Ganga GSoC exercise 2019

This document sets out the exercise that should be completed for students interested in applying to work on one of the two Ganga projects offered up as part of GSoC 2019. You can find the description of the projects at
https://hepsoftwarefoundation.org/gsoc/projects/2019/project_Ganga.html

Code for all work done should be placed in a GitHub repository. Also place in the repository the tests that demonstrate that your code is working correctly and a README file that explains everything in the repository.

Please complete the exercise at your own pace. When finished, please send an email to the three mentors as listed on the project page and included in the email announcing this exercise. In the email point to the Github repository that you have created with the documentation of the exercise.

You are also welcome to ask questions from the mentors at any time. You are welcome to complete the exercise for both of the projects, but if you are only interested in one of them, it is fine to simply ignore the exercise related to the other one.

Regards,
Ulrik Egede, Alex Richards and Mark Smith.

Task to be completed for both projects

- Install Ganga based on online information and demonstrate that you can run a “Hello World” job that executes on a “Local” backend.
- Create a job in Ganga that demonstrates splitting a job into multiple pieces and then collates the results at the end.
  - Use the included file CERN.*pdf*.
  - In python, or through using system calls, split the pdf file into individual pages.
  - Create a job in Ganga that will count the number of occurences of the word “the” in the text of the PDF file.
  - Using the ArgSplitter create subjobs that each will count the occurences for a single page.
  - Create a merger that adds up the number extracted from each page and places the total number into a file.

Container project

- Demonstrate that you can accomplish the task above by creating a docker container that when stated will execute all of the above and then exit. The docker container should be uploaded to dockerhub and the corresponding DockerFile placed in Github.
- Create a Ganga job that executes a docker container of choice as its job.

Memory management

In this part there is no need to use the Ganga framework at all. It should be completed in Python 2.7.

- Write a simple python programme that creates a configurable number of simple objects where each is a deep copy of the previous one. Create them inside a delay loop.
- Demonstrate that you can monitor the memory usage of the programme and that you understand the amount of memory used at a given time.
- Modify programme to release any references to the objects one-by-one and demonstrate that you observe the reduction in allocated memory.
- Now implement the same with shallow copies and demonstrate that you understand the reduced memory usage.
- Implement a simple copy-on-modify alogorithm that cobines the memory efficient use of the shallow copy with the ability for some of the objects to be modified.