OOKAMI PROJECT APPLICATION

Date: 22/03/28

Project Title:Calculation of electron collisions with molecular targets using the convergent close-coupling method

Usage:

- Testbed

Principal Investigator: Barry I. Schneider

- National Institute of Standards and Technology, , Gaithersburg, Maryland 20899-8422, USA
- 703-589-2528
- bis@nist.gov

Co Principal Investigator: Igor Bray

- Department of Physics and Astronomy, Curtin University, Perth, Western Australia 6102, Australia
- 08 9266 4416
- igor.bray@curtin.edu.au

Names & Email of initial project users:

Barry I. Schneider: bis@nist.gov
Igor Bray: igor.bray@curtin.edu.au

Usage Description:

Over the past few years the Co-PIs have been extending the very successful atomic convergent close coupling (CCC) method to small molecules such as H_2 and LiH by employing the prolate spheroidal coordinate system. The CCC method is a systematic and state-of-the-art approach to study electron scattering in quasi one-and-two electron atoms and molecules. Recent publications in

Phys. Rev.(1) and Phys. Rev. Letts.(3) are indicative of how our colleagues view the research.

While the PIs have benefited significantly from having access to a number of the XSEDE supercomputers and Frontera, we felt it would be quite interesting to see how our codes would perform on OOKAMI, given its very different hardware. The code has already been shown to perform quite well on GPUs as well as CPU based supercomputers and we would be quite interested in seeing how it will perform on the ARM processors. This is a request for a 10,000 node hour testbed allocation. Not knowing anything more at present on whether our codes can run on OOKAMI, and if so, how efficiently, it is difficult to say more.

Computational Resources:

- Total node hours per year: 10,000
- Size (nodes) and duration (hours) for a typical batch job: Variable but in initial development phase is likely to vary between 1 and 30 nodes, from minutes to an hour or so.
- Disk space (home, project, scratch): Standard allocations will suffice at the outset.

Personnel Resources (assistance in porting/tuning, or training for your users):

This will depend greatly on how the initial phase of compiling, linking and running the test calculations proceeds. We have always benefitted from the support given at other NSF supercomputer centers when we had issues. And, there were issues, especially with the GPU porting to Expanse.

Required software:

Fortran compilers, mathematical libraries equivalent to Intel MKL including ScalaPack. Thus far, the CCC code has been shown to perform well utilizing Intel, PGI and gfortran compilers. Presently, the implementation of a Cray fortran compiler with AMD CPUS and GPUS is under development. The GPU acceleration is implemented using OpenAcc directives.

If your research is supported by US federal agencies:

- Schneider is supported by NIST and has an XSEDE award
- Bray has several grants from the Australian Research Council.

Production projects:

1. The code performs well on the XSEDE machines Stampede2, Expanse and Bridges2, and also on Frontera of TACC.

2. The code has both a CPU and GPU modes of operation