Reducing Technical Debt with Reproducible Containers

Tanu Malik
2019 BSSw Fellow
Assistant Professor
School of Computing
DePaul University
Chicago, IL

IDEAS-ECP Webinar, November 4th, 2020

WhoamI

Tanu Malik
Assistant Professor,
School of Computing
Director, Data Systems and Opt. Lab
DePaul University
Chicago, IL
https://facsrv.cs.depaul.edu/~tmalik1
Tanu.Malik@depaul.edu

My expertise is:
Databases and distributed computing
Data provenance: history and lineage of data and software
Computational reproducibility: Repeating and recreating some one else's work

Systems built: http://sciunit.run

I want to know more about:
Reproducibility case studies in HPC and how containers are used.

Problems I'm currently working on:
Provenance alignment: Using provenance to highlight sources of irreproducibility
State maintenance in lineage graphs: Making Jupyter Notebooks reproducible

Outline

PART 1: How technical debt affects reproducibility?

PART 2: If reproducible containers provide a start?

PART 3: Guidance and summary

PART 1: How technical debt affects reproducibility?

Monetary debt

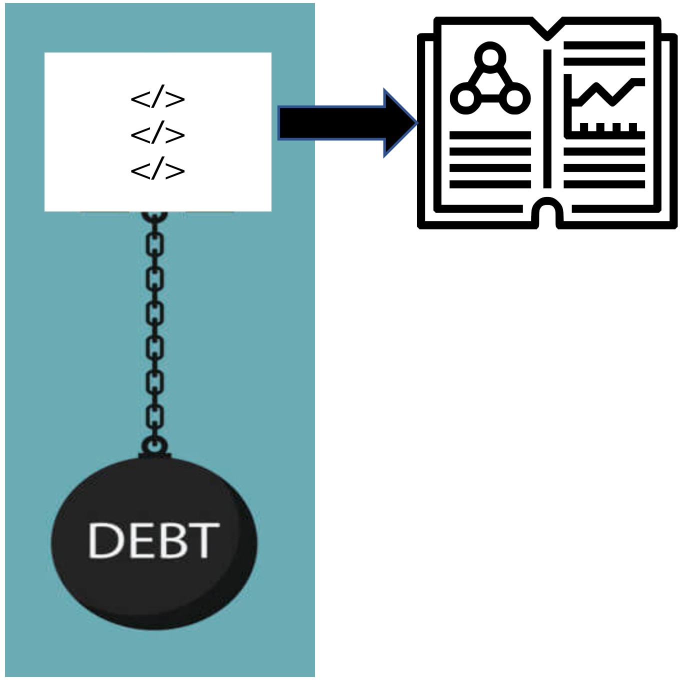
Monetary debt meets the objective "sooner"

Technical debt[1] is no different

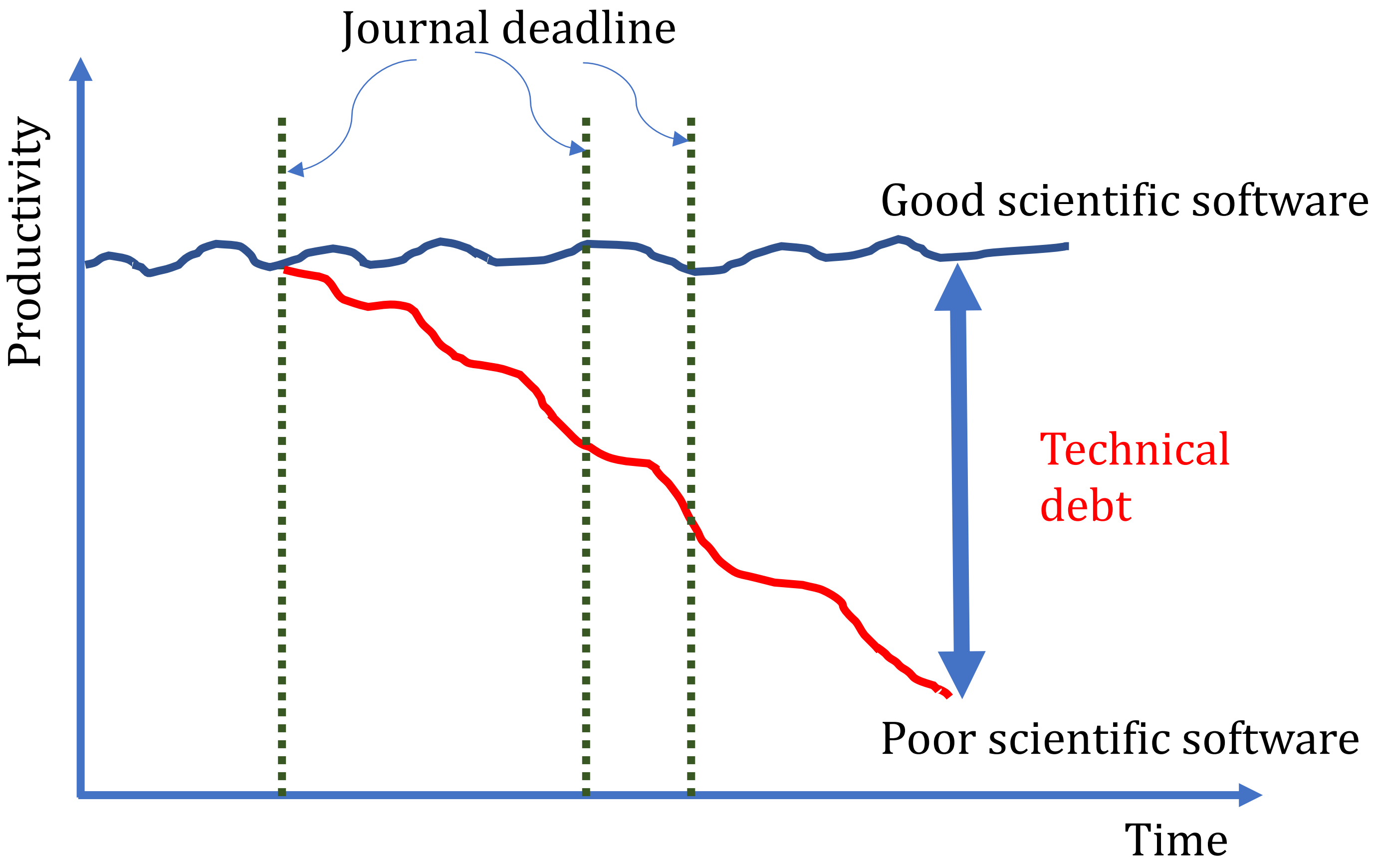
[1]A metaphor introduced by Ward Cunningham in 1992.

Technical debt[1] is no different

[1]A metaphor introduced by Ward Cunningham in 1992.



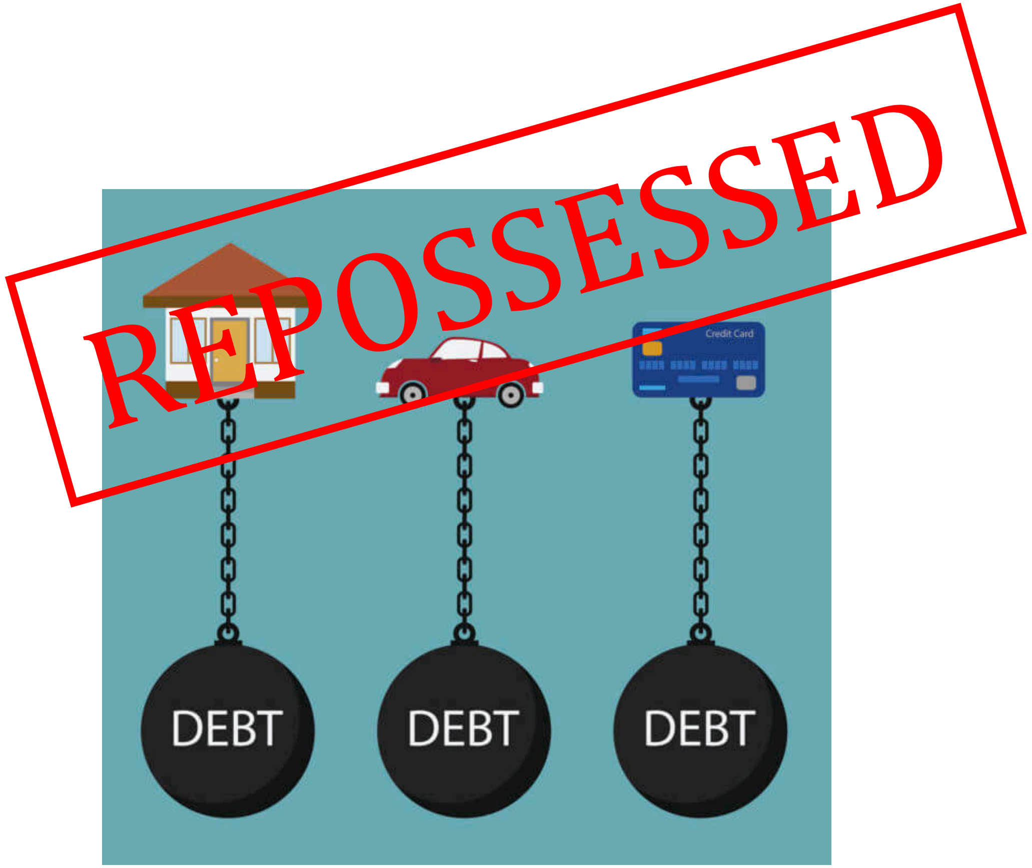
Technical debt is no different.



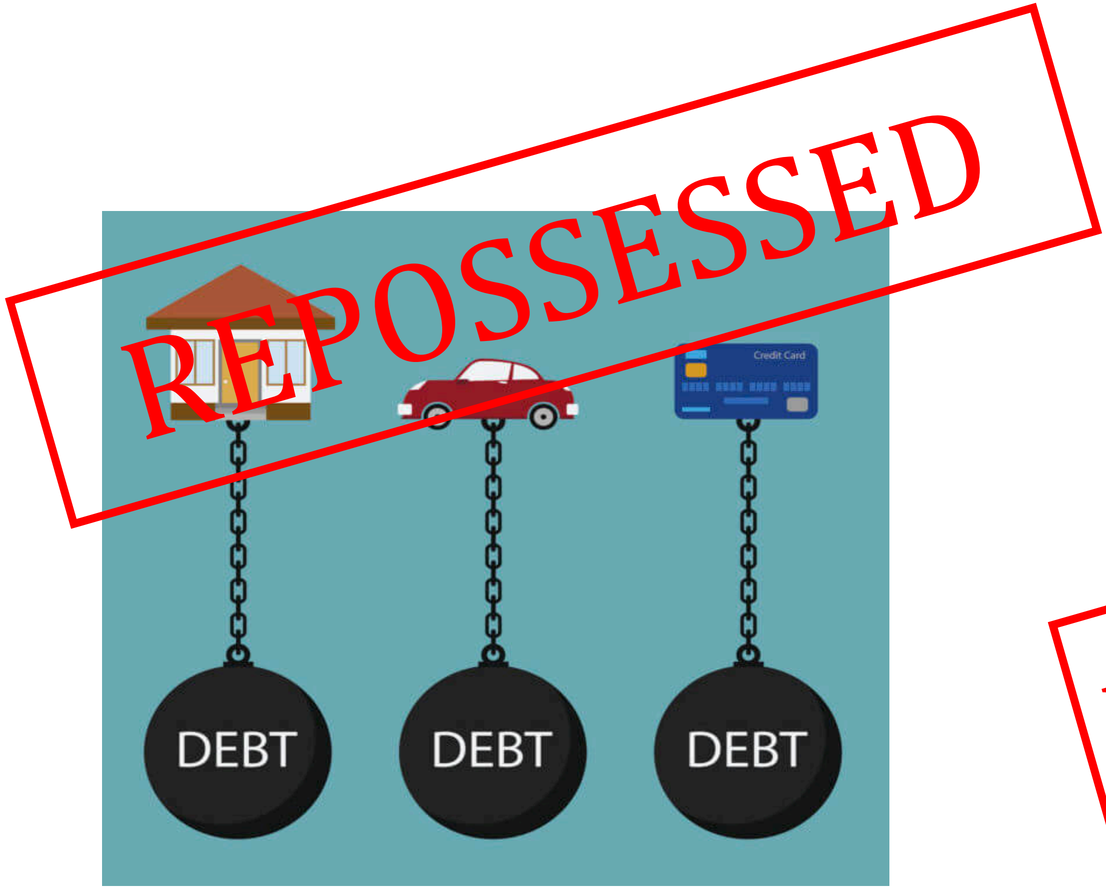
Dimensions of Technical Debt

- Poor quality code
- Poor design
- Environment debt
- Documentation debt
- Testing debt

Consequence of Mismanaged Debt



Consequence of Mismanaged Debt



Dimensions of Scientific Technical Debt

- Poor quality code
- Poor design
- Environment debt
- Documentation debt
- Testing debt

[1]E. Tom, A. Aurum, R. Vidgen, An exploration of technical debt, Journal of Systems and Software, Volume 86, Issue 6, 2013, Pages 1498-1516, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2012.12.052.

Dimensions of Scientific Technical Debt

- ~~Poor quality code~~
- ~~Poor design~~

✓Environment debt

✓Documentation debt

- ~~Testing debt~~

Bad bugs: The worst disasters caused by software fails

5 June 2013

Clever software can make our lives easier but a glitch can have disastrous consequences. In the past decades, computer bug catastrophes have caused deaths and disrupted lives on a large scale. **Sally Adee** takes us through six major software fails.

https://www.newscientist.com/gallery/software-bugs

A Scientist's Nightmare: Software Problem Leads to Five Retractions

Greg Miller

+ See all authors and affiliations

Science 22 Dec 2006:
Vol. 314, Issue 5807, pp. 1856-1857
DOI: 10.1126/science.314.5807.1856

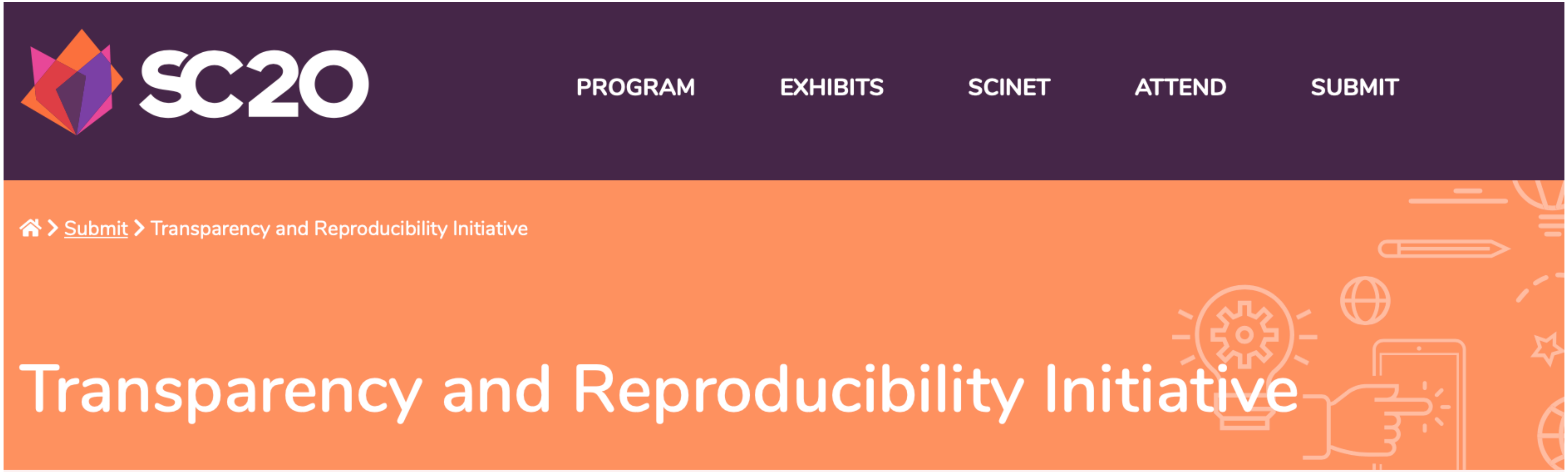
Critiqued coronavirus simulation gets thumbs up from code-checking efforts

Influential model judged reproducible – although software engineers called its code 'horrible' and 'a buggy mess'.

https://www.nature.com/articles/d41586-020-01685-y

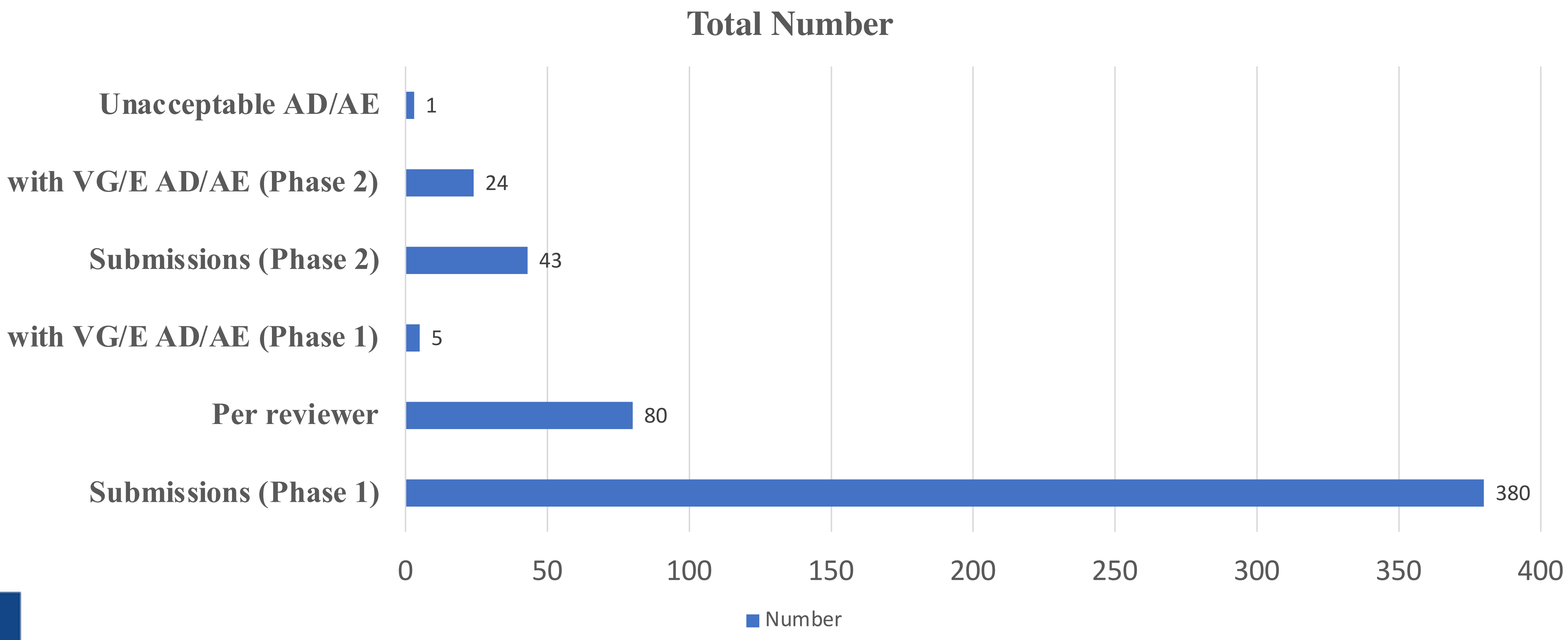
Cost of Scientific Technical Debt

Supercomputing Artifact Description and Evaluation Initiative



https://sc20.supercomputing.org/planning-committee/

Lack of artifacts will reject a paper



Technical debt incurs burden

- Reproducibility is an after thought.
- Identifying files for an application is a challenge
- Missing workflows
  - Really, that data/algorithm should be part of the bundle?
- “Sticks” from reviewers work
  - Authors who have not taken AD/AE process seriously do submit additional work
- Time consuming task
  - No tools to check if everything relevant for the publication is submitted
- No mapping of experiments to content in the paper.
  - No infrastructure for efficiently verifying claimed results

PART 2: Do reproducible containers provide a start?

Reproducibility ecosystem

Zenodo.org

Github

OpenData.gov

Package managers

Figshare

Sharing images via the cloud

Docker.com

[An introduction to **Docker** for **reproducible research**]
[C Boettiger] - ACM SIGOPS Operating Systems Review, 2015 - dl.acm.org

Docker: Using containers from build to run

Build

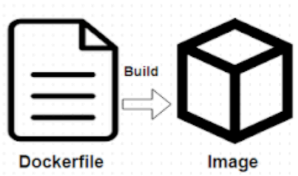
- Build images that captures applications requirements.
- Manually commit or use a recipe file.

Ship

- Push an image to DockerHub, a hosted registry, or a private Docker Registry.
- Share Images

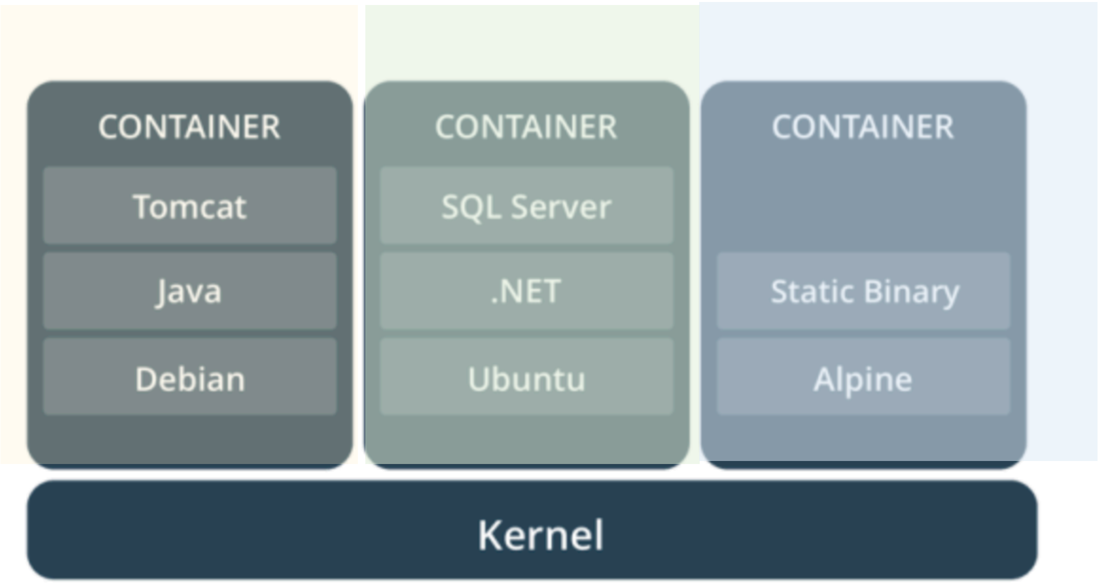
Run

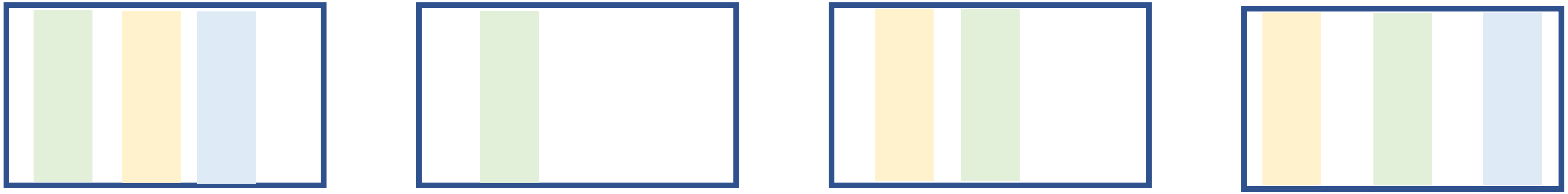
- Use Docker Engine to pull images down and execute a container from the image.



https://www.exascaleproject.org/event/conthpc

Containers provide constrained resource isolation





CPU Memory Filesystem Network

Authors must program a Dockerfile

```
# fetch node v4 LTS codename argon
FROM node:argon

# Request samplename build argument
ARG samplename

# Create app directory
RUN mkdir -p /usr/src/spfx-samples
WORKDIR /usr/src/spfx-samples

#Install app dependencies
RUN git clone https://github.com/SharePoint/sp-dev-fx-webparts.git .
WORKDIR /usr/src/spfx-samples/samples/$samplename

# install gulp on a global scope
RUN npm install gulp -g

# RUN ["npm", "install", "gulp"]
RUN npm install
RUN npm cache clean

# Expose required ports
EXPOSE 4321 35729 5432

# Run sample
CMD ["gulp", "serve"]
```

```
FROM buildkit-export AS buildkit-buildkitd.oci_only
COPY --from=buildkitd.oci_only /usr/bin/buildkitd.oci_only /usr/bin/
COPY --from=buildctl /usr/bin/buildctl /usr/bin/
ENTRYPOINT ["buildkitd.oci_only"]

# Copy together all binaries for containerd worker mode
FROM buildkit-export AS buildkit-buildkitd.containerd_only
COPY --from=runc /usr/bin/runc /usr/bin/
COPY --from=buildkitd.containerd_only /usr/bin/buildkitd.containerd_only /usr/bin/
COPY --from=buildctl /usr/bin/buildctl /usr/bin/
ENTRYPOINT ["buildkitd.containerd_only"]

FROM alpine AS containerd-runtime
COPY --from=runc /usr/bin/runc /usr/bin/
COPY --from=containerd /go/src/github.com/containerd/containerd/bin/containerd* /usr/bin/
COPY --from=containerd /go/src/github.com/containerd/containerd/bin/ctr /usr/bin/
VOLUME /var/lib/containerd
VOLUME /run/containerd
ENTRYPOINT ["containerd"]

FROM buildkit-${BUILDKIT_TARGET}
```

Containers do not reduce technical debt

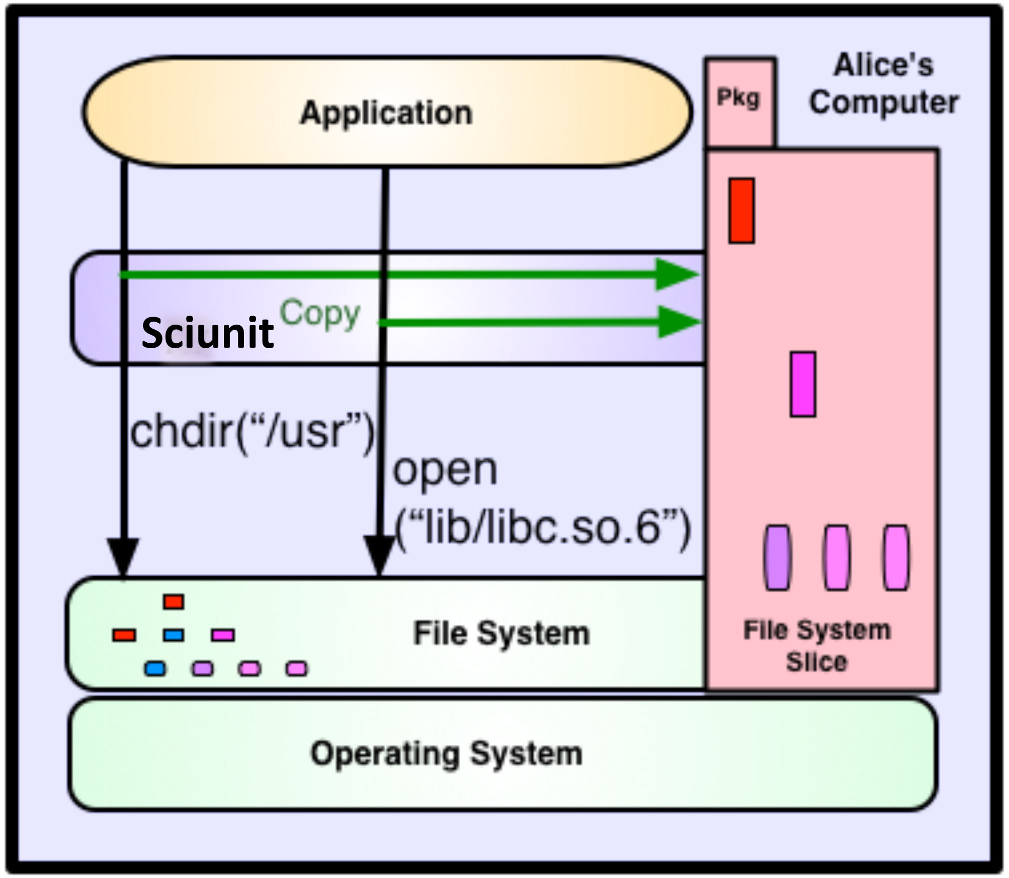
- Declarative encapsulation of dependencies for isolated execution
  - E.g. various shell utilities and library versions unknown to user

Automatic Encapsulation of Dependencies: The Sciunit

Key Idea: Identify dependencies during program execution

- Captures application dependencies during executions
- Repeats executions (with guarantees) within isolated environments

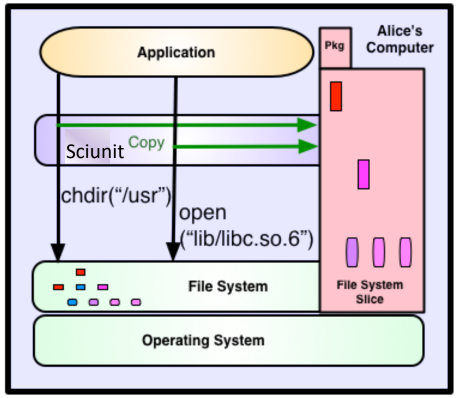
Sciunit: Audit

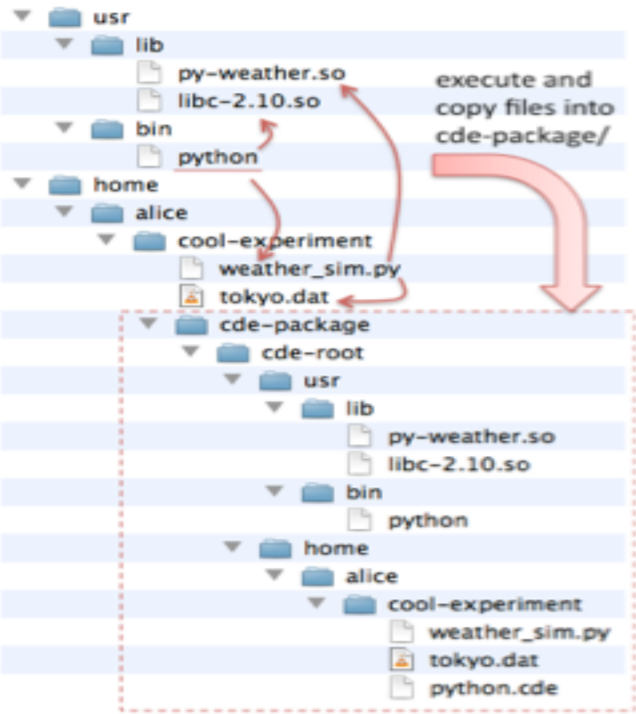


- Audit uses *ptrace* to observe dependencies and environment variables
  - Identifies binaries, libraries, scripts, and environment variables that application is dependent on.
- Dependencies are copied into a directory in the filesystem
- Inclusion of data files is optional
  - user may or may not want to package based on the size of the dataset.

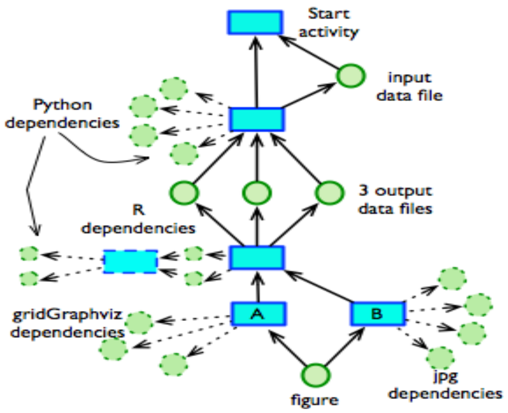
D.H. Ton That, G. Fils, Z. Yuan, T. Malik. Sciunits: Reusable Research Objects. In *IEEE eScience Conference* (eScience), 374-383, 2017

Audits provenance during execution time



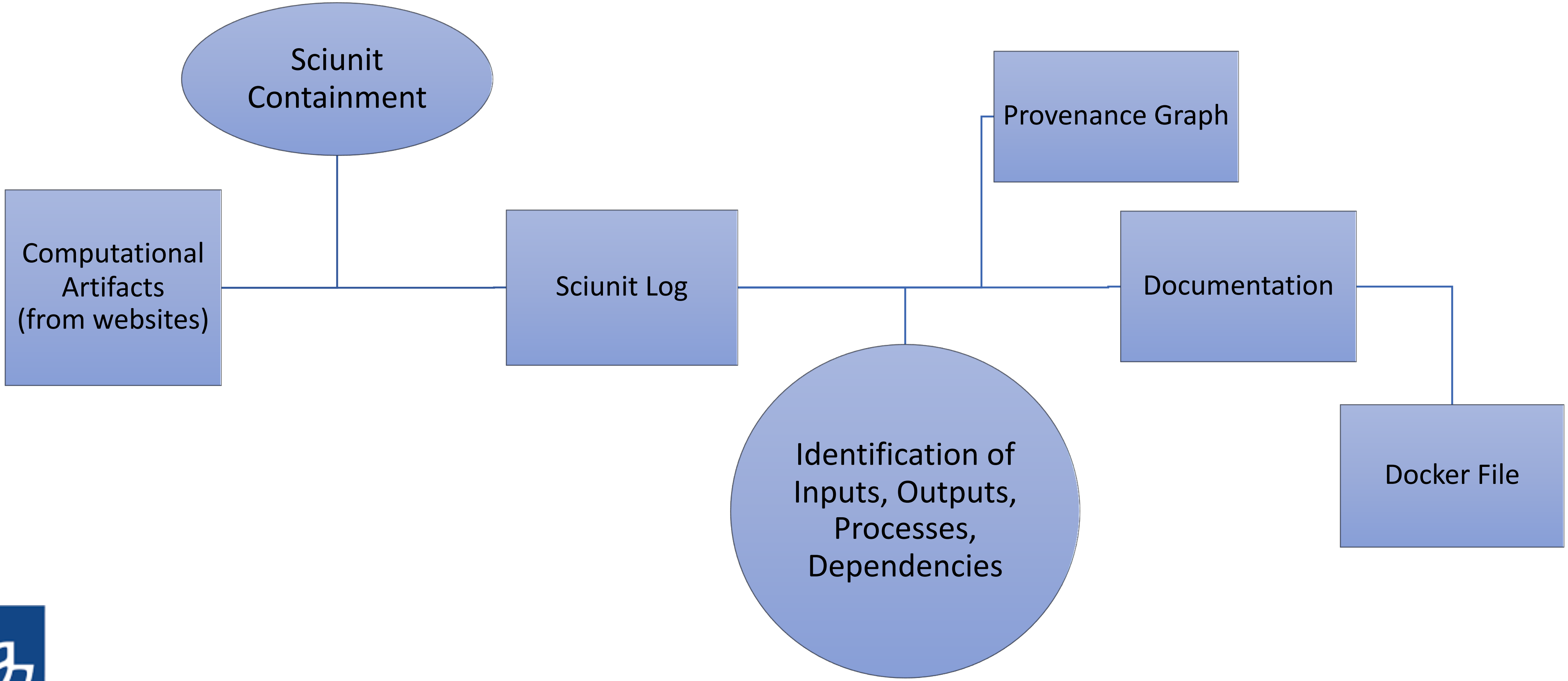



```
1542206708 2617 EXECVE 2623 /usr/bin/python3 /home/ubuntu/scripts/pysumma ["python3", "python_script.py"]
1542206708 2623 READ /etc/ld.so.cache
1542206708 2623 READ /lib/x86_64-linux-gnu/libpthread.so.0
1542206708 2623 READ /usr/lib/locale/locale-archive
1542206708 2623 READ /usr/lib/python3.6/__pycache__/_sysconfigdata_m_linux_x86_64-linux-gnu.cpython-36.pyc
1542206708 2623 READ /usr/lib/python3.6/collections
1542206708 2623 READ /home/ubuntu/scripts/pysumma/python_script.py
1542206708 2623 READ /etc/localtime
1542206708 2623 READ /home/ubuntu/.local/lib/python3.6/site-packages/dateutil
1542206709 2623 READ /home/ubuntu/.local/lib/python3.6/site-packages/numpy
1542206709 2623 READ /home/ubuntu/.local/lib/python3.6/site-packages/numpy/core
1542206709 2623 READ /home/ubuntu/.local/lib/python3.6/site-
1542206709 2623 READ /usr/lib/python3.6/unittest
```



Utilizing Provenance in Reusable Research Objects, In *Special Issue on Using Computational Provenance*, MDPI Informatics, Vol 5(1), 2018.
Light-weight Database Virtualization. In *IEEE International Conference on Data Engineering*, ICDE, 2015.
Auditing and Maintaining Provenance in Software Packages. In *International Provenance and Annotation Workshop* (IPAW), 97-109, 2014

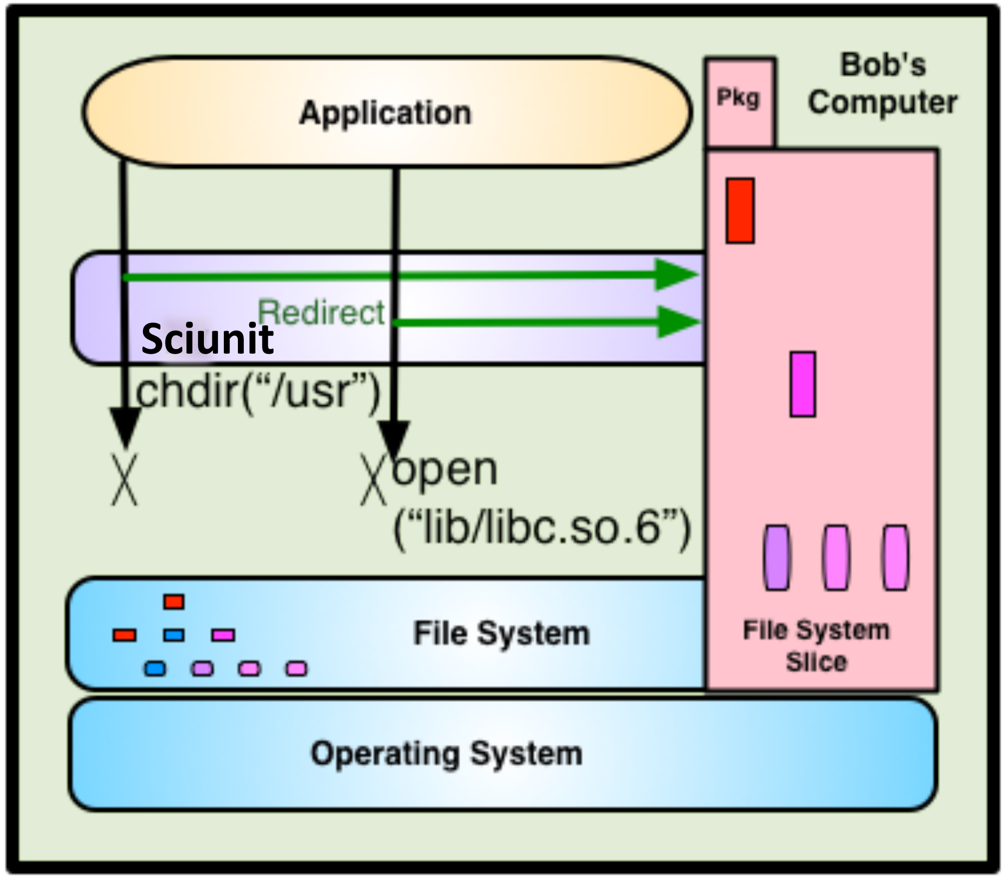
Sciunit: Share as a Zip file or Docker container



Documenting Computing Environments for Reproducible Experiments, In *Parallel Computing: Technology Trends, 756-765, 2020*

Sciunit: Repeat

- Sciunit uses namespace isolation during repeat
- Redirection of each call into the package



Efficient Provenance Alignment in Reproduced Executions, In *Theory and Practice of Provenance*, 2020.
SciInc: A Container Runtime for Incremental Recomputation", In*IEEE 15th International Conference on eScience* (eScience), 291-300, 2019, *doi:* 10.1109/eScience. 2019.00040.

Sciunit steps and external requirements

1. Create

2. Share

3. Repeat

figshare

HYDROSHARE

XSEDE
Extreme Science and Engineering Discovery Environment

CloudLab

C/C++

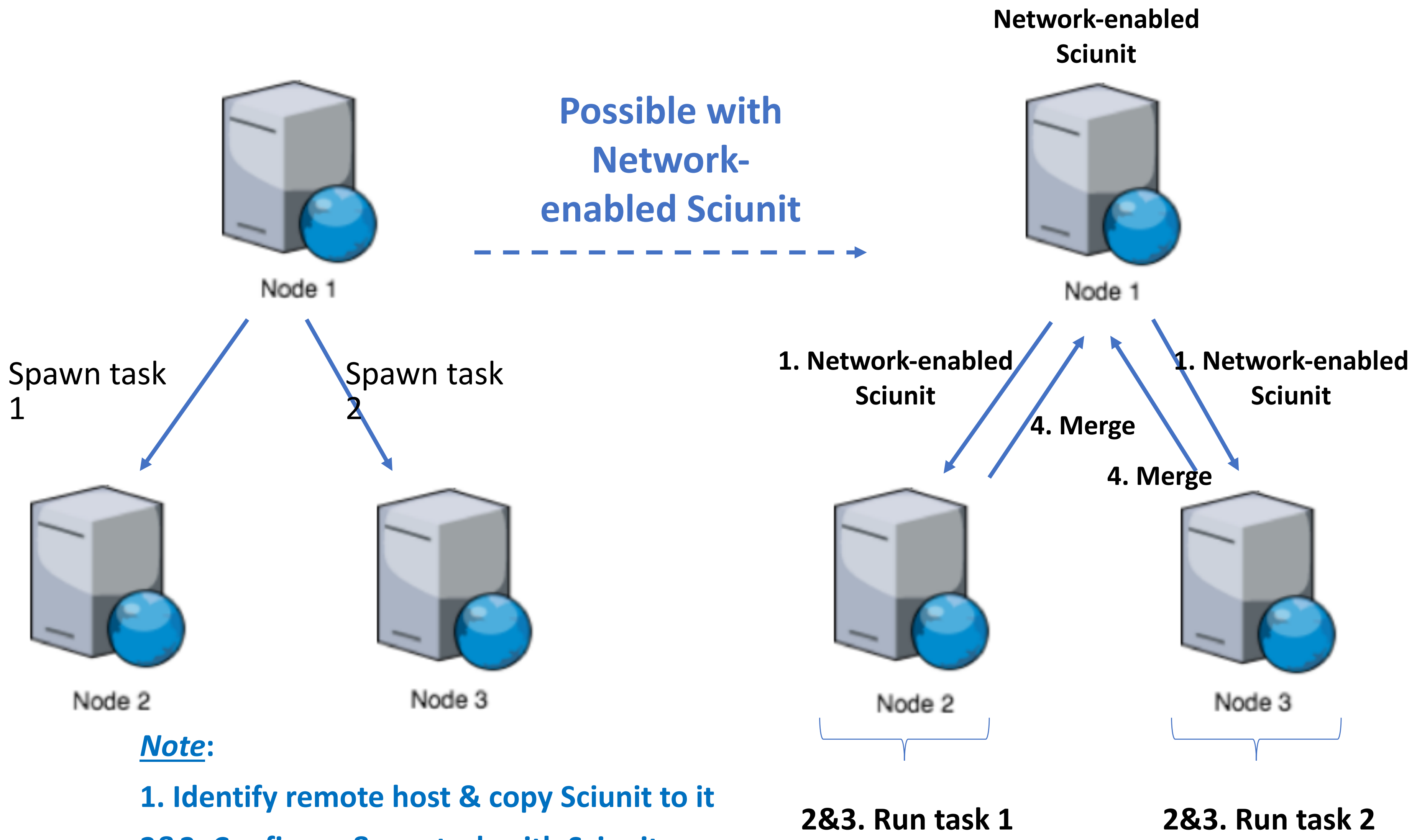
fortran
THE ORIGINAL HANGING CHAD

julia

jupyter

OPEN MPI

Network-enabled Sciunit: Audit



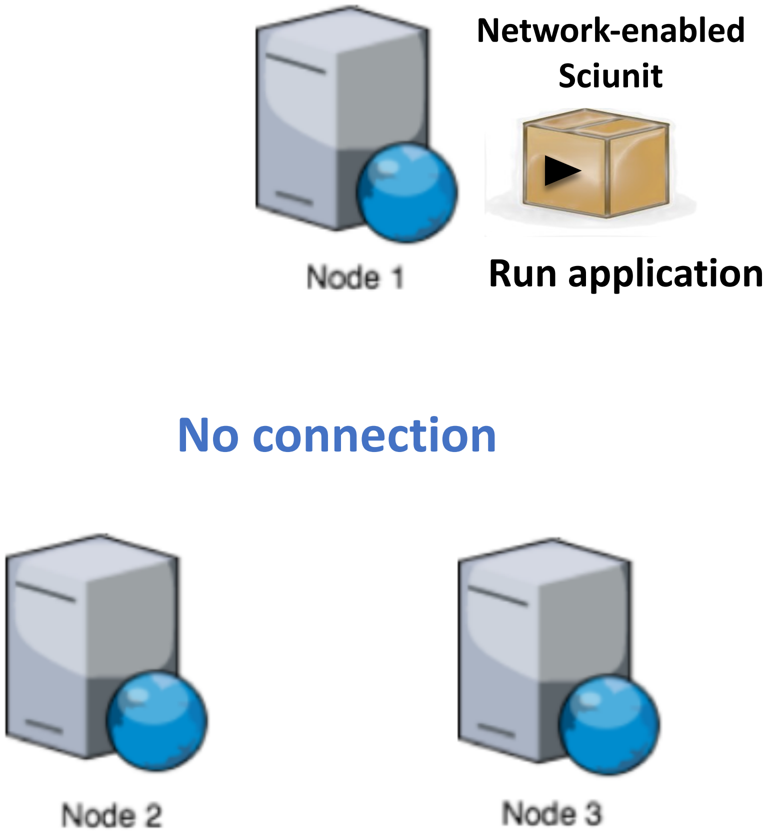
Note:

1. Identify remote host & copy Sciunit to it

2&3. Configure & run task with Sciunit

4. Retrieve & manually merge

Network-enabled Sciunit: Repeat on single node



***Note*:**

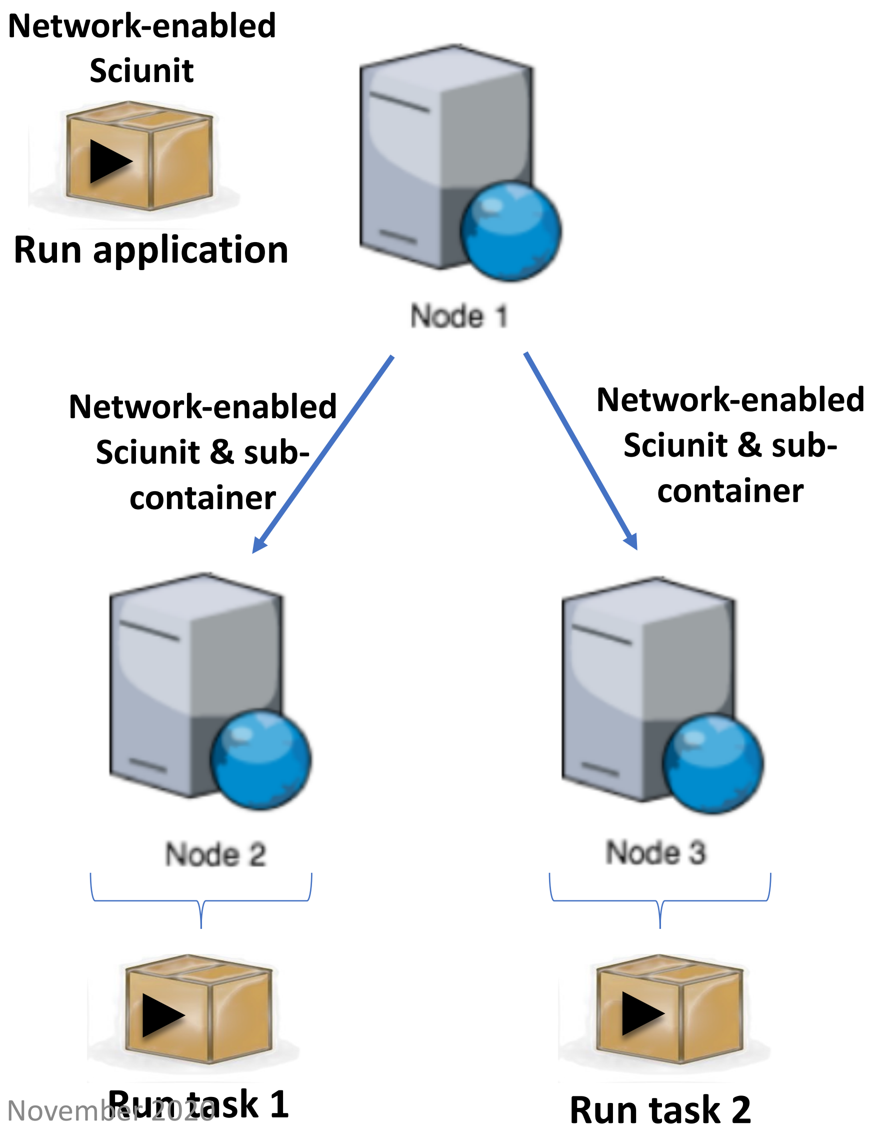
1. Repeat all computations at root node.

2. Network system calls are supplied through the content data captured during the original audit.

Network-enabled Sciunit: Repeat on multiple nodes

Requirements:

1. Identical number of nodes
2. Descriptions of new hostnames or IP addresses



Usecases

TABLE I: Usecases descriptions.

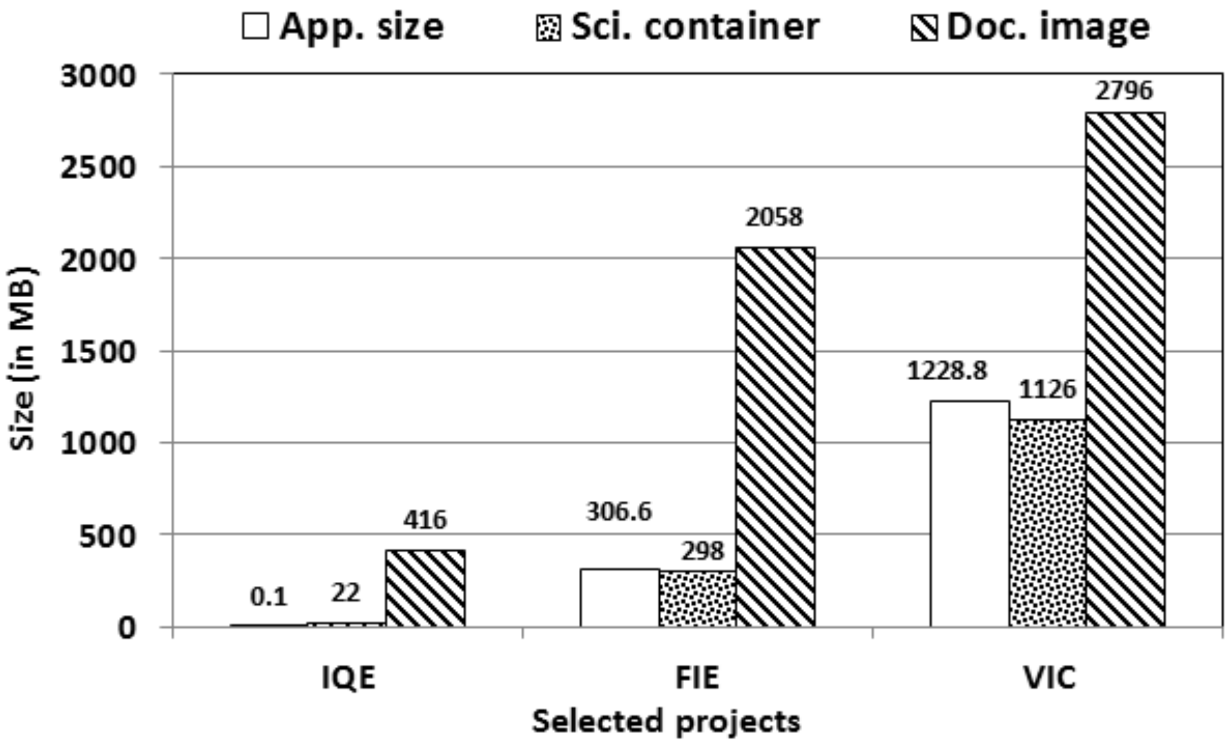
	FIE [16]	**VIC [17]**	**IQE [18]**
Source code languages	R, Bash	C, C++, Python, C shell script, Fortran	Python
Source code files	29	97	5
Data files	14	11,481	5
Dependency files	659	357	112
Size of all files	306.6 MB	1.2 GB	22 MB
Normal run time	286.756 s	40.259 s	5.226 s

[16[City of Chicago, "Food Inspection Evaluation," https://chicago.github. io/food-inspections-evaluation/, 2017, [Online; accessed 7-May-2017].

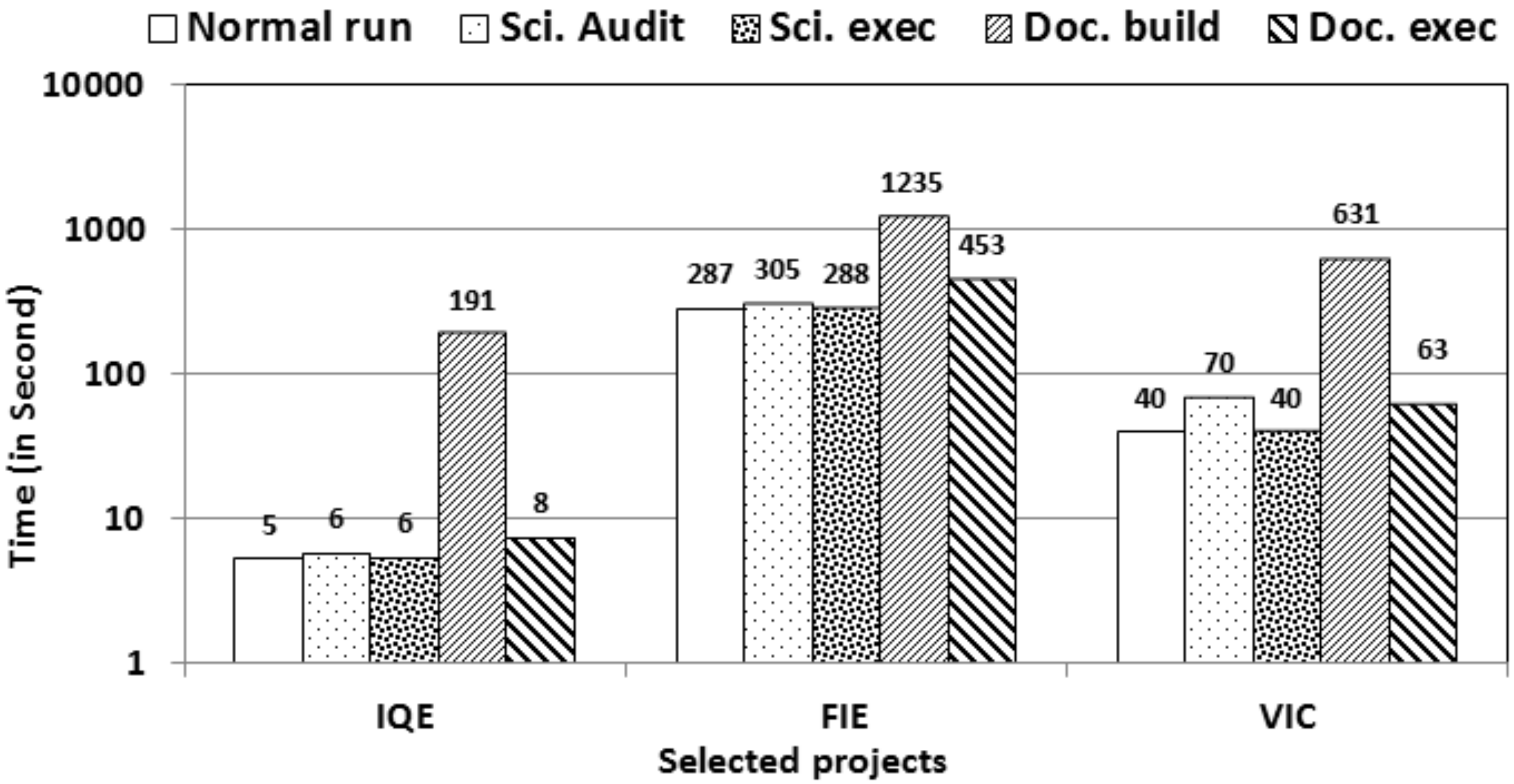
[17] M. M. Billah, J. L. Goodall *et al.*, "Using a data grid to automate data preparation pipelines required for regional-scale hydrologic modeling," *Environmental Modelling & Software*, vol. 78, 2016.

18] D. DBGroup, "Incremental Query Execution," 2019, [Online; accessed 3-April-2019]. [Online]. Available: https://TonHai@bitbucket. org/TonHai/iqe.git

Sciunit (native) versus Docker sizes



Sciunit audit and repeat times



Experiments

- NASA Parallel Benchmark:
  - Data transferred (~524 KB (class A) & ~268 KB (class B))

	Normal	#Calls	Meta Audit	Content Audit
NPB BT-MZ.A.2	20.30			
NPB LU-MZ.A.2	15.74	190	$\sim 2.1\%\uparrow$	$\sim 5.3\%\uparrow$
NPB SP-MZ.A.2	14.84			
NPB BT-MZ.B.2	83.95			
NPB LU-MZ.B.2	71.02	190	$\sim 0.8\%\uparrow$	$\sim 3.2\%\uparrow$
NPB SP-MZ.B.2	59.12			

- VIC

	Normal	Meta Audit	Content Audit	Replay
Data retrieval	146.5±1.8	$0.2\%\uparrow$	$134.5\%\uparrow$	53.0±3.0
Other steps	varies	2% - 30% ↑ overhead		

Sample Interaction of Sciunit

```
1. > sciunit create FIE
2. > sciunit exec ./FIE.sh ./DATA/weather_201710.Rds
     0. Download…
     1. Calculate violation matrix…
     2. Calculate heat map…
     3. Generate model data with ./DATA/weather_201710.Rds…
     4. Apply random forest model…
     5. Evaluation…
3. > sciunit list
     e1 Dec 4 12:44 ./FIE.sh ./DATA/weather_201710.Rds
4. > sciunit show
     id: e1
     sciunit: FIE
     command: ./FIE.sh ./DATA/weather_201710.Rds
     size: 306.6 MB
     started: 2017-12-04 12:44
5. > sciunit push
     …
     Title for the new article: FIE
     new: 306.6 MB [01:05, 4.72MB/s]
6. > sciunit copy
     mSLLTj#
```

Alice's Computer

```
1. > sciunit open mSLLTj#
     Opened Sciunit FIE
2. > sciunit list
     e1 Dec 4 12:44 ./FIE.sh ./DATA/weather_201710.Rds
3. > sciunit repeat e1
     …
     0. Download…
     1. Calculate violation matrix…
     2. Calculate heat map…
     3. Generate model data with ./DATA/weather_201710.Rds
     4. Apply random forest model…
     5. Evaluation…
4. > sciunit given '/tmp/weather_201801.Rds' e1 %
     …
     1. Generate model data with '/tmp/weather_201801.Rds
     2. Apply random forest model…
     3. Evaluation…
5. > sciunit list
     e1 Dec 4 12:44 ./FIE.sh ./DATA/weather_201710.Rds
     e2 Dec 14 2:44 ./FIE.sh ./tmp/weather_201801.Rds
```

Bob's Computer

Container Limitations

- Container either include the data or exclude the data
  - The decision is binary but does not consider necessary and sufficient data

Container Debloating: MiDAS

Example

```
void file_read(int bytes) {
        int fd, sz;
        char *c = (char *) calloc(bytes, sizeof(char));
        fd = open("test.txt", O_RDWR);
        lseek(fd,100,SEEK_SET);
        sz = read(fd, c, bytes);
}
```

```
test.txt, 100, 150
test.txt, 100, 190
test.txt, 100, 230
test.txt, 100, 450
```

Example

```
void file_read(int bytes) {
	int fd, sz;
	char *c = (char *) calloc(bytes, sizeof(char));
	fd = open("test.txt", O_RDWR);
	lseek(fd,100,SEEK_SET);
	sz = read(fd, c, bytes);
}
```

```
test.txt, 100, 150
test.txt, 100, 190
test.txt, 100, 230
test.txt, 100, 450
```

MiDAS: Minimizing DAtasetS

WHAT

Automatically identify & include **ONLY relevant data chunks** with application

HOW

Map **high level user inputs** to **file offsets**

Partial Evaluation & LLVM

- **Partial Evaluation**→optimization technique to **prune codebase**
  - Uses static inputs to generate a **specialized program** to accept remaining dynamic inputs

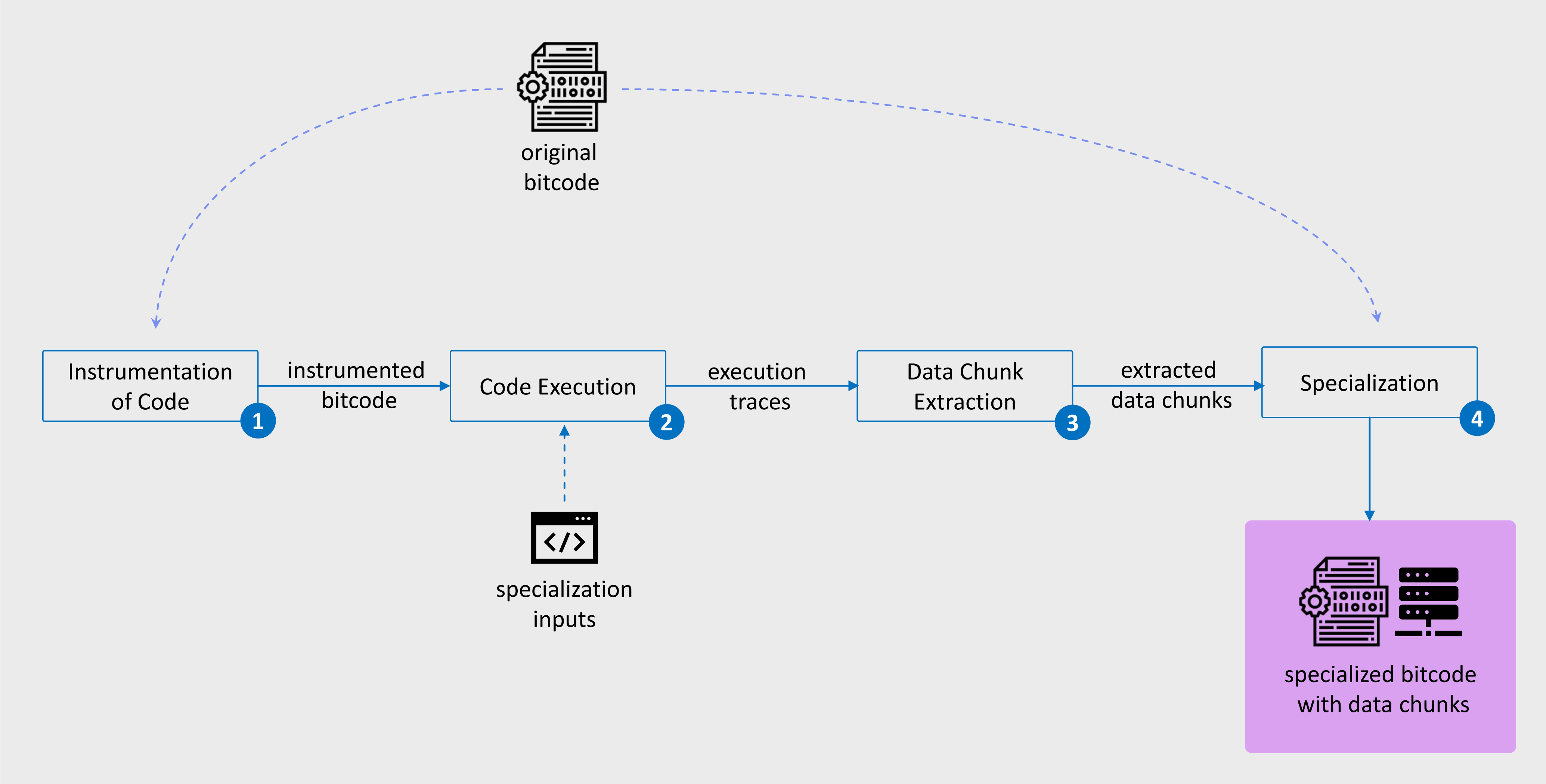
```
#include <math.h>
float compute_building_height(float building_distance){
        float viewing_angle = pi/4;
        float building_height =
          compute_opposite(building_distance,
            viewing_angle);
        return building_height;
}
float compute_opposite(float adjacent, float angle){
        float opposite = adjacent * tan(angle);
        return opposite;
}
```

(a) Original code

```
float compute_building_height(float building_distance){
        float building_height =
          compute_opposite_specialized(building_distance);
        return building_height;
}
float compute_opposite_specialized(float adjacent){
        float opposite = adjacent * 1;
        return opposite;
}
```

(b) Specialized code

MiDAS



original bitcode
Instrumentation of Code
1
instrumented bitcode
Code Execution
2
execution traces
Data Chunk Extraction
3
extracted data chunks
Specialization
4
specialization inputs
specialized bitcode with data chunks

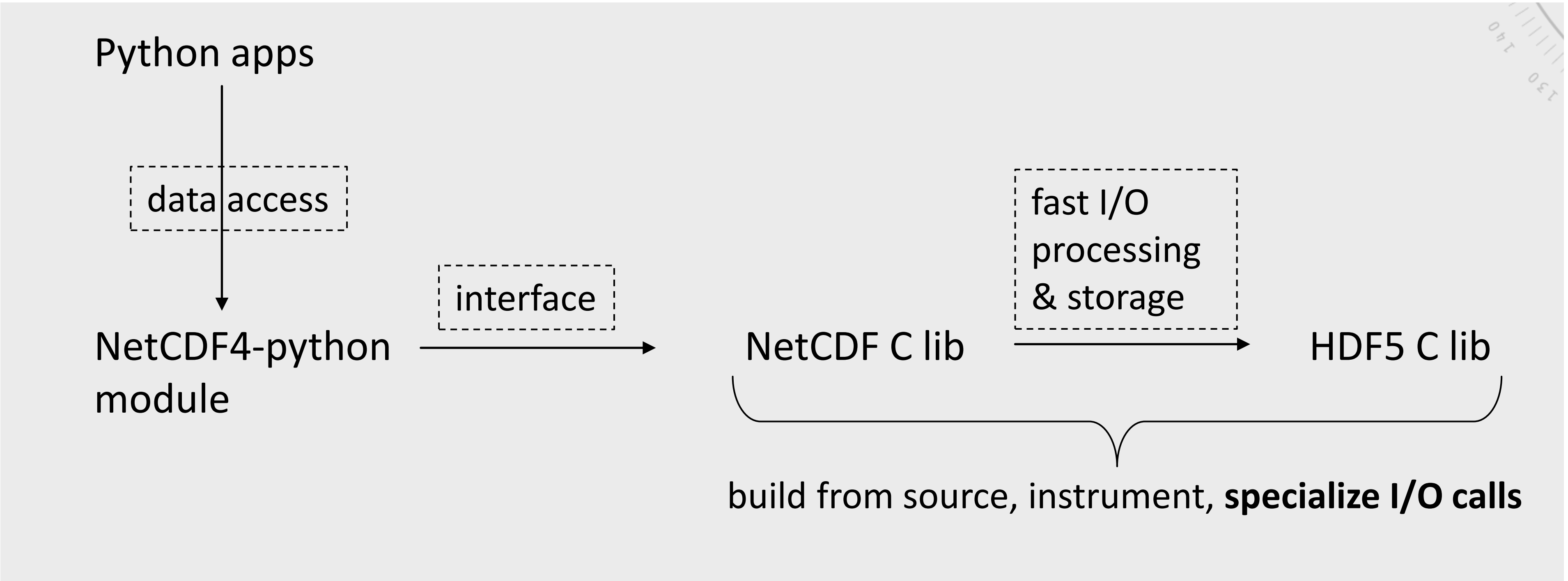
I/O Specialization

- **Replace I/O call & *preserve functionality***
  - Extracted file data in global variable→ fileData
  - Copy from global variable to *read* buffer → *memcpy*
  - Update all I/O call variables→return value of *read*
  - I/O call instruction removed → read

```
%94 = load i32, i32* %9, align 4
%95 = sext i32 %94 to i64
%96 = call i64 @read(i32 %92, i8* %93, i64 %95)
%97 = trunc i64 %96 to i32
store i32 %97, i32* %13, align 4
%98 = load i32, i32* %12, align 4
```

```
%94 = load i32, i32* %9, align 4
%95 = sext i32 %94 to i64
%96 = bitcast [17 x i8]* @fileData to i8*
call void @llvm.memcpy.p0i8.p0i8.i64(i8* %93,
        i8* %96, i64 %95, i32 1, i1 false)
%97 = alloca i64
store i64 %95, i64* %97
%loadRetVal = load i64, i64* %97
%98 = trunc i64 %loadRetVal to i32
store i32 %98, i32* %13, align 4
%99 = load i32, i32* %12, align 4
```

Specializing I/O Calls in Scientific Libraries



Results | Percentage of File Accessed

- Larger files generated from 30 MB NetCDF data file
- Rewriting data for multiple timesteps
- Data accessed corresponding to *temperature* attribute

Total Size	Accessed Size
30 MB	6.6 MB
700 MB	154 MB
1.4 GB	0.3 GB
9 GB	1.98 GB
12.8 GB	2.82 GB

APPLICATIONS OFTEN ACCESS
ONLY A SUBSET
OF A LARGE DATASET

PART 3: Summary and Guidance

Summary

- Technical debt affects reproducibility of scientific claims.
  - Process for evaluation of scientific claims is being rethought.
  - Artifact description and evaluation are becoming part of conferences
- Better reliability is needed.
  - Containers will be a prominent choice but their reliability is poor
    - Dependencies must be specified
    - Inefficient to use
    - No guarantees for execution verification
    - Not meant for interactive programs
- New light-weight methods: Sciunit, MiDAS

Use Sciunit for your next paper submission!

1. Tools downloaded ~850 times (tracked using pip)
2. 8 active contributors to the project
3. Actively used in geoscience disciplines that develop computational models and data-analytic pipelines

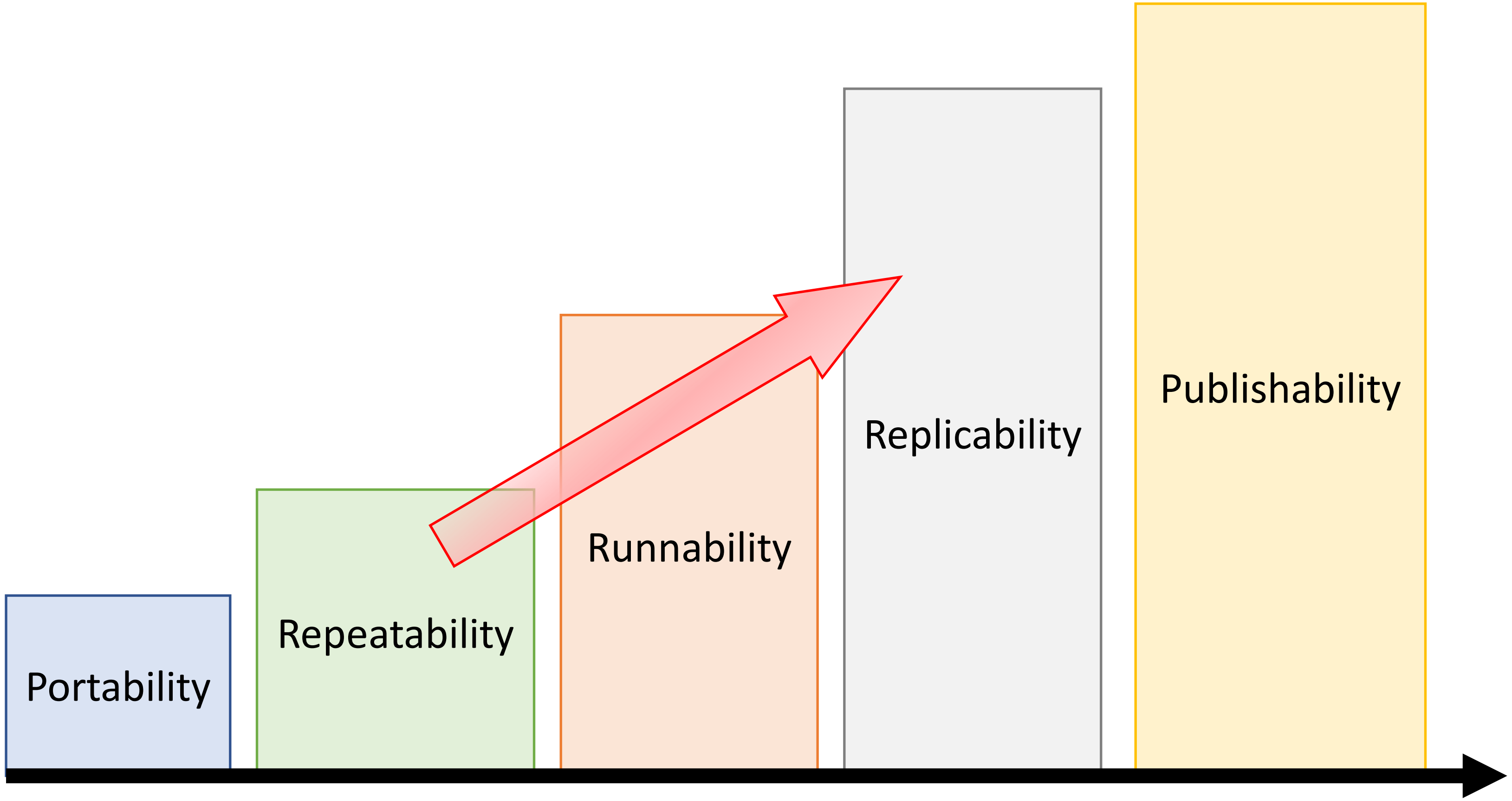
Website: http://sciunit.run

Issues and contribution: pr@sciunit.run

Guidance for Improving Reproducibility

Guidance for Improving Reproducibility

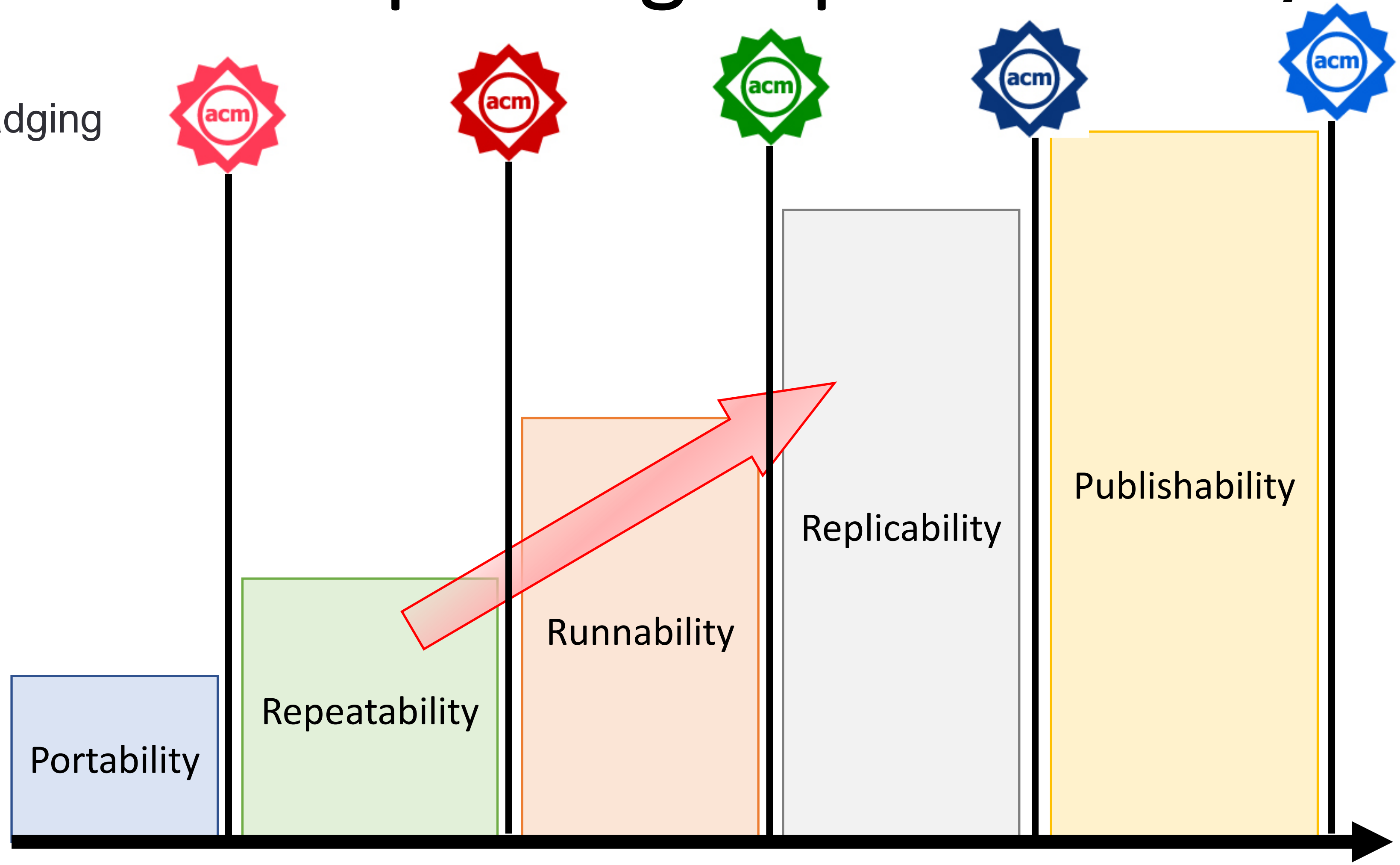
https://bssw.io/items?topic=reproducibility



1. J. Freire, N. Fuhr, and A. Rauber. *Reproducibility of data-oriented experiments in e-science* (Dagstuhl seminar 16041) Dagstuhl reports. 6(1):108–159, 2016. [Online; accessed 10-Sep-2017].

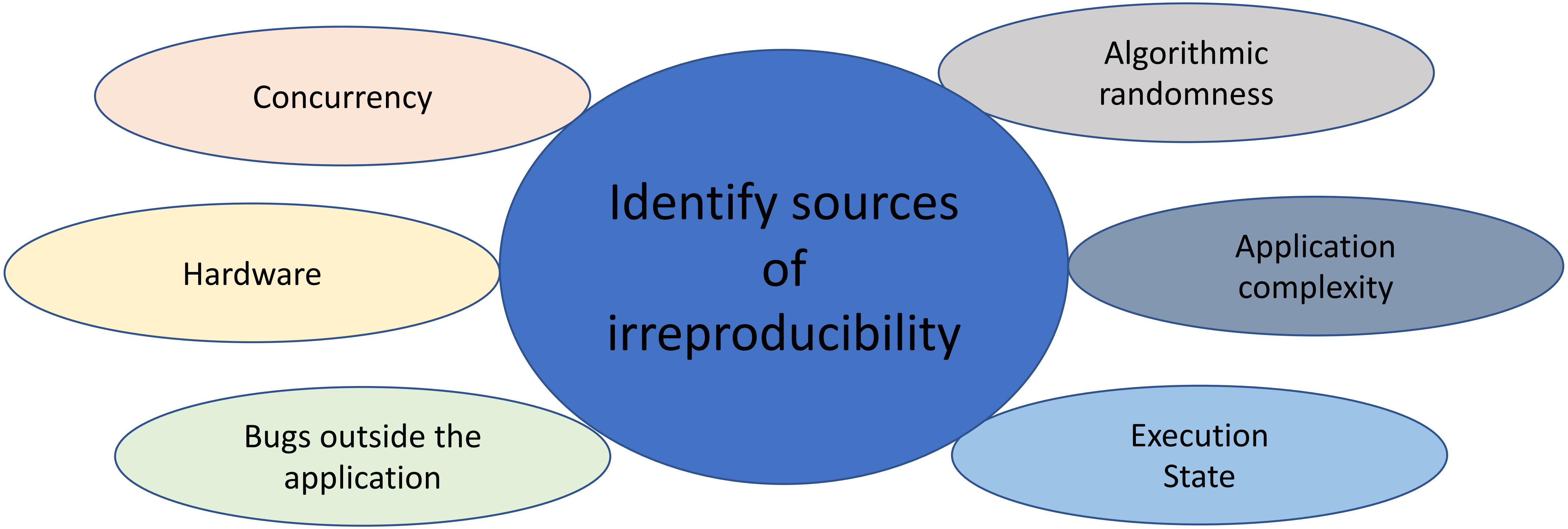
Guidance for Improving Reproducibility

Artifact Review and Badging

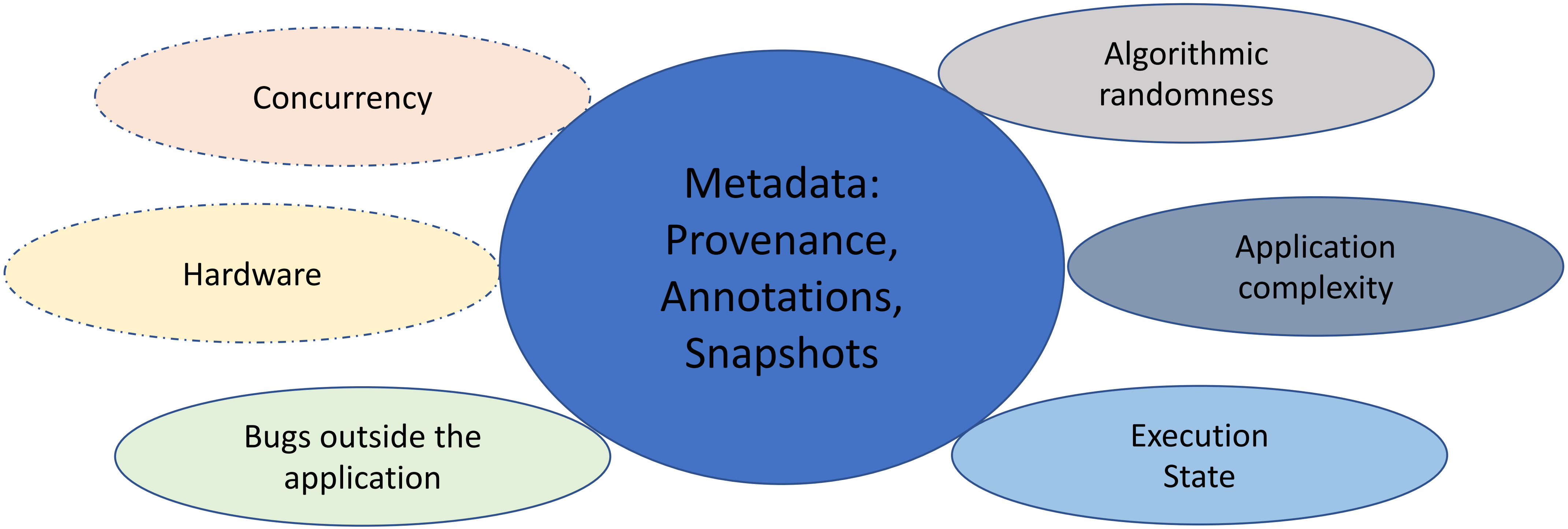


1. J. Freire, N. Fuhr, and A. Rauber. *Reproducibility of data-oriented experiments in e-science* (Dagstuhl seminar 16041) Dagstuhl reports. 6(1):108–159, 2016. [Online; accessed 10-Sep-2017].

Guidance for Improving Reproducibility

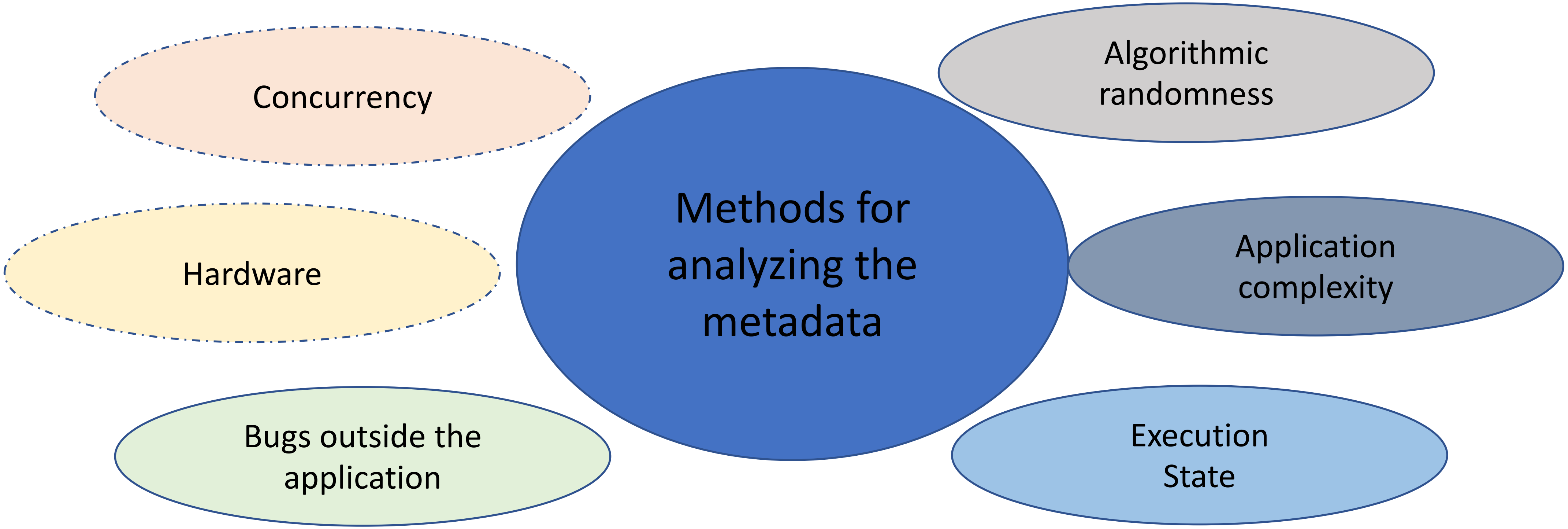


Guidance for Improving Reproducibility



https://bssw.io/items?topic=reproducibility

Guidance for Improving Reproducibility



https://bssw.io/items?topic=reproducibility

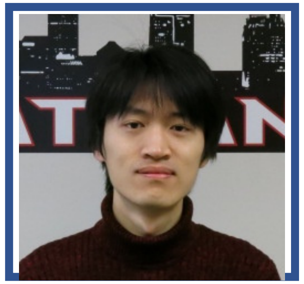
Acknowledgements

Yuta Nakamura
Ph.D. student

Raza Ahmad
Research Engineer

Nithin Manne
M.S Student

Jason Chuah
M.S. student



Zhihao Yuan
Research Engineer

Ton That Dai Hai
Postdoctoral Associate

Jason Chuah
Ph.D student @UVA

Acknowledgements

Ashish Gehani
SRI International

Ian Foster
UChicago & ANL

Dave Tarboton
Utah State

Jon Goodall
Univ. of Virginia

Scott Peckham
Univ of Colorado
Boulder

Eunseo Choi
Univ of Memphis

Acknowledgements | Funding

NSF CNS-1846418, ICER-1639759, ICER-1661918

BSSw Fellowship

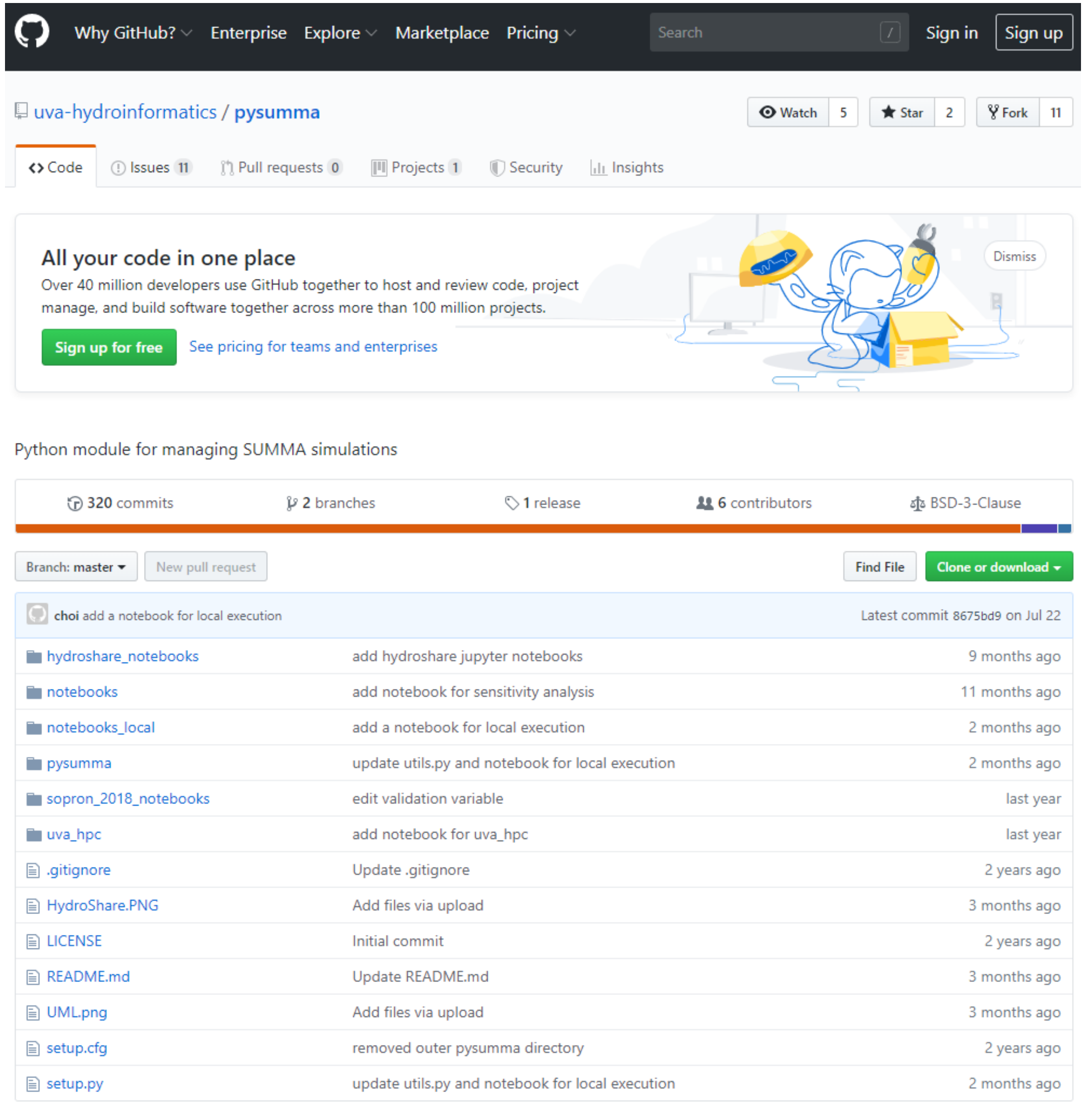
Bloomberg Foundation

DePaul Seed Grants

Questions

- tanu.malik@depaul.edu

Example



How to run pySUMMA locally

Installation and Usage

pySUMMA requires Python 3.6 and following packages :

- xarray 0.10.7 : N-D labeled arrays and datasets in python
- numpy 1.16.1 : the fundamental package for scientific computing with Python
- matplotlib 3.0.2 : a Python 2D plotting library
- seaborn 0.9.0 : statistical data visualization
- jupyterthemes 0.20.0 : select and install a Jupyter notebook theme
- hs-restclient 1.3.3 : HydroShare REST API python client library
- ipyleaflet 0.9.2 : A jupyter widget for dynamic Leaflet maps
- Linux Environment (VirtualBox 5.2.8)
  - lubuntu-16.10 executable
  - lubuntu-16.04.4 executable

Download and Install pySUMMA:

1.) Download pySUMMA

```
~/Downloads$ git clone https://github.com/uva-hydroinformatics/pysumma.git
```

Result

Listed Packages	Identified Packages
xarray{0.10.7} numpy{1.16.1} matplotlib{3.0.2} hs-restclient{1.3.3} ipyleaflet{0.9.2} seaborn{0.9.0} jupyterthemes{0.20.0}	xarray{0.10.7} numpy{1.16.1} matplotlib{3.0.2} hs-restclient{1.3.3} ipyleaflet{0.9.2}
Identified Sub-Packages	
Pygments{2.2.0} asyncio backcall{0.1.0} blinker{1.3} certifi{2018.10.15} cftime{1.0.2.1} geopandas{0.4.0} html http ipykernel{5.1.0} ipython-genutils{0.2.0} ipython{7.1.1} ipywidgets{7.4.2} jedi{0.13.1} jupyter-core{4.4.0} netCDF4{1.4.2} pandas{0.23.4} parso{0.3.1} pexpect{4.6.0} prompt-toolkit{2.0.7} ptyprocess{0.6.0} pyparsing{2.3.0} pysumma{0.1} pytz{2018.7} pyzmq{17.1.2} requests-oauthlib{1.0.0} requests-toolbelt{0.8.0} tornado{5.1.1} traitlets{4.3.2} traittypes{0.2.1} wcwidth{0.1.7}	
Python Built-In Packages	
chardet collections concurrent ctypes dateutil distutils email encodings idna importlib jinja2 json logging markupsafe multiprocessing oauthlib pkg_resources pydoc_data requests sqlite3 unittest urllib urllib3 xml	

Current and Future Work

- Developing Sciunit audit and repeat with checkpoint-restart
  - Compute- and data-analytic models that vary several parameters and are reexecuted multiple times to test their reproducibility.
  - Useful for Jupyter Notebooks

- Sciunit for reproducibility will provide provenance-based guarantees
  - Several cyberinfrastructure for Artifact Evaluation (OCCAM, CKFoundation)
  - Provenance-based guarantees are missing

- Developing MiDAS for different inputs