GPUPWA
Getting started

Zhang Jingqing

2016.03.24

Outline

1. Get the GPUPWA framework

2. Set GPUPWA environment and run a gpu program

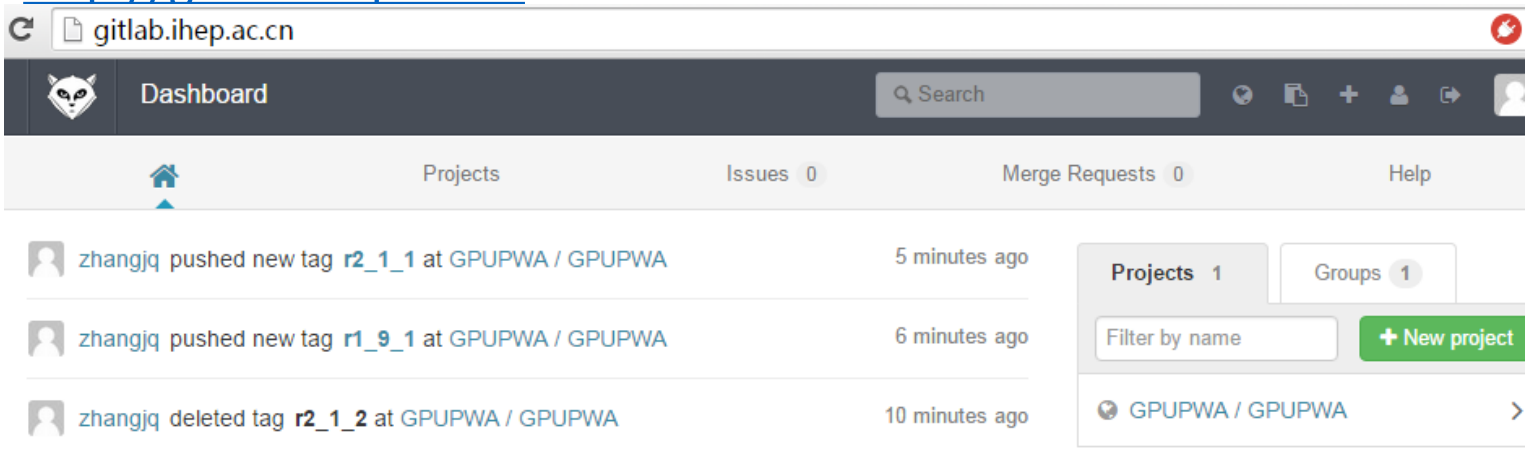
3. Submit a job to gpu queue

Part 1: get the GPUPWA framework from gitlab.ihep.ac.cn

GPUPWA project @gitlab.ihep.ac.cn

ihep gitlab：

http://gitlab.ihep.ac.cn

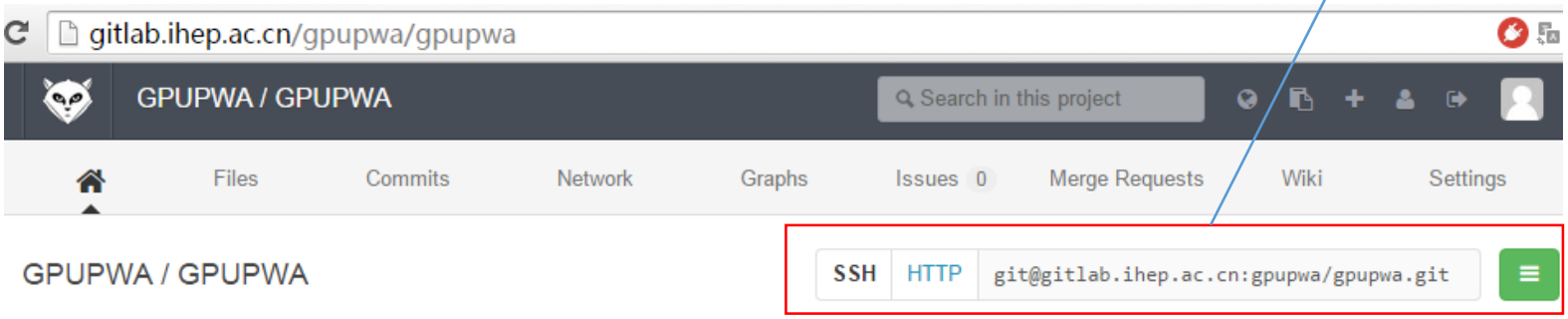


GPUPWA at gitlab :

(this link can be accessed even if you don't have a gitlab account)

http://gitlab.ihep.ac.cn/gpupwa/gpupwa

Address for clone this project



Clone the project

- If you have an account of the ihep gitlab, you can clone the GPUPWA project and then checkout version you are interested in:
- mkdir temp; cd temp;
- git clone git@gitlab.ihep.ac.cn:gpupwa/gpupwa.git

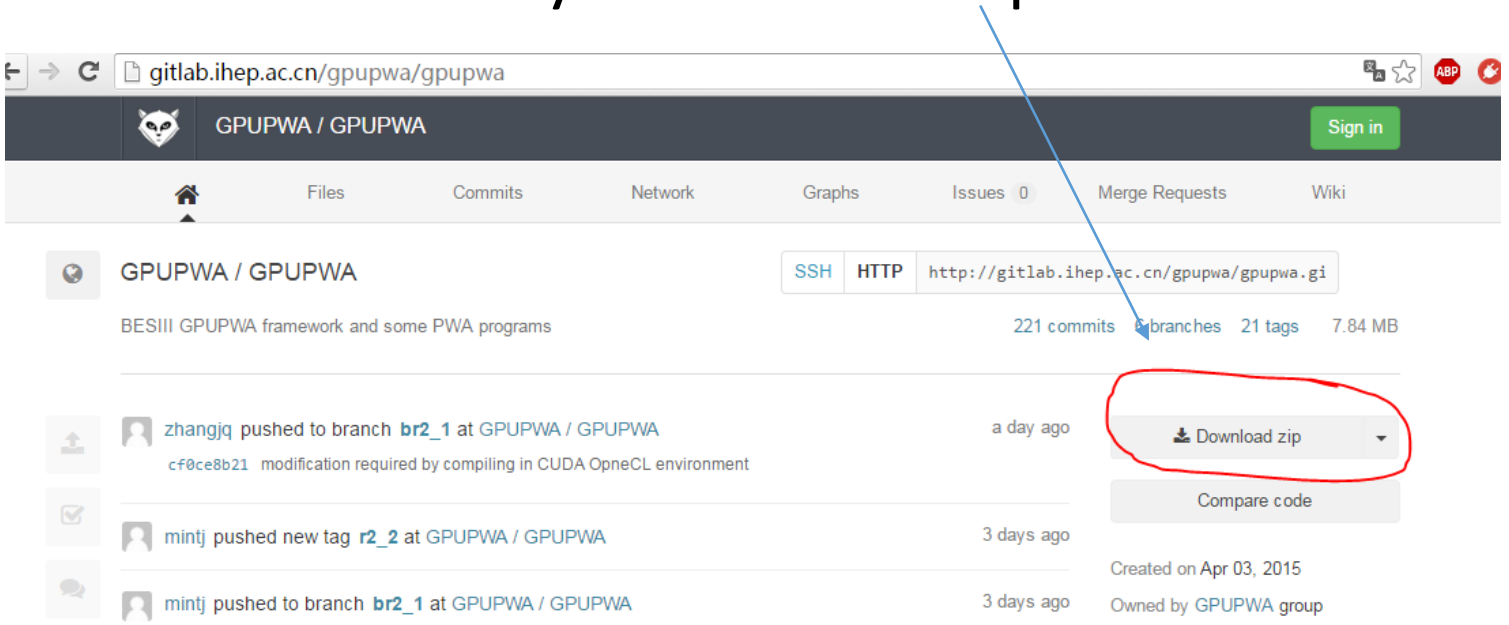
```
[zhangjq@gpu154 temp]$
[zhangjq@gpu154 temp]$ git clone git@gitlab.ihep.ac.cn:gpupwa/gpupwa.git
\Initialized empty Git repository in /besfs/groups/gpugroup/zhangjq/temp/gpupwa/.git/
Enter passphrase for key '/afs/ihep.ac.cn/users/z/zhangjq/.ssh/id_rsa':
remote: Counting objects: 2973, done.
remote: Compressing objects: 100% (638/638), done.
remote: Total 2973 (delta 2398), reused 2871 (delta 2331)
Receiving objects: 100% (2973/2973), 6.71 MiB | 8.90 MiB/s, done.
Resolving deltas: 100% (2398/2398), done.
[zhangjq@gpu154 temp]$ ls
gpupwa
[zhangjq@gpu154 temp]$ cd gpupwa/
[zhangjq@gpu154 gpupwa]$ ls
```

When this comes out, input the password of your git lab account

Clone the project (continued)

- If you don't have an account of gitlab.ihep.ac.cn, you may be not able to clone this project.

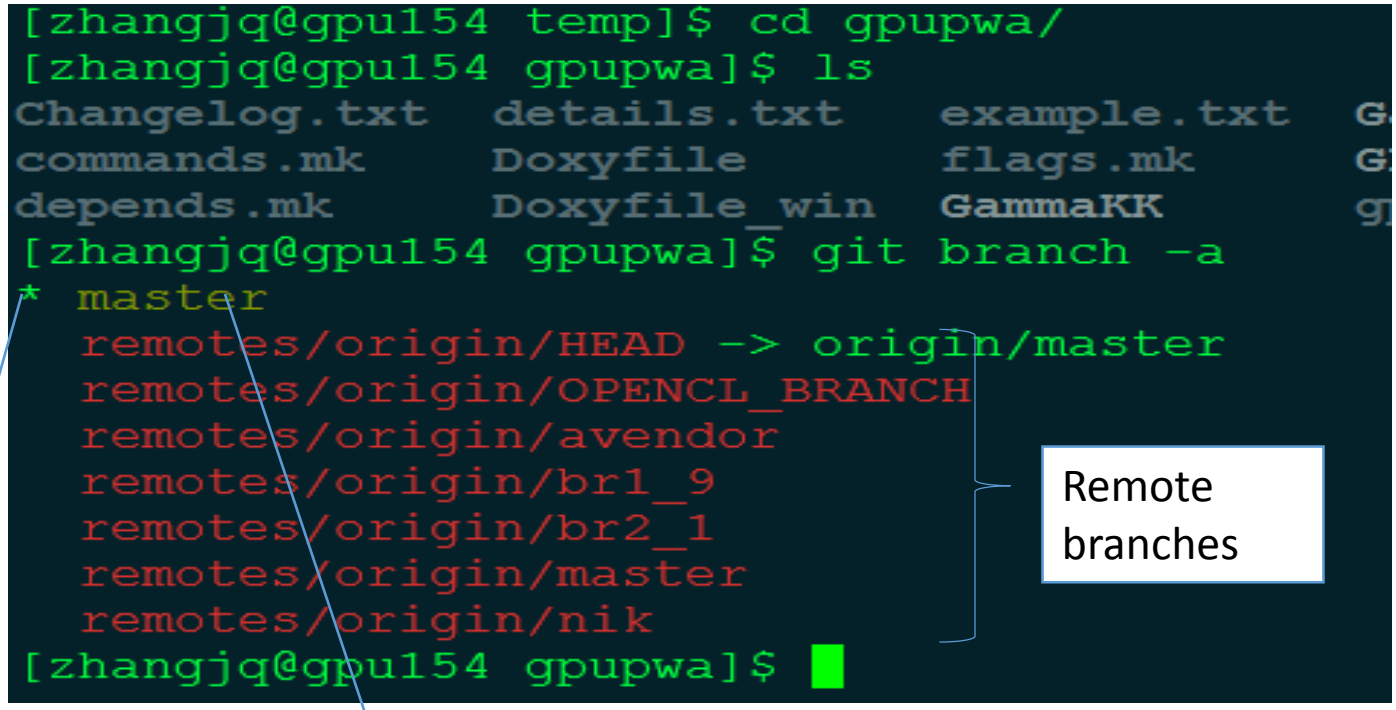
✓You can download the zip package to your PC, and then transfer it to your lxslc disk space



The whole repository is in the zip package. You can find what you get from a "git clone" in this zip package.

Branches in the project

- You can “use git branch –a” to see all the branches



Star means active branch: the branch you are in

Local branches

br1_9/br2_1 is the branch based on the CVS version r1_9/r2_1 and update some modifications required by new gcc/amdappsdk/atibrook

Tags in the project

- You can use “git tag” to see all tags of this project(one tag labeled one version)

```
[zhangjq@gpu154 gpupwa]$ git tag
arelease
r0
r0_1
r0_2
r1_0win
r1_1
r1_2
r1_3
r1_4
r1_5
r1_51
r1_6
r1_7
r1_8
r1_9
r1_9_1
r2_0
r2_0pre
r2_1
r2_1_1
[zhangjq@gpu154 gpupwa]$
```

tag

All of these tags are imported from CVS and each one represents one version in CVS history, except r1_9_1 and r2_1_1.

r1_9_1 and r2_1_1 labeled the version which can be run in gpu machines with gcc4.4.7 and new AMDAPPSDK/atibrook version/path.
The are updated from r1_9/r2_1 and only modified those which are required by new gcc/AMDAPPSDK/atibrook

Checkout the version you needed

- If you need the version 1.9 and hope it can run on the gpu machines. You need the version r1_9_1.
- You can check out the branch br1_9 by : git checkout br1_9

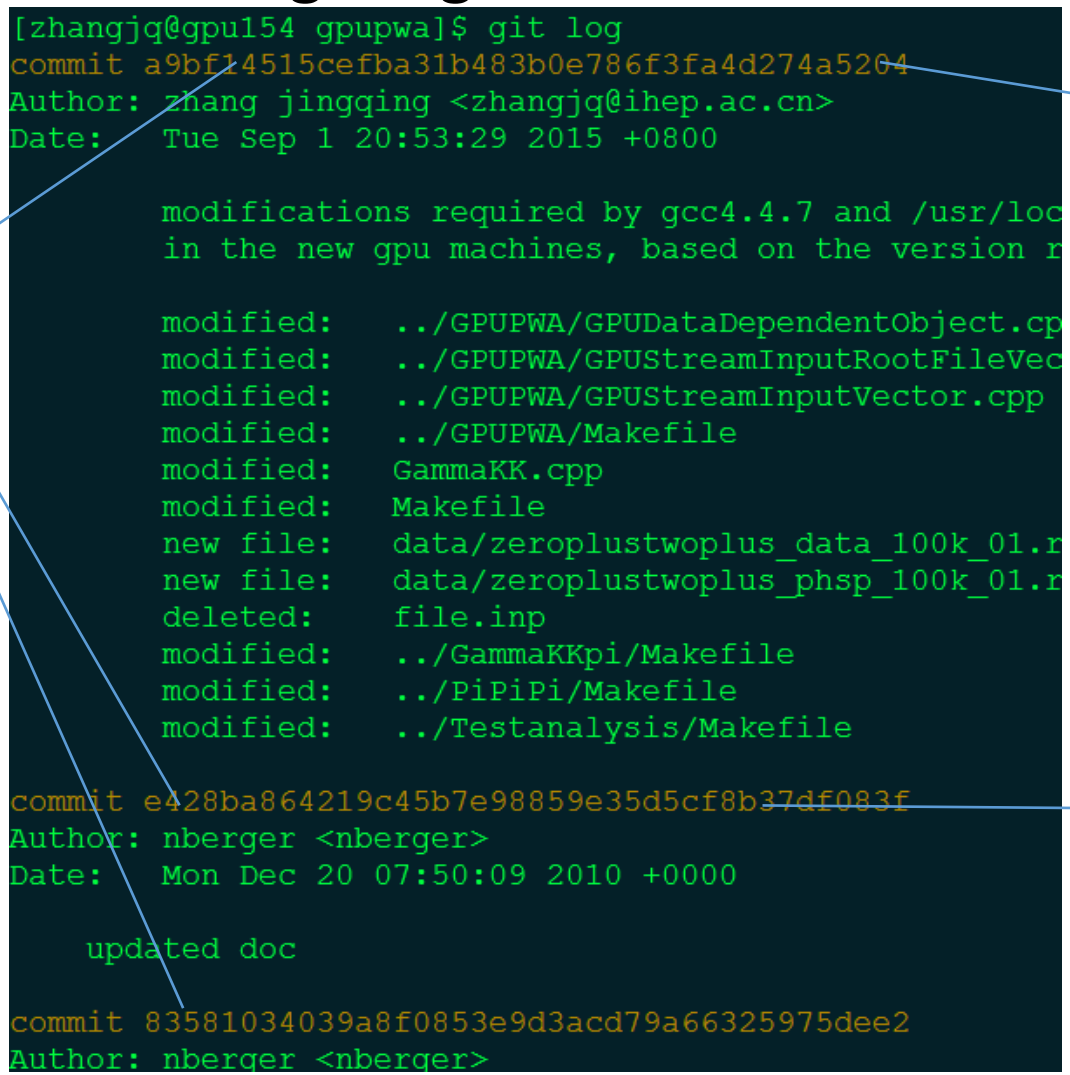
Star means active branch

Local branches

```
[zhangjq@gpu154 gpupwa]$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/OPENCL_BRANCH
  remotes/origin/avendor
  remotes/origin/br1_9
  remotes/origin/br2_1
  remotes/origin/master
  remotes/origin/nik
[zhangjq@gpu154 gpupwa]$ git checkout br1_9
Branch br1_9 set up to track remote branch br1_9 from origin.
Switched to a new branch 'br1_9'
[zhangjq@gpu154 gpupwa]$ git branch -a
* br1_9
  master
  remotes/origin/HEAD -> origin/master
  remotes/origin/OPENCL_BRANCH
  remotes/origin/avendor
  remotes/origin/br1_9
  remotes/origin/br2_1
  remotes/origin/master
  remotes/origin/nik
[zhangjq@gpu154 gpupwa]$
```

Checkout the version you needed

- And use “git log” to see the commit history



Commit ID

This version is the current version in branch br1_9 and this is the version r1_9_1.
You can do your modification based on this version

The version r1_9 in CVS history

Checkout r2_1_1

- If you want the version r2_1 and hope it can run on new gpu machines. You need r2_1_1

Checkout r2_1_1

This is the current version on branch br2_1. And this is the version r2_1_1

The version r2_1 in CVS history

```
[zhangjq@gpu154 gpupwa]$ git log
commit ab40e5a33180eaa9c155e4bc157abf3f1dc2caaa
Author: zhang jingqing <zhangjq@ihep.ac.cn>
Date:   Tue Sep 1 18:58:50 2015 +0800

        modifications required by gcc4.4.7 and AMDAPPSDK-2.9-1,
        based on version r2_1(the version in cvs r2_1/in gitlab tag r2_1)

        modified:   GPUPWA/Complex.cl
        modified:   GPUPWA/GPUDataDependentObject.cpp
        modified:   GPUPWA/GPUDataStream.cpp
        modified:   GPUPWA/GPUMinimizationHistory.cpp
        modified:   GPUPWA/GPUPWAAmplitudeCalculator.cpp
        modified:   GPUPWA/GPUPartialWave.h
        modified:   GPUPWA/GPUPartialWaveAnalysis.cpp
        modified:   GPUPWA/GPUPartialWaveAnalysis.h
        modified:   GPUPWA/GPUPropagatorType.cpp
        modified:   GPUPWA/GPUStreamInputVector.cpp
        modified:   GPUPWA/Opencl_interface/Compiler.cpp
        modified:   GPUPWA/Orbitals.cl
        modified:   GPUPWA/Propagators.cl
        modified:   GPUPWA/Tensors.cl
        new file:   GammaKK/data/zeroplustwoplus_data_100k_01.root
        new file:   GammaKK/data/zeroplustwoplus_phsp_100k_01.root
        modified:   GammaKK/files.txt

commit 1d905a827fa65576a618bb14df472fc5ad3add67
Author: nberger <nberger>
Date:   Tue May 17 03:41:44 2011 +0000

    Updated docu

commit 2ef678d5260b9b38c2f18b7baeda3888b1a65bea
Author: liubj <liubj>
Date:   Tue May 17 03:09:49 2011 +0000
```

Checkout a version with tag or commit ID

- You can also check out a version according to a commit ID or a tag:

git checkout ab40e

git checkout r2_1_1

```
[zhangjq@gpu154 gpupwa]$ git checkout r2_1_1
Note: checking out 'r2_1_1'.

You are in 'detached HEAD' state. You can look around, make
changes and commit them, and you can discard any commits yo
state without impacting any branches by performing another

If you want to create a new branch to retain commits you cr
do so (now or later) by using -b with the checkout command

  git checkout -b new_branch_name

HEAD is now at ab40e5a...       modifications required by g
n in cvs r2_1/in gitlab tag r2_1)
[zhangjq@gpu154 gpupwa]$ git branch -a
* (no branch)
  br1_9
  br2_1
  master
  remotes/origin/HEAD -> origin/master
  remotes/origin/OPENCL_BRANCH
  remotes/origin/avendor
  remotes/origin/br1_9
  remotes/origin/br2_1
  remotes/origin/master
  remotes/origin/nik
[zhangjq@gpu154 gpupwa]$
```

Now we are in here. This is the version r2_1_1.

You'd better first create a new branch and switch to it and then do your work:

git checkout -b my_branch_name

Switch on different branches

- You can use “git checkout branch_name” to swith the branches.
- But if you do not commit your current modifications and just checkout to another branch, your modifications will BE LOST.

Update the local/remote GPUPWA project

- You can use git pull to update your local GPUPWA project. Update of the GPUPWA in remote, i.e., GPUPWA@gitlab.ihep.ac.cn, will be checkout to your local project. (Please commit or backup the modification of your local project)
- You can use git push to push your local commit to the remote, i.e., the GPUPWA@gitlab.ihep.ac.cn. After successful git push, your modification will be in gitlab. (Please first use git pull and merge/solve the potential conflicts, and then use git push. Otherwise, you may not fail when use git push)

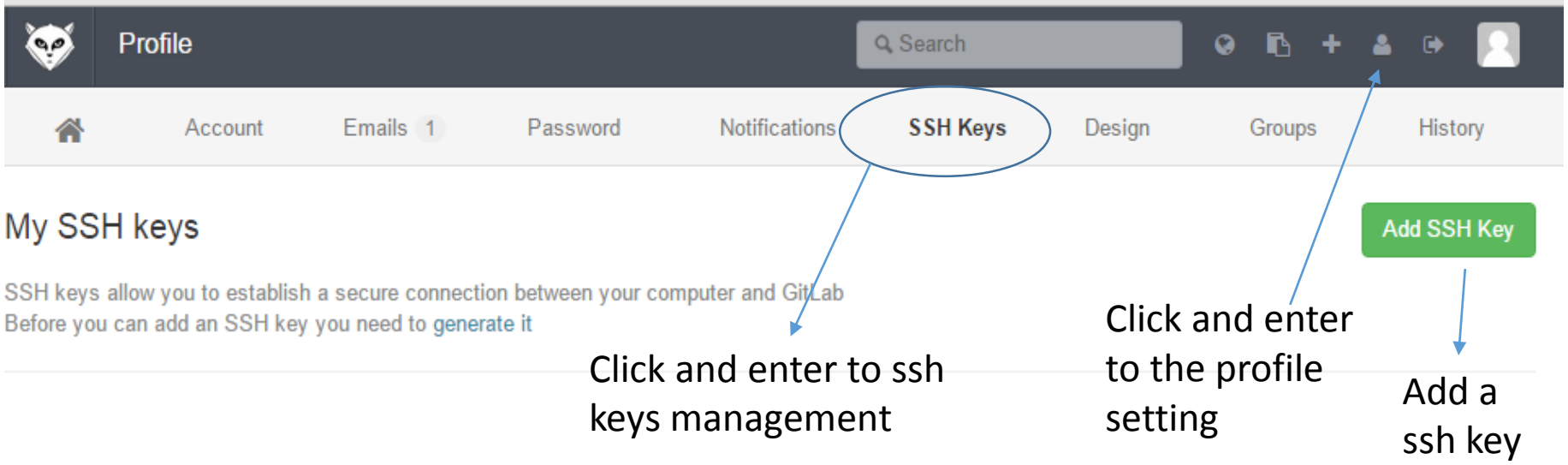
- You can find more information about git by google, for example:

http://www.bootcss.com/p/git-guide/

- Suggestion: Use a clean local project which has your modification/updates and a brief commit history when use git push

Potentail problem with git clone

- If you have some problem when git clone this project and you ensure that you have an account and the password is correct. Then maybe you need do some thing on your account's profile first, I guess.
- Login your account at gitlab and do some update according to the requirements of your profile setting



After adding a ssh key, you can try to git clone this project again

Part2: Set GPUPWA environment

Account for gpu machine/cluster

- Usually you need apply a gpu account to login a gpu machine (for developing program and test) and run jobs in gpu job queue. (Please contact wensp@ihep.ac.cn and liubj@ihep.ac.cn for application)

GPUPWA Environment

1. You need to set the path of OpenCL driver:

setenv AMDAPPSDKROOT /opt/AMDAPPSDK-2.9-1 (tcsh) or

AMDAPPSDKROOT= /opt/AMDAPPSDK-2.9-1 (bash)

The path /opt/AMDAPPSDK-2.9-1 may be different due to updation and installation.

2. You need to set the path of Root (different to those installed in lxslc):

setenv ROOTSYS /afs/ihep.ac.cn/bes3/gpupwa/root534/root (tcsh) or

ROOTSYS=/afs/ihep.ac.cn/bes3/gpupwa/root534/root (bash)

3. And add the two libs to LD_LIBRARY_PATH:

setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${AMDAPPSDKROOT}/lib/x86_64:${AMDAPPSDKROOT}/lib/x86:${ROOTSYS}/lib:/usr/local/lib (tcsh) or

LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${AMDAPPSDKROOT}/lib/x86_64:${AMDAPPSDKROOT}/lib/x86:${ROOTSYS}/lib:/usr/local/lib

4. You need to set the GPUPWA path:

setenv GPUPWA /xxx/xxx/the/path/to/your/GPUPWA/project/

For example: I cloned the project to /besfs/groups/gpugroup/zhangjq/temp/gpupwa, and if I want to develop/compile in this path, I should setenv GPUPWA /besfs/groups/gpugroup/zhangjq/temp/gpupwa

You can also copy/move the gpupwa to other path/names, just set the right path to GPUPWA

5. You need to set DISPLAY:

setenv DISPLAY :0.0 (tcsh) or

DISPLAY=:0.0

GPUPWA Environment (continued)

- Set the below environment variables if required when apply a front-test gpu account:

setenv BRT_ADAPTER your_val

setenv GPUPWA_GPU_NR your_val

setenv GPU_DEVICE_ORDINAL your_val

The below set may be also useful

- setenv PATH ${PATH}:$AMDAPPSDKROOT/bin/x86_64:$ROOTSYS/bin:/afs/ihep.ac.cn/bes3/gpupwa/bin

(you can set all of this variables in a file and source it when use gpu)

After all of this environment are set, we can run a GPUPWA program.

NOTE: do not set BOSS environment when use want to use gpu, because the Root in BOSS and in gpu are not same. Otherwise, your compiling will fail.

Run a GPUPWA program

- If you have got a GPUPWA copy, and set all the environment, then you can try to run a gpupwa program
- For example:

cd /besfs/groups/gpugroup/zhangjq/temp/gpupwa

make

cd GammaKK

cp _x86_64/gammakk .

./gammakk

And wait for the program finish

Note: do not cut a running gpu program process or a running gpu compiling process, otherwise, the gpu machine may not work and need reboot.

Run a gpu program

- If you made some modification in gpupwa/GammaKK/, you can make in GammaKK directory (if you have done make at the top directory at least one time).
- To easily debug, You can remove these directories by hand before make, and then make at the top directory:

GammaKK/_x86_64, GPUPWA/_x86_64, GPUPWA/_common/* (keep the _common directory), GPUPWA/Opencl_interface/_x86_64

- More information about GPUPWA

http://docbes3.ihep.ac.cn/twiki/bin/view/Main/PartialWaveAnalysisOnGPU

- And GPUPWA framework manual

http://docbes3.ihep.ac.cn/twiki/bin/view/Main/GpuPwa

General knowledge about OpenCL programming:

OpenCL in Action (My favourite ☺)

http://book.douban.com/subject/7163433/

http://www.amazon.cn/OpenCL-in-Action-How-to-Accelerate-Graphics-and-Computation-Scarpino-Matthew/dp/1617290173

Part 3: submit jobs in gpu queue

Submit jobs in gpu job queue

- There are two gpu job queue, gpuq@pbssrv02 and gpu2q@pbssrv02. gpuq@pbssrv02 only support GPUPWA_v1.9.x. gpu2q@pbssrv02 support GPUPWA_v2.1.x (and GPUPWA_v1.9.x)
- If you want to submit jobs to gpu queue, please use the script located at:

/afs/ihep.ac.cn/bes3/gpupwa/bin, they are just short sh script and python script.

gpu2q for gpu2q@pbssrv02 and gpuq for gpuq@pbssrv02.

You can copy the files you needed in your bin path and made some modification if necessary.

If you want to submit gammakk (the compiled executable file) to gpu queue.

Just go to the path where gammakk located at, and:

gpu2q gammakk

Then a job script is produced and the job is submitted to gpu queue.

You can use qstat to check the status, qstat –u username @pbssrv02.

Other pbs command can also be used if you specified the queue @pbssrv02.

Note: Please do not use qdel to delete the running job in gpu queue. If you do this, the gpu machine may do not work after this and may need reboot!