Interactive applications on HPC systems

Erich Birngruber
(erich.birngruber@gmi.oeaw.ac.at, @ebirn)
Vienna BioCenter

FOSDEM20
Interactive applications on HPC systems

Erich Birngruber
(erich.birngruber@gmi.oeaw.ac.at, @ebirn)
Vienna BioCenter

FOSDEM20
sh$ not good enough?
XPRA

- https://xpra.org/
- “screen for X11”
- Allows disconnect / re-connect to existing X sessions
- Web interface for X11 rendering (HTML5 canvas)
- For arbitrary GUI applications
- Containerized in SLURM
- Custom middleware for job management
Launch XPRA job

Xpra - Run cluster jobs with a web UI

Application
- Fiji

Cores
- 4

Memory (GB)
- 16

Walltime (h)
- 1

GPUs
- 0

Start
### XPRA job submitted

<table>
<thead>
<tr>
<th>ID</th>
<th>Application</th>
<th>Hours</th>
<th>Cores</th>
<th>Memory (GB)</th>
<th>GPUs</th>
<th>State</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>313</td>
<td>Fiji</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>Job running</td>
<td>Join / Settings</td>
</tr>
<tr>
<td>56</td>
<td>X-Term</td>
<td>1</td>
<td>12</td>
<td>10</td>
<td>0</td>
<td>Job finished</td>
<td></td>
</tr>
</tbody>
</table>

Items per page: 5  
1 - 2 of 2
XPRA session
XPRA setup

submit request

IT services

launch job

middleware

connect to xpra client

batch scheduler
• https://rstudio.com/

• IDE for R language

• Desktop and Web version (RStudio server)

• Commercial version for advanced features

• RStudio company has become a public benefit company https://blog.rstudio.com/2020/01/29/rstudio-pbc
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> print("I don't know R")
[1] "I don't know R"
> print(x, ...)
print(x, ...) prints its argument and returns it invisibly (via invisible(x)). It is a generic function which means that new printing methods can be easily added for new classes.
Press F1 for additional help
RStudio setup

RStudio server

batch scheduler

session

connect session

job launcher
• https://galaxyproject.org/

• Web based workflow tool

• Tools as building blocks (parameters, input, output)

• Tool definitions in XML

• Multiple instances: dev - testing - production
Galaxy setup

Git repo branches

- develop
- testing
- production

batch scheduler

session

job

test job
- https://jupyter.org/
- Web-Based IDE (standalone vs. hub)
- Notebooks = Code + Outputs
- Interpreters as “Kernels”
Spawner Options

Job type
CPU short (4c, 16gb, 4h)

Jupyter environment
Environment based on CBE env modules (Python 3.6.6)

Logging
☐ enable logging to $HOME/jupyterhub_[jobid].log

Environment variables (one per line)
MY_VAR=myvalue123

Spawn
```python
from mpl_toolkits.mplot3d import axes3d

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Grab some test data.
X, Y, Z = axes3d.get_test_data(0.05)

# Plot a basic wireframe.
ax.plot_wireframe(X, Y, Z, rstride=10, cstride=10)

# fig.canvas.layout.max_width = '1000px'

plt.show()
```
JupyterHub setup

- JupyterHub
- batch scheduler
- job
- connects
- proxy
- api
- session
Summary

- XPRA
  Special use cases: X11 applications (Fiji) in Containers
- RStudio
  R (from env modules), web-based IDE
- Galaxy
  pre-configured workflows
- JupyterHub
  Python (per-user kernels), plugins
Others

- OpenOnDemand: interactive/remote desktop portal
  https://openondemand.org/

- Apache Zeppelin: data exploration “notebooks”
  https://zeppelin.apache.org/

- Eclipse Che: cloud-based editor
  https://www.eclipse.org/che/
Then this happened
What is wrong?

What’s Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities

Souti Chattopadhyay, Ishita Prasad, Austin Z. Henley, Anita Sarma, Titus Barik
Oregon State University, Microsoft, University of Tennessee-Knoxville
{chattops, anita.sarma}@oregonstate.edu, {ishita.prasad, titus.barik}@microsoft.com, azh@utk.edu

ABSTRACT
Computational notebooks—such as Azure, Databricks, and Jupyter—are a popular, interactive paradigm for data scientists to author code, analyze data, and interleave visualizations, all within a single document. Nevertheless, as data scientists incorporate more of their activities into notebooks, they encounter unexpected difficulties, or pain points, that impact their productivity and disrupt their workflow. Through a systematic, mixed-methods study using semi-structured interviews (n = 20) and survey (n = 156) with data scientists, we catalog nine pain points when working with notebooks. Our findings suggest that data scientists face numerous pain points throughout the entire workflow—from setting up notebooks to deploying to production—across many notebook environments. Our data scientists report essential notebook requirements, such as supporting data exploration and visualization. The results of our study inform and inspire the design of computational notebooks.

Author Keywords
Computational notebooks; challenges; data science; interviews; pain points; survey

CCS Concepts
• Human-centered computing→Interactive systems and development frameworks and environments; • Software and its engineering→Computational notebooks; challenges; data science; empirical studies in HCI; pain points; survey

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI '20, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6708-0/20/04...
http://dx.doi.org/10.1145/3313831.3376729

What’s Wrong with Computational Notebooks?

Pain Points, Needs, and Design Opportunities

Our findings identify that data scientists face considerable pain points through the entire analytics workflow—from setting up the notebook to deploying to production—across many notebook environments. While prior work has studied specific facets of difficulties in notebooks [24, 17], such as versioning [18, 19] or cleaning unused code [13, 34], the central contribution of this paper is a taxonomy of validated pain points across data scientists’ notebook activities.
References

• XPRA https://xpra.org/

• RStudio https://rstudio.com/

• Jupyterhub https://jupyter.org/hub

• Galaxy https://galaxyproject.org/

• What is wrong with computational notebooks?
  http://web.eecs.utk.edu/~azh/blog/notebookpainpoints.html