## Problem Statement
- Large-scale scientific applications’ usage patterns lead to resource contention and load imbalance.
- Integration of BPIO, a method to resolve contention, with ADIOS provides a way to balance resource usage in a transparent way.

## Introduction
Balanced Placement I/O (BPIO) Library
- Topology-aware and balanced data placement [1]
- Resolves application level I/O contention
- Large-scale or wide-striped I/O improves substantially the application performance
- Computes placement cost for each I/O client

\[ \text{Placement Cost} = w_1 R_1 + w_2 R_2 + w_3 R_3 \]

Adaptable I/O System (ADIOS)
- ADIOS [2] provides portable, fast, scalable, easy-to-use, metadata rich output
- Change I/O method on-the-fly

Overall ADIOS/BPIO Architecture
- Scientific Applications
- Interface layer
- Kernel layer
- Transfer methods on top of BPIO
- ADIOS API
- Internal API
- External APIs
- Metadata

## Methodology
- IOR benchmark
  - Default
  - With BPIO
  - With ADIOS
  - With ADIOS/BPIO

- Large-scale application

## Metrics of Interest
- IOR synthetic benchmark
  - Measures parallel file system I/O performance
  - POSIX and MPI-IO API available
- Large-scale application (e.g. XGC1 or S3D)

## Results – Performance
- IOR with BPIO
  - POSIX: improved by up to 47%
  - MPI-IO: improved by up to 20%

- IOR with ADIOS/BPIO
  - POSIX is improved by up to 18%
  - MPI-IO (MPI_AMR) needs some more investigation

## Conclusions
- Performance improvements for IOR with BPIO
- Transparent integration of BPIO in ADIOS
- Ongoing work:
  - XGC1 ADIOS/BPIO performance evaluation

## References

## Acknowledgement
This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.