

OpenACC Performance

	Time in seconds (GPU)	Time in seconds (CPUs)			
Serial time	4.907738	6.369754			
Parallel threads	0.645384	2	4	8	16
Parallel time		3.111549	1.686830	1.098088	0.612320

Command Line Instructions

OpenACC on GPU:

1.Batch Script

```
[xsun2@bridges2-login014 openacc]$ cat cpu_script
#!/bin/bash
#SBATCH -A see200002p          # specify the project or allocation number
#SBATCH -p RM-shared# RM-shared partition
#SBATCH -J myjob              # Job name
#SBATCH --mail-user=sunxinyi@udel.edu
#SBATCH --mail-type=ALL

#SBATCH -N 1                  # Number of nodes, not cores (16 cores/node)
#SBATCH -n 16                 # Number of cores requested in total

#SBATCH -t 00:30:00          # set maximum run time of 30 minutes

export ACC_NUM_CORES=16
./mm_acc
```

2.Compile

```
[xsun2@bridges2-login014 openacc]$ module load nvhpc/21.7
[xsun2@bridges2-login014 openacc]$ nvc -acc -gpu=cc70 -Minfo=accel mm_acc.c -o mm_acc
main:
  30, Generating Tesla code
  32, #pragma acc loop gang, vector(128) /* blockIdx.x threadIdx.x */
  33, #pragma acc loop seq
  34, #pragma acc loop seq
  30, Generating implicit copyin(matrix_A[:1024][:1024],matrix_B[:1024][:1024]) [if not already present]
  Generating implicit copy(result[:1024][:1024]) [if not already present]
  33, Complex loop carried dependence of matrix_B->,matrix_A->,result-> prevents parallelization
  34, Complex loop carried dependence of matrix_B->,matrix_A->,result-> prevents parallelization
  Loop carried dependence of result-> prevents parallelization
  Loop carried backward dependence of result-> prevents vectorization
[xsun2@bridges2-login014 openacc]$
```

OpenACC on CPUs:

1. Batch Script

```
[xsun2@bridges2-login014 openacc]$ cat cpu_script
#!/bin/bash
#SBATCH -A see200002p          # specify the project or allocation number
#SBATCH -p RM-shared# RM-shared partition
#SBATCH -J myjob              # Job name
#SBATCH --mail-user=sunxinyi@udel.edu
#SBATCH --mail-type=ALL

#SBATCH -N 1                  # Number of nodes, not cores (16 cores/node)
#SBATCH -n 16                 # Number of cores requested in total

#SBATCH -t 00:30:00          # set maximum run time of 30 minutes

export ACC_NUM_CORES=16
./mm_acc
nsys profile -o cpufile --trace openacc ./mm_acc
[xsun2@bridges2-login014 openacc]$
```

2. Compile

```
[xsun2@bridges2-login014 openacc]$ nvc -acc -ta=multicore -Minfo=accel mm_acc.c -o mm_acc
main:
  30, Generating Multicore code
  32, #pragma acc loop gang
  33, Complex loop carried dependence of matrix_A->,result->,matrix_B-> prevents parallelization
  34, Complex loop carried dependence of matrix_A->,result->,matrix_B-> prevents parallelization
    Loop carried dependence of result-> prevents parallelization
    Loop carried backward dependence of result-> prevents vectorization
[xsun2@bridges2-login014 openacc]$
```

Execution Time(CPUs)

Thread(2)

```
[xsun2@bridges2-login013 openacc]$ more slurm-5271642.out
entered main function!
const set-up done!
matrix initialization done!
matrix multiplication done!
3.111549
[xsun2@bridges2-login013 openacc]$
```

Thread(4)

```
-bash: moreslurm-5271655.out: command not found
[xsun2@bridges2-login013 openacc]$ more slurm-5271655.out
entered main function!
const set-up done!
matrix initialization done!
matrix multiplication done!
1.686830
[xsun2@bridges2-login013 openacc]$
```

Thread(8)

```
[xsun2@bridges2-login013 openacc]$ more sl  
entered main function!  
const set-up done!  
matrix initialization done!  
matrix multiplication done!  
1.098088  
[xsun2@bridges2-login013 openacc]$
```

Thread(16)

```
[xsun2@bridges2-login013 openacc]$ more slurm-5271785.out  
entered main function!  
const set-up done!  
matrix initialization done!  
matrix multiplication done!  
0.612320  
[xsun2@bridges2-login013 openacc]$
```

Execution Time(GPU)

```
[xsun2@bridges2-login013 openacc]$ more slurm-5272106.out  
entered main function!  
const set-up done!  
matrix initialization done!  
matrix multiplication done!  
0.645384
```