AGU Advances

Supporting Information for

Science Storms the Cloud

C. L. Gentemann[1,2], C. Holdgraf[3,4], R. Abernathey[3,5], D. Crichton[6], J. Colliander[3,7,8], E. J. Kearns[9], Y. Panda[3], R. P. Signell[10]

[1]Farallon Institute, Petaluma, CA, [2]Earth and Space Research, Seattle, WA, [3]2i2c, Berkeley, CA, [4]International Computer Science Institute, Berkeley, CA, [5]Lamont Doherty Earth Observatory of Columbia University, Palisades, NY, [6]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, [7]Pacific Institute for the Mathematical Sciences, Vancouver, BC, Canada, [8]University of British Columbia, Vancouver, BC, Canada, [9]First Street Foundation, Brooklyn, NY, [10]US Geological Survey, Woods Hole, MA

Contents of this file

Introduction

The supporting information contains links to open data repositories, and other examples referenced in the paper, a discussion of other potential challenges to cloud-based science, and some simple guidance to get started:

Text S1.

Public cloud-based open data:
Amazon Web Services: https://registry.opendata.aws/
Google Cloud: https://cloud.google.com/public-datasets
Microsoft: https://azure.microsoft.com/en-us/services/open-datasets/

Text S2.

Examples using Binder for tutorials or teaching:
https://mybinder.readthedocs.io/en/latest/examples/examples.html
https://www.inferentialthinking.com/chapters/intro
https://gallery.pangeo.io/
https://github.com/fangohr/introduction-to-python-for-computational-science-and-engineering
https://github.com/jgomezdans/accra_wkshp

Examples using Binder for reproducible science in peer-reviewed publications:
There are 129 examples at google scholar search:
https://scholar.google.com/scholar?start=20&q=mybinder.org/v2

Some examples:
https://github.com/cgentemann/2020_FluxSat_MDPI_RemoteSensing
https://github.com/sjvrijn/cma-es-configuration-data-mining
https://github.com/martibosch/swiss-urbanization
https://github.com/johnjarmitage/flem
https://github.com/cboettig/noise-phenomena
https://github.com/LiYingWang/kwl.pottery
https://github.com/davidcortesortuno/paper-2019_nanoscale_skyrmions_target_states_confined_geometries

Instructions to creating Binder interactive notebooks:
https://mybinder.readthedocs.io/en/latest/introduction.html
https://earth-env-data-science.github.io/lectures/environment/binder.html

Text S3.

Additional discussion of challenges:

Vendor Lock-in. As we utilize the cloud, we must recognize that we are using a company's infrastructure. These companies have incentives that may mis-align with the goals of science. Possible Solution: Prioritize using and supporting open source tools that are governed and developed by diverse, multi-stakeholder communities. Support organizations that make it easier to use these tools in a way that minimizes vendor lock-in. Put pressure on cloud companies, universities, and funding agencies to prioritize support of these communities.

Software and Data Standards. Leveraging similar cloud infrastructure provides an opportunity for diverse scientific communities to adopt scientific software that is largely overlapping (the Pydata ecosystem is an example of this success). However, doing so will require some re-tooling in many pre-existing stacks, particularly around data standards and scripts. This will require both new efforts in infrastructure development as well as massive educational efforts to teach people the skills to work cloud-natively. Possible Solution: see above - support diverse, multi-stakeholder communities that define and oversee standards in data specifications and formats, as well as tools that interact with data. Agree to build tooling around these standards, rather than building institution- or field-specific toolchains.

Data inertia. Data can be expensive and hard to move. Once a cloud provider has your data, they have a lot of potential leverage. Moreover, processing, munging, uploading, and accessing data is often a labor-intensive process (with more or less pain depending on the original format and structure of the data). This can be a big barrier to scientists that want to do their work in the cloud, but are not equipped with the skills, tools, or support to do this efficiently and in a way that avoids the challenges laid out above. Possible Solution: Put pressure on cloud providers to minimize the costs associated with egress (taking data out of the cloud), in combination with pressure to support "cloud-agnostic" stacks as first-class citizens (or to partner with organizations that are cloud-agnostic). Put institutional support into open source communities that build infrastructure to facilitate the processing and distribution of (potentially large) datasets for cloud use. Invest in training opportunities for scientists who wish to do their work in the cloud utilizing this open source stack.

Institutional administration for cloud infrastructure. Computing infrastructure has traditionally been centrally procured by a research institution, which pays for the hardware and full-time employees to maintain that hardware and provide services upon it. Cloud infrastructure allows us to outsource all of that labor to a hardware stack that is much more flexible, agile, and cutting-edge. However, institutional purchasing processes often introduce artificial barriers to making this switch, such as charging full indirect costs on payments for

cloud infrastructure. This passes extra costs to researchers (instead of being borne centrally by the institution, as is done with local infrastructure). Possible Solution: Build communities of practice for institutional administrators that wish to facilitate the use of cloud infrastructure. Support the creation of strategic plans for how institutions can transition their infrastructure approach from "fully centralized" to a "hybrid local+cloud" model.

Text S4.

Code of conduct examples:
https://numfocus.org/code-of-conduct
https://ropensci.org/code-of-conduct/

Text S5.

Introductions to open science and open project design
https://www.openscapes.org/
https://the-turing-way.netlify.app/welcome.html
https://mozillascience.github.io/open-science-leadership-workshop/index.html

Simple Guidance to get started:

Learn Python or R and how to use GitHub. There are many online tutorials and in-person classes. We provide some links below. When you get stuck, Google or https://stackoverflow.com/ are very helpful. Twitter has a very active open science community that is also a useful resource. Providing links to your software in publications (https://zenodo.org/). If you don't currently have access to a cloud-based JupyterHub, the article lists several companies that can provide them.

Skills for scientific computing:
https://carpentries.org/
https://guides.github.com/
https://scipy-lectures.org/intro/

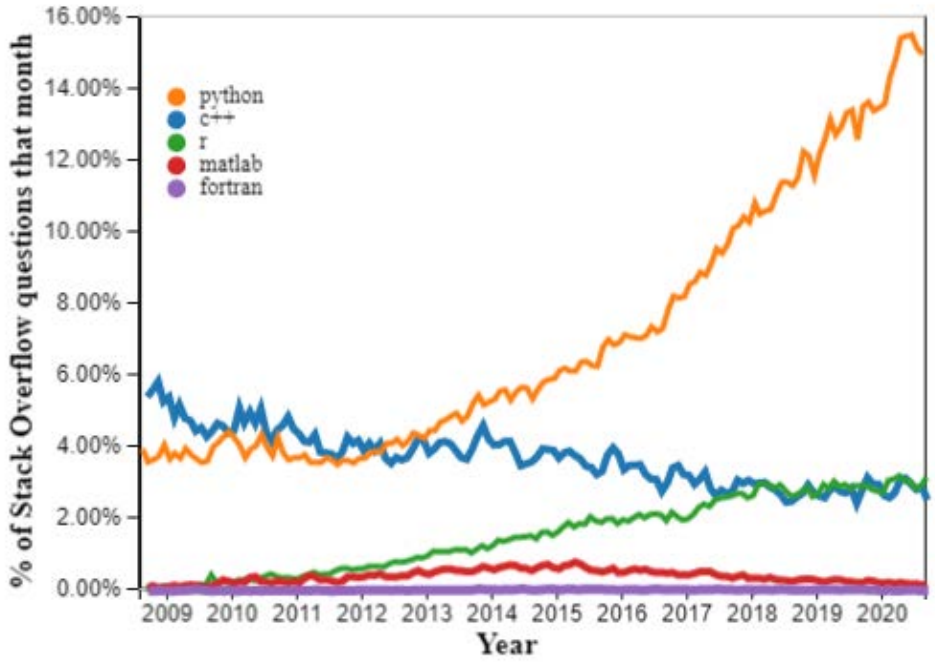

Figure S1. The popularity and growth (decay) of common scientific programming languages as measured by the number of tags for each language on the website Stack Overflow (https://tinyurl.com/fig1gentemann).