

An Evaluation of Science Data Formats and Their Use at the Community Coordinated Modeling Center

Marlo Maddox Code 587

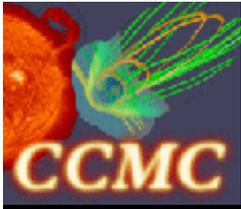
**Advanced Data Management &
Analysis Branch**



HDF/HDF-EOS Workshop VII - Silver Spring, MD

September 23 – 25, 2003



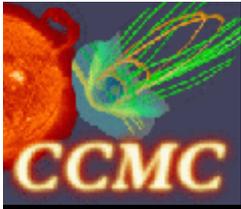


The Community Coordinated Modeling Center

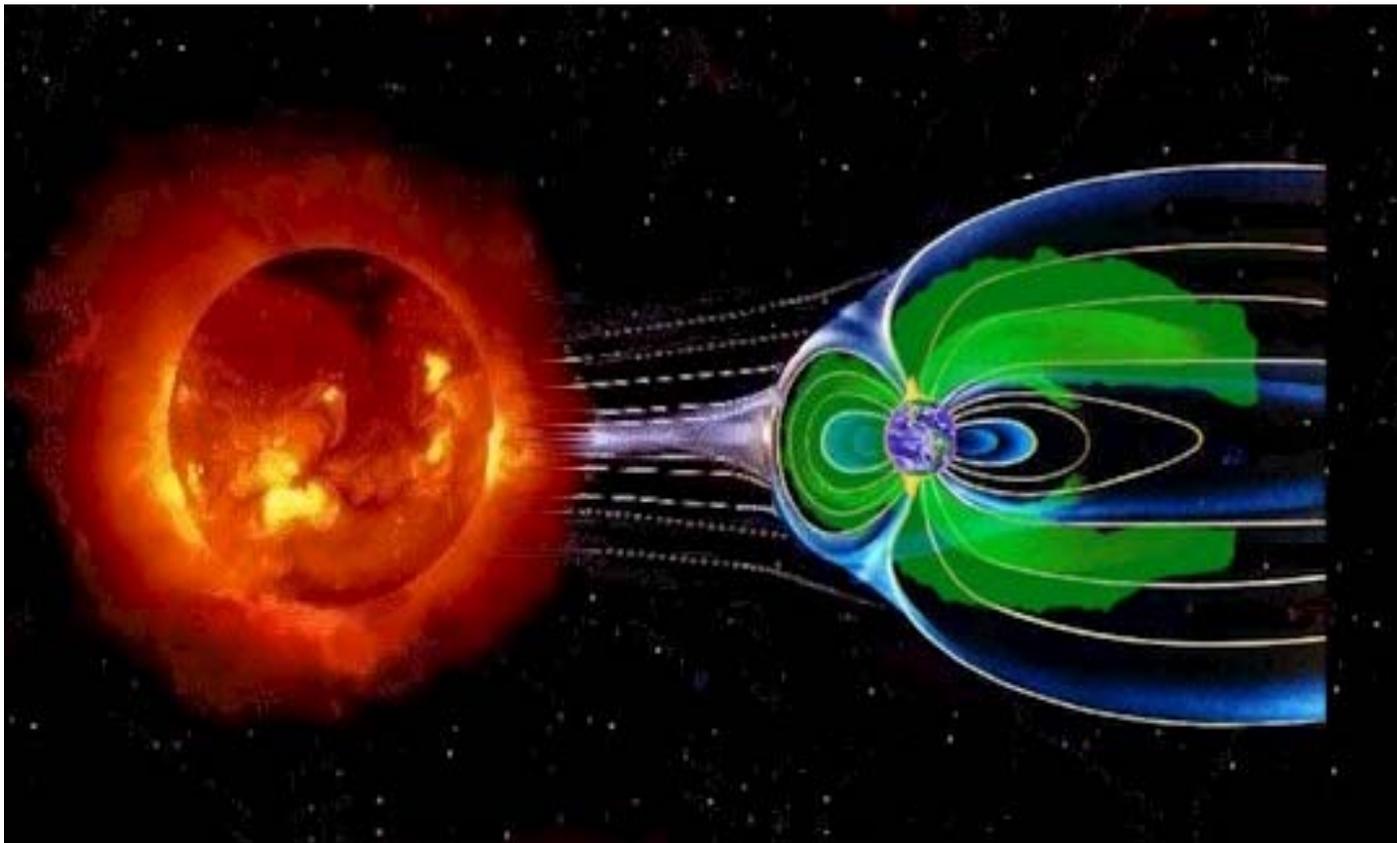
What the CCMC provides:

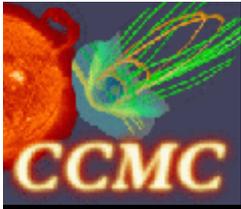
- Scientific validation
- Model coupling
- Metrics implementations
- Advanced visualization
- Model runs on request



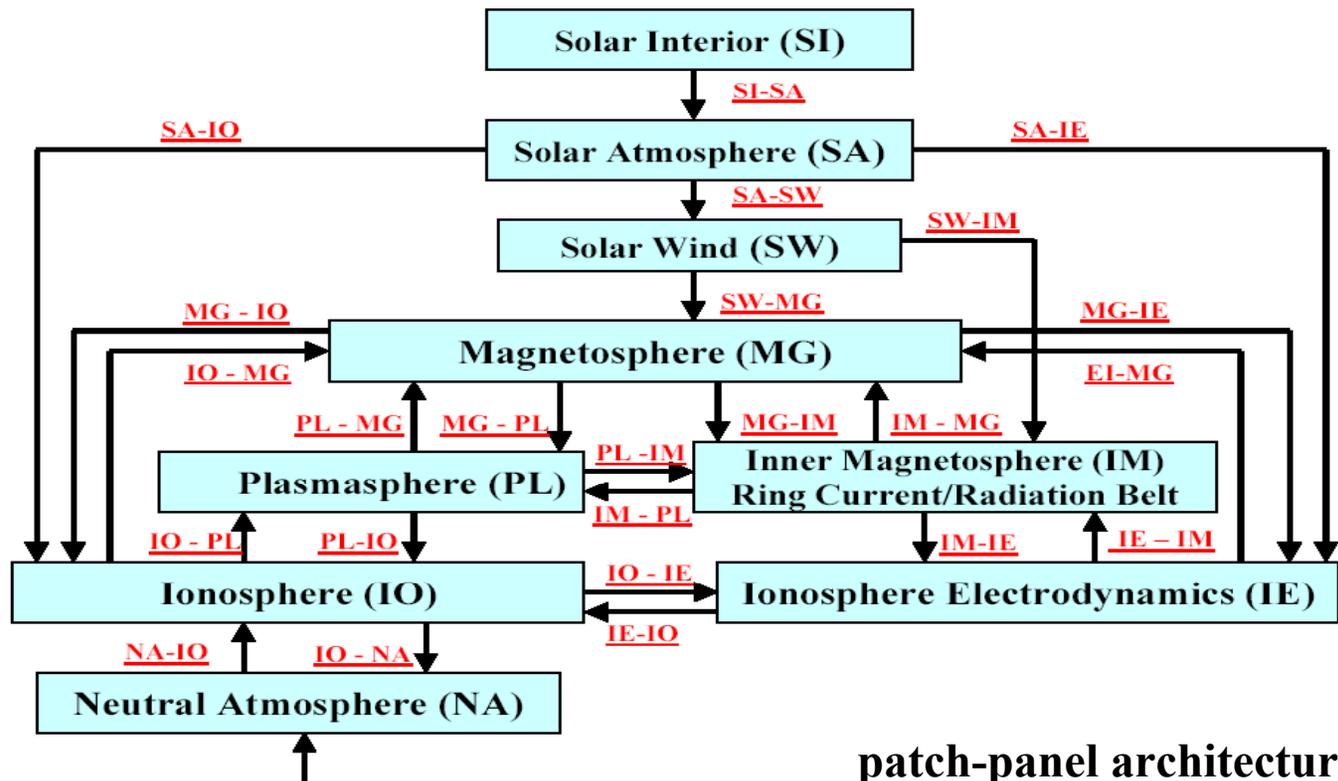


Covering the Entire Domain





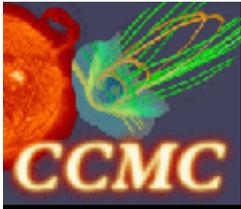
Space Weather Models





Challenges

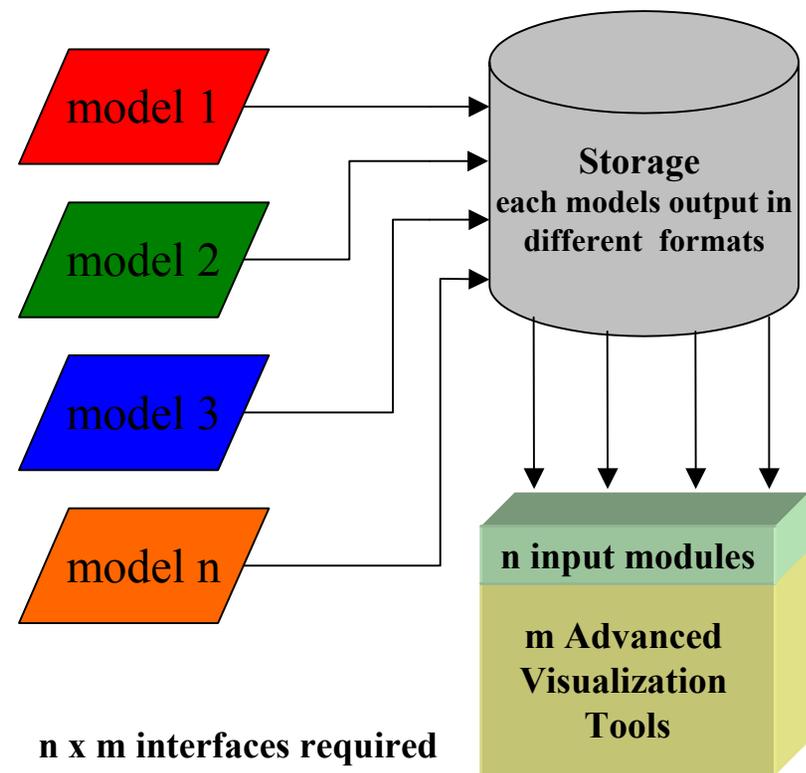
- No rules for standard model interfaces
- Each new model has unique output format
- Developer/user needs to become familiar with internal structure of each output file
- Custom read routines to access model data
- Data is not self describing
- Reduces portability and reuse of
 - Data output itself
 - Tools created to analyze data

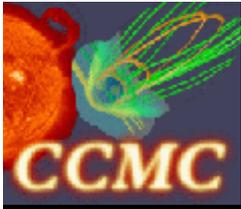


Every Models Output Is Unique

Environment Without Standard

- Specialized I/O routines required for every interface
- Unsuitable for use in flexible model chain
- No commonality between data passing through interfaces

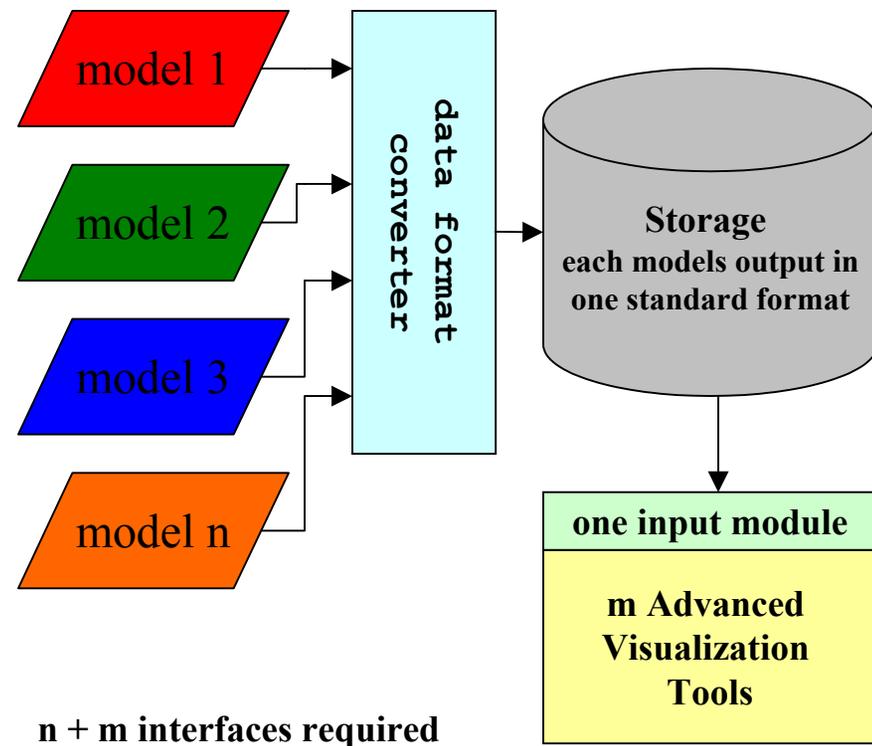




Every Models Output Is Unique

Standardized Environment

- **Original output can be preserved**
- **Standard format for storage, coupling, & visualization**
- **Model developers continue to have freedom of choice**
- **Ensures compatibility between models for coupling**
- **Ground work for which standard, reusable interfaces and tools can be developed**





Model Selected for Testing

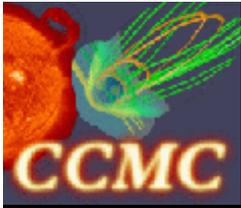
- Block-adaptive-tree-Solarwind-roe-upwind-scheme (BATSRUS) global magnetosphere MHD model
 - Developed by CSEM at university of Michigan
 - Uses MPI and Fortran 90 standard
 - Executes on massively parallel computer systems
 - Adaptive grid of blocks arranged in varying degrees of spatial refinement levels
 - Solves 3D MHD equations in finite volume form using numerical methods related to roe's approximate Riemann solver
 - Attached to an ionospheric potential solver that provides electric potentials and conductances in the ionosphere



Understanding the BATSRUS Models Output

General Scientific Output

- magnetospheric plasma parameters
 - Atomic mass unit density
 - Pressure
 - Velocity
 - Magnetic field
 - Electric currents
- ionospheric parameters
 - Electric potential
 - Hall and Pedersen conductances



BATSRUS .OUT File

byte	value
1	number of bytes n for next record
2	
3	
4	
5	n bytes containing units for variables R amu/cm ³ km/s nT nPa J/m ³ uA/m ²
n	
n+1	number of bytes n for previous record
n+2	
n+3	
n+4	

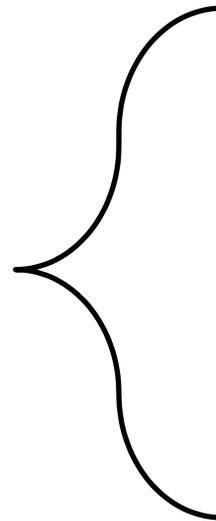
units
time step information
dimension sizes
special parameters
data variables names
grid information
variable values



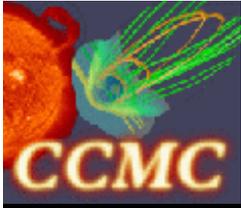


BATSRUS .OUT File

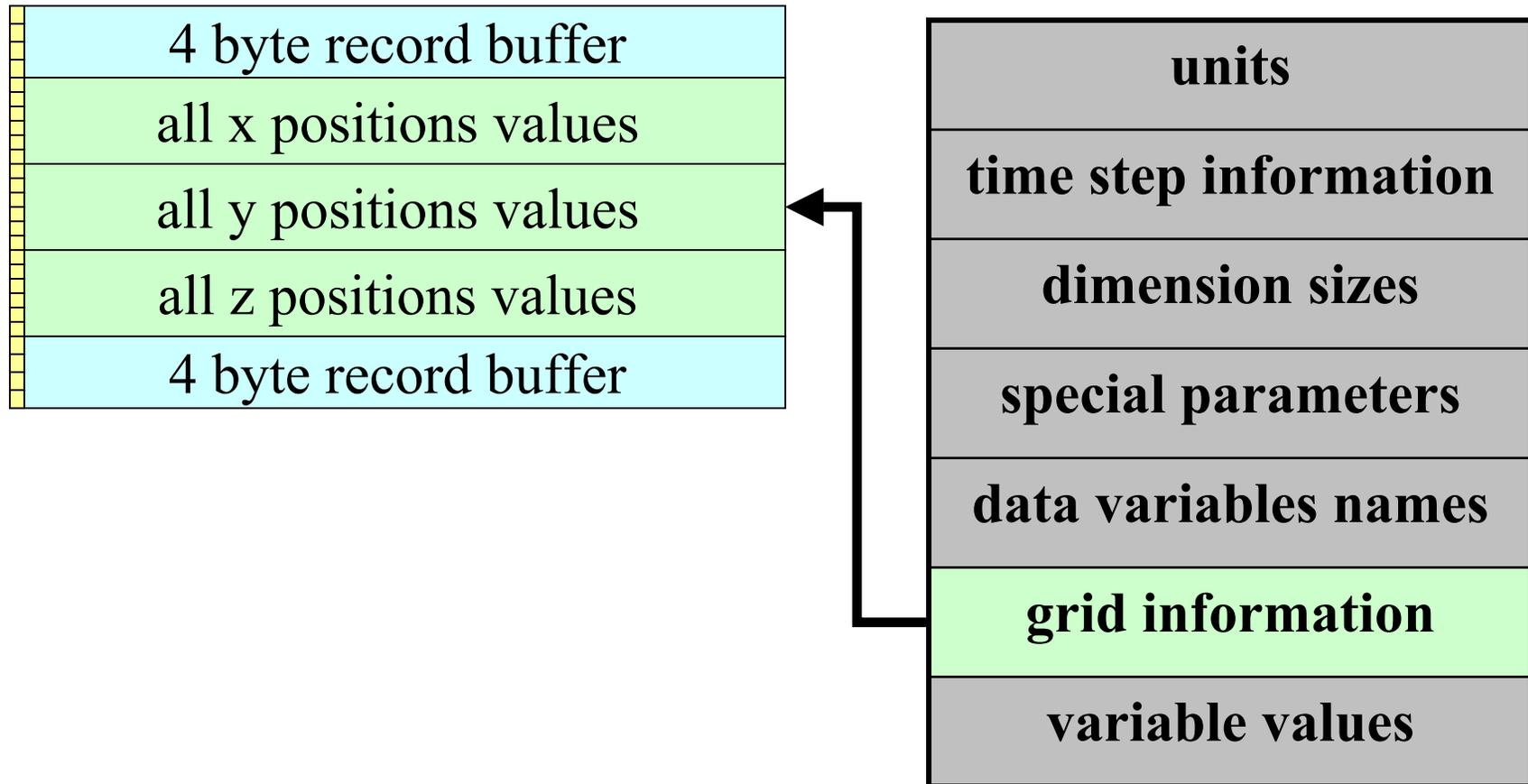
- general information
- static non-variant data

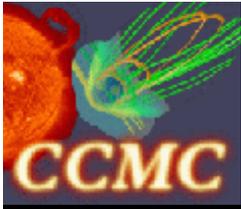


units
time step information
dimension sizes
special parameters
data variables names
grid information
variable values

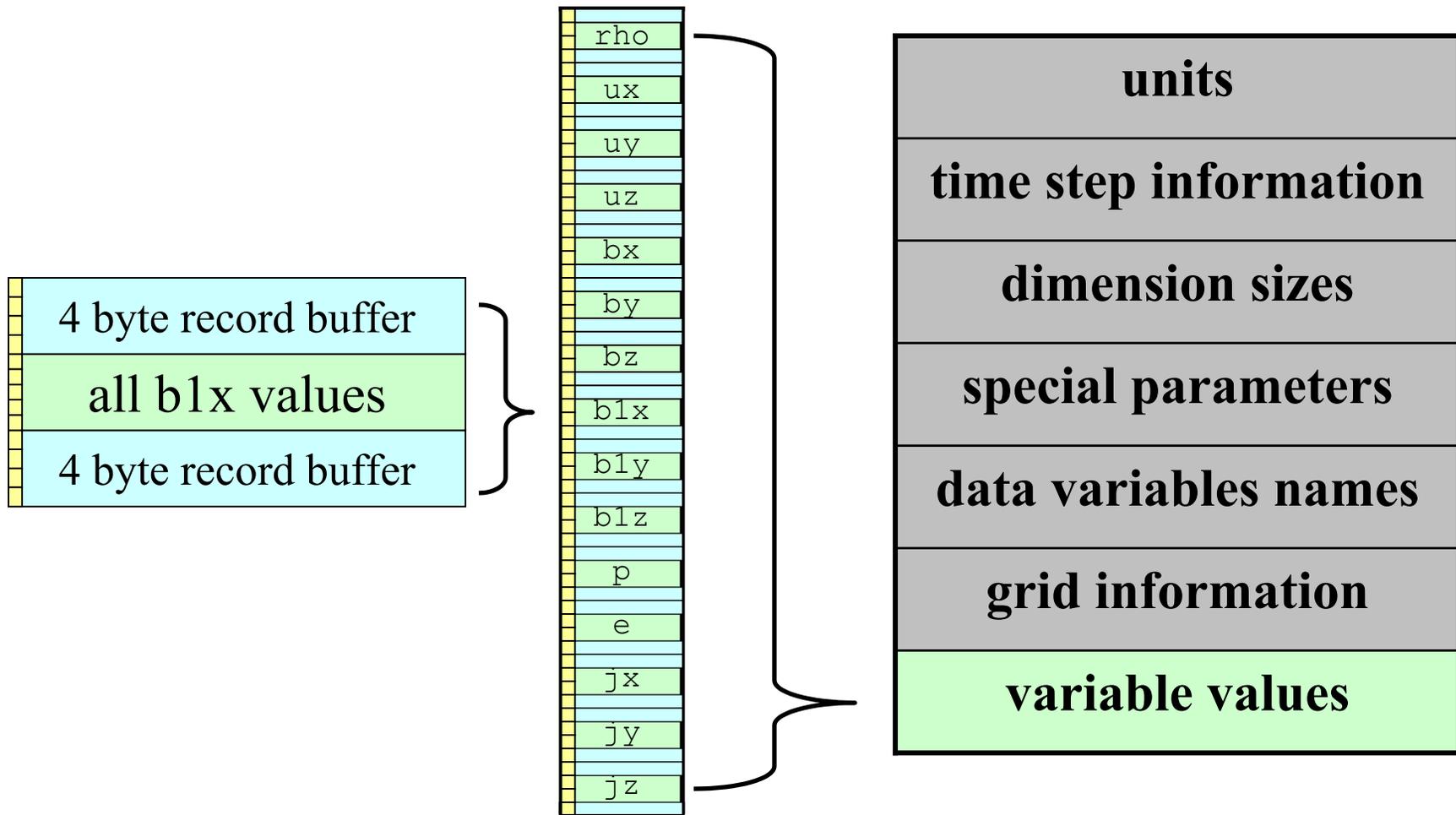


BATSRUS .OUT File





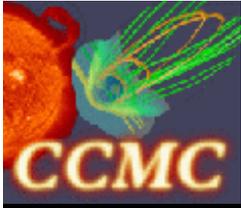
BATSRUS .OUT File





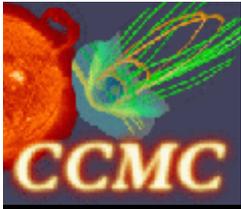
Designing the CDF

- CDF files have two main components
 - Attributes – metadata describing contents of CDF
 - Global – describe CDF as a whole
 - Variable – describe specific characteristics of the variables
 - Records – collections of variables
 - Scalar
 - Vector
 - N-dimensional arrays (where $n \leq 10$)
- Identify potential metadata (or any static data) from original output file
- Include this data in the global attributes portion of the CDF



CDF Variables

- CDFs contain two types of variables
 - rVariables – all have the same dimensionality
 - zVariables – can each have different dimensionalities
- CDF Dimensionality
 - a variable with one dimension is like an array
 - number of elements in array correspond to the dimension size

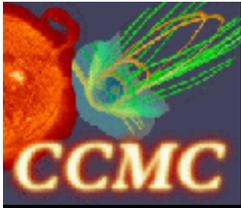


CCMC CDF Variables

x
y
z

rho
ux
uy
uz
bx
by
bz
b1x
b1y
b1z
p
e
jx
jy
jz

- BATSRUS model contains 18 dynamic variables
 - 3 position variables
 - 15 plot variables
- 18 CDF rVariables
 - one record per variable
 - one dimensional variables
 - dimension size = number of cells in grid
 - 18 records vs. 10.4 million in previous scheme



BATRUS .OUT to CDF

first column indicates
current record number

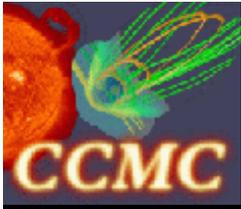
column two references the current records element
index – each element of the record stores a value for
the current variable

1:[1] = -251.0
1:[2] = -243.0
1:[3] = -235.0
1:[4] = -227.0
1:[5] = -219.0
1:[6] = -211.0
1:[7] = -251.0
1:[8] = -243.0

1:[9] = -235.0
1:[10] = -227.0
1:[11] = -219.0
1:[12] = -211.0
1:[13] = -251.0
1:[14] = -243.0
1:[15] = -235.0
1:[16] = -227.0

1:[17] = -219.0
1:[18] = -211.0
1:[19] = -251.0
1:[20] = -243.0
1:[21] = -235.0
1:[22] = -227.0
1:[23] = -219.0
1:[24] = -211.0

1:[1283401] = -251.0
1:[1283402] = -243.0
1:[1283403] = -235.0
1:[1283404] = -227.0
1:[1283405] = -219.0
1:[1283406] = -211.0
1:[1283407] = -251.0
1:[1283408] = -243.0



CDF Attributes

```
! Skeleton table for the "bats_2_cdf_OUTPUT.cdf" CDF.  
! Generated: Monday, 22-Sep-2003 17:06:08  
! CDF created/modified by CDF V2.7.1  
! Skeleton table created by CDF V2.7.1
```

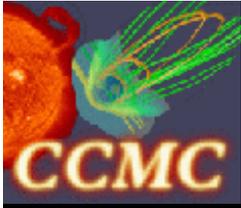
```
#header
```

```
          CDF NAME: bats_2_cdf_OUTPUT.cdf  
DATA ENCODING: NETWORK  
MAJORITY: ROW  
FORMAT: SINGLE
```

```
! Variables  G.Attributes  V.Attributes  Records  Dims  Sizes  
! -----  -  
          18/0           22             4         1/z    1  1293408
```

```
#GLOBALattributes
```

```
! Attribute      Entry      Data  
! Name           Number     Type      Value  
! -----      -  
  
"Project"        1:        CDF_CHAR  { "CCMC" } .  
  
"Disclaimer"     1:        CDF_CHAR  { "INSERT TERMS OF USAGE HERE" } .  
  
"Generated_By"   1:        CDF_CHAR  { "Marlo Maddox" } .  
  
"Generation_Date" 1:        CDF_CHAR  { "3/27/2003" } .  
  
"Simulation_Model" 1:        CDF_CHAR  { "BATSRUS" } .
```



CDF Attributes

```
"Elapsed_Time_In_Seconds"
    1:  CDF_FLOAT  { 4200.16 } .

"Number_Of_Dimensions"
    1:  CDF_INT4   { -3 } .

"Number_Of_Special_Parameters"
    1:  CDF_INT4   { 10 } .

"Special_Parameters"
    1:  CDF_FLOAT  { 1.66667 }
    2:  CDF_FLOAT  { 2248.43 }
    3:  CDF_FLOAT  { -0.368162 }
    4:  CDF_FLOAT  { 3.0 }
    5:  CDF_FLOAT  { 1.0 }
    6:  CDF_FLOAT  { 1.0 }
    7:  CDF_FLOAT  { 3.0 }
    8:  CDF_FLOAT  { 6.0 }
    9:  CDF_FLOAT  { 6.0 }
   10:  CDF_FLOAT  { 6.0 } .

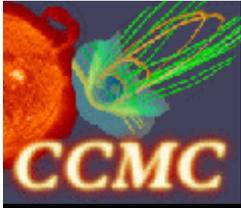
"Number_Of_Plot_Variables"
    1:  CDF_INT4   { 15 } .

"X_Dimension_Size"
    1:  CDF_INT4   { 1293408 } .

"Y_Dimension_Size"
    1:  CDF_INT4   { 1 } .

"Z_Dimension_Size"
    1:  CDF_INT4   { 1 } .

"Current_Iteration_Step"
    1:  CDF_INT4   { 22924 } .
```



CDF Variables

```
#variables
```

```
! Variable      Data      Number      Record      Dimension
! Name          Type      Elements    Variance    Variances
! -----      ----      -
```

```
"x"            CDF_FLOAT      1           T           T
```

```
! Attribute     Data
! Name          Type      Value
! -----      ----      -
```

```
"Description"
```

```
CDF_CHAR      { "X position for center of cell in grid..." }
```

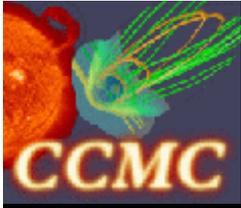
```
"Dictionary_Key"
```

```
CDF_CHAR      { "CCMC/SWMF Data Dictionary Entry" }
```

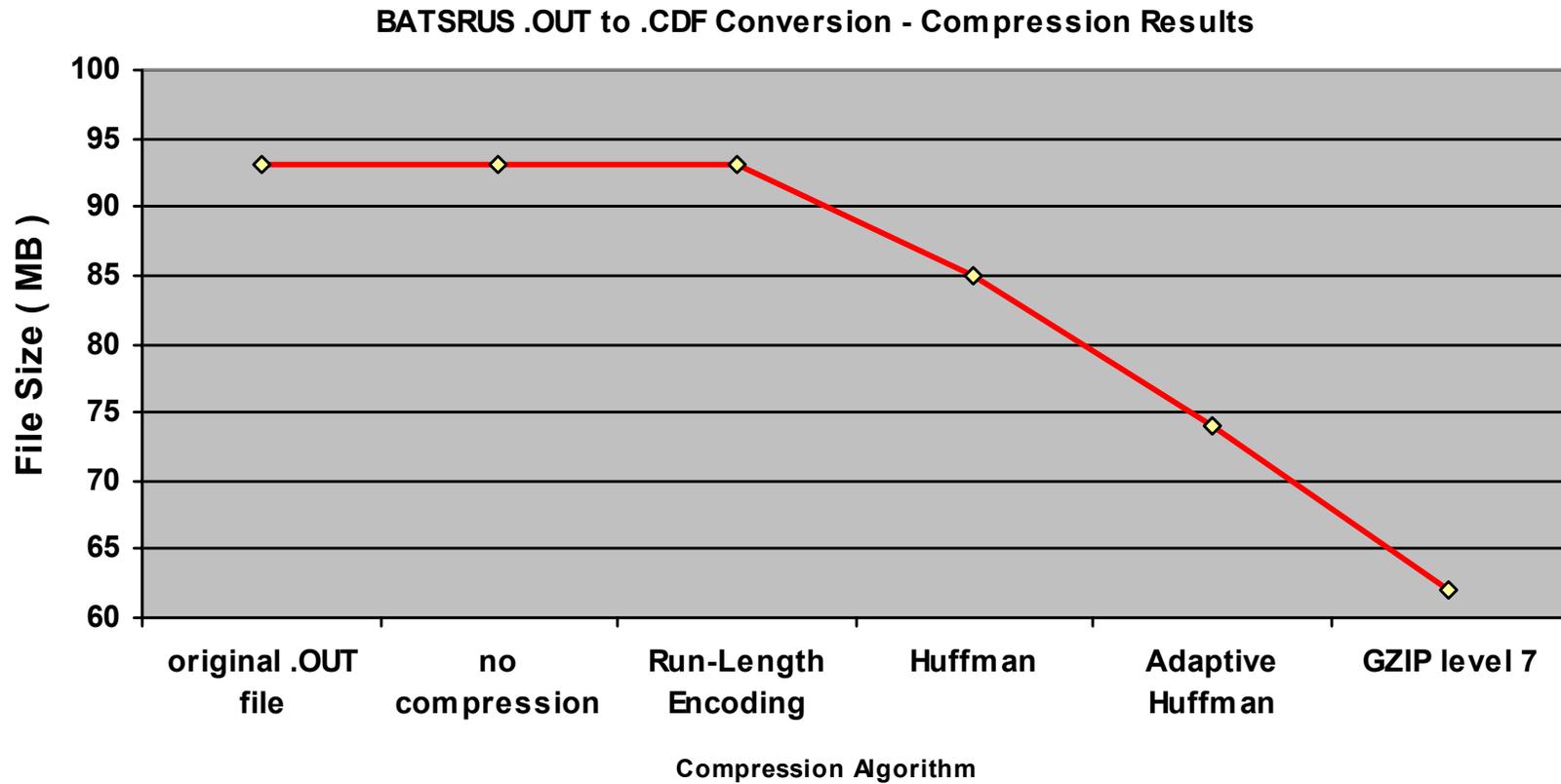
```
"Valid_Min"    CDF_FLOAT      { -100000.0 }
```

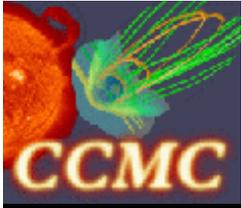
```
"Valid_Max"    CDF_FLOAT      { 100000.0 } .
```

```
! RV values were not requested.
```



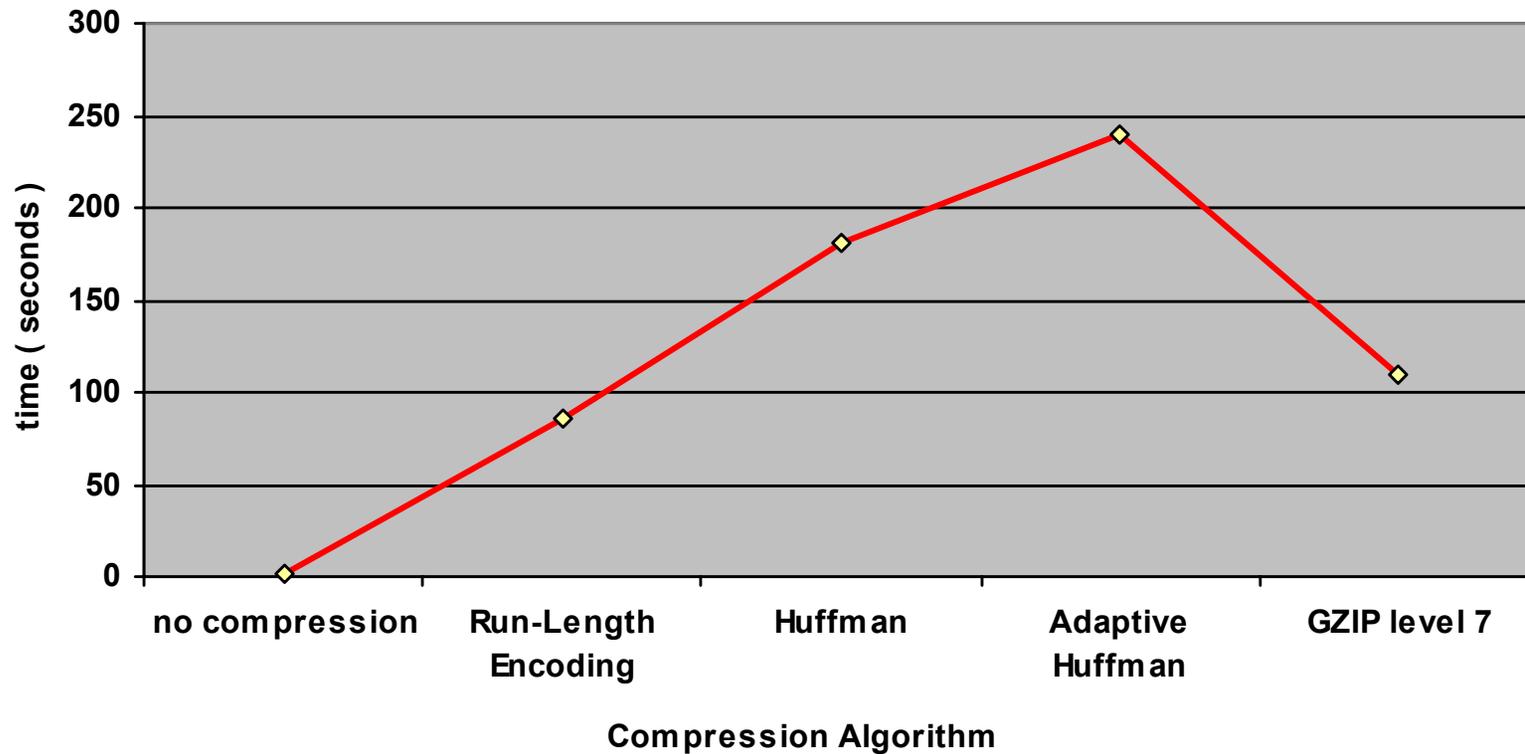
Compression Performance Tests

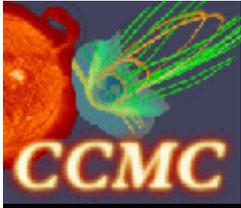




Compression Performance Tests

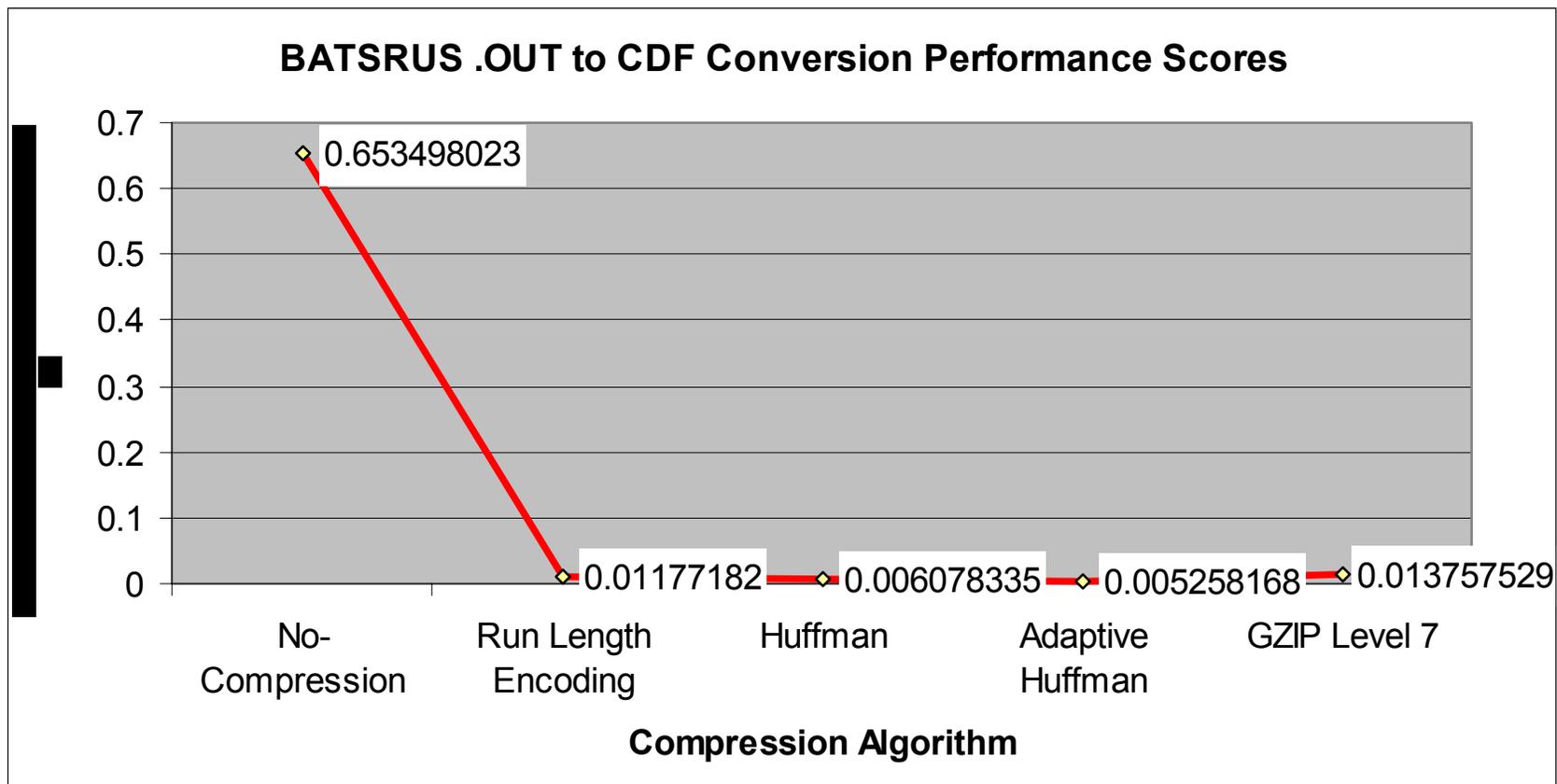
BATSRUS .OUT to .CDF Conversion - Wall Clock Time Results





Performance Score

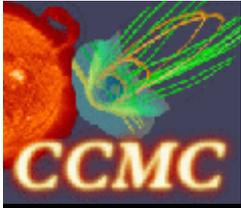
$$\frac{(\text{original_file_size})}{(\text{cdf_file_size})} * \frac{1}{t}$$





Performance Results

- Optimal CDF storage format
 - Single one-record rVariables
 - Dimension size equal to number of cells in grid
- Uncompressed CDF creation time of 1.5 seconds
- CDF file size virtually the same as original BATSRUS output file size
- Method could be applied to additional models in similar fashion



Conclusion

- BATRUS .Out to CDF conversion results promising
 - 1.5 second uncompressed CDF creation time
 - Resulting file size virtually unchanged
- OpenDx successfully imported CDF data using standard input module (*only had to specify input file name*)
 - Requires minimal initial development to correctly categorize imported data
- Closer to establishing a data format standard within the CCMC



Future Work

- Research HDF 5 data standard
- Test BATRUS output conversion performance with HDF 5
- Compare CDF vs. HDF 5 performance
- Propose use of either or both
- Develop standard naming conventions for variables (similar to ISTP program)

Conversion Software Architecture

