Predictions of Large-scale QMCPack I/Os on Titan using Skel

Matthew Wezowicz[1], Michael Matheny[1], Stephen Herbein[1], Jeremy Logan[2], Jeongnim Kim[2], Jaron Krogel[2], Scott Klasky[2], Michela Taufer[1]

[1] University of Delaware [2] Oak Ridge National Laboratory

ABSTRACT

Large-scale applications such as the Quantum Monte Carlo code QMCPack face a major challenge to preserve their scalability when fine-grained data is gathered and stored to disk or used for in-situ analysis. Using an I/O framework such as ADIOS allows us to address the trade-off by deploying different I/O methods with little code modifications. Still, the search for the most suitable methods and settings can be challenging. Ideally such a search can be conducted using a tool such as Skel that allows us to decouple the I/O from the computation in real applications and to tune the I/O on real supercomputers. To allow tuning for applications with unbalanced generation of I/O, we extended Skel to integrate the I/O variability shown by real world applications. We validated the Skel results against actual QMCPack I/O performance data and showed that Skel results can be used to predict I/O and steer choices for applications.

1. MOTIVATION AND GOALS

As supercomputers begin to move towards exascale, there is an increasing discrepancy between I/O performance and flops. Developers must balance the trade-off between application scalability and the amount of data they want to save. QMCPack, a Quantum Monte Carlo code, has reached this point for petascale. QMCPack's existing I/O methods prevent saving fine-grain data at scale. The current solution is to save averages that provide only a general view.

Users want to be able to tune their I/O configuration to maintain scalability while achieving maximum I/O performance. Ideally this can be done without the expense of running the application. One such tool for this task is Skel [3]. Skel takes an I/O descriptor and generates a full I/O benchmark, but it has several limitations. Our main goal is to extend Skel to remove some of these limitations and show that Skel can be used to steer design choices by validating the simulation results against a real application such QMCPack at the large scale.

2. BACKGROUND

QMCPack is a Quantum Monte Carlo simulation that finds solutions to Schrödinger's equation [1]. The algorithm is loosely coupled and the code exploits both the CPU and GPU. In theory, QMCPack is the perfect exascale application.

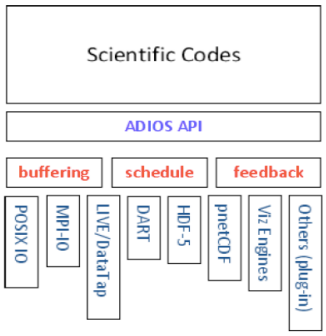


Fig. 1 Overview of ADIOS

ADIOS is an I/O framework that provides a suite of simple APIs (Figure 1), while using metadata that is stored in an external XML file [2]. The metadata contains a description of the data and how the data should be written out to disk. The simple API allows for minimal changes to existing code bases while the metadata enables I/O method switching without recompiling the code.

Skel is a tool for generating an I/O skeleton for benchmarking. Skel uses the ADIOS metadata descriptor to generate C or Fortran source code. The current implementation of Skel has a major limitation that reduces its applicability to applications with variable I/O: it uses a fixed I/O size per "time step."

3. MODELING VARIABLE I/Os

A fixed I/O model is not realistic for many applications, QMCPack included. QMCPack processes have a variable number of trial solutions called walkers. As the simulation evolves, walkers are generated or terminated based on their energy values making the I/O of each process variable in size.

We modeled the variability of the number of walkers through the behavior of a spring. We approximate the dampened spring equation using the Midpoint method, and then make small random changes to the spring's position. We empirically validated the typical fluctuations of the number of walkers per process versus the values generated by the model and found similar patterns as shown in Figure 2.

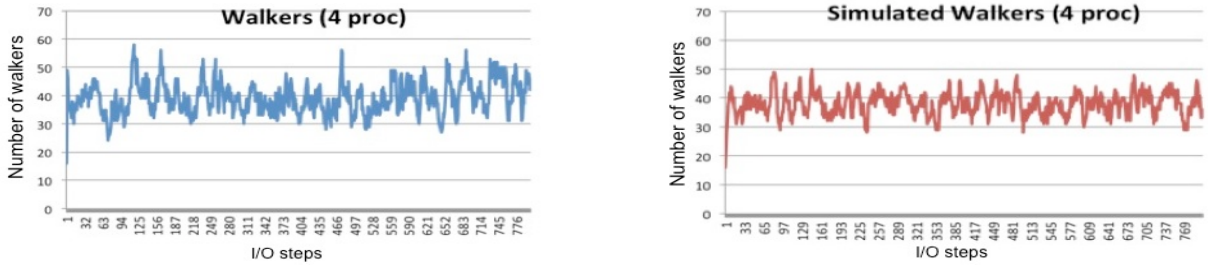
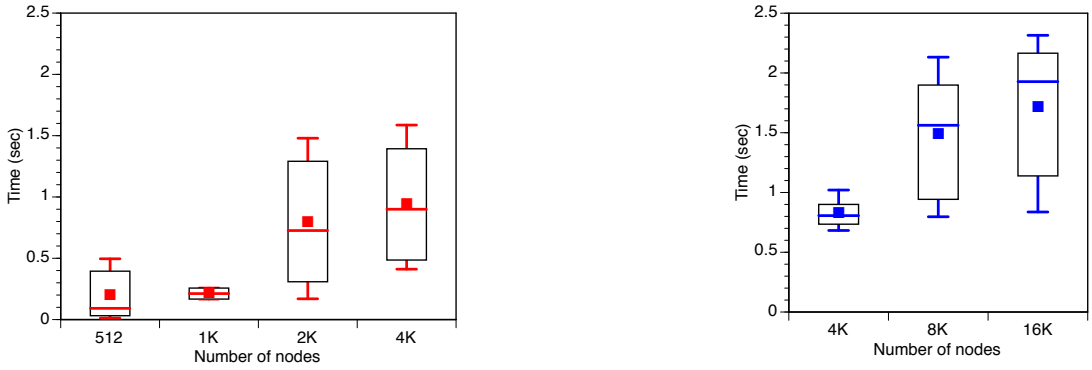


Fig. 2 Empirical validation of simulated number of walkers.



(a) (b)

Fig. 3 I/O times of QMCPack simulations at the small scale (a) and of Skel mimicking the QMCPack I/Os at the large scale (b).

We extended Skel to integrate user-defined models for the specification of variable I/O sizes. Our extension allows these size models to be described through inline C and Fortran functions or by the inclusion of code from files. This was accomplished by adding support for additional tags to the Skel's XML description of the I/O. Note that the correctness of benchmarks generated by Skel was validated in previous work [3] and, thus, the validation is not part of the poster contributions.

4. TESTS AND PREDICTIONS

Our goal is to estimate the I/O time in QMCPack with ADIOS at a large scale (i.e., on all the nodes of Titan). To this end, we first measured I/O times of QMCPack simulations at the small scale, i.e., on 512, 1K, 2K and 4K nodes with two MPI processes per node, using ADIOS MPI AGGREGATE as presented in Figure 5(a). When moving to a larger scale, we used Skel rather than QMCPack to predict the code's I/O performance on 4K, 8K and 16K nodes using the same ADIOS method, as presented in Figure 5(b). Note that we considered the ADIOS MPI_AGGREGATE method for the measurements because it is more stable in QMCPack runs and generally performs better, as we will show in Figure 4. Overall, Skel allows us to identify the cost and variability of QMCPack at a lower cost in terms of resources and time.

The comparison of ADIOS MPI_AGGREGATE and POSIX methods was performed on the Titan supercomputer for the same graphite system but for simulation sizes of 128, 256, 512, 1024, 2048, and 8192 nodes with two MPI processes per node. Figure 4 shows how the performance of the different I/O methods on Titan follows the expected trend of increasing divergence between POSIX and MPI_AGGREGATE due to increased strain on the Lustre metadata server. The figure also shows how the ADIOS's MPI_AGGREGATE method exhibits excellent scalability when the number of nodes reaches larger values. In exascale systems this method is expected to mitigate I/O costs and support application scalability.

5. CONCLUSION AND FUTURE WORK

In this paper we present the expansion of Skel to integrate models of variable I/O in scientific applications. After validating the modified Skel on a small-scale cluster, we used it to predict the I/O of QMCPack simulations on a large scale. Our results show the potential of the ADIOS MPI_AGGREGATE method to support low-cost I/O on exascale platforms.

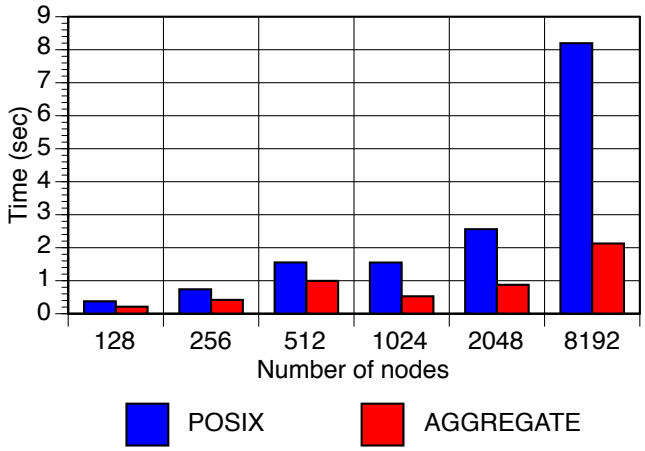


Fig. 4 I/O predictions of QMCPack on Titan.

In future work, we plan on adding additional capabilities to Skel, including the capability to specify variable I/O with a library of distributions and user specified lists that can emulate traces of actual execution. Simulation of read patterns is another feature we wish to add to allow developers to understand the different factors in optimizing for read access.

ACKNOWLEDGMENTS

S. Herbein, M. Matheny, and M. Wezowicz equally contributed to this work. This work is supported in part by the NSF grants: CNS 1318417 and CNS 1217812. It was also partially supported by the projects Predictive Theory and Modeling for Materials and Chemical Science by the Department of Energy (DOE), Basic Energy Science (BES), the Laboratory Directed Research and Development Program of ORNL, and the DOEs SULI Program. The authors want to thank Dr. Norbert Podhorszki for his advice on using ADIOS.

REFERENCES

[1] Kim et al., Hybrid Algorithms in Quantum Monte Carlo. *Journal of Physics: Conference Series* 402:1, 012008, 2012.

[2] Liu et al., Hello ADIOS: The Challenges and Lessons of Developing Leadership Class I/O Frameworks. *Concurrency and Computation: Practice and Experience*, August, 2013.

[3] Logan et al., Skel: Generative Software for Producing Skeletal I/O Applications. *eScience Workshops*, 2011.