg_/gen_mom_`
 - Converts kinematic variables (x, p) to MC integration variables



MadGraph5_aMC Phase Space Sampling

- However, there are major issues that complicate conversion:
 - `x_to_f_arg` is **NOT reentrant!**
 - Many common blocks accessed to read and write state variables (particularly event-level information that is important for MC later)
 - “Stateful” operations are not thread-safe :(
 - `x_to_f_arg` has lots of **branching!**
 - Within the function body, a lot of operations depend on event-level information which gives rise to MANY branch divergences.
- To make the conversion more tractable and eliminate many complications, development is focused only on a single process and reduced-scale vectorization.
 - $pp \rightarrow e^+ e^-$ process to trigger only a small part of the control flow
 - Vectorize over a single warp as opposed to all events.
 - A. Valassi's slides (20/08) mention two low hanging fruit: modifications within `sample_get_x`; this work is broader — does not change `sample_get_x`, modifies the host instead.

Phase Space Sampling: Work Done So Far

- Modify lines in `gen_mom()` triggered for a 2->1 process
- Basic strategy is to wrap the lines within a `do..while` loop over the current warp and populate the vectorized variables (`all_*`)
 - `sample_get_x` unchanged.

Before

After

```
c-----
c tjs 5/24/2010 for 2->1 process
c-----
      call sample_get_x(sjac,x(ndim-1),ndim-1,mincfg,0d0,1d0)
      xtau = x(ndim-1)
      if(nexternal .eq. 3) then
        x(ndim-1) = pmass(3)*pmass(3)/stot
        sjac=1 / stot      !for delta function in d_tau
      endif
      call sample_get_x(sjac,x(ndim),ndim,mincfg,0d0,1d0)
      CALL GENCMS(STOT,Xbk(1),Xbk(2),X(ndim-1), SMIN, SJAC)
      x(ndim-1) = xtau      !Fix for 2->1 process
c      Set CM rapidity for use in the rap() function
      cm_rap=.5d0*dlog(xbk(1)*ebeam(1)/(xbk(2)*ebeam(2)))
      set_cm_rap=.true.
c      Set shat
c all xbks to populate s
      s(-nbranch) = xbk(1)*xbk(2)*stot
      endif
```

```
c-----
c tjs 5/24/2010 for 2->1 process
c-----
      do idx = 1 + loop_offset, WARP_SIZE + loop_offset
        call sample_get_x(all_sjac(idx),all_x(ndim-1,idx),ndim-1,mincfg,0d0,1d0)
        all_lastbin(ndim-1, idx) = lastbin(ndim - 1)
        xtau = all_x(ndim-1,idx)

        if(nexternal .eq. 3) then
          all_x(ndim-1, idx) = pmass(3)*pmass(3)/stot
          all_sjac(idx)=1 / stot      !for delta function in d_tau
        endif

        call sample_get_x(sjac,all_x(ndim, idx),ndim,mincfg,0d0,1d0)
        CALL GENCMS(STOT,all_xbk(1, idx),all_xbk(2, idx), all_x(ndim-1,idx), SMIN, all_sjac(idx))
        all_x(ndim-1,idx) = xtau      !Fix for 2->1 process

c      Set CM rapidity for use in the rap() function
        all_cm_rap(idx)=.5d0*dlog(all_xbk(1, idx)*ebeam(1)/(all_xbk(2, idx)*ebeam(2)))
c      Set shat
        all_s(-nbranch, idx) = all_xbk(1,idx)*all_xbk(2,idx)*stot
      enddo
      set_cm_rap=.true.
      endif
```

Phase Space Sampling: Work Done So Far

- Need to also make changes to the event loop in `dsample.f`
- Snippet on the left can all be replaced by a single call to `x_to_f_arg` with `all_x`, `all_wgt`, and `all_p` as arguments instead.
- Vectorizing beyond one warp is becoming a problem, too many branching statements.
- Whole workflow with changes doesn't work :(at the moment because I'm trying to also vectorize the momentum transformations in lines 402-443 and I don't know which lines are activated.
- Only way to exchange one vectorized variable (`all_lastbin`) between `gen_mom` and the main sampling loop was to add it to a common block, which might not be the best strategy.
- Would love to hear feedback about what the best way forward will be.

`dsample.f`

```
c
c      call sample_get_config(wgt,iter,ipole)
c      if (iter .le. itmax) then
c          write(*,*) 'iter/ievent/ivec', iter, ievent, ivec
c          ievent=ievent+1
c pass all_p, all_x all_wgt, all_cm_rap as arg
c      call x_to_f_arg(ndim,ipole,mincfg,maxcfg,ninvar,wgt,x,p)
c      CUTSDONE=.FALSE.
c      CUTSPASSED=.FALSE.
c      if (passcuts(p,VECSIZE_USED)) then
c          ivec=ivec+1
c          ilock = ilock+1
c          if (ilock.gt.WARP_SIZE)then
c              ilock = 1
c              iwarp = iwarp +1
c          endif
c      write(*,*) 'pass_point ivec is ', ivec
c      all_p(:,ivec) = p(:)
c      all_wgt(ivec) = wgt
c      all_x(:,ivec) = x(:)
c      all_xbk(:, ivec) = xbk(:)
c      all_q2fact(:, ivec) = q2fact(:)
c      all_cm_rap(ivec) = cm_rap
c      all_lastbin(:, ivec) = lastbin(:)
c
c      i = ivec
c      fx = dsig(all_p(1,i),all_wgt(i),0)
c      bckp(i) = fx
c      write(*,*) i, all_wgt(i), fx, all_wgt(i)*fx
c      all_wgt(i) = all_wgt(i)*fx
c      if (ilock.ne.WARP_SIZE)then
c          cycle
c      endif
```