Simulations for the Energy Frontier

S. Chekanov
HEP/ANL

JLab Detector Working Group meeting
+ ANL

Jan 13, 2016
Future of particle collisions

- is an exploration territory. Future projects include:

High-Luminosity LHC

ILC (International Linear Collider)

FCC (Future Circular Collider). FCC-ee and FCC-hh

CEPC (Circular Electron Positron Collider)

SPPC (Super Proton-Proton Collider)

EIC (Electron Ion Collider)

In the next decade we will deal with exploration of physics reach, detector parameters and new technology options for post-LHC era

→ Requires detailed simulation of physics processes and detector responses
Monte Carlo simulation for DPF (Snowmass 2013)

- First Snowmass meeting with large-scale MC production with open data access
  - ~billion events with Delphes fast simulation
  - 140 pileup scenarios for HL-LHC
- Used OSG-grid and other resources

Lessons learned

- General community (especially theorists) is reluctant to use grid to access data
  - security certificate & approvals are too complicated?
    → Use OSG grid to make samples, but use HTTP to get them!
- Limited file storage & large EVGEN event files when using pileup
  - EVGEN files deleted, ROOT files slimmed → EVGEN is lost!
- No sufficient MC description → logfiles removed!
- No sustainable web servers for data preservation
  → Most MC files cannot be accessed any longer

Each experiment has its own resources & proprietary tools. How to share resources using project-specific infrastructure?
Technology choices for post-DPF simulations

- **Output from Monte Carlo generators (EVGEN)**
  - STDHEP, HEPMC, LHE, formats etc → new ProMC format
  - Keep NLO, logfiles etc. in a single format → everything you need for long-term archiving

- **Fast detector simulation**
  - DELPHES 3.3 as for DPF 2015. Maintained by Université catholique de Louvain
  - Easy to install
  - Available ATLAS, CMS, ILD, LHC-B and “HERA-like” detectors
  - Output: ROOT files

- **Full detector simulation:**
  - no “Project” or R&D money to develop → reuse the existing software.
  - ANL/Fermilab choice: Simulator For The Linear Collider (SLIC) developed at SLAC
    - Easy to use and configure detectors
    - Expertise in the USA
  - Output: SLCIO files

- **Analysis:** C++/ROOT, CPython/ROOT, Jython/Java
Moving forward: HepSim

Learning from DPF & building a public Monte Carlo repository

- Non-proprietary software
- Open data access
- Simple deployment on personal computers (Windows / Linux / Mac)

MC models → EVGEN

- web-optimized, compact archive format

EVGEN → SIM/RECO

- Fast detector simulation
- Full (Gean4-based) simulation with easy-to-use detector description

→ http://

- Non-proprietary software
- Open data access
- Simple deployment on personal computers (Windows / Linux / Mac)

→ http://

Open access
New data format for EVGEN

- Simple & can be deployed on Mira (unlike ROOT)
- Supported by C++, Java, etc.
- 30% smaller files than any HEP format after compression
  - Uses “Varint” for int64 instead of “fixed bytes”
  - Each byte in a varint, except the last byte. The most significant bit indicates that there are further bytes to come.

  Value: 0-10: 2 bytes  
  Value: 10-1000: 4 bytes  
  Value: 1000-10,000: 6 bytes  
  >10,000: 8 bytes

- ~20-100 times faster than XML and 3-10 times smaller
- “Archive” format → Keeps event records, original logfiles, PDG tables etc
- Separate events can be streamed as “records”  
  - similar to avi frames for web video players
- Self-describing format: C++/ Java code can be created from files

When the file size is matter

- A typical at 100 TeV event with 200 pileup MinBias events can contain up to 20,000 particles
- Low momenta → small values

Number of used bytes depends on values. Small values use small number of bytes

- Key for data reduction for high-luminosity LHC
  - effective compression of pile-up particles
- Many convertors & tools
- FORTRAN Monte Carlo generators also supported (FortranProMC by K.Strand/E.May)

https://atlaswww.hep.anl.gov/asc/promc/
## Benchmarks for EVGEN files

ProMC files are 12 times smaller than HEPMC and 30% smaller than ROOT and ~30% faster to process.

<table>
<thead>
<tr>
<th>File format</th>
<th>File Size (MB)</th>
<th>C++ (sec)</th>
<th>CPython (sec)</th>
<th>Java (sec)</th>
<th>Jython (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProMC</td>
<td>307</td>
<td>15.8</td>
<td>980</td>
<td>11.7 (12.1 +JVM startup)</td>
<td>33.3 (35 +JVM startup)</td>
</tr>
<tr>
<td>ROOT</td>
<td>423</td>
<td>20.4</td>
<td>66.7 (PyROOT)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LHEF</td>
<td>2472</td>
<td>84.7</td>
<td>30.4</td>
<td>9.0 (9.6 +JVM startup)</td>
<td>-</td>
</tr>
<tr>
<td>HEPMC</td>
<td>2740</td>
<td>175.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LHEF (gzip)</td>
<td>712</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LHEF (bzip2)</td>
<td>552</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LHEF (Izma)</td>
<td>513</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HEPMC (gzip)</td>
<td>1021</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HEPMC (bzip2)</td>
<td>837</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HEPMC (Izma)</td>
<td>802</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1. Benchmark tests for reading files with 10,000 ttbar events stored in different file formats. For each test, the memory cache on Linux was cleared. In case of C++, the benchmark program reads complete event records using appropriate libraries. CPython code for ProMC file is implemented in pure CPython and does not use C++ binding (unlike PyROOT that uses C++ libraries). In case of LHEF files, JAVA and CYPYTHON benchmarks only parse lines and tokenize the strings, without attempting to build an event record, therefore, such benchmarks may not be accurate while comparing with ProMC and ROOT.

HepSim simulation

- 2013-14: A community project to keep EVGEN files
- 2015-now: Stores fast and full simulations using “tags”
- Open data access
- Maintained at HEP-ANL
- Used for future circular collider studies (ANL/Fermilab/CERN):
  - LHC physics
  - Phase-II LHC upgrade
  - HL-LHC (pp 14 TeV 3000 fb-1)
  - FCC-hh studies (100 TeV pp, 3 ab-1)
  - HGCAL for CMS
  - Circular Electron Positron Collider (since Sep 2015)
  - EIC?
- Theorists can add their MC simulations
  - .. and analyze events the way experimentalists do!
- Can be used for outreach too

http://atlaswww.hep.anl.gov/hepsim/
HepSim simulation

http://atlaswww.hep.anl.gov/hepsim/

MC generator files (including NLO), fast simulations (ROOT), full simulations (SLCIO)
HepSim  
Repository with Monte Carlo predictions for HEP experiments

Information about "tev100_qcd_herwigpp_pt2700" dataset

- Name: tev100_qcd_herwigpp_pt2700
- Collisions: pp
- CM Energy: 100 TeV
- Entry ID: 7
- Topic: SM
- Generator: HERWIG4
- Calculation level: LO+PS+hadronisation
- Process: All dijet QCD events
- Total events: 1160000
- Number of files: 116
- Cross section (\sigma): 34.7 \pm 0.0 pb
- Luminosity (L): 33,429.3948 pb^{-1} (or) 33,429.4 fb^{-1} (or) 0.0334 ab^{-1}
- Format: ProMC
- Submission date: Fri Oct 31 14:20:17 CDT 2014
- Download URL: http://mc.hep.anl.gov/esc/hepsim/events/pp/100tev/qcd_herwigpp_full/qcd_herwigpp_pt2700
- Dataset size: 12.03 GB

Description: Inclusive QCD dijets. The log file is attached to ProMC.

Dataset files: View files

<table>
<thead>
<tr>
<th>Nr</th>
<th>Analysis code</th>
<th>Output plot (SVG)</th>
<th>Output (XML)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>qcd_herwigpp_pt2700.py</td>
<td>![SVG Image]</td>
<td>![XML Image]</td>
</tr>
</tbody>
</table>

Author: S.Crekanov

URL for download or data streaming

Format with “variable-byte” encoding based on Google's protocol buffers

40% smaller than fixed-bytes in ROOT

Shows a typical validation distribution created using Jython script.

Also supports Java, Groovy, (J)Ruby, CPython and C++

The manual explains how to download or stream events using client-side Java analysis tool
Available Monte Carlo generators

- MG5/PY6 (NLO+PS+hadr): TTbar, Higgs+jj, Higgs+TTbar etc
- MG5/Herwig (NLO+PS+hadr)
- PYHIA8 (many processes)
- FPMC (exclusive WW, Higgs)
- HERWIG++ pp collisions (QCD dijets)
- SuperChic 2 - A Monte Carlo for Central Exclusive Production
- MCFM (NLO):: Higgs -> γγ , Inclusive gamma, TTbar
- NLOjet++ (NLO) for inclusive jets (bins in pT)
- JETPHOX (NLO) for inclusive photons (bins in pT)
- PYTHIA6 for e+e and mu+mu- collisions
- LEPTO/PYTHIA for ep DIS
- LEPTO/ARIADNE for ep DIS
- Single particle guns (+ pileup)

~20% samples generated on BlueGene/Q (Mira) (Jetphox, MCFM)
~40% HEP-ANL (mainly Madgraph)
~40% OSG-CI grid (ANL/UChicago) and USATLAS CI (for phase II)
The browser unpacks “varints” into the usual numbers and show particle names using a look-up table.

Can view event records from very large files (tested with ~10 GB files)
Example: Exploring high-pT jets


Run the analysis code from the Web browser (without installing the toolkit)

To use Java Web Start, configure Java permissions:

- For Linux, run "ControlPanel", go to the "Security" tab and add "http://atlaswww.hep.anl.gov" to the exception list
- For Windows, find "Java Control Panel" and do the same.

Click here to see the editor with Python script. Click “run”

You will see updated in real time jet pT distribution as you stream over online events
HepSim dataset statistics

~ 170 data samples
(some are “compound”, i.e. consists of subsamples)

~1.4 billion events
How it works: EVGEN

**Event Generators**

- PYTHIA6
- PYTHIA8
- HERWIG++
- Madgraph5
- MCFM
- JetPhox
- FPMC
- NLOjet++
- LEPTO/Ariadne

Files created on HEP servers, Mira, OSG-grid (CI connect) U Chicago / ANL comp. Institute Or users

- **Unified ProMC format**
- varint encoding
- C++, Java, Python
- Web streaming
- Can be installed on BG/Q

EVGEN files stored on several public web servers (Apache)

HepSim:
- index files
- create metadata
- prepare for batch download
- validate with Jython scripts
- create search database
Simulation of detector response

- **EVGEN**
  - fast
  - full

- **Detector description (XML)**

- **Delphes fast simulation**
  - (HL-LHC, FCC, ILC, EIC..)

- **SLIC full simulation software**

- **Gean4 simulation**

- **Reconstruction**
  - (org.lcsim)

- **Particle flow algorithm**
  - (Pandora)

- **ROOT files organized in directories for each detector geometry tag**

- **HepSim**
  - index reco files
  - create metadata
  - prepare for batch download
  - validation scripts
  - search database
  ...

- **SLCIO file format**
  - (C++/Java/Python)
Full G4 simulation & analysis

Developed at SLAC (T.Johnson, N.Graph, J.McCormick) for the SiD detector (ILC)
Included to ilcsoft (J.Srube, PNNL)
Includes analysis tools (Jas3, Wired4)

"Provides a complete and flexible detector simulation package capable of simulating arbitrarily complex detectors with runtime detector description".
Example: Software for future circular colliders

Adopted for future collider studies at ANL/Fermilab (S.C., A.Kotwal, J.Strube):
- Integrated with HepSim. Output files are publicly accessible
- Supported by HEP ANL and deployed on OSG-grid
- User analysis package:
  - **FPaDsoft** - software for “Future Particle Detector” studies
  - Uses Python on the Java platform (C++ can be used too)
  - Does not require “installation”. Runs on Linux/Mac/Windows

Data analysis software (FPadSoft)

User analysis:


Note:
Jas3 with Wired4 can be used too!
SiD detector

- A multi-purpose detector
- Optimized for separate particles (unlike LHC)
- The key characteristics of the SiD detector:
  - 5 Tesla solenoid
  - Silicon tracker:
    - 50 um readout readout pitch
    - 25 um distance between pitches
  - 3.5 mm cell size for ECAL
  - W absorber with silicon readout)
  - 10x10 mm cell size for HCAL:
    - Steal (absorber with RPC)
    - 40 layers for barrel (HCAL)
    - 45 layers in forward (FCAL)
- Optimized for particle-flow algorithms.
SiD detector

<table>
<thead>
<tr>
<th>Barrel</th>
<th>Technology</th>
<th>Inner radius</th>
<th>Outer radius</th>
<th>z extent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertex detector</td>
<td>Silicon pixels</td>
<td>1.4</td>
<td>6.0</td>
<td>+/- 6.25</td>
</tr>
<tr>
<td>Tracker</td>
<td>Silicon strips</td>
<td>21.7</td>
<td>122.1</td>
<td>+/- 152.2</td>
</tr>
<tr>
<td>ECAL</td>
<td>Silicon pixels-W</td>
<td>126.5</td>
<td>140.9</td>
<td>+/- 176.5</td>
</tr>
<tr>
<td>HCAL</td>
<td>RPC-steel</td>
<td>141.7</td>
<td>249.3</td>
<td>+/- 301.8</td>
</tr>
<tr>
<td>Solenoid</td>
<td>5 Tesla SC</td>
<td>259.1</td>
<td>339.2</td>
<td>+/- 298.3</td>
</tr>
<tr>
<td>Flux return</td>
<td>Scintillator-steel</td>
<td>340.2</td>
<td>604.2</td>
<td>+/- 303.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Endcap</th>
<th>Technology</th>
<th>Inner z</th>
<th>Outer z</th>
<th>Outer radius</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertex detector</td>
<td>Silicon pixels</td>
<td>7.3</td>
<td>83.4</td>
<td>16.6</td>
</tr>
<tr>
<td>Tracker</td>
<td>Silicon strips</td>
<td>77.0</td>
<td>164.3</td>
<td>125.5</td>
</tr>
<tr>
<td>ECAL</td>
<td>Silicon pixel-W</td>
<td>165.7</td>
<td>180.0</td>
<td>125.0</td>
</tr>
<tr>
<td>HCAL</td>
<td>RPC-steel</td>
<td>180.5</td>
<td>302.8</td>
<td>140.2</td>
</tr>
<tr>
<td>Flux return</td>
<td>Scintillator/steel</td>
<td>303.3</td>
<td>567.3</td>
<td>604.2</td>
</tr>
<tr>
<td>LumiCal</td>
<td>Silicon-W</td>
<td>155.7</td>
<td>170.0</td>
<td>20.0</td>
</tr>
<tr>
<td>BeamCal</td>
<td>Semiconductor-W</td>
<td>277.5</td>
<td>300.7</td>
<td>13.5</td>
</tr>
</tbody>
</table>

All of this can be changed using XML configuration files.
Designing future detectors

Focus on physics & performance studies of certain detector aspects and physics processes, rather than creating “complete” detectors

Studies of CAL transverse and longitudinal granularity, depth, material, magnetic fields, pixel sizes etc, responses to particles etc.

simplifying for CPEC (e+e- at 250 GeV)

expanding to FCC-hh (pp at 100 TeV)
Programming languages

- **EVGEN: ProMC format → C++ (or) Java. Support for Fortran**

- **Delphes fast simulation → C++/ROOT**

- **SLIC software:**
  - Geant4 simulation → C++/C
  - Reconstruction → Java
  - Pandora particle flow algorithm → C++

- **Analysis: C++/ROOT or Jython/Java (Python on the Java platform)**
  - No manpower to maintain platform specific libs → minimize the usage of C++
  - Currently, many studies are done using Python on the Java platform
    - can read PROMC and SLCIO files
    - easy to deploy, no LINUX specific libraries
    - runs on Windows/Mac

Java

- Most popular object-oriented programming language
- TIOBE Index for January 2016:
  
  http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

<table>
<thead>
<tr>
<th>Jan 2016</th>
<th>Jan 2015</th>
<th>Change</th>
<th>Programming Language</th>
<th>Ratings</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>▲</td>
<td>Java</td>
<td>21.465%</td>
<td>+5.94%</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>▼</td>
<td>C</td>
<td>16.036%</td>
<td>-0.67%</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>▲</td>
<td>C++</td>
<td>6.914%</td>
<td>+0.21%</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>▲</td>
<td>C#</td>
<td>4.707%</td>
<td>-0.34%</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>▲</td>
<td>Python</td>
<td>3.854%</td>
<td>+1.24%</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
<td>PHP</td>
<td>2.706%</td>
<td>-1.08%</td>
</tr>
</tbody>
</table>

- Students who want to find a job outside HEP should learn Java
- Remember, no Project or R&D money for software support at this stage
- Cannot spend research time on compilation / chasing constantly changing Linux libraries
MC simulations for the HEP community

Phys. Rev. D 91 (2015) 034014

Usage:
- Snowmass papers for HL-LHC
- ATLAS run I & II analyses: excl. H\(^0\), excl. WW, direct photons with MCFM NLO, JETPHOX NLO, Long-lived particles, ADD model for gravitons, H \(\rightarrow \phi \gamma\) → validated and shipped to ATLAS
- FCC physics studies, CPEC (recently)

One of the world's largest public MC simulation for 100 TeV:
- \(~40,000\) CPU*h to create
- \(~2\) days for download & analyse
Contributions

- E. May - ProMC format development, benchmarks on BlueGene/Q (ANL)
- K. Strand (SULI 2014) - ProMC conversion tools
- P. Van Gemmeren - testing ProMC format
- T. Sjöstrand - ProMC integration with Pythia8
- P. Demin - ProMC integration with Delphes
- I. Pogrebnyak - (U.Michigan) software validation toolkit, fastjet in Java
- D. Wilbern (SULI 2015) - Pileup mixing tool based on ProMC
- M. Selvaggi - Delphes card for ILD geometry and “EIC”-like (requested by S.C.)
- H. Gray - Delphes card for FCC-hh geometry
- J. Strube (PNNL) - LCIO/SLIC for full simulation
- A. Kotwal (Duke Univ.) - LCIO/SLIC for full simulation
- J. Adelman (NIU) – H+tt sample + post-Snowmass Delphes 3.3 card for 13/14 TeV
- S. Padhi - prototyping Snowmass Delphes 3.1 during Snowmass 2013
- K. Pedersen - alternative b-tagging for rfast003 in HepSim
- Shin-Shan Yu - Heavy Higgs MG5 simulations for HepSim

A lot of help / advise from J.McCormick and N.Graph (SLAC)
How to contribute

- Generate EVGEN files and point to their location
  - Fill ProMC correctly, i.e. append complete logfile, cross sections etc.
  - Validate using the HEPSIM tools (if can)
- Contribute to the software tools
- Run Apache server and maintain your own EVGEN & full simulation files

Support (limited, on a voluntary basis): (contact hepsim@anl.gov)

- HEPSIM integration, deployment, OSG-grid, EVGEN MC, fast sim etc.
  - ANL: S.C.
- Some support for changes inside SLIC soft (used for ILC)
  - SLAC: N.Graf & J.McCormick
  - PNNL: J.Strube
- Configure detectors, physics, analysis package for circular colliders
  - ANL/Fermilab: S.C., A.Kotwal
Test samples for EIC

- DIS. e-p collisions. 141 GeV: LEPTO+PYTHIA, LEPTO+ARIADNE

Selected: e⁻ p collisions, 141 GeV CM energy. p(250 GeV), e(20 GeV), all type

![Table showing test samples for EIC](image)

ep. 141 CM energy. Several Q2 regions
- Includes fast and full simulations for testing:
  - SiD detector full simulations (rfull001 tag)
  - ILD fast simulation (rfast001 tag)

- Moving Monte Carlo code used for HERA (ARIADNE, LEPTO, AROMA, CASCADE, RAPGAP, PHOJET, PEPSI) from the RunMC project http://runmc.hepforge.org/
Analyzing EVGEN & full simulation files for e+p (DIS)

- Then click “rfull001”. Find *.slcio files.
- Download samples as:

```bash
source hs-toolkit/setup.sh
hs-get gev141ep_lepto6_dis100q2
hs-get gev141ep_lepto6_dis100q2%rfull001
```

- EVGEN files can be analyzed as explained in [http://atlaswww.hep.anl.gov/hepsim/description.php](http://atlaswww.hep.anl.gov/hepsim/description.php)
- Analysis of full simulation files (reconstructed particles & calorimeter hits, tracks) can be done as explained here for e+e-:
  - pay attention to “electron container” (DIS!)

Good Luck!