

## Lessons Learned Developing Performance Portable QMCPACK

Date: May 10, 2023

Presented by: Paul Kent (Oak Ridge National Laboratory)

(The slides will be available under “Materials from the Webinar” in the above link.)

---

**Q.** Is the data movement managed using unified memory?

**A.** No, because support for unified memory wasn't as widespread back then, and we didn't want to rely on it being available and performant. We also want to be able to run on older hardware too that might not have that capability. However the new design supports.

**Q.** I wish MiniQMC keeps getting updated, the HPCToolkit team has been using it as one of the codes as a tutorial example... maybe we should switch to actual QMCPACK now with a tiny input file? (I'll follow up later with an email)

**A.** Good point. Unfortunately keeping the miniapp current takes resources and we are fully focused on the main application at the moment. What to do post ECP is a question, we would like to maintain the miniapp “sustainably”. E.g. As I mentioned the Kokkos version is supposedly broken since it has not been updated in a few years. While it might only take a few minutes to fix, someone still has to do it. And of course the fix might take longer.

**Q.** Are you using NVIDIA GPU MPS on Summit and Perlmutter for the multiple crowd approach, or only relying on asynchronous GPU kernel launches? Not sure what the AMD and Intel equivalent for MPS. And how do you manage multiple CPU threads “talking” to the same GPU?

**A.** Not using MPS. We rely on async and it is critical that the runtimes support async well and do not unnecessarily serialize. The current design is more efficient than MPS, particularly in terms of memory. QMCPACK runs are often memory limited, so using the minimum number of processes to drive the GPUs is important. Work from different CPU threads is mapped to different GPU streams by the compiler runtime. The vendors and compiler teams have improved these assignments.

**Q.** Is the delayed update algorithm numerically equivalent or are you trading some accuracy?

**A.** The equations are mathematically equivalent but in practice there are some small numerical differences. Stability appears to be the same as the old algorithm but it is something we will investigate further.

**Q.** Do you perform facility testing too? (e.g. actually running QMCPACK on Summit/Frontier/Perlmutter/Sunspot with small inputs over several compute nodes)

**A.** Yes but not automatically. We run by hand, but it would be much more efficient to have this done automatically. We can size the runs to the resource availability. Knowing whether the code compiles and the unit tests pass can be done very cheaply. This would catch compiler, library, wrapper (etc.) problems promptly and let us be more efficient.

**Q.** Any comments on compiling QMCPACK with Cray CC, AMD ROCm amdclang, or Intel icpx?

**A.** These are moving targets. Email for current status.