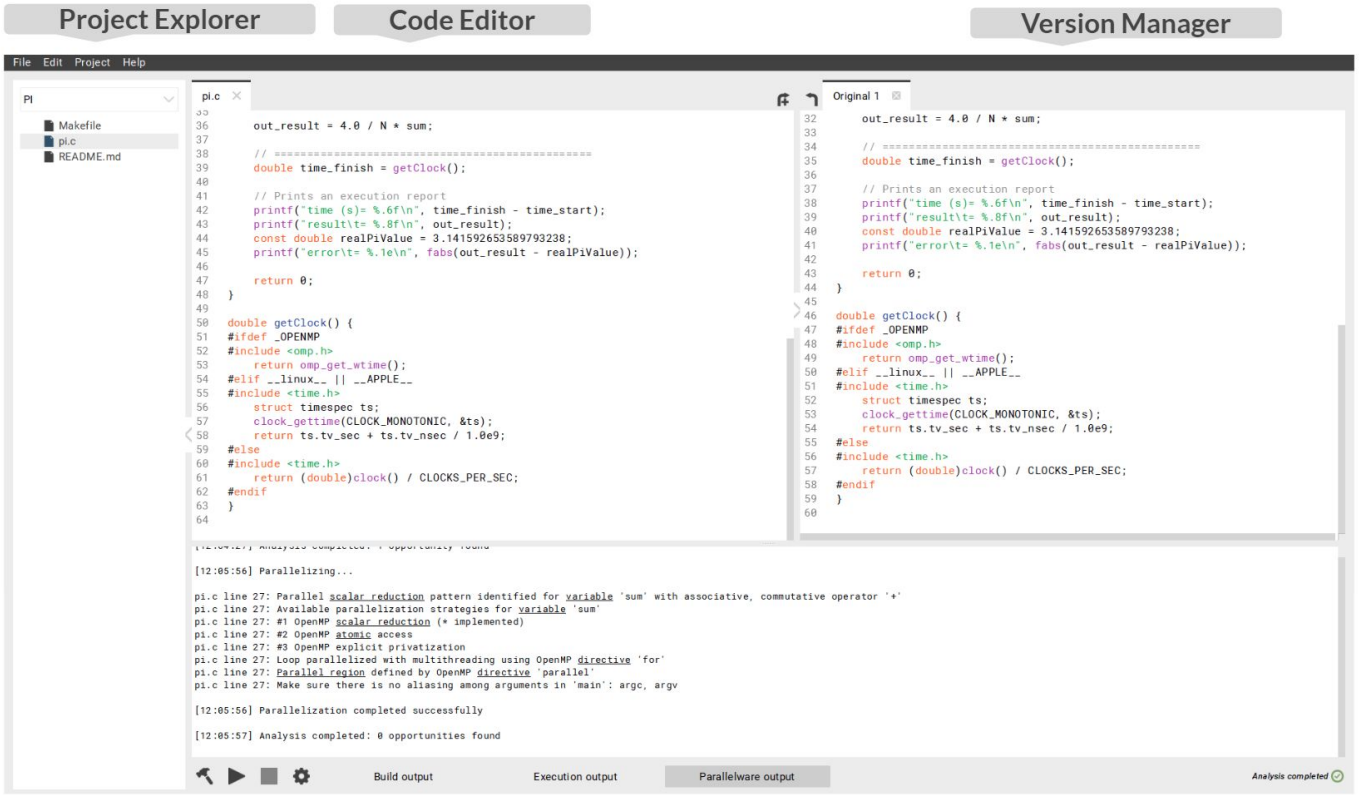
Parallelizing the calculation of π on the GPU with OpenMP/OpenACC

Goals:

- Build & run an OpenMP code on the GPU (for problem size N=900000000)
- Build & run an OpenACC code on the GPU (for problem size N=900000000)

Parallelware Trainer



Run Parallelware Trainer on CORI

Step 1: Log in to CORI enabling X11 forwarding

```
$ ssh -X <your_login>@cori.nersc.gov
```

Step 2: Copy the sample codes to be used during the course

```
$ cp -a /global/common/cori/software/pwtrainer-1.2.0_linux-x64/examples.zip ~
```

Step 3: Prepare the environment by loading the appropriate modules

```
$ module purge && module load esslurm cuda/10.0 gcc/8.1.1-openacc-gcc-8-branch-20190215 pgi/19.4 pwtrainer
```

Step 4: Open an interactive session in a GPU node

```
$ salloc -t 60 -N 1 -c 16 -C gpu --gres=gpu:2 --mem=32GB
```

Step 5: Run the Parallelware Trainer tool from the GPU node

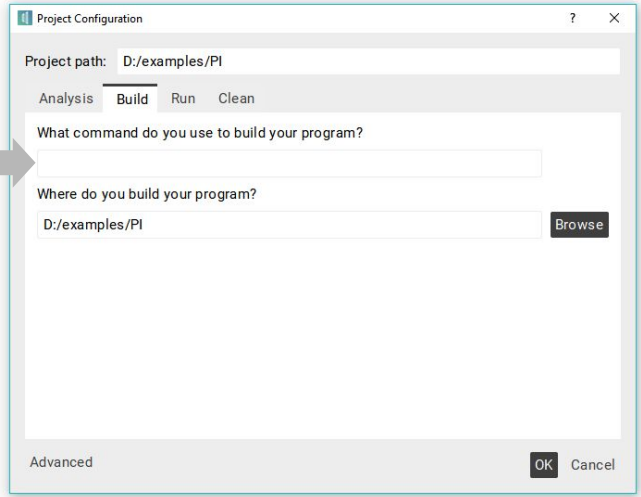
```
$ pwtrainer &
```

Step 6: Activate Parallelware Trainer

Click on 'Click here to install a license' and enter the following in the 'File name' text box:
/global/common/cori/software/pwtrainer-1.2.0_linux-x64/pwtrainer-Jun19-NERSC.lic

Build in Parallelware Trainer

	OpenMP Enabling HPC since 1997	OpenACC More Science, Less Programming
GCC	`gcc -fopenmp -foffload=nvptx-none="-Ofast -lm -misa=sm_35" -lm -o pi pi.c`	`gcc -fopenacc -foffload=nvptx-none="-Ofast -lm -misa=sm_35" -lm -D_OPENMP -o pi pi.c`
PGI	`pgcc -mp -ta=tesla:cc70,cuda10.0 -Minfo=accel -lm -o pi pi.c`	`pgcc -acc -ta=tesla:cc70,cuda10.0 -Minfo=accel -lm -D_OPENMP -o pi pi.c`

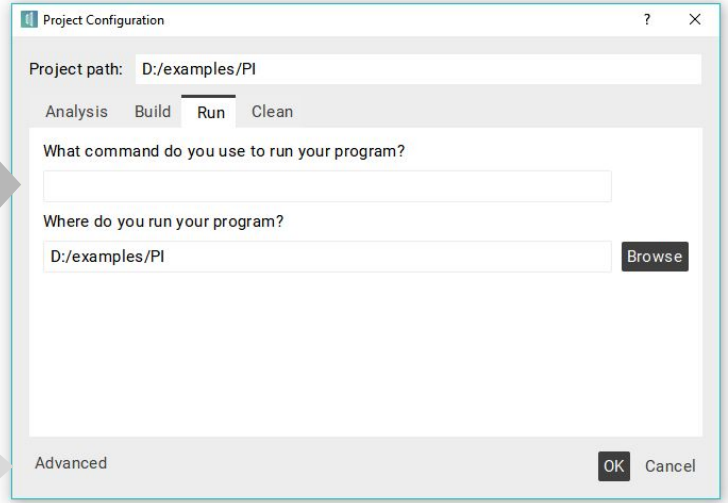


N.B. For Macs, check that your compiler supports OpenMP/OpenACC, and update with the appropriate compiler (e.g. gcc-mp-7)

Run in Parallelware Trainer

OpenMP Enabling HPC since 1997	`srun env OMP_NUM_THREADS=4 ./pi 900000000`
OpenACC More Science, Less Programming	`srun env ACC_DEVICE_TYPE=nvidia ./pi 900000000`

N.B. Instead of using the **env** command, you can add this variable by clicking the Advanced button.



Decomposition of PI into Patterns

Code file "pi.c"		Pattern			
Function	Line	Forall	Scalar reduction	Sparse reduction	Convergence loop
main()	29		sum		