Track Highlights

Paul LAYCOCK (BNL), NAKAHAMA Yu (Nagoya), Steven SCHRAMM (UniGe)
KHOO Teng Jian
& T1,2,9 Track Conveners
Number \( \frac{X}{e} \) ascale
Online SW

Melanie Hava: platypus swimming with the fishes

Offline SW
1. Optimisation / acceleration

2. Collaborative and common software
What has changed since CHEP’18?

FogKV became DAQDB and is available @github: github.com/daq-db/daqdb.

Intel Optane DC Persistent Memory is now available.

DAQDB has been integrated into ATLAS TDAQ framework.

Distributed mode has been enabled.

DAQDB is in the performance range required by ATLAS/CMS for local access, but requires optimization in remote mode.
Quantum Tracking

Quantum Computer

Universal quantum circuit

\[ |0\rangle \rightarrow H \rightarrow u_1(\theta_1) \rightarrow U_\theta \rightarrow U_\theta^+ \rightarrow U_{\theta'} \]

- Arrange gates for each problem
- General-purpose computer

D-Wave

Quantum annealer

- Find the minimum energy state of a given Hamiltonian
- Suitable for an optimization problem

Focus on quantum annealer

Results

1600 particles (20\% of HL-LHC) - 11000 hits

Input  \rightarrow Doublet selection  \rightarrow Annealing

- Reconstructed high \( pT \) tracks
- Reconstructed low \( pT \) tracks
- Not reconstructed tracks
- Fake tracks

390000 Doublets
Purity 0.22 \%
Efficiency 99.5 \%

2445 Doublets

1424 Doublets
Purity 98.5 \%
Efficiency 96.4 \%

Not just a pipe dream!?
potential doublets

- filter doublets
- create triplets
- create qplets

build QUBO

solve

kept triplets

- doublets
- tracks

precision
recall
trackml score

final doublets forming track candidates

Quantum annealing

preprocessing / model building

sampling

processing

scoring
Top Tagging with ResNet-50

- ResNet-50: image recognition network
  - 25M parameters, 8B operations
  - Available on coprocessors
  - vs. CMS DeepAK8 network: 500K parameters, 50M operations
- Retrain on publicly available top quark tagging dataset
  - New set of weights, optimized for physics
  - Add custom classifier layers to interpret features from ResNet-50
  - ResNet-50 model that runs on FPGAs is “quantized”
    - Tune weights to achieve similar performance
- State-of-the-art results vs. other leading algorithms
  - Demonstrates viability of large industry networks for physics

Performance Comparisons

<table>
<thead>
<tr>
<th>Type</th>
<th>Note</th>
<th>Latency [ms]</th>
<th>Throughput [img/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU*</td>
<td>Xeon 2.6 GHz</td>
<td>1750</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>i7 3.6 GHz</td>
<td>500</td>
<td>2</td>
</tr>
<tr>
<td>GPU†**</td>
<td>batch = 1</td>
<td>7</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>batch = 32</td>
<td>1.5</td>
<td>667</td>
</tr>
<tr>
<td>Brainwave</td>
<td>remote</td>
<td>60</td>
<td>660</td>
</tr>
<tr>
<td></td>
<td>on-prem</td>
<td>10 (1.8 on FPGA)</td>
<td>660</td>
</tr>
</tbody>
</table>

*Performance depends on clock speed, TensorFlow version, # threads (1)
†Directly connected to CPU via PCIe – not a service
**Performance depends on batch size & optimization of ResNet-50

- SONIC w/ Brainwave achieves:
  - 175× (30×) on-prem (remote) improvement in latency vs. CMSSW CPU!
  - Competitive throughput vs. GPU, w/ single-image batch as a service!
## Tracking ML challenge

### Leaderboard

<table>
<thead>
<tr>
<th>#</th>
<th>User</th>
<th>Entries</th>
<th>Date of Last Entry</th>
<th>score ▲</th>
<th>accuracy_mean ▲</th>
<th>accuracy_std ▲</th>
<th>computation time (sec) ▲</th>
<th>computation speed (sec/event) ▲</th>
<th>Duration ▲</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sgorbuno</td>
<td>9</td>
<td>03/12/19</td>
<td>1.1727</td>
<td>0.944 (2)</td>
<td>0.00 (14)</td>
<td>28.06 (1)</td>
<td>0.56 (1)</td>
<td>64.00 (1)</td>
</tr>
<tr>
<td>2</td>
<td>fastrack</td>
<td>53</td>
<td>03/12/19</td>
<td>1.1145</td>
<td>0.944 (1)</td>
<td>0.00 (15)</td>
<td>55.51 (16)</td>
<td>1.11 (16)</td>
<td>91.00 (6)</td>
</tr>
<tr>
<td>3</td>
<td>cloudkitchen</td>
<td>73</td>
<td>03/12/19</td>
<td>0.9007</td>
<td>0.928 (3)</td>
<td>0.00 (13)</td>
<td>364.00 (18)</td>
<td>7.28 (18)</td>
<td>407.00 (8)</td>
</tr>
</tbody>
</table>

**HEP people**

**PH+CS**
Accuracy phase

- **First**: Top Quarks
  - Johan Sokrates is an industrial Mathematics master student
  - Pair seeding, triplet extension, trajectory following, track cleaning, all with machine learning for quality selection

- **Second**: Outrunner
  - Pei-Lien Chou is a software engineer in image-based deep learning in Taiwan
  - Machine learning to predict the adjacency matrix

- **Third**: Sergey Gorbunov
  - Sergey Gorbunov is a physicist, expert in tracking
  - Iterative steps, triplet seeding, trajectory following

Throughput phase

- **First**: Sgorbuno
  - Sergey Gorbunov is a physicist, expert in tracking
  - 3rd position at the accuracy phase
  - Iterative steps, triplet seeding, trajectory following

- **Second**: fastrack
  - Dmitry Emelyanov is a physicist
  - Graph representation of neighbors, cellular automaton, track following

- **Third**: cloudkitchen
  - Marcel Kunze is a former physicist.
  - Direct acyclic graph of voxels, pair and triplet classification + Top Quark solution of accuracy phase (trajectory following, track cleaning, all with machine learning for quality selection)

Incidentally, best solutions are also best accuracy and best timing. Software will be submitted and analyzed in the coming weeks.

Still a niche for old-fashioned algs where we have decades of experience
Turnkey Software Stack
— FCC pioneering

Framework migration need not be a massive barrier

Marlin & Gaudi

Apart from some naming conventions, very similar ideas in the two frameworks:

<table>
<thead>
<tr>
<th></th>
<th>Marlin</th>
<th>Gaudi</th>
</tr>
</thead>
<tbody>
<tr>
<td>language</td>
<td>c++</td>
<td>c++</td>
</tr>
<tr>
<td>working unit</td>
<td>Processor</td>
<td>Algorithm</td>
</tr>
<tr>
<td>configuration language</td>
<td>XML</td>
<td>Python</td>
</tr>
<tr>
<td>set up function</td>
<td>init</td>
<td>initialize</td>
</tr>
<tr>
<td>working function</td>
<td>processEvent</td>
<td>execute</td>
</tr>
<tr>
<td>wrap up function</td>
<td>end</td>
<td>finalize</td>
</tr>
<tr>
<td>Transient data format</td>
<td>Lcio</td>
<td>anything</td>
</tr>
</tbody>
</table>

- To start using Gaudi: use a generic wrapper around the processors.
- Prototype: [https://github.com/andresaider/GMP](https://github.com/andresaider/GMP)
- Read Lcio files and pass the `Lcio::Event` to our processors

_Surprisingly_ little changes needed in Marlin

- Make `marlin::Processor::setParameters` and `marlin::Processor::setName` public
- Actually make the Marlin `EventSelector` part of the namespace to avoid clash with `EventSelector` from Gaudi
- Make it possible to call the `marlin::ProcessorEventSeeder` from the wrapper; move to functions from private to public
ACTS

Gaudi

TrackML

Turnkey stack
Thanks for a great conference!