Uni.lu HPC School 2021

PS08: Advanced Distributed Computing with Python

Uni.lu High Performance Computing (HPC) Team

Dr. E. Kieffer

University of Luxembourg (UL), Luxembourg

http://hpc.uni.lu

Latest versions available on Github:

UL HPC tutorials: https://github.com/ULHPC/tutorials

UL HPC School: hpc.uni.lu/education/hpcschool

PS08 tutorial sources: ulhpc-tutorials.rtfd.io/en/latest/python/advanced/

Summary

Main Objectives

1. How to parallelise your python code ?
2. Hereafter, we are going to see two alternatives :
   - ↪ High-level approach with ipyparallel for scikit-learn
   - ↪ Low-level approach with scoop
   - ↪ Flexible approach with Dask

Summary

Scikit-learn + ipyparallel

- **Scikit-learn** is a well-known python scientific package:
  - ↪ Machine learning algorithms (e.g. SVM)
  - ↪ Data analysis approaches (e.g. PCA)
  - ↪ Data mining techniques (e.g. Clustering)
- Scikit-learn algorithms can be parallelised
- Especially useful for Hyper-parameters search
- Scikit-learn relies on **ipyparallel** and **joblib** libraries to parallelism algortihms

Ipyparallel

- Originally designed under **Ipython**
- IPython's architecture for parallel and distributed computing
- Support many different styles of parallelism:
  - ↪ Single program, multiple data (SPMD) parallelism
  - ↪ Multiple program, multiple data (MPMD) parallelism
  - ↪ Message passing using MPI
  - ↪ Task farming
  - ↪ Hybrid approaches combined the above ones
- Ipyparallel can detect a job scheduler (e.g. Slurm) when started on a HPC platform

Practical session

Please go to https://ulhpc-tutorials.readthedocs.io/en/latest/python/advanced/scikit-learn/

Summary

Scoop + deap

- **Deap**
- Python evolutionary computing library:
  - ↪ Genetic algorithms
  - ↪ Particle swarm algorithms
  - ↪ Evolutionary strategies
  - ↪ Estimation of Distribution algorithms
- Deap relies on **scoop**

Scoop

- SCOOP => Scalable COncurrent Operations in Python
- Applications of SCOOP:
  - ↪ Evolutionary algorithms
  - ↪ Monte Carlo simulations
  - ↪ Data mining
  - ↪ Data processing
  - ↪ Graph traversam
- Very simple to use
- Override default map (reduce) function

Practical session

Please go to https://ulhpc-tutorials.readthedocs.io/en/latest/python/advanced/scoop-deap/

Summary

Dask

Dask is a flexible library to perform parallel computing Data Science tasks in Python. Although multiple parallel and distributed computing libraries already exist in Python, Dask remains **Pythonic** while being very efficient (see Diagnosing Performance). Dask is composed of two parts:

- **Dynamic task scheduling**: Optimized computational workloads (see distributed dask)
- **Big Data collections**: Parallel and distributed equivalent data collecting extending Numpy array, Pandas dataframes

An interesting feature of Dask is Python iterators for large-than-memory or distributed environments. Dask tries to provide different qualities:

- **Familiar**: Provides parallelized NumPy array and Pandas DataFrame objects
- **Flexible**: Provides a task scheduling interface for more custom workloads and integration with other projects.
- **Native**: Enables distributed computing in pure Python with access to the PyData stack.

Practical session

Please go to
https://ulhpc-tutorials.readthedocs.io/en/latest/python/advanced/dask-ml/

Thank you for your attention...

Questions?

ulhpc-tutorials.rtfd.io/en/latest/python/advanced/

High Performance Computing @ Uni.lu

University of Luxembourg, Belval Campus
Maison du Nombre, 4th floor
2, avenue de l'Université
L-4365 Esch-sur-Alzette
mail: hpc@uni.lu

1. Introduction
2. Parallel machine learning with ipyparallel
3. Parallel evolutionary computing with scoop
4. Dask: Scalable analytics in Python

Uni.lu HPC School 2021 Contributors

hpc.uni.lu