## Algo\_launcher Architecture

### 2.1 Algo\_launcher framework

### 

The overall Algo\_launcher framework contains three main modules, such as Adaptor, launcher, logs, and Algo\_Bucket, and project configuration files such as rabbitmq\_wrapper.json and launch\_confXXX.json. The function of each module and the configuration file are as follows:：

**Algo\_Bucket：**Adaptation module, which abstracts the algorithm module. The purpose is to not rely on the algorithm code in the Algo\_Bucket file when the code is executed. Only the algorithm integration part of the Algo\_Bucket module is encapsulated into the module to achieve decoupling. purpose。

**Adaptor：**Adaptation module, which abstracts the algorithm module. The purpose is to not rely on the algorithm code in the Algo\_Bucket file when the code is executed. Only the algorithm integration part of the Algo\_Bucket module is encapsulated into the module to achieve decoupling. purpose。

**rabbitmq\_wrapper.json：**The rabbitmq server configuration file has the following configuration file structure:：

{

"mq\_host":"localhost", #rabbitmq server ip adress

"mq\_port":5672, #rabbitmq server port

"mq\_vhost":"/", #default config

"mq\_connect\_retry\_count":10 #rabbitmq server Maximum allowed connections 10

}

Each time you re-deploy a new project, you only need the corresponding new configuration parameters.

**launch\_confXXX.json:**Project algorithm requirements and operational profiles. Its file structure is as follows：

{

"Algo\_List": [

{

"algo\_path":"~/NFS", #Algo\_BucketAlgorithm module path

"adaptor": "ad\_pyevent", #Corresponding algorithm integration code file name

"Q\_name": "people", #The name of the algorithm corresponding to the message queue

"timeout": 600, # Algorithm timeout

"gpu": [

3,3,3 ] #The GPU allocation of the algorithm selects the corresponding parameters according to the hardware conditions.For example, the current configuration means three processes on the first, second, and third cards.

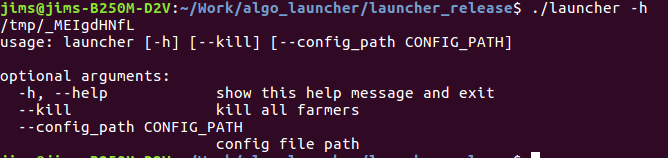
}

]

}

**Launcher:** The startup entry of the entire project, the following modes of operation：

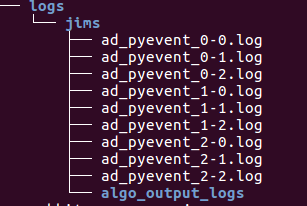
./luncher -h View related commands

./luncher --kill The algorithmic process that ends the project run，You must run this command before each restart of the algorithm

./launcher --config\_path ./launch\_confXXXX.json run The strategy of the algorithm requirement configuration file launch\_confXXXX.json in the current path。

**logs：**When the project starts, it will generate a logs folder under the project file, which is the log module of the entire project framework..

logs Directory Structure：



logs :Folder name for the log directory

jims:The name of the computer to run the launcher

ad\_pyevent\_0-0.log :Is the first process log of the da\_pyevent algorithm on the first card

algo\_output\_logs Is to call the log record folder output by the algorithm module

ad\_pyevent\_out\_0-0.log :Is the algorithm output log called by the first process on the first card by the da\_pyevent algorithm.