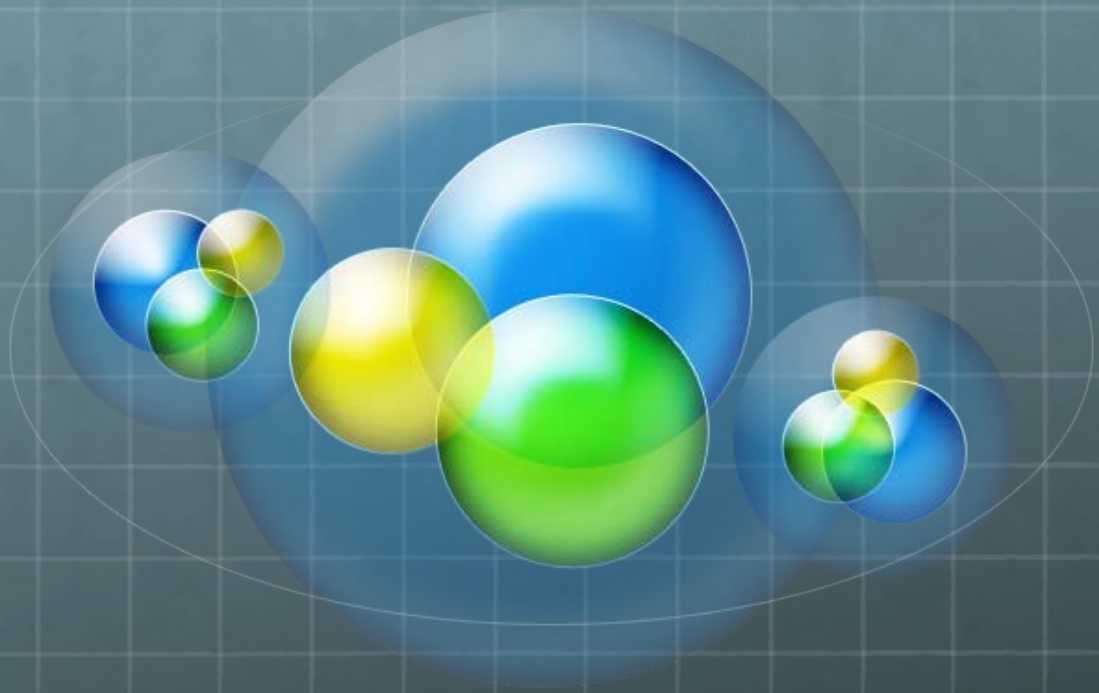


PIO Update

John Dennis

June 29, 2010

Parallel I/O library (PIO)

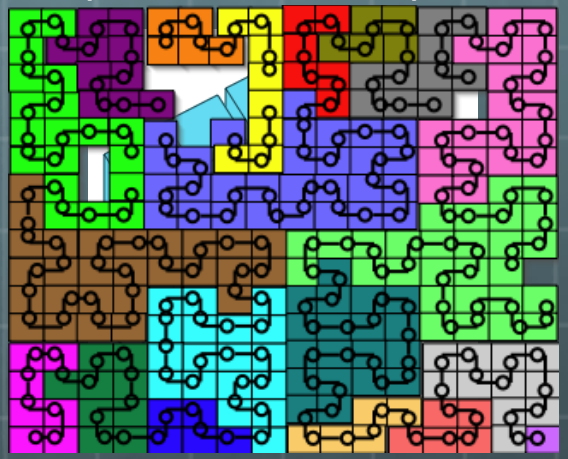
- Goals:
  - Reduce memory usage
  - Improve performance
- Principle Developers:
  - Loy (ANL)
  - Edwards (IBM)
  - Dennis (NCAR)
- Contributions from many in SEWG
- Writing a single file from parallel application
  - Flexibility in I/O libraries
  - MPI-IO,NetCDF3, NetCDF4, pNetCDF

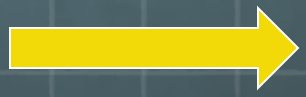
PIO Status

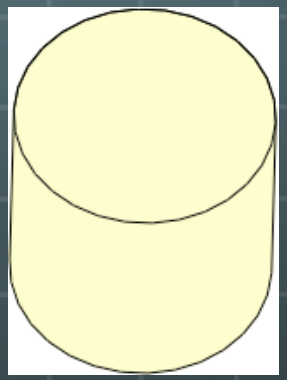
- Supported parallel I/O library in CCSM4 & CESM1 release
- Addition of Flow-control algorithms (Worley)
- Initial documentation using Doxygen
- Small but growing user base
  - ESMF
  - VAPOR + wavelet compression
- Performance optimization on Blue Gene
  - Improved robustness

PIO: Writing distributed data (I)

Computational decomposition





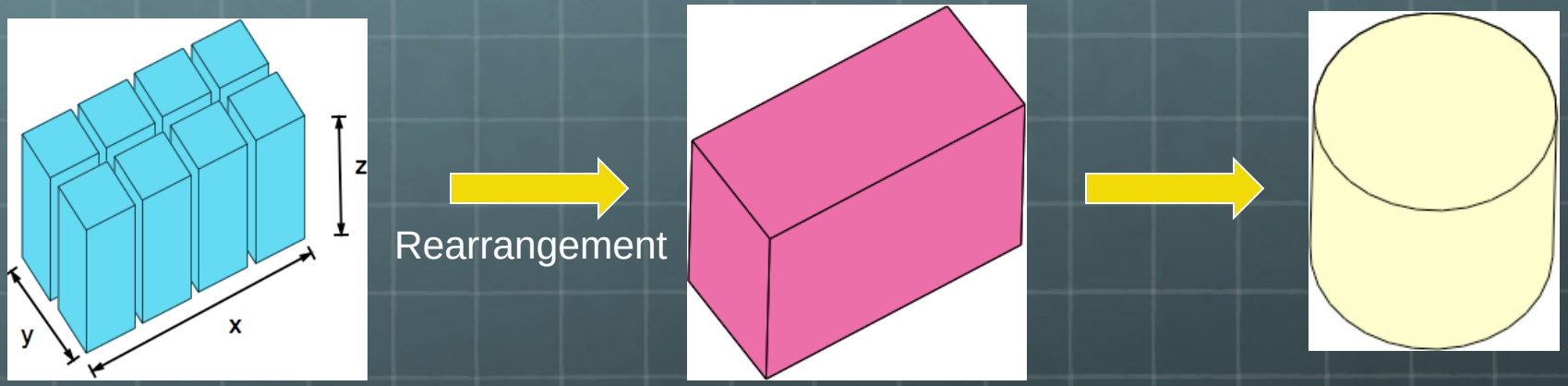


+ Simple
+ Most versions of MPI-IO will do aggregation
- Computational decomposition may not be optimal for disk access
- pNetCDF requires block cyclic decompositions

PIO: Writing distributed data (II)

Computational decomposition

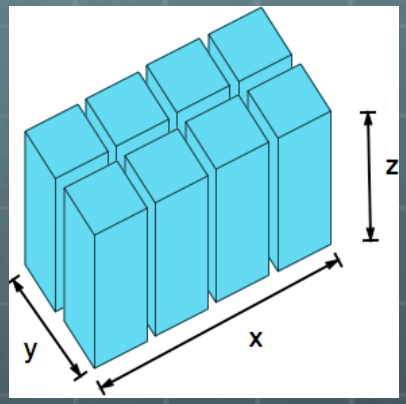
I/O decomposition



+ Maximize size of individual io-op's to disk
- Non-scalable user space buffering
- Very large fan-in → large MPI buffer allocations

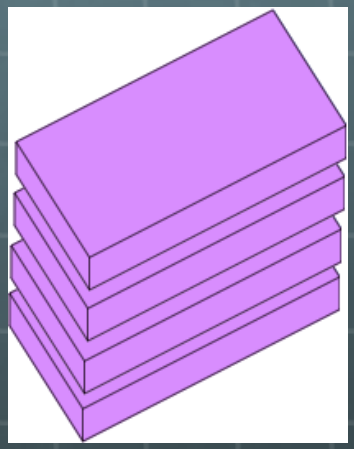
PIO: Writing distributed data (III)

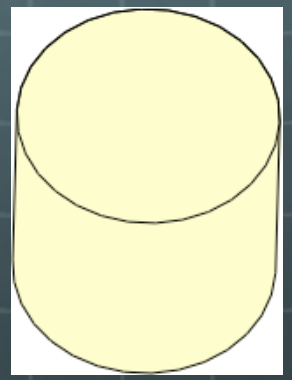
Computational decomposition



Rearrangement

I/O decomposition

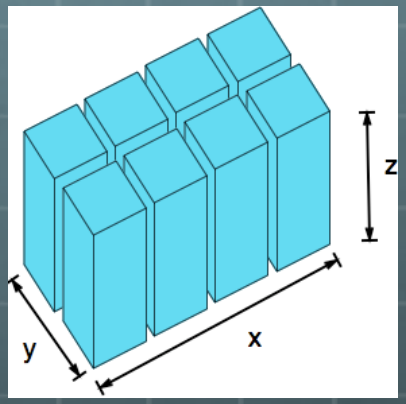




+ Scalable user space memory
+ Relatively large individual io-op's to disk
- Very large fan-in → large MPI buffer allocations

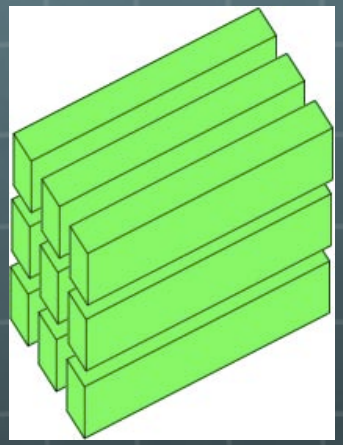
PIO: Writing distributed data (IV)

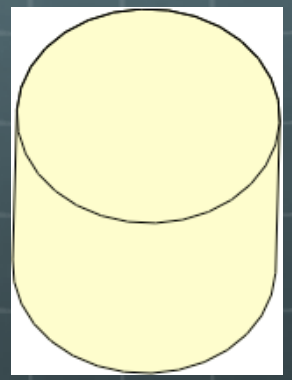
Computational decomposition



Rearrangement

I/O decomposition

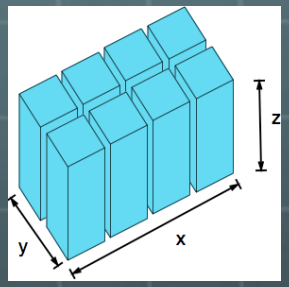




+ Scalable user space memory
+ Smaller fan-in -> modest MPI buffer allocations
- Smaller individual io-op's to disk

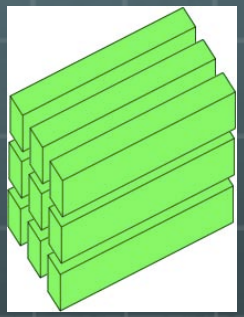
Writing data to Lustre file system

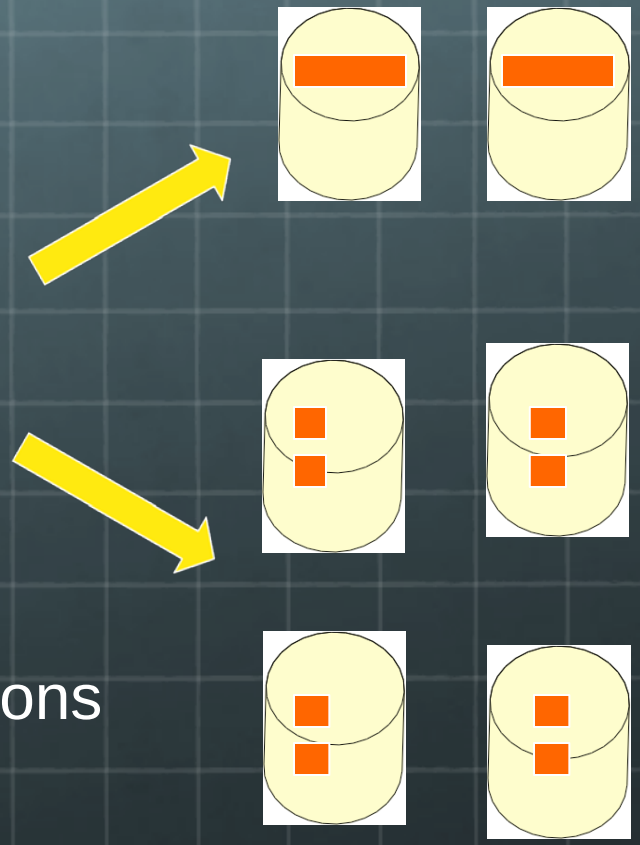
Computational decomposition



Rearrangement

I/O decomposition

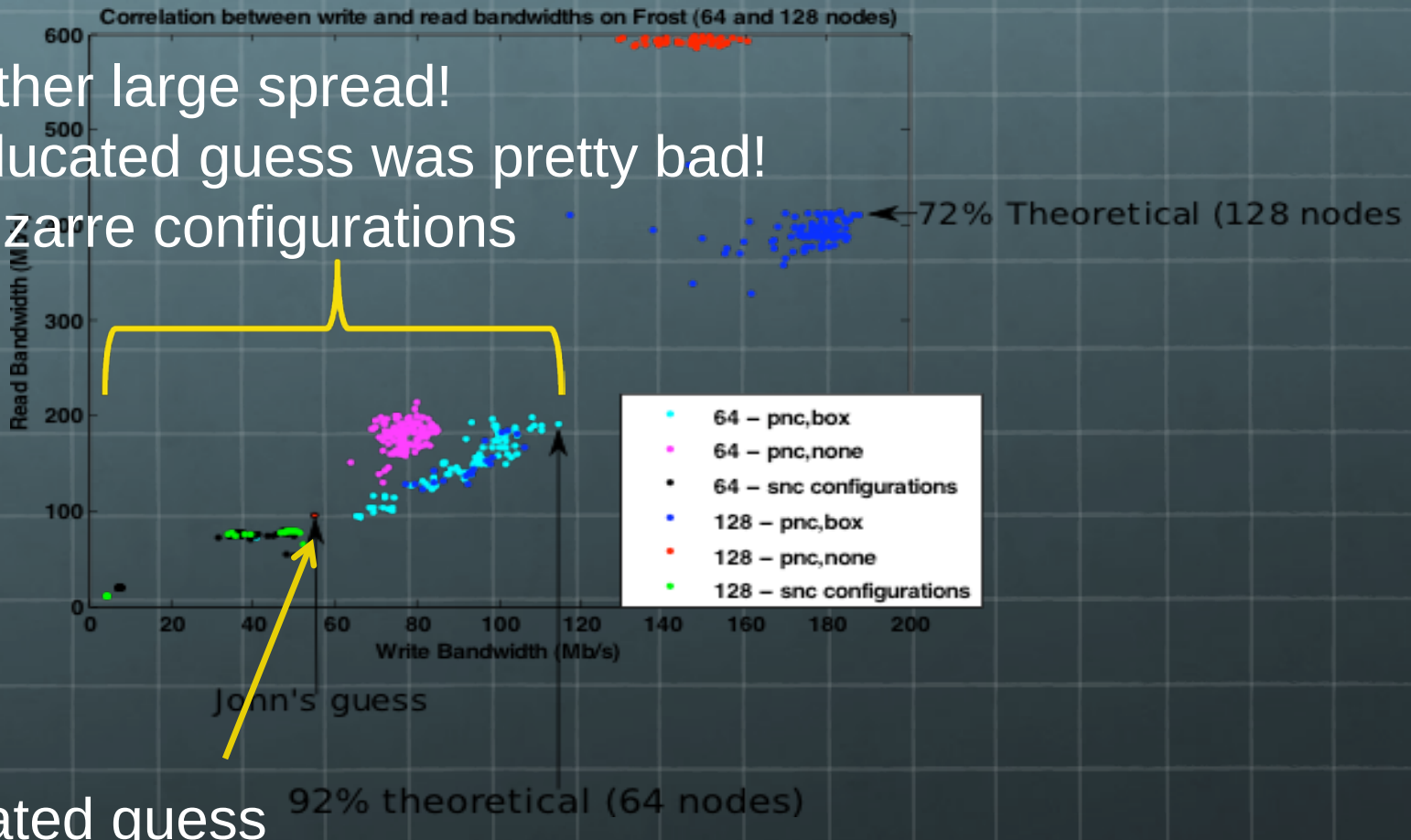




> 260,000,000,000 diff configurations

SIParCS:
Searching multi-dimension search space

(I) Rather large spread!
(II) Educated guess was pretty bad!
(III) Bizarre configurations

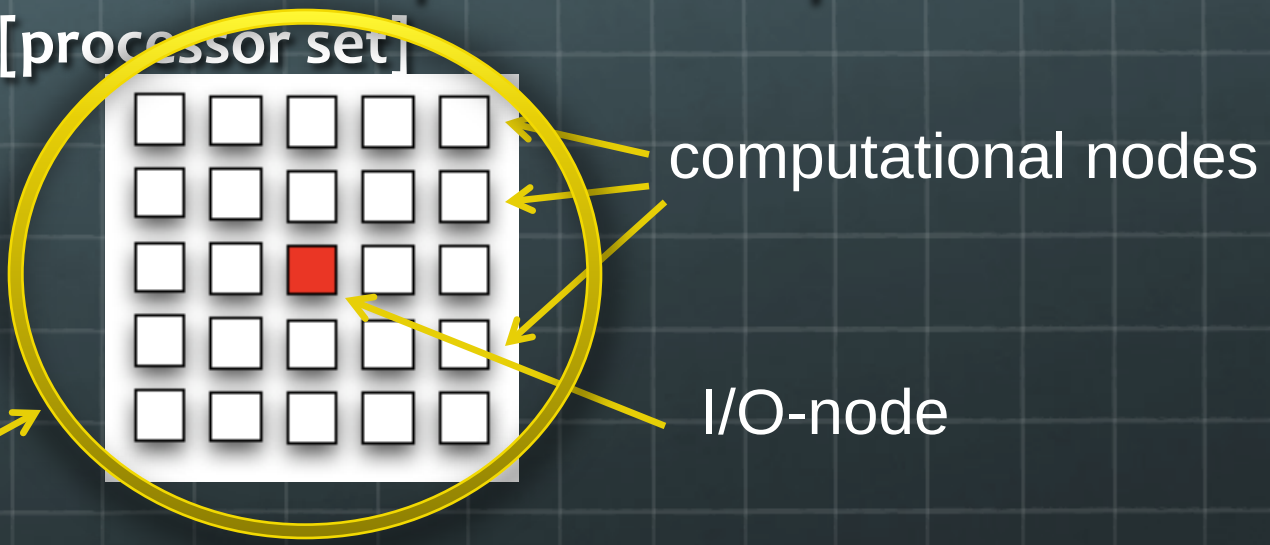


Educated guess

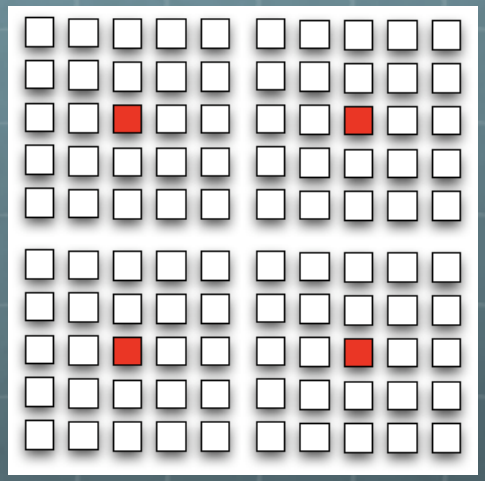
K. Ericson

Good configurations on Blue Gene ?

- Load imbalance across I/O nodes
- PIO was not using Blue Gene specific topological information
  - One or more IO-node per set of computational nodes [processor set]

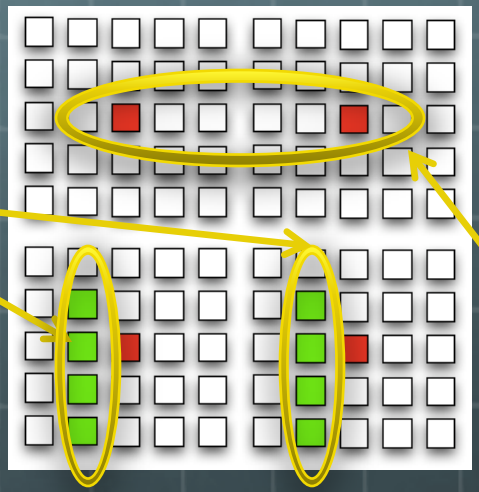


Jobs contain
one or more Psets

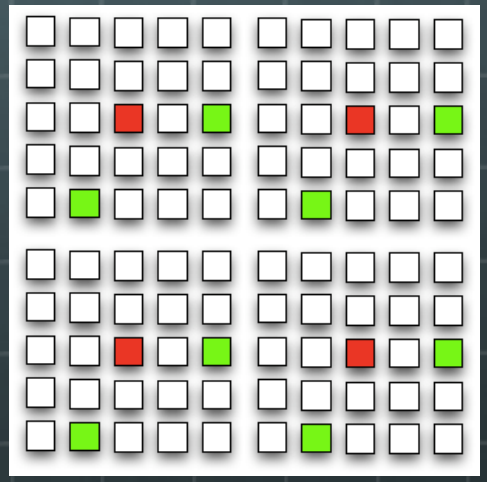


MPI-IO tasks

Unbalanced allocation of
I/O tasks to I/O-nodes



Idle
I/O nodes

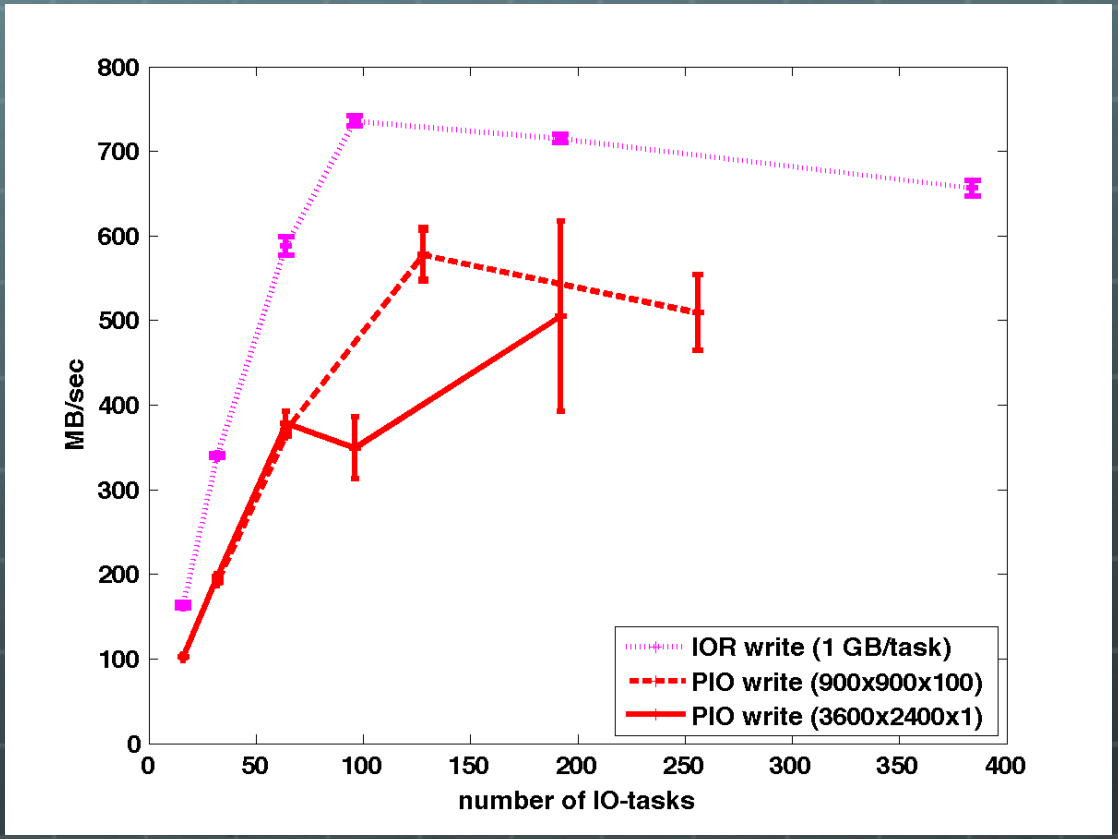


Balanced allocation of
I/O tasks to I/O nodes

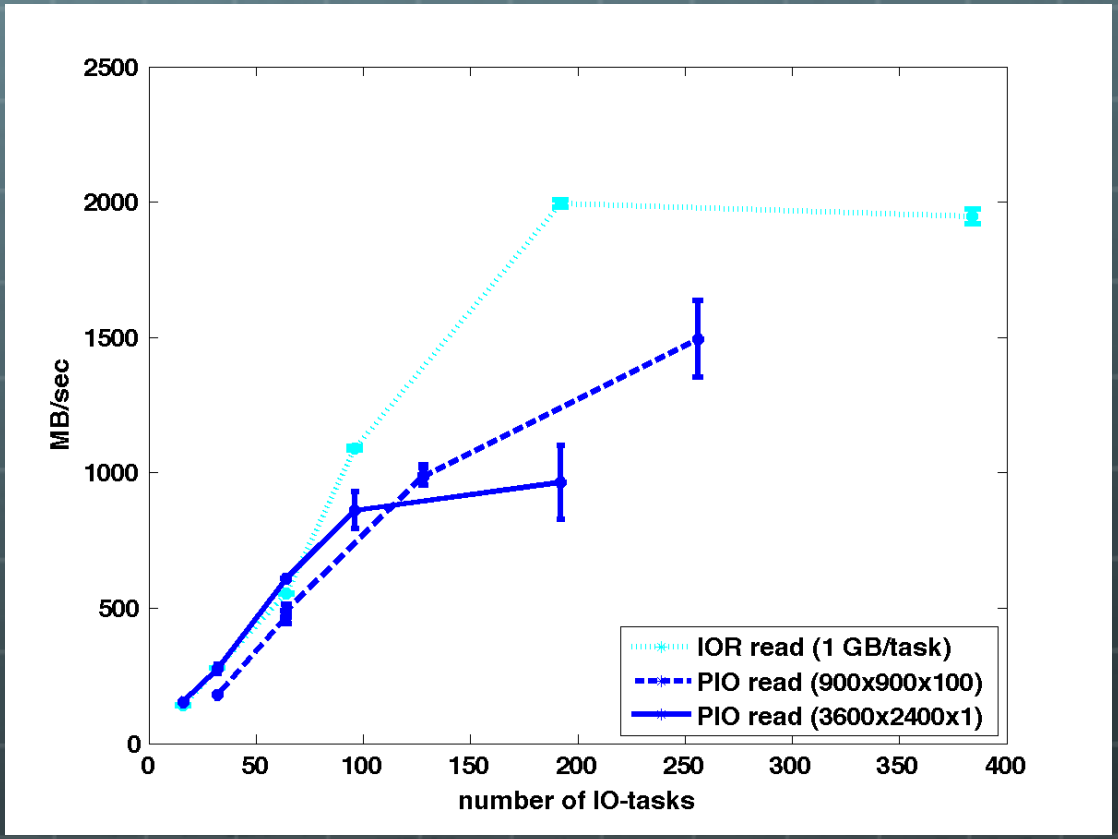
Optimizing for Blue Gene

- Conceptual bug in PIO
- Modified library to balance I/O tasks to I/O nodes
- PIO configuration
  - [3600x2400x1] x 8 bytes x 10 variables x 10 files
  - [900x900x100] x 8 bytes x 10 variables x 10 files
- IOR configuration: 1GB/io-task

Write performance on BG/L



Read performance on BG/L



Questions

dennis@ucar.edu