

# The ADDE GRIB server

Scott Lindstrom

[Mr. Server]

Dave Parker

[Mr. MySQL]

Kevin Baggett

[Mr. XCD]

# The ADDE GRIB Server

- Run in real time with XCD decode
- Run with archived data
- Run with and without a database to speed execution

# ADDE GRIB Server Requirements

- Requires up-to-date gbtbpds files
  - Converts numbers in GRIB files to
    - Generating Stations (NCEP, ECMWF, ....)
    - Level
    - Parameter
      - 4-character limitations
    - etc.

# ADDE GRIB Server Requirements

- Problems:
  - GRIB files can number in the 1000s
    - Significantly slows searching times
  - Tables can become out of date
    - Values can also be defined within the GRIB file

# GRIB Server with Database

- Run in real time with XCD
- Run with archived data

# WHY?

- GRDLIST RTGRIDS/ALL DAY=#Y  
TIME=0 F HOUR=108 SRC=GFS  
LEVEL=1000 PAR=U GRIB=2526 33 94  
100 FORM=ALL took 3.5 minutes
- GRDLIST RTGRIBS/ALL DAY=#Y  
TIME=0 F HOUR=108 SRC=GFS  
LEVEL=1000 PAR=U GRIB=2526 33 94  
100 FORM=ALL took 2 seconds

# Step 0

- Have system administrator install MySQL and configure the root mysql user password
- Verify install using test scripts that come with MySQL

# The 'gribadmin' Script

- `/home/oper/bin/gribadmin`
  - Creates/removes the grib database
    - `./gribadmin makedb`
      - [enter the mysql root password] to create database
    - `./gribadmin fields`
      - [prints out the field names from the “grib” database]
    - `./gribadmin removedb`
      - [enter the mysql root password] to delete database



# What Does the Database Hold?

- Everything that is filed in the McIDAS grid header
- GRIB file name, and byte offset to grid data
  - [used to speed access]
- Extended parameter/projection names
- GRIB version (1 vs. 2)

# What Does the Database Hold?

- XCD must initially partially decode grid, but the directory server then just retrieves all the data from the database
- Get server still has to decode entire grid, but the stored GRIB file and offset speed the decoding

# What Does the Database Hold?

| grib\_type | gen\_proc\_ID | j\_day | runtime |  
forecast\_day | forecast\_hour | forecast\_time |  
geo\_ID | level\_ID | param\_ID | ptr\_to\_grid |  
param\_short | param\_long | param\_scale |  
param\_units | level | level\_scale | level\_units |  
model\_name | proj\_short | proj\_long |  
grib\_file\_name | grib\_version | grid\_size |  
num\_rows | num\_columns | num\_points | nav1 |  
nav2 | nav3 | nav4 | nav5 | nav6 | type\_mask |  
time\_ave\_diff | lev\_ave\_diff | period\_1 | period\_2 |  
modtime |

# Are There Any Issues?

- Because the number of bytes to be transmitted must be known, must know number of grids going; must know number of stitched grids, too
- Each packet of stored information: ~3500 bytes
- If you are listing ALL -- >100000 grids -- you can bump up against a resource limit if your server machine is heavily used

# Are There Any Issues?

- Comparisons with RTGRIDS datasets have been very favorable
- Access via RTGRIBS is usually faster
- Real-time testing to begin in SSEC Data Center by end of December

# Things to Consider with Archived Data

- Fully resolved pathname determined using `Mcpathname --` make sure the path is in `MCPATH` (and watch the **REDIRECTs**)
- `LD_LIBRARY_PATH` must include link to mysql libraries in `mcadde (.profile or .mcenv)`
  - `LD_LIBRARY_PATH=/usr/local/mysql/lib/mysql:$LD_LIBRARY_PATH`

# Things to Consider with Archived Data

- You can build a database to access archived data as well
- Running two databases on one machine has not yet been tested