PRUNE: A Preserving Run Environment for Reproducible Scientific Computing

CCTools

UNIVERSITY OF NOTRE DAME

Reproducibility

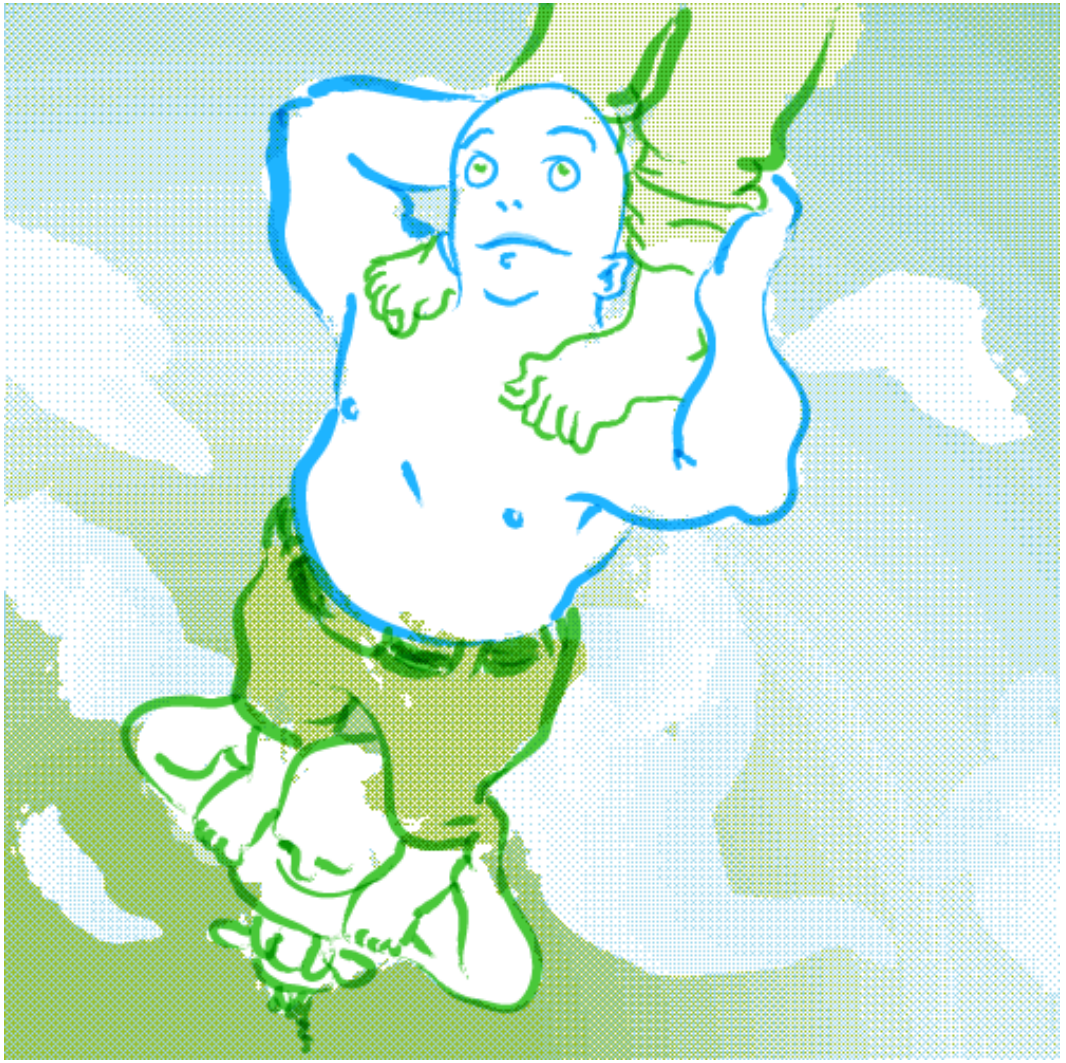
- "[An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions]" –Jon Claerbout

Verify **and Extend**

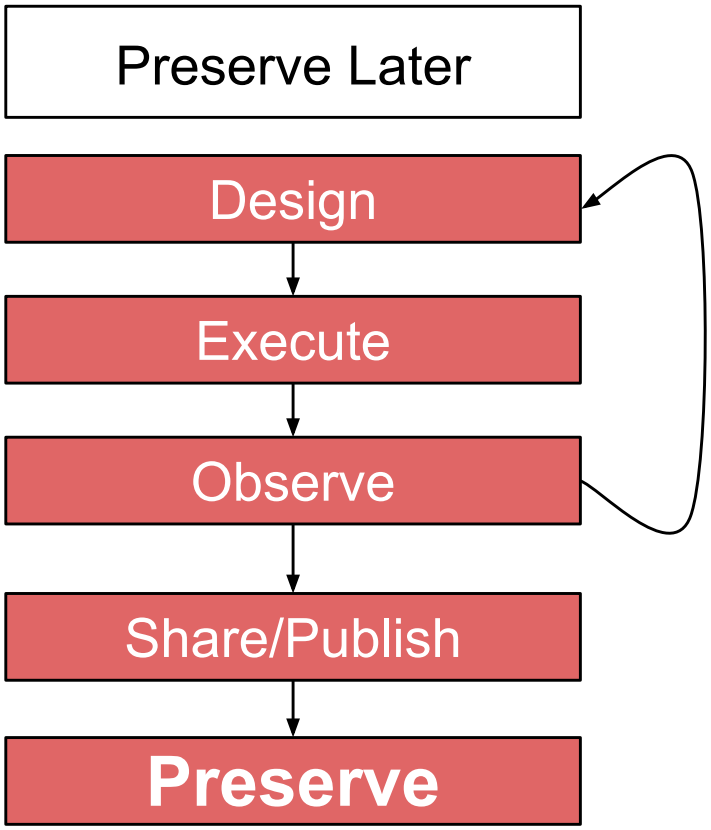
CCTools

UNIVERSITY OF NOTRE DAME

- Don’t re-invent the wheel
- Stand on the shoulders of giants



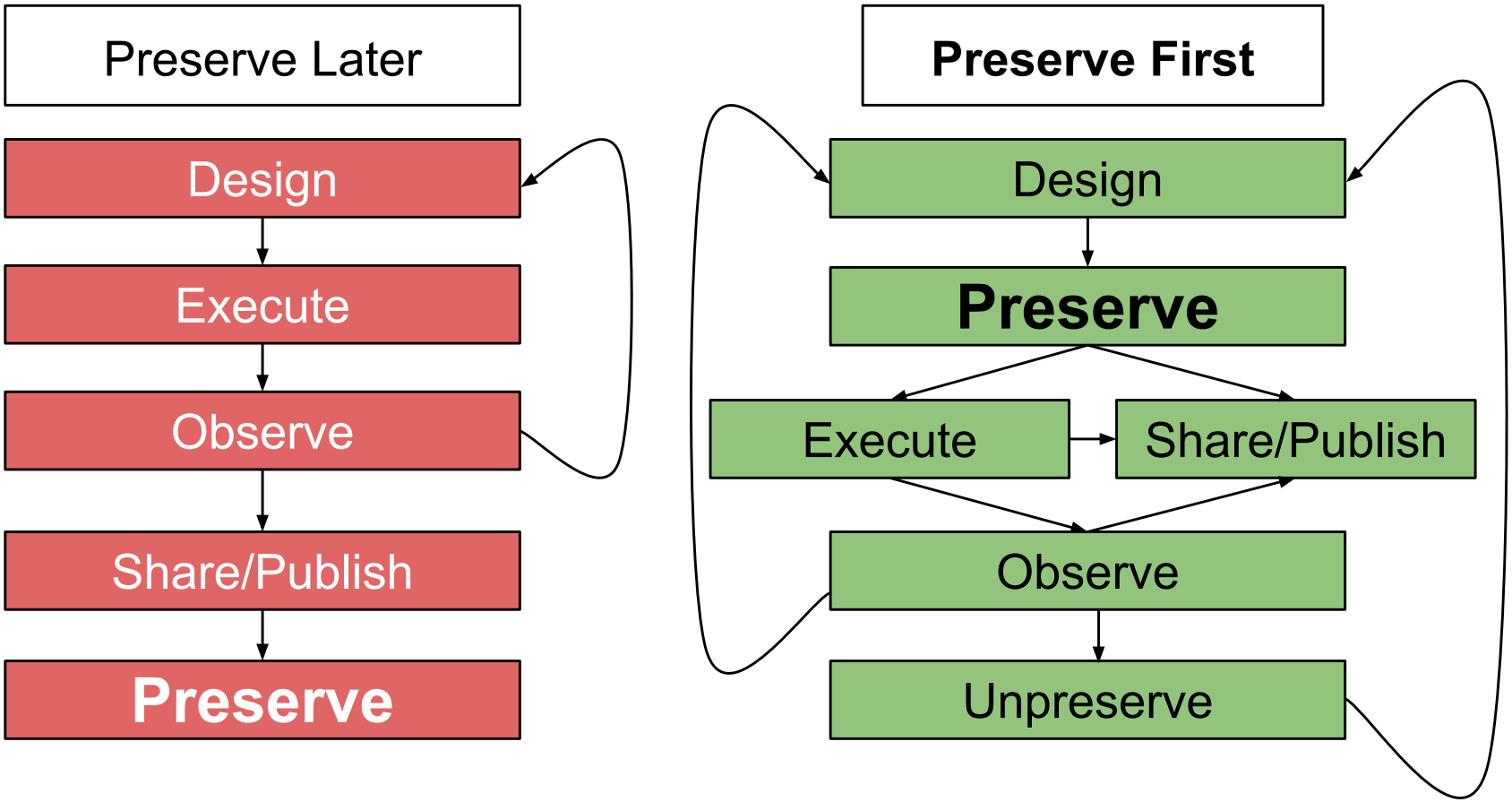
Accepted philosphy



- Libraries
- Hardware
- Network

- System Administrators
- Remote Collaborators
- Graduated Students

Proposed philosophy

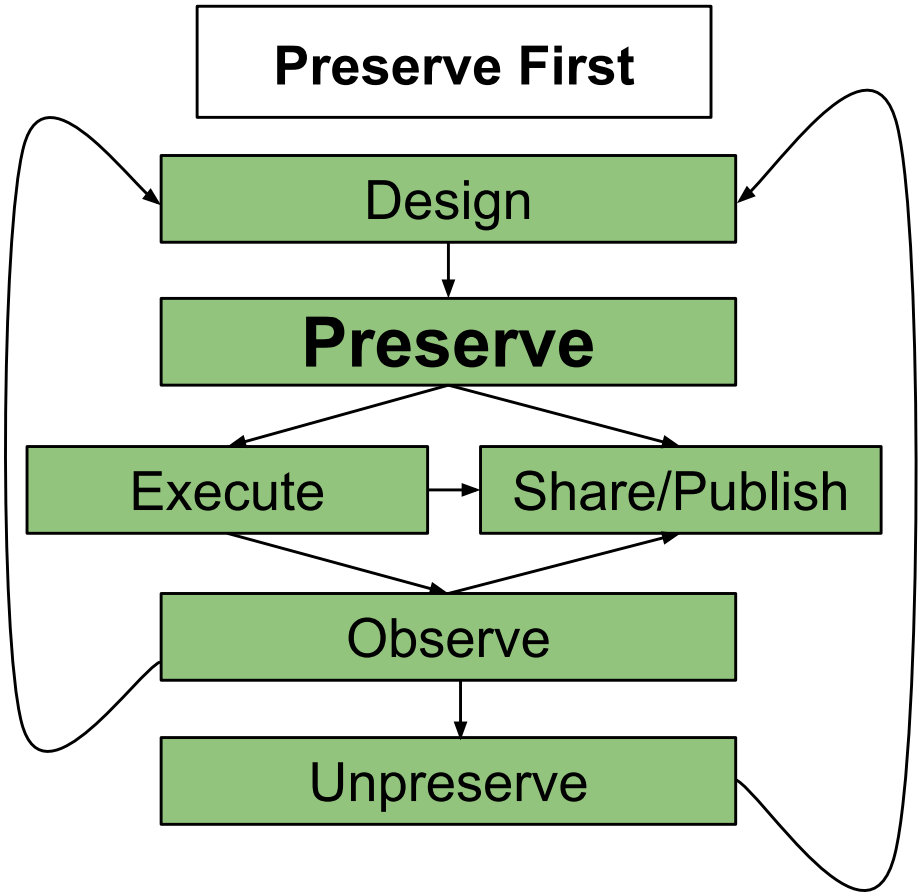


Differences

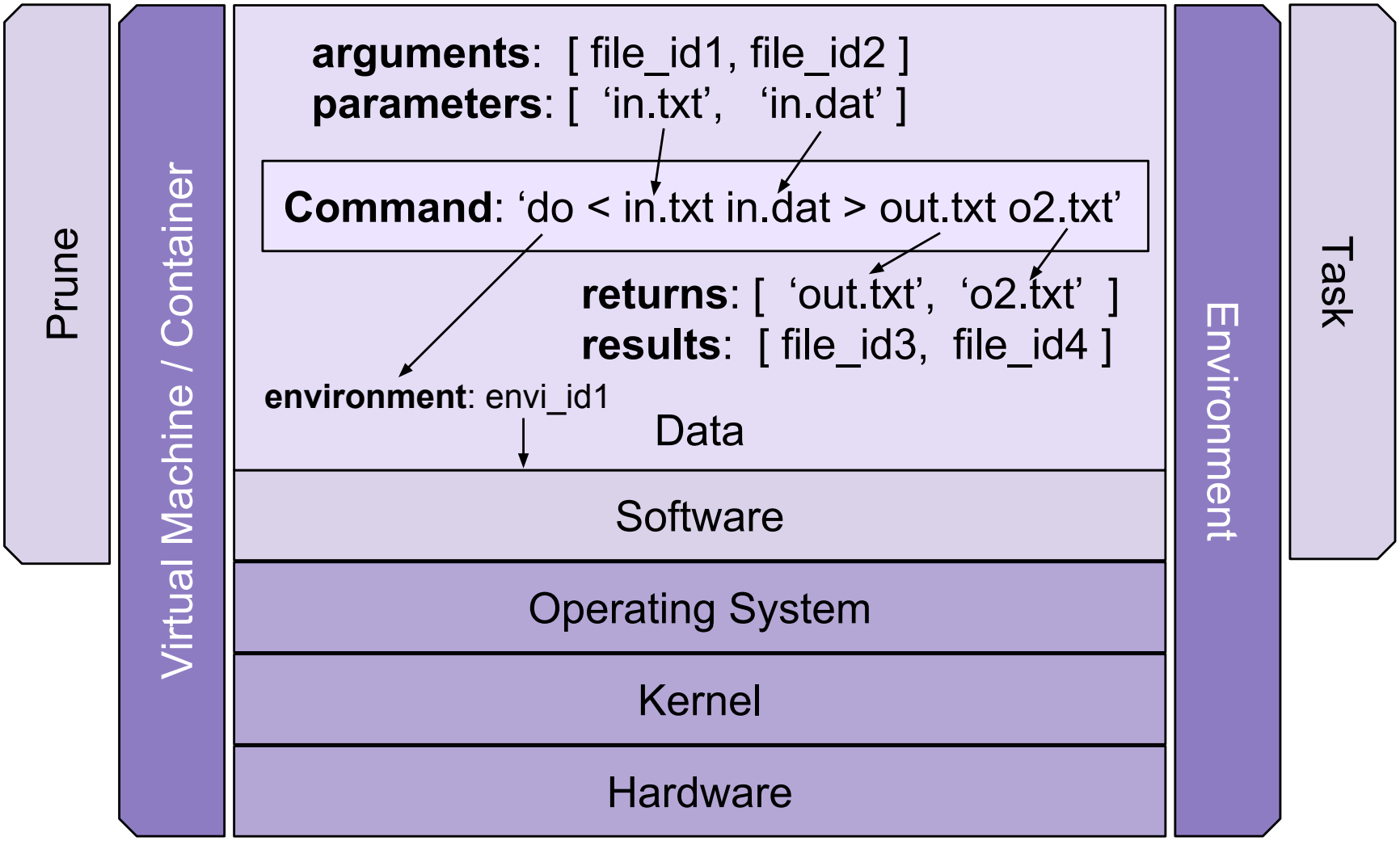
- ~~Git: User decides when to preserve~~
- Preserve ALL specification changes

- ~~Git: Code Commits separate from Code Execution~~
- System Manages ALL computation

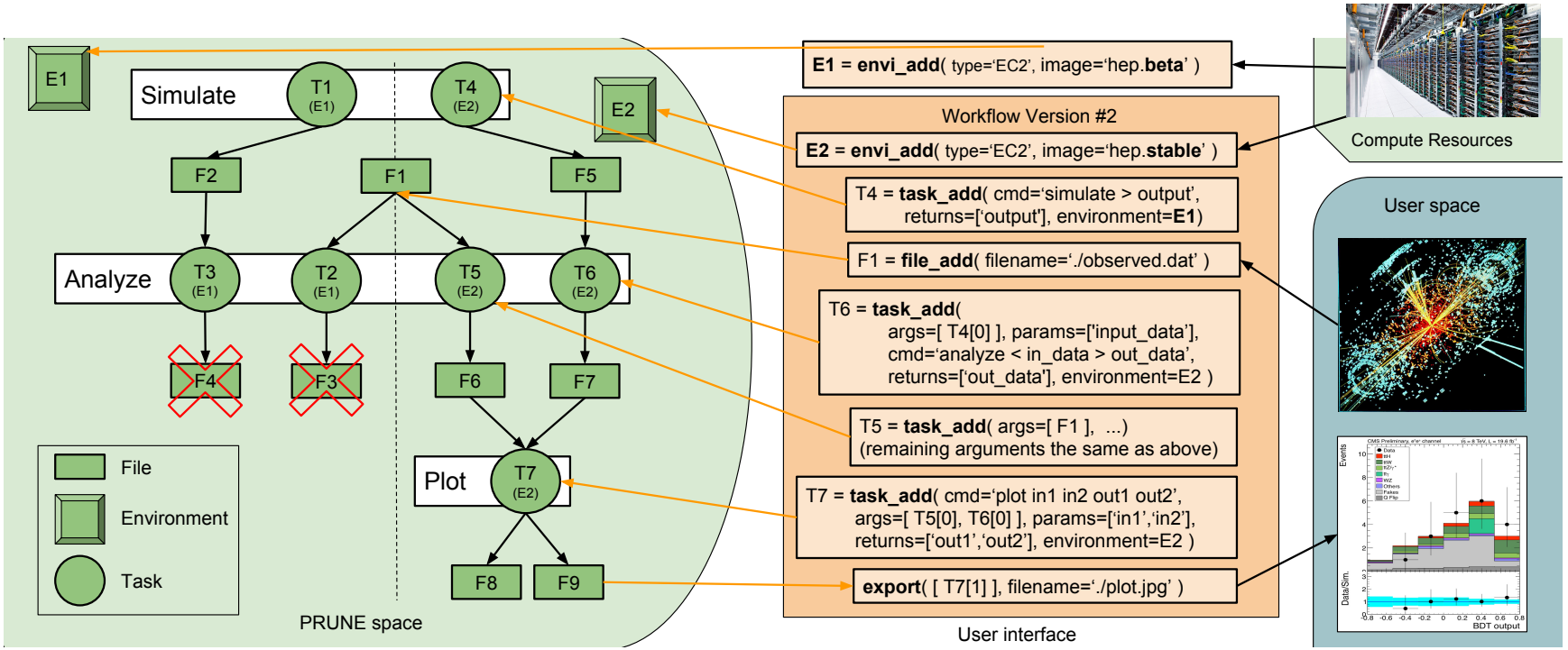
- Remove unneeded code later on



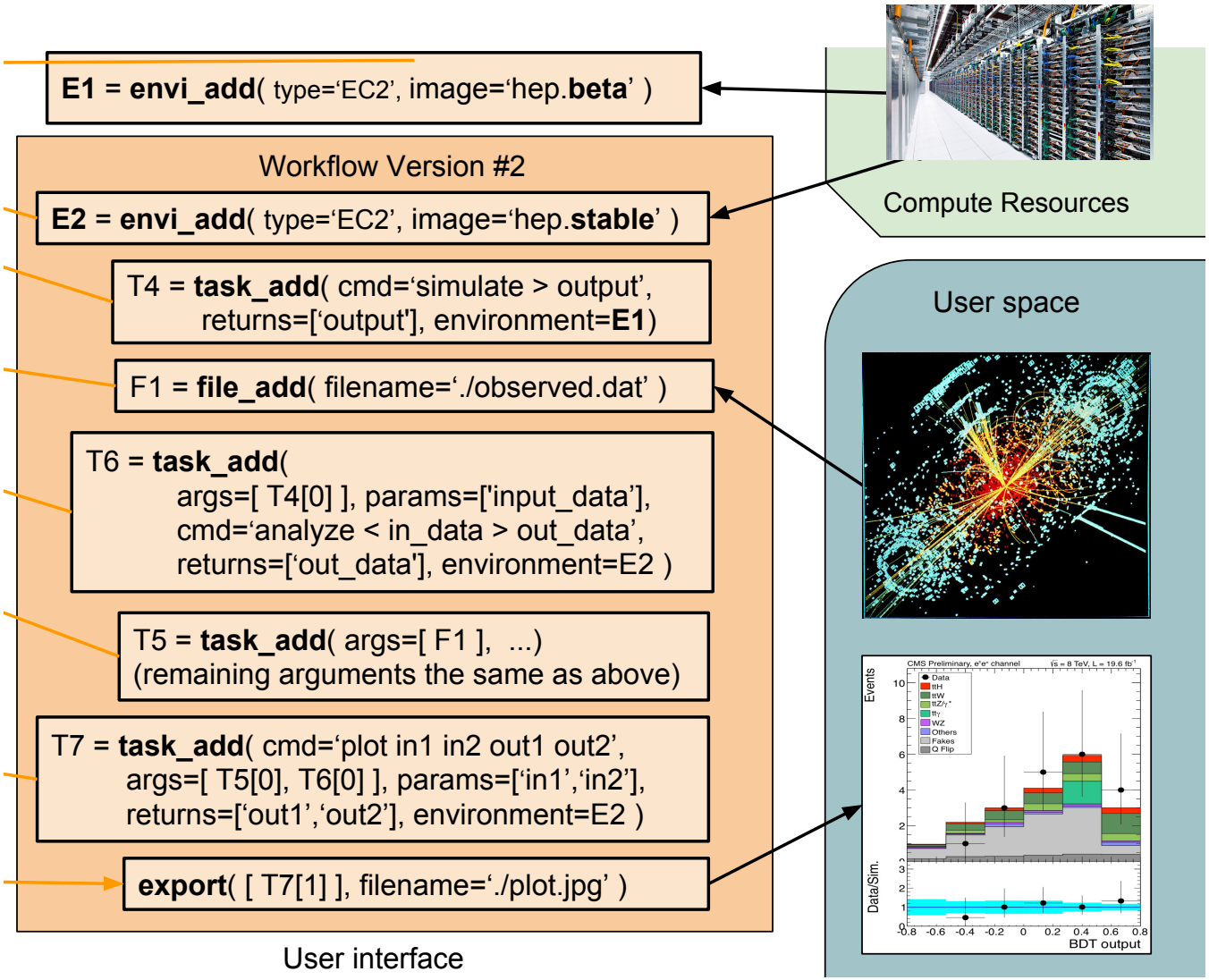
What to Preserve



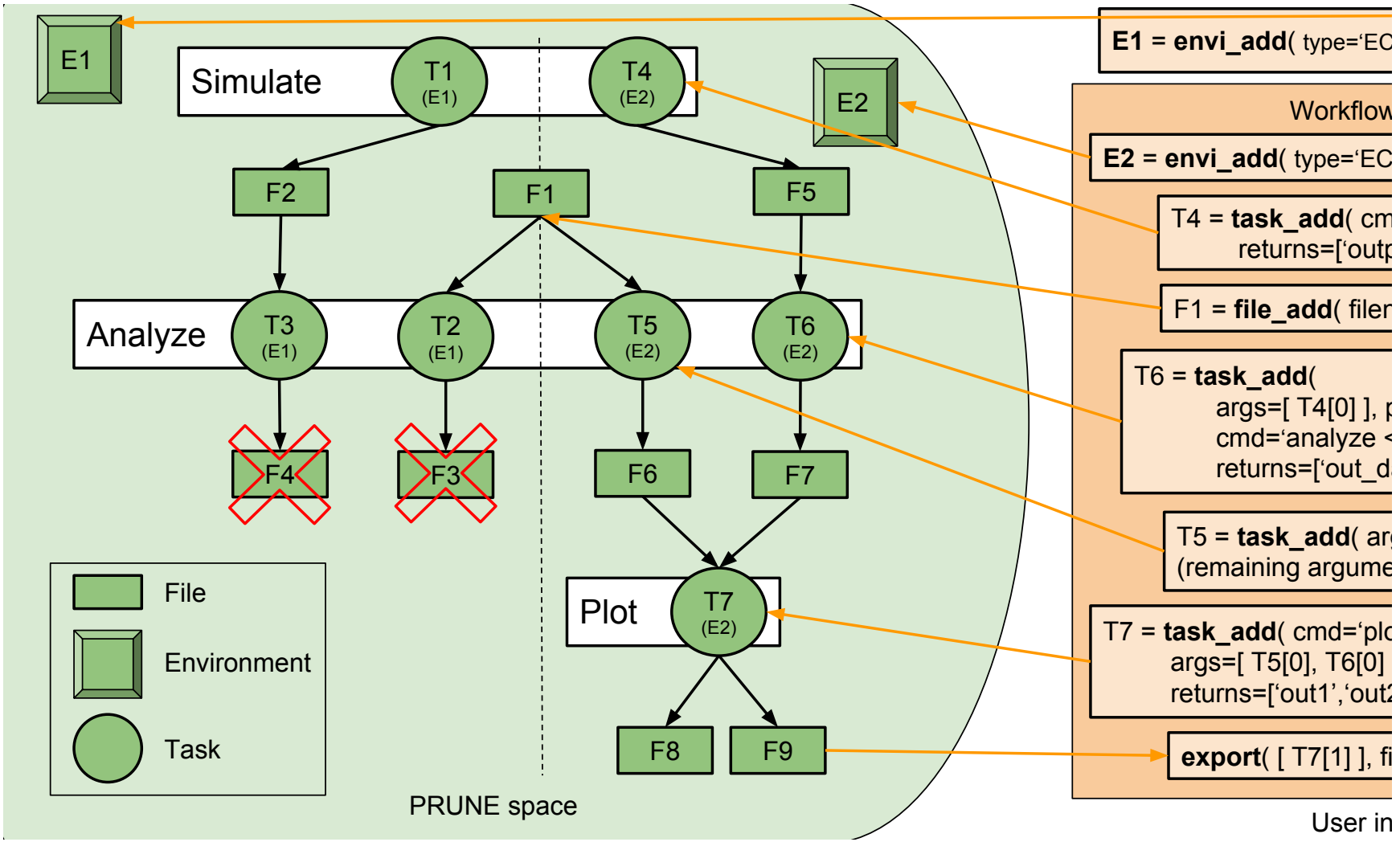
Overview


E1
Simulate
T1
(E1)
T4
(E2)
E2
F2
F1
F5
Analyze
T3
(E1)
T2
(E1)
T5
(E2)
T6
(E2)
F4
F3
F6
F7
Plot
T7
(E2)
F8
F9
File
Environment
Task
PRUNE space
E1 = envi_add(type='EC2', image='hep.beta')
Workflow Version #2
E2 = envi_add(type='EC2', image='hep.stable')
T4 = task_add(cmd='simulate > output',
returns=['output'], environment=E1)
F1 = file_add(filename='./observed.dat')
T6 = task_add(
args=[T4[0]], params=['input_data'],
cmd='analyze < in_data > out_data',
returns=['out_data'], environment=E2)
T5 = task_add(args=[F1], ...)
(remaining arguments the same as above)
T7 = task_add(cmd='plot in1 in2 out1 out2',
args=[T5[0], T6[0]], params=['in1','in2'],
returns=['out1','out2'], environment=E2)
export([T7[1]], filename='./plot.jpg')
User interface
Compute Resources
User space
Events
Data/Sim.
BDT output

User Interface



Overview


E1
Simulate
T1
(E1)
T4
(E2)
E2
F2
F1
F5
Analyze
T3
(E1)
T2
(E1)
T5
(E2)
T6
(E2)
F4
F3
F6
F7
Plot
T7
(E2)
F8
F9
File
Environment
Task
PRUNE space
E1 = envi_add(type='EC
Workflow
E2 = envi_add(type='EC
T4 = task_add(cm
returns=['outp
F1 = file_add(filer
T6 = task_add(
args=[T4[0]], p
cmd='analyze <
returns=['out_d
T5 = task_add(ar
(remaining argume
T7 = task_add(cmd='plo
args=[T5[0], T6[0]
returns=['out1','out2
export([T7[1]], fi
User in

Sample code: Merge sort

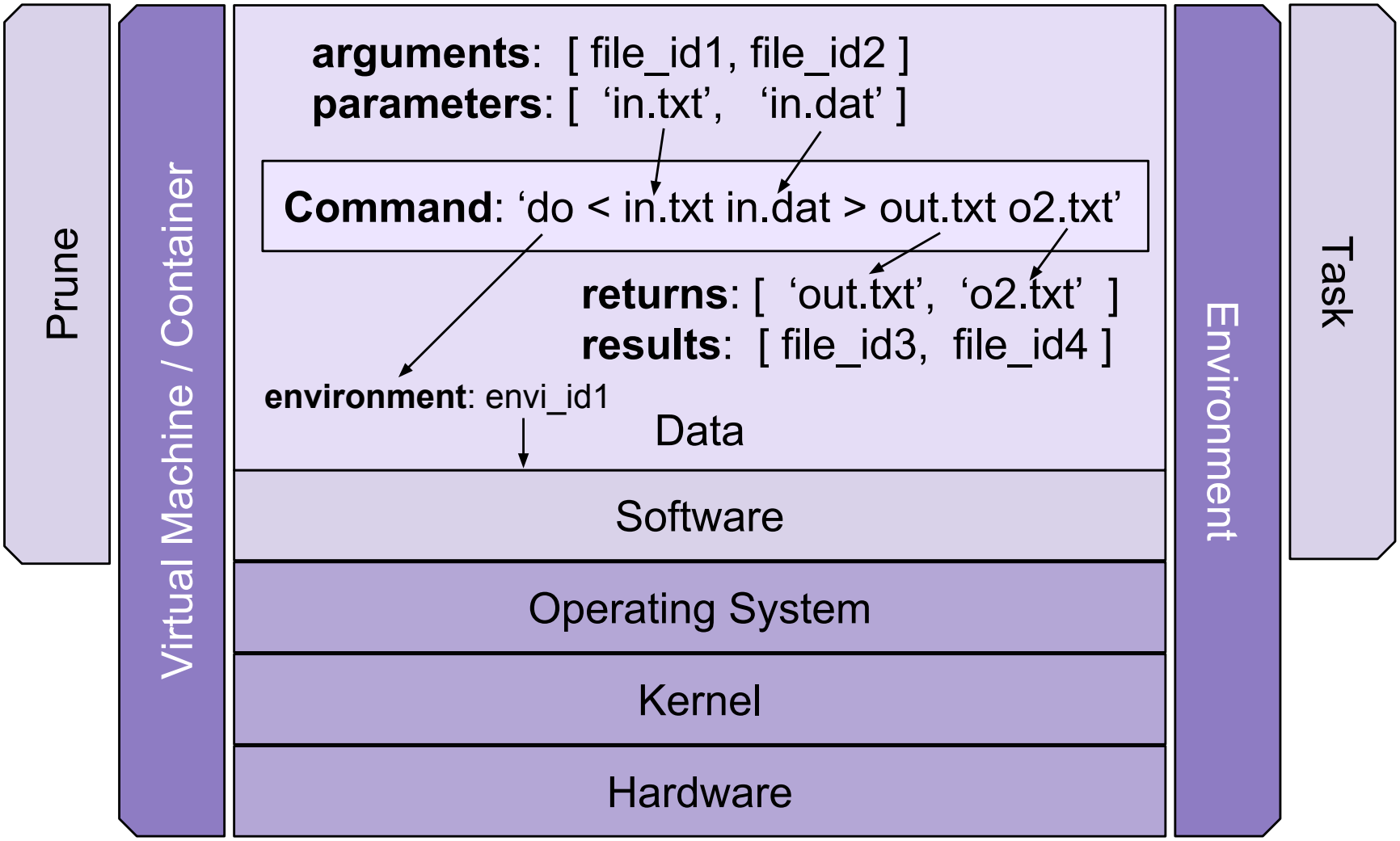
UNIVERSITY OF NOTRE DAME

```
#!/usr/bin/env python
from prune import client
prune = client.Connect() #Use SQLite3

####### Import sources stage #######
E1 = prune.env_add(type=`EC2', image=`ami-b06a98d8')
D1, D2 = prune.file_add( `nouns.txt', `verbs.txt' )
```

Sample code: Merge sort

UNIVERSITY OF NOTRE DAME

```
####### Sort stage #######
D3, = prune.task_add( returns=[`output.txt'],
    env=E1, cmd=`sort input.txt > output.txt',
    args=[D1], params=[`input.txt'] )
D4, = prune.task_add( returns=[`output.txt'],
    env=E1, cmd=`sort input.txt > output.txt',
    args=[D2], params=[`input.txt'] )

####### Merge stage #######
D5, = prune.task_add(
    returns=[`merged_out.txt'], env=E1,
    cmd=`sort -m input*.txt > merged_out.txt',
    args=[D3,D4], params=[`input1.txt',`input2.txt'] )
```


Prune
Virtual Machine / Container
arguments: [file_id1, file_id2]
parameters: ['in.txt', 'in.dat']
Command: 'do < in.txt in.dat > out.txt o2.txt'
returns: ['out.txt', 'o2.txt']
results: [file_id3, file_id4]
environment: envi_id1
Data
Software
Operating System
Kernel
Hardware
Environment
Task

Sample code: Merge sort

UNIVERSITY OF NOTRE DAME

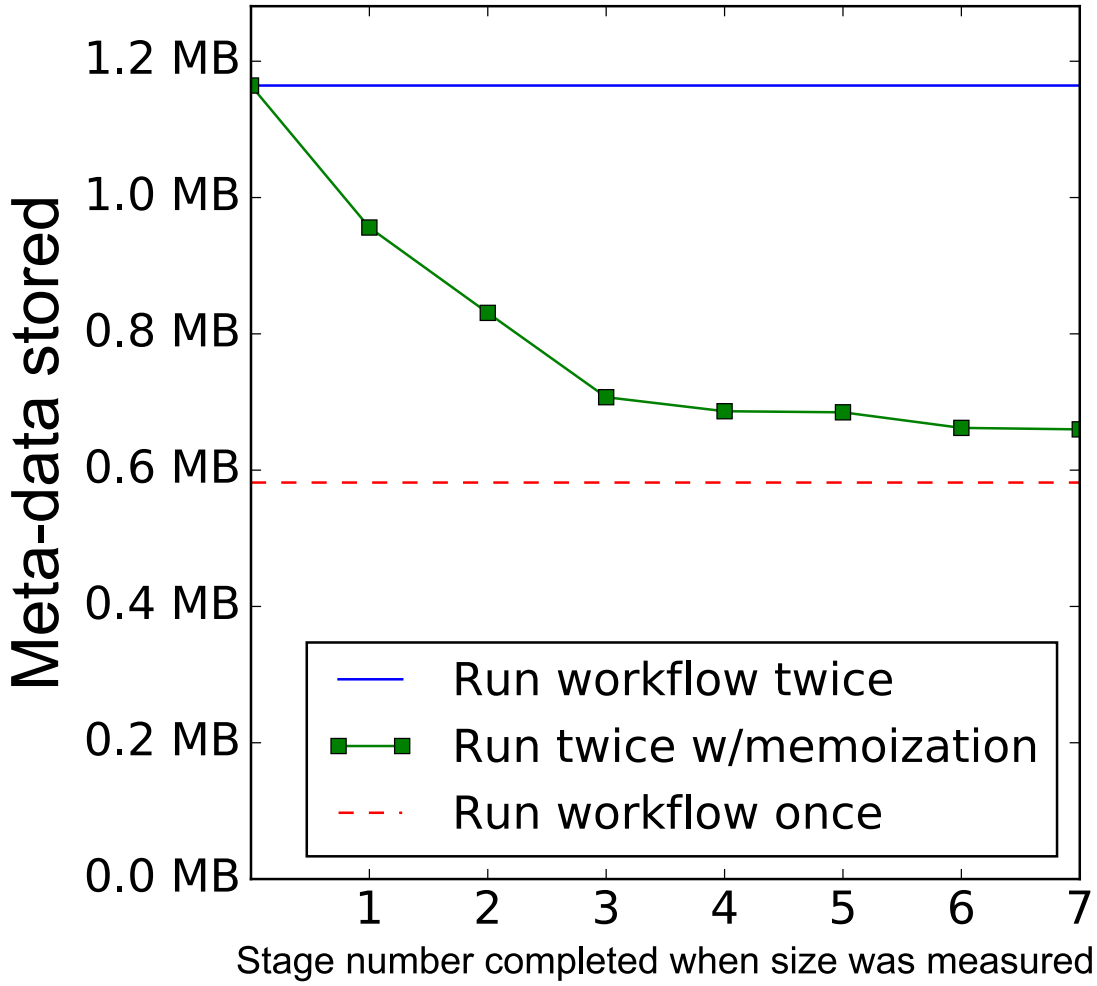
```
####### Sort stage #######
D3, = prune.task_add( returns=[`output.txt'],
    env=E1, cmd=`sort input.txt > output.txt',
    args=[D1], params=[`input.txt'] )
D4, = prune.task_add( returns=[`output.txt'],
    env=E1, cmd=`sort input.txt > output.txt',
    args=[D2], params=[`input.txt'] )

####### Merge stage #######
D5, = prune.task_add(
    returns=[`merged_out.txt'], env=E1,
    cmd=`sort -m input*.txt > merged_out.txt',
    args=[D3,D4], params=[`input1.txt',`input2.txt'] )
```

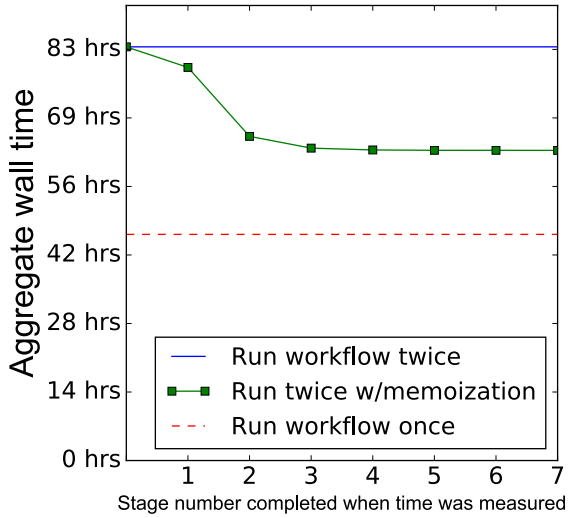
Sample code: Merge sort

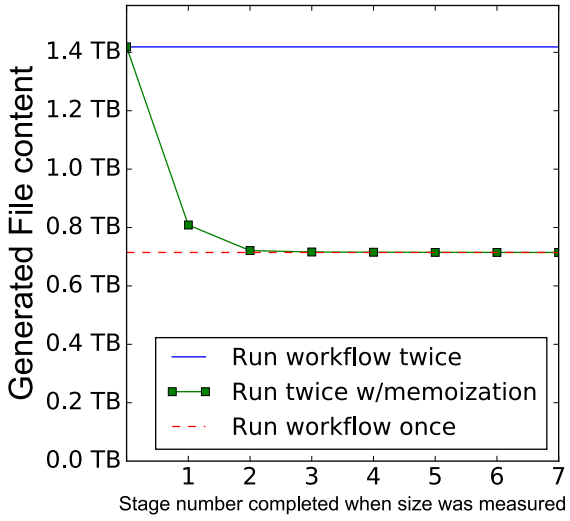
UNIVERSITY OF NOTRE DAME

```
####### Execute the workflow #######
prune.execute( worker_type='local', cores=8 )

####### Export #######
prune.export( D5, `merged.txt' ) # Final data
prune.export( D5, `wf.prune', lineage=2 )
```

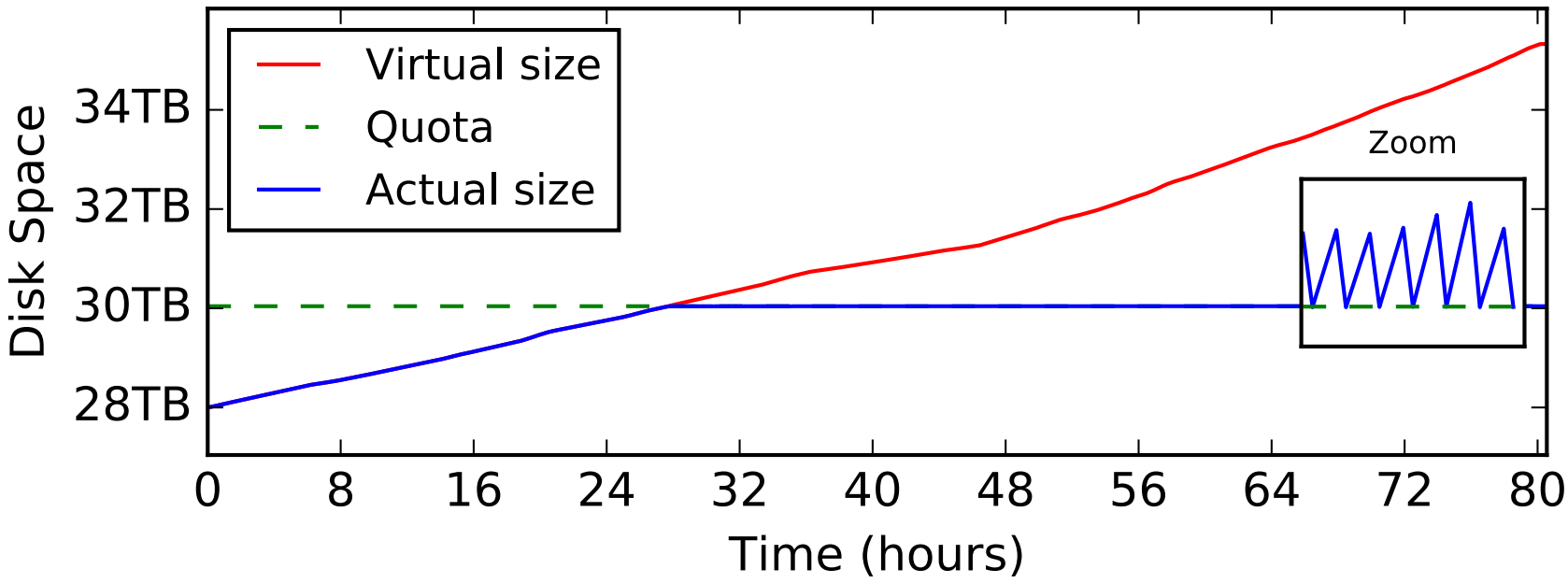
Derivation History = Cachable Results





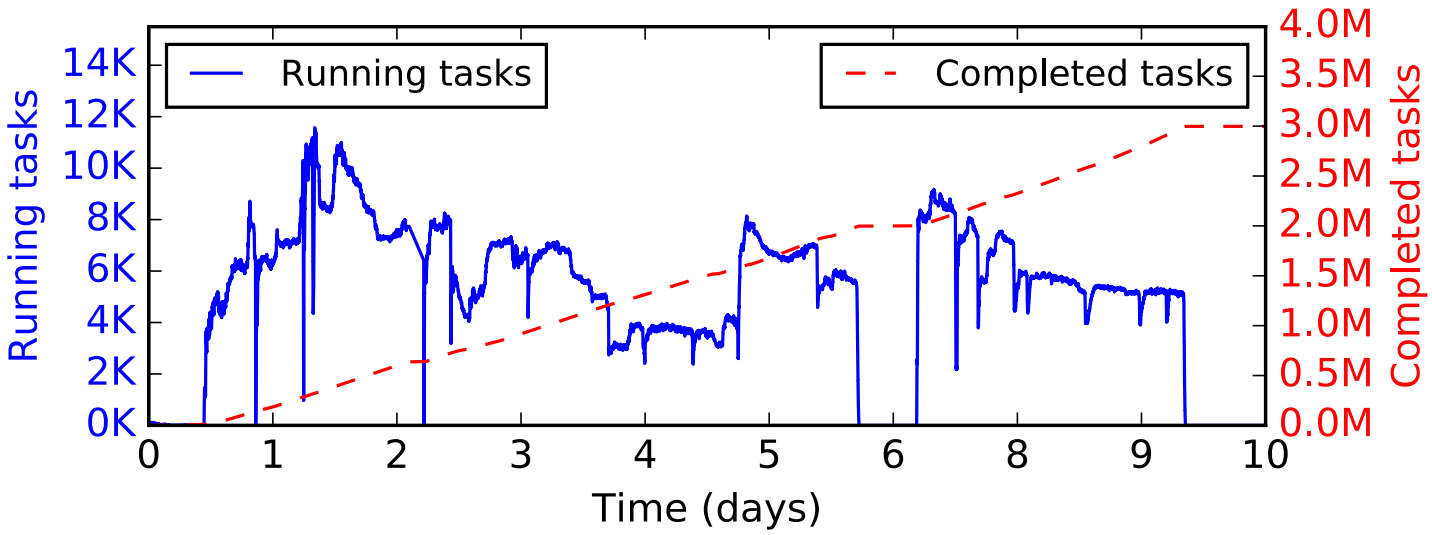


Quotas


Virtual size
Quota
Actual size
Zoom
Disk Space
34TB
32TB
30TB
28TB
0
8
16
24
32
40
48
56
64
72
80
Time (hours)

Scalability

- ~12,000 parallel cores
- ~3 million tasks
- Wall clock overhead
  - ~1% above native



Thank You!

- Sample workflows
- http://ccl.cse.nd.edu/software/prune/prune.html
  - Merge sort
  - Pairwise comparisons (US Censuses)
  - High-energy Physics

CCTools

UNIVERSITY OF NOTRE DAME