HPS Software

Presentation for DOE Review at SLAC

January 18, 2019

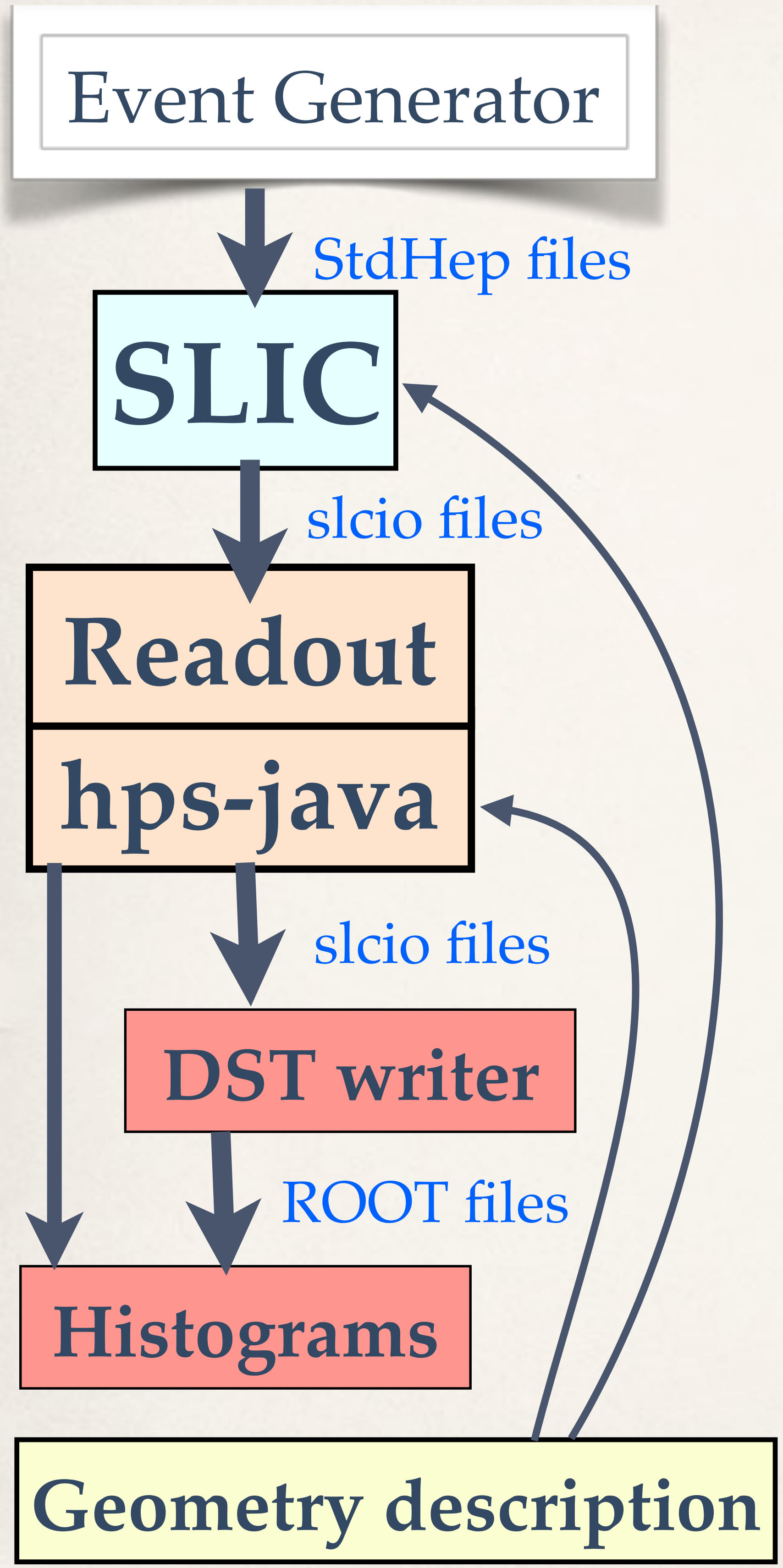
Overview

- Introduction
    - History
    - System Overview
    - Software Organization Overview
    - Software Group
- Outstanding Task List
- Historic and projected manpower
- System resource utilization
- Conclusions

Introduction - history

- Early decision by collaboration to leverage the existing expertise in the SLAC group with the Linear Collider Simulation, LCSim software framework.
  - JLab (CLAS12) software was too immature, and would not suffice for expected 6-GeV era run.
  - Not enough time and manpower to start from scratch.
- Result:
  - Development of “hps-java” code, which utilizes the “lcsim” framework.
    - +/- Main code development is in Java.
    - + Robust framework to develop on.
    - + Existing tracking component: seed tracker.
    - - No overlap with JLab code.
  - Main data storage model: LCIO.
    - + Read/write capabilities from Java and C++.
    - - Less flexibility in contents.

Introduction - System Overview: MC



A′ events,
Background events

SLIC or hps-sim : Main GEANT4 based simulation.

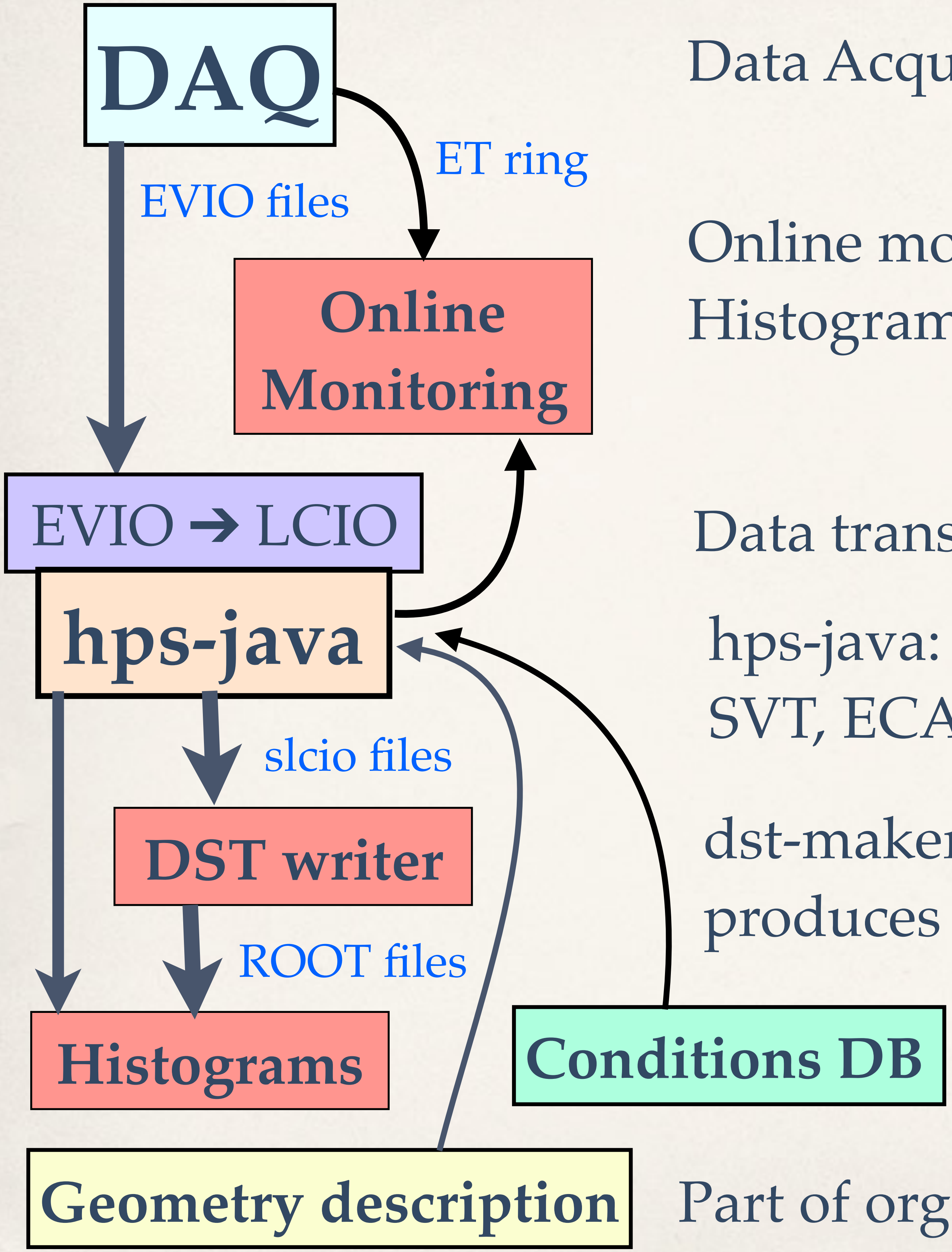
Readout: Simulates electronics and trigger.

hps-java: Analysis framework:
SVT, ECAL, Tracking, Hodoscope

dst-maker or hpstr - reads slcio files and produces data summary files.

Part of org.lcsim that computes geometries.

Introduction - System Overview: Data



Data Acquisition System

Online monitoring system
Histograms and Event display.

Data translation layer.

hps-java: Analysis framework:
SVT, ECAL, Tracking, Hodoscope

dst-maker or hpstr - reads slcio files and produces data summary files.

Stores run by run information.

Part of org.lcsim that computes geometries.

Introduction - System Overview

- Calibrations:
  - SVT online calibration code - timing in, pedestals, gains.
    - Existing code that runs during commissioning to time in and check SVT.
  - ECal calibration - Cosmic ray calibration, Full Energy Electron calibration.
    - Existing code to calibrate ECal, pedestals and gains.
  - Hodoscope calibration
    - Code needs to be written, but can borrow from ECal code.
  - Detector Alignment - Millipede II
    - Complicated procedure for getting a good alignment.
    - High on Tracking Group priority list to simplify and improve this procedure.
- Physics Analysis Code:
  - Runs after data reconstruction.
  - Was in the domain of individual analyzers, but is now becoming more centralized.
  - See presentation by Nathan Baltzell.

Introduction - Software Organization

- Code repository - GitHub
  - Tracks code, allows development on branches
  - Merging only through “pull requests”, which must be approved.
- Issue Tracking - GitHub
  - Couples code issues with branches.
- Code Documentation - Confluence Wiki + Java Doc
- Build System - Maven
- Testing - Maven integration tests.
- Continuous integration testing - Jenkins / Hudson
- Code profiles - JProfiler
- Releases: - Github + Maven + Nexus.
  - Release is tagged on GitHub.
  - Resulting JAR file is available for download from Nexus.

Introduction - Software Group

- Bi- Weekly meetings with online presentations.
- Software group mailing list
- SLACK for more immediate communication.

- Lead: Maurik Holtrop
    - Tracking lead: Norman Graf
    - MC Generators: Takashi Maruyama
        - MC data production: Bradley Yale
    - Trigger: Valeri Kubarovsky
        - Trigger code: Kyle McCarty
    - Data Processing: Rafayel Paremuzyan
    - Analysis software: Matt Graham, Nathan Baltzell
    - Specific codes:
        - DST code: Omar Moreno
        - MC Simulation code, conditions system: Jeremy McCormick

Monte Carlo Generators

Monte Carlo Detector Model

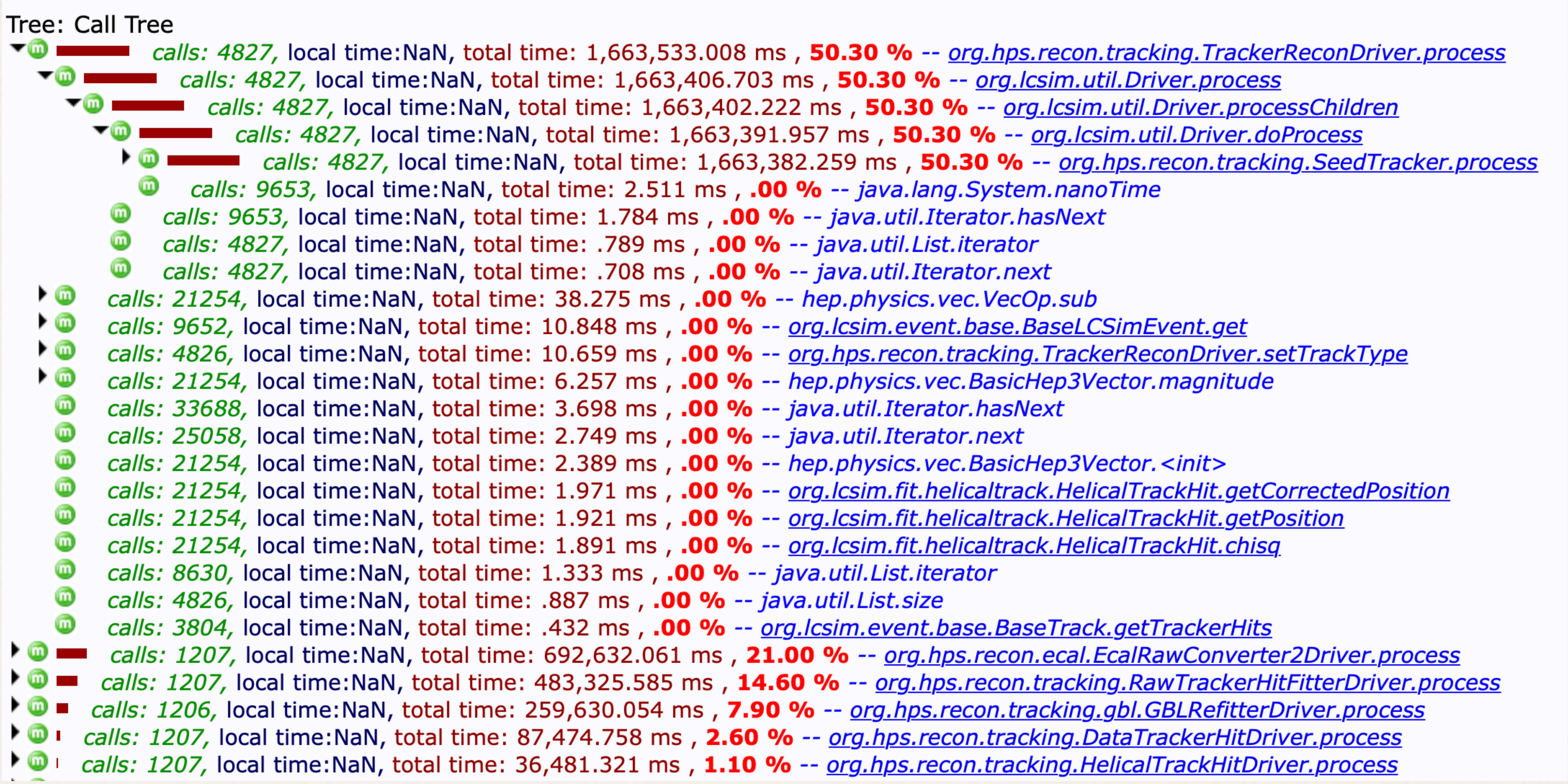
Tracking

Alignment

Monitoring

Resource use, CPU

- CPU requirements for main data processing step.
  - 2017 - May - 165 ms/event/core
  - 2018 - Nov - 53 ms/event/core
  - Profile: http://nuclear.unh.edu/HPS/Profiles/Call_Tree_doProcess_2018_12_11.xml



```
Tree: Call Tree
calls: 4827, local time:NaN, total time: 1,663,533.008 ms , 50.30 % -- org.hps.recon.tracking.TrackerReconDriver.process
  calls: 4827, local time:NaN, total time: 1,663,406.703 ms , 50.30 % -- org.lcsim.util.Driver.process
    calls: 4827, local time:NaN, total time: 1,663,402.222 ms , 50.30 % -- org.lcsim.util.Driver.processChildren
      calls: 4827, local time:NaN, total time: 1,663,391.957 ms , 50.30 % -- org.lcsim.util.Driver.doProcess
        calls: 4827, local time:NaN, total time: 1,663,382.259 ms , 50.30 % -- org.hps.recon.tracking.SeedTracker.process
        calls: 9653, local time:NaN, total time: 2.511 ms , .00 % -- java.lang.System.nanoTime
      calls: 9653, local time:NaN, total time: 1.784 ms , .00 % -- java.util.Iterator.hasNext
      calls: 4827, local time:NaN, total time: .789 ms , .00 % -- java.util.List.iterator
      calls: 4827, local time:NaN, total time: .708 ms , .00 % -- java.util.Iterator.next
  calls: 21254, local time:NaN, total time: 38.275 ms , .00 % -- hep.physics.vec.VecOp.sub
  calls: 9652, local time:NaN, total time: 10.848 ms , .00 % -- org.lcsim.event.base.BaseLCSimEvent.get
  calls: 4826, local time:NaN, total time: 10.659 ms , .00 % -- org.hps.recon.tracking.TrackerReconDriver.setTrackType
  calls: 21254, local time:NaN, total time: 6.257 ms , .00 % -- hep.physics.vec.BasicHep3Vector.magnitude
  calls: 33688, local time:NaN, total time: 3.698 ms , .00 % -- java.util.Iterator.hasNext
  calls: 25058, local time:NaN, total time: 2.749 ms , .00 % -- java.util.Iterator.next
  calls: 21254, local time:NaN, total time: 2.389 ms , .00 % -- hep.physics.vec.BasicHep3Vector.<init>
  calls: 21254, local time:NaN, total time: 1.971 ms , .00 % -- org.lcsim.fit.helicaltrack.HelicalTrackHit.getCorrectedPosition
  calls: 21254, local time:NaN, total time: 1.921 ms , .00 % -- org.lcsim.fit.helicaltrack.HelicalTrackHit.getPosition
  calls: 21254, local time:NaN, total time: 1.891 ms , .00 % -- org.lcsim.fit.helicaltrack.HelicalTrackHit.chisq
  calls: 8630, local time:NaN, total time: 1.333 ms , .00 % -- java.util.List.iterator
  calls: 4826, local time:NaN, total time: .887 ms , .00 % -- java.util.List.size
  calls: 3804, local time:NaN, total time: .432 ms , .00 % -- org.lcsim.event.base.BaseTrack.getTrackerHits
calls: 1207, local time:NaN, total time: 692,632.061 ms , 21.00 % -- org.hps.recon.ecal.EcalRawConverter2Driver.process
calls: 1207, local time:NaN, total time: 483,325.585 ms , 14.60 % -- org.hps.recon.tracking.RawTrackerHitFitterDriver.process
calls: 1206, local time:NaN, total time: 259,630.054 ms , 7.90 % -- org.hps.recon.tracking.gbl.GBLRefitterDriver.process
calls: 1207, local time:NaN, total time: 87,474.758 ms , 2.60 % -- org.hps.recon.tracking.DataTrackerHitDriver.process
calls: 1207, local time:NaN, total time: 36,481.321 ms , 1.10 % -- org.hps.recon.tracking.HelicalTrackHitDriver.process
```

Resource use, CPU

- CPU requirement for MC production.

Resource use, disk

- Estimated disk space usage.

Software Task List

Mostly, our software is in reasonably good shape, but many improvement are desirable: directly related to 2019 running, smoothing operations, speeding up processing.

- **Critical Tasks for 2019 run:**
  - Complete Hodoscope simulation and new trigger optimization analysis.
    - Extensive task which is already well underway. See Rafayel Paremuzyan's talk.
  - Add FADC bit-packed data decoder to hps-java.
    - Already exists for CLAS12, so not expected to be too complicated.
  - Update monitoring histograms.
    - Needs hodoscope and L0 histograms added.
    - Cleaning up existing histograms.
  - Improve/update data quality monitoring.
    - Update for hodoscope and L0.

Software Task List

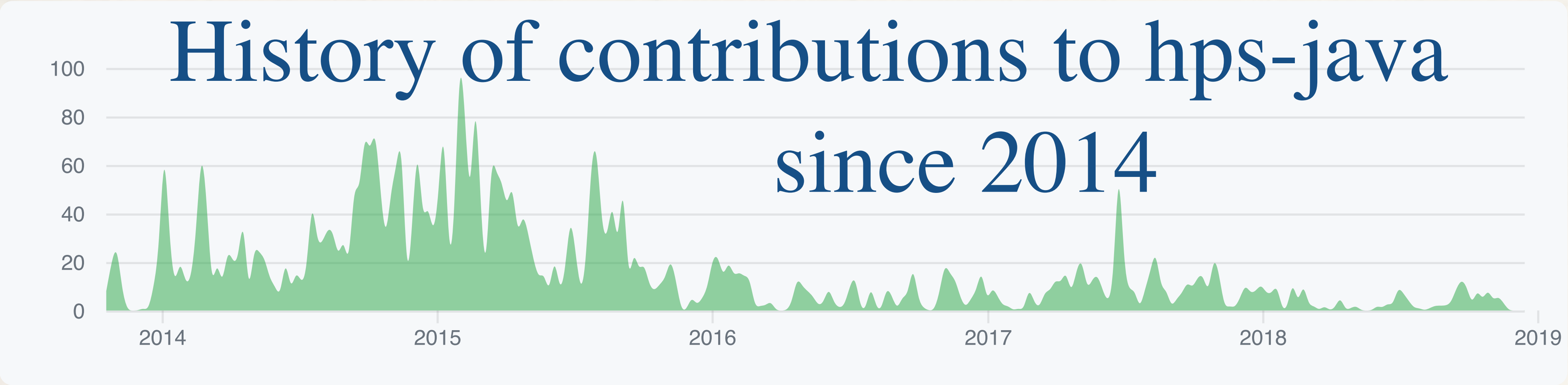
- **Important Tasks, highly desirable:**
  - Improve the alignment procedures.
    - We need to get detector alignment to be easier so results can be obtained more quickly.

Software Task List

- **Other Important Tasks:**
  - Improve the alignment procedures.
    - We need to get detector alignment to be easier so results can be obtained more quickly.
  - Revisit all other calibrations and see where updates are needed.
    - It has been a little while since we last needed a full calibration.
  - Improve processing speed of the code.
    - Further improve the speed of the tracking code.
    - Complete the investigation of alternate tracking: Kalman filter and different seed finder.
    - Possibly: preprocess the FADC and SVT pulse fits.
  - Learn to use the Open Science Grid for simulation.

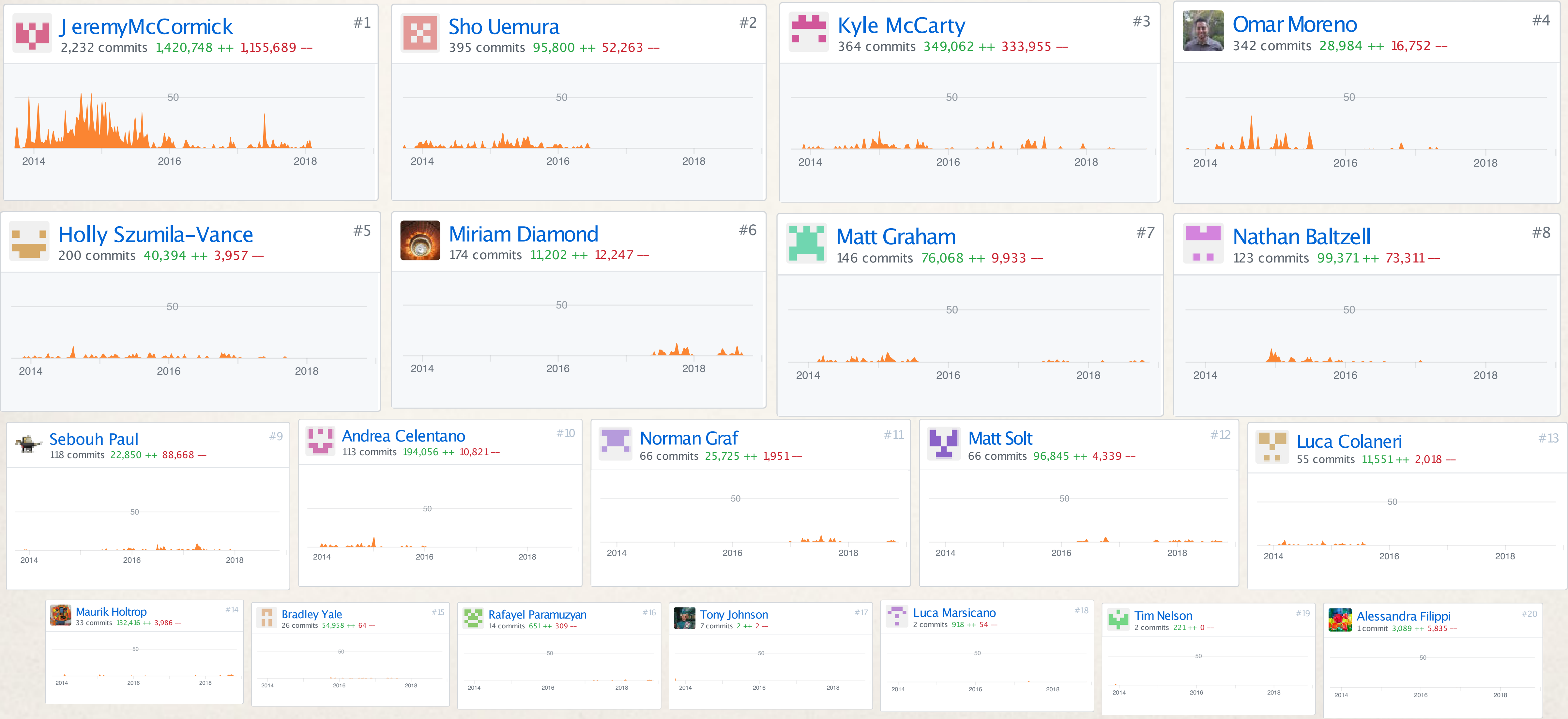
  - Lots of minor issues, maintenance issue, code improvements on issues lists.

Software contributions



History of contributions to hps-java since 2014

Caveat:
Number of lines
or number of commits
does not linearly
correspond to software
productivity



J eremyMcCormick #1
2,232 commits 1,420,748 ++ 1,155,689 --

Sho Uemura #2
395 commits 95,800 ++ 52,263 --

Kyle McCarty #3
364 commits 349,062 ++ 333,955 --

Omar Moreno #4
342 commits 28,984 ++ 16,752 --

Holly Szumila-Vance #5
200 commits 40,394 ++ 3,957 --

Miriam Diamond #6
174 commits 11,202 ++ 12,247 --

Matt Graham #7
146 commits 76,068 ++ 9,933 --

Nathan Baltzell #8
123 commits 99,371 ++ 73,311 --

Sebouh Paul #9
118 commits 22,850 ++ 88,668 --

Andrea Celentano #10
113 commits 194,056 ++ 10,821 --

Norman Graf #11
66 commits 25,725 ++ 1,951 --

Matt Solt #12
66 commits 96,845 ++ 4,339 --

Luca Colaneri #13
55 commits 11,551 ++ 2,018 --

Maurik Holtrop #14
33 commits 132,416 ++ 3,986 --

Bradley Yale #15
26 commits 54,958 ++ 64 --

Rafayel Paramuzyan #16
14 commits 651 ++ 309 --

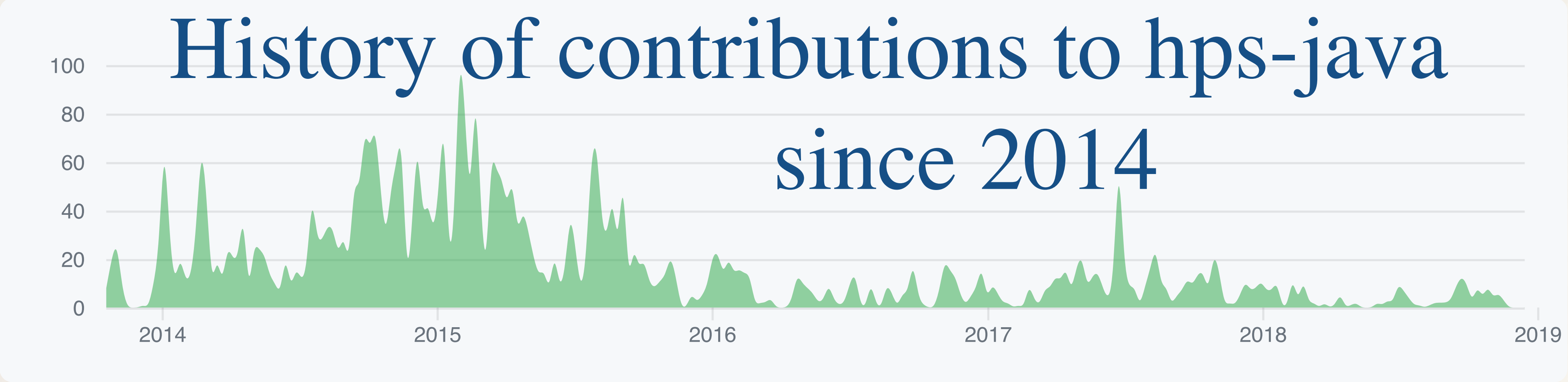
Tony Johnson #17
7 commits 2 ++ 2 --

Luca Marsicano #18
2 commits 918 ++ 54 --

Tim Nelson #19
2 commits 221 ++ 0 --

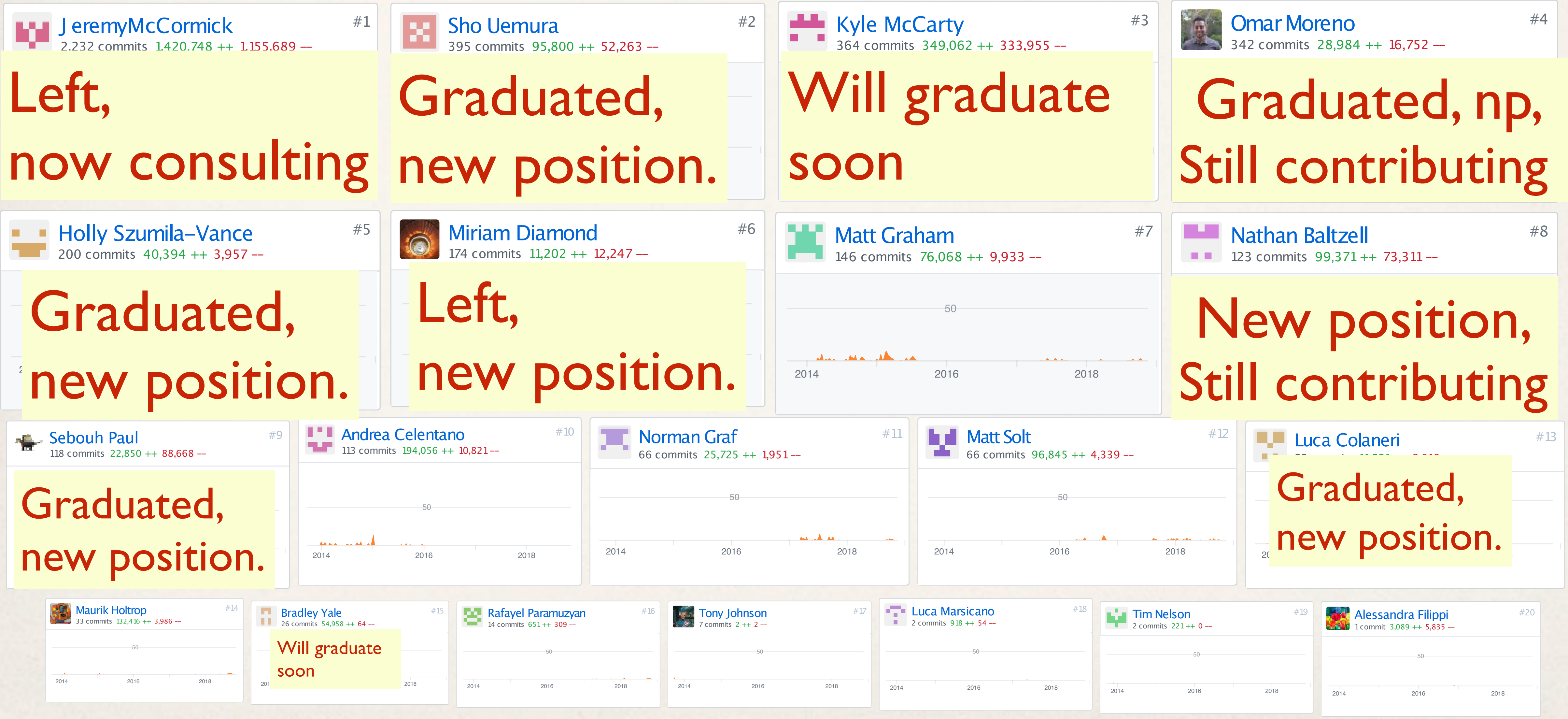
Alessandra Filippi #20
1 commit 3,089 ++ 5,835 --

Software contributions



History of contributions to hps-java since 2014

Not too surprising, there is a fair bit of turnover in the contributors to the software.



J eremyMcCormick #1
2,232 commits 1,420,748 ++ 1,155,689 --

Left,
now consulting

Sho Uemura #2
395 commits 95,800 ++ 52,263 --

Graduated,
new position.

Kyle McCarty #3
364 commits 349,062 ++ 333,955 --

Will graduate
soon

Omar Moreno #4
342 commits 28,984 ++ 16,752 --

Graduated, np,
Still contributing

Holly Szumila-Vance #5
200 commits 40,394 ++ 3,957 --

Graduated,
new position.

Miriam Diamond #6
174 commits 11,202 ++ 12,247 --

Left,
new position.

Matt Graham #7
146 commits 76,068 ++ 9,933 --

Nathan Baltzell #8
123 commits 99,371 ++ 73,311 --

New position,
Still contributing

Sebouh Paul #9
118 commits 22,850 ++ 88,668 --

Graduated,
new position.

Andrea Celentano #10
113 commits 194,056 ++ 10,821 --

Norman Graf #11
66 commits 25,725 ++ 1,951 --

Matt Solt #12
66 commits 96,845 ++ 4,339 --

Luca Colaneri #13

Graduated,
new position.

Maurik Holtrop #14
33 commits 132,416 ++ 3,986 --

Bradley Yale #15
26 commits 54,958 ++ 64 --

Will graduate
soon

Rafayel Paramuzyan #16
14 commits 651 ++ 309 --

Tony Johnson #17
7 commits 2 ++ 2 --

Luca Marsicano #18
2 commits 918 ++ 54 --

Tim Nelson #19
2 commits 221 ++ 0 --

Alessandra Filippi #20
1 commit 3,089 ++ 5,835 --

Conclusions