Conducting Reproducible Research with Umbrella: Tracking, Creating, and Preserving Execution Environments

Haiyan Meng, Alexander Vyushkov, Matthias Wolf, Anna Woodard, and Douglas Thain
University of Notre Dame, Notre Dame, IN 46556, USA **Email**:{hmeng, avyushko, mwolf3, awoodard, dthain}@nd.edu

UNIVERSITY OF NOTRE DAME

1. ABSTRACT

Publishing scientific results without the detailed execution environments describing how the results were collected makes it difficult or even impossible for the reader to reproduce the work. However, the configurations of the execution environments are too complex to be described easily by authors. To solve this problem, we propose a framework facilitating the conduct of reproducible research by tracking, creating, and preserving the comprehensive execution environments with Umbrella. The framework includes a lightweight, persistent and deployable execution environment specification, an execution engine which creates the specified execution environments, and an archiver which archives an execution environment into persistent storage services like Amazon S3 and Open Science Framework (OSF). The execution engine utilizes sandbox techniques like virtual machines (VMs), Linux containers and user-space tracers, to create an execution environment, and allows common dependencies like base OS images to be shared by sandboxes for different applications.

We evaluate our framework by utilizing it to reproduce three scientific applications from epidemiology, scene rendering, and high energy physics. We evaluate the time and space overheads of reproducing these applications using different sandbox techniques – Parrot, Docker and the Amazon EC2. Our results show that these applications can be reproduced using different sandbox techniques successfully and efficiently, even through the overhead and performance slightly vary.

2. Tracking Execution Environment: Umbrella Spec

```
{
   "description": "A ray-tracing application which creates video frames.",
   "hardware": {
      "arch": "x86_64",
      "cores": "1",
      "memory": "1GB",
      "disk": "3GB"
   },
   "kernel": {
      "name": "linux",
      "version": ">=2.6.18"
   },
   "os": {
      "name": "redhat",
      "version": "6.5",
      "mountpoint": "/",
      "source": [ "http://ccl.cse.nd.edu/.../redhat-6.5-x86_64.tar.gz" ],
      "format": "tgz",
      "action": "unpack",
      "checksum": "669ab5ef94af84d273f8f92a86b7907a",
      "size": "633848940",
      "uncompressed_size": "1743656960",
      "ec2": {
         "ami": "ami-2cf8901c",
         "region": "us-west-2",
         "user": "ec2-user"
      }
   },
   "software": {
      "povray-3.6.1-redhat6-x86_64": {
         "mountpoint": "/software/povray-3.6.1-redhat6-x86_64",
         "source": [ "http://ccl.cse.nd.edu/.../povray-3.6.1-redhat6-x86_64.tar.gz" ],
         "format": "tgz",
         "action": "unpack",
         "checksum": "b02ba86dd3081a703b4b01dc463e0499",
         "size": "1471452",
         "uncompressed_size": "3010560"
      }
   },
   "data": {
      "4_cubes.pov": {
         "mountpoint": "/tmp/4_cubes.pov",
         "source": [ "http://ccl.cse.nd.edu/.../4_cubes.pov" ],
         "format": "plain",
         "action": "none",
         "checksum": "c65266cd2b672854b821ed93028a877a",
         "size": "1757"
      },
       ...
   },
   "environ": {
      "PWD": "/tmp"
   },
   "cmd": "povray +I/tmp/4_cubes.pov +O/tmp/frame000.png +K.0  -H50 -W50",
   "output": {
      "files": [ "/tmp/frame000.png" ],
      "dirs": [ "/tmp/output" ]
   }
}
```

Fig. 1. Umbrella Specification Example – povray.umbrella

3. Creating Execution Environment: Umbrella Execution Engine

Hardware	Kernel	OS	Sandbox Techniques
Yes	Yes	Yes	Utilize the current OS directly (Fig. 3)
Yes	Yes	No	OS-level virtualization (Docker, Parrot) (Fig. 3)
Yes/No	No	No	Hardware Virtualization (VirtualBox, VMWare, EC2) (Fig. 4)

Fig. 2. Sandbox Techniques for Creating Execution Environments

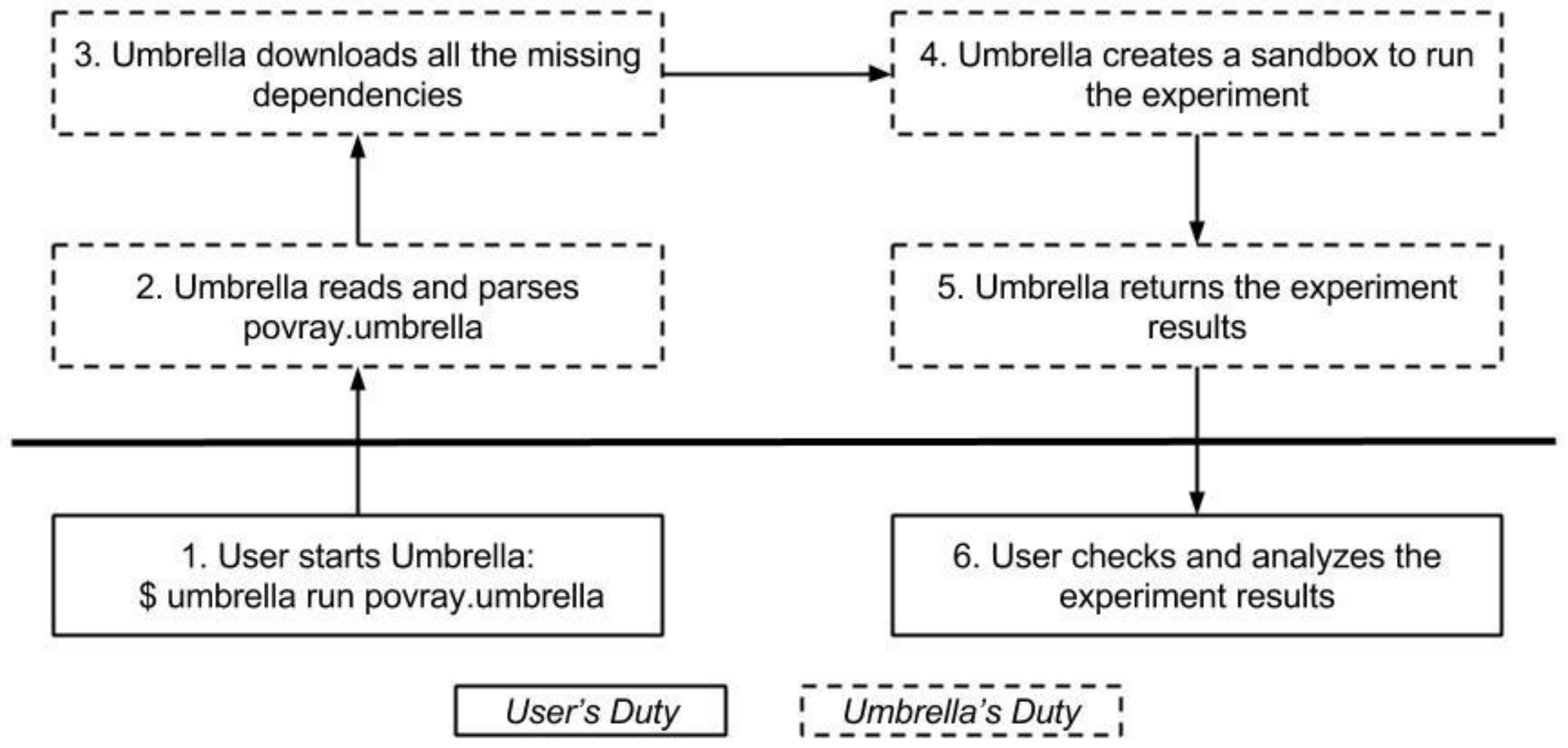

3. Umbrella downloads all the missing dependencies
4. Umbrella creates a sandbox to run the experiment
2. Umbrella reads and parses povray.umbrella
5. Umbrella returns the experiment results
1. User starts Umbrella: $ umbrella run povray.umbrella
6. User checks and analyzes the experiment results
User's Duty
Umbrella's Duty

Fig. 3. Workflow of Umbrella Execution Engine (local)

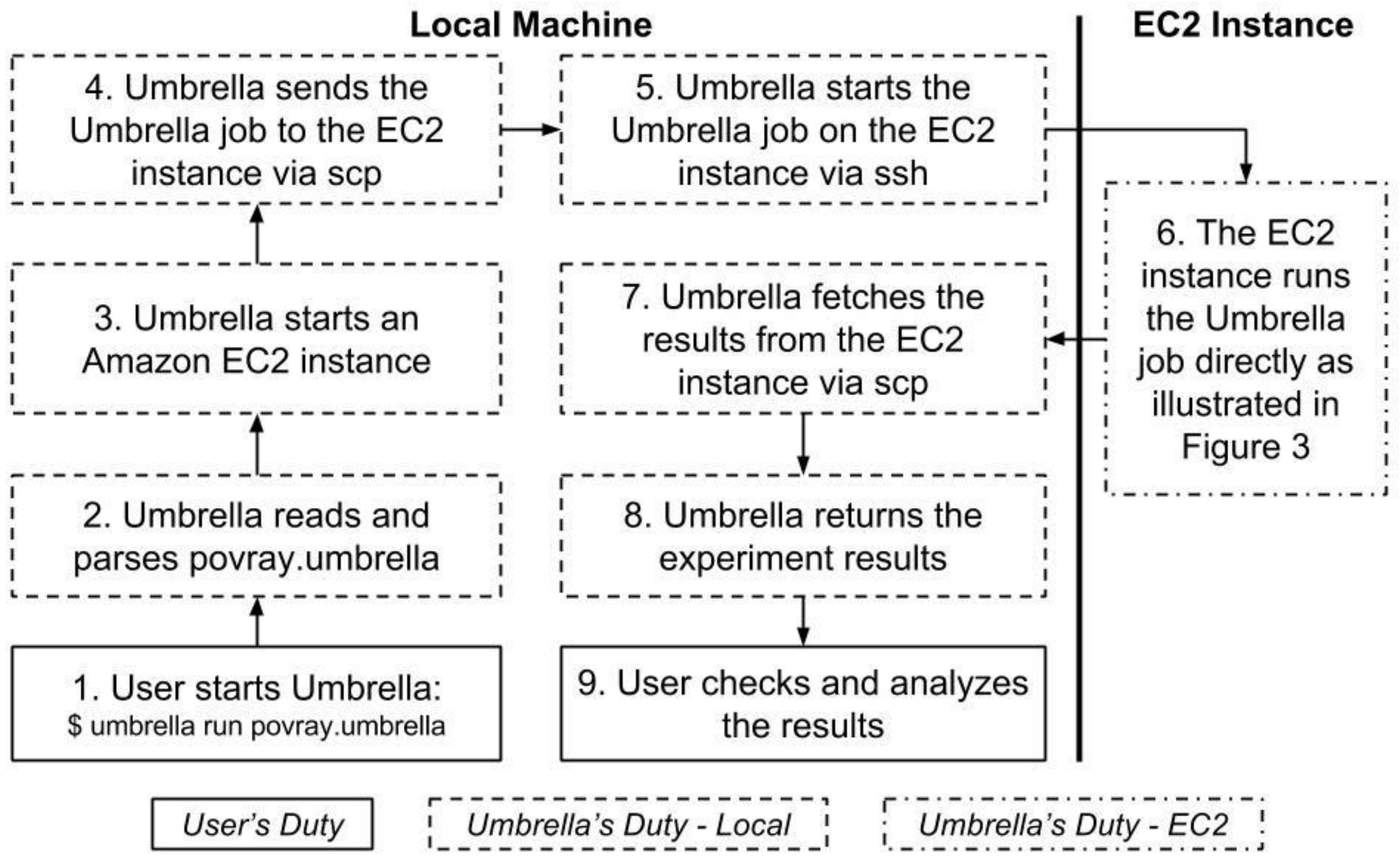

Local Machine
EC2 Instance
4. Umbrella sends the Umbrella job to the EC2 instance via scp
5. Umbrella starts the Umbrella job on the EC2 instance via ssh
6. The EC2 instance runs the Umbrella job directly as illustrated in Figure 3
3. Umbrella starts an Amazon EC2 instance
7. Umbrella fetches the results from the EC2 instance via scp
2. Umbrella reads and parses povray.umbrella
8. Umbrella returns the experiment results
1. User starts Umbrella: $ umbrella run povray.umbrella
9. User checks and analyzes the results
User's Duty
Umbrella's Duty - Local
Umbrella's Duty - EC2

Fig. 4. Workflow of Umbrella Execution Engine (EC2)

4. Preserving Execution Environment: Umbrella Archiver

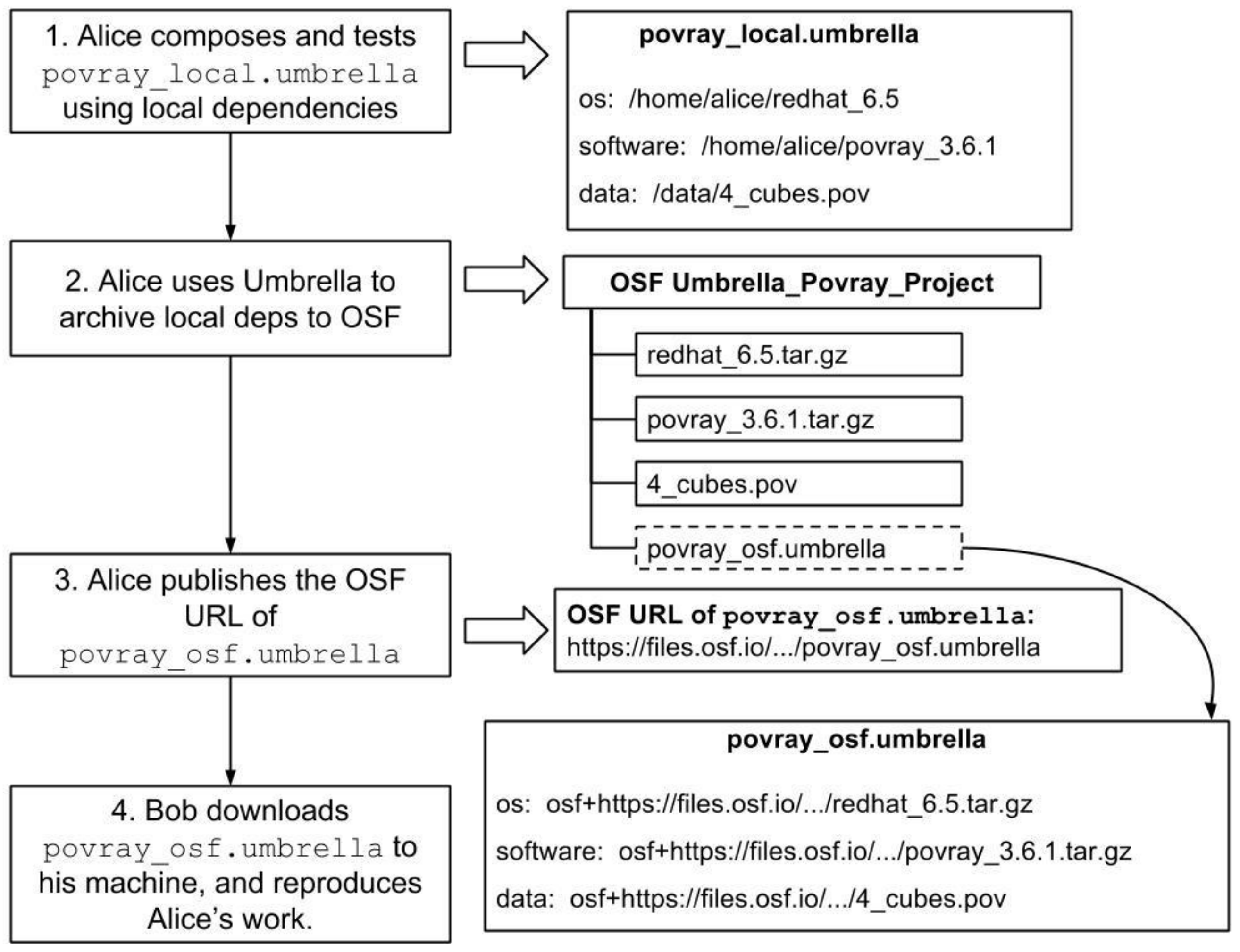

1. Alice composes and tests povray_local.umbrella using local dependencies
povray_local.umbrella
os: /home/alice/redhat_6.5
software: /home/alice/povray_3.6.1
data: /data/4_cubes.pov
2. Alice uses Umbrella to archive local deps to OSF
OSF Umbrella_Povray_Project
redhat_6.5.tar.gz
povray_3.6.1.tar.gz
4_cubes.pov
povray_osf.umbrella
3. Alice publishes the OSF URL of povray_osf.umbrella
OSF URL of povray_osf.umbrella: https://files.osf.io/.../povray_osf.umbrella
povray_osf.umbrella
os: osf+https://files.osf.io/.../redhat_6.5.tar.gz
software: osf+https://files.osf.io/.../povray_3.6.1.tar.gz
data: osf+https://files.osf.io/.../4_cubes.pov
4. Bob downloads povray_osf.umbrella to his machine, and reproduces Alice's work.

Fig. 5. Conducting Reproducible Research Using Umbrella – Local + OSF

4. Preserving Execution Environment: Umbrella Archiver – cont'd

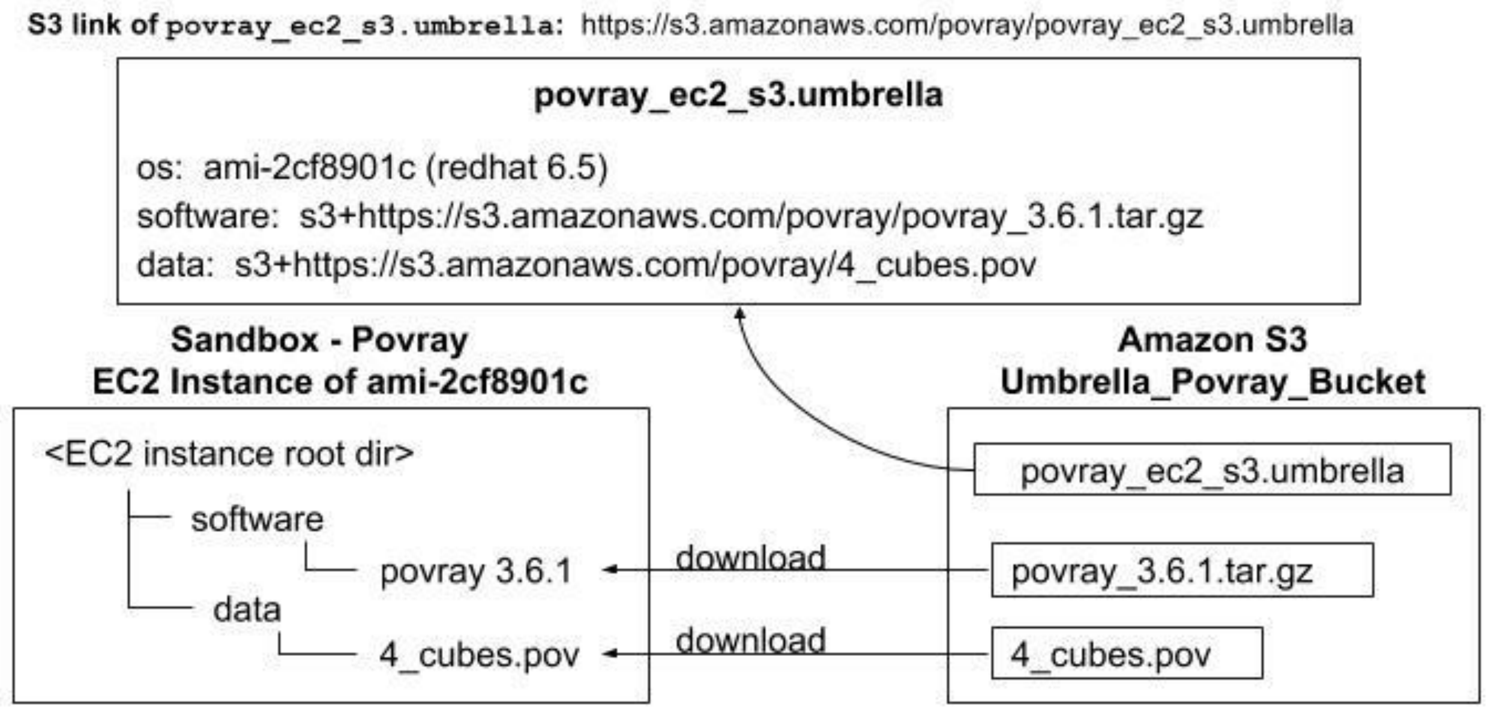

S3 link of povray_ec2_s3.umbrella: https://s3.amazonaws.com/povray/povray_ec2_s3.umbrella
povray_ec2_s3.umbrella
os: ami-2cf8901c (redhat 6.5)
software: s3+https://s3.amazonaws.com/povray/povray_3.6.1.tar.gz
data: s3+https://s3.amazonaws.com/povray/4_cubes.pov
Sandbox - Povray
EC2 Instance of ami-2cf8901c
<EC2 instance root dir>
software
povray 3.6.1
data
4_cubes.pov
download
download
Amazon S3
Umbrella_Povray_Bucket
povray_ec2_s3.umbrella
povray_3.6.1.tar.gz
4_cubes.pov

Fig. 6. Conducting Reproducible Research Using Umbrella – EC2 + S3

5. Evaluation

5.1 Umbrella Specification File Sizes

Application	OpenMalaria	Povray	CMS
Umbrella Spec Size	3.3KB	2.4KB	1.9KB

5.2 Overheads of Creating Execution Environments

Application	OS Deps	Software Deps	Data Deps
OpenMalaria	CentOS 6.6 (69MB/218MB)	openMalaria(2.9MB/13MB) .rpm packages (209MB) epel.repo (<1KB)	.xml (28KB) .csv (<1KB) .xsd (196KB)
Povray	RedHat 6.5 (605MB/1.8GB)	povray (1.5MB/2.9MB)	.pov (1.8KB) .inc (28KB)

CMS	RedHat 6.5 (605MB/1.8GB)	cmssw(1.3GB)
		Parrot(23MB/71MB)
			.sh (<1KB)

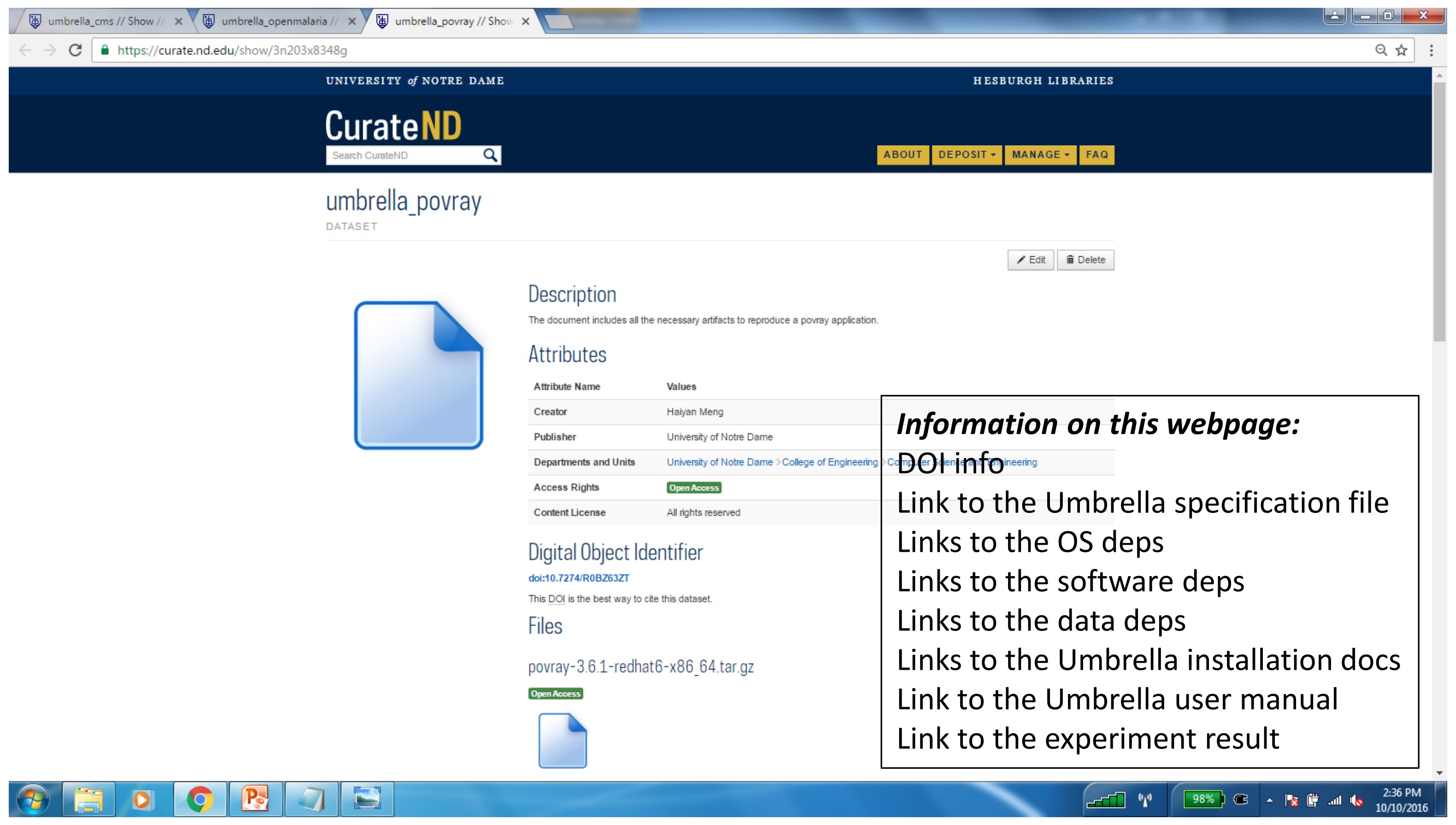
Fig. 7. Size of Application Dependencies

Application	OpenMalaria	Povray	CMS	Permission / Location
Parrot	N/A	65min (2.40GB)	79min (2.39GB)	non-root/local
Docker	57min (1.53GB)	68min (4.11GB)	82min (4.19GB)	root/local
EC2 – m3.medium	113min (225MB)	130min (4.4MB)	211min (94MB)	non-root/remote
EC2 – m3.large	58min (255MB)	65min (4.4MB)	108min (94MB)	non-root/remote

Fig. 8. Time and Space Overheads of Creating Execution Environments

5.3 Last Step to Enhance Reproducibility - DOI

Application	DOI URL
OpenMalaria	http://dx.doi.org/doi:10.7274/R03F4MH3
Povray	http://dx.doi.org/doi:10.7274/R0BZ63ZT
CMS	http://dx.doi.org/doi:10.7274/R0765C7T



umbrella_povray
DATASET

Description
The document includes all the necessary artifacts to reproduce a povray application.

Attributes

Attribute Name	Values
Creator	Haiyan Meng
Publisher	University of Notre Dame
Departments and Units	University of Notre Dame › College of Engineering › Computer Science and Engineering
Access Rights	Open Access
Content License	All rights reserved

Digital Object Identifier
doi:10.7274/R0BZ63ZT
This DOI is the best way to cite this dataset.

Files
povray-3.6.1-redhat6-x86_64.tar.gz

Information on this webpage:
DOI info
Link to the Umbrella specification file
Links to the OS deps
Links to the software deps
Links to the data deps
Links to the Umbrella installation docs
Link to the Umbrella user manual
Link to the experiment result

DASPOS Data and Software Preservation for Open Science

DASPOS: www.daspos.org Cooperative Computing Lab: ccl.cse.nd.edu

This work was supported in part by National Science Foundation grants PHY-1247316 (DASPOS), OCI-1148330 (SI2) and PHY-1312842.

The University of Notre Dame Center for Research Computing scientists and engineers provided critical technical assistance throughout this research effort.