How to Pickle
Serialisation in Python

Martin Kerscher

Fakultät für Physik
Ludwig–Maximilians–Universität München

code coffee - May 16, 2023

Serialisation

Translate a data structure or an object into a format that can be transmitted or stored.

Transfer or store data, e.g. for (remote) procedure/method calls

- Property Lists, COM, XML, YAML, JSON, ...

Scientific data formats:

- flat: csv, txt,
- hierarchical: NetCDF, HDF5, FITS, feather, ...

Python:

- modules `marshall`, `h5py`, `astropy`, `json`, `yaml`, `...`
- numpy, pandas offer some routines

and there is **pickle** (with `dill` on top).

A simple example: `prog1.py, prog2.py`

Write:

```
import pickle
fname = "thestate.pickle"

x = 11

with open(fname,'wb') as f:
    pickle.dump(x, f)

print(x)
```

- Dump "object" x to a file
- Inspect on the commandline with
  `python -m pickle thestate.pickle`
- Load object from file
  (you know what has been saved).
- Also with byte object:
  .dumps, .loads

Read:

```
import pickle
fname = "thestate.pickle"

with open(fname,'rb') as f:
    x = pickle.load(f)

print(x)
```

(Almost) everything can be pickled,
see `more.py, more2.py`.

Why does this work?

```
...
print(type(tree))

import inspect
l = inspect.getmembers(tree)
print(l)
```

```
class bare:
    pass

o = bare()
o.x = 1
print(o.x)
```

- Inspect an object at runtime: type, all members, ...
- Build an object at runtime, e.g. add a new member.

Reusable calculations

```
def save_state(fname, *args):
    with open(fname,'wb') as f:
        pickle.dump(args, f)


def load_state(fname, *args):
    try:
        f = open(fname,'rb')
    except IOError:
        # if no state was saved yet
        S = args
    else:
        # load state
        S = pickle.load(f)
        # simple versioning:
        # do not use if args[0]>=S[0]
        if args[0] >= S[0]:
            S = args
        f.close()
    return S
```

- `*args` – variable number of arguments returned as tuple
- `save_state`
  a simple wrapper
- `load_state`
  only if file is there
  only if file has newer version
- version saved as `int` as the first component.

There is more to serialisation

Pickle is not ment for

- long-term storage
- interoperability (platforms, versions)

For complex objects consider `dill`.

Support in other languages?

- C++: supported with boost, but not automated
- Fortran: rudimentary support, Transfer perhaps TOML
- Julia: Serialization.serialize()
- R: serialize()
- Java: certainly, but I dont know

Green calculations

- efficient cooling and reuse of waste heat
- modern hardware and energy management
- efficient algorithms - this is on you

Use computing resources at the LRZ[1]

There is a price – `slurm`:

```
Slurm Job_id=453774 Name=a12 Ended, Run time 3-00:00:01, TIMEOUT, ExitCode 0
```

Workaround astonishingly simple with pickle.

[1] `https://www.lrz.de/wir/green-it_en/`