A Case Study in Preserving a High Energy Physics Application

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

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.

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
<table>
<thead>
<tr>
<th>Task Category</th>
<th>Original Script</th>
<th>Reduced Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obtain Namelist</td>
<td>N/A</td>
<td>28min 28s</td>
</tr>
<tr>
<td>Generate Package</td>
<td>N/A</td>
<td>26min 19s</td>
</tr>
<tr>
<td>Obtain Software</td>
<td>6min 11s</td>
<td>N/A</td>
</tr>
<tr>
<td>Build Environment</td>
<td>5min 49s</td>
<td>4s</td>
</tr>
<tr>
<td>Analyze Code</td>
<td>20min 31s</td>
<td>13min 04s</td>
</tr>
</tbody>
</table>

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

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

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.