# A Case Study in Preserving a High Energy Physics Application

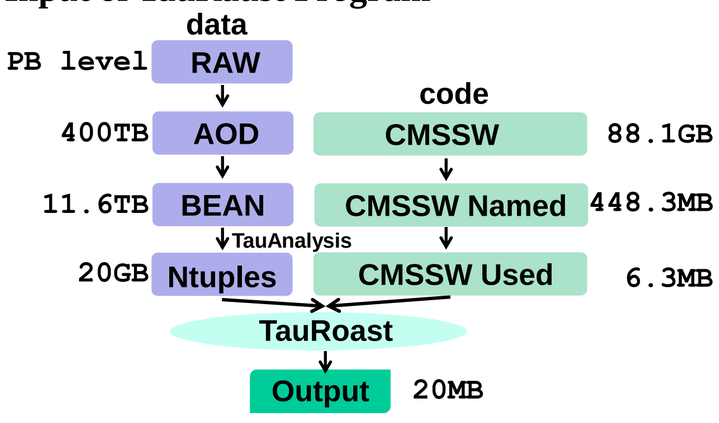
Haiyan Meng, Matthias Wolf, Peter Ivie, Anna Woodard, Michael Hildreth, and Douglas Thain
Department of Computer Science and Engineering / Department of Physics

UNIVERSITY OF NOTRE DAME

## ABSTRACT

**The reproducibility of scientific results increasingly depends upon the preservation of computational artifacts. Although preserving a computation to be used later sounds easy, it is surprisingly difficult due to the complexity of existing software and systems. Implicit dependencies, networked resources, and shifting compatibility all conspire to break applications that appear to work well. To investigate these issues, we present a case study of a complex high energy physics application. We analyze the application and attempt several methods at extracting its dependencies for the purposes of preservation. We propose one fine-grained dependency management toolkit to preserve the application and demonstrate its correctness in two different environments - one virtual machine from the Notre Dame Cloud Platform and one virtual machine from the Amazon EC2 Platform. We report on the completeness, performance, and efficiency of each technique, and offer some guidance for future work in application preservation.**

## Input of TauRaust Program




*high selectivity of data and code*

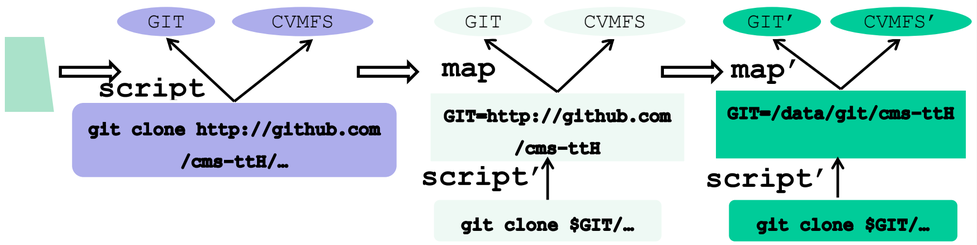
## Observations

**Many Explicit External Dependencies**
**Many Implicit Local Dependencies**
**Configuration Complexity**
**High Selectivity**
**Rapid Changes in Dependencies**

## High Selectivity of Data and Software Dependencies

| Name | Location | Total Size | Named Size | Used Size |
|---|---|---|---|---|
| CMSSW code | CVS | 88.1 GB | 448. 3 MB | 6.3 MB |
| Tau source | Git | 73.7 MB | 73.7 MB | 6.7 MB |
| PyYAML binaries | HTTP | 52 MB | 52 MB | 0 KB |
| .h file | HTTP | 41 KB | 41 KB | 0 KB |
| Ntuples data | HDFS | 11.6 TB | N/A | 20 GB |
| Configuration | CVMFS | 7.4 GB | N/A | 103 MB |
| Linux commands | localFS | 110 GB | N/A | 68.4 MB |
| Home dir | AFS | 12 GB | N/A | 32 MB |
| Misc commands | PanFS | 155 TB | N/A | 1.6 MB |
| Total | | 166.8 TB | N/A | 21 GB |

## Challenge 1: How to redirect data source of each dependency?

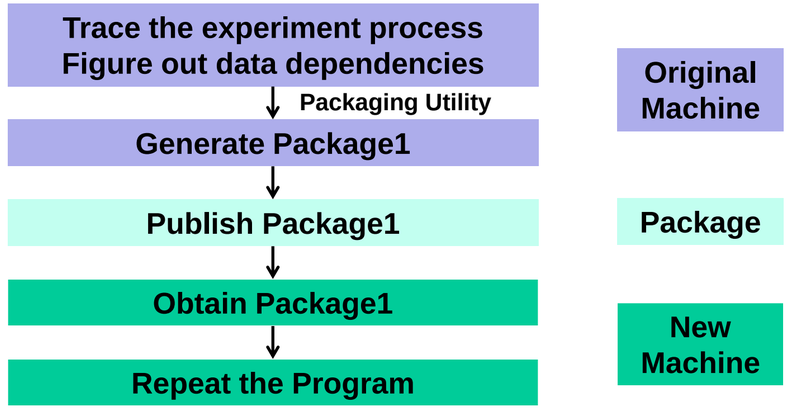



**Email** **Script** **Abstract Script** **Package**

`GIT + CVMFS` : original data `GIT' + CVMFS'`: data copied into the package

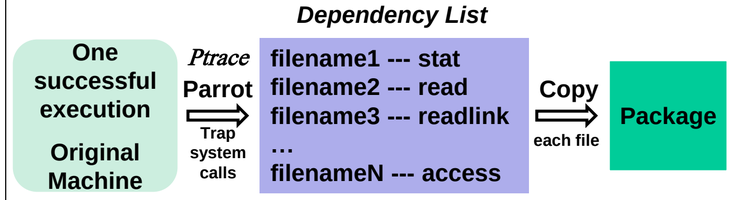
*Evolution of Preservation Methods*

## Challenge 2: How to figure out the really used data?




## One Implementation of Package Method

1. Obtain one successful execution
2. Generate a dependency list
3. Generate a Package containing all the dependencies




## Evaluation

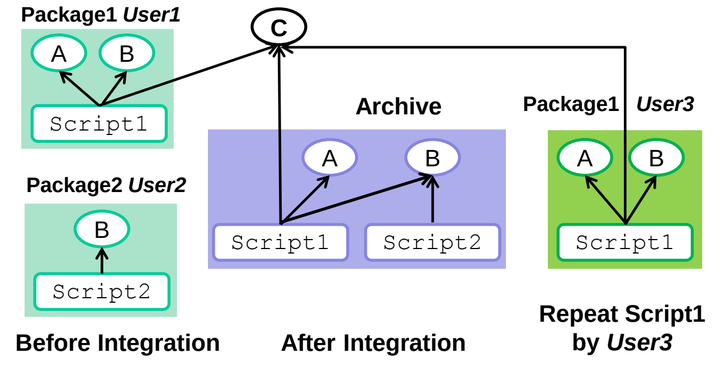
### 1. Execution time

| Task Category | Original Script | Reduced Package |
|---|---|---|
| Obtain Namelist | N/A | 28min 28s |
| Generate Package | N/A | 26min 19s |
| Obtain Software | 8min 11s | N/A |
| Build Environment | 5min 49s | 4s |
| Analyze Code | 20min 31s | 13min 04s |

### 2. Correctness

| Machine Type | Distribution Version | CPU Cores | Memory (GB) | Execution Time |
|---|---|---|---|---|
| Original Machine | Red Hat 5.10 | 64 | 125 | 13min 04s |
| KVM (Notre Dame) | CentOS 5.10 | 4 | 2 | 21min 38s |
| Xen (EC2) | Red Hat 5.9 | 16 | 60.5 | 13min 30s |

## Preserving Multiple Artifacts




## Open Problems

**Measure the Mess or Force Cleanliness?**
**Granularity of Dependencies**
**Scope of Reuse**
**Dependency Detection**

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.