Understanding and Optimizing Data Input/Output of Large-Scale Scientific Applications

Presented by

Jeffrey S. Vetter (Leader)

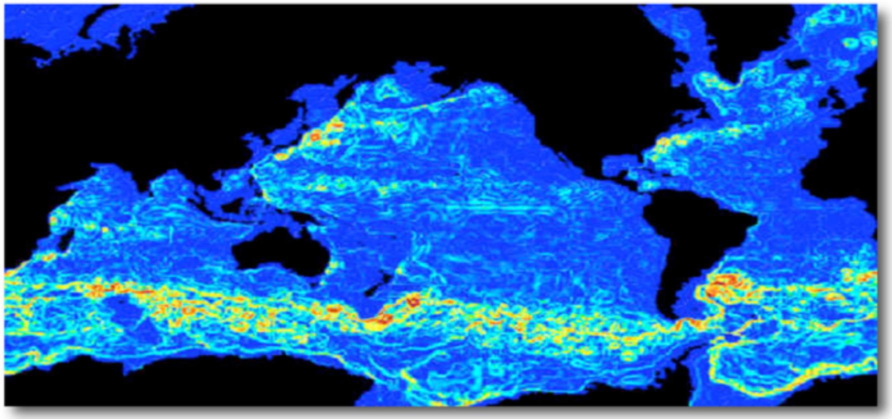
Weikuan Yu, Philip C. Roth

Future Technologies Group

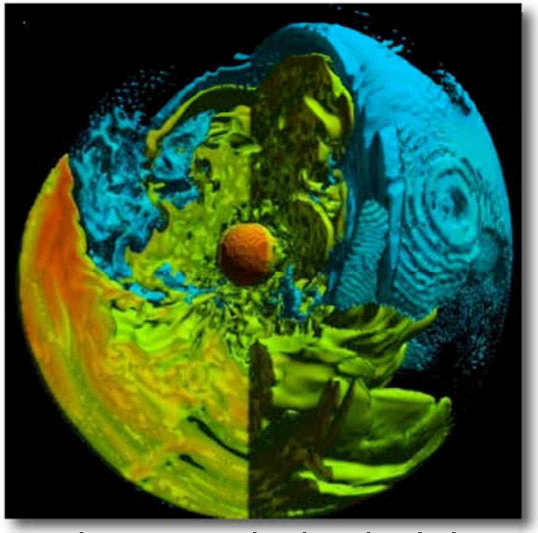
Computer Science and Mathematics Division

I/O for large-scale scientific computing

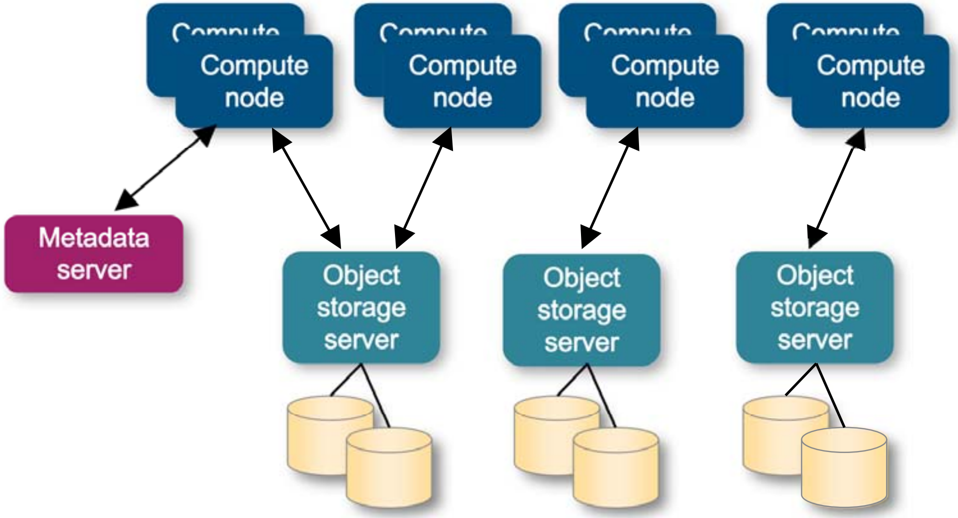
- Reading input and restart files
- Writing checkpoint files
- Writing movie, history files
- Gaps of understanding across domains; efficiency is low



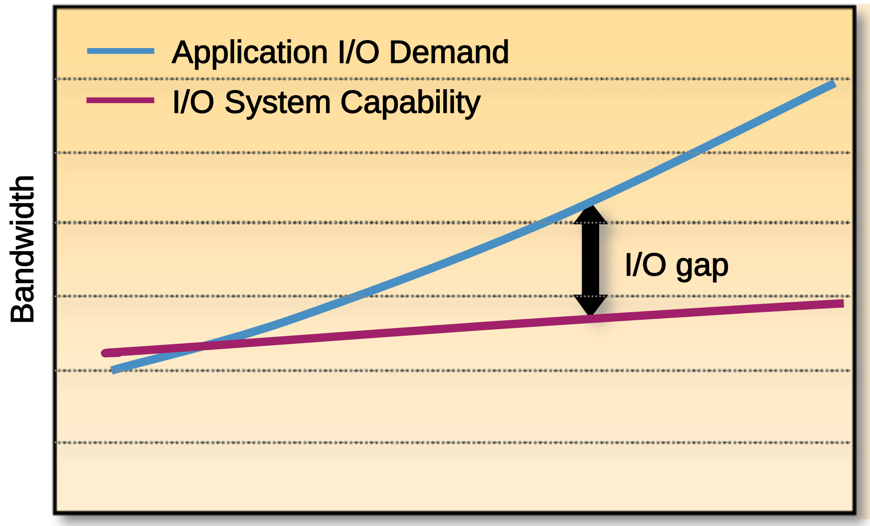
SciDAC climate studies visualization at ORNL



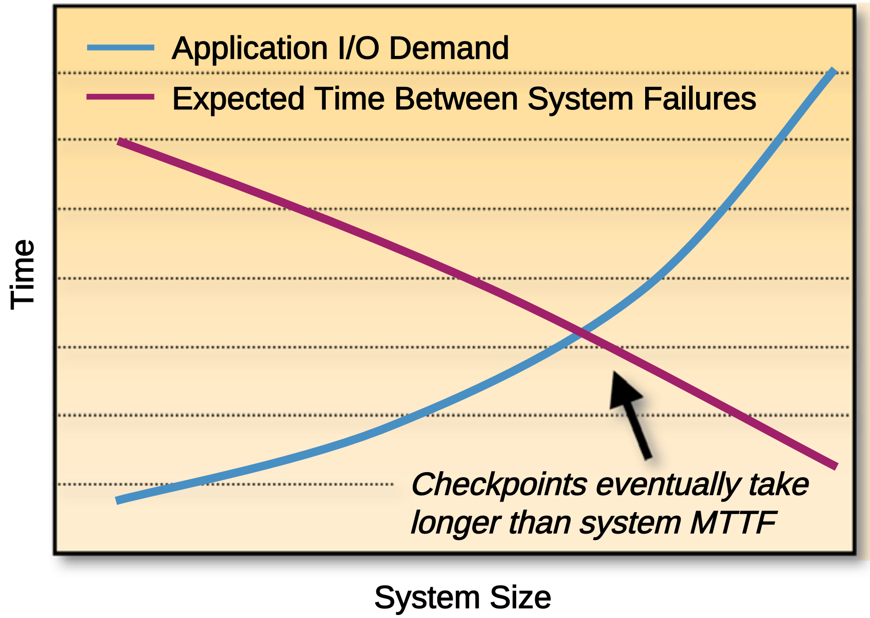
SciDAC astrophysics simulation visualization at ORNL



The I/O gap



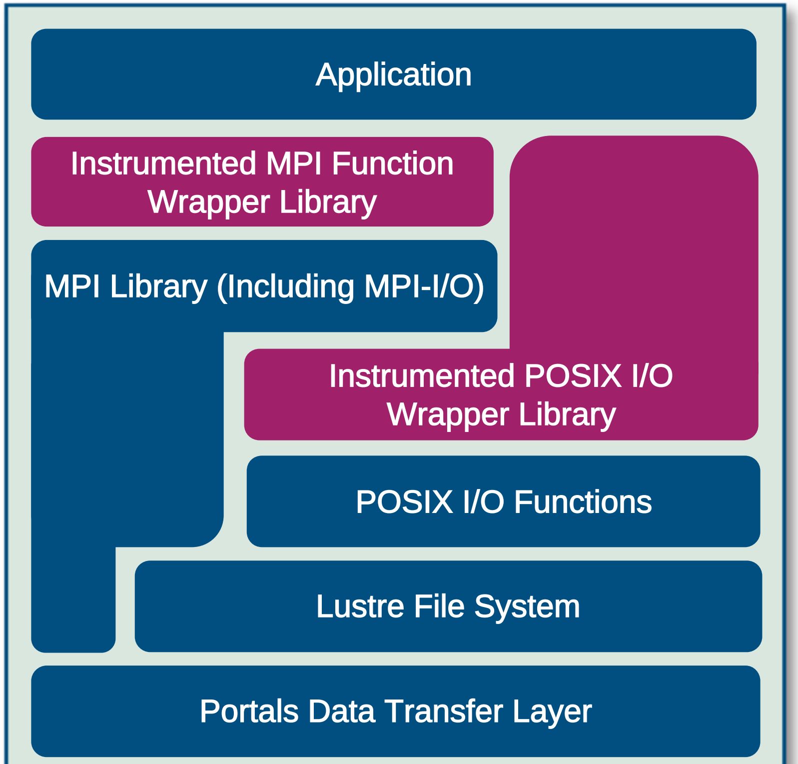
Widening gap between application I/O demands and system I/O capability.



Gap may grow too large for existing techniques (e.g.,checkpointing) to be viable because of decreases in system reliability as systems get larger.

Insight into I/O behavior

- Performance data collection infrastructure for Cray XT
- Gathers detailed I/O request data without changes to application source code
- Useful for
  - Characterizing application I/O
  - Driving storage system simulations
  - Deciding how and where to optimize I/O



Instrumented Cray XT MPI Software Stack

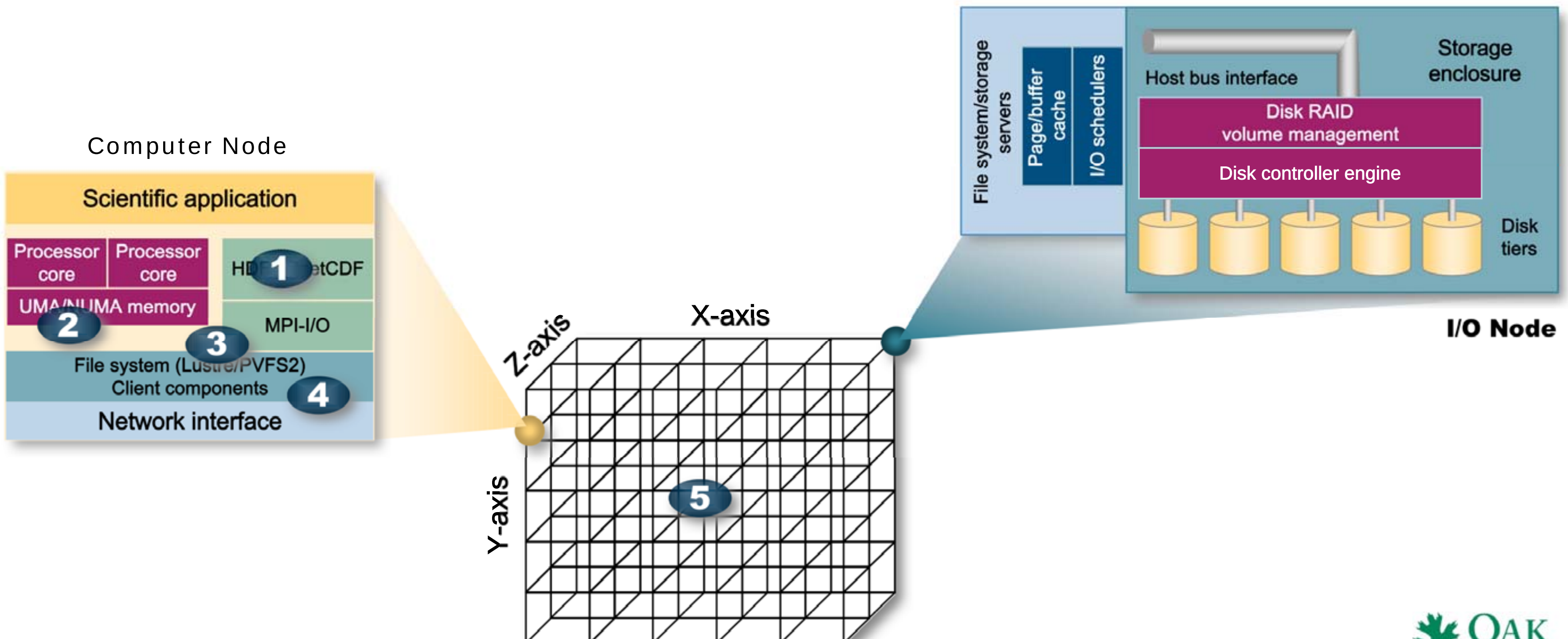
Jaguar Cray XT system at ORNL

Oak Ridge National Laboratory

Optimization through parallel I/O libraries

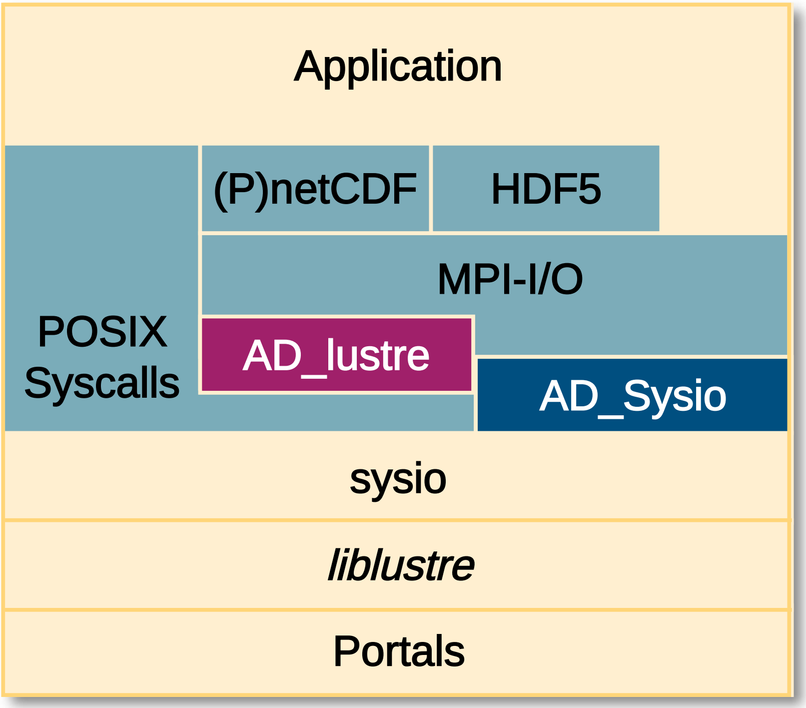
- Advantages from parallel I/O libraries
  - Interfacing application, runtime, and operating system
  - Ease of solution deployment

- Challenges approachable via libraries:
  1. Application hints and data manipulation
  2. Processor/memory architecture
  3. Parallel I/O protocol processing overhead
  4. File-system–specific techniques
  5. Network topology and status



Opportunistic and adaptive MPI-I/O for Lustre (OPAL)

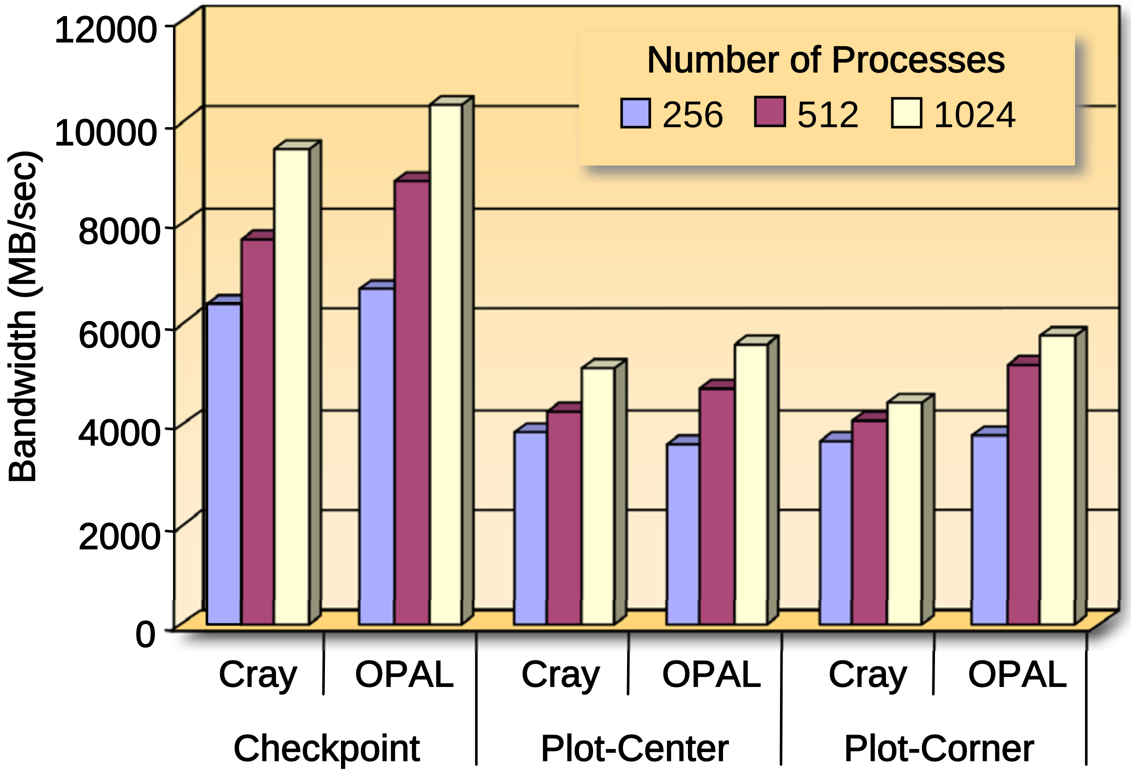
- An MPI-I/O package optimized for Lustre
- A unified code base for Cray XT and Linux
- Open source, better performance
  - Improved data-sieving implementation
  - Enabled arbitrary striping specification over Cray XT
  - Lustre stripe-aligned file domain partitioning



- http://ft.ornl.gov/projects/io/#download
- Provides better bandwidth and scalability than default Cray XT MPI-I/O library in many cases

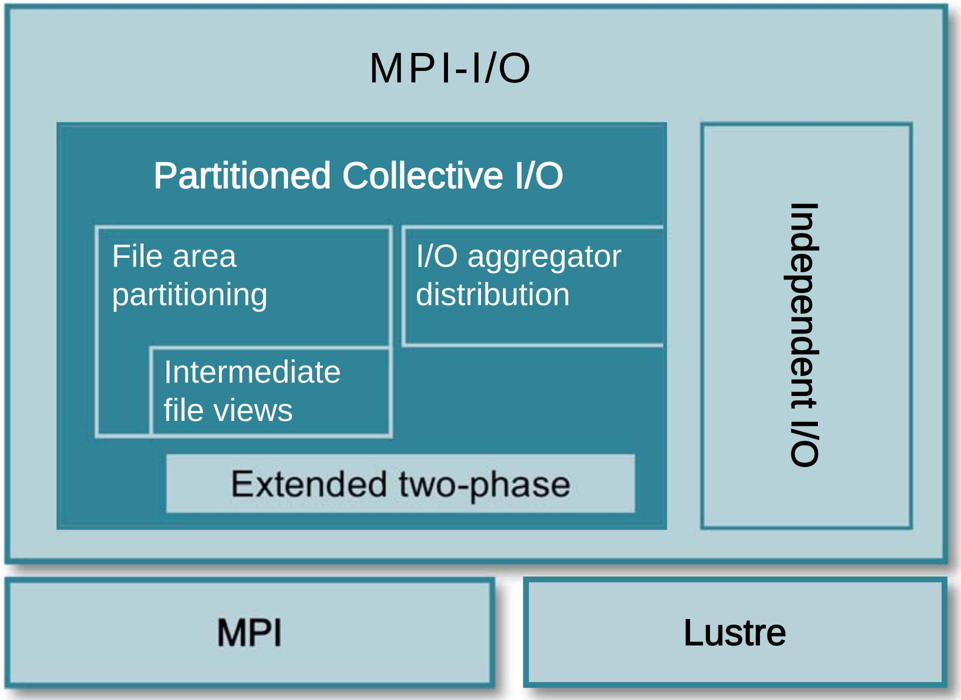
Example results: OPAL

- Bandwidth for FLASH I/O benchmark.
- OPAL provided better bandwidth and scalability than default Cray XT MPI-I/O library.



Partitioned collective I/O (ParColl)

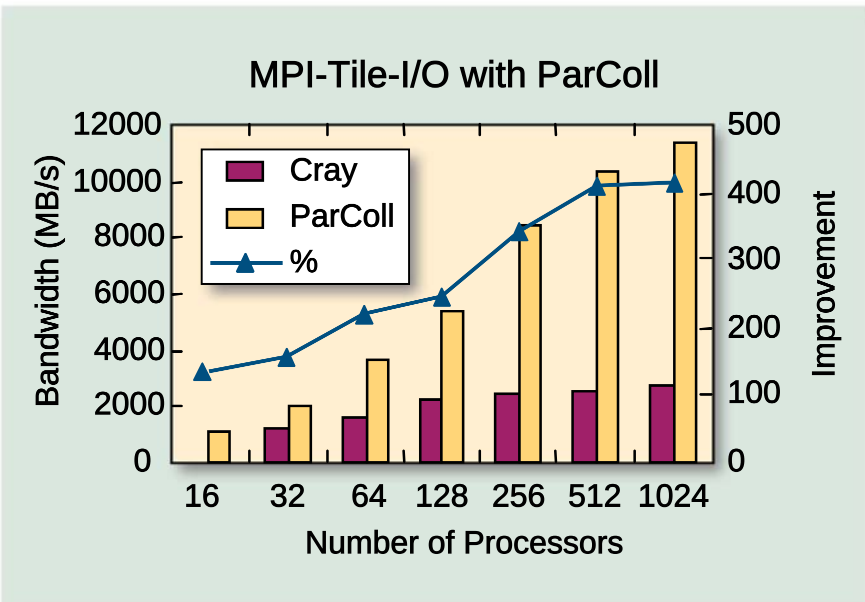
- Collective I/O is good for small I/O request aggregation.
- But global synchronization within is a barrier to scalability.
- ParColl partitions global processes, I/O aggregators, and the shared global file appropriately for scalable I/O aggregation.

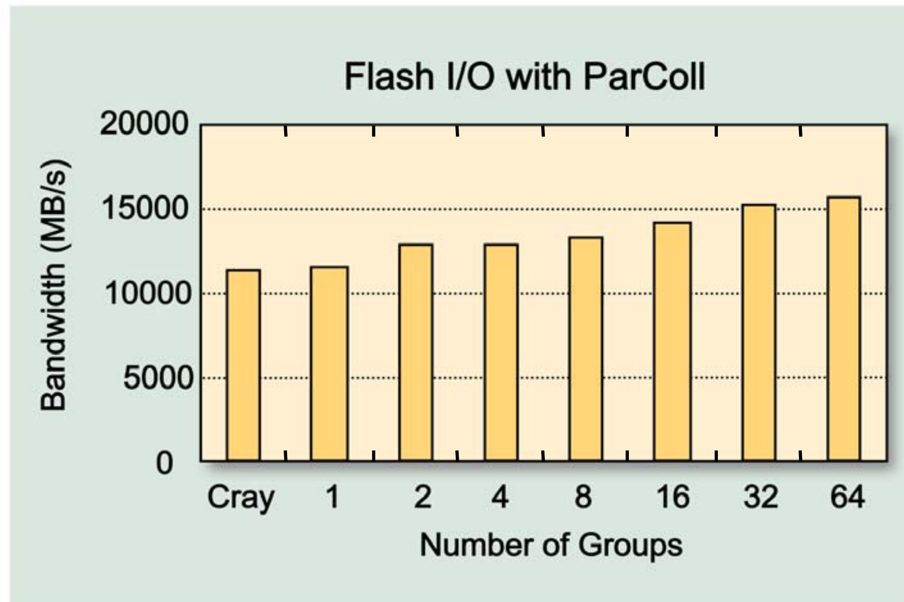


Oak Ridge National Laboratory

Example results: ParColl

- Evaluated benchmarks: MPI-Tile-I/O and Flash I/O.
- ParColl improves collective I/O for various benchmarks.





Contacts

Jeffrey S. Vetter

Leader
Future Technologies Group
Computer Science and Mathematics Division
(865) 356-1649
vetter@ornl.gov

Weikuan Yu

(865) 574-7990
wyu@ornl.gov

Philip C. Roth

(865) 241-1543
rothpc@ornl.gov

For more information, including code downloads, see http://ft.ornl.gov/projects/io/