



# Update on GLM Cluster/Filter Algorithm Testing

Douglas Mach, UAHuntsville

Monte Bateman, USRA

GLM AWG/R3 Science Meeting 2009



# Outline

1. Status of GLM CFA
  - a. Version 3 Details
  - b. Horseshoes\_Hangrenades Routine
2. CFA Improvements
  - a. Memory Management
  - b. Error Detection
3. Proxy Data
  - a. Type Definitions
  - b. Details on GLM Proxy Data
  - c. New Proxy Data Requests
4. Future Tasks
5. Questions



# 1. Status of GLM CFA

- Version 3 code submitted to NOAA AIT on 6/30/09
- Version 4 due to NOAA AIT 2/25/10
- Version 3 currently being assimilated into NOAA AIT Framework
- Current version is able to process ~20k events/s
- Uses single processing thread (no regionalization or parallelization)

(details next set of slides)



# 1a. Version 3 CFA Details

- Version 3 code submitted to NOAA AIT on 6/30/09
  - Better memory management
  - Error detection
  - Numerous code tweaks to speed processing



# 1a. Version 3 CFA Details (cont.)

- Version 3 currently being assimilated into NOAA AIT Framework
  - Global variables to be handled by Framework, not CFA
  - Most memory allocation to be handled outside of CFA code (by Framework)
  - I/O to be handled by Framework (NetCDF format)
  - In contact with AIT folks for any integration issues



# 1a. Version 3 CFA Details (cont.)

- Current version is able to process 20k events/s
  - Reasonable datasets with data rates up to 25k events/s can be processed by the CFA within 1 s of computer time
  - Somewhat non-linear function of event rate
  - “Pending flashes/groups/events count” is key parameter



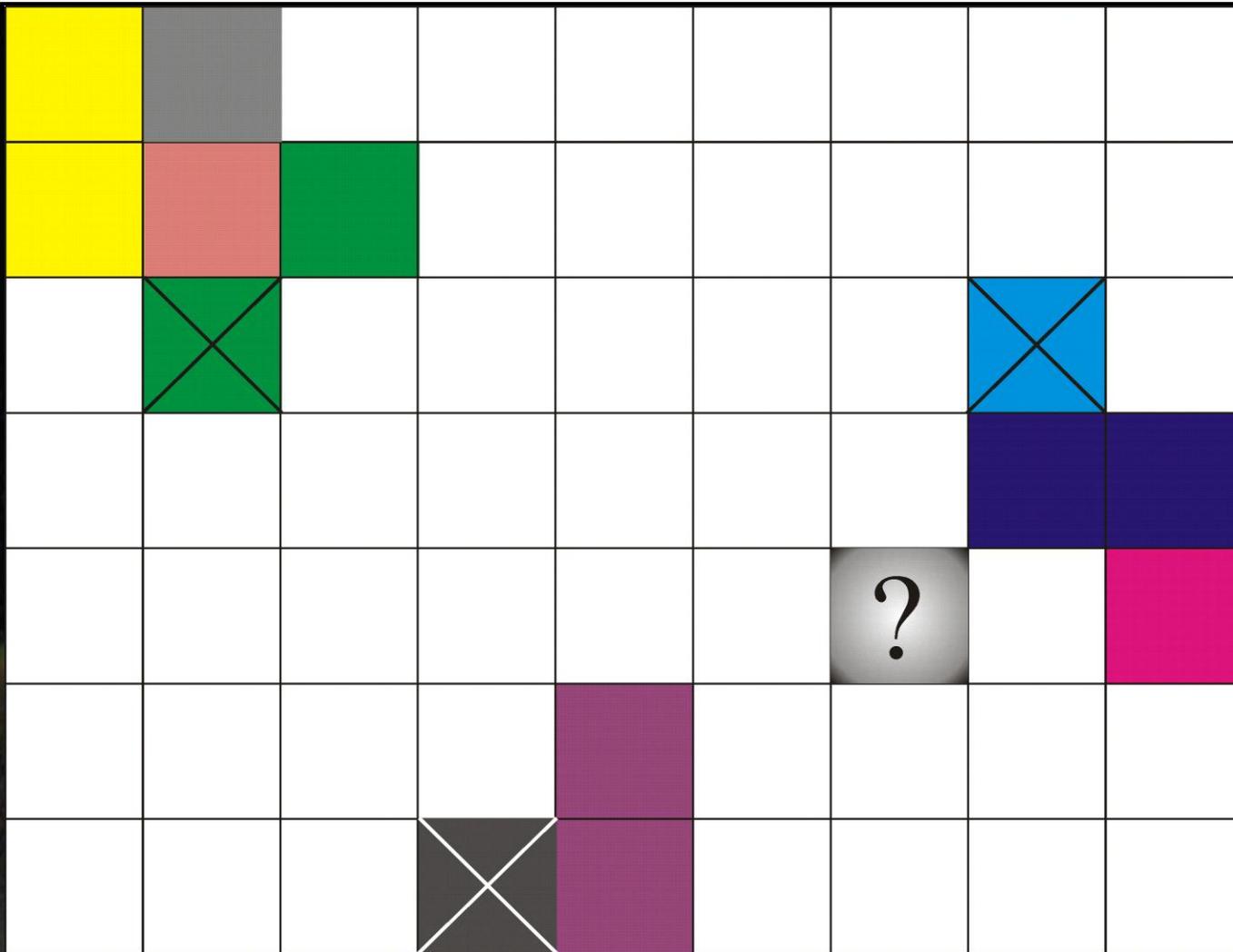
# 1a. Version 3 CFA Details (cont.)

- Uses single processing thread (no regionalization or parallelization)
  - Overhead to keep track of regionalization much greater than originally thought
    - Boundaries of regions
    - Join/split regions
    - Tracking regions
  - Found alternative method to “cheat” effects of regionalization without the overhead code
  - Horseshoes\_Handgrenades (H\_H) routine (details next)





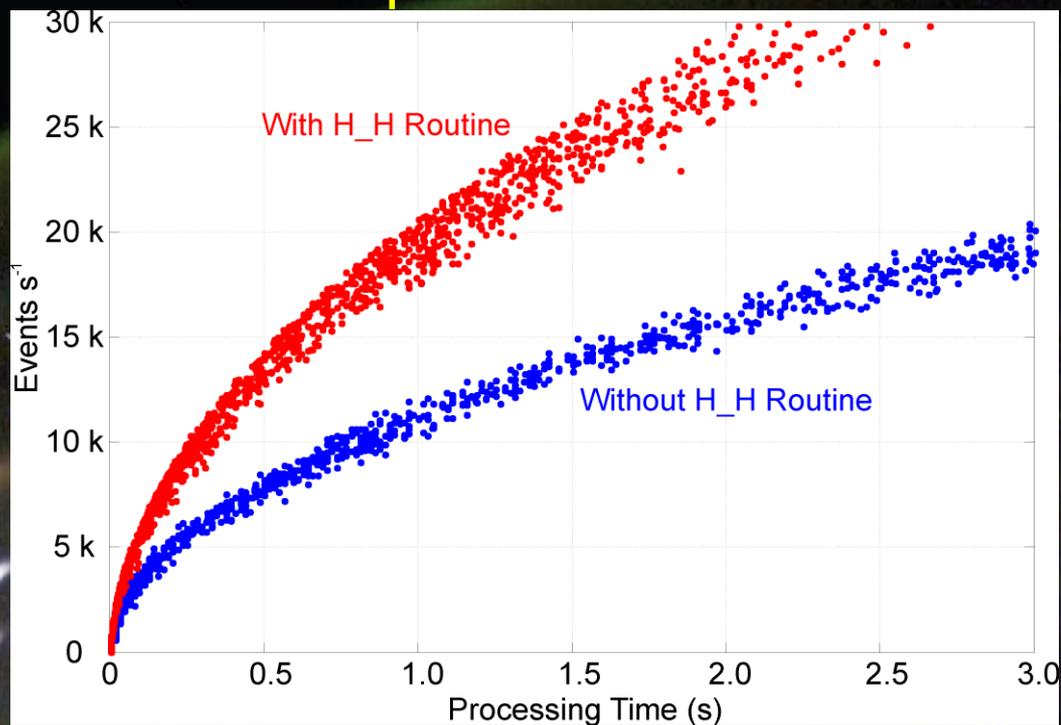
# 1b. Secret of H\_H Routine (cont.)





# 1b. Secret of H\_H Routine (cont.)

- Definition of “close” needs to be tuned
  - Too large & time savings decrease
  - Too small & large flashes could be split
  - Currently defined as 3° Lat/Lon
- Able to speed clustering without regionalization code overhead





## 2. CFA Improvements

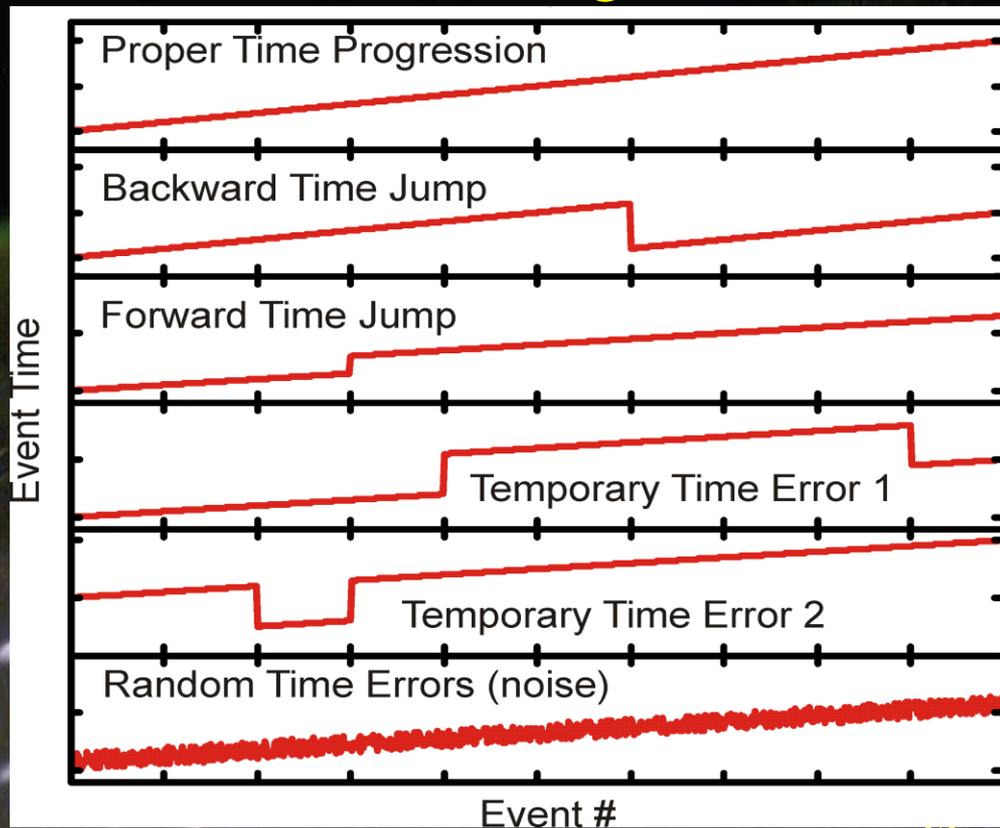
- Memory Management
  - Code allocates “typical” memory buffers for events, groups, flashes, children pointers, parent pointers, etc.
  - As needs increase, the code is able to allocate more memory for buffers (but never de-allocates)
  - Code handles inability to allocate more memory (computer error) by truncating flashes (no data loss) rather than crashing, which is what it did before
  - Need to tune the definition of “typical” for memory management
    - Use too much memory vs.
    - Spending too much time allocating new buffers (with possibility of memory leaks)





## 2. CFA Improvements (cont.)

- Time errors are handled differently
  - Time errors do not result in events being thrown out
  - Forward time jumps are “ignored”  
if large enough, more or less “flush” pending flash list
  - Backward time jumps “flushes” the pending flash list





# 3. Proxy Data

- Types of Proxy Data

- Speed Tests

- A speed test checks how many events per second the LCFA can process, and determines if the LCFA can keep up with latency requirements.

- Accuracy Tests

- An accuracy test involves first constructing (by simulation or by using a data source) an event-level dataset. Next, one clusters the event-level dataset to construct a known flash-level dataset. Finally, the event-level dataset is input to the LCFA and the LCFA output is compared to the known flash-level dataset to assess LCFA clustering accuracy. Note that the input event-level dataset has no errors (effect of input errors is examined in resiliency tests).

- Resiliency Tests

- A resiliency test examines how well the LCFA can control/minimize the deleterious effects of input errors or other peculiar/unusual input conditions (e.g., event radiance noise or location errors, corrupted data packets, extremely long-duration flashes, events near the prime-meridian or international date-line, negative event times, event times that run backwards, latitudes/longitudes that are out-of-range, etc.).



## 3. Proxy Data (cont.)

- Details on GLM Proxy Data

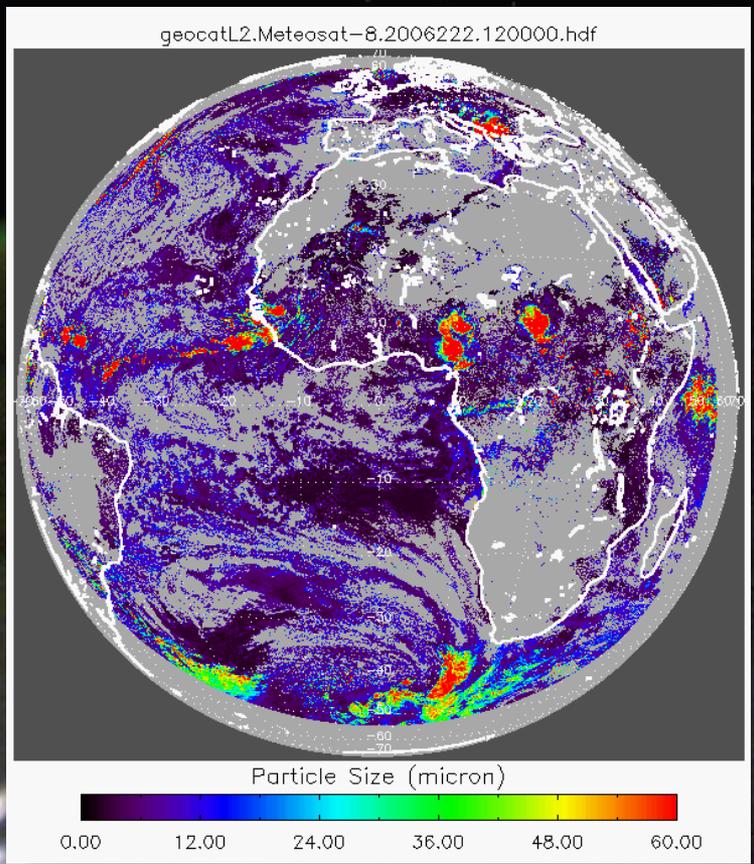
- The best proxy we have for GLM is the LIS (optical)
- LIS is a LEO sensor, so it presents a transient view of storms
- The LMA (VHF) is a 24x7 total lightning sensor
- VHF and optical instruments are sensitive to different parts of the lightning flash
- Need a mapping from LMA (24x7) to LIS (optical)
- That mapping has been developed using a coincidence study incorporating many years of LMA+LIS data
- Using this mapping, proxy data based on real LMA data have been generated and processed through the Mach's cluster algorithm.
- Results are being evaluated...



# 3. Proxy Data (cont.)

- Need new proxy data types to test code

- Large mesoscale systems with frequent, realistic lightning
  - test code's ability to separate individual lightning flashes
- “Realistic” large (full GLM FOV) datasets with “bursty” lightning
  - to test code's ability to handle and then recover from high data rate periods
- “Realistic” datasets tied to some ABI test datasets
  - so that lightning can be associated with ABI features
  - may need to be in other regions of the world





# 4. Future Tasks

- Assist with integration of V3 code into AIT Framework system
- Develop V4 code
  - Within Framework
  - Under software configuration management (Rational® ClearCase®)
  - NetCDF I/O
  - Better error handling (some error correction)
  - Optimize code for further speed improvements
- Assist in developing new proxy datasets
- Look at parallelization
  - Code has 3 “threads”
    - Input read/filter
    - Cluster/filter
    - Output write/filter
  - May not be needed



# 5. Questions?