TACC

TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU

TEXAS

The University of Texas at Austin

Tracking imports for Python, R

Robert McLay

Feb. 17, 2022

XALT: Outline

- XALT can track executable that are run
- Also the shared libraries
- Can we track imports for Python and R?
- How could we do it?

How can we track imported packages?

- It will require special code unique for each tool
- Somehow we have to “insert” code into the import process
- This is typically accomplished by using some Hook provide by the tools’ developers

Tracking imports of packages for R

- It started with R.
- James McComb & Michael Scott from IU developed the R part
- They wrote code that intercepts the import action.
- XALT provides a program to take that data: xalt_record_pkg
- All packages tracker use this program to collect the data.

XALT Prerequisites

- A `path_pattern` in a sites' Config.py.
- `['PKGS', r'.*R'],`
- `['PKGS', r'.*python[0-9.]*'],`

XALT Env. Vars for PKGS

- Since the execution is happenning during XALT Tracking
- The following environment variables are defined
  - XALT_DIR: The root directory of where xalt_record_pkg
  - XALT_RUN_UUID: The run uuid for the current R or Python program running
- The R and Python Import hooks only collect data if XALT_RUN_UUID is defined

xalt_record_pkg usage

- The hook routine does the following
- Gets XALT_DIR and XALT_RUN_UUID from env.
- Builds command path to xalt_record_pkg using XALT_DIR
- The rest of the command line is:
  - -u <run_uuid>
  - program <name>
  - xalt_run_uuid <run_uuid>
  - package_name <pkg_name>
  - package_version <pkg_version>
  - package_path <pkg_path>

xalt_record_pkg execution

- xalt_record_pkg builds a json string w/ data
- Every import will generate a record
- Why?

`xalt_record_pkg` execution (II)

- The hook code in R, Python is called dynamically
- There is nothing recording that can be called at the end.
- Originally XALT was going to make PKGS not be sampled,
- Also generate a start record
- This way import records would have something to connnect with.
- However this is a bad idea!

`xalt_record_pkg` execution (III)

- XALT needs an execution record stored to save import data
- But there are too much Python runs to store every one
- Instead XALT uses /dev/shm in a unique directory (UUID)
- This avoids overlap with other executions
- But why?

Why write package import data to /dev/shm?

- Speed
- Python and R can be sampled
- Data is only sent on the “wire” at end of program if sampled
- Delete data otherwise.

Sent on the “wire”?

- Import records are saved on /dev/shm
- This is independent of $XALT_TRANSMISSION_STYLE
- At the end of the execution the end record and import records are sent via $XALT_TRANSMISSION_STYLE
- This is only if sampled.

What to do with this data?

- Find the list of heavily imported packages
- Find who is using conda python
- XALT won't know if something is imported but not used
- Track down heavily used packages and try to speed them up.

Python hook: py_src/xalt_sitecustomize.py ⇒ sitecustomize.py

- Python 2 and 3 both look for sitecustomize.py when starting
- Help from Riccardo Murri
- All Pythons uses sys.meta_path to locate files to import
- Can register object to capture imports.
- Just add XALT's location of sitecustomize.py to PYTHONPATH

Filtering python packages via site's Config.py

```
python_pkg_patterns = [
{ 'k_s':'SKIP','kind':'path','patt': r"^[^/]" },
{ 'k_s':'SKIP','kind':'name','patt': r"^_"    },
{ 'k_s':'SKIP','kind':'name','patt': r".*\." },
{ 'k_s':'KEEP','kind':'path','patt': r".*/.local/" },
]
```

Conclusions

- We have a way to track imports from R and Python
- It works well but there are a few conflicts with sitecustomize.py
- We have yet more data to try to figure out what to do with.

Future Topics?

- Others?