

上机作业A5 提示

2022年12月1日 19:42

任务

具体参考L13b课件的71~73页，在DevCloud上使用DPC++尝试异构编程，完成一系列的任务：

71页：Part I 熟悉DevCloud

1. 使用clinfo命令【自行尝试，不必提交】
2. 使用基本的Linux命令【自行尝试，不必提交】
3. 使用dpcpp编译第一个DPC++的程序（可以任选课上的示例代码），并在Login Node上使用CPU设备运行
4. 使用pbsnodes检查可用的计算节点【自行尝试，不必提交】
5. 编写Shell脚本，使用qsub命令提交到Compute Node上使用GPU设备执行，并检查返回的结果

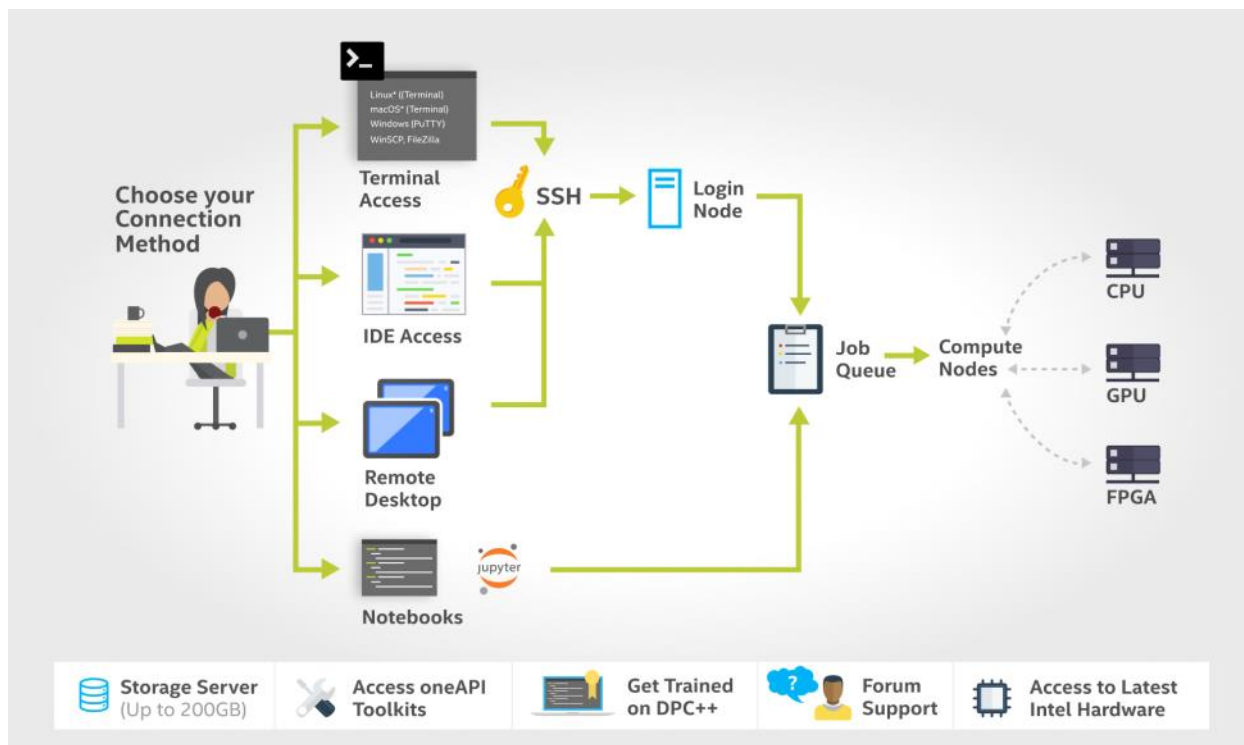
72页：Part II 根据要求更改DPC++的代码

1. 学习basic_parafor.cpp代码，了解本地与设备内存分配、拷贝、释放等函数的使用，以及并行计算的parallel_for函数的使用
2. 修改basic_parafor.cpp代码，将申请的内存空间修改为本机与设备共享的内存空间
3. 使用nd_range实现一个向量加法程序vector_add.cpp

73页：课后作业【选做】

1. 改写gemm_basic代码26, 27行，利用work group和local work item的坐标来计算global坐标
 2. 修改程序输入数据的大小，设定非 $M=N=K=2000$ ，修改程序，并使其通过正确性测试
- 课件71页之前的示例代码在代码仓库中也有提供，可以自行尝试，不做统一要求。

了解DevCloud的架构



参考: [Overview | Intel® DevCloud](#)

需要理解登陆节点 (Login Node) 与计算节点 (Compute Node) 之间的关系, 登录节点只有CPU设备, 如果需要在DevCloud上使用异构设备执行DPC++的程序, 需要借助作业队列 (Job Queue), 将作业提交到计算节点进行, 这样才能获取到CPU、GPU、FPGA全部类型的设备, 计算节点完成后会将结果 (标准输出流与标准错误流的输出) 以文本形式返回登陆节点。

推荐的作业提交方法

推荐大家全部在DevCloud上操作, 将原始代码仓库克隆到DevCloud上, 签出一个分支, 在分支上对代码仓库做根据任务要求做修改, 例如更改代码、编写作业脚本等, 中间生成的结果文件直接保留在仓库里。做完任务之后, 直接将该分支推送到水杉码园, 本次作业无需撰写实验报告。

1. 注册DevCloud, 登录并进入Jupyter Notebooks操作界面, 相当于一个在Login Node上的云主机
2. 在Jupyter中, 新建一个Console, 由于需要在DevCloud上使用git提交代码与结果到水杉, 需要初始化git的设置:
 - a. 使用水杉码园-右上角“设置”-个人信息栏目显示的用户名与电子邮箱地址, 在DevCloud上使用git config设置用户名与邮箱:

```
$ git config --global user.name yourname
$ git config --global user.email your@email.com
```
 - b. 在DevCloud上生成SSH密钥, 将生成的密钥添加到水杉码园设置的SSH/GPG密钥栏目中:

```
$ ssh-keygen -t ed25519 -C "your@email.com"
$ cat ~/.ssh/id_ed25519.pub
```
3. 从GitHub上克隆实验代码: [pengzhao-intel/oneAPI_course](https://github.com/pengzhao-intel/oneAPI_course): [oneAPI - Data Parallel C++ course for students \(github.com\)](https://github.com/pengzhao-intel/oneAPI_course/tree/master/Data%20Parallel%20C%20%2B%2B)

4. 进入oneAPI_course代码仓库，新建一个名为homework07的分支
\$ git checkout -b homework07
5. 在homework07的分支上根据任务要求进行操作，完成一个任务可以commit一次
6. 完成所有任务之后，将水杉码园的个人作业仓库添加为远程仓库，远程仓库名称可以命名为shuishan，例如：
\$ git remote add shuishan
https://gitea.shuishan.net.cn/Soft_Sys_Opt.James.2022Fall.DaSE/<stu_id>.git
其中<stu_id>是学号
7. 将DevCloud上本地的homework07分支推送到水杉码园的远程仓库，推送时需要输入水杉码园的用户名与密码，还需要先将水杉码园上原有的homework07分支删除后再进行操作（因为原有homework07分支已经有一个README.md文件，直接推送会产生冲突）：
\$ git push shuishan homework07

由于需要将任务写成脚本，提交到队列中等待Compute Node执行并返回结果，建议每一份代码对应一个脚本，每一个脚本对应返回的2个结果文件，例如：

- 代码：basic_parafor.cpp
- 脚本：basic_parafor.sh
- 结果文件：basic_parafor.sh.o2067174（标准输出流），
basic_parafor.sh.e2067174（标准错误流）

结果文件可以证明你已经完成了对应任务，**截图不必另外上传**。

完成任务后，直接按照上述流程提交你的分支到水杉码园的作业仓库即可。

脚本编写示例

作业是需要编写脚本提交到Compute Node进行的，在Login Node上使用source设置环境变量无法作用于Compute Node，因此需要在脚本的基本流程应当是：设置环境变量 → 编译 → 执行。

另外，同时为了了解作业的执行时间，可以在Shell脚本的开头与结尾添加显示时间的命令。

示例脚本gpu_selector.sh如下（工作目录为~/oneAPI_course/code）：

```
#!/bin/bash
echo
echo start: $(date "+%Y%m%d.%H%M%S.%3N")
echo

# TODO list
source /opt/intel/oneapi/setvars.sh > /dev/null 2>&1
dpcpp gpu_selector.cpp -o gpu_selector
if [ $? -eq 0 ]; then ./gpu_selector; fi

echo
echo stop: $(date "+%Y%m%d.%H%M%S.%3N")
echo
```

编写脚本后，使用qsub提交作业

```
qsub -l nodes=1:gpu:ppn=2 -d . run.sh
```

Note:

- `-l nodes=1:gpu:ppn=2` (**lower case L**) is used to assign one full GPU node to the job.
- The `-d .` is used to configure the current folder as the working directory for the task.
- `run.sh` is the script that gets executed on the compute node.

相关资料请参考: [Job Submission | Intel® DevCloud](#)