I. SCALING INFORMATION

A. Performance/Scaling of the B-Spline Atomic R-Matrix Code

Figure 1 provides some scaling information for major parts of the BSR package. While BSR_BREIT and BSR_MAT scale well until about 800 cores in the example shown, the performance of BSR_HD deteriorates rapidly when the number of cores exceeds about 400-500. This lack of scaling efficiency in the BSR_HD step is due to; 1) the SCALAPACK [1] diagonalization. and 2) the ALL-TO-ALL operation of required to collect the eigenvectors from the distributed matrix. By simply collecting only the surface amplitudes, the scaling would be improved significantly. This will be implemented in the codes during the upcoming year.

Specifically, runs were performed on the Knight's Landing (KNL) and Skylake (SKX) nodes on Stampede 2 with 256, 512, 768, and 1024 cores. We used 64 cores per node on the KNL nodes and 48 cores per node on the SKX nodes. On SuperMIC, 20 cores per node were used. If nodes rather than cores are a measure of performance then the KNL system on Stampede-2 and SuperMIC are comparable. The figure also shows that the SKX cores outperform the KNL cores by about a factor of three. This is disappointing but in line with what others have seen on the two systems. Unfortunately, we found that using more than one thread on the KNL cores slows the program down significantly. We currently do not have a clear explanation for this, but we intend to look into it in the next allocations period. Finally, while we realize that SuperMIC is no longer available to XSEDE, we trust that this information is still useful. It basically shows that we can expect to move rather easily to other machines, e.g., Comet and Bridges.

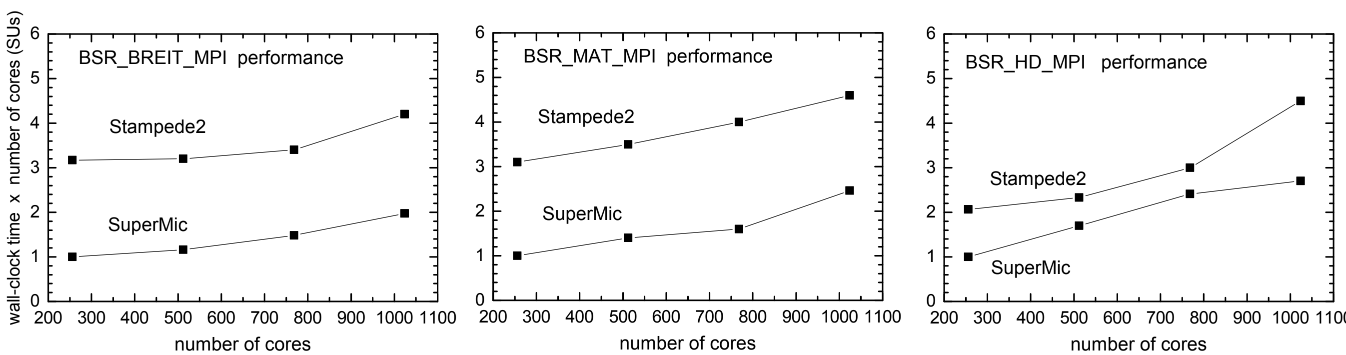

FIG. 1. The product of wall-time and the number of cores (normalized to the smallest number as a function of the latter) for `BSR_BREIT`, `BSR_MAT`, and `BSR_HD` on Stampede 2 and SuperMIC. The rank of the matrix was approximately 50,000.

We emphasize that in practice we do *not* waste SUs to the extent illustrated above, where doubling the number of cores from 512 to 1,024 increased the product of wall-time and the number of cores by a factor of $\approx 2 - 2.5$ on Stampede 2, i.e., it resulted in an increased wall-time as well. In fact, we always try to optimize the number of cores by using the following criteria:

- The matrix has to fit into the available memory, which sets the minimum number of cores.
- Increase the number of cores only to the extent that scaling does not deteriorate significantly while also ensuring that the maximum allowed wall-time is not exceeded.

This strategy ultimately sets the maximum size of the matrices that we can handle, and hence the size of the physical problems that we can tackle. To date, we have been able to deal with matrices of rank up to 400,000. Running `BSR_HD` represents the most expensive part of our work, and it has to be done for every partial-wave symmetry.