Pynamic: The Python Dynamic Benchmark

Summary Version

1.0

Purpose of Benchmark

Pynamic is designed to test a system's ability to handle the heavy use of dynamically linked libraries exhibited by large Python-based applications.

Characteristics of Benchmark

Pynamic is based on pyMPI, an MPI extension to the Python programming language. Pynamic adds a code generator that creates a user-specified number of Python modules and utility libraries to be linked into pyMPI. With the appropriate parameters, Pynamic can build a dummy application that closely models the footprint of an important Python-based multiphysics code at LLNL. This multiphysics code uses about five hundred dynamically linked libraries (DLLs) and stresses a system's dynamic loading ability.

Mechanics of Building Benchmark

Pynamic includes the source for pyMPI, which requires a Python installation. In addition, two of the key Pynamic files are themselves Python scripts. The preferred configuration parameters are as follows:

```
./config_pynamic.py 496 1850 –e –u 215 1850 –n 100 –t
```

This will create a standalone pyMPI executable, as well as a pynamic-pyMPI executable with all of the DLLs linked in.

Mechanics of Running Benchmark

```
srun ./pynamic-pyMPI pynamic_driver.py
```

```
srun ./pyMPI pynamic_driver.py
```

Verification of Results

A successful run of Pynamic (i.e., no errors) is sufficient verification of functionality. A time comparison between pynamic-pyMPI and pyMPI provides insight into the benefits and penalties of linking against the generated shared libraries.