Appendix: Artifact Description/Artifact Evaluation

I. Overview of Contributions and Artifacts

A. Paper's Main Contributions

C_1 Methods for reshaping high energy physics applications from long-running to near-interactive using TaskVine.

C_2 Evaluation of serverless task execution for physics application reshaping.

C_3 Evaluation of utilizing peer transfers to for physics application reshaping.

B. Computational Artifacts

A_1 github.com/BarrySlyDelgado/SC_TaskVineHEP_AD

Artifact ID	Contributions Supported	Related Paper Elements
A_1	C_1, C_2, C_3,	Figures 7-8, 11-15 Table 1

II. Artifact Identification

A. Relation To Contributions

The artifact provided (A_1) includes plotting tools used to generate the original figures in the paper along with some logs. Additionally, the artifact includes scaled down experiments of used to ensure and verify the reproducibility of the artifact. The provided experiments use a reduced data set to scale down execution time and needed resources. A set of experiments can executed either locally or on distributed resources. The resources provided within the artifact substantiate our paper contributions by providing the original logs of the experiments used in the paper and scaled down experiments for replication.

B. Expected Results

The scaled down experiments within the artifact should replicate the significant contributions shown in the paper. Specifically, for figures 7-8 and Figures 11 - 15. As the artifact produced replicated figures for the ones shown in the paper, the expected results are as follows: the replication of figure 7 should depict two runs of the application DV3 the plot on the left depicting no data transfers between peer workers while the plot on the right will depict more data transfer among peer workers, where the max transfers between any two peers is smaller when peer transfers are not enables. The replication of figure 8 produces a plot depicting the runtime of tasks during an execution of DV3 in two modes, Tasks and Function Calls. Here, the tasks executed in Function Calls mode should be faster. The replication of figure 11 should depict a scaled down version of the application RS-Triphoton, depicting worker cache usage over the applications runtime. Specifically, this is executed in two modes, single and binary graph reeducation. Binary graph reduction should use less distributed disk space and use execute faster. The replication of Figure 12 depicts executions of DV3 on various stack configurations as mentioned in the paper. The replicated figure depicts 2 graphs, consisting of pertinent information during runtime. Specifically, concurrent task and waiting tasks. This replication contains 3 lines instead of the 4 shown in the paper. The reason being that stack 1 and 2 use different hardware that may not be available for individuals execution artifact experiments. Instead. stack 1 and 2 are depicted as a single entity. The expected results for this replicate figure should show stack 4 having high concurrent tasks while having low waiting tasks. Stack 3 should have less concurrency resulting in more waiting tasks . The combination of stack 1 and 2 (noted as stack 2) should have even less concurrency and even more waiting tasks. The replication of figure 13 should depict the task stream of executions of DV3 on a set of workers for stacks 3 and 4. Here, the task stream for stack 3 should be more segmented than the stream of stack 4. The replication of figure 14 should depict the executions of DV3 and RS-Triphoton at different scales (cores) with workflow executors TaskVine and dask.distributed. Here TaskVine should perform better than dask.distributed at each scale Additionally, scaling is shown for DV3 and RS-TriPhoton.The replication for figure 15 should depict the amount of concurrent tasks during an execution of DV3. Here, Concurrency should be maintained high for a majority of the duration of the execution.

C. Estimated Runtime

There are two main experiments presented in the artifact, DV3 and RSTriPhoton. The experiment for DV3 runs in 4 configurations where each configuration is run 3 times execution for each configurations at 2 scales. RS-TriPhoton is run in 2 configurations, each 3 times at 3 scales.

1) Artifact Setup: Once downloaded, there is little setup for execution (less than 10 minutes).

2) Artifact Execution: Each execution of DV3 can take from 10 - 15 minutes. For all runs at all scales and configurations this can take from 4 - 6 hours. Each execution of RS-TriPhoton completes in roughly a minute. This experiment takes roughly 20 minutes.

3) Artifact Analysis: Once execution is complete, the analysis of the reproduced experiments should generate graphs. Graph generation should take less than 10 minutes.

D. Artifact Setup

1) Hardware: If executed on a compute cluster, compute nodes should be equipped with a local disk and have access to a shred filesystem.

2) Software: The artifact should be executed on a LINUX based machine. For graph generation ,latest versions of Python and the library matplotlib are required. Most importantly the latest installation of ndcctools is required, which can be installed via conda. Additional can be found on the cctools GitHub.

3) Datasets/Inputs: Datasets are to be downloaded and included in the respective experiment's directory under the directory name "data". Reduced Datasets are to be made available here: github.com/BarrySlyDelgado/SC_TaskVineHepData

4) Installation and deployment: Once downloaded, there are two directories, "paper_figures" and "experiments" within the experiments directory, traversing to either directory "DV3" or "RS-TriPhoton", individuals can execute an experiment via the command "./poncho_package_run -e ENV_NAME.tar.gz ./RUN_SCRIPT". To generate the referenced tarball, one should first install ndcctools as mentioned before. Then, one can create said tar ball via command "poncho_package_create env.yml ENV_NAME.tar.gz" Once each experiment is complete, a user can run gen_plots.sh located in the experiments directory. This will produce the replicated figures mentioned above. Notably, Values are to be entered manually for replications of figure 14. these values can be retrieved from the logs for dask.distributed executions and generated using command "vine_plot.py LOG_NAME –compute-time".

E. Artifact Evaluation

The steps to evaluate the artifact is as follows: once data has been downloaded, execute the experiments located in the directories experiments/DV3 and experiments/RS-TriPhoton as shown in the Artifact Setup (installation and deployment) section above. This produces the logs used for artifact replication. Then generate the replicated figure by executing "./gen_plots.sh". For each experiment 4 core workers are used to reduce the scale. Additionally, 64GB of memory and 100GB of disk space is allocated for execution. Each configuration, for each scale during an experiment is repeated 3 times. Most of these parameters can be modified within the run script. Notably the size of the datasets for DV3 and RS-TriPhoton are 5 and 8 gigabytes respectively. Additionally,

F. Artifact Analysis

For the figures, information is taken from logs generated on each execution. penitent information is computed from these logs to generate for displays. For each execution, logs will be outputted into the experiments/logs directory. To generate the displays the logs are parsed and relevant information is computed. information such as task placement, data transfer events and task run times are computed from these logs. There are various events that are accumulated to produce a figure. For example, figure is 12 is generated via computing task placement and execution duration from the logs.