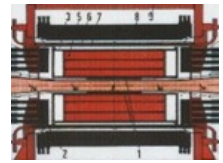
listing



Data preservation

(JADE status)

Andrii Verbytskyi

Remote DPHEP meeting,
March 1, 2021

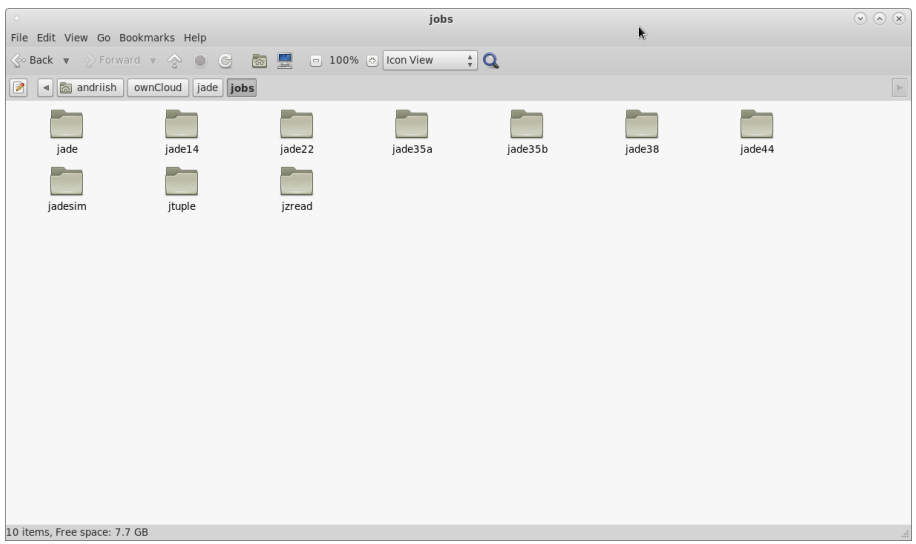
JADE experiment

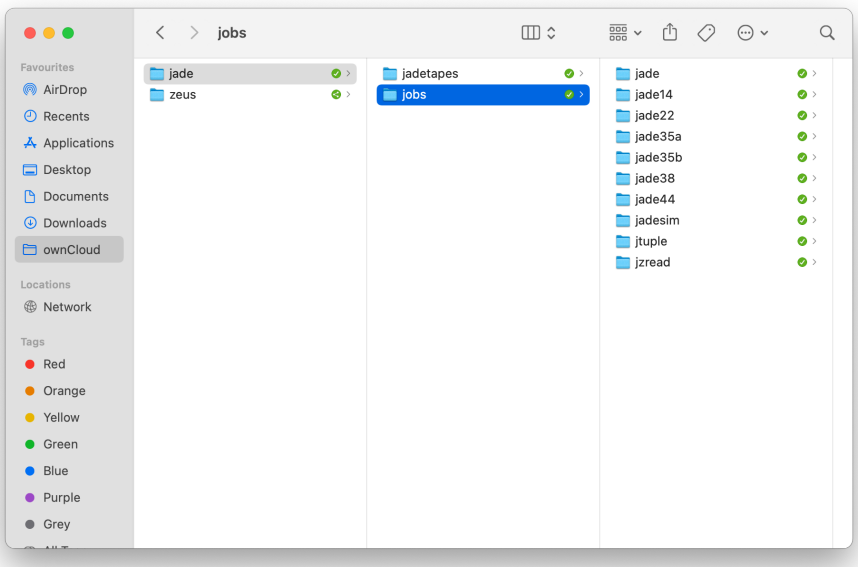
The JADE experiment waslocated at the PETRA e^+e^- storage ring at DESY in Hamburg, Germany. The experiment took data between 1979 and 1986 in the center-of-mass range between 12 and 46.6 GeV.

- Site: `https://wwwjade.mpp.mpg.de/`
- People@MPP: S. Bethke, S. Kluth, H. von der Schmitt, A. Verbytskyi.
- Data: All data of JADE, some old and more new MC.
- Documentation: Scanned internal notes.
- Services: Storage for user analysis, computing resources, MC generation.
- Physics: All included.

JADA data and documentation

- As of March 1, 2021 JADE data are available in MPP/MPCDF either in the dCache, archive system and in ownCloud *NEW*.
- The size of JADE data is much smaller (600Gb) than of modern experiments → one can use MPCDF ownCloud to store/access it. Works on Windows, MacOSX and Linux.
- The internal notes are now available at `https://wwwjade.mpp.mpg.de/compnotes.html` *NEW*





Data is accessible from anywhere with a **standard software.**

JADE software

The main thing about the JADE software: it is Fortran.

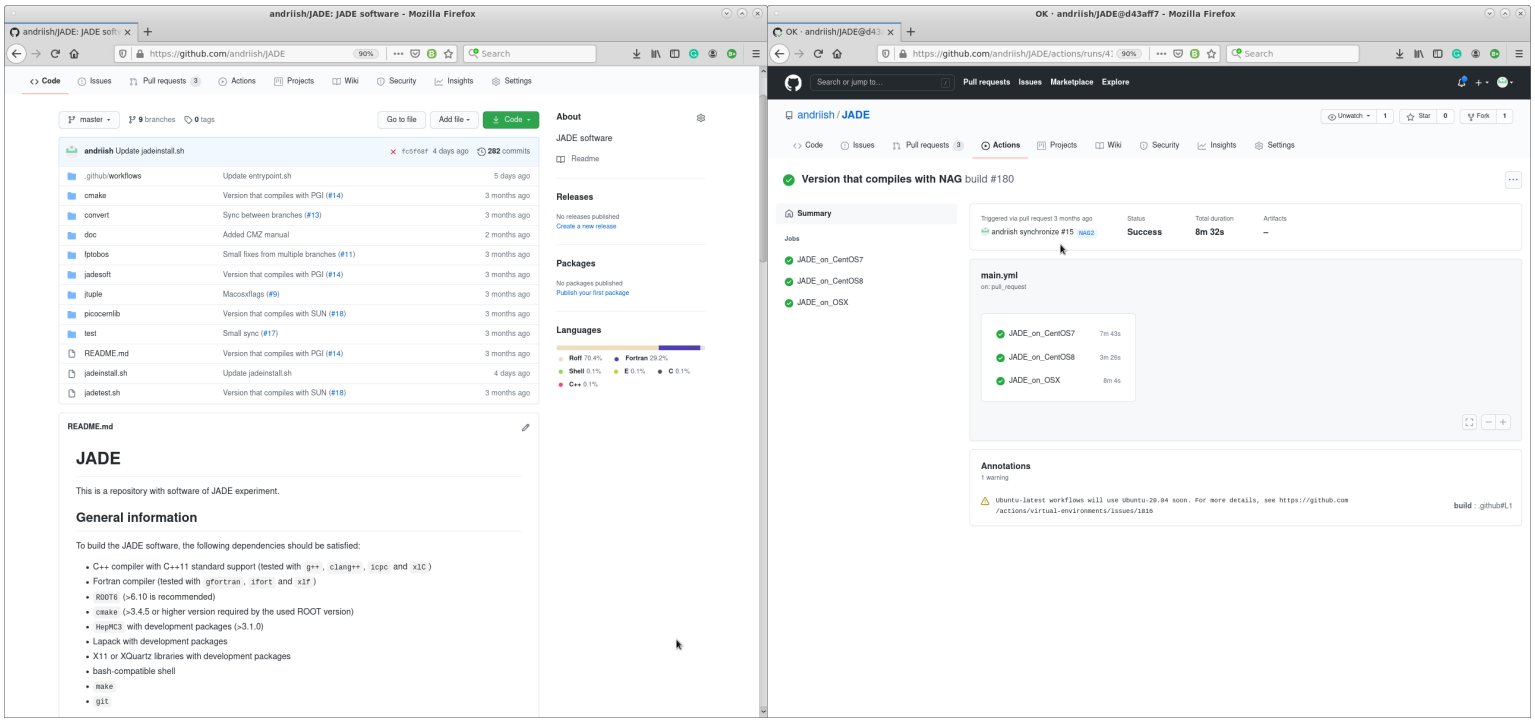
- The software is de-facto public for a long time from `https://wwwjade.mpp.mpg.de/`.
- The GitHub repository `https://github.com/andriish/JADE`.
- The software consists of
  - An **obsolete** set of MC generation codes, e.g. pythia5, jetset (obolete).
  - An interface to modern MC generators (OK).
  - The detector "simulation" software from 198x+ (OK).
  - The reconstruction software from 198x+ (OK).
  - The analysis software from 200x+ (OK).
  - Event display (partially OK).
  - Utilities to deal with data and calibration from 198x+ (OK).
- The repository is a standard GitHub repository with documentation^{NEW}, CI^{NEW} and so on.

To compile and run the JADE software one needs a **suitable** Fortran/C++ compiler toolchain, cmake and ROOT. Again: **standard software.**

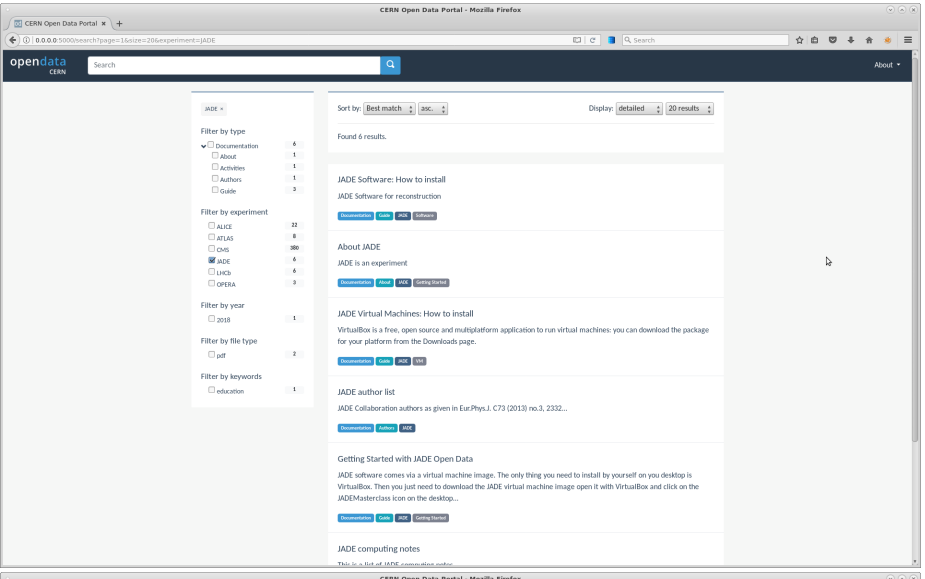
JADE software: recent developments

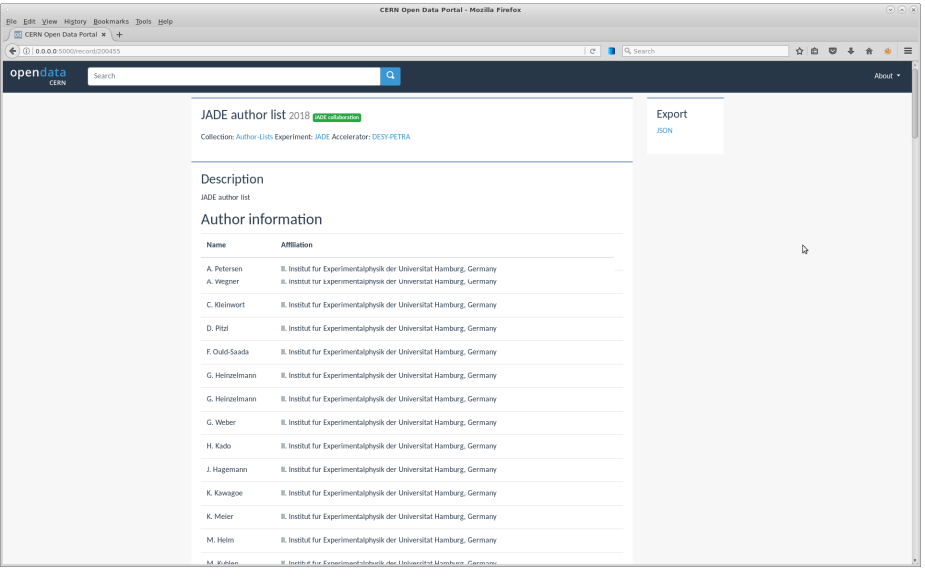
More portability, testing and documentation.

- GNU and IBM toolchains support extended with preliminary Intel[NEW] and NAG[NEW]. GNU is still the most stable one.
- More CI tests[NEW].
- Updated the site and documentation[NEW].
- Support for CentOS8[NEW] and MacOSX10.15+ on x86_64[NEW]



JADE @ opendata.cern.ch ?





- JADE data, software, and documentation could be stored also in CERN if an agreement is reached.
- **. . . however, even without this agreement one can technically prepare the data to be accessible in a similar way in MPP/MPCDF and in GitHub[a] or any similar service.**

[a] GitHub has easier access to MacOSX

Conclusions

- JADE DP stack is based on open standards, does not rely on specific SW and is extremely portable. One can run it completely on desktop.
- **“JADE – collider experiment on your desktop”.**

Backup slide: The CI tests

- Convert the preserved raw data into JADE-readable format
- Reconstruct the data and run analysis on the real data (several stages)
- Generate MC sample with a modern generator
- Convert the sample into JADE-readable format
- 'Simulate' the MC sample
- Reconstruct the MC event and run analysis on them (several stages)

The existing test setup can be scaled to full-scale analysis.
At some point one could make the available e^+e^- preliminary analyses public however one should think about the scheme and copyright for the involved people.

Backup slide: The CI configuration

Standard GitHub Actions VM for MacOSX/stock Docker for CentOS, e.g.

```
FROM centos:8
COPY entrypoint.sh /entrypoint.sh
ENTRYPOINT ["/entrypoint.sh"]
```

Updated software in each run using homebrew/CentOS repositories, e.g.

```
#!/bin/sh -l
set -x
yum -y install  epel-release dnf-plugins-core
dnf config-manager --set-enabled powertools
yum -y install  gcc gcc-c++ gcc-gfortran make which cmake cmake-data cmake-filesystem
     HepMC3*
yum -y install  lapack-static  lapack-devel lapack  gengetopt  blas-devel blas atlas-devel
      atlas  openblas-devel openblas openblas-serial64 openblas-threads  --skip-broken
yum -y install  libX11-devel libX11  libXmu-devel libXmu libXau-devel libXau libXcursor-
     devel  libXcursor  libSM-devel libSM libICE libICE-devel libXext-devel libXext
yum -y install  root-*6* --exclude=*doc* --exclude=*debug* --skip-broken

sh jadeinstall.sh

yum -y install pythia8-devel pythia8 pythia8-data

sh jadetest.sh

out=$?
echo ::set-output name=out::$out
```

Rebuilds on push and once per month → **Always updated.**