FAIR ASSESSMENT OF CLOUD-BASED EXPERIMENTS: AN EXPERIENCE WITH CHAMELEON ARTIFACTS

Krishna Kamath[a], Nicole Brewer[b],
Tanu Malik[a,c]
[a]School of Computing, DePaul University, Chicago, IL, USA
[c]Department of Electrical Engineering and Computer Science,
Univ. of Missouri, Columbia
[b]Center for Biology and Society, Arizona State University, Tempe
AZ USA

Introduction

- **Reproducibility** is essential to validate scientific research
  - *Experiments need to be validated*
- Cloud testbeds provide consistent hardware to simplify experiment reproducibility
  - *Provide consistent hardware*
- **But are the experiments hosted on cloud-based testbeds actually reproducible?**

Problem Statement

- FAIR: Findable, Accessible, Interoperable, Reusable
- Are FAIR guiding principles useful to test reproducibility of cloud testbed experiments?
- Analysis of cloud testbed experiments
  - *Are they reproducible?*
  - *What are some common issues when trying to reproduce experiments?*
- Are there recommendations to authors to improve reproducibility across cloud platforms?

Problem Statement

- FAIR: Findable, Accessible, Interoperable, Reusable
- Are FAIR guiding principles useful to test reproducibility of cloud testbed experiments?
- Analysis of cloud testbed experiments
  - *Are they reproducible?*
  - *What are some common issues when trying to reproduce experiments?*
- Are there recommendations to authors to improve reproducibility across cloud platforms?

FAIR Definitions

- Findable
  - *(meta)data are assigned a globally unique and persistent identifier*
  - *data are described with rich metadata*
  - *metadata clearly include the identifier of the data it describes*
  - *(meta)data are registered or indexed in a searchable resource*
- Accessible
  - *(meta)data are retrievable by their identifier using a standardized communications protocol*
  - *open, free, and universally implementable*
  - *protocol allows for authentication and authorization procedure*

Wilkinson, M., Dumontier, M., Aalbersberg, I. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data*

FAIR Definitions

- Interoperable
  - *(meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.*
  - *(meta)data use vocabularies that follow FAIR principles*
  - *(meta)data include qualified references to other (meta)data*
- Reusable
  - *meta(data) are richly described with a plurality of accurate and relevant attributes*
  - *(meta)data are released with a clear and accessible data usage license*
  - *(meta)data are associated with detailed provenance*
  - *(meta)data meet domain-relevant community standards*

Wilkinson, M., Dumontier, M., Aalbersberg, I. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data*

FAIRness for Cloud-based Experiments

- Cloud experiments must maintain continuous FAIRness
- Findable artifacts may be incomplete, i.e., some of the files are missing
- FAIR definitions are related to data & metadata; not to artifact execution

FAIRness for Cloud-based Experiments

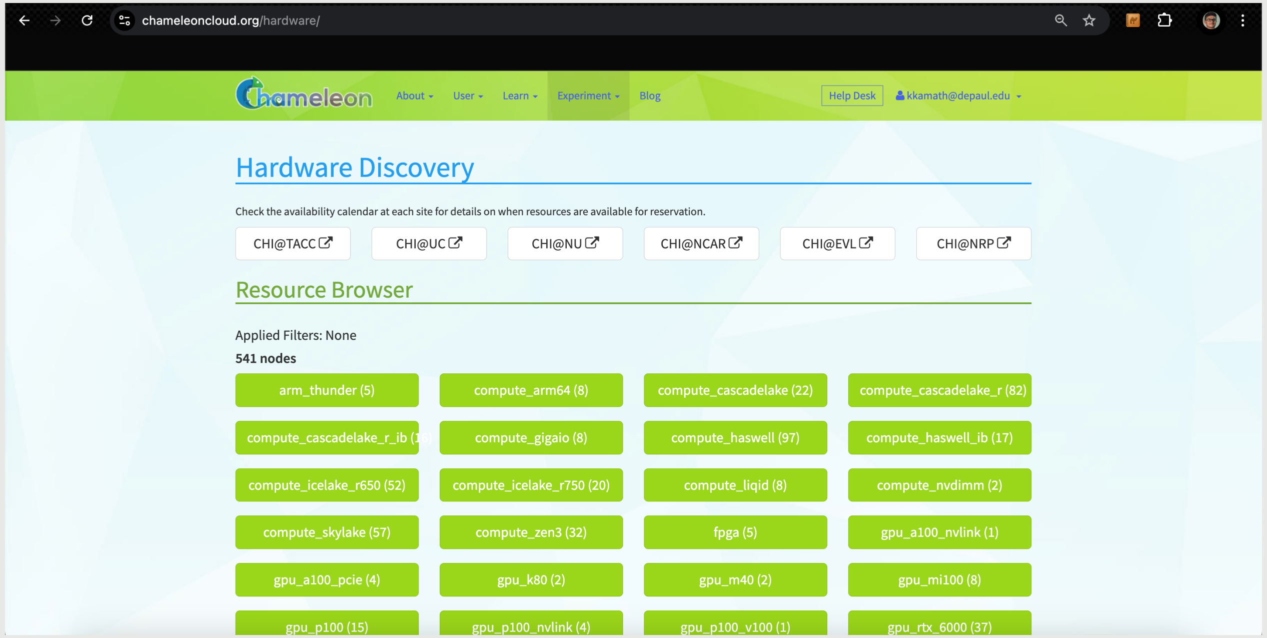
- Findable
  - *associated with a persistent URI that results in all the necessary files and data for a successful experiment run*
- Accessible
  - *necessary found files result in a successful execution and produces consistent results.*
- Interoperable
  - *if its bundled package can be successfully executed to generate standardized metadata about its execution*
- Reusable
  - *An artifact is reusable if it can be successfully executed in an alternate environment*

Problem Statement

- Are FAIR guiding principles useful to test reproducibility of cloud testbed experiments?
- Analysis of cloud testbed experiments
  - *Are they reproducible?*
  - *What are some common issues when trying to reproduce experiments?*
- Are there recommendations to authors to improve reproducibility across cloud platforms?

Chameleon

- Cloud testbed
- Features
  - *bare-metal reconfigurability*
  - *full control of the software stack*
  - *Specialized hardware*



Hardware Discovery

Check the availability calendar at each site for details on when resources are available for reservation.

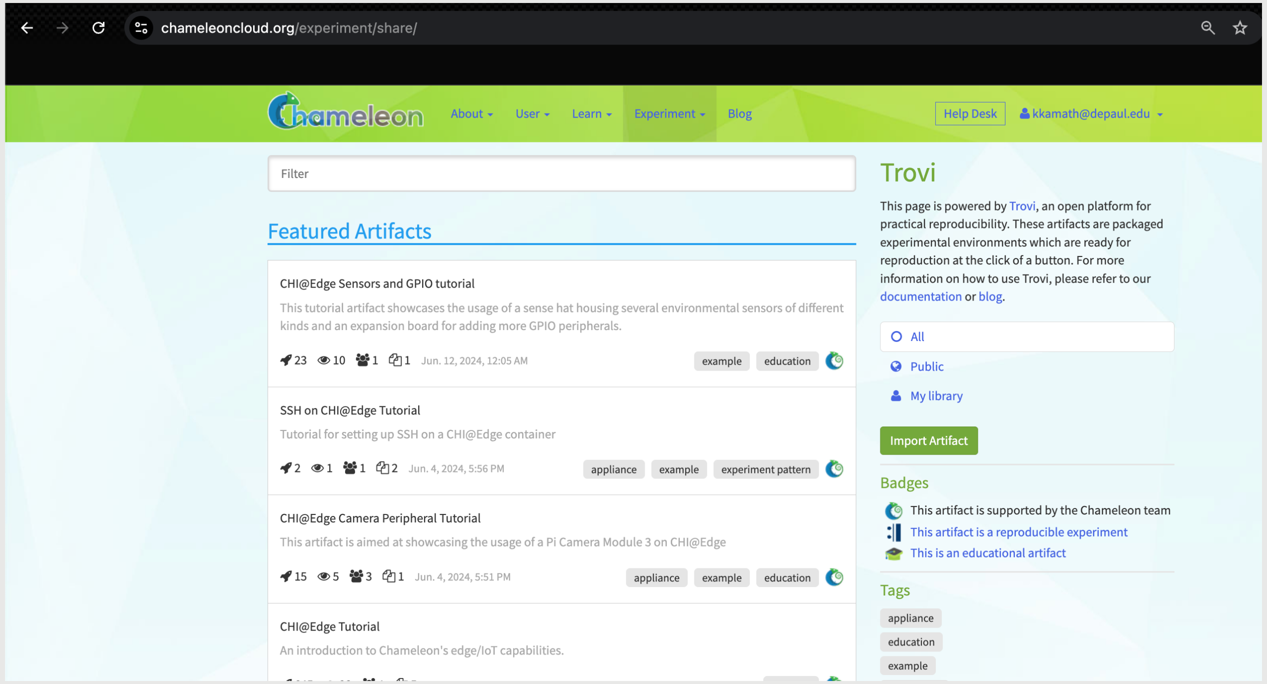
Resource Browser

Applied Filters: None

541 nodes

Trovi

- Portal to share & replay artifacts
- ~170 artifacts



chameleoncloud.org/experiment/share/

Featured Artifacts

CHI@Edge Sensors and GPIO tutorial

This tutorial artifact showcases the usage of a sense hat housing several environmental sensors of different kinds and an expansion board for adding more GPIO peripherals.

23 10 1 1 Jun. 12, 2024, 12:05 AM example education

SSH on CHI@Edge Tutorial

Tutorial for setting up SSH on a CHI@Edge container

2 1 1 2 Jun. 4, 2024, 5:56 PM appliance example experiment pattern

CHI@Edge Camera Peripheral Tutorial

This artifact is aimed at showcasing the usage of a Pi Camera Module 3 on CHI@Edge

15 5 3 1 Jun. 4, 2024, 5:51 PM appliance example education

CHI@Edge Tutorial

An introduction to Chameleon's edge/IoT capabilities.

Trovi

This page is powered by Trovi, an open platform for practical reproducibility. These artifacts are packaged experimental environments which are ready for reproduction at the click of a button. For more information on how to use Trovi, please refer to our documentation or blog.

Badges

This artifact is supported by the Chameleon team

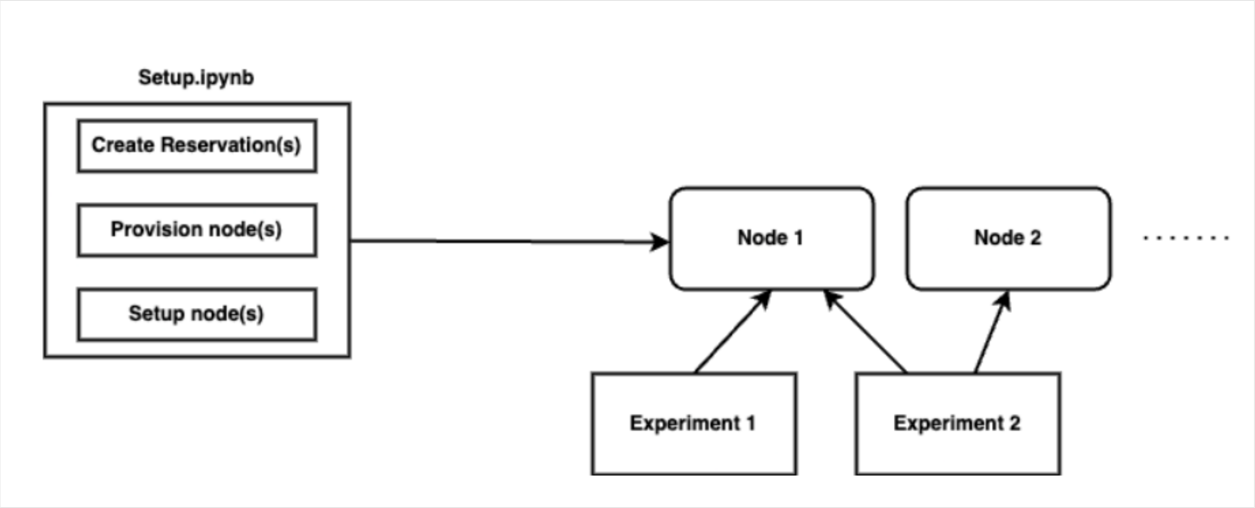
This artifact is a reproducible experiment

This is an educational artifact

Tags

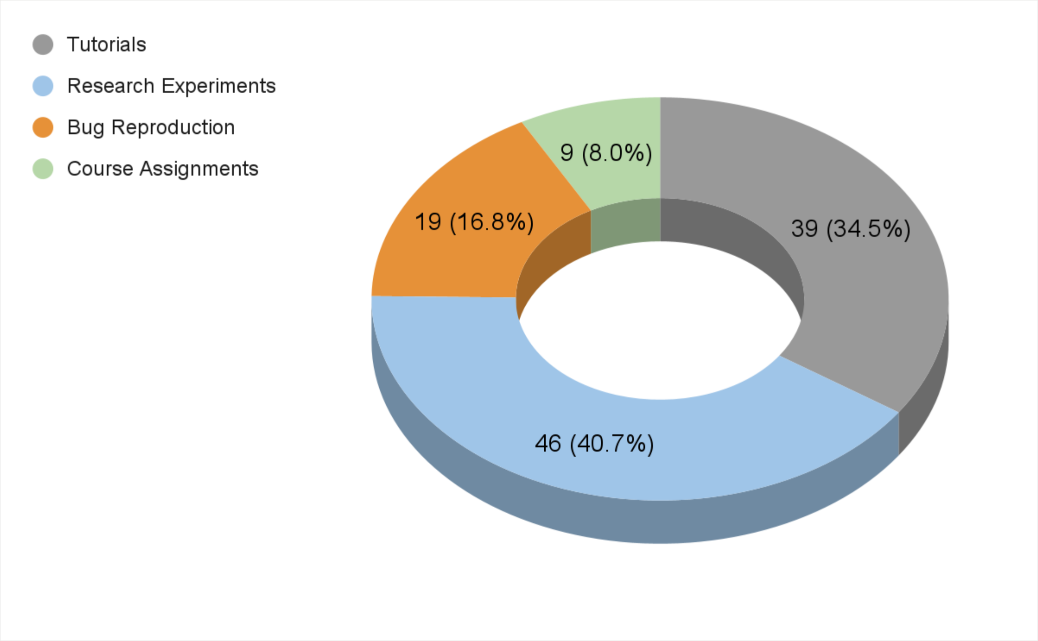
Generic Chameleon Artifact Template

- Most artifacts contain these 3 steps:
  - *System setup*
  - *Performing the experiment*
  - *Conduct the analysis*



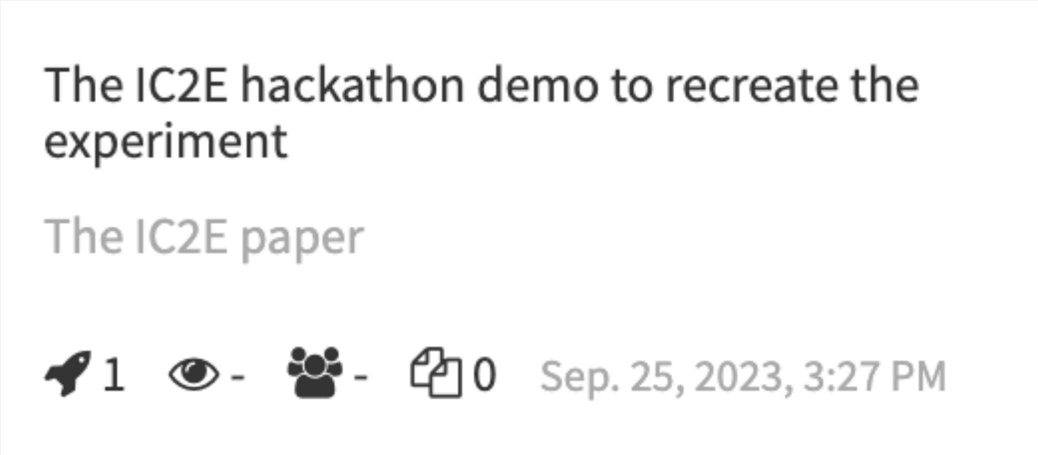
Data Collection and Artifact Types

- Chameleon artifacts
  - *113 artifacts*
  - *Ranging from Apr 2022 – Dec 2023*
- Types of artifacts
  - *Tutorials*
  - *Research experiments*
  - *Bug reproduction*
  - *Course Assignments*



Chameleon Artifact Findability

- 104/113 artifacts had all the files completely
- 9 were not findable



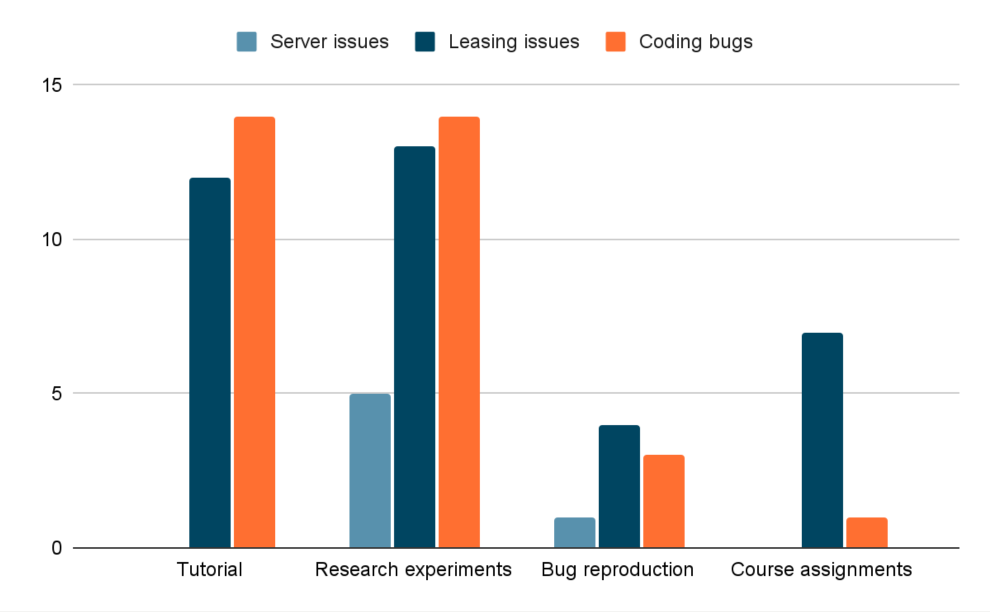
The IC2E hackathon demo to recreate the experiment

The IC2E paper

🚀 1 👁 - 👥 - ⧉ 0 Sep. 25, 2023, 3:27 PM

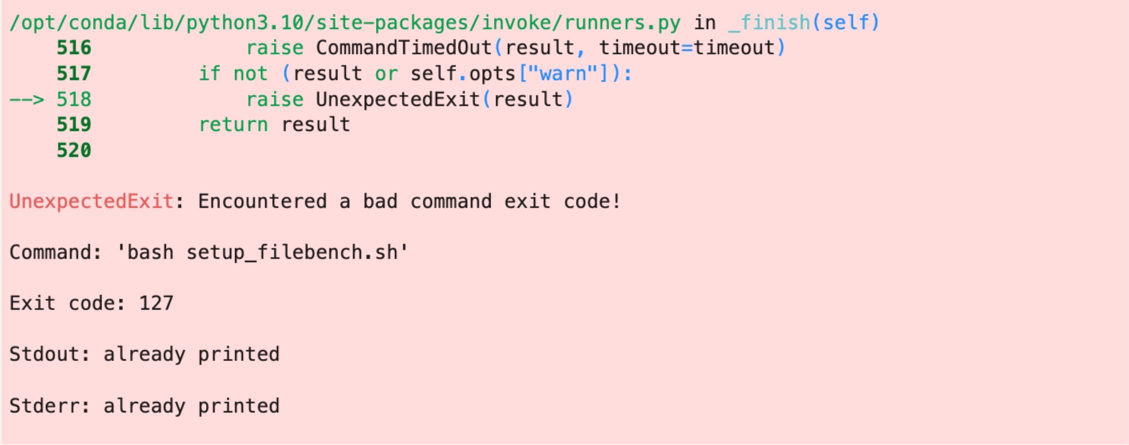
Chameleon Artifact Accessibility

- Out of 104 experiments, we were able to repeat 30 successfully
- Common issues:
  - *Code bugs*
  - *Leasing issues*
  - *Server Issues*



Chameleon Artifact Accessibility

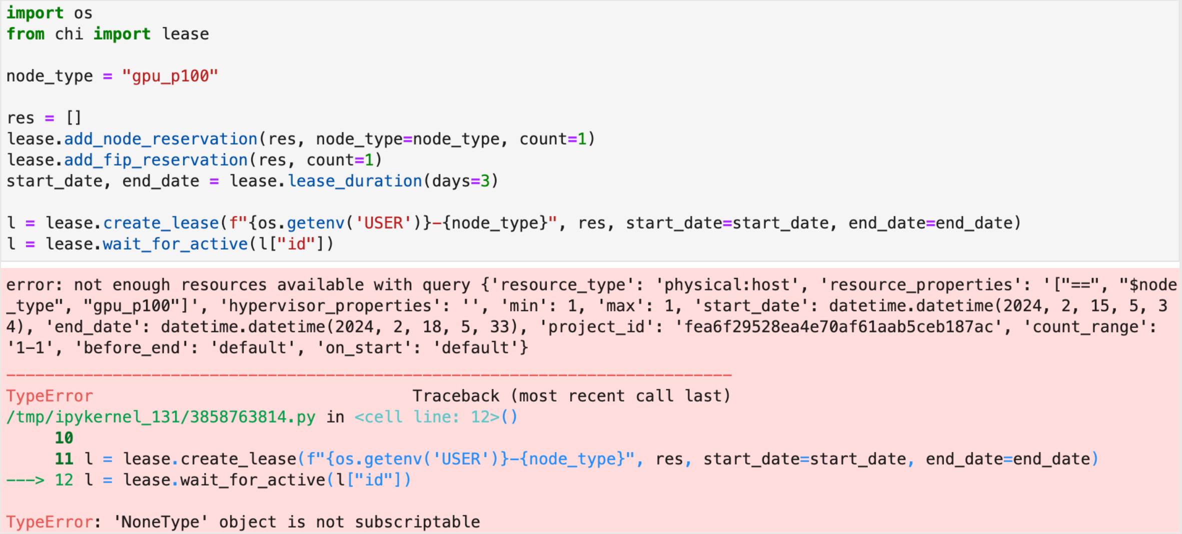
- Out of 104 experiments, we were able to repeat 30 successfully
- Common issues:
  - *Code bugs*
  - *Leasing issues*
  - *Server Issues*



```
/opt/conda/lib/python3.10/site-packages/invoke/runners.py in _finish(self)
    516             raise CommandTimedOut(result, timeout=timeout)
    517         if not (result or self.opts["warn"]):
--> 518             raise UnexpectedExit(result)
    519         return result
    520 

UnexpectedExit: Encountered a bad command exit code!

Command: 'bash setup_filebench.sh'

Exit code: 127

Stdout: already printed

Stderr: already printed
```

Chameleon Artifact Accessibility

- Out of 104 experiments, we were able to repeat 30 successfully
- Common issues:
  - *Code bugs*
  - *Leasing issues*
  - *Server Issues*

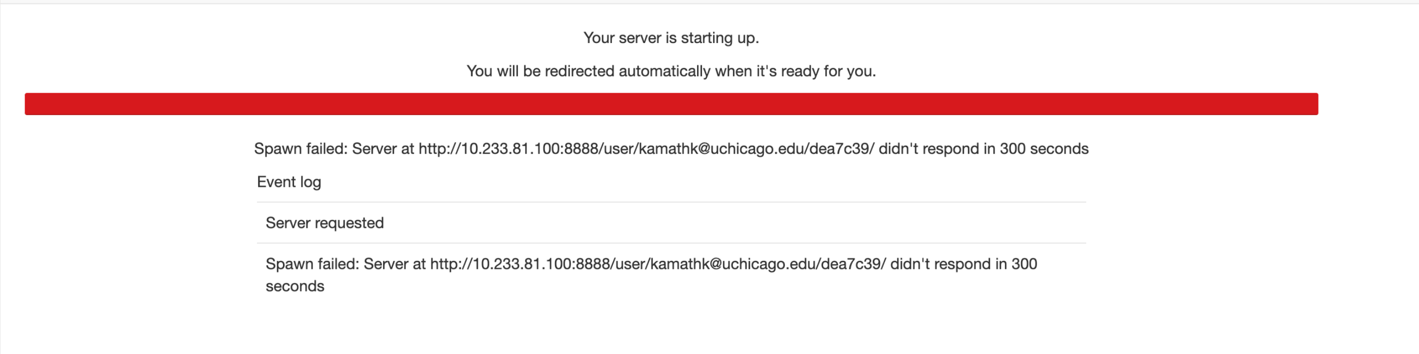


```
import os
from chi import lease

node_type = "gpu_p100"

res = []
lease.add_node_reservation(res, node_type=node_type, count=1)
lease.add_fip_reservation(res, count=1)
start_date, end_date = lease.lease_duration(days=3)

l = lease.create_lease(f"{os.getenv('USER')}-{node_type}", res, start_date=start_date, end_date=end_date)
l = lease.wait_for_active(l["id"])
```

```
error: not enough resources available with query {'resource_type': 'physical:host', 'resource_properties': '["==", "$node
_type", "gpu_p100"]', 'hypervisor_properties': '', 'min': 1, 'max': 1, 'start_date': datetime.datetime(2024, 2, 15, 5, 3
4), 'end_date': datetime.datetime(2024, 2, 18, 5, 33), 'project_id': 'fea6f29528ea4e70af61aab5ceb187ac', 'count_range':
'1-1', 'before_end': 'default', 'on_start': 'default'}

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipykernel_131/3858763814.py in <cell line: 12>()
     10
     11 l = lease.create_lease(f"{os.getenv('USER')}-{node_type}", res, start_date=start_date, end_date=end_date)
---> 12 l = lease.wait_for_active(l["id"])

TypeError: 'NoneType' object is not subscriptable
```

Chameleon Artifact Accessibility

- Out of 104 experiments, we were able to repeat 30 successfully
- Common issues:
  - *Code bugs*
  - *Leasing issues*
  - *Server Issues*



Your server is starting up.

You will be redirected automatically when it's ready for you.

Spawn failed: Server at http://10.233.81.100:8888/user/kamathk@uchicago.edu/dea7c39/ didn't respond in 300 seconds

Event log

Server requested

Spawn failed: Server at http://10.233.81.100:8888/user/kamathk@uchicago.edu/dea7c39/ didn't respond in 300 seconds

Chameleon Artifact Interoperability

- Package experiments with application virtualization (AV) tools
  - *Sciunit, Reprozip*
- AV tools:
  - *Intercept system calls to determine files used*
  - *Audit and repeat mode*
- 18 out of the 30 experiments were packaged, and could be repeated on Chameleon
- 12 of the artifacts had:
  - *Leasing issues*
  - *packaging issues*

Chameleon Artifact Reusability

- Determine if the package is runnable on a public cloud infrastructure (AWS)
- 5 experiments were reusable on different cloud platforms
  - *artifacts which didn’t need specialized hardware*

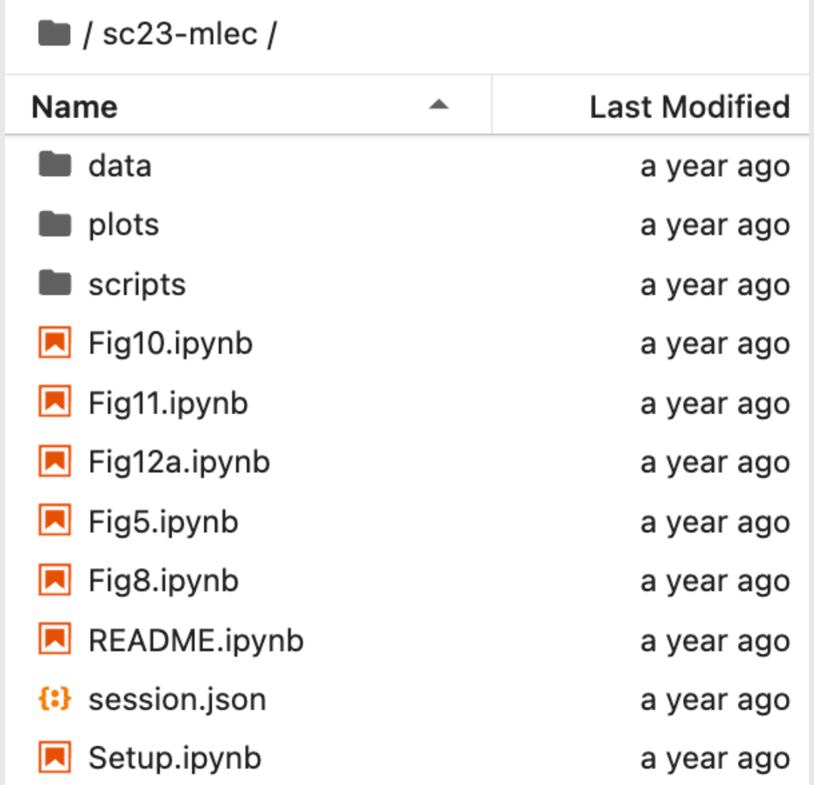
Recommendations

Recommendations

- Organize notebooks by purpose
- Ensure Long-term Preservation
- Install Dependencies During Setup
- Automate instructions
- Reduce Calls to Cloud-Specific APIs
- Indicate estimated wait times

Recommendation 1

- Organize Notebooks by Purpose
  - *Setup should be different from experiment*
  - *Break large experiments into multiple notebooks*



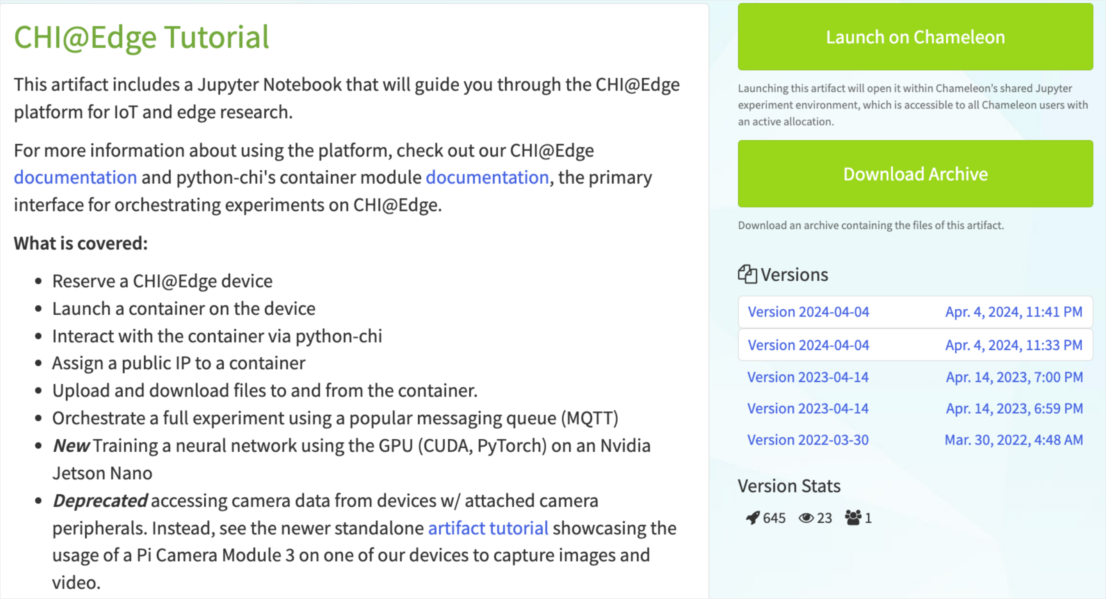
/ sc23-mlec /

Name	Last Modified
data	a year ago
plots	a year ago
scripts	a year ago
Fig10.ipynb	a year ago
Fig11.ipynb	a year ago
Fig12a.ipynb	a year ago
Fig5.ipynb	a year ago
Fig8.ipynb	a year ago
README.ipynb	a year ago
session.json	a year ago
Setup.ipynb	a year ago

Wang, Meng, et al. "Design Considerations and Analysis of Multi-Level Erasure Coding in Large-Scale Data Centers."

Recommendation 2

- Ensure long term preservation
  - *Most open-source software decays without maintenance*
  - *frequently try repeating and updating artifacts*



CHI@Edge Tutorial

This artifact includes a Jupyter Notebook that will guide you through the CHI@Edge platform for IoT and edge research.

For more information about using the platform, check out our CHI@Edge documentation and python-chi's container module documentation, the primary interface for orchestrating experiments on CHI@Edge.

What is covered:

- Reserve a CHI@Edge device
- Launch a container on the device
- Interact with the container via python-chi
- Assign a public IP to a container
- Upload and download files to and from the container.
- Orchestrate a full experiment using a popular messaging queue (MQTT)
- ***New*** Training a neural network using the GPU (CUDA, PyTorch) on an Nvidia Jetson Nano
- ***Deprecated*** accessing camera data from devices w/ attached camera peripherals. Instead, see the newer standalone artifact tutorial showcasing the usage of a Pi Camera Module 3 on one of our devices to capture images and video.

Launching this artifact will open it within Chameleon's shared Jupyter experiment environment, which is accessible to all Chameleon users with an active allocation.

Download an archive containing the files of this artifact.

Versions

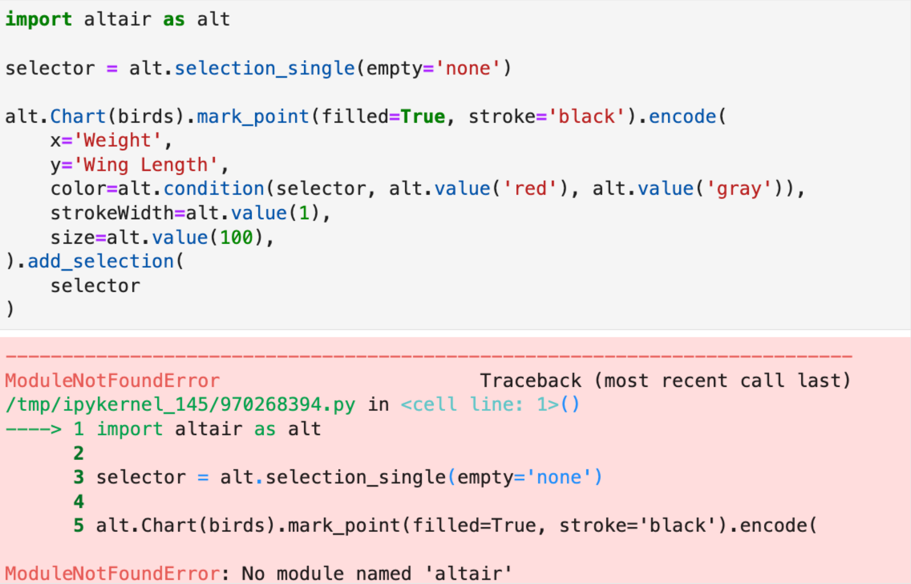
Version 2024-04-04	Apr. 4, 2024, 11:41 PM
Version 2024-04-04	Apr. 4, 2024, 11:33 PM
Version 2023-04-14	Apr. 14, 2023, 7:00 PM
Version 2023-04-14	Apr. 14, 2023, 6:59 PM
Version 2022-03-30	Mar. 30, 2022, 4:48 AM

Version Stats

🚀 645 👁 23 👥 1

Recommendation 3

- Install dependencies during setup

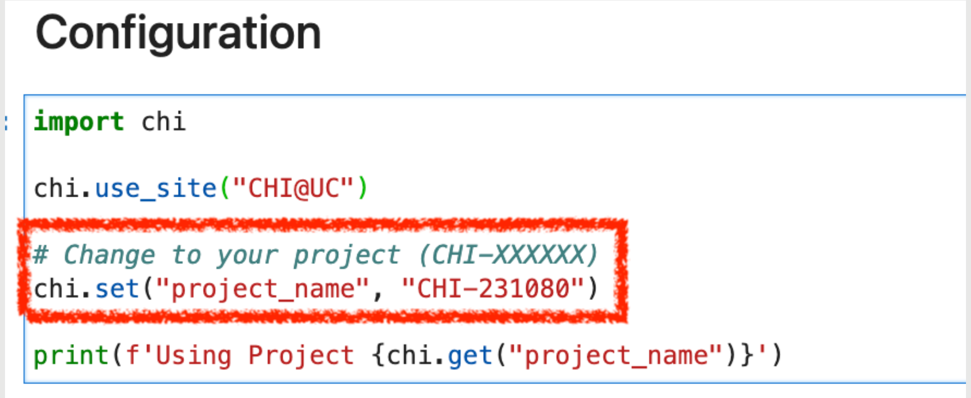


```
import altair as alt

selector = alt.selection_single(empty='none')

alt.Chart(birds).mark_point(filled=True, stroke='black').encode(
    x='Weight',
    y='Wing Length',
    color=alt.condition(selector, alt.value('red'), alt.value('gray')),
    strokeWidth=alt.value(1),
    size=alt.value(100),
).add_selection(
    selector
)
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
/tmp/ipykernel_145/970268394.py in <cell line: 1>()
----> 1 import altair as alt
      2 
      3 selector = alt.selection_single(empty='none')
      4 
      5 alt.Chart(birds).mark_point(filled=True, stroke='black').encode(

ModuleNotFoundError: No module named 'altair'
```

Recommendation 4

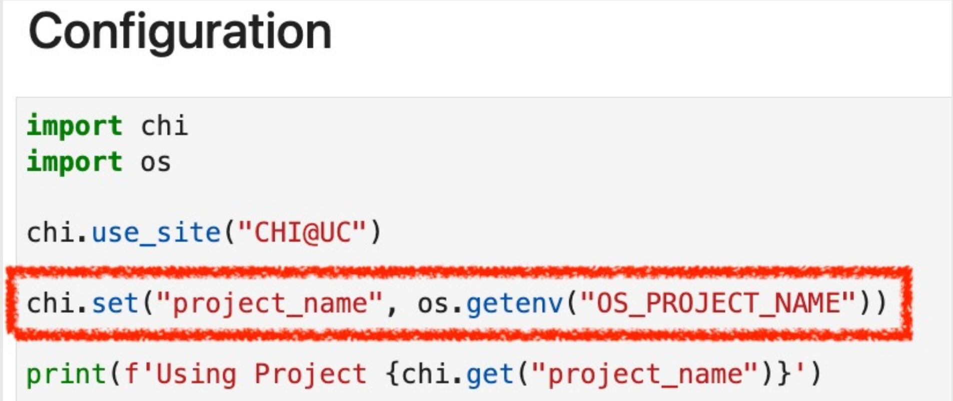
- Automate instructions
  - *Use environment variables*
  - *Include tear-down commands*



Configuration

```
import chi

chi.use_site("CHI@UC")

# Change to your project (CHI-XXXXXX)
chi.set("project_name", "CHI-231080")

print(f'Using Project {chi.get("project_name")}')
```

Requires manual changes



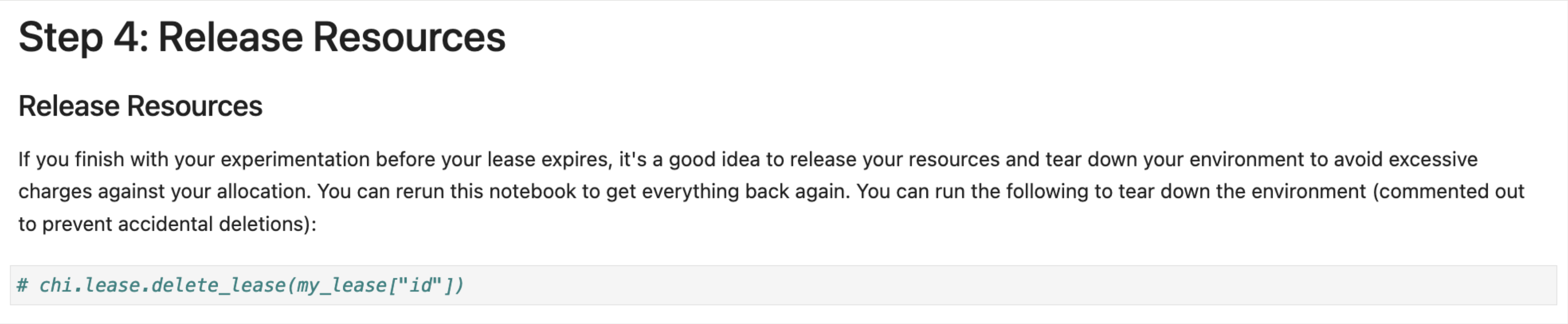
Configuration

```
import chi
import os

chi.use_site("CHI@UC")

chi.set("project_name", os.getenv("OS_PROJECT_NAME"))

print(f'Using Project {chi.get("project_name")}')
```

Using env vars

Recommendation 4

- Automate instructions
  - *Use environment variables*
  - *Include tear-down commands*



Step 4: Release Resources

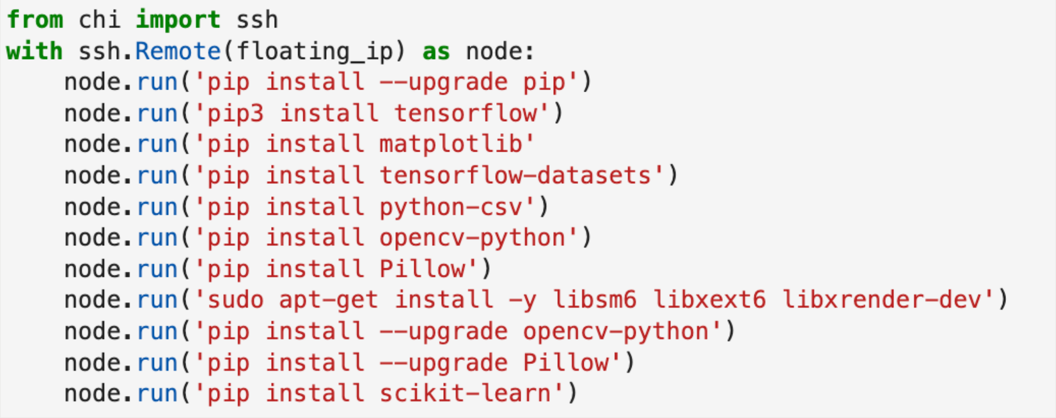
Release Resources

If you finish with your experimentation before your lease expires, it's a good idea to release your resources and tear down your environment to avoid excessive charges against your allocation. You can rerun this notebook to get everything back again. You can run the following to tear down the environment (commented out to prevent accidental deletions):

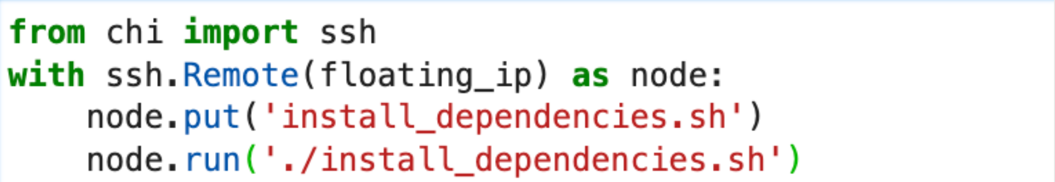
```
# chi.lease.delete_lease(my_lease["id"])
```

Recommendation 5

- Reduce Calls to Cloud-Specific API



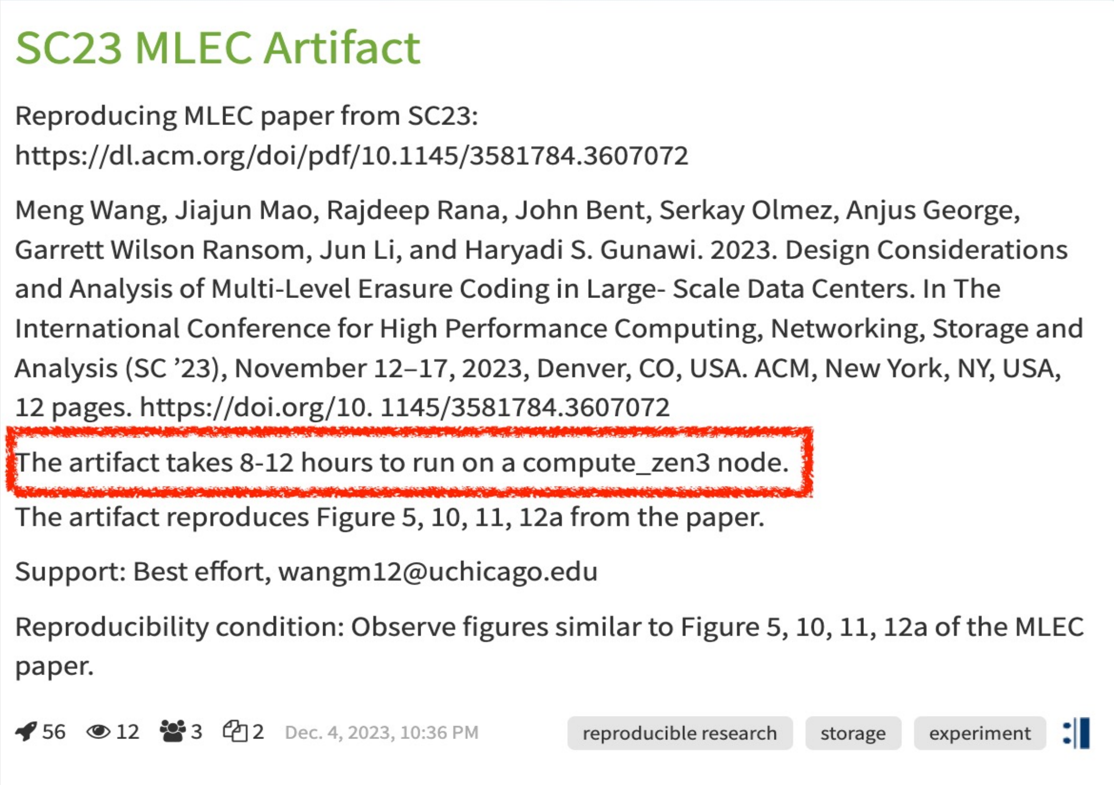
```
from chi import ssh
with ssh.Remote(floating_ip) as node:
    node.run('pip install --upgrade pip')
    node.run('pip3 install tensorflow')
    node.run('pip install matplotlib'
    node.run('pip install tensorflow-datasets')
    node.run('pip install python-csv')
    node.run('pip install opencv-python')
    node.run('pip install Pillow')
    node.run('sudo apt-get install -y libsm6 libxext6 libxrender-dev')
    node.run('pip install --upgrade opencv-python')
    node.run('pip install --upgrade Pillow')
    node.run('pip install scikit-learn')
```



```
from chi import ssh
with ssh.Remote(floating_ip) as node:
    node.put('install_dependencies.sh')
    node.run('./install_dependencies.sh')
```

Recommendation 6

- Indicate wait times
  - *Authors have tacit understanding of experiment duration; but maybe the user reproducing the experiment doesn't*



SC23 MLEC Artifact

Reproducing MLEC paper from SC23: https://dl.acm.org/doi/pdf/10.1145/3581784.3607072

Meng Wang, Jiajun Mao, Rajdeep Rana, John Bent, Serkay Olmez, Anjus George, Garrett Wilson Ransom, Jun Li, and Haryadi S. Gunawi. 2023. Design Considerations and Analysis of Multi-Level Erasure Coding in Large- Scale Data Centers. In The International Conference for High Performance Computing, Networking, Storage and Analysis (SC '23), November 12–17, 2023, Denver, CO, USA. ACM, New York, NY, USA, 12 pages. https://doi.org/10. 1145/3581784.3607072

The artifact takes 8-12 hours to run on a compute_zen3 node.

The artifact reproduces Figure 5, 10, 11, 12a from the paper.

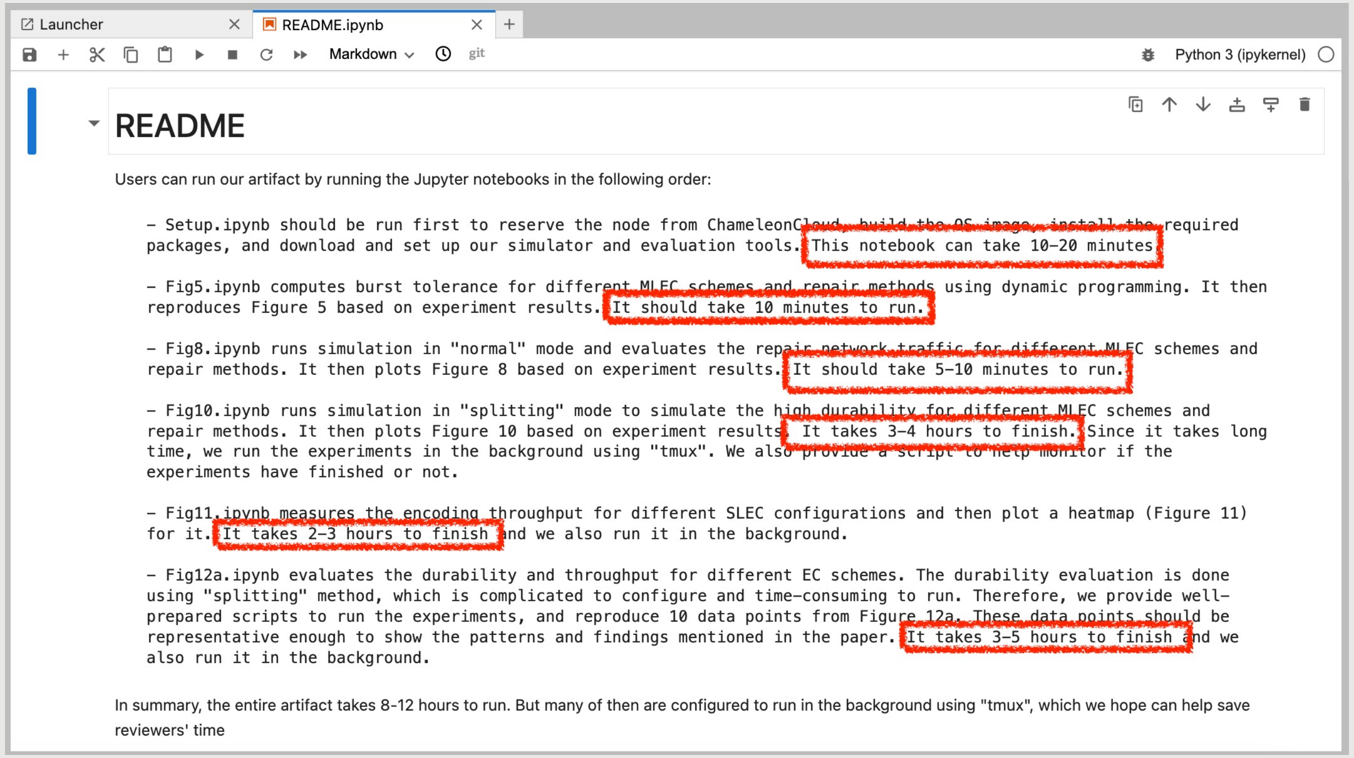
Support: Best effort, wangm12@uchicago.edu

Reproducibility condition: Observe figures similar to Figure 5, 10, 11, 12a of the MLEC paper.

Wang, Meng, et al. "Design Considerations and Analysis of Multi-Level Erasure Coding in Large-Scale Data Centers."

Recommendation 6

- Indicate wait times
  - *For larger experiments, including wait times for individual experiments is a good idea*



README

Users can run our artifact by running the Jupyter notebooks in the following order:

```
- Setup.ipynb should be run first to reserve the node from ChameleonCloud, build the OS image, install the required
packages, and download and set up our simulator and evaluation tools. This notebook can take 10-20 minutes

- Fig5.ipynb computes burst tolerance for different MLEC schemes and repair methods using dynamic programming. It then
reproduces Figure 5 based on experiment results. It should take 10 minutes to run.

- Fig8.ipynb runs simulation in "normal" mode and evaluates the repair network traffic for different MLEC schemes and
repair methods. It then plots Figure 8 based on experiment results. It should take 5-10 minutes to run.

- Fig10.ipynb runs simulation in "splitting" mode to simulate the high durability for different MLEC schemes and
repair methods. It then plots Figure 10 based on experiment results. It takes 3-4 hours to finish. Since it takes long
time, we run the experiments in the background using "tmux". We also provide a script to help monitor if the
experiments have finished or not.

- Fig11.ipynb measures the encoding throughput for different SLEC configurations and then plot a heatmap (Figure 11)
for it. It takes 2-3 hours to finish and we also run it in the background.

- Fig12a.ipynb evaluates the durability and throughput for different EC schemes. The durability evaluation is done
using "splitting" method, which is complicated to configure and time-consuming to run. Therefore, we provide well-
prepared scripts to run the experiments, and reproduce 10 data points from Figure 12a. These data points should be
representative enough to show the patterns and findings mentioned in the paper. It takes 3-5 hours to finish and we
also run it in the background.
```

In summary, the entire artifact takes 8-12 hours to run. But many of then are configured to run in the background using "tmux", which we hope can help save reviewers' time

Wang, Meng, et al. "Design Considerations and Analysis of Multi-Level Erasure Coding in Large-Scale Data Centers."

Recommendations

- Organize notebooks by purpose
- Ensure Long-term Preservation
- Install Dependencies During Setup
- Automate instructions
- Reduce Calls to Cloud-Specific APIs
- Indicate estimated wait times

Conclusions

- Trovi is great! We need to do better.
  - *Reusability of cloud-based experiments is difficult to achieve in practice*
- FAIR definitions must be redefined for cloud-based experiments
- Why is this the case?
  - *Lack of maintainence*
  - *Leasing issues*
- Recommendations
  - *Organize notebooks by purpose*
  - *Ensure Long-term Preservation*
  - *Install Dependencies During Setup*
  - *Automate instructions*
  - *Reduce Calls to Cloud-Specific APIs*
  - *Indicate estimated wait times*

Open Post-doc Position

The Department of Electrical Engineering and Computer Science in the College of Engineering at University of Missouri, Columbia is looking for talented and motivated postdoctoral fellows to become part of our team working on exciting science research projects.

Researchers working at the intersection of systems, workflows, data management, and high-performance computing or any relevant data and computational science discipline, and who have received their Ph.D. within the last three years are encouraged to apply. The successful applicant will receive a competitive salary, and excellent benefits. **The position is for up to four years beginning anytime after Jan 1st, 2025.**

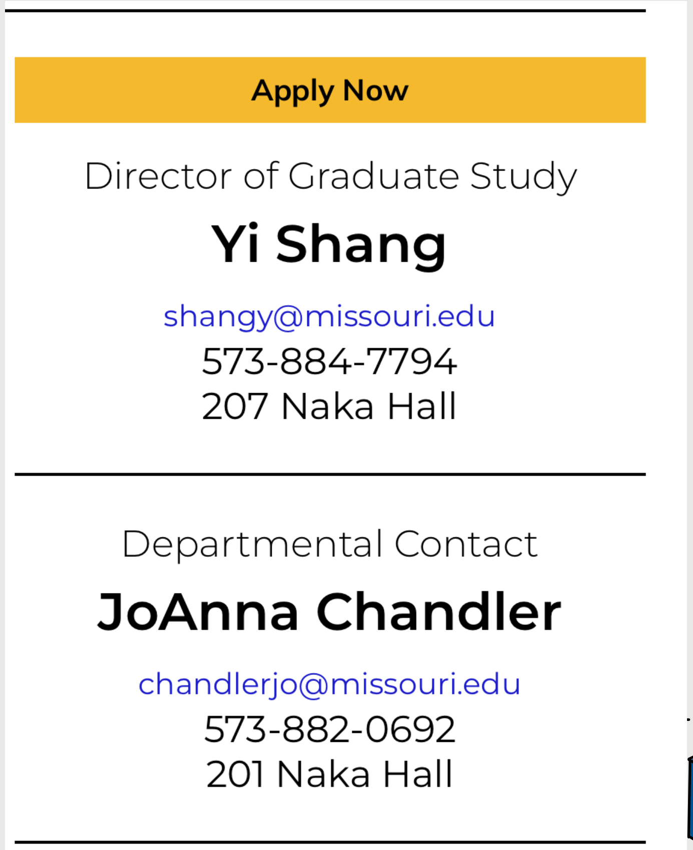
This post-doctoral position will be in the broad areas of graph data management, data provenance/lineage, workflow systems, notebooks and scientific data management. **The objective will be to create new ways of capturing, tracking and making understandable large-scale, distributed scientific experiments.**

To apply please email **your CV and a Research Statement to tmnhx at missouri.edu**

Open PhD Position

The Department of Electrical Engineering and Computer Science in the College of Engineering at University of Missouri, Columbia is looking for talented and motivated graduate student to become part of our team working on exciting science research projects.

- https://engineering.missouri.edu/degree/phd-in-computer-science/
- https://gradschool.missouri.edu/degreecategory/computer-science/



Apply Now

Director of Graduate Study

Yi Shang

shangy@missouri.edu
573-884-7794
207 Naka Hall

Departmental Contact

JoAnna Chandler

chandlerjo@missouri.edu
573-882-0692
201 Naka Hall

Thank You!

Email: tanu.malik@depaul.edu, tanu@missouri.edu