Performance Evaluation of Application I/O Kernels Using MPI Collective I/O and Caching

Wei-keng Liao and Alok Choudhary
Northwestern University

In Collaboration with
Rob Ross, Rob Latham, and Rajeev Thakur
Argonne National Laboratories

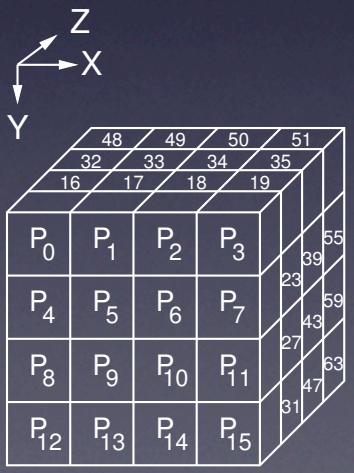
Jacqueline Chen
Sandia National Laboratories

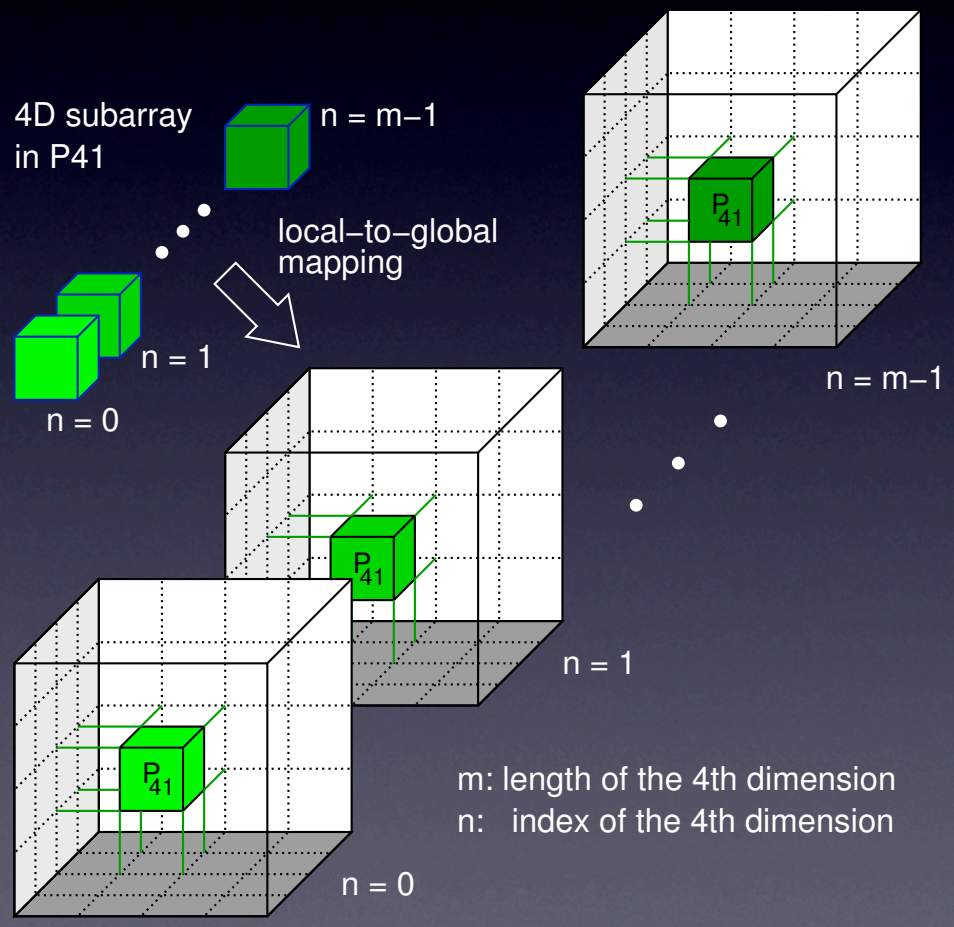
Ramanan Sankaran and Scott Klasky
Oak Ridge National Laboratory

SDM All-hand Meeting, Nov. 2007

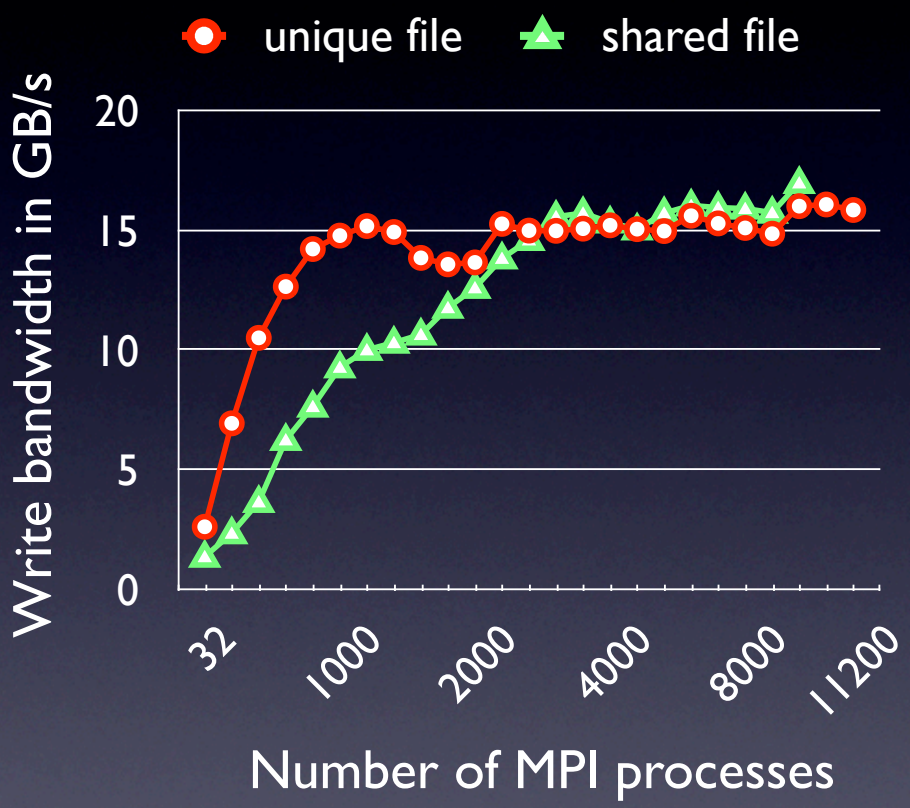
S3D I/O Kernel

- S3D is a turbulent combustion application developed at SNL
- Production runs usually require > 10,000 cores
- Checkpoint writes





Results on Cray XT @ ORNL



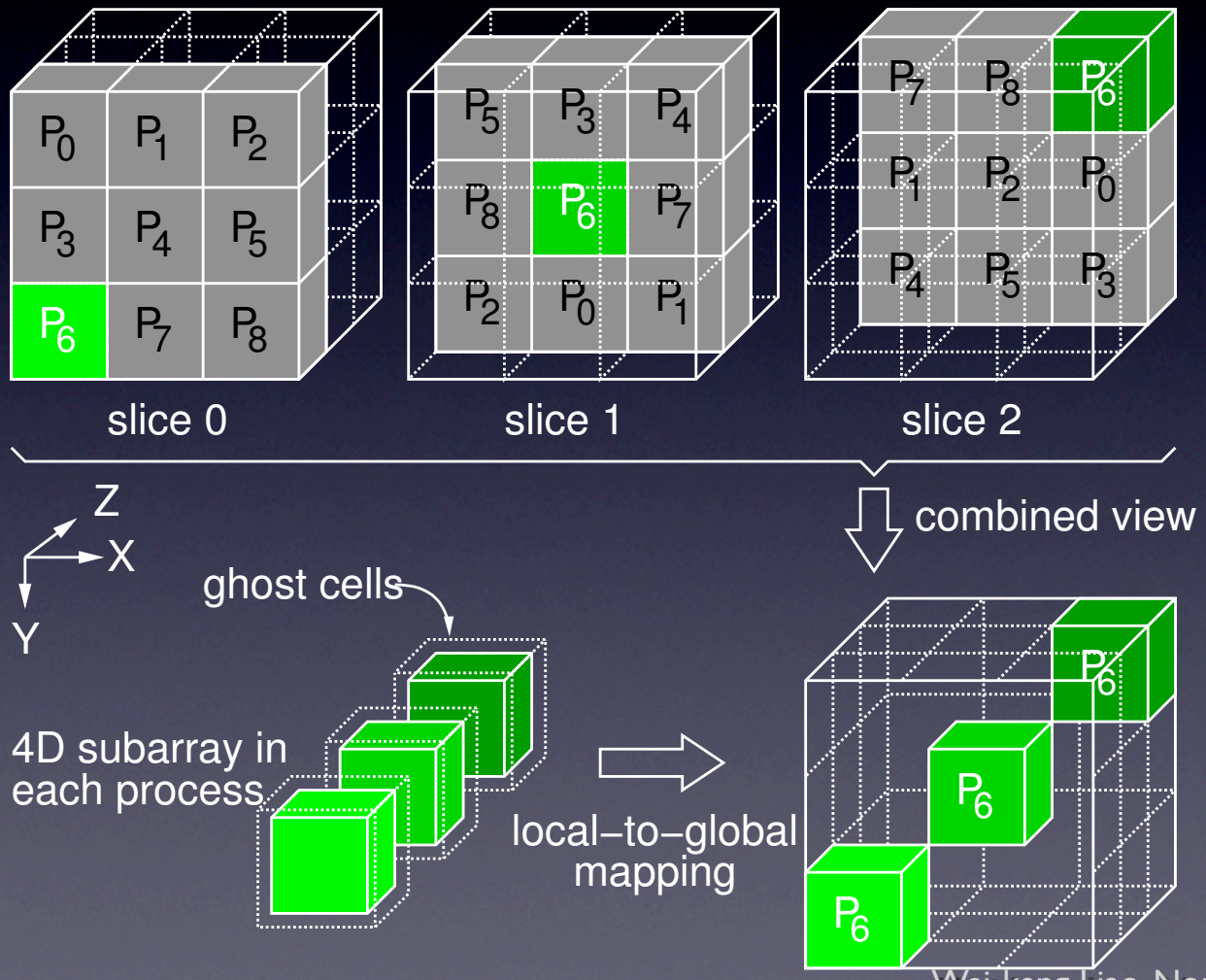
- Original design
  - One file per process per checkpoint
- Using MPI-IO
  - One file per checkpoint
- Experiments
  - 10 checkpoints
  - Fixed subarray size: 50x50x50
  - Lustre file system

Client Process Collaboration

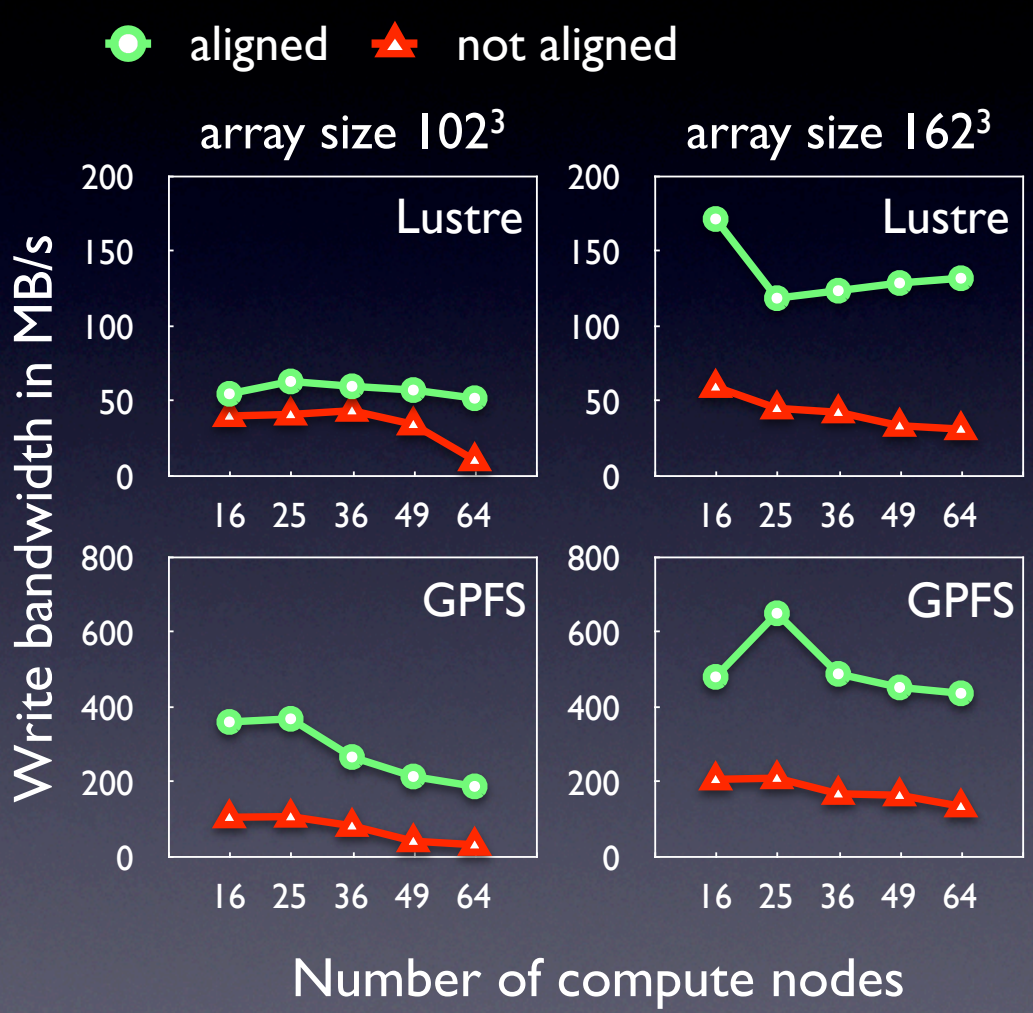
- MPI collective I/O proves collaboration scalable
  - 2-phase I/O: data is redistributed among processes to generate large contiguous I/O requests
- Other optimizations
  - I/O alignment with file system lock boundaries
  - File caching

BTIO benchmark

- 3D block tri-diagonal array partitioning pattern



I/O Aligned with Lock Boundaries

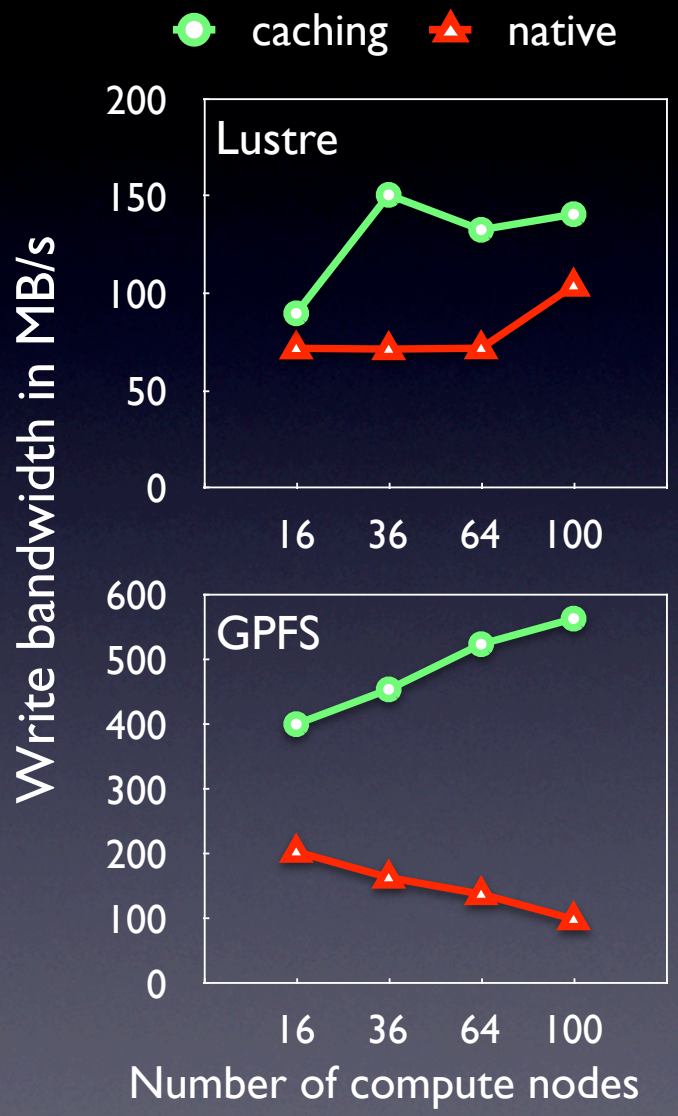


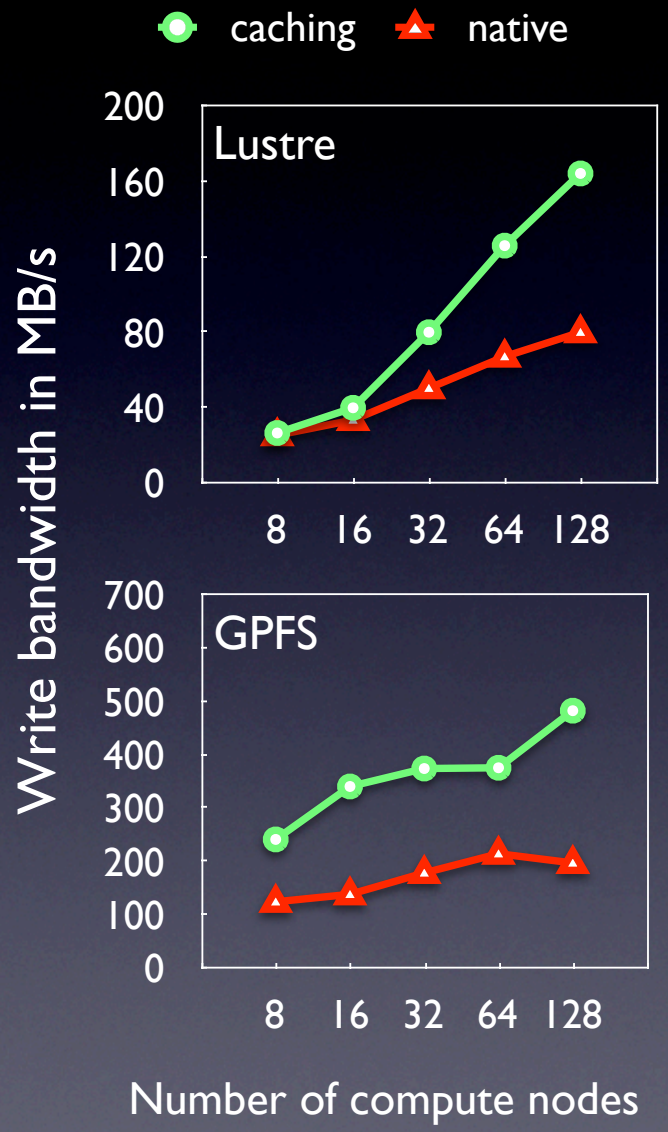
- MPI collective I/O
  - Divide aggregate access range among processes
  - Each process is responsible for the I/O in its file domain
- Comparison
  - Even division
  - Division aligned with lock boundaries

MPI-IO Client-side File Caching

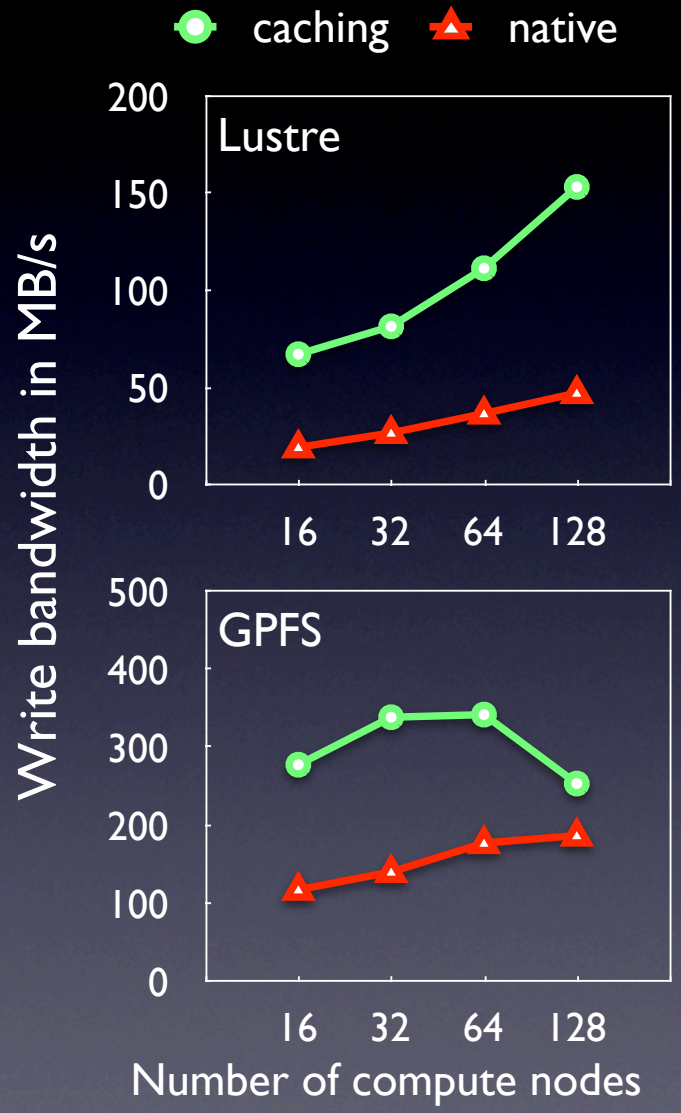
- A fully functional caching layer in MPI library
  - Clients handle cache coherence control
  - Handle read/write, collective/independent I/O
- Mechanisms
  - An I/O thread in each MPI process
  - Cache metadata management
  - Local/global caching policies

BTIO and S3D I/O





FLASH I/O



- I/O kernel of the FLASH astrophysics application developed at U. of Chicago
- I/O method: HDF5
- Each process writes 80 subarrays
  - 16x16x16 doubles
- Writes are not interleaved among processes

Summary

- MPI-IO for S3D application
  - One file per checkpoint
  - Data arrays are stored in canonical order
  - Performance is comparable to one-file-per-process approach
- A file caching layer for MPI-IO
  - Aggregate I/O for better performance
  - File accesses aligned with lock boundaries