# Accuracy meets precision
# MC event simulation in a decade

Stefan Höche

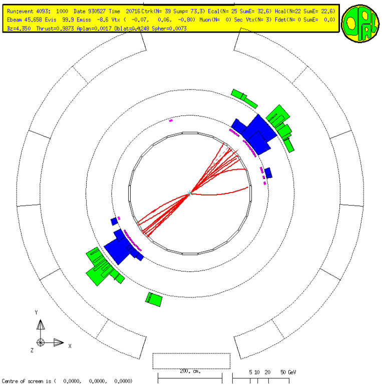
SLAC National Accelerator Laboratory

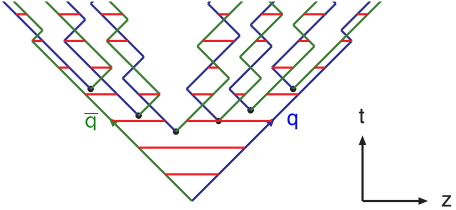
Future Trends in Nuclear Physics Computing

Jefferson Lab, 05/02/2017

# Event generators in 1978

[Andersson,Gustafson,Ingelman,Sjöstrand] Phys.Rept.97(1983)31




- Lund string model: $\sim$ like rubber band that is pulled apart and breaks into pieces, or like a magnet broken into smaller pieces.
- Complete description of 2-jet events in $e^+e^-$ $\rightarrow$hadrons

# Event generators in 1978

[Andersson,Gustafson,Ingelman,Sjöstrand] Phys.Rept.97(1983)31

```
      SUBROUTINE JETGEN(N)
      COMMON /JET/ K(100,2), P(100,5)
      COMMON /PAR/ PUD, PS1, SIGMA, CX2, EBEG, WFIN, IFLBEG
      COMMON /DATA1/ MESO(9,2), CMIX(6,2), PMAS(19)
      IFLSGN=(10-IFLBEG)/5
      W=2.*EBEG
      I=0
      IPD=0
C 1 FLAVOUR AND PT FOR FIRST QUARK
      IFL1=IABS(IFLBEG)
      PT1=SIGMA*SQRT(-ALOG(RANF(0)))
      PHI1=6.2832*RANF(0)
      PX1=PT1*COS(PHI1)
      PY1=PT1*SIN(PHI1)
  100 I=I+1
C 2 FLAVOUR AND PT FOR NEXT ANTIQUARK
      IFL2=1+INT(RANF(0)/PUD)
      PT2=SIGMA*SQRT(-ALOG(RANF(0)))
      PHI2=6.2832*RANF(0)
      PX2=PT2*COS(PHI2)
      PY2=PT2*SIN(PHI2)
C 3 MESON FORMED, SPIN ADDED AND FLAVOUR MIXED
      K(I,1)=MESO(3*(IFL1-1)+IFL2,IFLSGN)
      ISPIN=INT(PS1+RANF(0))
      K(I,2)=1+9*ISPIN+K(I,1)
      IF(K(I,1).LE.6) GOTO 110
      TMIX=RANF(0)
      KM=K(I,1)-6+3*ISPIN
      K(I,2)=8+9*ISPIN+INT(TMIX+CMIX(KM,1))+INT(TMIX+CMIX(KM,2))
C 4 MESON MASS FROM TABLE, PT FROM CONSTITUENTS
  110 P(I,5)=PMAS(K(I,2))
      P(I,1)=PX1+PX2
      P(I,2)=PY1+PY2
      PMTS=P(I,1)**2+P(I,2)**2+P(I,5)**2
C 5 RANDOM CHOICE OF X=(E+PZ)MESON/(E+PZ)AVAILABLE GIVES E AND PZ
      X=RANF(0)
      IF(RANF(0).LT.CX2) X=1.-X**(1./3.)
      P(I,3)=(X*W-PMTS/(X*W))/2.
      P(I,4)=(X*W+PMTS/(X*W))/2.
C 6 IF UNSTABLE, DECAY CHAIN INTO STABLE PARTICLES
  120 IPD=IPD+1
      IF(K(IPD,2).GE.8) CALL DECAY(IPD,I)
      IF(IPD.LT.I.AND.I.LE.96) GOTO 120
C 7 FLAVOUR AND PT OF QUARK FORMED IN PAIR WITH ANTIQUARK ABOVE
      IFL1=IFL2
      PX1=-PX2
      PY1=-PY2
C 8 IF ENOUGH E+PZ LEFT, GO TO 2
      W=(1.-X)*W
      IF(W.GT.WFIN.AND.I.LE.95) GOTO 100
      N=I
      RETURN
      END
```

```
      SUBROUTINE LIST(N)
      COMMON /JET/ K(100,2), P(100,5)
      COMMON /DATA3/ CHA1(9), CHA2(19), CHA3(2)
      WRITE(6,110)
      DO 100 I=1,N
      IF(K(I,1).GT.0) C1=CHA1(K(I,1))
      IF(K(I,1).LE.0) IC1=-K(I,1)
      C2=CHA2(K(I,2))
      C3=CHA3((47-K(I,2))/20)
      IF(K(I,1).GT.0) WRITE(6,120) I, C1, C2, C3, (P(I,J), J=1,5)
  100 IF(K(I,1).LE.0) WRITE(6,130) I, IC1, C2, C3, (P(I,J), J=1,5)
      RETURN
  110 FORMAT(////T11,'I',T17,'ORI',T24,'PART',T32,'STAB',
     &T44,'PX',T56,'PY',T68,'PZ',T80,'E',T92,'M'/)
  120 FORMAT(10X,I2,4X,A2,1X,2(4X,A4),5(4X,F8.1))
  130 FORMAT(10X,I2,4X,1X,I2,2(4X,A4),5(4X,F8.1))
      END
```

```
      SUBROUTINE DECAY(IPD,I)
      COMMON /JET/ K(100,2), P(100,5)
      COMMON /DATA1/ MESO(9,2), CMIX(6,2), PMAS(19)
      COMMON /DATA2/ IDCO(12), CBR(29), KDP(29,3)
      DIMENSION U(3), BE(3)
C 1 DECAY CHANNEL CHOICE, GIVES DECAY PRODUCTS
      TBR=RANF(0)
      IDC=IDCO(K(IPD,2)-7)
  100 IDC=IDC+1
      IF(TBR.GT.CBR(IDC)) GOTO 100
      ND=(59+KDP(IDC,3))/20
      DO 110 I1=I+1,I+ND
      K(I1,1)=-IPD
      K(I1,2)=KDP(IDC,I1-I)
  110 P(I1,5)=PMAS(K(I1,2))
C 2 IN THREE-PARTICLE DECAY CHOICE OF INVARIANT MASS OF PRODUCTS 2+3
      IF(ND.EQ.2) GOTO 130
      SA=(P(IPD,5)+P(I+1,5))**2
      SB=(P(IPD,5)-P(I+1,5))**2
      SC=(P(I+2,5)+P(I+3,5))**2
      SD=(P(I+2,5)-P(I+3,5))**2
      TDU=(SA-SD)*(SB-SC)/(4.*SQRT(SB*SC))
      IF(K(IPD,2).GE.11) TDU=SQRT(SB*SC)*TDU**3
  120 SX=SC+(SB-SC)*RANF(0)
      TDF=SQRT((SX-SA)*(SX-SB)*(SX-SC)*(SX-SD))/SX
      IF(K(IPD,2).GE.11) TDF=SX*TDF**3
      IF(RANF(0)*TDU.GT.TDF) GOTO 120
      P(100,5)=SQRT(SX)
C 3 TWO-PARTICLE DECAY IN CM, TWICE TO SIMULATE THREE-PARTICLE DECAY
  130 DO 160 IL=1,ND-1
      IO=(IL-1)*100-(IL-2)*IPD
      I1=I+IL
      I2=(ND-IL-1)*100-(ND-IL-2)*(I+IL+1)
      PA=SQRT((P(IO,5)**2-(P(I1,5)+P(I2,5))**2)*
     &(P(IO,5)**2-(P(I1,5)-P(I2,5))**2))/(2.*P(IO,5))
  140 U(3)=2.*RANF(0)-1.
      PHI=6.2832*RANF(0)
      U(1)=SQRT(1.-U(3)**2)*COS(PHI)
      U(2)=SQRT(1.-U(3)**2)*SIN(PHI)
      TDA=1.-(U(1)*P(IO,1)+U(2)*P(IO,2)+U(3)*P(IO,3))**2/
     &(P(IO,1)**2+P(IO,2)**2+P(IO,3)**2)
      IF(K(IPD,2).GE.11.AND.IL.EQ.2.AND.RANF(0).GT.TDA) GOTO 140
      DO 150 J=1,3
      P(I1,J)=PA*U(J)
  150 P(I2,J)=-PA*U(J)
      P(I1,4)=SQRT(PA**2+P(I1,5)**2)
  160 P(I2,4)=SQRT(PA**2+P(I2,5)**2)
C 4 DECAY PRODUCTS LORENTZ TRANSFORMED TO LAB SYSTEM
      DO 190 IL=ND-1,1,-1
      IO=(IL-1)*100-(IL-2)*IPD
      DO 170 J=1,3
  170 BE(J)=P(IO,J)/P(IO,4)
      GA=P(IO,4)/P(IO,5)
      DO 190 I1=I+IL,I+ND
      BEP=BE(1)*P(I1,1)+BE(2)*P(I1,2)+BE(3)*P(I1,3)
      DO 180 J=1,3
  180 P(I1,J)=P(I1,J)+GA*(GA/(1.+GA)*BEP+P(I1,4))*BE(J)
  190 P(I1,4)=GA*(P(I1,4)+BEP)
      I=I+ND
      RETURN
      END
```

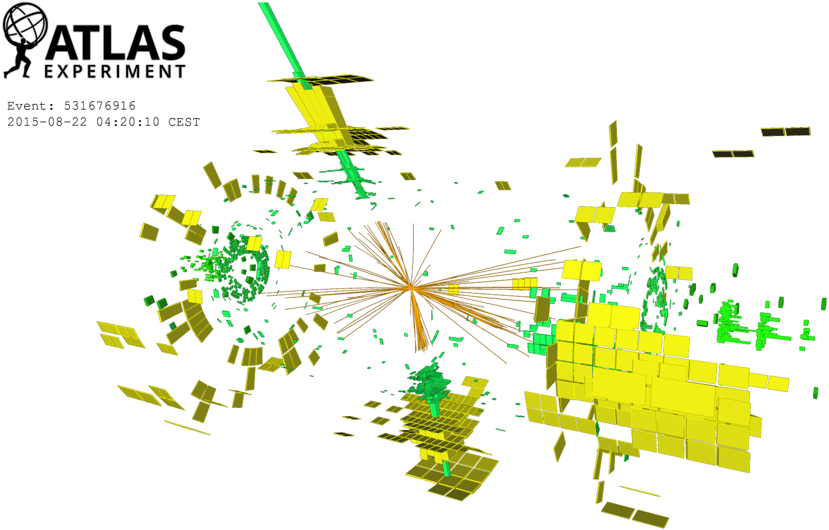
≈ 200 punched cards

Fortran code

```
      SUBROUTINE EDIT(N)
      COMMON /JET/ K(100,2), P(100,5)
      COMMON /EDPAR/ ITHROW, PZMIN, PMIN, THETA, PHI, BETA(3)
      REAL ROT(3,3), PR(3)
C 1 THROW AWAY NEUTRALS OR UNSTABLE OR WITH TOO LOW PZ OR P
      I1=0
      DO 110 I=1,N
      IF(ITHROW.GE.1.AND.K(I,2).GE.8) GOTO 110
      IF(ITHROW.GE.2.AND.K(I,2).GE.6) GOTO 110
      IF(ITHROW.GE.3.AND.K(I,2).EQ.1) GOTO 110
      IF(P(I,3).LT.PZMIN.OR.P(I,4)**2-P(I,5)**2.LT.PMIN**2) GOTO 110
      I1=I1+1
      K(I1,1)=IDIM(K(I,1),0)
      K(I1,2)=K(I,2)
      DO 100 J=1,5
  100 P(I1,J)=P(I,J)
  110 CONTINUE
      N=I1
C 2 ROTATE TO GIVE JET PRODUCED IN DIRECTION THETA, PHI
      IF(THETA.LT.1E-4) GOTO 140
      ROT(1,1)=COS(THETA)*COS(PHI)
      ROT(1,2)=-SIN(PHI)
      ROT(1,3)=SIN(THETA)*COS(PHI)
      ROT(2,1)=COS(THETA)*SIN(PHI)
      ROT(2,2)=COS(PHI)
      ROT(2,3)=SIN(THETA)*SIN(PHI)
      ROT(3,1)=-SIN(THETA)
      ROT(3,2)=0.
      ROT(3,3)=COS(THETA)
      DO 130 I=1,N
      DO 120 J=1,3
  120 PR(J)=P(I,J)
      DO 130 J=1,3
  130 P(I,J)=ROT(J,1)*PR(1)+ROT(J,2)*PR(2)+ROT(J,3)*PR(3)
C 3 OVERALL LORENTZ BOOST GIVEN BY BETA VECTOR
  140 IF(BETA(1)**2+BETA(2)**2+BETA(3)**2.LT.1E-8) RETURN
      GA=1./SQRT(1.-BETA(1)**2-BETA(2)**2-BETA(3)**2)
      DO 160 I=1,N
      BEP=BETA(1)*P(I,1)+BETA(2)*P(I,2)+BETA(3)*P(I,3)
      DO 150 J=1,3
  150 P(I,J)=P(I,J)+GA*(GA/(1.+GA)*BEP+P(I,4))*BETA(J)
  160 P(I,4)=GA*(P(I,4)+BEP)
      RETURN
      END
```

```
      BLOCK DATA
      COMMON /PAR/ PUD, PS1, SIGMA, CX2, EBEG, WFIN, IFLBEG
      COMMON /EDPAR/ ITHROW, PZMIN, PMIN, THETA, PHI, BETA(3)
      COMMON /DATA1/ MESO(9,2), CMIX(6,2), PMAS(19)
      COMMON /DATA2/ IDCO(12), CBR(29), KDP(29,3)
      COMMON /DATA3/ CHA1(9), CHA2(19), CHA3(2)
      DATA PUD/0.4/, PS1/0.5/, SIGMA/350./, CX2/0.77/,
     &EBEG/10000./, WFIN/100./, IFLBEG/1/
      DATA ITHROW/1/, PZMIN/0./, PMIN/0./, THETA,PHI,BETA/5*0./
      DATA MESO/7,1,3,2,8,5,4,6,9,7,2,4,1,8,6,3,5,9/
      DATA CMIX/2*0.5,1.,2*0.5,1.,2*0.25,0.5,2*0.,1./
      DATA PMAS/0.,2*139.6,2*493.7,2*497.7,135.,548.8,957.6,
     &2*765.9,2*892.2,2*896.3,770.2,782.6,1019.6/
      DATA IDCO/0,1,6,11,12,13,15,17,19,21,22,25/
      DATA CBR/1.,0.381,0.681,0.918,0.969,1.,0.426,0.662,0.959,
     &0.980,1.,1.,1.,0.667,1.,0.667,1.,0.667,1.,0.667,1.,1.,
     &0.899,0.987,1.,0.486,0.837,0.984,1./
      DATA KDP/1,1,8,2,1,1,2,8,1,1,1,2,3,6,4,7,5,4,6,5,7,2,2,
     &1,2,4,6,2,1,1,1,8,3,2,1,3,8,17,18,1,8,8,2,8,3,8,3,8,2,8,
     &3,3,8,3,5,7,3,9,0,0,8,8,3,8,9,9,14*0,8,4*0,8,0/
      DATA CHA1/'UD','DU','US','SU','DS','SD','UU','DD','SS'/
      DATA CHA2/'GAMM','PI+','PI-','K+','K-','K0','KB0','PI0','ETA',
     &'ETAP','RHO+','RHO-','K*+','K*-','K*0','KB*0','RHO0','OMEG','PHI'/
      DATA CHA3/'    ','STAB'/
      END
```

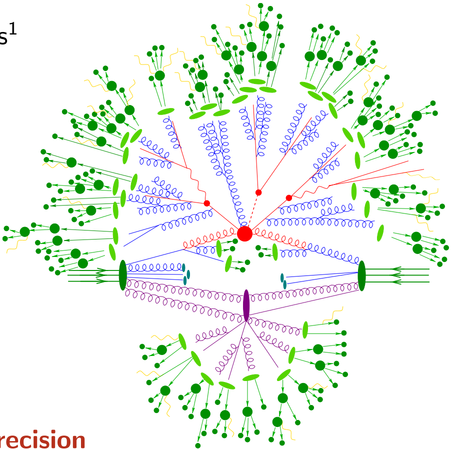
# Experimental situation in 2016



# Event generators in 2016

[Buckley et al.] arXiv:1101.2599

- (N)LO Matrix Element generators[1]
  Comix, MadGraph5, Whizard, ...
- Parton showers, mostly based
  on dipole/antenna picture
- Multiple interaction models
  possibly interleaved with shower
- Hadronization models
  string/cluster fragmentation
- Hadron decay packages
- Photon emission generators
  YFS formalism or QED shower



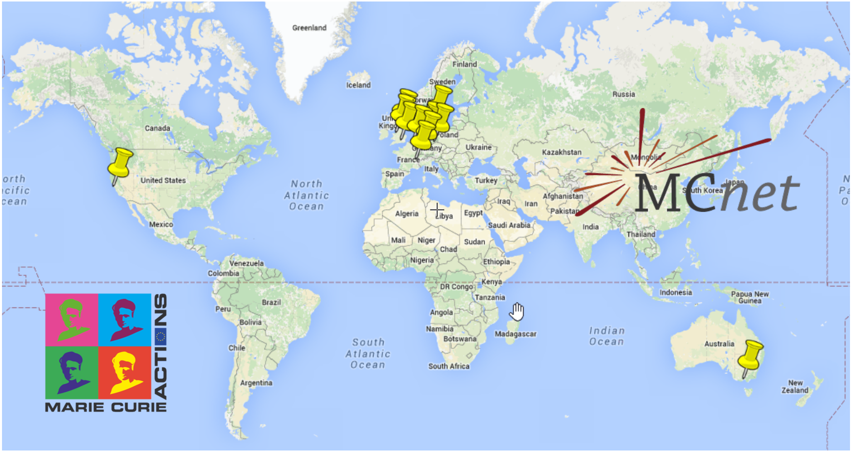
**Development focused on higher precision**

[1] Virtual corrections sold separately ↗ later

# Accuracy and Precision (A. David)


[N. Glover at Scales WS 2017]

# The developers



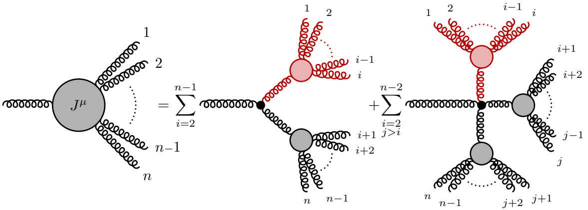

7 Countries, 12+ Institutes, $\sim$ 100 Scientists

# Computing short-distance cross sections – LO

- Computing Feynman diagrams gets complicated very quickly

| # of gluons | # of diagrams | # of gluons | # of diagrams |
|---|---|---|---|
| 5 | 25 | 9 | 559405 |
| 6 | 220 | 10 | 10525900 |
| 7 | 2485 | 11 | 224449225 |
| 8 | 34300 | 12 | 5348843500 |

- Must eliminate common subexpressions [Berends,Giele] NPB306(1988)759




- Ideally also parallelize the calculation [Gleisberg,SH] arXiv:0808.3674

| gg → ng | Cross section [pb] | | | | |
|---|---|---|---|---|---|
| n | 8 | 9 | 10 | 11 | 12 |
| arXiv:0808.3674 | 0.755(3) | 0.305(2) | 0.101(7) | 0.057(5) | 0.026(1) |
| PRD67(2003)014026 | 0.70(4) | 0.30(2) | 0.097(6) | | |
| NPB539(1999)215 | 0.719(19) | | | | |

# Computing short-distance cross sections – NLO

- Infrared subtraction at NLO generalized ∼20 years ago
  [Frixione,Kunszt,Signer] hep-ph/9512328, [Catani,Seymour] hep-ph/9605323
  [Catani,Dittmaier,Seymour,Trocsanyi] hep-ph/0201036
- Generalized unitarity / reduction at integrand level and advanced tensor reduction ushered in NLO revolution ∼10 years ago
  [Bern,Dixon,Dunbar,Kosower] hep-ph/9409265 hep-ph/9708239,
  [Denner,Dittmaier] hep-ph/0509141, [Binoth,Guillet,Pilon,Heinrich,Schubert] hep-ph/0504267
  [Ossola,Papadopoulos,Pittau] hep-ph/0609007, arXiv:0802.1876, [Forde] arxiv:0704.1835
  [Ellis,Giele,Kunszt] arXiv:0708.2398 [Giele,Kunszt,Melnikov] arXiv:0801.2237 . . .
- Distributed responsibility allows for more challenging calculations
  [Binoth et al.] arXiv:1001.1307, [Alioli et al.] arXiv:1308.3462

$$\sigma_{\mathrm{NLO}} = \int d\Phi_{\mathrm{B}} \sum \left(\mathrm{B} + \tilde{\mathrm{V}} + \mathrm{I}\right) + \int d\Phi_R \sum \left(\mathrm{R} - \mathrm{S}\right)$$

- Experimenters wishlist completed in 2012 [LesHouches WG] hep-ph/0604120
- But field still developing, in particular for NLO EW, BSM and EFTs
- Resurgence of interest due to new techniques for NNLO computations

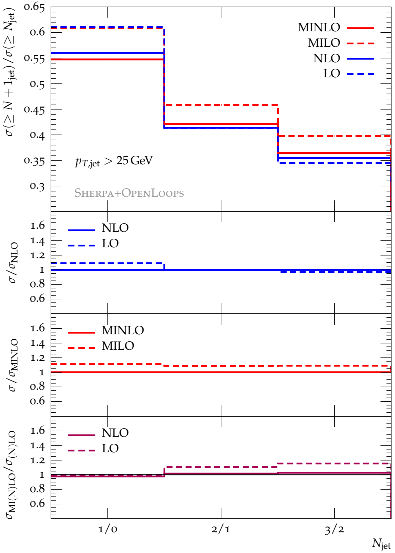
# Example: $t\bar{t}$+3jets at NLO QCD

[Maierhöfer,Moretti,Pozzorini,Siegert,SH] arXiv:1607.06934

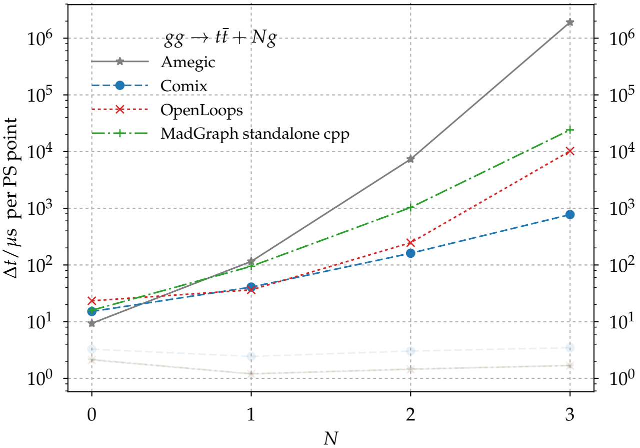
- Background to many new physics searches at LHC
- Interesting phenomenology, e.g. behavior of Berends ratio
- Challenging number of loop graphs

| Channel \ $n_{light}$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $gg \to t\bar{t}$ | 47 | 630 | 9'438 | 152'070 |
| $u\bar{u} \to t\bar{t}$ | 12 | 122 | 1'608 | 23'835 |
| $u\bar{u} \to t\bar{t}u\bar{u}$ | – | – | 506 | 6'642 |
| $u\bar{u} \to t\bar{t}d\bar{d}$ | – | – | 252 | 3'321 |

- Not the most complicated case $W^{\pm}$+5/$Z$+4 jets more involved



# Example: $t\bar{t}$+3jets at LO – The code base




- Bad computational strategy costs orders of magnitude in performance (This is effectively comparing Feynman diagrams to recursion)
- Most challenging calculations done with best technology (Comix) but still sacrificing speed for ability to maintain code base (Consider that development mostly done by graduate students)

# Parallel vs. distributed computing

MC simulations / NLO pQCD calculations typically split into

- Optimization step
  - Determine total cross section and maximum for MC simulation
  - Use adaptive MC integrators to reduce variance
  - Store results in form of weight factors / grids
- Event generation step
  - Use weight factors / grids to increase efficiency
  - Produce full events rather than cross sections only
    (Parton shower, Hadronization, Hadron decays, ...)

Natural separation into HPC and HTC domain

# Example: $t\bar{t}$+3jets on Bullet Cluster (SLAC)

Integrator optimization only

| Process Type | RAM/core [GB] | # Cores & Time |
|---|---|---|
| I gg→3g | 85/127MB | 64 × 4.5h |
| I qq→3g † | 13s | 64 × 1.1h |
| I gq→2g1q | 72/114MB | 64 × 58m |
| I gg→1g2q | 48/90MB | 64 × 31m |
| I qq→1g2q † | 43s | 64 × 3.2h |
| I gq→3q | 87/126MB | 64 × 1.3h |
| RS gg→4g | 501/574MB | 128 × 20.5h |
| RS qq→4g † | 548/614MB | 128 × 7.7h |
| RS gq→3g1q | 553/621MB | 128 × 10.7h |
| RS gg→2g2q | 248/315MB | 128 × 4.3h |
| RS qq→2g2q† | 1.9/1.9GB | 128 × 18.8h |
| RS gq→1g3q | 670/737MB | 128 × 1.08d |
| RS gg→4q | 208/274MB | 128 × 4.5h |
| RS qq→4q † | 1.8/1.8GB | 128 × 1.5d |

† memory & timing now reduced by factor 2

# To scale or not to scale

[Bauerdick et al.] arXiv:1401.6117, [Habib et al.] arXiv:1603.09303

- Complex final states require complex simulations
- High-Performance-Computers become cost-effective option, but ...
- Adaptive MC algorithms major bottleneck to scaling:
  Current limitations at $\mathcal{O}(1 - 10k)$ cores during optimization
- Scaling or no scaling in event generation, depending on sample size
  With 786k cores on Mira, each rank may produce only one event!
  $\rightarrow$ timing differences during MC unweighting do not average out


# Asynchronous MPI during optimization

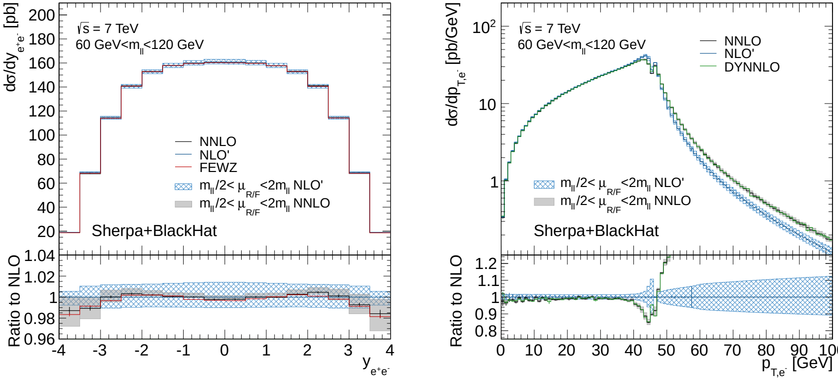
- Compute nodes sit idle during MPI communication until they receive new integrator parameters
- Possible solution: asynchronous communication
  - Computation threaded on each node, one thread handles MPI while the other integrates
  - When the MPI thread has finished an Allreduce call, it interrupts the calculator thread
  - The calculator thread switches to the next process and continues, while the MPI thread calls Allreduce on the process just computed
- Very efficient, except for the fact that one needs two threads and therefore may lose up to a factor of 2 globally in runtime. In return we do not waste time waiting for an MPI answer while we could have computed points

# Computing short distance cross sections – NNLO

- NLO calculations conveniently recycle tree-level results into Born, real-emission and infrared subtraction terms
  $\rightarrow$ ideally recycle existing NLO results into parts of NNLO
- $q_T$ cutoff method [Catani,Grazzini] hep-ph/0703012, [Gao,Li,Zhu] arXiv:1210.2808
  partitions phase-space into $q_T \approx 0$ bin and finite $q_T$ region
- Prediction for zero-$q_T$ bin from resummation
  [Becher,Neubert] arXiv:1007.4005, [Gehrmann,Lübbert,Yang] arXiv:1209.0682, arXiv:1403.6451
- Finite $q_T$ region can be taken from from automated NLO frameworks

# Drell-Yan at NNLO QCD

[Li,Prestel,SH] arXiv:1405.3607

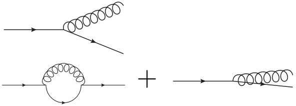



| $E_{\rm cms}$ | 7 TeV | 14 TeV | 33 TeV | 100 TeV |
|---|---|---|---|---|
| VRAP | $973.99(9)^{+4.70}_{-1.84}$ pb | $2079.0(3)\ ^{+14.7}_{-6.9}$ pb | $4909.7(8)\ ^{+45.1}_{-27.2}$ pb | $13346(3)\ ^{+129}_{-111}$ pb |
| SHERPA | $973.7(3)\ ^{+4.78}_{-2.21}$ pb | $2078.2(10)^{+15.0}_{-8.0}$ pb | $4905.9(28)^{+45.1}_{-27.9}$ pb | $13340(14)^{+152}_{-110}$ pb |

# Adding QCD evolution – Parton showers

[Marchesini,Webber] NPB238(1984)1, [Sjöstrand] PLB157(1985)321

- Parton "decay" can occur in two ways:

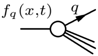


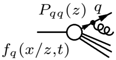
- observed
- unobserved

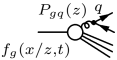
- Impose probability conservation $\Rightarrow$ observed + unobserved = 1
  Splitting governed by Poisson statistics $\rightarrow$ survival probability

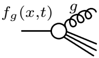
$$\Delta(t,t') := \mathcal{P}_{\rm nosplit}(t,t') = \exp\left\{-\int_t^{t'}\frac{{\rm d}\bar{t}}{\bar{t}}\,\mathcal{P}_{\rm split}(\bar{t}\,)\right\}$$

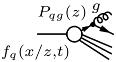
- Key to Monte-Carlo simulation of arbitrarily many emissions

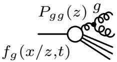
$$\frac{\rm d}{{\rm d}\log(t/\mu^2)}$$


$$= \int_x^1 \frac{{\rm d}z}{z}\frac{\alpha_s}{2\pi}$$


$$+ \int_x^1 \frac{{\rm d}z}{z}\frac{\alpha_s}{2\pi}$$



$$\frac{\rm d}{{\rm d}\log(t/\mu^2)}$$


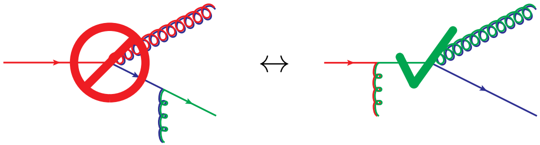
$$= \sum_{i=1}^{2\,n_f}\int_x^1 \frac{{\rm d}z}{z}\frac{\alpha_s}{2\pi}$$


$$+ \int_x^1 \frac{{\rm d}z}{z}\frac{\alpha_s}{2\pi}$$


# Color coherence and the dipole picture

[Marchesini,Webber] NPB310(1988)461

- Individual color charges inside a color dipole cannot be resolved by gluons of wavelength larger than the dipole size
  $\rightarrow$ emission off combined mother parton instead



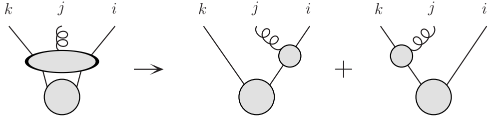
- Net effect is destructive interference outside cone with opening angle defined by emitting color dipole
  $\rightarrow$ Soft anomalous dimension halved due to reduced phase space
- Formerly implemented by angular ordering / angular veto
- Alternative description in terms of color dipoles

  [Gustafsson,Pettersson] NPB306(1988)746, [Kharraziha,Lönnblad] hep-ph/9709424
  [Winter,Krauss] arXiv:0712.3913

## The midpoint between dipole and parton showers

- Angular ordered / vetoed parton shower does not fill full phase space
  Dipole shower lacks parton interpretation $\to$ prefer alternative to both
- Can preserve parton picture by partial fractioning soft eikonal
  $\leftrightarrow$ soft enhanced part of splitting function [Catani,Seymour] hep-ph/9605323

$$\frac{p_i p_k}{(p_i p_j)(p_j p_k)} \to \frac{1}{p_i p_j}\frac{p_i p_k}{(p_i + p_k) p_j} + \frac{1}{p_k p_j}\frac{p_i p_k}{(p_i + p_k) p_j}$$



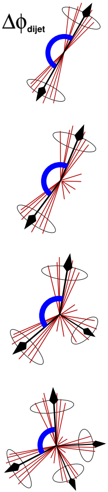

- "Spectator"-dependent kernels, singular in soft-collinear region only
  $\to$ capture dominant coherence effects (3-parton correlations)

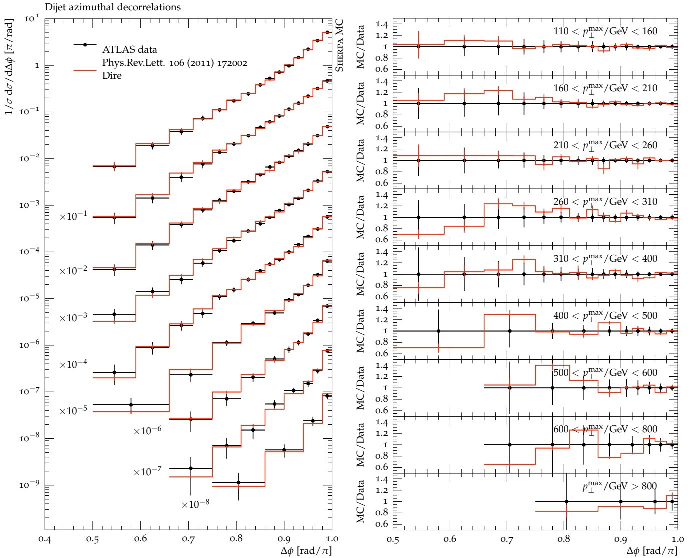
$$\frac{1}{1-z} \to \frac{1-z}{(1-z)^2 + \kappa^2} \qquad \kappa^2 = \frac{k_\perp^2}{Q^2}$$

- For correct soft evolution, ordering variable must be identical at both "dipole ends" ($\to$ recover soft eikonal at integrand level)

# Accuracy of the parton shower - Jets at the LHC

[Prestel,SH] arXiv:1506.05057



# Including higher-order effects in parton showers

- NLO DGLAP kernels known since long

  [Curci,Furmanski,Petronzio] NPB175(1980)27, PLB97(1980)437
  [Floratos,Kounnas,Lacaze] NPB192(1981)417

  - Incorrect scheme for parton showers evolving in $k_T$, but correction terms are simple phase-space factors

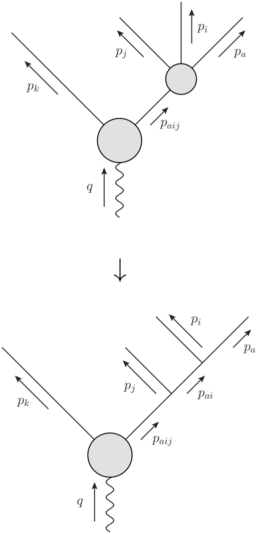
- Current focus on MC implementation technology
  - Subtract 2-loop cusp term from NLO kernels and combine with LO soft (CMW rescaling [Catani,Marchesini,Webber] NPB349(1991)635)
    Include 3-loop term in the same way [Moch,Vermaseren,Vogt] hep-ph/0403192
  - Redefine time-like Sudakovs to recover NLO DGLAP evolution

    [Jadach,Skrzypek] hep-ph/0312355
  - Mostly negative NLO corrections require weighted veto algorithm

    [Schumann,Siegert,SH] arXiv:0912.3501, [Lönnblad] arXiv:1211.7204

# $q\bar{q}$ mixing and new splitting kernels at NLO

- New topology at NLO from $q \to \bar{q}$ and $q \to q'$ splittings
- Generic $1 \to 3$ process in parton shower $2 \to 4$ process in dipole(-like) shower
- First branching treated as soft gluon radiation, second as collinear splitting (to match diagrammatic structure)
- Requires new kinematics mapping and phase-space parametrization



# Collinear factorization and resummation

- Phase space factorization can be derived as in [Dittmaier] hep-ph/9904440

$$\int \mathrm{d}\Phi(p_a, p_i, p_j \,|\, \tilde{p}_{aij}, \tilde{p}_k) =$$

$$\left[\frac{J^{(1)}_{\mathrm{FF}}}{4(2\pi)^3} \int \frac{\mathrm{d}t}{t} \int \mathrm{d}z_a \int \mathrm{d}\phi_j\right]\left[\frac{1}{4(2\pi)^3} \int \mathrm{d}s_{ai} \int \frac{\mathrm{d}x_a}{x_a} \int \mathrm{d}\phi_i \, J^{(2)}_{\mathrm{FF}}\right] 2\, p_{ai} p_j$$

- Combination with massless matrix element in collinear limit leads to

$$\int \mathrm{d}\Phi(p_a, p_i, p_j, p_k \,|\, q)\, |M_{n+2}(a, i, j, k|\, q)|^2$$

$$= \int \frac{\mathrm{d}t}{t} \int \mathrm{d}z_a \int \mathrm{d}s_{ai} \int \frac{\mathrm{d}x_a}{x_a} \int \frac{\mathrm{d}\phi_i}{2\pi} \frac{2p_{ai}p_j}{s_{aij}}$$

$$\times \left(\frac{\alpha_s}{2\pi}\right)^2 \frac{P_{(aij)a}(p_a, p_i, p_j)}{s_{aij}} \int \mathrm{d}\Phi(\tilde{p}_{aij}, \tilde{p}_k \,|\, q)\, |M_n(\widetilde{aij}, \tilde{k}|\, q)|^2$$

- Write as differential branching probability

$$\frac{\mathrm{d}\ln \Delta^{1\to 3}_{(aij)a}}{\mathrm{d}\ln t} = \int \mathrm{d}z_a\, z_a \int \mathrm{d}s_{ai} \int \frac{\mathrm{d}x_a}{x_a} \int \frac{\mathrm{d}\phi_i}{2\pi} \left(\frac{\alpha_s}{2\pi}\right)^2 \frac{P_{(aij)a}(p_a, p_i, p_j)}{s^2_{aij} /\, 2\, p_{ai} p_j}$$

- LO PS accounts for iterated collinear limit, hence we must subtract

$$\frac{\mathrm{d}\ln \Delta^{(1\to 2)^2}_{(aij)a}}{\mathrm{d}\ln t} = \int \mathrm{d}z_a\, z_a \int \frac{\mathrm{d}s_{ai}}{s_{ai}} \int \frac{\mathrm{d}\xi}{\xi} \left(\frac{\alpha_s}{2\pi}\right)^2 \frac{\sum_{(ai)} P^{(0)}_{(aij)(ai)}(\xi) P^{(0)}_{(ai)a}(z_a/\xi)}{s_{aij} /\, 2\, p_{ai} p_j}$$

# Collinear factorization and resummation

- Simplest possible configuration $q \to q'$ [Catani,Grazzini] hep-ph/9908523

$$P_{qq'} = \frac{1}{2} C_F T_R \frac{s_{aij}}{s_{ai}} \left[ -\frac{t_{ai,j}^2}{s_{ai} s_{aij}} + \frac{4\, z_j + (z_a - z_i)^2}{z_a + z_i} + (1 - 2\varepsilon) \left( z_a + z_i - \frac{s_{ai}}{s_{aij}} \right) \right]$$

  where $(z_a + z_i)\, t_{ai,j} = 2(z_a s_{ij} - z_i s_{aj}) + (z_a - z_i) s_{ai}$

- Apparent collinear singularity in $s_{ai}$ that cancels
  upon azimuthal averaging against iterated LO splitting
- But integrand locally divergent $\to$ not amenable to MC simulation
- Solved by subtraction of spin-correlated LO splitting functions
  [Somogyi,Trocsanyi,del Duca] hep-ph/0502226

$$P_{qg}^{\mu\nu} = C_F \left[ -2 \frac{z}{1-z} \frac{k_T^\mu k_T^\nu}{k_T^2} + \frac{1-z}{2} \left( -g^{\mu\nu} + \frac{p^\mu n^\nu + p^\nu n^\mu}{np} \right) \right]$$

$$P_{gq}^{\mu\nu} = T_R \left[ -g^{\mu\nu} + 4\, z(1-z)\, \frac{k_T^\mu k_T^\nu}{k_T^2} \right]$$

- Leads to additional subtraction term

$$\Delta P_{qq'} = C_F T_R \frac{4 z_a z_i z_j}{(1 - z_j)^3} \left( 1 - 2\cos^2\phi \right) \;, \qquad \cos\phi = \frac{s_{ai} s_{jk} + s_{ak} s_{ij} - s_{aj} s_{ik}}{\sqrt{4\, s_{ai} s_{ak}\, s_{ij} s_{jk}}}$$

# Collinear factorization and resummation

- Reference for $q \to q'$ upon integration over $s_{ai}, x_a, \phi_j$ given by NLO kernel

$$P_{qq'}(z) = C_F T_R \left[ (1+z)\log^2 z - \left(\frac{8}{3}z^2 + 9z + 5\right)\log z + \frac{56}{9}z^2 + 4z - 8 - \frac{20}{9z} \right]$$

- So far we have

$$P_{qq'}(z) = -C_F T_R \left[ 5(1-z) + 2(1+z)\log z \right]$$

- The difference lies in $\mathcal{O}(\varepsilon^0)$ contributions from renormalization $\times$ LO

$$\Delta P^{(\mathrm{R})}_{qq'} = \int_{z_a} \frac{\mathrm{d}\xi}{\xi} C_F \left( \frac{1+(1-\xi)^2}{\xi} \log(\xi(1-\xi)) + \xi \right) P_{gq}(z_a/\xi)$$

- ... and in integrating the iterated LO kernels over the $D$-dimensional $1 \to 3$ phase space as required for local subtraction vs. integrating over the $D$-dimensional $1 \to 2$ phase space as required for $\Delta P^{(\mathrm{R})}_{qq'}$

$$\int \frac{dx_a}{x_a} C_F T_R \left[ \frac{1+z_j^2}{1-z_j} + \left(1 - \frac{2\, z_a z_i}{(z_a+z_i)^2}\right) \left(1 - z_j + \frac{1+z_j^2}{1-z_j}\right) \left(\log(z_a z_i z_j) - 1\right) \right]$$

# Collinear factorization and resumption

[Prestel,SH] arXiv:1705.tonight

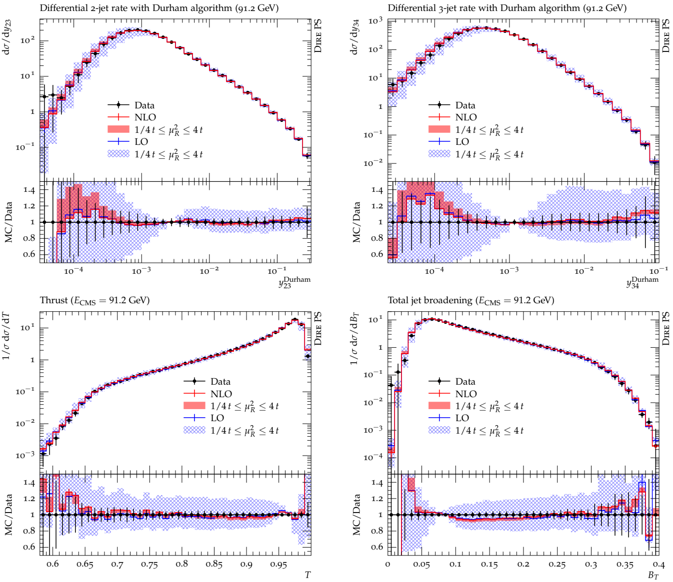
- Contributions due to $D$-dimensionality of phase space must not spoil differential radiation pattern (we live in 4D)
- Hence simulate as endpoint contributions:
  - Generated using triple collinear phase space, but retroactively projected onto $s_{ai} = 0$
  - Guarantees phase-space coverage identical to fully differential simulation
  - Remainder taken care of by parton shower unitarity
- Full calculation schematically similar to **MC@NLO** matching, but **in the exponent of the all-orders Sudakov** factor of the shower

$$P_{qq'}(z) = \left(\mathrm{I} + \frac{1}{\varepsilon}\,\mathcal{P} - \mathcal{I}\right)_{qq'}(z) + \int \mathrm{d}\Phi_{+1}(\mathrm{R}-\mathrm{S})_{qq'}(z,\Phi_{+1})$$

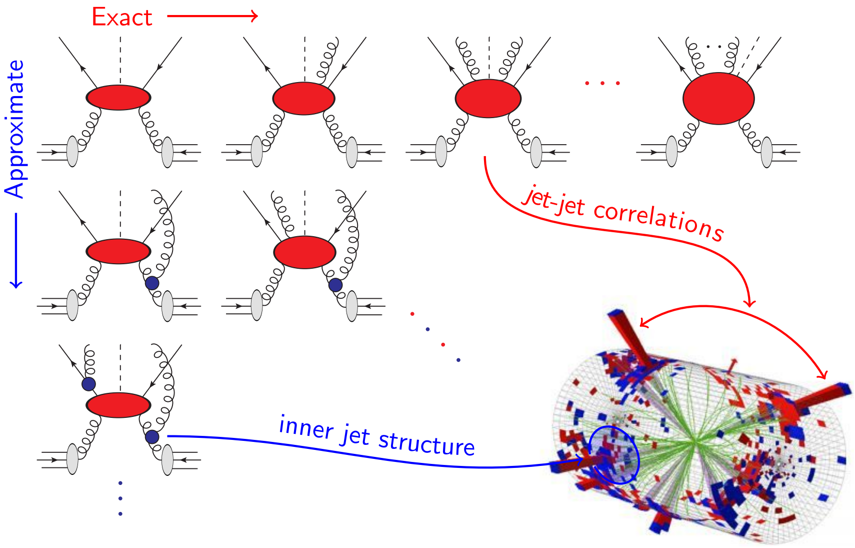
- Most importantly, we never have to compute any integral analytically
- Natural extension to kernels with Born and virtual corrections

# First phenomenological predictions

[Krauss,Prestel,SH] arXiv:1705.tonight



# Parton-shower matching & merging



# NLO matching

Two possible ways to match NLO calculations and parton showers

**Additive** (MC@NLO-like)

[Frixione,Webber] hep-ph/0204244

- Use parton-shower splitting kernel as NLO subtraction term
- Multiply LO event weight by Born-local K-factor including integrated subtraction term and virtual corrections
- Add hard remainder function consisting of subtracted real-emission correction

**Multiplicative** (POWHEG-like)

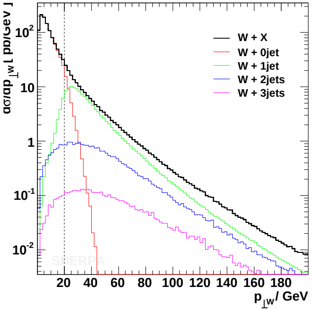
[Nason] hep-ph/0409146

- Use matrix-element corrections to replace parton-shower splitting kernel by full real-emission matrix element in first shower branching
- Multiply LO event weight by Born-local NLO K-factor (integrated over real corrections that can be mapped to Born according to parton-shower kinematics)

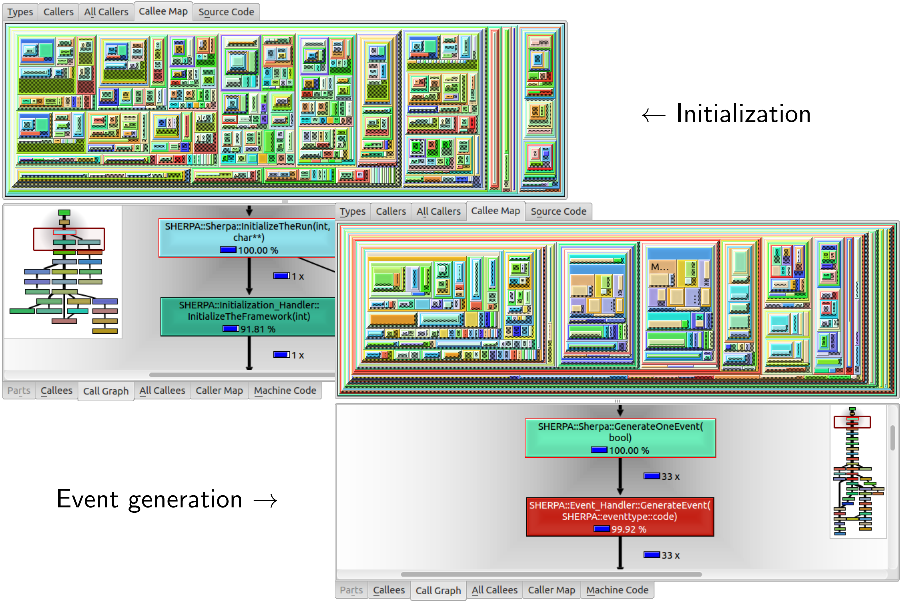
Both cases: Beware of sub-leading color corrections and spin correlations!
They have been included in [Krauss,Schönherr,Siegert,SH] arXiv:1111.1220

# ME+PS merging

- Separate phase space into "hard" and "soft" region
- Matrix elements populate hard domain
  Made exclusive by truncated vetoed parton shower Sudakov factors
- Parton shower populates soft domain
  Vetoed in hard domain to compute Sudakov factors (pseudo-shower)
- Need criterion to define "hard" & "soft"
  $\rightarrow$ jet measure $Q$ and corresponding cut, $Q_{\text{cut}}$



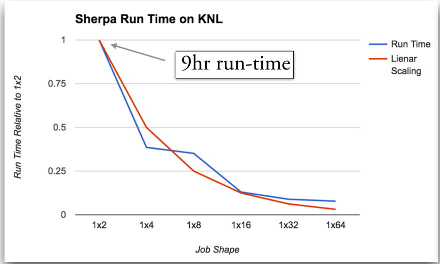
# The caveat of object oriented code

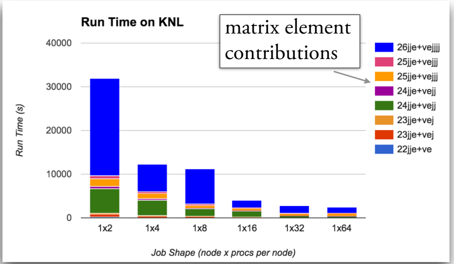


# Tomorrow's architectures - Brace for impact

[Childers,Uram,LeCompte,Benjamin,SH] CHEP 2016

- Need to run codes on tomorrow's (& today's) architectures
- 2000's paradigm: Memory is free, Flops are expensive
  Example: 16-core Xeon, 20MB L2 Cache, 64GB RAM
- 2020's paradigm: Flops are free, Memory is expensive & must be managed
  Example: 68-core Xeon KNL, 34MB L2 Cache, 16GB HBM, 96GB RAM
- Improvements in memory consumption & scalability, but
  No scaling beyond $\mathcal{O}(1k)$ cores so far (problem specific)
  Example: Sherpa NLO simulation at particle level on KNL




[T. Childers at CHEP 2016]

# Lessons from Mira - A sampler

Tried to run full-fledged Sherpa MC at particle level on MIRA (ANL)
We still don't have a fully working setup, and the reasons are

- **I/O strategy**
  - Sherpa code performed thousands of file operations at startup (hadron decay information, process information & libraries, integrator, ...) causing distress due to the large number of concurrent processes
  - 1$^{st}$ attempt: Combine info files into SQLite databases – no luck
  - 2$^{nd}$ attempt: Remove unnecessary stat calls – getting there
  - 3$^{rd}$ attempt: Clean up code – better, but still not optimal
- **Code design**
  - Automated codes are stupid, ME generators in particular
  - You can afford being stupid once, but not a million times over
  - 1$^{st}$ attempt: Redesign common subexpression elimination – better
  - 2$^{nd}$ attempt: Redesign process initialization – even better
    Example on E5-2699: $pp \to W$+7j (all) 1.4GB, 1.5d to init, 15s to start
    $pp \to W$+8j (all) 5.9GB, 14d to init, 74s to start

Classical examples of simple engineering and sacrificing speed for flexibility
Issues get worse with decreasing core-by-core performance (BG/Q, KNL!)

## Outlook

- NLO (and often NNLO) fixed-order calculations now standard
- Most simulators include NLO calculations fully automatically
- Simple NNLO calculations incorporated, more to be expected soon
- Parton showers still lack formal precision compared to analytics
  Unfortunately, they are most important for getting jet shapes right
- First steps towards NLO showers have been made, most importantly
  an algorithm to incorporate any NLO kernel in the MC now exists
- (N)NLO matching & merging methods under rapid development

## Outlook

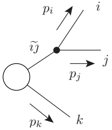
- First steps for usage of general-purpose MC on HPC made
- Good experience with fixed-order calculations, which always repeat the same, simple computational task
- Current best performance at $\sim 100 - 1000$ cores per job (different at different orders, NNLO requires $\mathcal{O}(10 - 100)$ more)
- More intricate case is full, unweighted event generation, as program flow and memory usage both unpredictable (MC!) Therefore we appreciate an increase in L2 cache much more than an increase in the number of vector units
- Unfortunately, this is not the way industry is heading, hence a truly HPC capable future MC simulation will require redesign of algorithms and code base from scratch

# Backup Slides

# The midpoint between dipole and parton showers

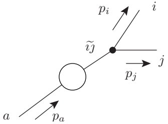
Choose parametrization such that soft term is $\frac{1-z}{(1-z)^2+\kappa^2}$ in all dipole types

(1) FF



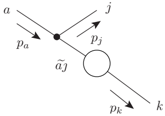

$$\kappa^2 = \frac{p_i p_j \, p_j p_k}{(p_{\widetilde{ij}} p_{\tilde{k}})^2} \qquad z_j = \frac{p_j p_k}{p_{\widetilde{ij}} p_{\tilde{k}}}$$

(2) FI



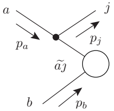

$$\kappa^2 = \frac{p_i p_j \, p_j p_a}{(p_{ij} p_a)^2} \qquad z_j = \frac{p_j p_a}{p_{ij} p_a}$$

(3) IF




$$\kappa^2 = \frac{p_a p_j \, p_j p_k}{(p_{jk} p_a)^2} \qquad z_j = \frac{p_j p_k}{p_{jk} p_a}$$

(4) II




$$\kappa^2 = \frac{p_a p_j \, p_j p_b}{(p_a p_b)^2} \qquad z_j = \frac{p_j p_b}{p_a p_b}$$

Preserve collinear anomalous dimensions & sum rules $\rightarrow$ splitting functions fixed

$$P_{qq}(z,\kappa^2) = 2\,C_F \left[ \left( \frac{1-z}{(1-z)^2+\kappa^2} \right)_+ - \frac{1+z}{2} \right] + \gamma_q \, \delta(1-z)$$

$$P_{gg}(z,\kappa^2) = 2\,C_A \left[ \left( \frac{1-z}{(1-z)^2+\kappa^2} \right)_+ + \frac{z}{z^2+\kappa^2} - 2 + z(1-z) \right] + \gamma_g \, \delta(1-z)$$

$$P_{qg}(z,\kappa^2) = 2\,C_F \left[ \frac{z}{z^2+\kappa^2} - \frac{2-z}{2} \right] \qquad P_{gq}(z,\kappa^2) = T_R \left[ z^2 + (1-z)^2 \right]$$