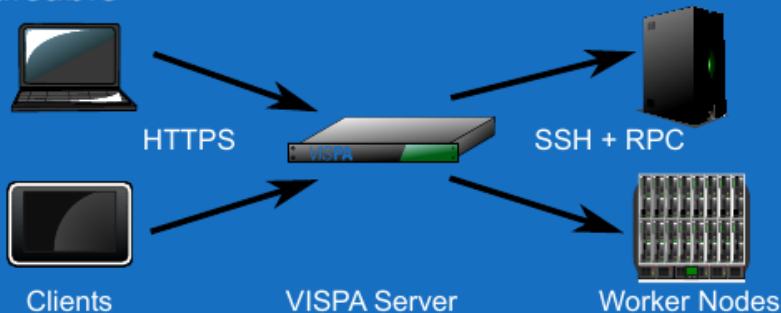


Christian Glaser, Martin Erdmann, Robert Fischer, Daniel von Asseldonk, Marcel Rieger, Gero Müller, Martin Urban, Thorben Quast
 Contact: glaser@physik.rwth-aachen.de

Data Analysis in a Web Browser

- Graphical front-end to your infrastructure
 - Makes your software, data and computing resources available through the web
 - Usable from any system
- Base functionality provided
 - User management
 - Apps: file browser, terminal, code editor
- System is extendable with own apps using most common web technologies
 - HTML5, CSS3, jQuery, bootstrap, template rendering
- Scalable



- Every computer supporting SSH + Python can be used as worker
- Bootstrap method: worker is configured automatically

What's new?

- Complete GUI redesign
 - Look and feel of a desktop software
 - Frequently used operations are accessible with a single mouse click or a shortcut
- Preference system for individual taste
- Code editor with direct Python script execution

```

def process(self, obj, sink):
    event = toEvent(obj)
    selectedLeptons = []
    for particle in event.getParticles():
        # loop over all particles to find leptons
        if particle.getName() in self._leptonName:
            # this particle has the correct name
            # it is considered a candidate for a lepton now
            leptonCandidate = particle
            # now check for required pT and eta criteria
            if leptonCandidate.getPt() < self._minLeptonPt:
                # not enough transverse momentum
                continue
            if leptonCandidate.getEta() > self._maxLeptonEta:
                # the pseudo-rapidity is too big
                # the candidate propagates too far into the
                # forward direction of the detector
                continue
            # all candidates here fulfill pT and eta criteria
            selectedLeptons.append(leptonCandidate)
    # selectedLeptons now contains all of the accepted leptons
    # need to check whether the total number of leptons is as desired
    if len(selectedLeptons) < self._minLeptons:
        # not enough leptons in this event
        return "veto"
    if len(selectedLeptons) > self._maxLeptons:
        # too many leptons in this event
        return "veto"
    # if this point is reached, the event contains the
    # desired number of selected leptons
    # the event is forwarded to the "passed" output port (aka source)
    return "passed"
def endJob(self):
    pass
    
```

executed "python /home/cglaser/CERN/.../Z_mass.py" ↑ Top ↓ Bottom ↻ Clear

OUTPUT:
 Info in <Namespace:Print>: png file /home/cglaser/CERN/Z_mass.png
 Lepton selection criteria:
 * particle name: Lepton
 * minimal transverse momentum: 30 GeV
 * maximum pseudo rapidity: 2.1
 * minimal number of leptons: 2
 * maximal number of leptons: 2

runtime: 2.13 s

HOME/CERN

Events

Events	71
Mean	88.91
RMS	6.66
Underflow	0
Overflow	0

m_2 [GeV/c²]

Code-Editor with direct Python Script Execution

Browser: VISPA | URL: https://vispa.physik.rwth-aachen.de/server/ | VISPA Cluster | Feedback | cglaser

Workspace: CERN | File: Z_mass.py

Buttons: Save, Save as..., Execute, Abort

```
52 def process(self, obj, sink):
53     event = toEvent(obj)
54
55     selectedLeptons = []
56     for particle in event.getParticles():
57         # loop over all particles to find leptons
58
59         if particle.getName() in self._leptonName:
60             # this particle has the correct name
61             # it is considered a candidate for a lepton now
62             leptonCandidate = particle
63
64             # now check for required pT and eta criteria
65             if leptonCandidate.getPt() < self._minLeptonPt:
66                 # not enough transverse momentum
67                 continue
68
69             if leptonCandidate.getEta() > self._maxLeptonEta:
70                 # the pseudo-rapidity is too big
71                 # the candidate propagates too far into the
72                 # forward direction of the detector
73                 continue
74
75             # all candidates here fulfill pT and eta criteria
76             selectedLeptons.append(leptonCandidate)
77
78     # selectedleptons now contains all of the accepted leptons
79     # need to check whether the total number of leptons is as desired
80     if len(selectedLeptons) < self._minNLeptons:
81         # not enough leptons in this event
82         return "veto"
83
84     if len(selectedLeptons) > self._maxNLeptons:
85         # too many leptons in this event
86         return "veto"
87
88     # if this point is reached, the event contains the
89     # desired number of selected leptons
90
91     # the event is forwarded to the "passed" output port (aka source)
92     return "passed"
93
94 def endJob(self):
95     pass
96
```

executed "python /home/cglaser/CERN/..."

OUTPUT:

Info in <TCanvas::Print>: png file /home/cglaser/CERN/Z_mass.png

Lepton selection criteria:

- * particle name: Lepton
- * minimal transverse momentum: 30 GeV
- * maximum pseudo rapidity: 2.1
- * minimal number of leptons: 2
- * maximal number of leptons: 2

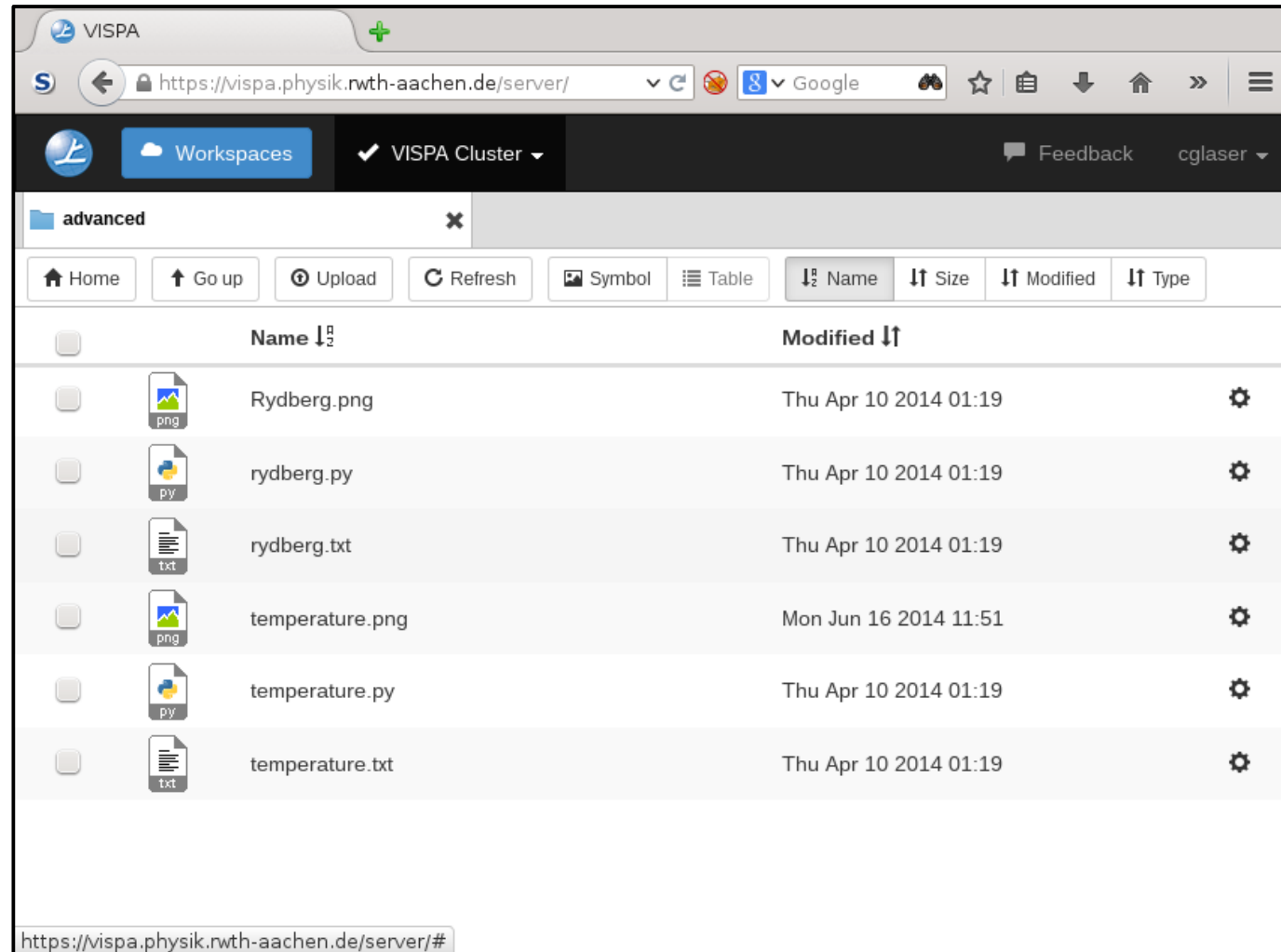
runtime: 2.13 s

\$HOME/CERN

Statistic	Value
Entries	71
Mean	89.91
RMS	5.965
Underflow	0
Overflow	0

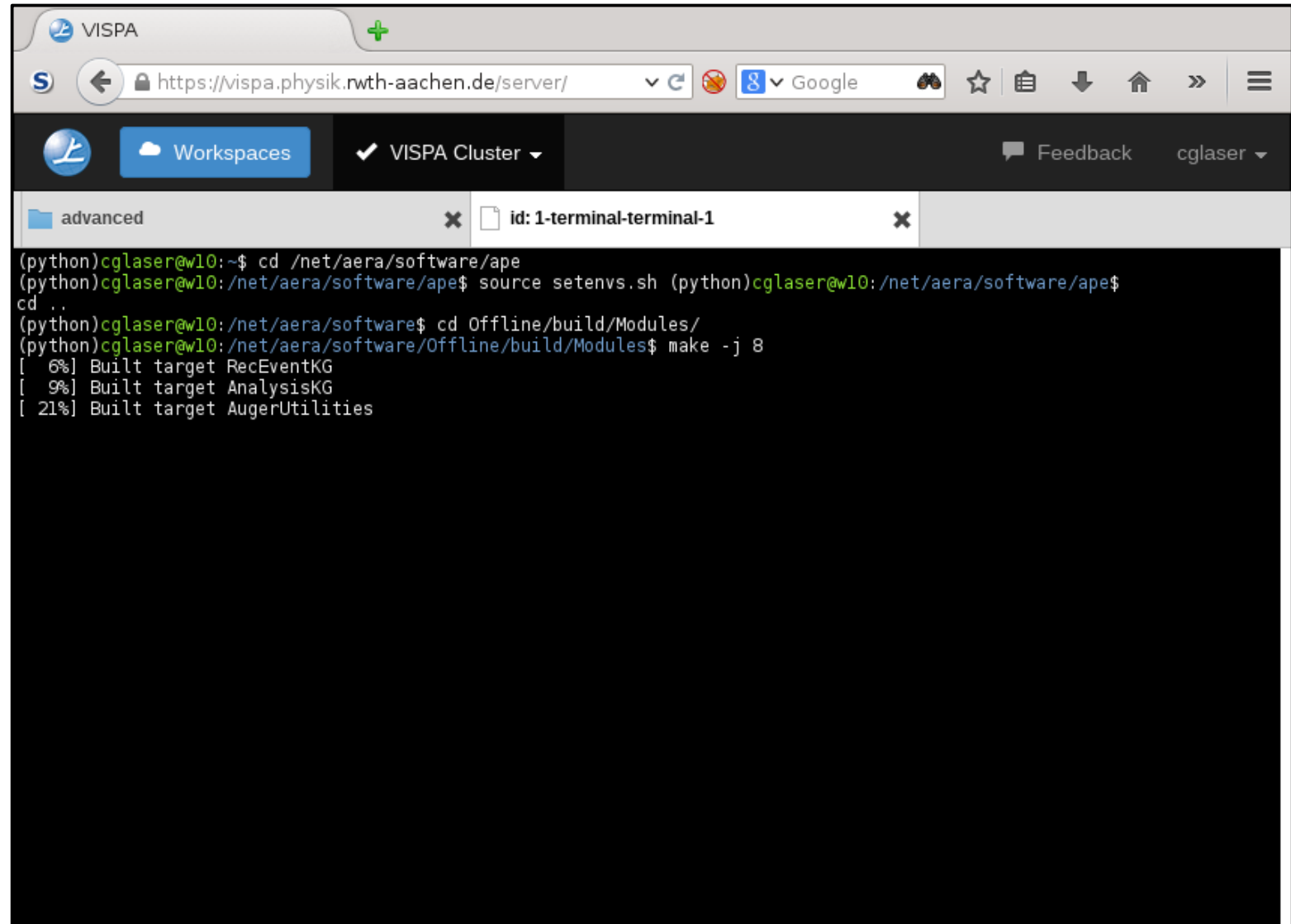
File-Browser

- Manage files on workspace
- Shortcuts supported
- Upload / Download
- View images



Terminal

- Features without GUI implementation can be accessed via terminal



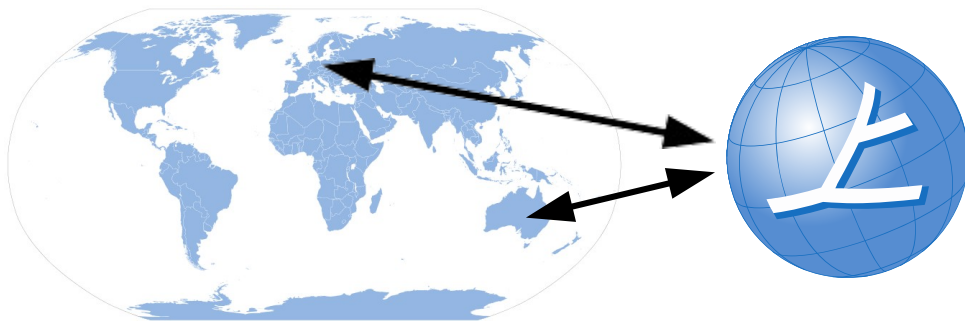
The screenshot shows a web browser window with the URL `https://vispa.physik.rwth-aachen.de/server/`. The browser interface includes a search bar with 'Google' and navigation icons. Below the browser, there is a terminal window titled 'id: 1-terminal-terminal-1'. The terminal shows the following commands and output:

```
(python)cglaser@w10:~$ cd /net/aera/software/ape
(python)cglaser@w10:/net/aera/software/ape$ source setenvs.sh (python)cglaser@w10:/net/aera/software/ape$
cd ..
(python)cglaser@w10:/net/aera/software$ cd Offline/build/Modules/
(python)cglaser@w10:/net/aera/software/Offline/build/Modules$ make -j 8
[ 6%] Built target RecEventKG
[ 9%] Built target AnalysisKG
[ 21%] Built target AugerUtilities
```

Use Cases

Collaborative Analysis

- Review and execute a colleague's analysis with just one click
- Joint analyses:
 - ➔ Directly on shared files or through a repository
 - ➔ No separate system setup necessary



University Education

- Data analyses in experimental physics lectures [1]

CERN Outreach

- Public data and example analyses available through web platform



Extend VISPA to your Own Needs

The screenshot displays the VISPA software interface. At the top, there is a navigation bar with a logo, 'Workspaces', and a 'local' status indicator. Below this is a file browser showing '\$HOME' and an open window titled 'Auger Offline'. A toolbar contains buttons for 'submit job', 'open bootstrap.xml', 'open module sequence', 'save XML files', 'save ModuleSequence', 'reset all module options to default', and 'set all infolevel to ...'. The main interface is divided into three panels:

- Left Panel:** A sidebar with expandable categories: 'Steering Symbols', 'All Modules', 'Fd Modules', 'Hybrid Reconstruction Modules', 'Radio Modules', 'SD Reconstruction Modules', and 'SdHAS Modules'. The 'Radio Modules' category is expanded, showing a list of modules including 'RdAntennaChannelToStationConverter', 'RdAntennaStationToChannelConverter', 'RdChannelADCToVoltageConverter', 'RdChannelAmplitudeCalibrator', 'RdChannelBandpassFilter', 'RdChannelBandstopFilter', 'RdChannelBeaconSimulator', 'RdChannelBeaconSuppressor', 'RdChannelBeaconTimingCalibrator', 'RdChannelDebugWriter', 'RdChannelLinearPredictorRFISuppressor', 'RdChannelMedianFilter', and 'RdChannelNoiseASCIIExporter'.
- Center Panel:** Titled 'Module Sequence', it shows a hierarchical list of modules in a sequence. The 'RdChannelResponseIncorporator' module is highlighted with a mouse cursor. The sequence includes: 'try', 'SdHorizontalReconstruction', 'try stop', 'RdEventInitializer', 'RdStationRejector', 'RdChannelADCToVoltageConverter', 'RdChannelSelector', 'RdChannelPedestalRemover', 'RdChannelResponseIncorporator', 'RdChannelBeaconSuppressor', 'RdChannelTimeSeriesTaperer', 'RdChannelBandstopFilter', 'RdChannelUpsampler', 'loop (numTimes=unbounded)', 'RdDirectionConvergenceChecker', 'RdAntennaChannelToStationConverter', 'RdStationSignalReconstructor', 'RdClusterFinder', 'RdPlaneFit', 'loop stop', 'RdLDFMultiFitter', 'RdChannelRiseTimeCalculator', and 'RdStationFieldVectorCalculator'.
- Right Panel:** Titled 'RdChannelResponseIncorporator', it displays configuration options in a table format.

Option	Value	Unit
InfoLevel	1	
ForwardResponseOnFirstCall	0	
InverseUpperFrequencyLimit	80	megahertz
InverseLowerFrequencyLimit	30	megahertz
OverrideForwardBandLimits	0	
NumResponsesToCache	500	
ForwardLowerFrequencyLimit	30	megahertz
OverrideInverseBandLimits	0	
ForwardUpperFrequencyLimit	80	megahertz

- Server/Workspace: Python
- Client: HTML/CSS/Javascript

Extend VISPA to your Own Needs

The screenshot shows the VISPA web interface. At the top, there are navigation elements like 'Workspaces' and 'local'. Below that, a toolbar contains buttons for 'submit job', 'open bootstrap.xml', 'open module sequence', 'save XML files', 'save ModuleSequence', 'reset all module options to default', and 'set all infolevel to ...'. The main area is divided into three panels: 'Steering Symbols', 'Module Sequence', and 'RdChannelResponseIncorporator'. The 'Module Sequence' panel shows a list of modules with up and down arrows, including 'RdChannelBandstopFilter', 'RdChannelUpsampler', 'loop (numTimes=unbounded)', 'RdDirectionConvergenceChecker', 'RdAntennaChannelToStationConverter', 'RdStationSignalReconstructor', 'RdClusterFinder', and 'RdPlaneFit'. The 'RdChannelResponseIncorporator' panel shows a table of options with columns for 'Option', 'Value', and 'Unit'. The 'ForwardUpperFrequencyLimit' option is highlighted with a value of 80 and a unit of megahertz.

Option	Value	Unit
		megahertz
		megahertz
		megahertz
ForwardUpperFrequencyLimit	80	megahertz

Try it out:

- vispa.physik.rwth-aachen.de
- vispa@lists.rwth-aachen.de
- All code open source*

*<https://forge.physik.rwth-aachen.de/projects/vispa-web/repository/v1-0>

SPONSORED BY THE