



# Metrics of computing trends in HEP

Jim Pivarski

Princeton University – IRIS-HEP

May 10, 2022

This is a talk about measuring *physicists*: what they talk about and what they do for computing.

This is a talk about measuring *physicists*: what they talk about and what they do for computing.

Well-defined metrics around software are a clear objective of the NSF OAC SI2/CSSI program: we've taken that very seriously, both to try to gauge our impact and guide our evolving plans.

This is a talk about measuring *physicists*: what they talk about and what they do for computing.

Well-defined metrics around software are a clear objective of the NSF OAC SI2/CSSI program: we've taken that very seriously, both to try to gauge our impact and guide our evolving plans.

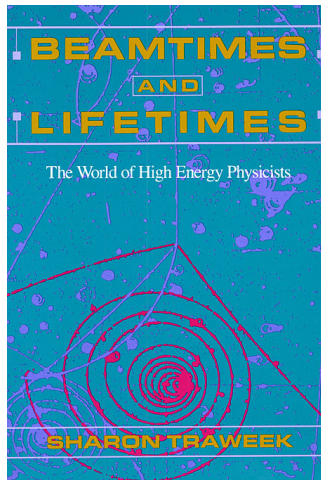
Before joining DIANA/HEP, I went from physics to data science, and had to get used to the idea of measuring people. It's a different kind of analysis: human events are *not* independent and systematic errors dominate.

This is a talk about measuring *physicists*: what they talk about and what they do for computing.

Well-defined metrics around software are a clear objective of the NSF OAC SI2/CSSI program: we've taken that very seriously, both to try to gauge our impact and guide our evolving plans.

Before joining DIANA/HEP, I went from physics to data science, and had to get used to the idea of measuring people. It's a different kind of analysis: human events are *not* independent and systematic errors dominate.

Nevertheless, these kinds of analyses are meaningful: social scientists do it all the time. Inspiration: read Sharon Traweek's anthropological study of physicists at SLAC and KEK in the 1970's. Physicists can be data points!



# My expectation 6 years ago: Spark, Hadoop, functional big data

## Big Data

Jim Pivarski 32/60

Google had an re-indexing problem: a set of webpages containing words had to be re-indexed as a set of words pointing to webpages, so that you can search for pages by keyword.

Their solution, called "map-reduce," was published as a white paper in 2004.

It was immediately reimplemented as an open source product, Apache Hadoop.



Hadoop is now almost synonymous with Big Data, and it has spawned an ecosystem of tools that interoperate with it, much like ROOT in HEP.

# Completely changed by focus group/interview/survey feedback

## Big Data

Lisa Rivardski 22/60

### C++ and Python (Aug 31, mostly)



Google had  
had to be n  
can search

Their solut  
"map-reduc  
as a white

It was imm  
reimplemen  
source prod

Hadoop is  
ecosystem

"I like to use PyROOT because the development time to write code... is very quick. Like, for me, it's probably an order of magnitude faster than C++."

"Something like Python is so much more attractive than something like C++."

"C++ is a language that invites mistakes."

"I'd say my C++ skills are somewhere in the collaboration— not the worst, not the best— but some of the code in CMSSW was written by people with way more appetite for C++ than I have. It can be hard to understand, just looking at the code, what the person was trying to achieve."

16 / 19

# Completely changed by focus group/interview/survey feedback

## Big Data

Google had  
had to be re  
can search

Their solution  
"map-reduce"  
as a white

It was impossible  
reimplement  
source prod

Hadoop is  
ecosystem

## C++ and Python

"I like to use Python  
code... is very  
magnitude faster

"Something like  
something like

"C++ is a language

"I'd say my C++  
the worst, not the  
written by people  
can be hard to  
person was trying to achieve."

## Half-hour interviews with physicists about array syntax



1 grad student, 2 postdocs (beginning & advanced), and 1 advanced researcher

Everyone had most experience in C++ (5 years to decades), less in Python, which was primarily PyROOT (6 months to 3–4 years), very little in Numpy (2 to 5 months).

Some found it easier, some more difficult.

- ▶ "Way, way much easier than applying cuts with for loops."
- ▶ "Surprised by how conceptually different you have to think about selections, combining objects." but "Not good or bad, just surprising that it has a learning curve."
- ▶ "Individual problems have been much more difficult than expected." and "Translating 'if' statements is where I get hung up." but "Not inherently harder; just harder now for those of us used to the 'for' loop version."

18 / 20

16 / 19



# Completely changed by focus group/interview/survey feedback

## Big Data

Google had  
had to be  
can search

Their solu  
"map-redu  
as a white

It was imm  
reimplemen  
source prod

Hadoop is  
ecosystem

## C++ and Python

"I like to use Python  
code... is very  
magnitude faster

"Something like  
something like

"C++ is a language

"I'd say my C++  
the worst, not  
written by people  
can be hard to  
person was trying to achieve."

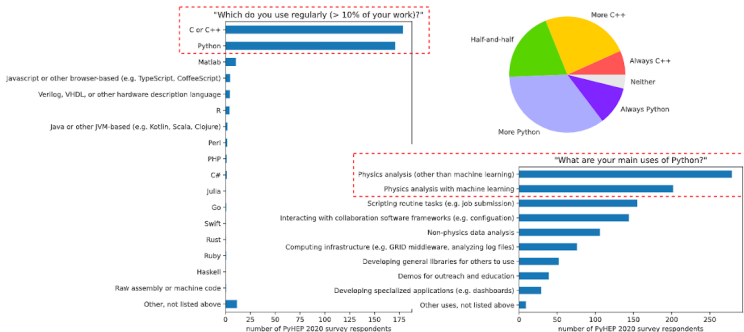
## Half-hour interviews with physicists about array syntax

1 grad student  
Everyone  
primarily from

Some four

- ▶ "Way, way, way faster"
- ▶ "Surprisingly easy to learn"
- ▶ "Individuals who are not familiar with C++"

## Consistent with survey results (PyHEP 2020 participants)



# Ways to study humans

(Important note: I am not an expert. Below is what I learned from conversations with them.)

- ▶ Qualitative:

- ▶ **Focus groups:** most open to unexpected ideas. Want to keep the group size and mix such that participants are willing to speak up. Goal is to discover new *dimensions* of the vector space, not just points within it.
- ▶ **One-on-one interviews:** can be deeper but less broad than focus groups. Lacks the multiplying effect of responding to each other's opinions.
- ▶ **History/documents:** observational, rather than experimental, but this method can reach further into the past.

# Ways to study humans

(Important note: I am not an expert. Below is what I learned from conversations with them.)

## ► Qualitative:

- **Focus groups:** most open to unexpected ideas. Want to keep the group size and mix such that participants are willing to speak up. Goal is to discover new *dimensions* of the vector space, not just points within it.
- **One-on-one interviews:** can be deeper but less broad than focus groups. Lacks the multiplying effect of responding to each other's opinions.
- **History/documents:** observational, rather than experimental, but this method can reach further into the past.

## ► Quantitative:

- **Surveys:** can get large, statistically meaningful datasets, at the cost of losing flexibility/openness to new ideas. Now you *are* filling in a vector space.
- **Proxy metrics:** can measure what people *do*, rather than what they *say*.

# Ways to study humans

(Important note: I am not an expert. Below is what I learned from conversations with them.)

## ► Qualitative:

- **Focus groups:** most open to unexpected ideas. Want to keep the group size and mix such that participants are willing to speak up. Goal is to discover new *dimensions* of the vector space, not just points within it.
- **One-on-one interviews:** can be deeper but less broad than focus groups. Lacks the multiplying effect of responding to each other's opinions.
- **History/documents:** observational, rather than experimental, but this method can reach further into the past.

## ► Quantitative:

- **Surveys:** can get large, statistically meaningful datasets, at the cost of losing flexibility/openness to new ideas. Now you *are* filling in a vector space.
- **Proxy metrics:** can measure what people *do*, rather than what they say.

# Proxy metrics: high statistics, cautious interpretation

google.org Flu Trends

Language: English (United States) ▼

[Google.org home](#)

[Dengue Trends](#)

Flu Trends

[Home](#)

Select country/region ▼

How does this work?

[FAQ](#)

## How does this work?

We've found that certain search terms are good indicators of flu activity. Google Flu Trends uses aggregated Google search data to estimate current flu activity around the world in near real-time.

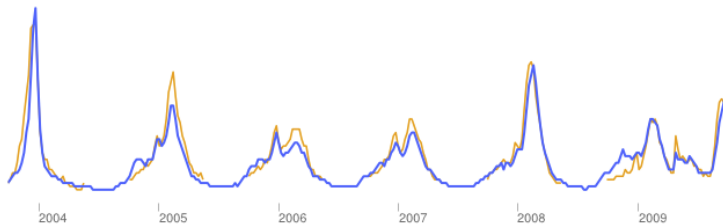
Historical estimates

See data for: United States ▼

### United States Flu Activity

Influenza estimate

● Google Flu Trends estimate ● United States data



United States: Influenza-like illness (ILI) data provided publicly by the [U.S. Centers for Disease Control](#).

# Proxy metrics: high statistics, cautious interpretation

google.org Flu Trends

Language: English (United States) ▼

[Google.org home](#)

[Dengue Trends](#)

Flu Trends

[Home](#)

Select country/region ▼

How does this work?

[FAQ](#)

## Google Flu Trends (2008–2015)

Count searches for things like “fever,” “cough,” interpret as flu activity.

(This was controversial.)

### How does this work?

We've found that certain search terms are good indicators of flu activity. Google Flu Trends uses aggregated Google search data to estimate current flu activity around the world in near real-time.

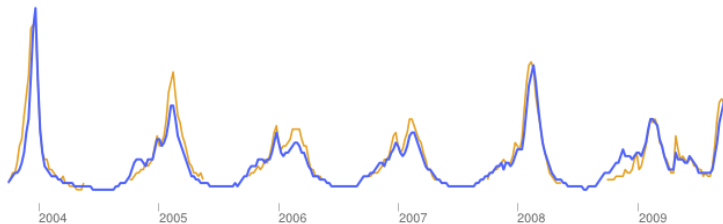
Historical estimates

See data for: United States ▼

### United States Flu Activity

Influenza estimate

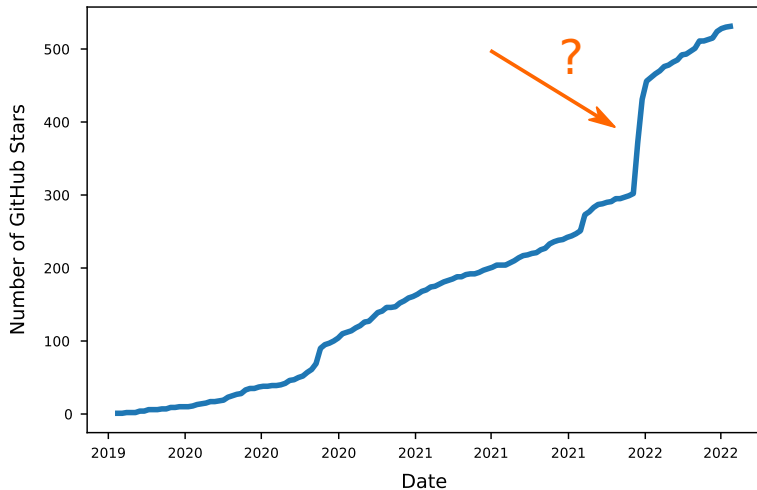
● Google Flu Trends estimate ● United States data



United States: Influenza-like illness (ILI) data provided publicly by the [U.S. Centers for Disease Control](#).

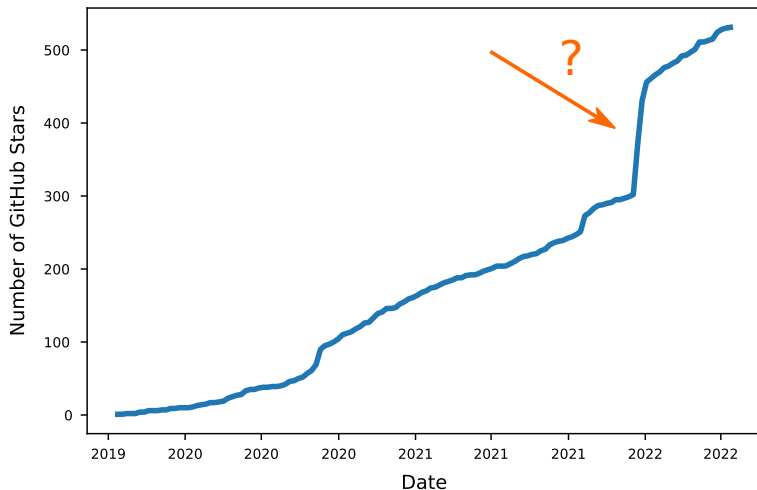
# Example: what happened here?

Number of stars on Awkward Array's GitHub repo versus time



# Example: what happened here?

Number of stars on Awkward Array's GitHub repo versus time

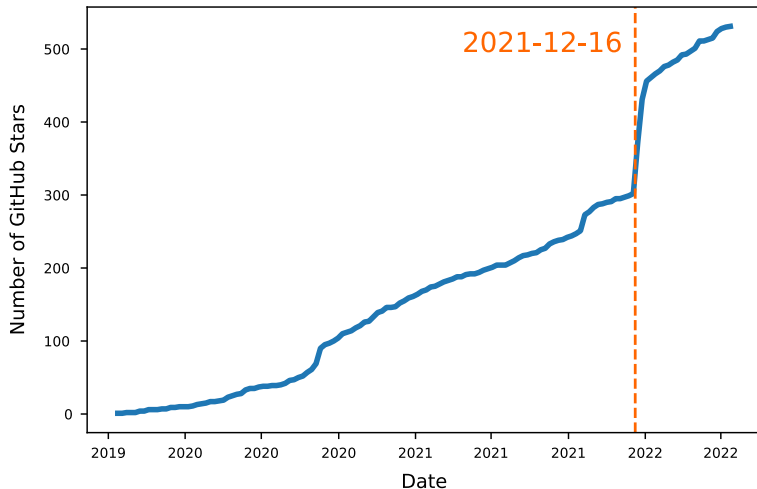


What we're trying to explain is a big, qualitative feature, not the little bumps.



# Example: what happened here?

Number of stars on Awkward Array's GitHub repo versus time



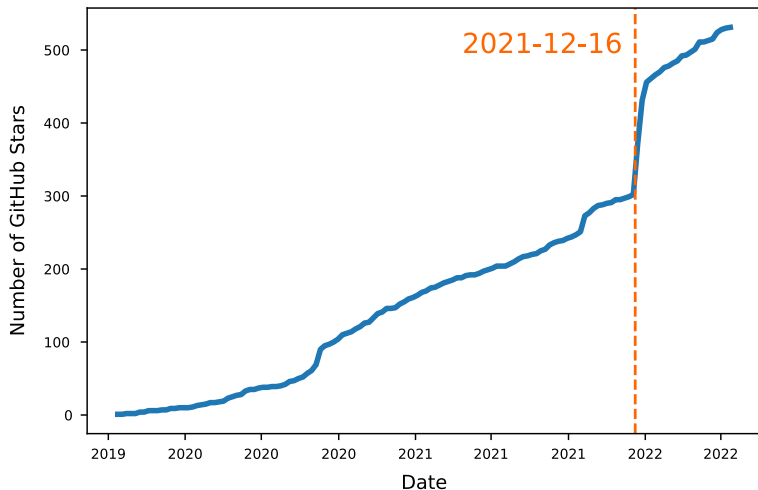
What we're trying to explain is a big, qualitative feature, not the little bumps.

Temporally coincides with Awkward Array on Hacker News:

<https://news.ycombinator.com/item?id=29576323>

# Example: what happened here?

Number of stars on Awkward Array's GitHub repo versus time



What we're trying to explain is a big, qualitative feature, not the little bumps.

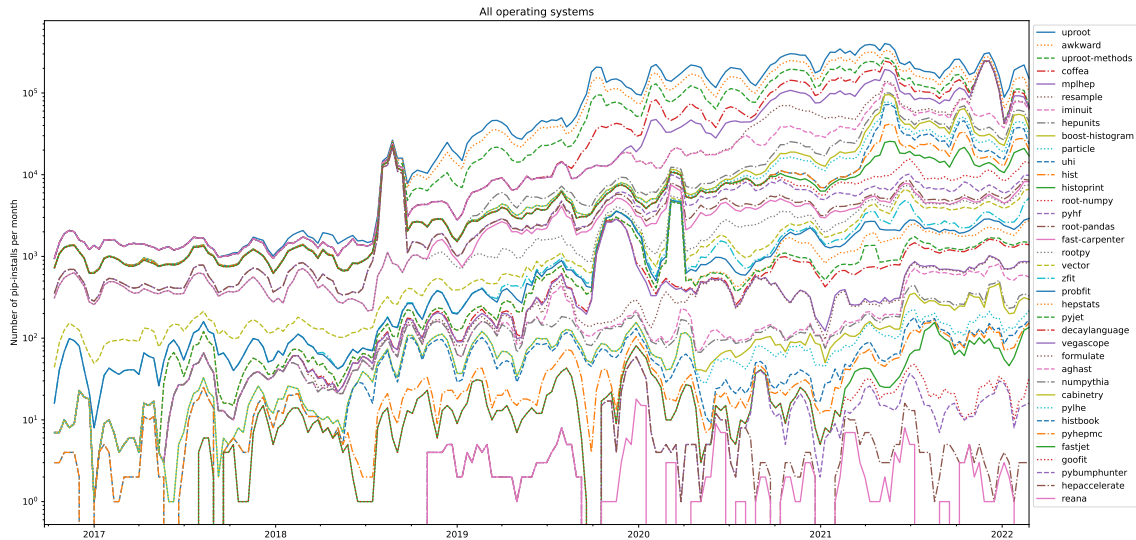
Temporally coincides with Awkward Array on Hacker News:

<https://news.ycombinator.com/item?id=29576323>

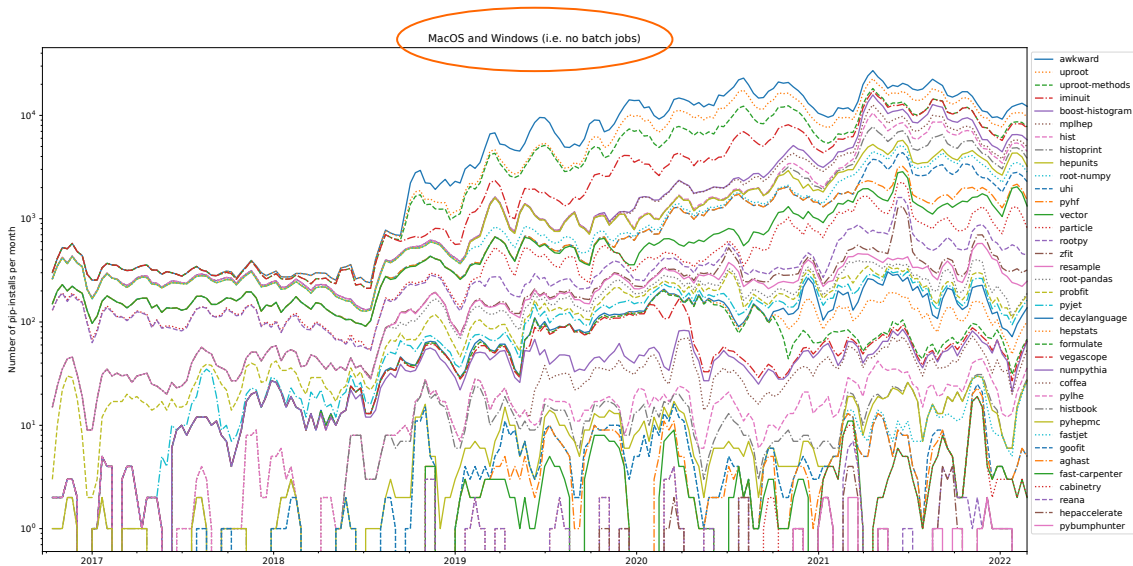
It shouldn't be too controversial to call a correlation like this "causal."

What about counting downloads (a traditionally favorite metric)?

# Stacked download statistics for Scikit-HEP and related packages



# Stacked download statistics for Scikit-HEP and related packages



Linux includes batch jobs, which sometimes `pip install` the same package on thousands of workers.

Linux includes batch jobs, which sometimes `pip install` the same package on thousands of workers.

Selecting only MacOS and Windows removes most batch jobs, but it excludes some individual users (like me), probably with a behavioral bias.

Linux includes batch jobs, which sometimes `pip install` the same package on thousands of workers.

Selecting only MacOS and Windows removes most batch jobs, but it excludes some individual users (like me), probably with a behavioral bias.

Still, there's continuous testing jobs on MacOS and Windows.



Linux includes batch jobs, which sometimes `pip install` the same package on thousands of workers.

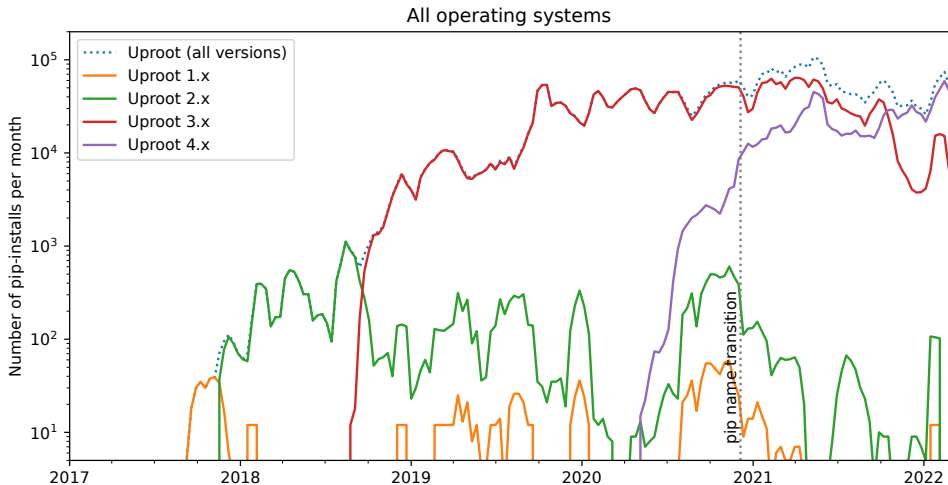
Selecting only MacOS and Windows removes most batch jobs, but it excludes some individual users (like me), probably with a behavioral bias.

Still, there's continuous testing jobs on MacOS and Windows.

Do we even *want* to exclude these things? What do we *want* the observable to quantify?

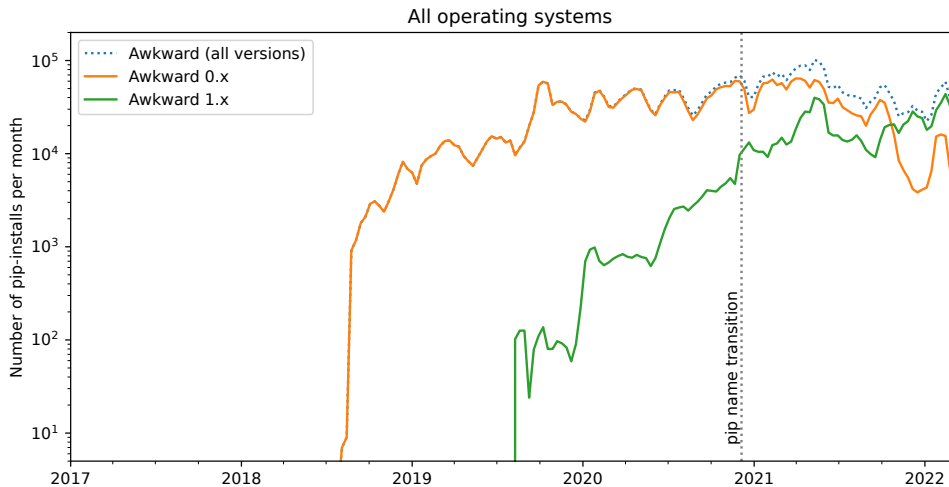
More often useful when *comparing* two things

## Transition from “old” Uproot to “new”



More often useful when *comparing* two things

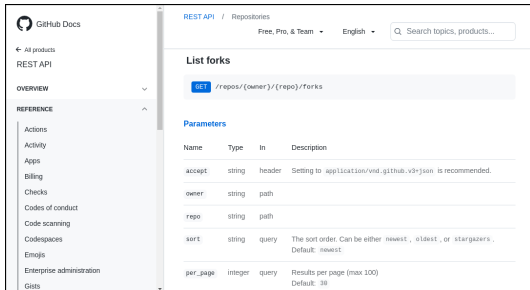
## Transition from “old” Awkward to “new”



# Directed study: how are physicists using C++ and Python?

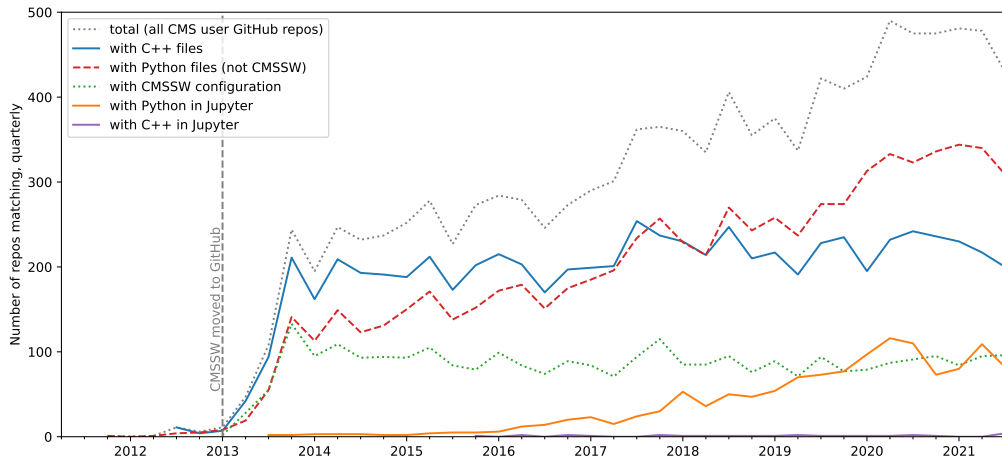
Analyze code in 11 635 GitHub repos by 2 172 physicists:

1. Ask GitHub which users forked CMSSW and call them “CMS physicists.” (CMSSW has been on GitHub for a long enough time to see trends.)
2. Clone all of the physicists’ repos (the ones that are not forks of something else).
3. Search the code of these repos and count matches.
4. Take care to exclude CMSSW configuration files, which are also Python.



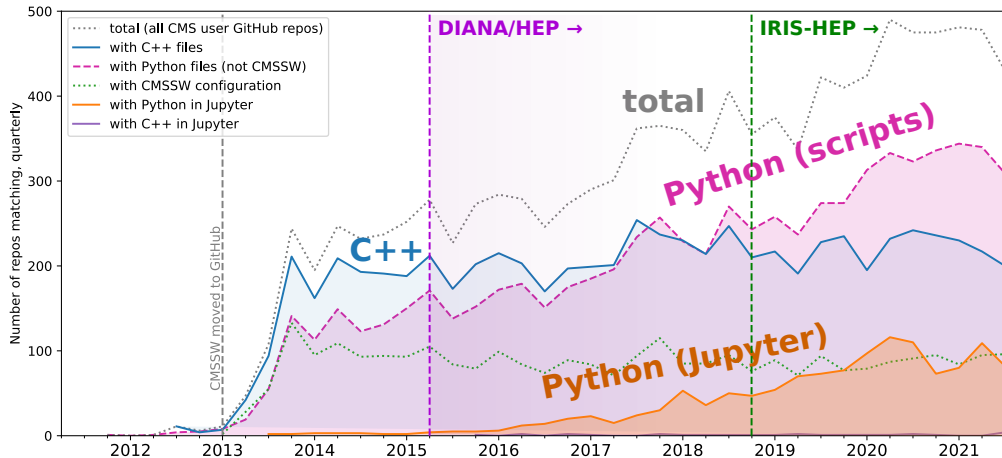
# Language use: C++, Python, and Jupyter

## Number of non-fork GitHub repos created by CMS physicists



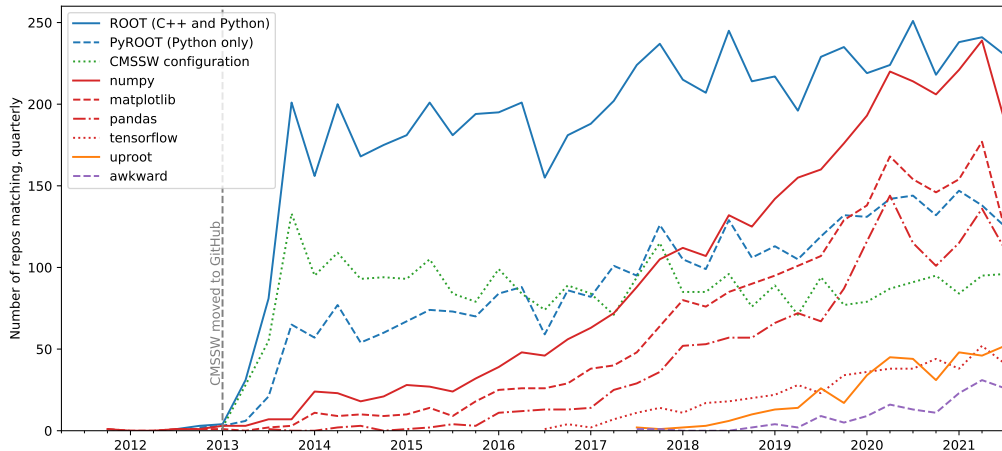
# Language use: C++, Python, and Jupyter

## Number of non-fork GitHub repos created by CMS physicists



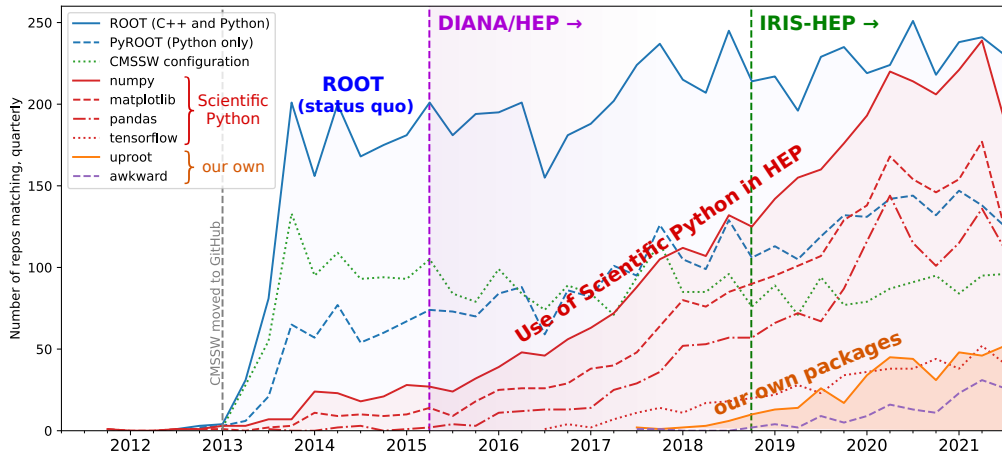
# Packages: ROOT, Scientific Python, Uproot/Awkward

Same sample, now counting matches for `import XYZ`, `from XYZ import`, etc.



# Packages: ROOT, Scientific Python, Uproot/Awkward

Same sample, now counting matches for `import XYZ`, `from XYZ import`, etc.





# Longer baseline: title/abstract matches in InspireHEP

- ▶ CHER and ACAT are the two major conferences for computing in HEP.
- ▶ CHER started in 1985 and includes 5 407 proceedings.
- ▶ ACAT started in 1990 and includes 1 446 proceedings.
- ▶ Search all the titles and abstracts for interesting keywords!

The screenshot shows the InspireHEP website header with logos for CERN, DESY, CERN, SLAC, and Fermilab. Below the header, there are navigation links for 'ABOUT' and 'GO TO INSPIRE'. The main content area features a green box with the number '15' and the date 'Jun 2020', followed by the headline 'WE RELEASED THE NEW INSPIRE API' by Stella Christodoulaki. The text below the headline states: 'We are happy to announce the release of the new INSPIRE REST API! Documentation: <https://github.com/inspirehep/rest-api-doc>'. It then lists the highlights of the new API: 'Access data from the search and detailed record pages programmatically', 'JSON-based: in contrast to the API on the old INSPIRE, the new API uses JSON as primary data format, which is easy to use in a vast number of programming languages, and uses human-readable and well-documented fields.', and 'Other formats, like BibTeX or LaTeX, are directly available and don't require extracting them from an HTML page'. On the right side, there is a search bar and a 'Categories' section listing: Authors (8), Collaborations (6), Collections (25), Conferences (1), Data (4), HEP (15), Jobs (4), INSPIRE (35), and References and Citations (26).

INSPIRE HEP

SLAC Fermilab

ABOUT GO TO INSPIRE

15 Jun 2020

WE RELEASED THE NEW INSPIRE API

by Stella Christodoulaki

We are happy to announce the release of the new INSPIRE REST API!

Documentation: <https://github.com/inspirehep/rest-api-doc>

The highlights of the new API are:

- Access data from the search and detailed record pages programmatically
- JSON-based: in contrast to the API on the old INSPIRE, the new API uses JSON as primary data format, which is easy to use in a vast number of programming languages, and uses human-readable and well-documented fields.
- Other formats, like BibTeX or LaTeX, are directly available and don't require extracting them from an HTML page

Search

Categories

Authors (8)

Collaborations (6)

Collections (25)

Conferences (1)

Data (4)

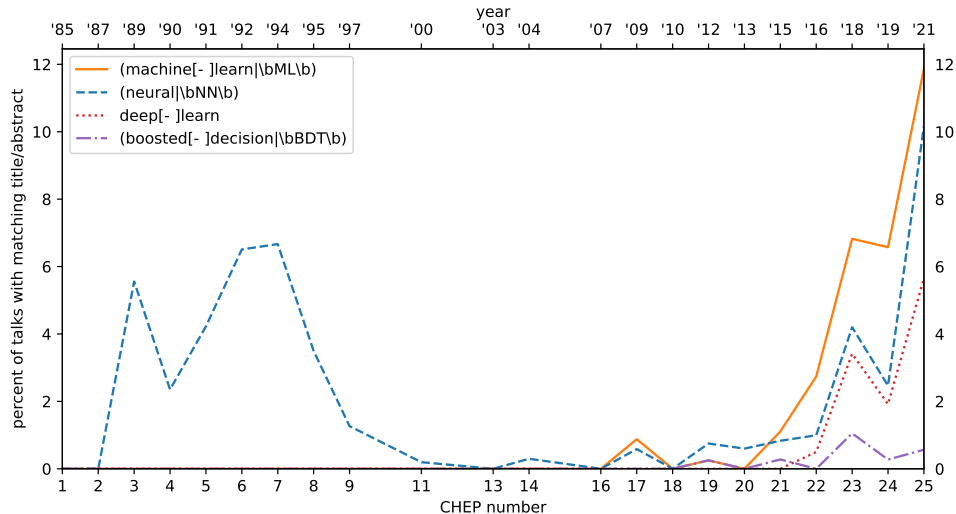
HEP (15)

Jobs (4)

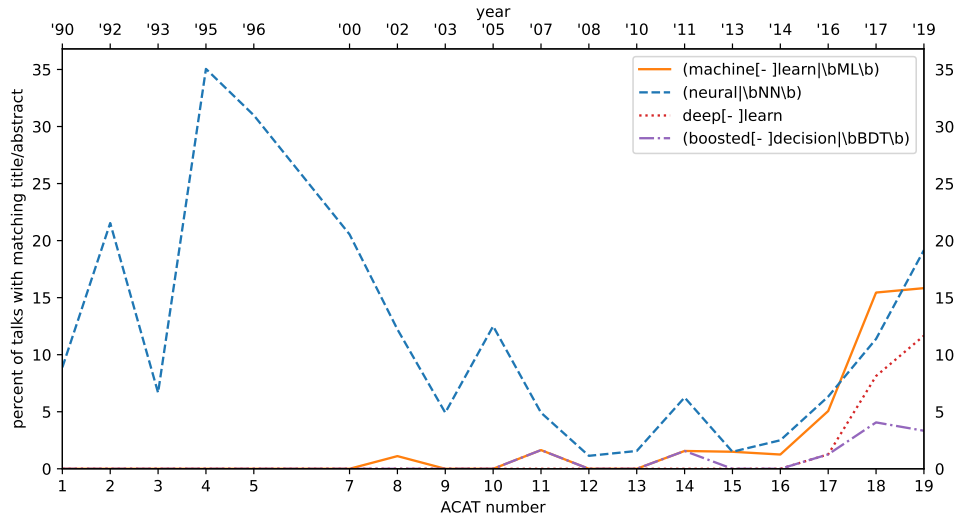
INSPIRE (35)

References and Citations (26)

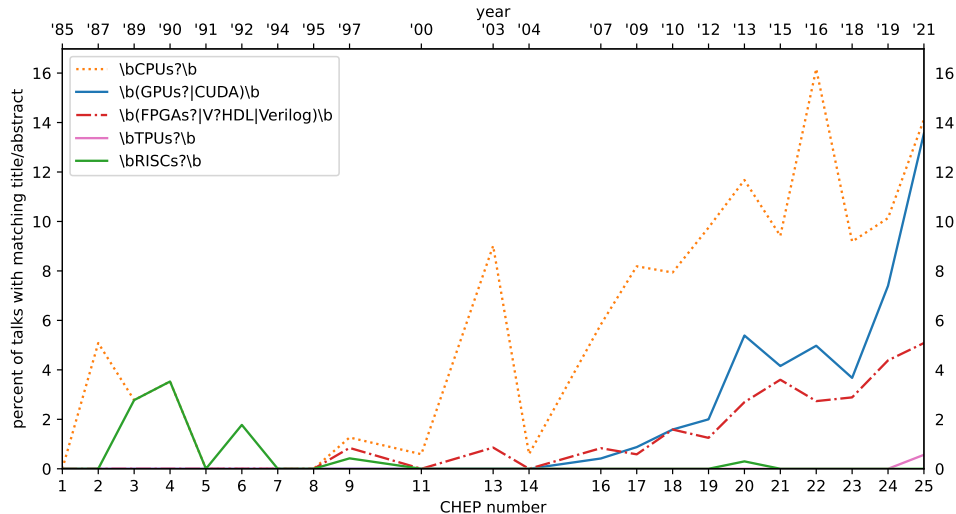
# Machine learning in CHEP papers



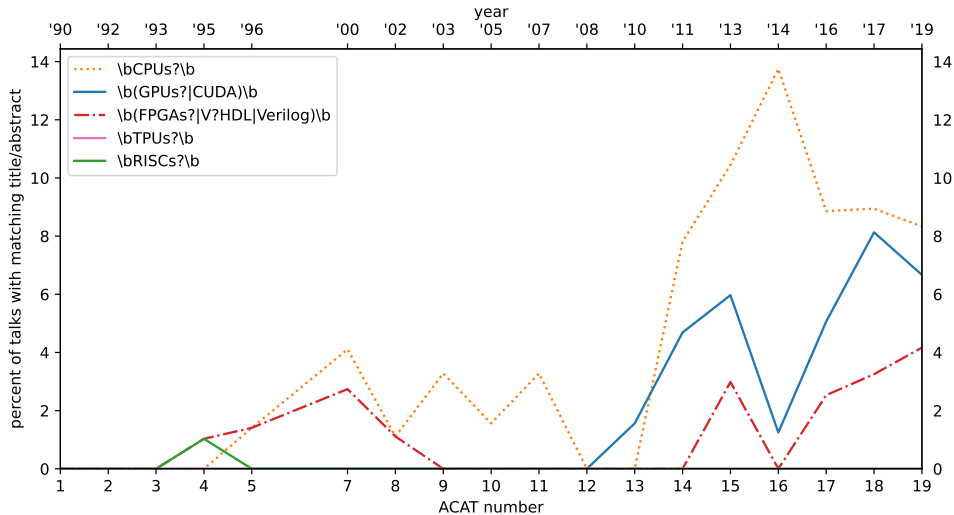
# Machine learning in ACAT papers



# Hardware accelerators in CHEP papers

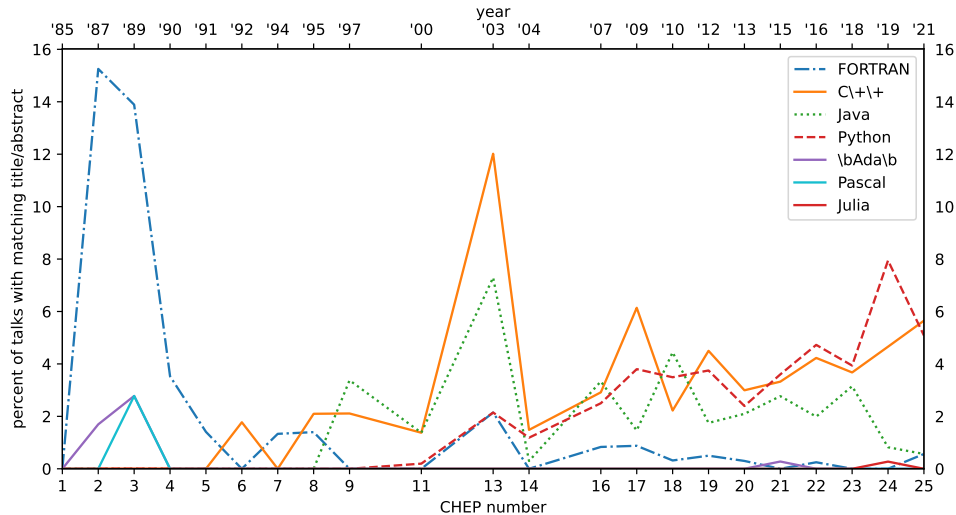


# Hardware accelerators in ACAT papers

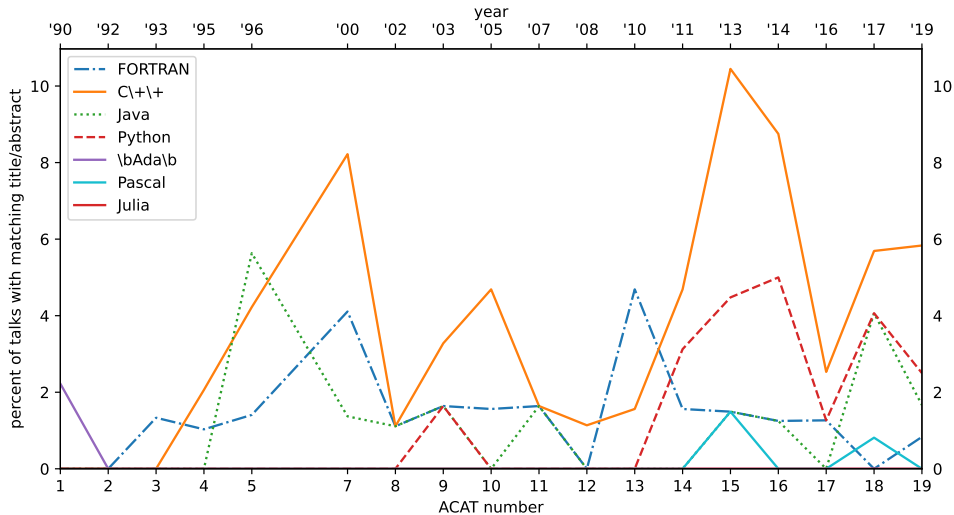


What about language transitions? Fortran  $\rightarrow$  C++  $\rightarrow$  Python?

# Programming languages in CHEP papers



# Programming languages in ACAT papers







## **Emerging Standard ? Python as “Software Glue”**

### ■ Clear trend towards Python

- ❖ Used by: ATLAS (Athena), CMS, D0, LHCb (Gaudi), SND,...
- ❖ Used by: Lizard/Anaphe, HippoDraw, JAS (Jython)...
- ❖ Architecturally, scripting is “just another service”
- ❖ ROOT is the exception to the “Python rule”
  - CINT interpreter plays a central role
  - Developers and users seem happy

### ■ Python is popular with developers...

- ❖ Rapid prototyping; gluing together code
- ❖ (Almost) auto-generation of wrappers (SWIG)

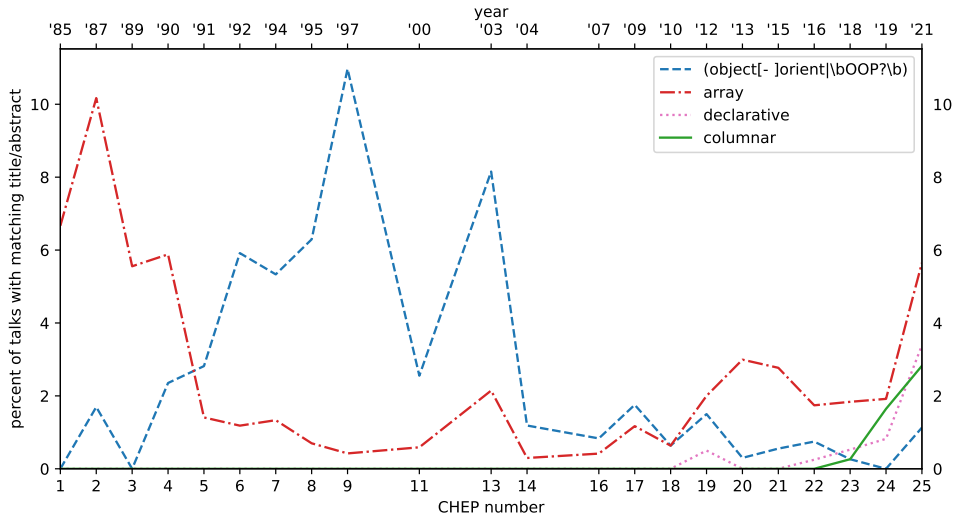
### ■ ...but acceptance by users not yet proven

- ❖ Another language to learn, syntax,...

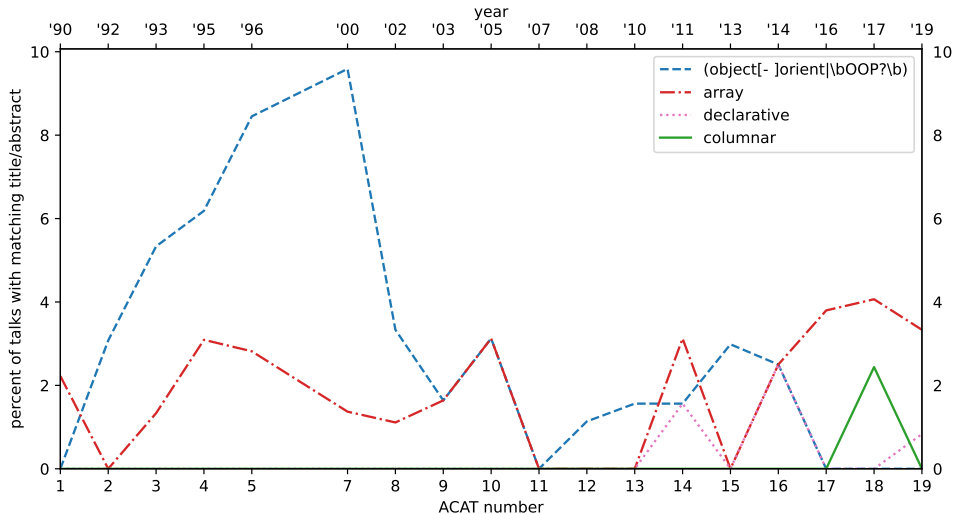
*“Summary of Track 2: Data Analysis and Visualisation”  
Lucas Taylor, Northeastern U. CHEP 01, Beijing, 3-7 S*

Note: PyROOT  
introduced in  
2004 (v4.00/04).

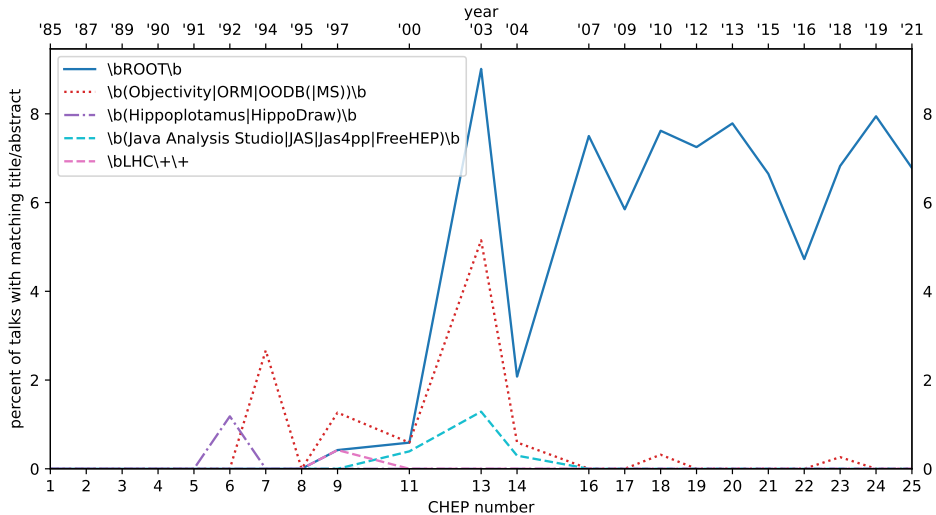
# Programming paradigms in CHEP papers



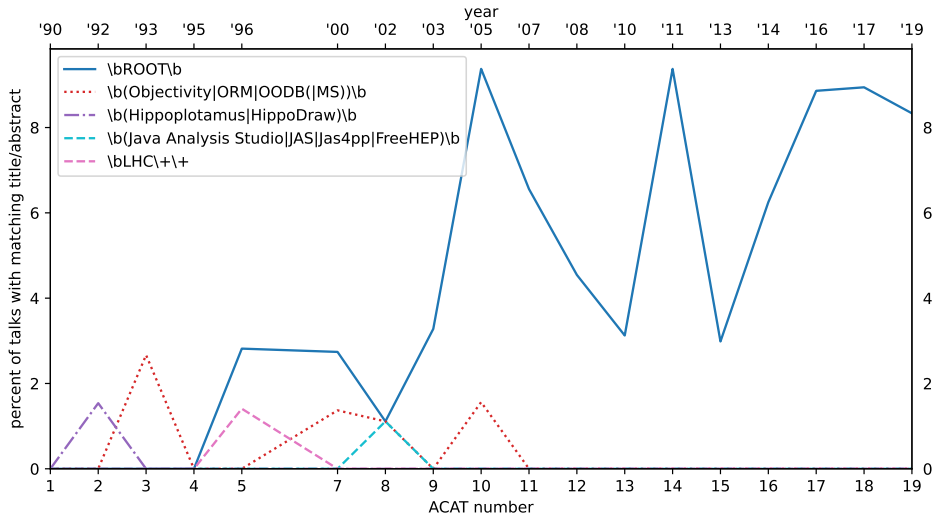
# Programming paradigms in ACAT papers



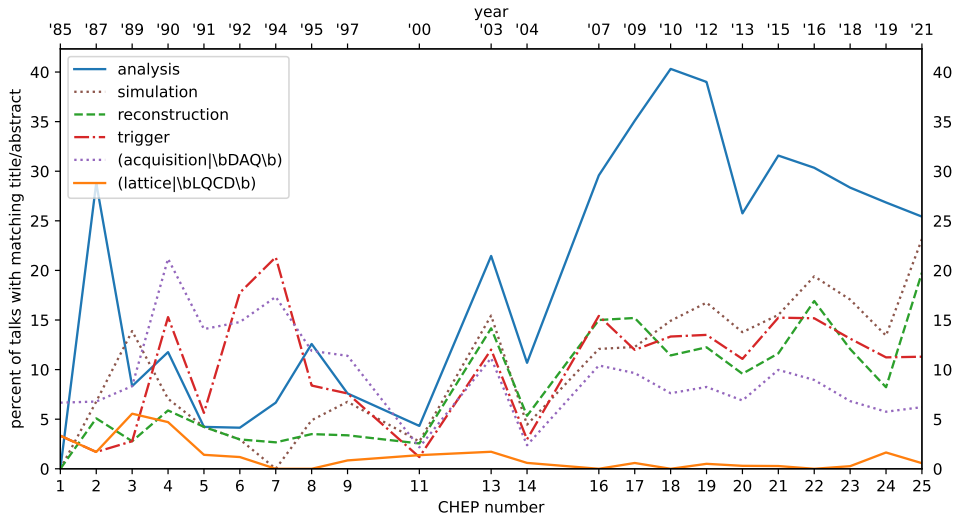
# Software frameworks in CHEP papers



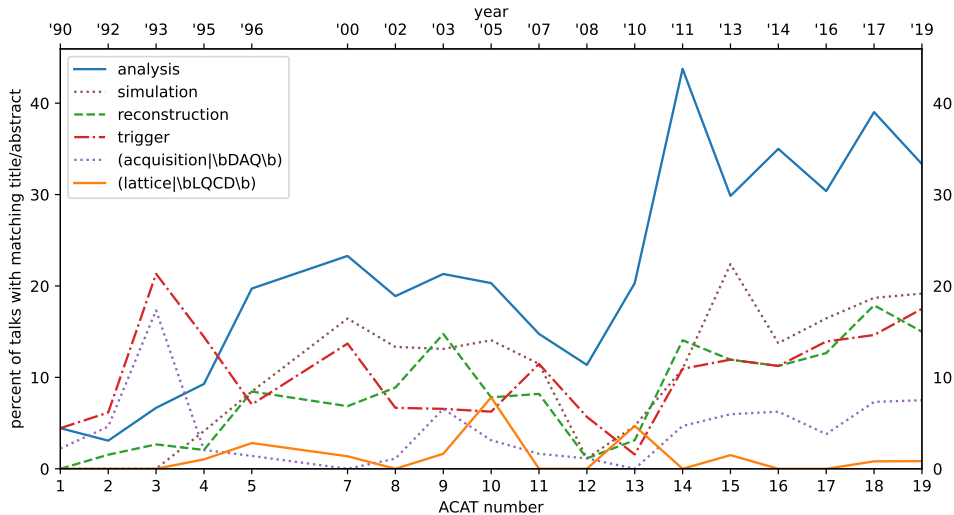
# Software frameworks in ACAT papers



# Kinds of tasks in CHEP papers



# Kinds of tasks in ACAT papers



- ▶ Different ways of understanding people, including the HEP software community: focus groups, interviews, historical documents, surveys, and proxy metrics.
- ▶ This talk focused on proxy metrics, which are quantitative, but you have to pay close attention to what they're quantifying.
- ▶ Some clear trends and conclusions emerged. Others are muddled.