UNIVERSITY OF MARYLAND

Adaptive Compression to Improve I/O Performance for Climate Simulations

Swati Singhal

Alan Sussman

UMIACS and Department of Computer Science

Scientific Data Compression?

- **Data reduction is growing concern for scientific computing**
  - Motivating Example: Meso-Scale Climate Simulation Application from Department of Atmospheric and Oceanic Sciences at UMD
    - An ensemble simulation with data assimilation on 1.3 million grid points that forecasts for a nine hour period, repeatedly
      - Simulation time: ~65 mins on a cluster for a 9 hour forecast (~60% of the time is spend in I/O and extraneous work)
      - Generates ~283 GB of data per simulation

- **Possible solution**
  - **Data compression**
    - Reduces data volume for I/O and increases effective I/O bandwidth

Scientific Data Compression

- **Scientific data often are multidimensional arrays of floating point numbers, stored in self-describing data formats (e.g., netCDF, HDF, etc.)**
  - Difficult to compress -> high entropy in lower order bytes

0.00589	00111011	1100000	**100000000**	**11100111**
0.00590	00111011	1100000	**101010100**	**11001010**

Hard to achieve good compression ratio.

- **What methods are available to compress scientific data?**
  - Two categories of data compression methods
    - Lossy compression
      E.g., ZFP (Linstrom), SZ (Di),
      ISABELA (Lakshminarasimhan)
    - Lossless compression
      - E.g., ZLIB, LZO, BZIP2, FPC (Burtscher),
        ISOBAR (Schendel)

Lossy methods provide high compression but **precision is lost**

Lossless methods retain precision but are not sufficient to achieve high compression
- **require preprocessing techniques**

Scientific Data Compression

- **Which compression method to use for the given data?**
  - Typically one-time offline analysis on small subset of data
  - Criteria is based on either compression ratio or compression speed depending on application needs
  - One compression method for all the data variables

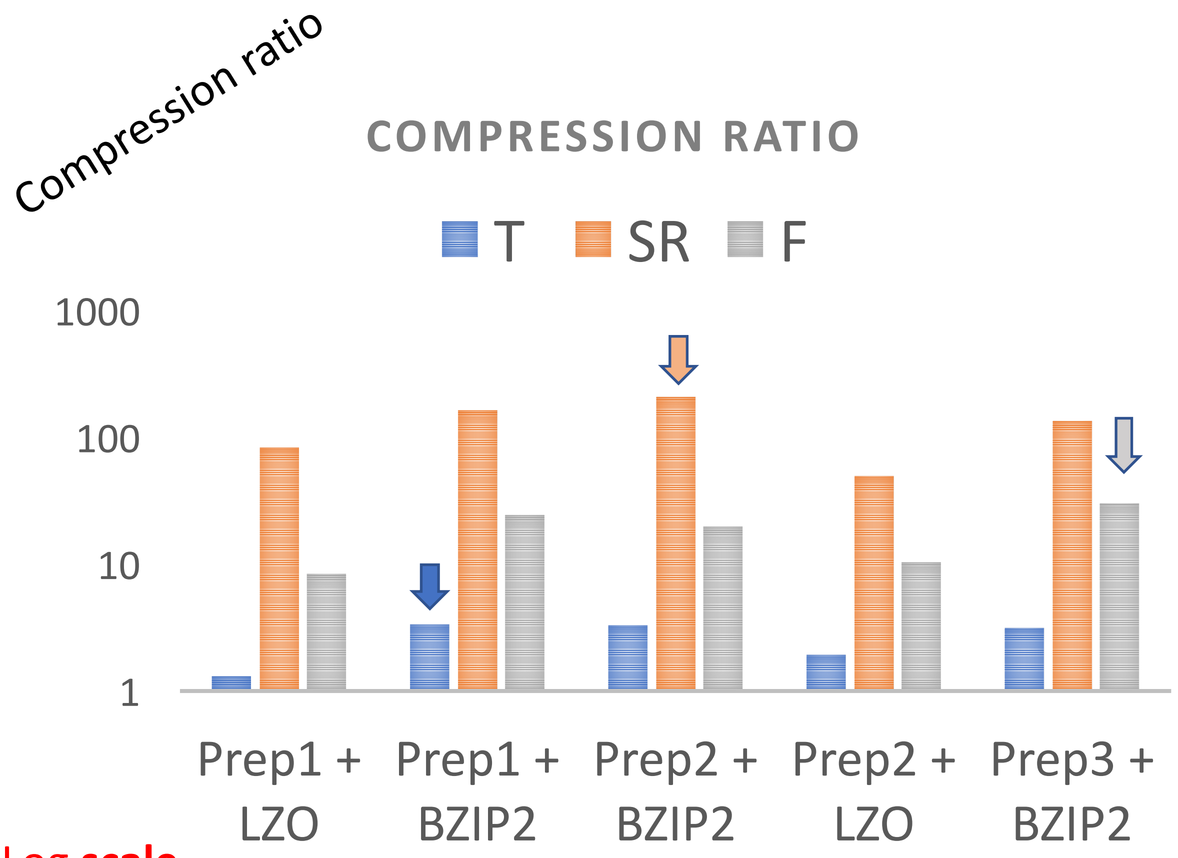
ISSUES?

- Manual effort required to select a compression scheme
- Limited measurements to define performance criteria
- **Loss of compression benefits -> best compression method differs for different variables (and may also change for same variable over time)**

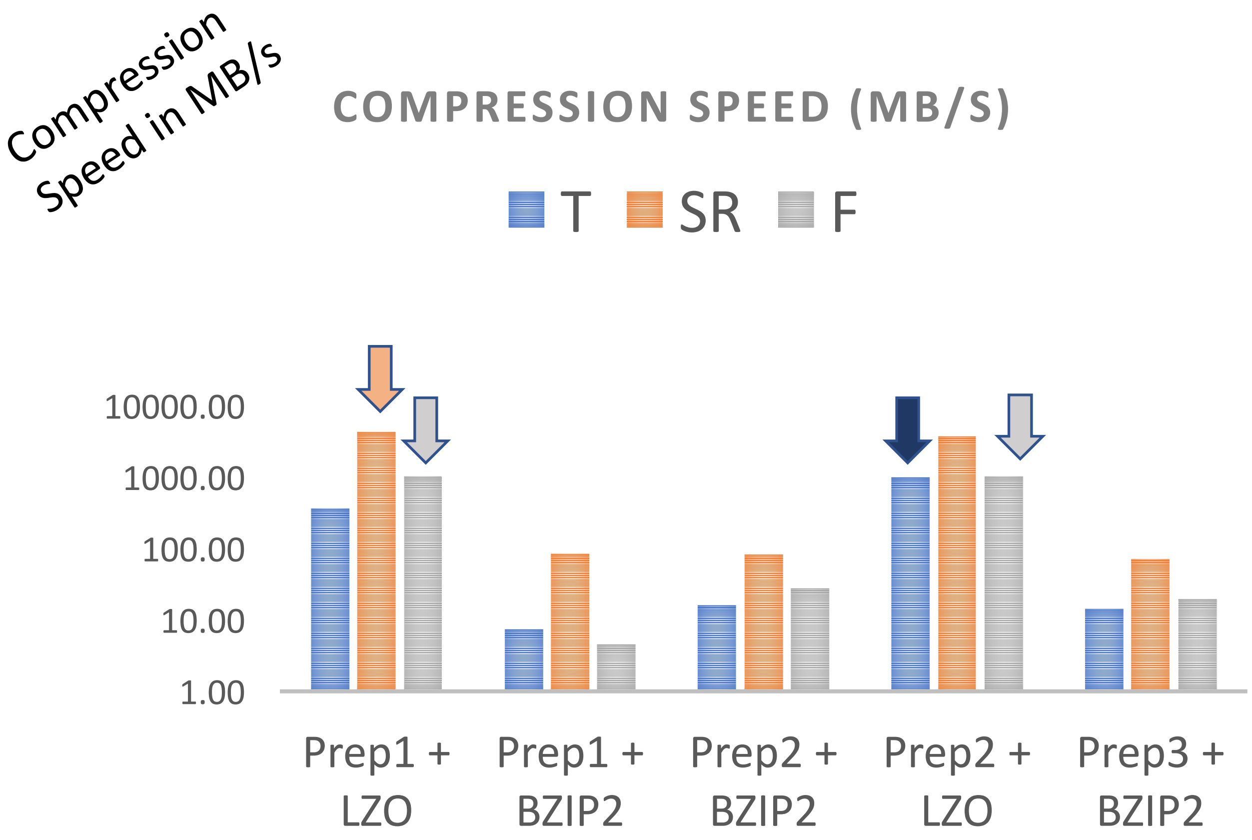
Can we do better?

Best compression method differs for different variables

Results from a single WRF output file



Log scale



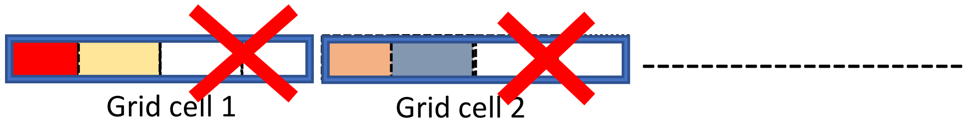
Log scale

ACOMPS:
Adaptive Compression Scheme

- **An Adaptive compression tool that**
  - Supports a set of lossless compression methods combined with different memory preprocessing techniques
  - **Automatically selects the best compression method for each variable in the dataset**
  - Allows flexible criteria to select the best compression method
  - Allows compressing data in smaller units/chunks for selective decompression to increase effective I/O bandwidth

ACOMPS: Adaptive Compression Scheme

Preprocessing techniques supported ***

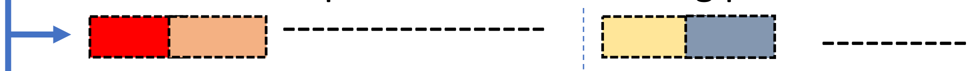


Bytes segregation (B)

Identify and segregate compressible bytes for compression

Byte-Wise segregation (BW)

Segregate compressible bytes and group these bytes based on their position in the floating point number.

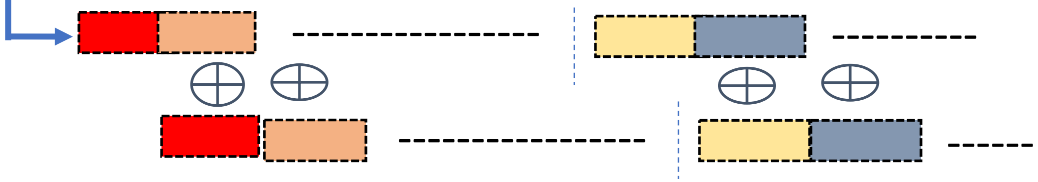


First compressible byte of all grid cells

Second compressible byte of all grid cells

Byte-Wise segregation and XOR (BWXOR)

Byte-Wise segregation + XOR



Lossless compression methods supported ***

LZO

ZLIB

BZIP2

Total 9 compression techniques

B - LZO
B - ZLIB
B - BZIP2

BW - LZO
BW - ZLIB
BW - BZIP2

BWXOR - LZO
BWXOR - ZLIB
BWXOR - BZIP2

***Other preprocessing and compression(both Lossy and Lossless) techniques can be added

ACOMPS : Adaptive Compression Scheme

Criteria to evaluate the performance of any compression technique X

$$\textbf{performance}_x = \text{compression\_speed}_x * W_{CS} + \text{compression\_ratio}_x * W_{CR}$$

User tunable parameters:

$\mathbf{W_{CR}}$ => compression ratio weighting for deciding best compression method.

$\mathbf{W_{CS}}$ => compression speed weighting for deciding best compression method.

$\triangle$ => small delta limit to define acceptance range.

Variable **A** to be compressed at **time step 0**

Determine the best technique, $\mathbf{T_x}$.
Record
$\mathbf{Best_A = T_x}$
$\mathbf{BestPerf_A = performance_{Tx}}$

Compress the data using technique $\mathbf{Best_A}$ and record $\mathbf{latestPerf_A = performance_{BestA}}$

ACOMPS : Adaptive Compression Scheme

Criteria to evaluate the performance of any compression technique X

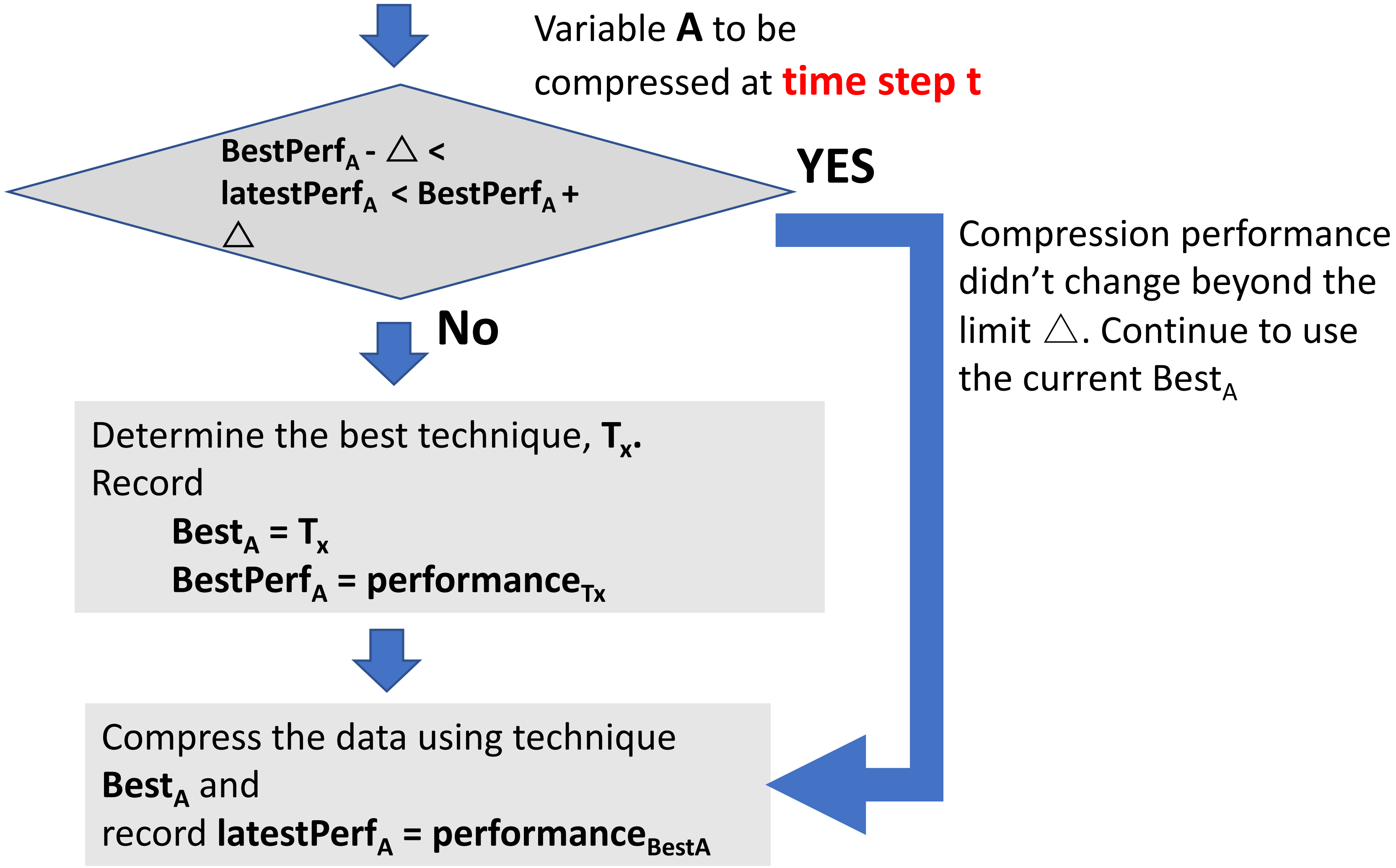
$$\textbf{performance}_x = \text{compression\_speed}_x * W_{CS} + \text{compression\_ratio}_x * W_{CR}$$

User tunable parameters:

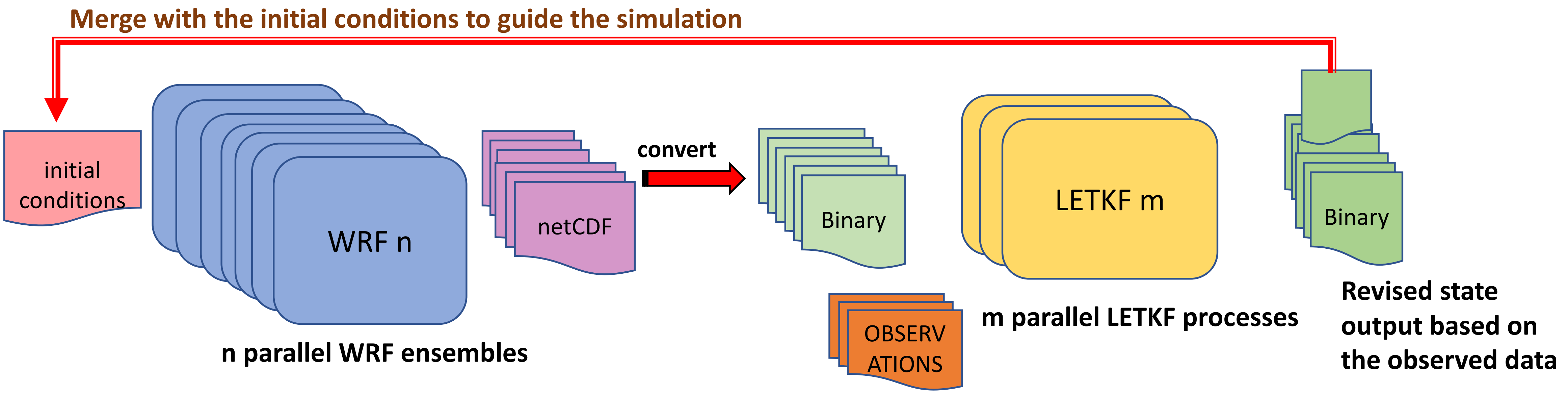
$\mathbf{W_{CR}}$ => compression ratio weighting for deciding best compression method.

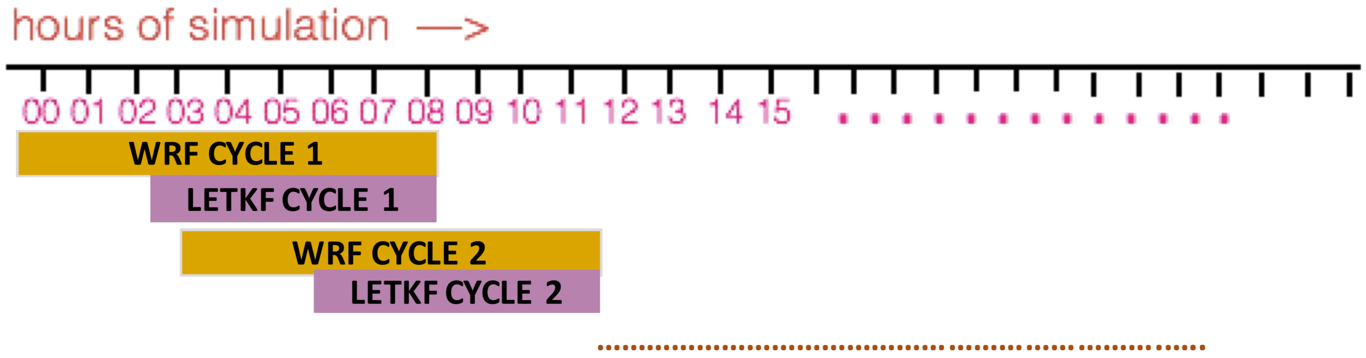
$\mathbf{W_{CS}}$ => compression speed weighting for deciding best compression method.

△ => small delta limit to define acceptance range.

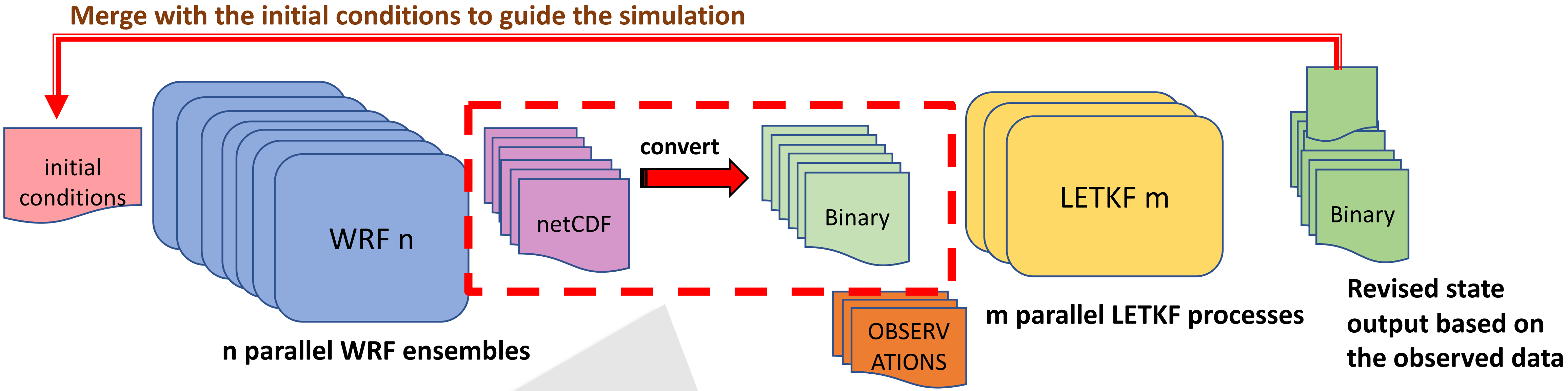


WRF-LETKF based climate simulations





WRF-LETKF based climate simulations



Example: n = 55, m = 400. Max grid size : 181 x 151 x 51

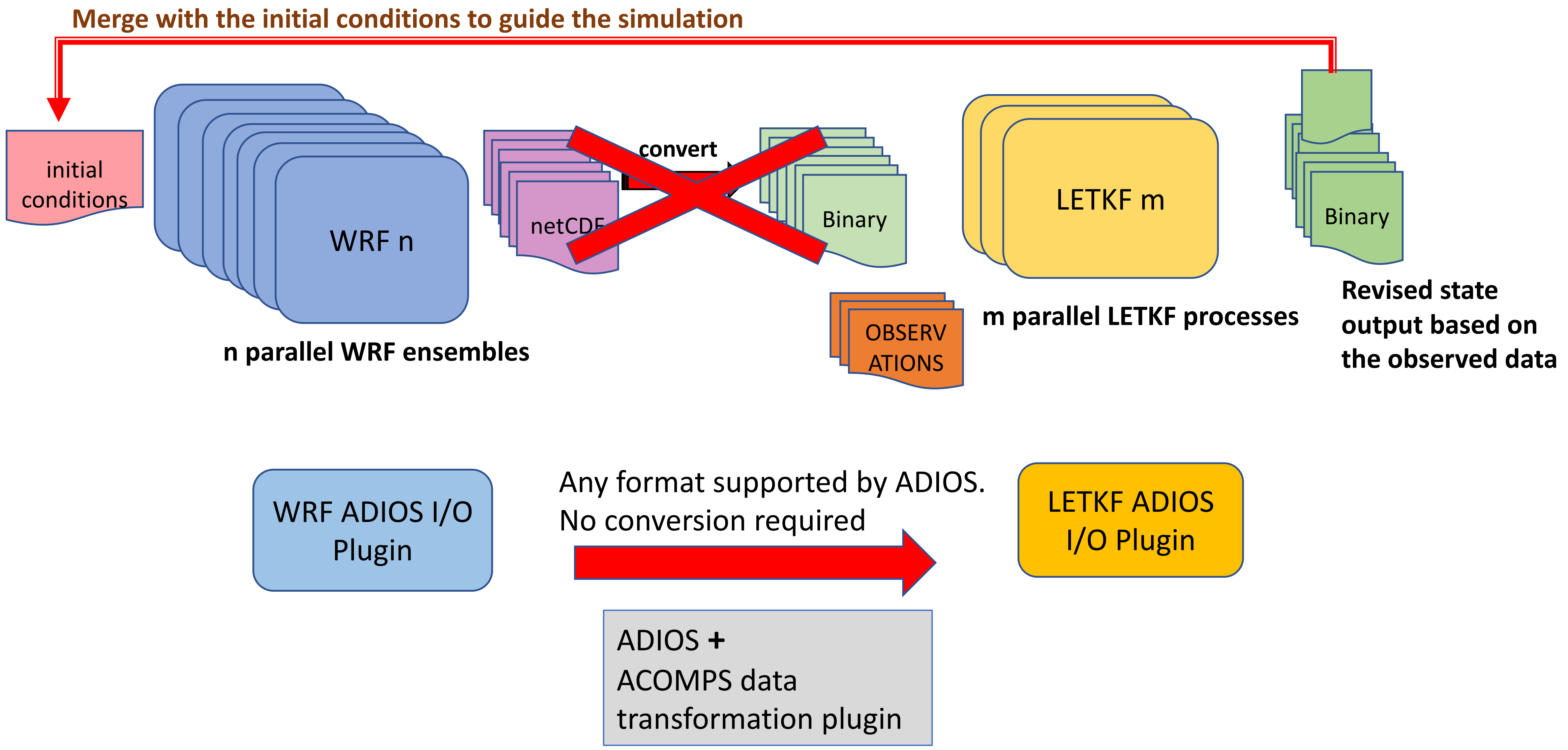
<u>Single cycle WRF + single cycle LETKF (9 hours simulation time)</u>

Total simulation time (single cycle WRF + single cycle LETKF) : ~ 65 mins on cluster

High conversion cost (9 x 55) files => 36.7 minutes => ~56% of the total simulation time

Large output data size : ~283 GB

WRF-LETKF based climate simulations



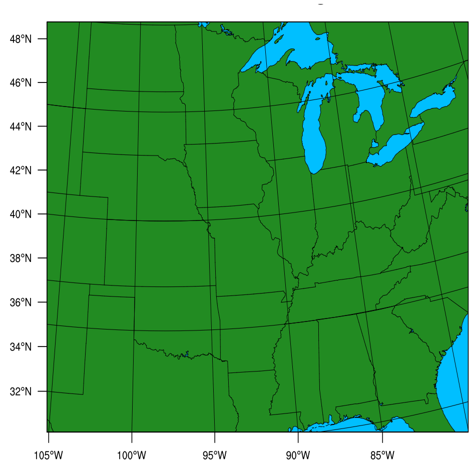
Experimental Setup

- **Deepthought2**

Campus cluster at UMD

Number of nodes : 484 with 20 cores/node + 4 nodes with 40 core/node
Memory/node ~ 128 GB (DDR3 at 1866 Mhz)
Processor : dual Intel Ivy Bridge E5-2680v2 at 2.8 GHz
Parallel File system : Lustre

- **Climate simulations with WRF-LETKF**



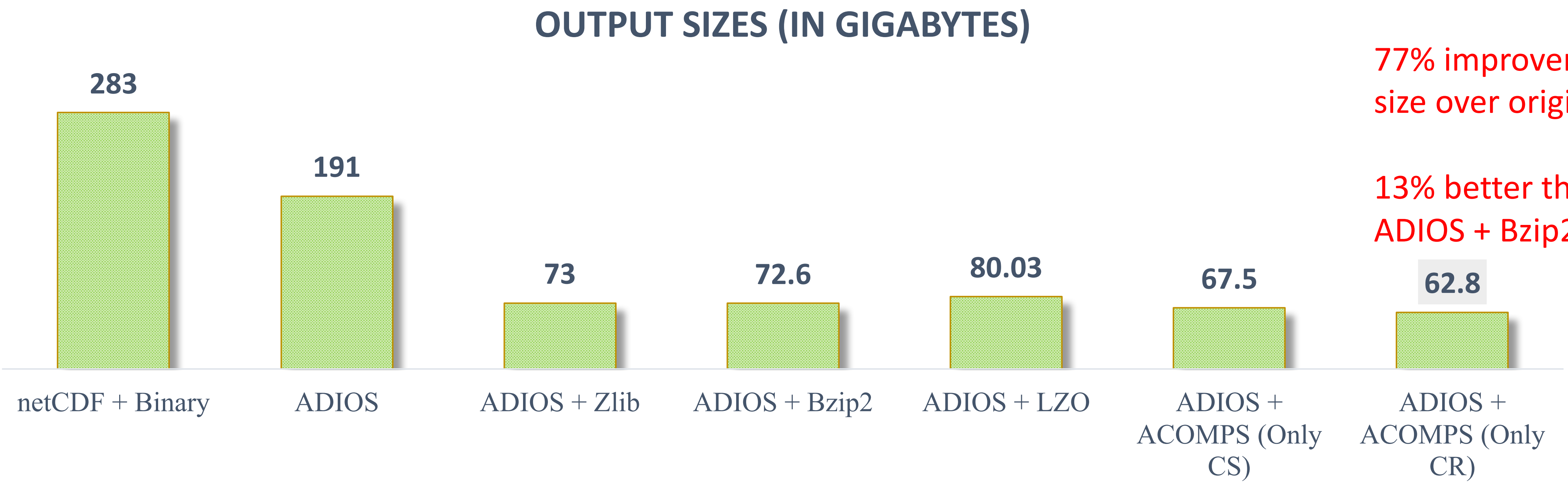
Domain size : 181 x 151 grid cells
Vertical levels : 51

Majority of variables are float type
3D variable : XLAT, XLONG, F, T…
4D variables: U, V, W, P, PB, RAINC..
…

WRF Ensemble **n = 55** => each uses 1 node
No. of MPI processes = (55 x 20) = 1100

LETKF => uses 20 nodes
No. of MPI processes = (20 x 20) = 400

Adaptive Vs Non-adaptive methods : Output sizes



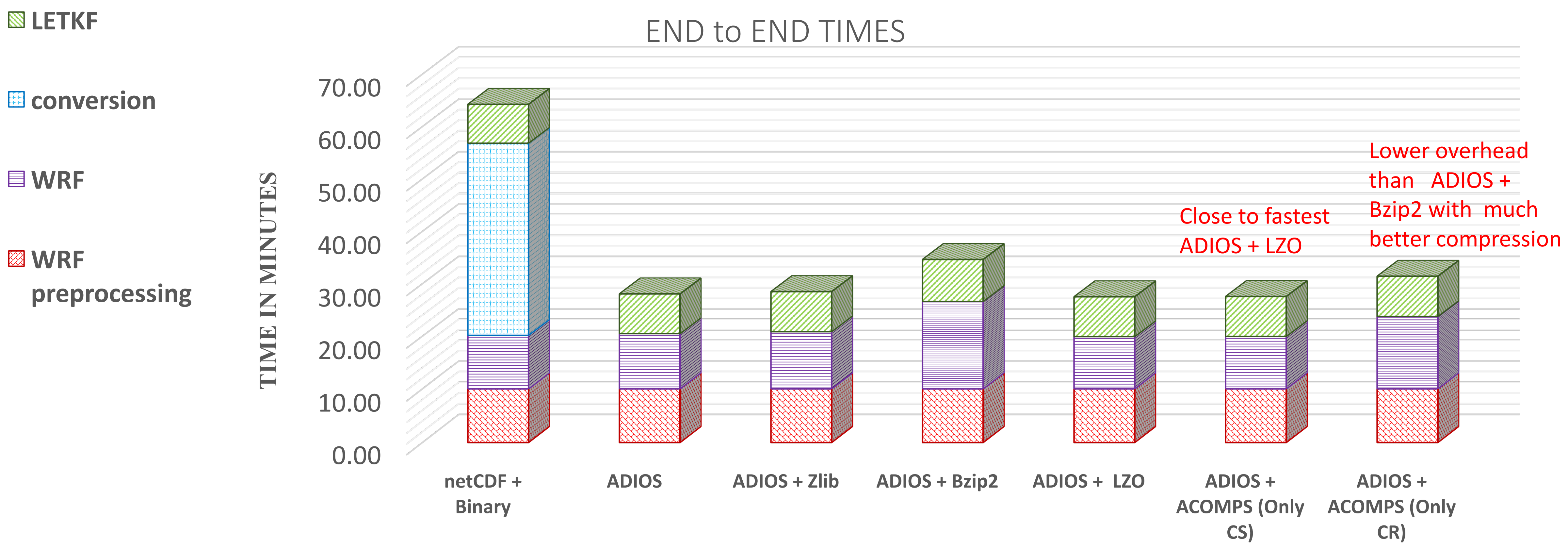
77% improvement in size over original

13% better than ADIOS + Bzip2

ACOMPS achieves better compression

Only CR (Best compression ratio, slower) => W_{CR} = 1, W_{cs} = 0
Only CS (Best speed, not as good compression) => W_{CR} = 0, W_{cs} = 1

Adaptive Vs Non-adaptive methods : Compression time



ACOMPS incurs low overhead

Only CR(Best compression ratio, slower) => $W_{CR} = 1, W_{cs} = 0$

Only CS(Best speed, not as good compression) => $W_{CR} = 0, W_{cs} = 1$

Future Directions

- Extend to support more compression methods including both lossless and lossy compression methods

- Thoroughly analyze how the best compression method for a given variable changes over time
  - How often it is advantageous to do the re-analysis?
  - How to enhance the criteria to decide when to re-evaluate in order to adapt to the changes quickly

- Parallelize the analysis phase using threads

UNIVERSITY OF MARYLAND

Adaptive Compression to Improve I/O Performance for Climate Simulations

Swati Singhal
swati@cs.umd.edu

Alan Sussman
als@cs.umd.edu

UMIACS and Department of Computer Science