

Sarah Neuwirth*, Feiyi Wang^S, Sarp Oral^S, Sudharshan Vazhkudai^S, and Ulrich Bruening*

*University of Heidelberg, Germany, {sarah.neuwirth,ulrich.bruening}@ziti.uni-heidelberg.de, ^SOak Ridge National Laboratory, USA, {fwang2,oralhs,vazhkudaiss}@ornl.gov

Problem Statement

- Large-scale scientific applications' usage patterns lead to I/O resource contention and load imbalance
- Implementation of a dynamic, shared library based on BPIO, a method to resolve contention, provides a transparent way to balance resource usage without source code modification or recompilation

Introduction

Balanced Placement I/O (BPIO) Library

- Topology-aware and balanced data placement [1]
- Resolves application-level I/O contention
- Computes placement cost for each I/O client based on a tunable, weighted cost function:

$$Placement\ Cost = w_1 * R_1 + w_2 * R_2 + w_3 * R_3 + w_4 * R_4$$

R_i : resource component w_i : weight factor

- Available as a user space library, but requires direct integration into the source code

Aequilibro – Integrating BPIO with ADIOS

- ADIOS [2] provides portable, fast, scalable, easy-to-use, metadata rich output and I/O interfaces can be changed during runtime
- Aequilibro [3] combines the optimization done at the interconnect level by BPIO with the benefits of the ADIOS I/O framework

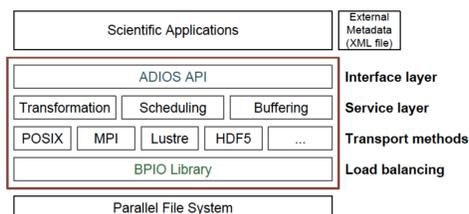


Fig. 1: Aequilibro software stack.

Research Objectives

- Design and implement a preloadable, shared library based on the BPIO library
- Evaluate the performance of the framework with a synthetic benchmark (IOR) for POSIX I/O and MPI-IO
- Evaluate with real-world HPC workloads on Titan

Aequilibro Performance Results

- IOR synthetic benchmark (POSIX and MPI-IO):
 - Setups: (I) Default, (II) ADIOS, (III) BPIO, (IV) Aequilibro
 - 10 scaled runs per test case on Titan, 3 repetitions per run
- Real-world HPC workload based on S3D physics code
- Metrics of interest:
 - Performance improvement with BPIO
 - Average bandwidth per second (GB/s)



Fig. 2: Titan supercomputing system.

$$Performance\ Improvement = 100 * \left(\frac{Bandwidth_{IOR_BPIO} - 1}{Bandwidth_{IOR_default}} \right) [1]$$

- Test System:
 - Titan
 - Cray XK7
 - 18,688 nodes
 - Spider II (Lustre-based) file system
 - 144 OSSs
 - 1,008 OSTs

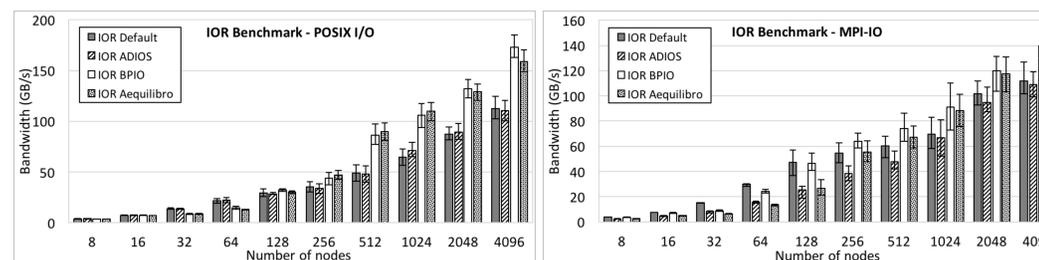


Fig. 3: IOR bandwidth performance for setup (I) to (IV) for POSIX I/O and MPI-IO including errors bars.

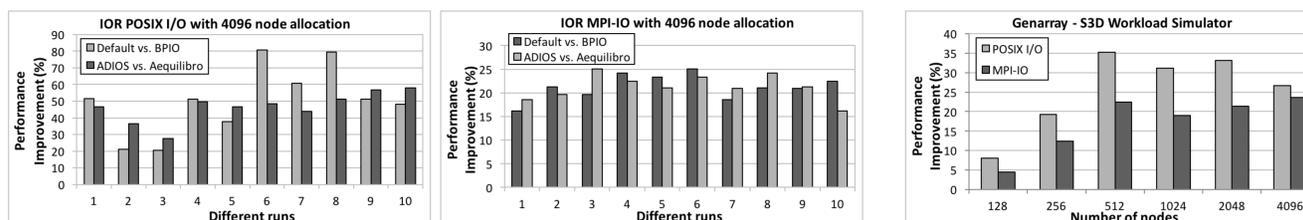


Fig. 4: Performance improvement for IOR (I) vs. (III) and (II) vs. (IV) at large-scale.

Fig. 5: S3D performance improvement.

Balanced Placement I/O Framework – BPIO 2.0

BPIO Runtime Environment

- Build as a shared, preloadable library (LD_PRELOAD)
- Utilizes BPIO library for balanced data placement
- Uses function interposition to prioritize itself over standard function calls
- End-to-end and per job load balancing
- Supported I/O interfaces include POSIX I/O and MPI-IO; HDF5 is under development

Dynamic Instrumentation

- Wrapper functions to intercept I/O functions
- Internal functions to initialize and maintain internal data structures and module-specific I/O characterization data
- Set of functions to interact with the BPIO library

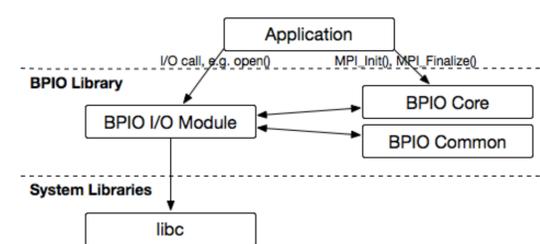


Fig. 6: BPIO runtime environment.

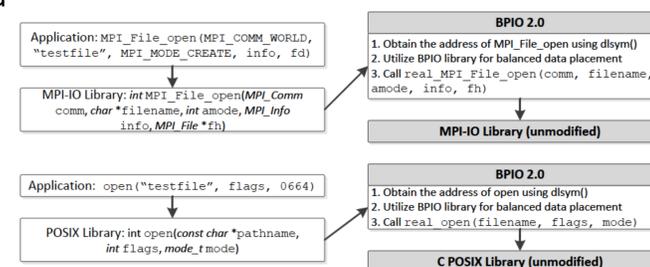


Fig. 7: Dynamic interception of I/O functions at runtime.

Experimental Results with BPIO 2.0

- Initial performance evaluation of the preloadable BPIO framework with the IOR benchmark
- Performance improvement and average bandwidth per second are similar to the direct integration of BPIO into an application

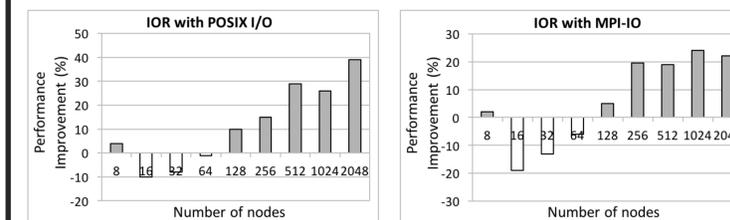


Fig. 8: Performance improvement for IOR BPIO 2.0 vs. IOR Default.

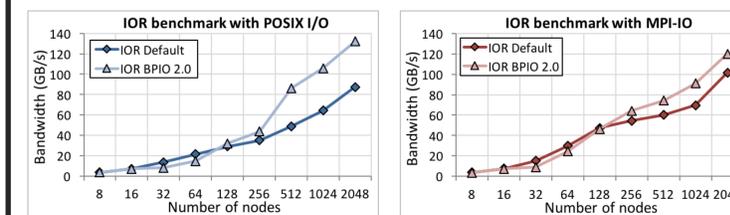


Fig. 9: Bandwidth performance for IOR Default and IOR BPIO 2.0.

Conclusions

- Aequilibro provides performance improvement, but requires the explicit BPIO integration into ADIOS' transport methods
- BPIO 2.0 is built as a dynamic library so it does not require any code modification or recompilation
- Initial experiments show similar performance improvement trends as a direct BPIO integration
- Ongoing and future work:
 - Single shared file support for MPI-IO
 - Extensive real-world HPC workload evaluation
 - Performance comparison of Aequilibro and BPIO 2.0
 - Support of HDF5

References

- F. Wang, S. Oral, S. Gupta, D. Tiwari, and S. Vazhkudai, *Improving Large-scale Storage System Performance via Topology-aware and Balanced Data Placement*, In ICPADS '14 (pp. 656-663).
- J. Lofstead, S. Klasky, K. Schwan, N. Podhorski, and J. Chen, *Flexible IO and Integration for Scientific Codes through the Adaptable IO System (ADIOS)*, In CLADE '08 (pp.15-24).
- S. Neuwirth, S. Oral, F. Wang, Q. Liu, and S. Vazhkudai, *Improving Large-scale Application Performance with ADIOS and BPIO*, Poster at SMC '15.