Improve Data Retriever efficiency for out-of-memory scale datasets

Basic Information

- ### Contact Information

  - Name : Xian Wang
  - Personal Email : wang000@mail.ustc.edu.cn
  - github : https://github.com/CoronaeW (https://github.com/CoronaeW)
  - Cell Phone : +86 13155655033

- ### Brief Introduction

  I am a senior of University of Science and Technology of China, major in Physics, also have a dual major in computer science. I have practiced Python programming in my daily study and researches of lab for almost 3 years. I have the knowledge and experience of making the code more efficient.

Abstract

- Organizing and managing data sets is a time-consuming task in many programming processes. We often spend lot of time in finding and sorting out data sets which we needs in programming. The Data-Retriever is a automated tool for developer to standardize data sets they want to use in programming.
- The goal of the project is to make Data-Retriever more efficient when dealing with large data sets by using Python. It could have faster speed and use less memory. Perhaps some parallel algorithm or using of image processing unit like CUDA coding of Python(like PyCUDA, based on C/C++) can accelerate the speed of codes. After that, an efficient databases should have strong-directional indexes to make finding and using data more easily.

Technical Details

- We can use these ideas to make Python code more efficient：
  - Making the code run faster and with minimal intrusion is to use a real-time compiler (JIT). In the past, we could import psyco after installation and call psyco.full(). The code speed can be significantly improved. JIT can monitor the program in real time and compile some code into machine code.
  - We can use image processing unit to achieve code acceleration, like PyCUDA,PyOpenGL. And these are accelerated from the hardware level. If there is a powerful GPU, we can use it to calculate, thereby reducing the valuable resources of the CPU and memory.
  - We can use Cython, Numba. These projects are dedicated to translating Python code into C, C++, and some other code which run faster.
  - Ctypes and can help us to achieve the operation of the Python underlying object. It can be used to build compiled C objects in memory. And call the function of C in the shared library. However, ctypes are already included in Python's standard library.
- Speed and memory footprint are often not optimized simultaneously, and we always look for their balance in program development. Through these ideas above we can make code more efficient both in speed and in weight.

Schedule

- ### March 28 - May 14 Community Bonding

- Learn the basic knowledge of Data Retriever by the docset to have a clear idea with how to manage data sets.
- Before starting with the project, I would like to participate in issues to find out what method is feasible.
- Discuss the pros and cons of the solution proposed.

- **May 2 - June 11 Phase 1**

  - Learn High-performance computing method and implement them in Python.
  - Start work on learning the structure of funcations in Data Retriever.
  - Put CUDA or other parallel algorithm into use to accelerate the speed of sorting or querying.

- **June 15 - July 8 Phase 2**

  - Start work on modifying the funcations in packages and modules.
  - Optimization the speed and memory footprint both on single processor and multiprocessor.
  - Start work on determining how fast and memory intensive areas of the codebase.

- **July 9- August 6 Phase 3**

  - Start work on adding indexes to the databases for efficient querying.
  - Optimization the speed of searching and querying the indexes.

- **August 7 - 14 Final Work**

  - Adding docstrings, fixing bugs， cleaning code.
  - Passing PR reviews.
  - Submit the code before the end of GSoC.
  - Merge the PRs.

Future works

- My contributions' link is https://github.com/weecology/retriever/commits?author=CoronaeW (https://github.com/weecology/retriever/commits?author=CoronaeW)
- After the GSoC I could provides correspondingly processed data sets with a high degree of precision according to my profession which can enrich the database of Data Retriever.

Why me

The GSoC provides me with the opportunity to contact with open-source python project which I am interested in. I have relevant programming skills and experience, so it is possible for me to succeed in this project. In the past I have been involved in many projects which need lots of data sets.The Retrieval of Land Surface Characteristic Parameters Based on Landsat-7 ETM+ in the Pan Third Pole Region Which is a project I have done. It used lots of satellite datasets when programming in Python so I had to consider to increase the speed of code. It may provide ideas to how to improve Data Retriever efficiency for large scale datasets. After that I am going to continue my study in my school as a postgraduate, and I should have enough time to finish this work.