



Q1: I implemented OpenACC directives into my code. When I run PGPROF and execute my code, I get some profiling information but I also get this some lines that look like this: ... (CCFF), line 693; error : Undefined object 'ACC259'.

A: Newer PGI compiler versions may contain messages unrecognized by the older profilers. These messages are skipped and this error messages is generated.

Q2: I like the hover help, does it have just the output from -Minfo=accel or does it contain -Minfo=all ?

A: The CCFF messages are generated at compile time with the -Minfo=ccff option. By default the CCFF messages are not generated by the compiler.

Q3: Can you capture trace one host from command line / via environment variables, and analyze it with GUI profiler on other one?

A: Yes, on the remote machine use `pgprof -o a.prof ./a.out`. Then on your local machine launch the Visual mode of PGPROF and select 'File -> Import'.

Q4: I am using OpenACC on a server and on my computer I don't have any NVIDIA GPU cards. Can I use PGI profiler?

A: Yes. PGPROF can still tell you about the performance on the CPU.

Q5: I can apply using fortran 90?

A: Yes, FORTRAN code is support by PGPROF.

Q6: Does guided analysis work with remote applications (e.g. on GUI-less node)?

A: Yes, on the remote system use: `pgprof --kernels <kernel specifier> --analysis-metrics -o a.prof ./a.prof` (More information on page 45 of the PGPROF users guide).

Q7: Can I increase the amount of stack / heap in the NetBeans IDE 8.0.2 shell using OpenACC?

A: NetBeans will treat OpenACC as pragmas in the code and I don't know of any plugins that provide additional OpenACC knowledge within the IDE, such as code completion or anything that would adjust the stack or heap at all.

Q8: If you have more than one GPU, can you target a specific card?

A: Yes, the OpenACC API has the `acc_set_device_num` function for selecting which device should be used. Please see lecture #4 from the original OpenACC course for more information.

Q9: Is it possible to use OpenACC in R?

A: OpenACC is only specified for C, C++, and FORTRAN, so there is no way to annotate R code directly with OpenACC. It is possible to write key functions in C, C++, or FORTRAN using OpenACC and then use language bindings to call those functions from other programming languages, however.

Q10: Is there a way to access the lab for free without registering a new account?

A: Taking the labs does require registering for an account, but should not require a new account. If you are having trouble using an existing account, please email openacc@nvidia.com with the email you used for your account and we'll get you help.

Q11: NVIDIA had announced LLVM based open source compiler last year. Is it any closer to release? For FORTRAN.

A: Effort continues and PGI is working towards a release late 2016 or early 2017.

Q12: like to know more about more complex routine usage, where nested functions call this func being marked by routine

A: For nested functions, each routine in the nest must be decorated.

Q13: Highly nonlinear problems can be extremely sensitive to order of operations. Does OpenACC and if so how does it address the issue?

A: Order of operations is a general problem with parallel programming. Some algorithms will get different, but still correct, answers when run in parallel (regardless of what is used to parallelize the program). Some compilers choose to add a reduced performance, bitwise reproducibility mode, but not all do. Programs that are sensitive to order of operations should investigate other, less sensitive approaches when targeting a parallel architecture (not just GPUs).

Q14: Is there a way to use OpenACC for Python?

A: No, OpenACC supports just C, C++, and FORTRAN. One could choose to use language bindings to call into an OpenACC-accelerated routine from Python, but OpenACC cannot be used directly in Python. Python users should take a look at Anaconda Accelerate for similar acceleration features.