DASPOS Data and Software Preservation for Open Science

UNIVERSITY OF NOTRE DAME

# 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, Indiana, USA

October 2016

## Observation: it is difficult to reproduce the experiment results published in academic papers!

Alice did the experiments for her paper:

**server**: lab01.phy.research.org

1) installed software deps (i.e., **sim_sort**) under /home/alice/software
2) configured environment variables (**SIMCOUNT**)
3) wrote the analysis script, **analysis.py**
   /usr/bin/python --> python2.7
4) downloaded the datasets to /home/alice/data

Experiment results -> Figures

Submitted the paper, and it got accepted.

Several months later, Bob read the paper and emailed Alice to ask for help to reproduce the experiment.

Alice searched for analysis.py and sent it to Bob.

**Problems Bob encountered:** 😲 😟

- analysis.py depends on the setting of the environment variable SIMCOUNT
- analysis.py expects an input file located at /home/alice/data/file1
- analysis.py attempts to utilize an executable named sim_sort
- the output of analysis.py overflows Bob's memory and disk
- /usr/bin/python on Bob's machine is Python 3.0, which is not backwards compatible with Python 2.7.

- Alice forgot to preserve the SIMCOUNT setting.
- Alice deleted the directory /home/alice/data by accident.
- sim_sort is under version control via Git and can be found, however, Alice forgot the commit id used.
- As for the memory and disk overflow, Alice realized she should have told Bob the experiment requires 6GB memory and 20GB disk space.

Sysadmins update kernel, OS, system software periodically

Hardware upgrade every several years

Network resources from third-party websites

….

**Experiment results can NOT be reproduced by others or even the original author!**

## Lessons

- 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.

- The configurations of the execution environments are too complex to be described easily by authors.

  hardware, kernel, OS, software, data, environ vars

# A Framework for Conducting Reproducible Research

- **Tracking execution environments**

allows the user to specify all the necessary details about a comprehensive execution environment

- **Creating execution environments**

sandbox techniques like VMs, Linux Containers (i.e., Docker) and user-space tracers (i.e., Parrot)

- **Preserving execution environments**

archives data and software deps in the first place into persistent storage services (i.e., Amazon S3)

# Tracking Execution Environments: Umbrella Specification

**Sections:**

hardware kernel
os software data
environ cmd output
description ….

**os/software/data sections:**

source
checksum
size
format
mountpoint

```
{
  "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" ]
  }
}
```

# Resource URLs Supported by Umbrella

| Resource | Example URL |
|---|---|
| Local Filesystem | /home/hmeng/data/input |
| HTTP | http://www.data.com/data/file1 |
| HTTPS | https://lab01.nd.edu/data/hep/file2 |
| Amazon S3 | s3+https://s3.aws.com/…/cubes.pov |
| Open Science Framework (OSF) | osf+https://files.osf.io/v1/…/7559c3a |
| Git Repository | git+https://github.com/…/cctools.git |
| CernVM File System | cvmfs://cvmfs/cms.cern.ch |

## Creating Execution Environment: Umbrella Execution Engine

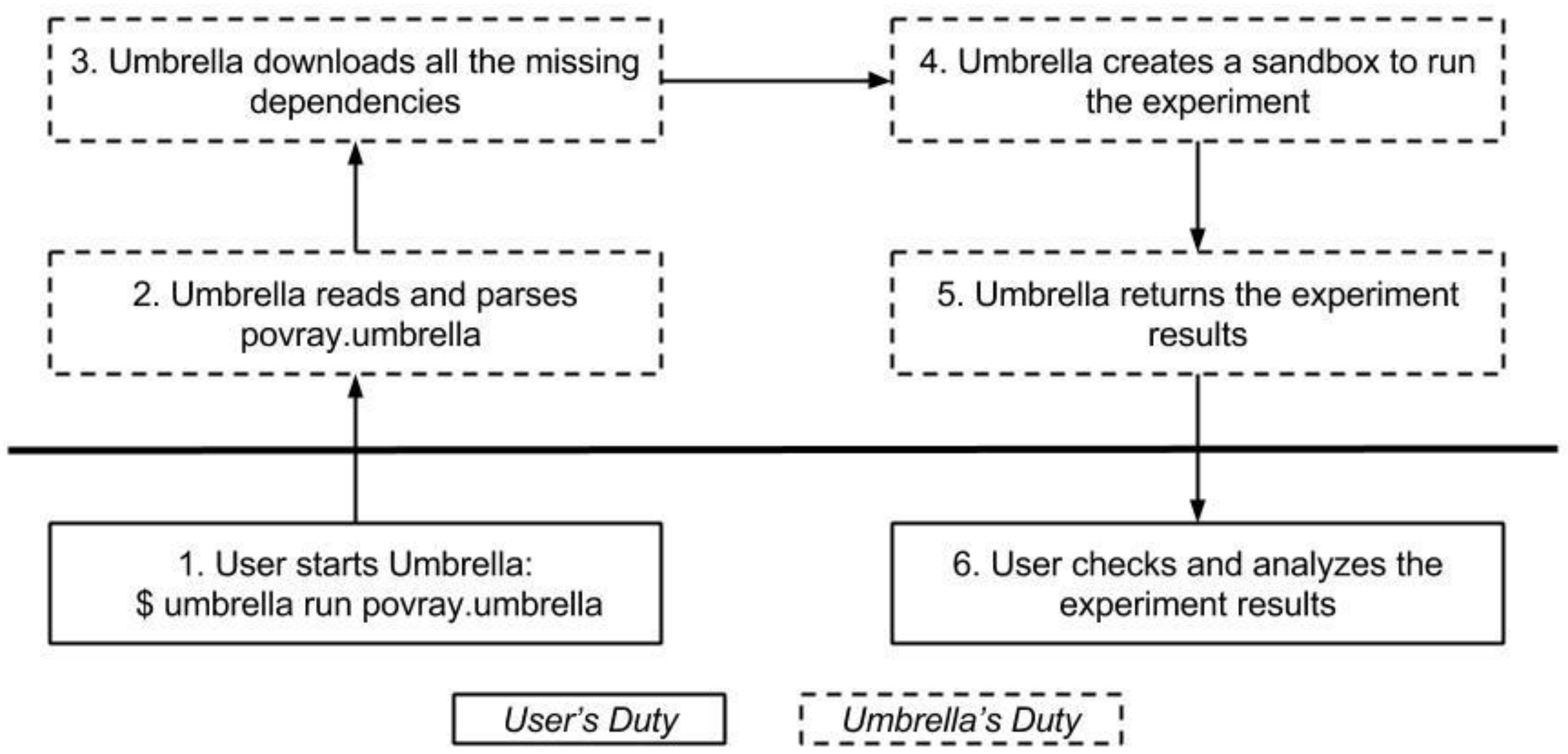
**Matching degree** between

-- the execution node

-- the specified execution environment

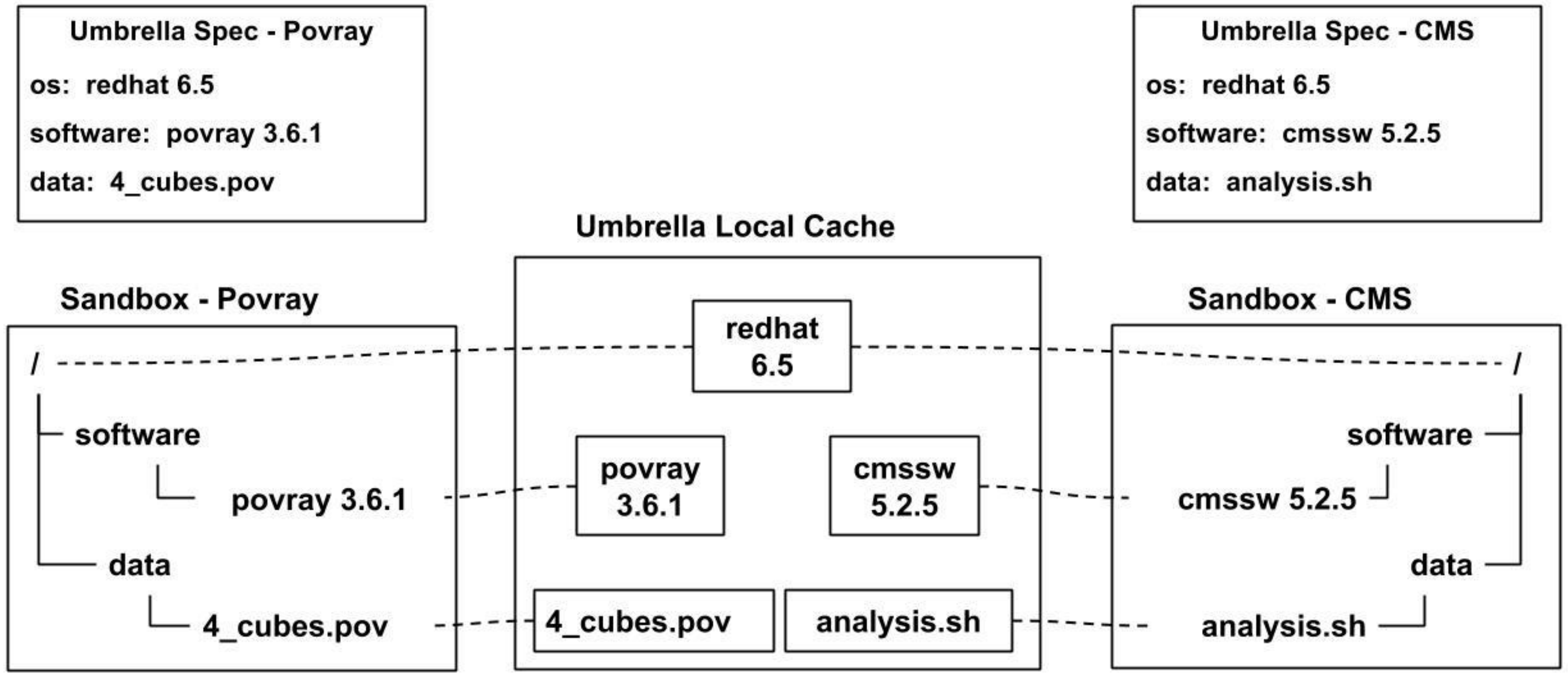
| Hardware | Kernel | OS | Sandbox Techniques |
|---|---|---|---|
| Yes | Yes | Yes | Utilize the current OS directly |
| Yes | Yes | No | OS-level Virtualization<br>Docker, Parrot |
| Yes/No | No | No | Hardware Virtualization<br>Local: VirtualBox, VMWare<br>Remote: Amazon EC2 |

# Umbrella Execution Engine - Local


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

# Umbrella Local Cache

- OS-level virtualization



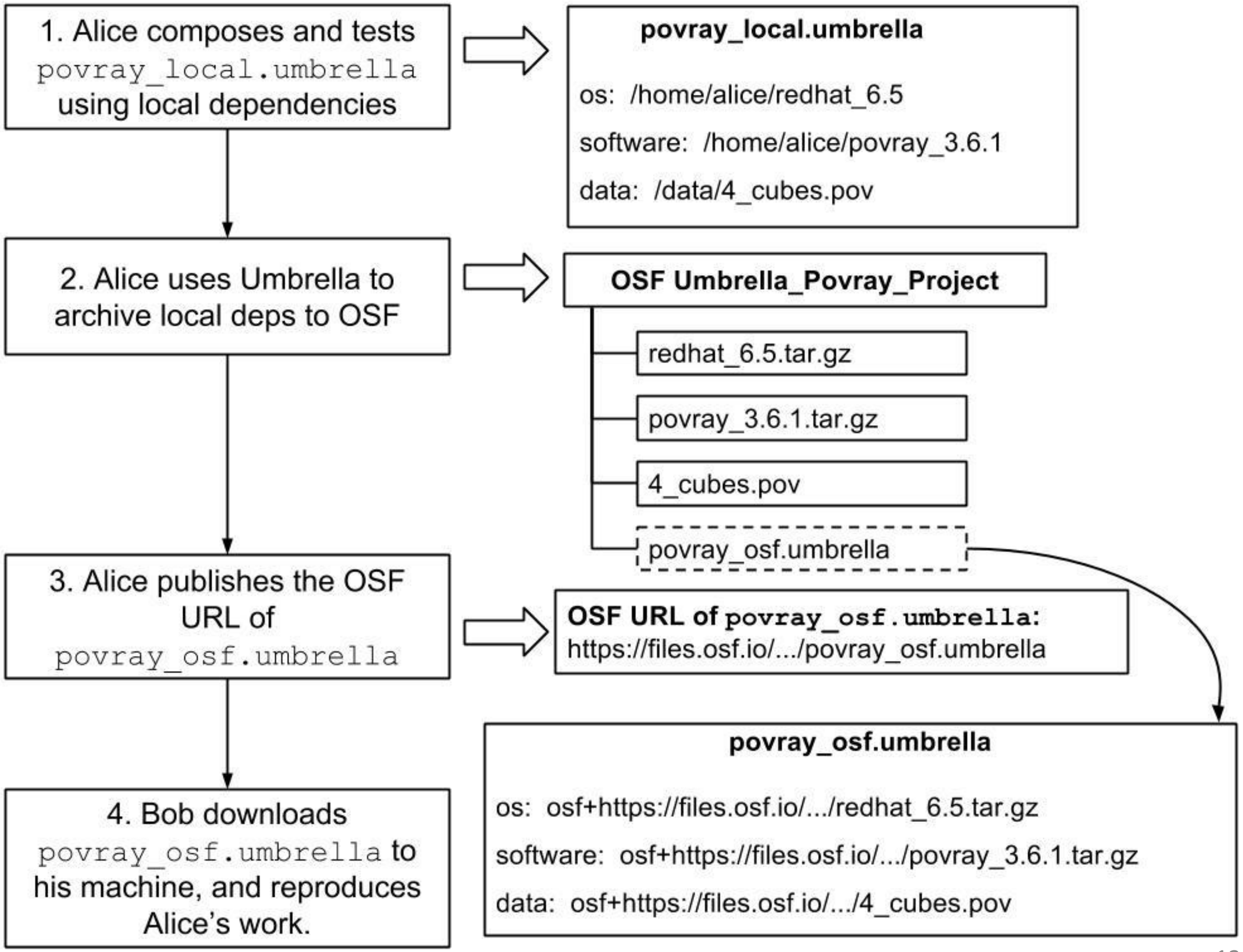
# Preserving Execution Environment: Umbrella Archiver

- Uploads the deps into persistent storage services
  - Amazon S3
  - OSF storage service
- Allows the user to mark unreliable deps

  Local dependencies

  Some third-party network dependencies
- Allows the user to set the access permission of uploaded resources

# How Our Framework can Help Alice and Bob?



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

## Evaluation

Umbrella – Python 2.6

Execution mode: Parrot, Docker, EC2

We evaluate our framework via three scientific applications:

- Epidemiology - OpenMalaria
- Scene Rendering - Povray
- High Energy Physics - CMS

## Umbrella Specification File Sizes:

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

## Sizes of os/software/data Dependencies of the Evaluated Applications:

| Application | OS Deps | Software Deps | Data Deps |
|---|---|---|---|
| OpenMalaria | CentOS 6.6 (69MB/218MB) | openMalaria(2.9MB/13MB)<br>.rpm packages (209MB)<br>epel.repo (<1KB) | .xml (28KB)<br>.csv (<1KB)<br>.xsd (196KB) |
| Povray | RedHat 6.5 (605MB/1.8GB) | povray (1.5MB/2.9MB) | .pov (1.8KB)<br>.inc (28KB) |
| CMS | RedHat 6.5 (605MB/1.8GB) | cmssw(1.3GB)<br>Parrot(23MB/71MB) | .sh (<1KB) |

## Sizes of os/software/data Dependencies of the Evaluated Applications:

| Application | OS Deps | Software Deps | Data Deps |
|---|---|---|---|
| OpenMalaria | CentOS 6.6 (69MB/218MB) | openMalaria(2.9MB/13MB)<br>.rpm packages (209MB)<br>epel.repo (<1KB) | .xml (28KB)<br>.csv (<1KB)<br>.xsd (196KB) |
| Povray | RedHat 6.5 (605MB/1.8GB) | povray (1.5MB/2.9MB) | .pov (1.8KB)<br>.inc (28KB) |
| CMS | RedHat 6.5 (605MB/1.8GB) | cmssw(1.3GB)<br>Parrot(23MB/71MB) | .sh (<1KB) |

## Overheads of Creating Execution Environments:

| 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 |

The parrot and docker sandbox modes are tested on the same machine:
hardware: x86 64 kernel: Linux 2.6.32 OS: RedHat 6.7

| Application | OS Deps | Software Deps | Data Deps |
|---|---|---|---|
| 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) |

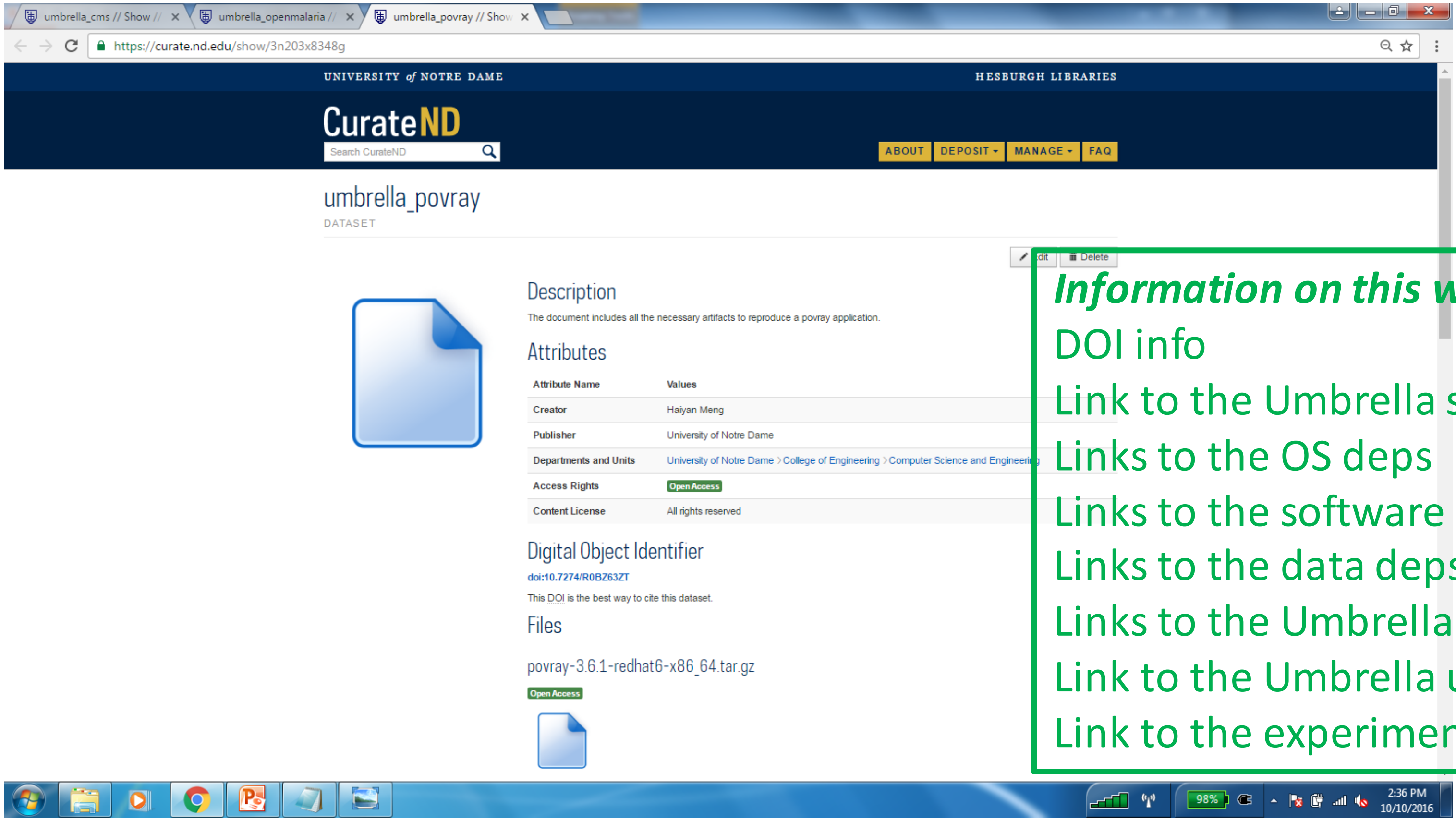
## Effectiveness of Umbrella Local Cache:

| Application (Deps Size) | Cache Size | Delta (Newly Added Deps) | Time |
|---|---|---|---|
| CMS (2.39GB) | 2.39GB | 2.39GB (all deps) | 79min |
| CMS - rerun | 2.39GB | 0 | 78min |
| Povray (2.40GB) | 2.40GB | 4.4MB (software and data deps) | 64min |
| Povray - rerun | 2.40GB | 0 | 64min |
| Povray – new software deps | 2.40GB | 4.4MB (software deps) | 64min |
| Povray – new data deps | 2.40GB | 28KB (data deps) | 64min |

The initial size of the Umbrella local cache is 0.
All the tests here were done with the parrot sandbox mode on the same machine:
hardware: x86 64 kernel: Linux 2.6.32 OS: RedHat 6.7

# 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

Open Access

***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

## Summary

**A Framework for Conducting Reproducible Research:**

- **Tracking execution environments** (Umbrella Specification)

Lightweight, persistent and deployable execution environment specs

Easily shared, expanded, and repurposed

- **Creating execution environments** (Umbrella Execution Engine)

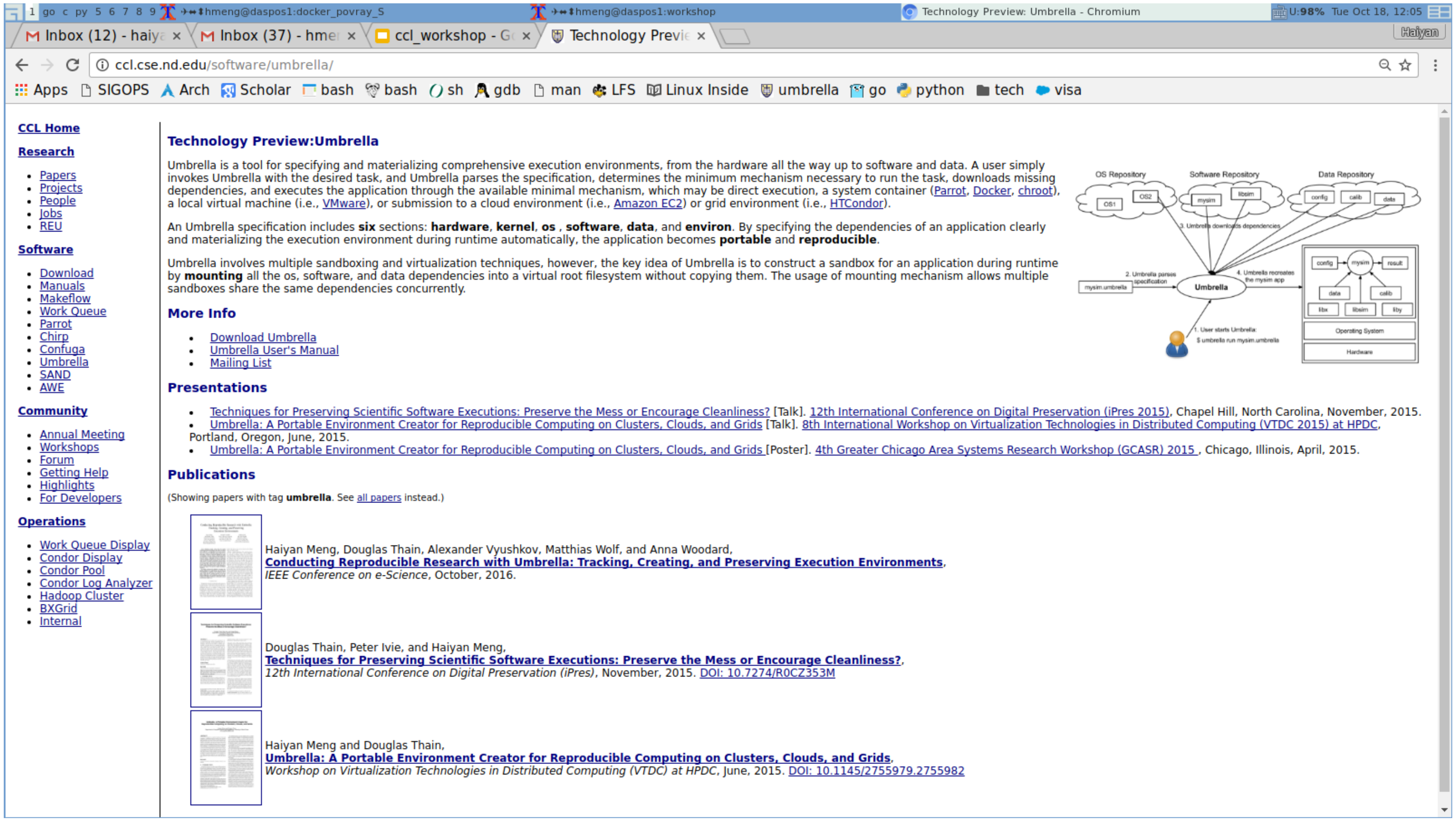
(re)create execution environments using sandbox techniques like VM, Docker and Parrot.

- **Preserving execution environments** (Umbrella Archiver)

persistent storage services like Amazon S3 and OSF

**tracking the execution environments as the research process goes**

## Umbrella: http://ccl.cse.nd.edu/software/umbrella/


1 go c py 5 6 7 8 9 hmeng@daspos1:docker_povray_S
hmeng@daspos1:workshop
Technology Preview: Umbrella - Chromium
U:98% Tue Oct 18, 12:05
Inbox (12) - haiy
Inbox (37) - hme
ccl_workshop - G
Technology Previe
Haiyan
ccl.cse.nd.edu/software/umbrella/
Apps SIGOPS Arch Scholar bash bash sh gdb man LFS Linux Inside umbrella go python tech visa

CCL Home

Research

- Papers
- Projects
- People
- Jobs
- REU

Software

- Download
- Manuals
- Makeflow
- Work Queue
- Parrot
- Chirp
- Confuga
- Umbrella
- SAND
- AWE

Community

- Annual Meeting
- Workshops
- Forum
- Getting Help
- Highlights
- For Developers

Operations

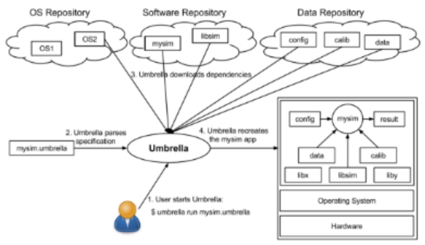
- Work Queue Display
- Condor Display
- Condor Pool
- Condor Log Analyzer
- Hadoop Cluster
- BXGrid
- Internal

**Technology Preview:Umbrella**

Umbrella is a tool for specifying and materializing comprehensive execution environments, from the hardware all the way up to software and data. A user simply invokes Umbrella with the desired task, and Umbrella parses the specification, determines the minimum mechanism necessary to run the task, downloads missing dependencies, and executes the application through the available minimal mechanism, which may be direct execution, a system container (Parrot, Docker, chroot), a local virtual machine (i.e., VMware), or submission to a cloud environment (i.e., Amazon EC2) or grid environment (i.e., HTCondor).

An Umbrella specification includes **six** sections: **hardware**, **kernel**, **os** , **software**, **data**, and **environ**. By specifying the dependencies of an application clearly and materializing the execution environment during runtime automatically, the application becomes **portable** and **reproducible**.

Umbrella involves multiple sandboxing and virtualization techniques, however, the key idea of Umbrella is to construct a sandbox for an application during runtime by **mounting** all the os, software, and data dependencies into a virtual root filesystem without copying them. The usage of mounting mechanism allows multiple sandboxes share the same dependencies concurrently.



**More Info**

- Download Umbrella
- Umbrella User's Manual
- Mailing List

**Presentations**

- Techniques for Preserving Scientific Software Executions: Preserve the Mess or Encourage Cleanliness? [Talk]. 12th International Conference on Digital Preservation (iPres 2015), Chapel Hill, North Carolina, November, 2015.
- Umbrella: A Portable Environment Creator for Reproducible Computing on Clusters, Clouds, and Grids [Talk]. 8th International Workshop on Virtualization Technologies in Distributed Computing (VTDC 2015) at HPDC, Portland, Oregon, June, 2015.
- Umbrella: A Portable Environment Creator for Reproducible Computing on Clusters, Clouds, and Grids [Poster]. 4th Greater Chicago Area Systems Research Workshop (GCASR) 2015 , Chicago, Illinois, April, 2015.

**Publications**

(Showing papers with tag **umbrella**. See all papers instead.)

Haiyan Meng, Douglas Thain, Alexander Vyushkov, Matthias Wolf, and Anna Woodard,
**Conducting Reproducible Research with Umbrella: Tracking, Creating, and Preserving Execution Environments**,
*IEEE Conference on e-Science*, October, 2016.

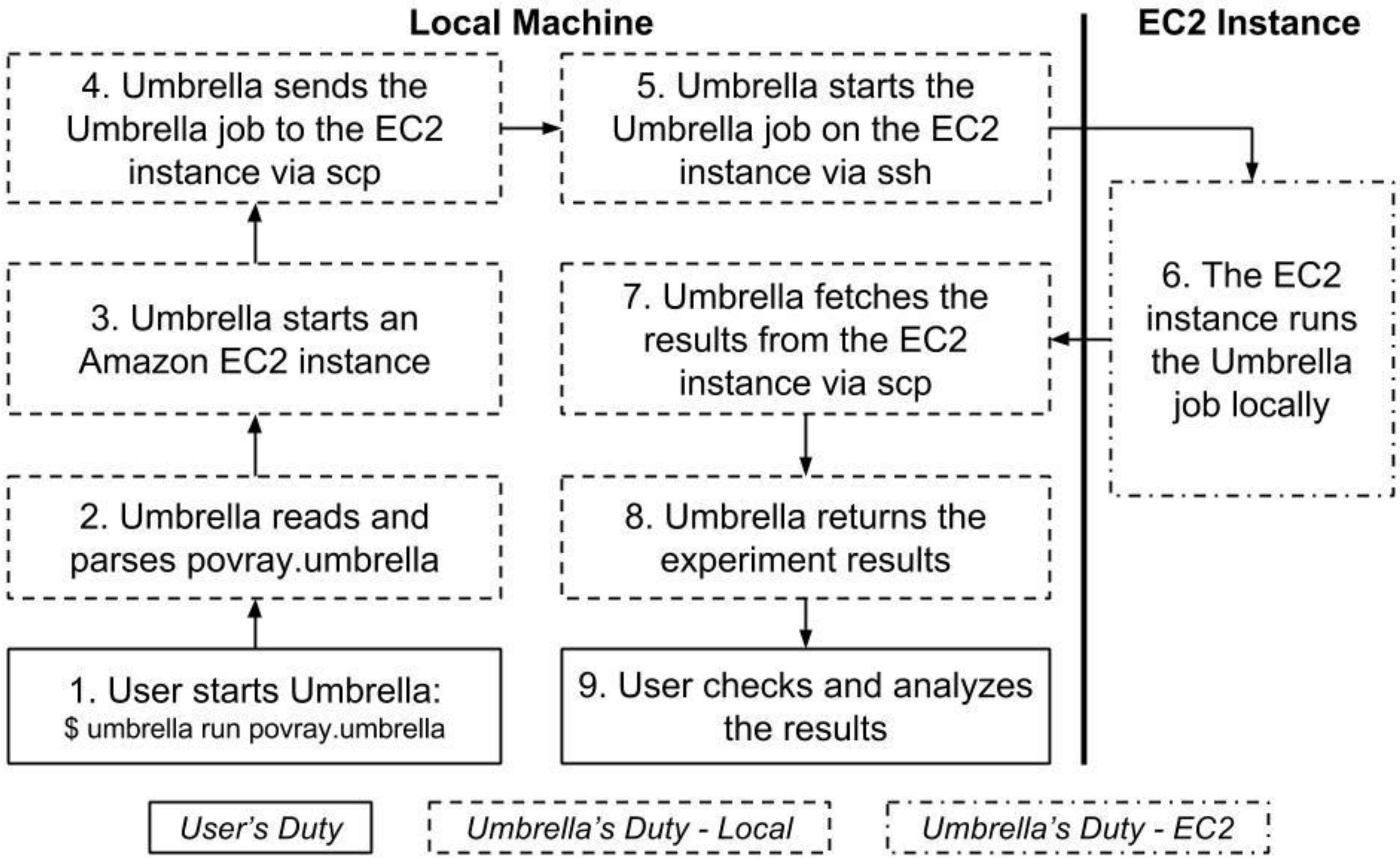
Douglas Thain, Peter Ivie, and Haiyan Meng,
**Techniques for Preserving Scientific Software Executions: Preserve the Mess or Encourage Cleanliness?**,
*12th International Conference on Digital Preservation (iPres)*, November, 2015. DOI: 10.7274/R0CZ353M

Haiyan Meng and Douglas Thain,
**Umbrella: A Portable Environment Creator for Reproducible Computing on Clusters, Clouds, and Grids**,
*Workshop on Virtualization Technologies in Distributed Computing (VTDC) at HPDC*, June, 2015. DOI: 10.1145/2755979.2755982

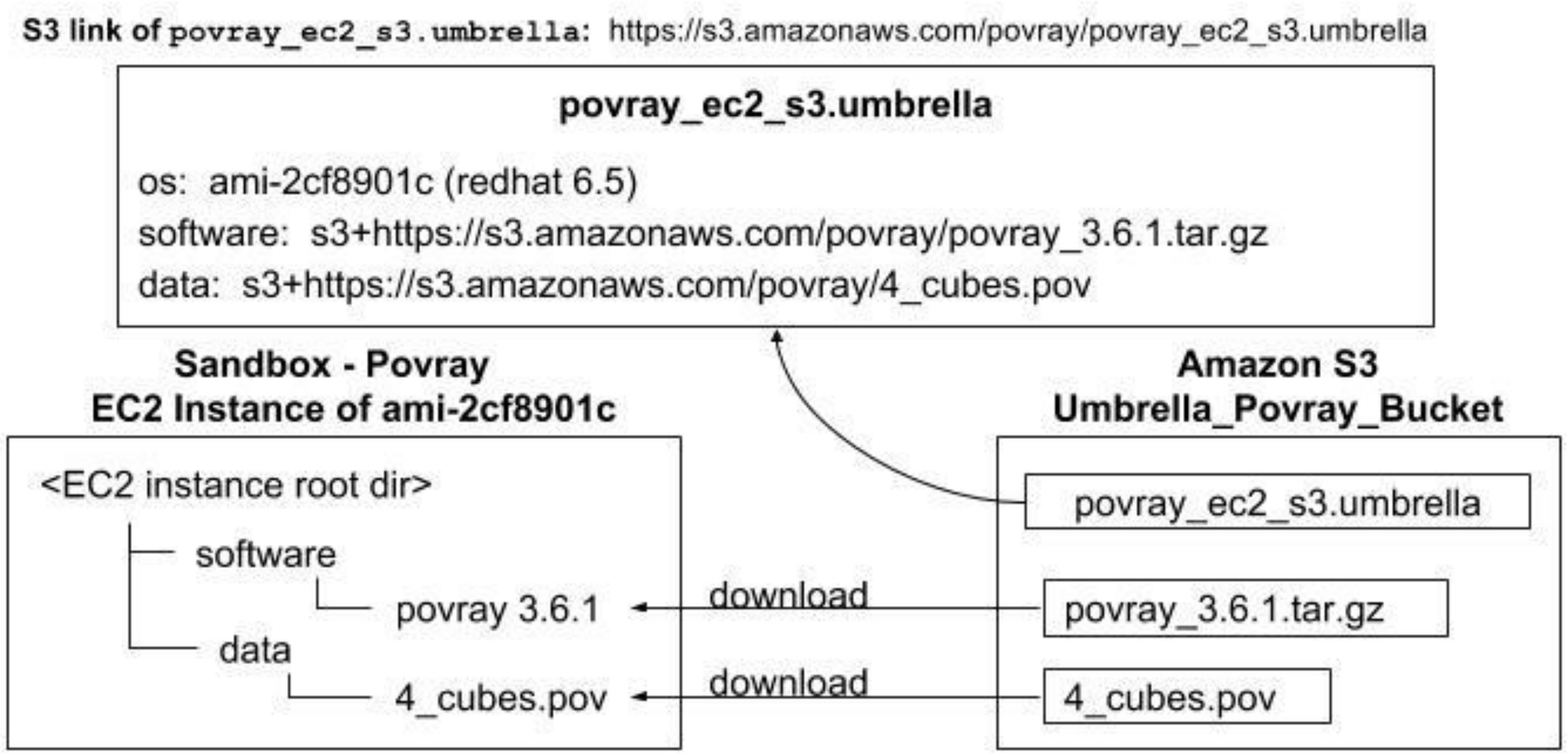
Name: Haiyan Meng Email: hmeng@nd.edu

***Questions?***

# Umbrella Execution Engine – EC2


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
3. Umbrella starts an Amazon EC2 instance
7. Umbrella fetches the results from the EC2 instance via scp
6. The EC2 instance runs the Umbrella job locally
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

## How Our Framework can Help Alice and Bob?


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