I/O and post-processing for High-resolution climate data

Dr. John Dennis
dennis@ucar.edu

Motivation

- High-resolution climate generates a large amount of data!

Outline

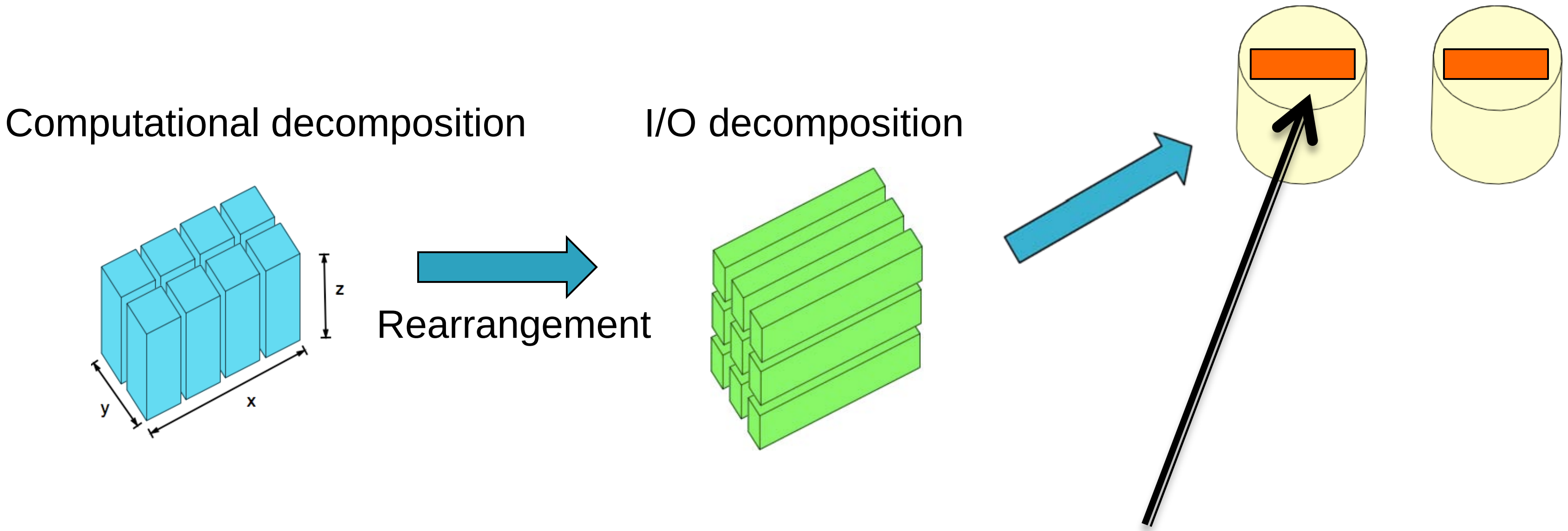
- PIO update and Lustre optimizations
- How do we analyze high-resolution climate data faster?
- Wavelet compression for climate data

PIO update and Lustre optimizations

Parallel I/O library (PIO)

- Goals:
  - Reduce memory usage
  - Improve performance
- Writing a single file from parallel application
  - Flexibility in I/O libraries
  - MPI-IO,NetCDF3, NetCDF4, pNetCDF

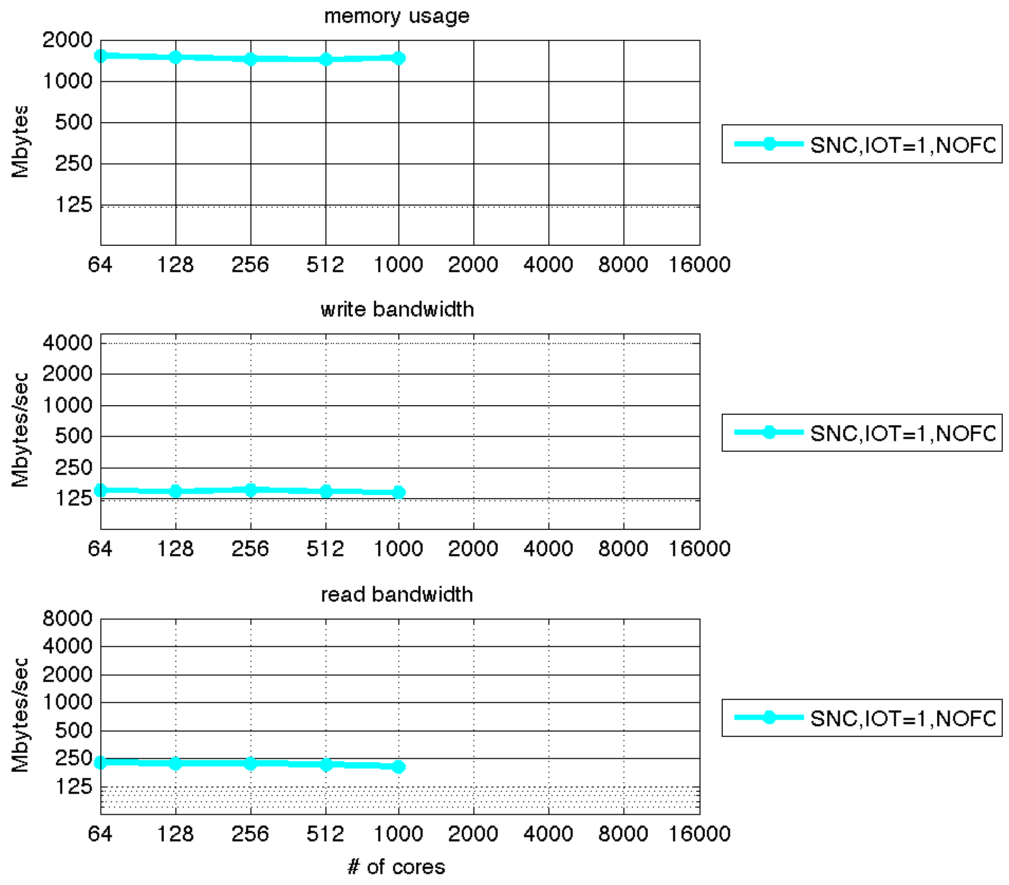
Optimizing writing data to Lustre file system

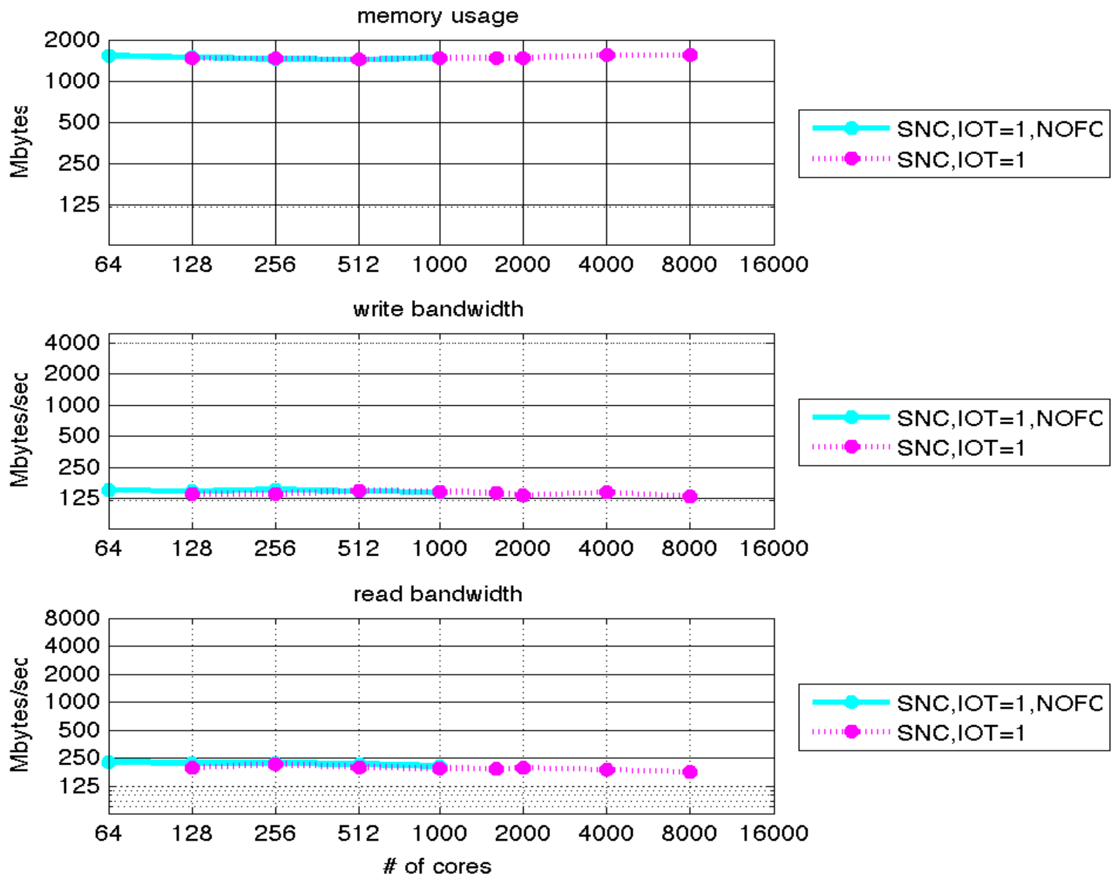


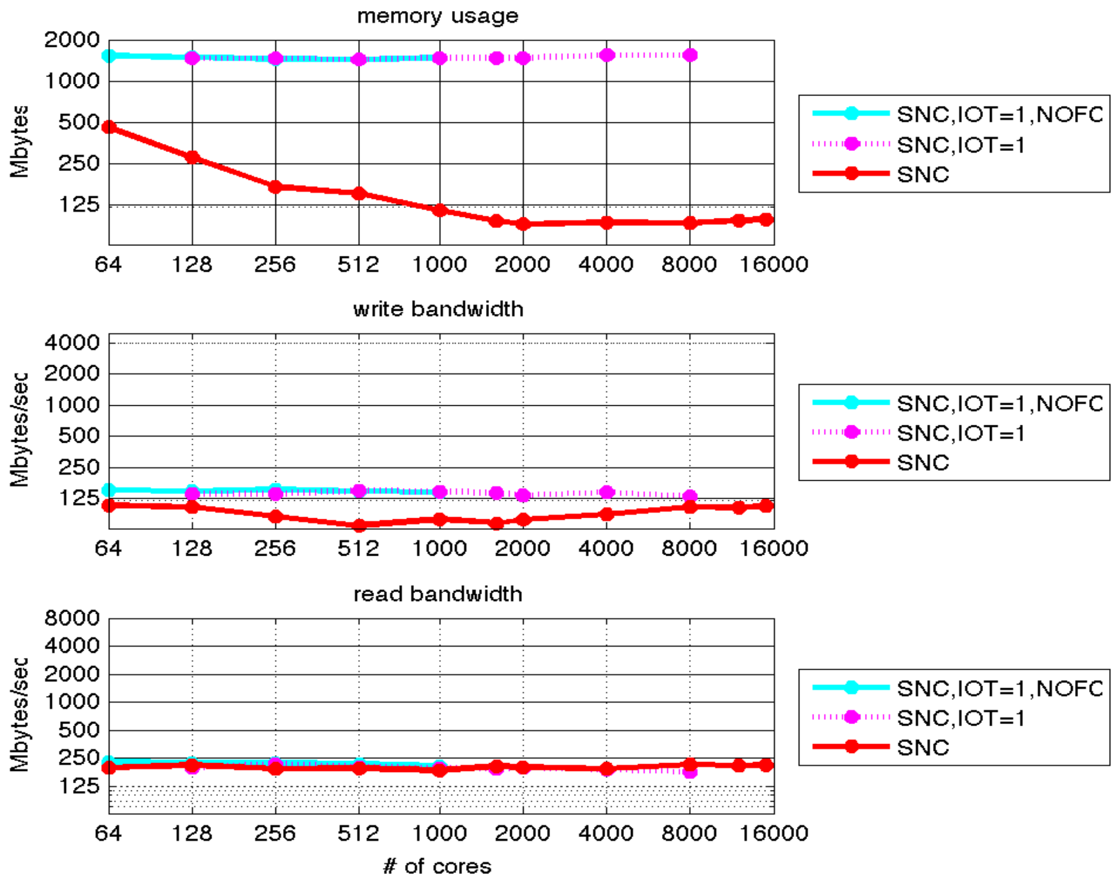
Match I/O decomposition to Lustre stripe size

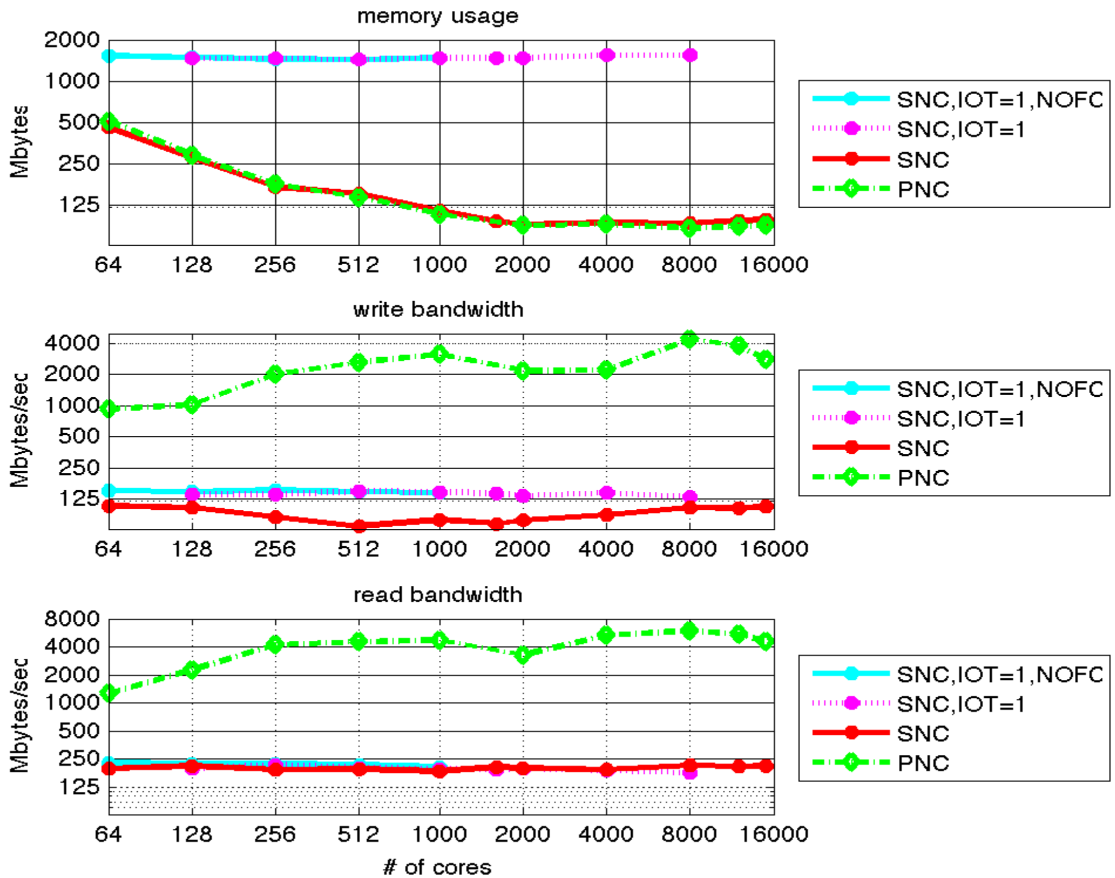
Experimental setup

- Read/write 3D POP sized variable [3600x2400x40]
- 10 files, 10 variables per file, [max bandwidth]
- Using Kraken (Cray XT5) + Lustre filesystem
  - Used 16 of 336 OST
- Impact of PIO features
  - Flow-control
  - Vary number of IO-tasks
  - Different general I/O backends
- Did we achieve our design goals?


memory usage
Mbytes
2000
1000
500
250
125
64
128
256
512
1000
2000
4000
8000
16000
SNC,IOT=1,NOFC
write bandwidth
Mbytes/sec
4000
2000
1000
500
250
125
SNC,IOT=1,NOFC
read bandwidth
Mbytes/sec
8000
4000
2000
1000
500
250
125
SNC,IOT=1,NOFC
# of cores


memory usage
Mbytes
2000
1000
500
250
125
64
128
256
512
1000
2000
4000
8000
16000
SNC,IOT=1,NOFC
SNC,IOT=1
write bandwidth
Mbytes/sec
4000
2000
1000
500
250
125
read bandwidth
8000
# of cores


memory usage
Mbytes
2000
1000
500
250
125
64
128
256
512
1000
2000
4000
8000
16000
SNC,IOT=1,NOFC
SNC,IOT=1
SNC
write bandwidth
Mbytes/sec
4000
2000
1000
500
250
125
read bandwidth
8000
4000
2000
1000
500
250
125
# of cores


memory usage
Mbytes
2000
1000
500
250
125
64
128
256
512
1000
2000
4000
8000
16000
SNC,IOT=1,NOFC
SNC,IOT=1
SNC
PNC
write bandwidth
Mbytes/sec
4000
2000
1000
500
250
125
read bandwidth
8000
4000
# of cores

How do we analyze high-resolution climate data faster?

John Dennis, Matthew Woitaszek (NCAR)
Taleena Sines* (Frostburg State)

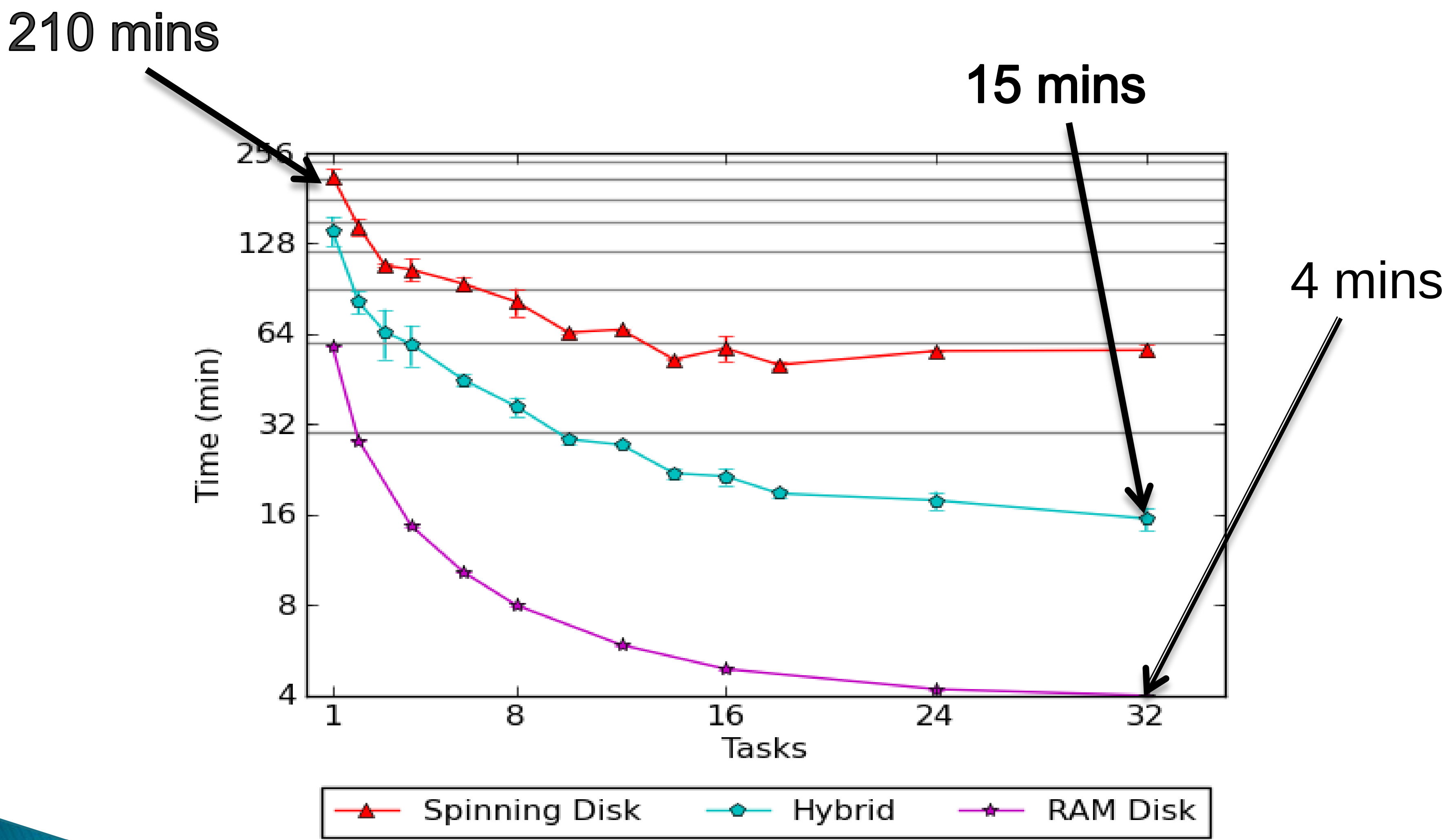
“Parallel high-resolution climate data analysis using Swift”, Linux Clusters [under review]

* SIParCS Intern

Parallelizing diagnostics

- Used Swift a workflow language (UC/ANL)
  - Parallel scripting language
  - Data dependency driven
- Examine performance on several data-intensive architectures
  - Flash memory: Dash (SDSC)
  - SGI UV: Nautilus (NICS)
  - Large memory node: Polynya (NCAR)
    - 32 cores
    - 1 TB ram [512 GB memory/ 512 GB ramdisk]
    - 120 TB GPFS file-system (old hardware)

Polynya: 0.5° /10yr



Wavelet compression for Climate data

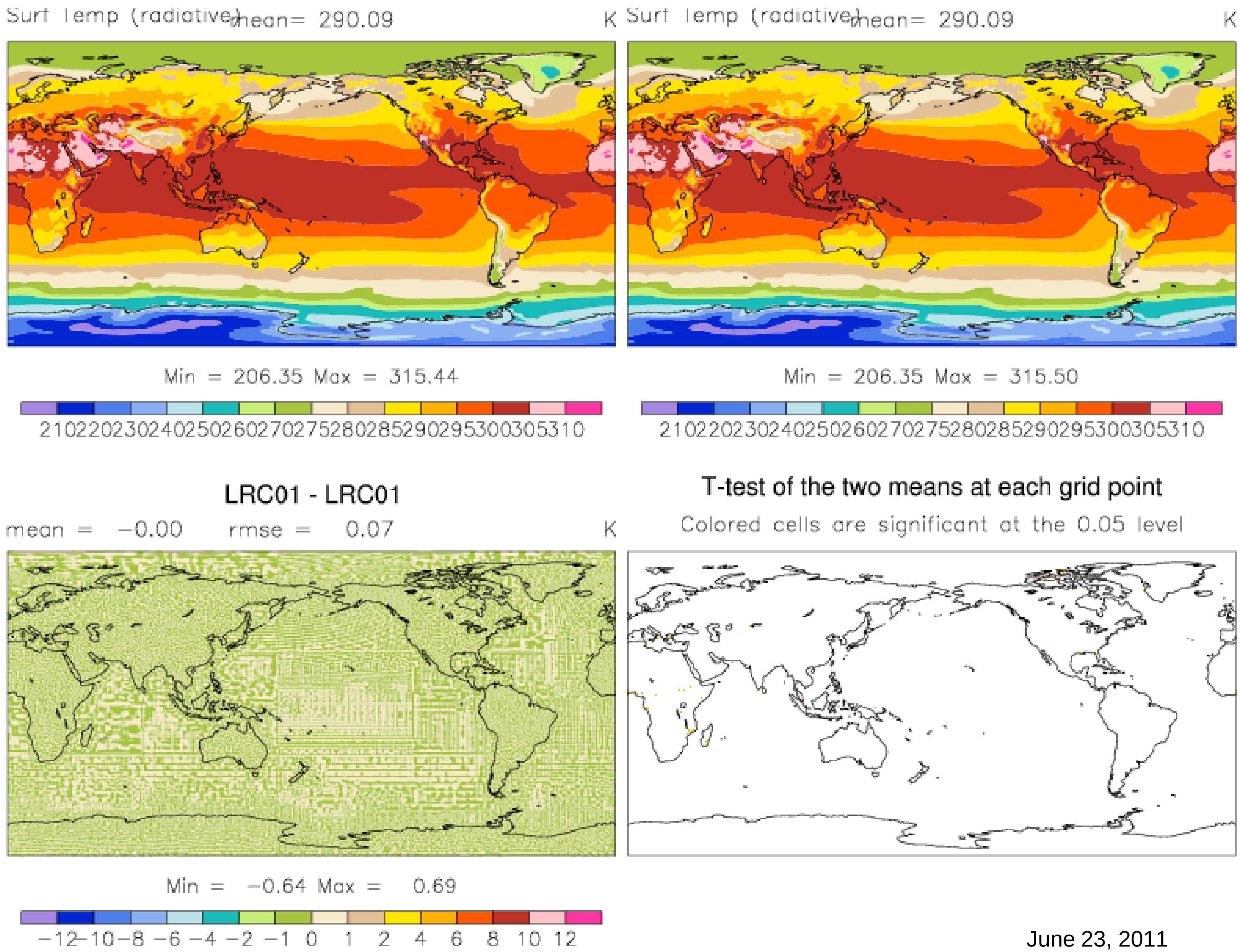
John Clyne, Yanick Polius, John Dennis
(NCAR)

Wavelet compression of Climate data

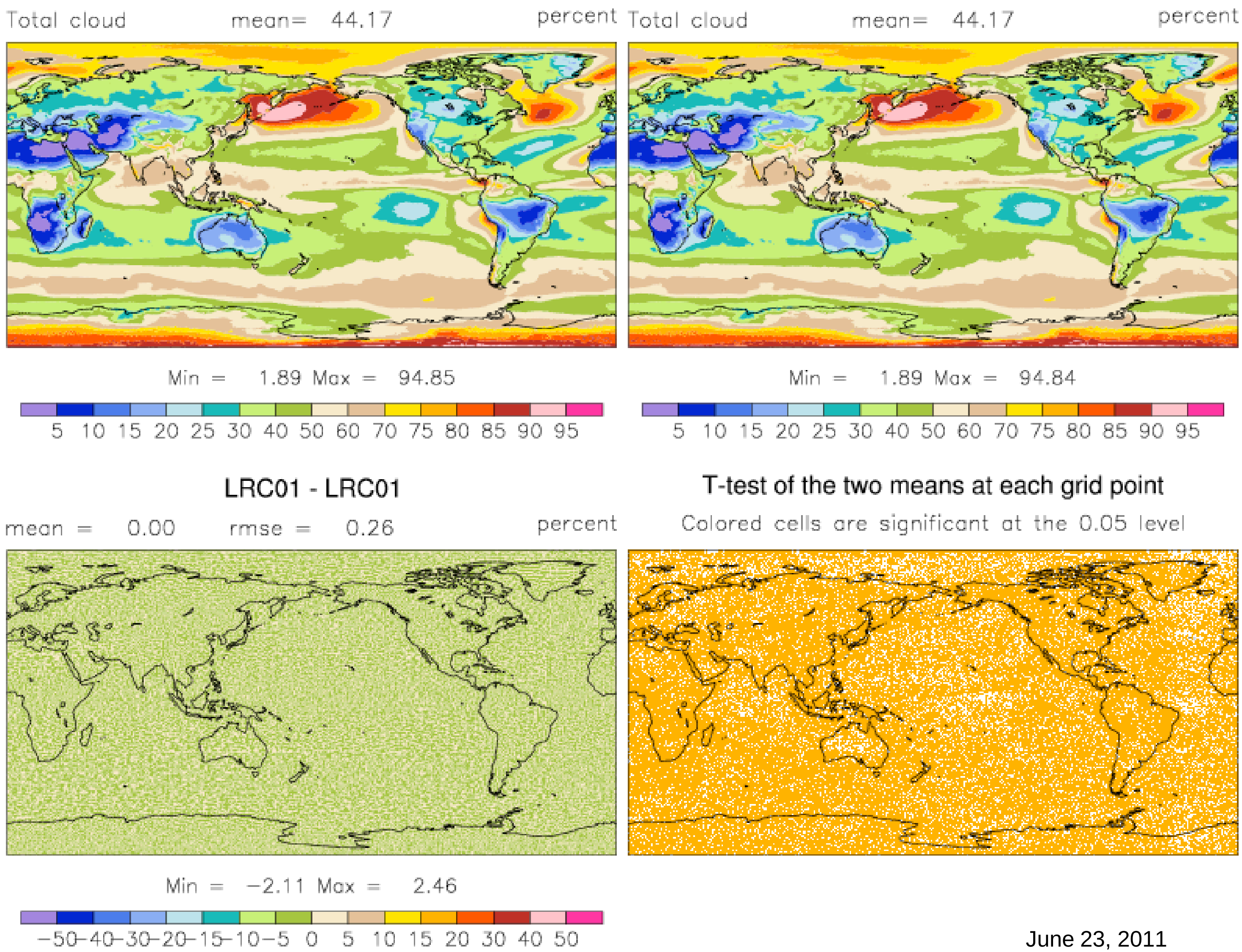
- Compression algorithm:
  - Apply wavelet transform to model outputs
  - Sort resulting wavelet coefficients by absolute magnitude
  - Discard smallest coefficients
  - Reconstruct an approximation of original data from remaining coefficients
- Compare original and reconstructed data using CESM AMWG Diagnostic Package

- Preliminary experiment
- 10 years of 0.5 deg CAM
- Compression
  - 2D variables – 2:1
  - 3D variables – 8:1
- Evaluate using student T-test
- Outcomes/Issues
  - Looks great!
  - Limited temporal variability
  - Does not preserve zeros

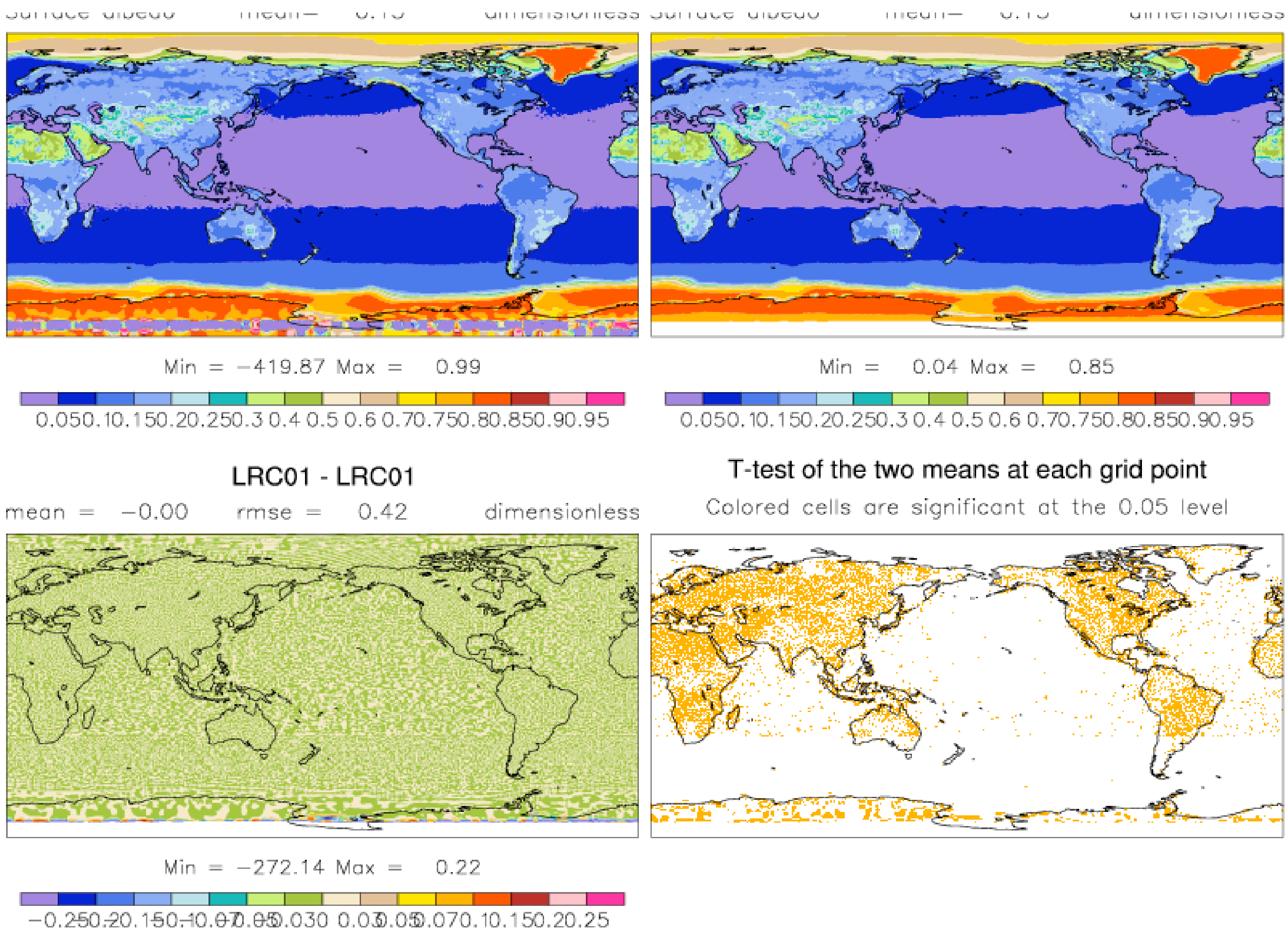
TS (Surface Temperature)



CLDTOT (Total Cloud)



ALBSURF (Surface Albedo)



Acknowledgements

- NCAR:
  - D. Bailey
  - F. Bryan
  - T. Craig
  - B. Eaton
  - J. Edwards [IBM]
  - N. Hearn
  - K. Lindsay
  - N. Norton
  - M. Vertenstein
- COLA:
  - J. Kinter
  - C. Stan
- U. Miami
  - B. Kirtman
- U.C. Berkeley
  - W. Collins
  - K. Yelick (NERSC)
- U. Washington
  - C. Bitz

- NICS:
  - M. Fahey
  - P. Kovatch
- ANL:
  - R. Jacob
  - R. Loy
- LANL:
  - E. Hunke
  - P. Jones
  - M. Maltrud
- LLNL
  - D. Bader
  - D. Ivanova
  - J. McClean (Scripps)
  - A. Mirin
- ORNL:
  - P. Worley

- Grant Support:
  - DOE
    - DE-FC03-97ER62402 [SciDAC]
    - DE-PS02-07ER07-06 [SciDAC]
  - NSF
    - Cooperative Grant NSF01
    - OCI-0749206 [PetaApps]
    - CNS-0421498
    - CNS-0420873
    - CNS-0420985
- Computer Allocations:
  - TeraGrid TRAC @ NICS
  - DOE INCITE @ NERSC
  - LLNL Grand Challenge
- Thanks for Assistance:
  - Cray, NICS, and NERSC

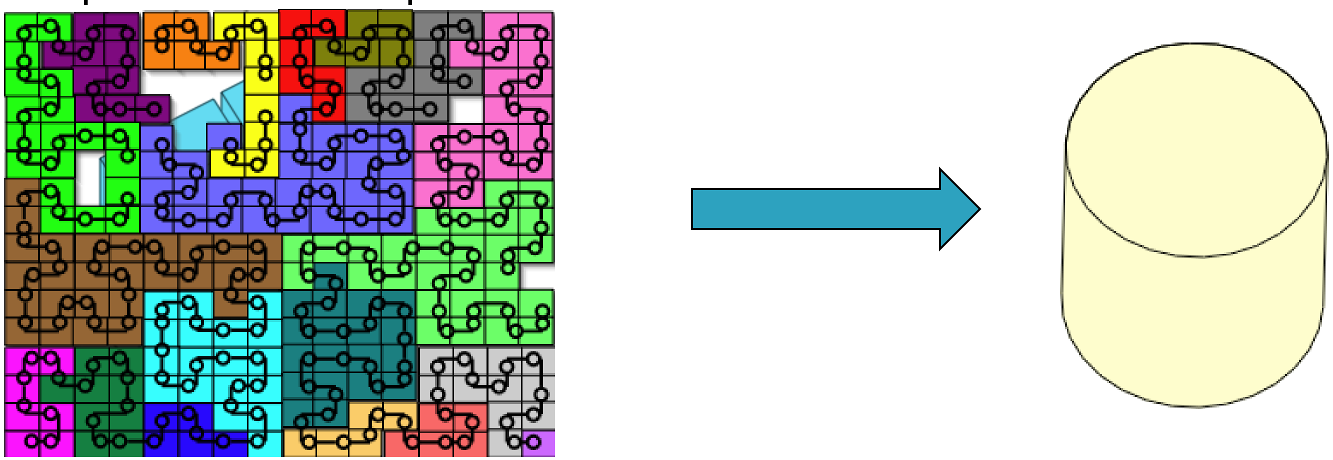
and many more…

Questions?

dennis@ucar.edu

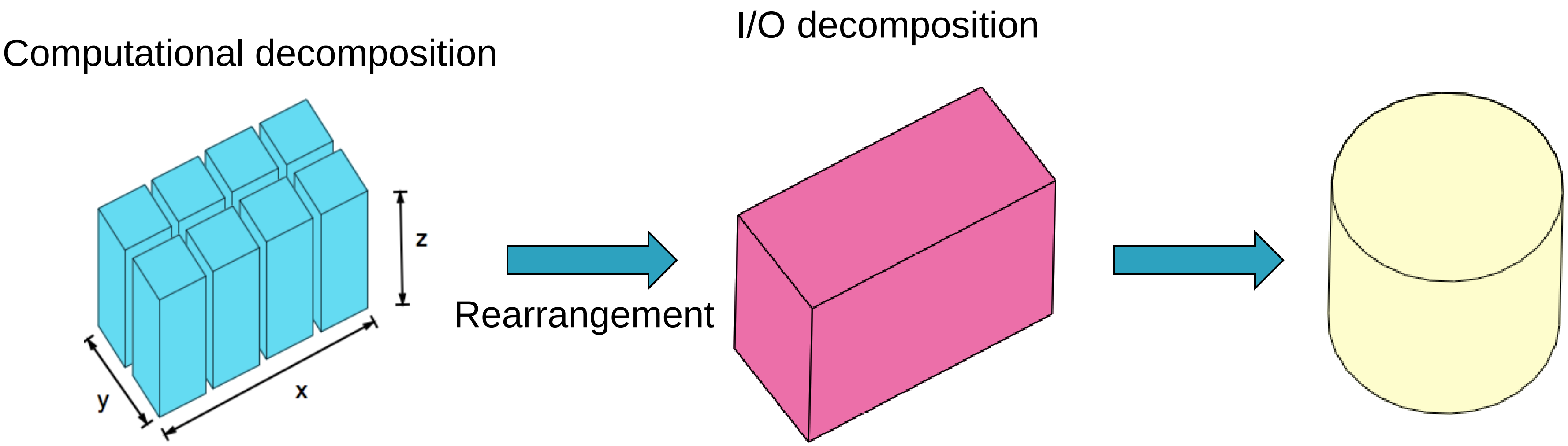
PIO:
Writing distributed data (I)

Computational decomposition



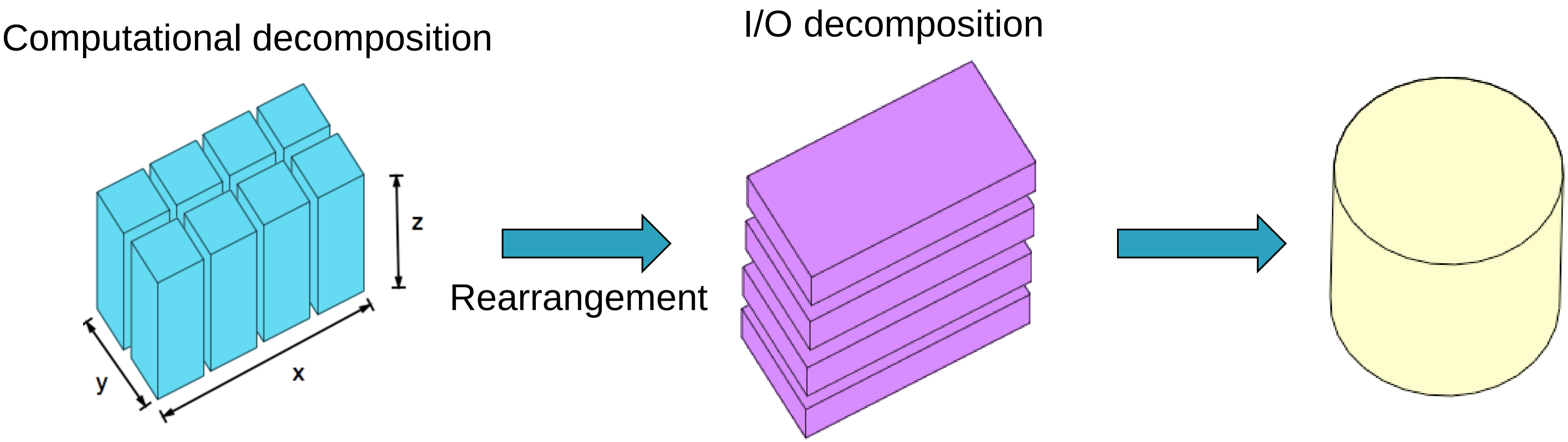
+ Simple
+ Most versions of MPI-IO will do aggregation
- Computational decomposition may not be optimal for disk access
- pNetCDF requires block cyclic decompositions

PIO:
Writing distributed data (II)



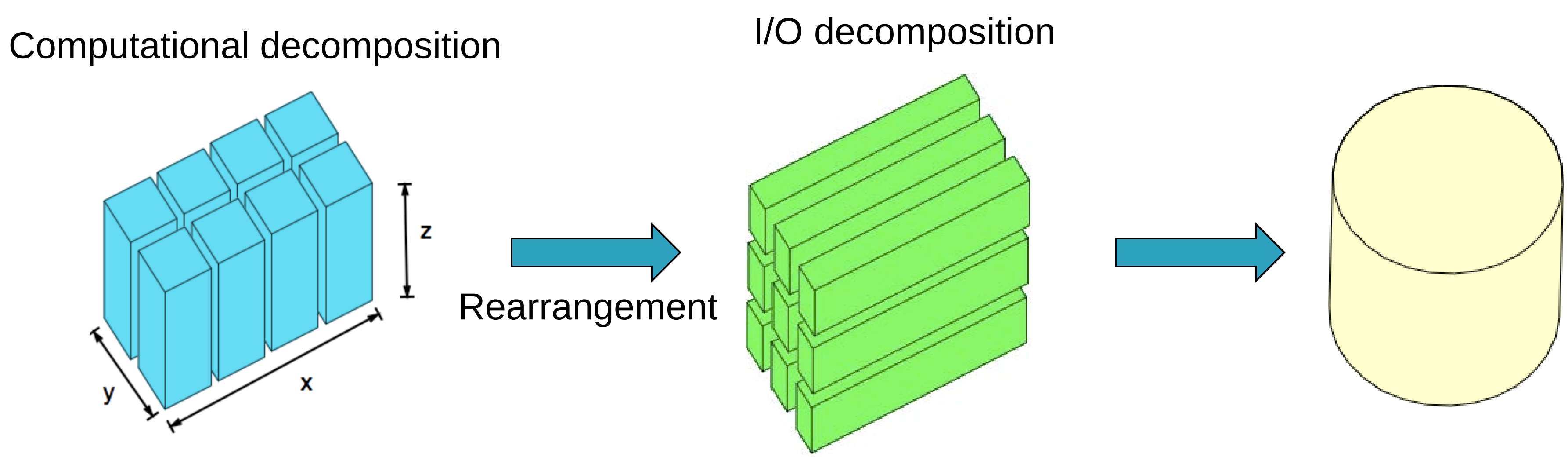
+ Maximize size of individual io-op's to disk
- Non-scalable user space buffering
- Very large fan-in → large MPI buffer allocations

PIO:
Writing distributed data (III)

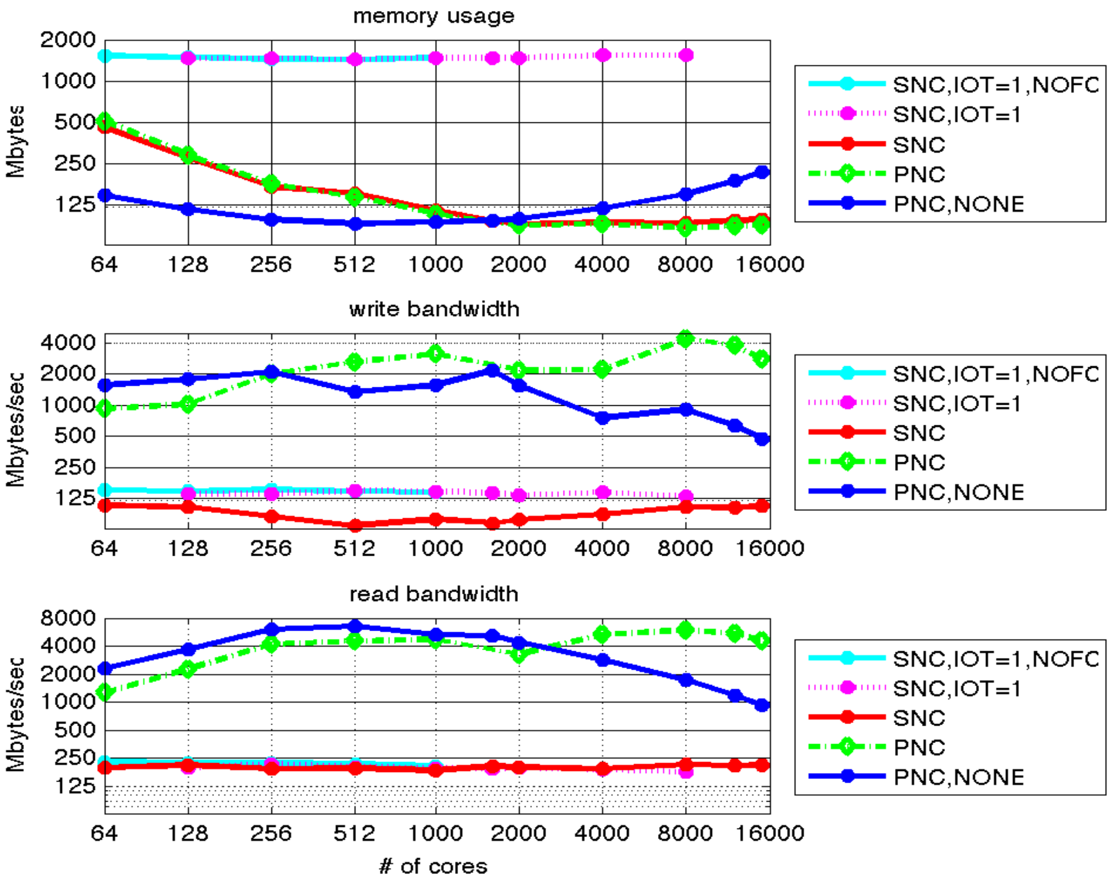


+ Scalable user space memory
+ Relatively large individual io-op’s to disk
- Very large fan-in → large MPI buffer allocations

PIO: Writing distributed data (IV)



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


memory usage
Mbytes
2000
1000
500
250
125
64
128
256
512
1000
2000
4000
8000
16000
SNC,IOT=1,NOFC
SNC,IOT=1
SNC
PNC
PNC,NONE
write bandwidth
Mbytes/sec
4000
2000
1000
500
250
125
read bandwidth
8000
# of cores

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