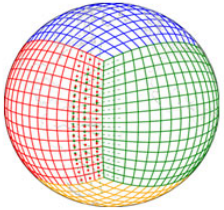
WCOSS/EMC SCIENCE QUARTERLY BRIEFING

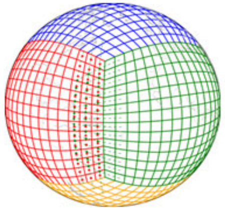
NGGPS FV3GFS

Fanglin Yang
VijayTallapragada

November 2, 2016



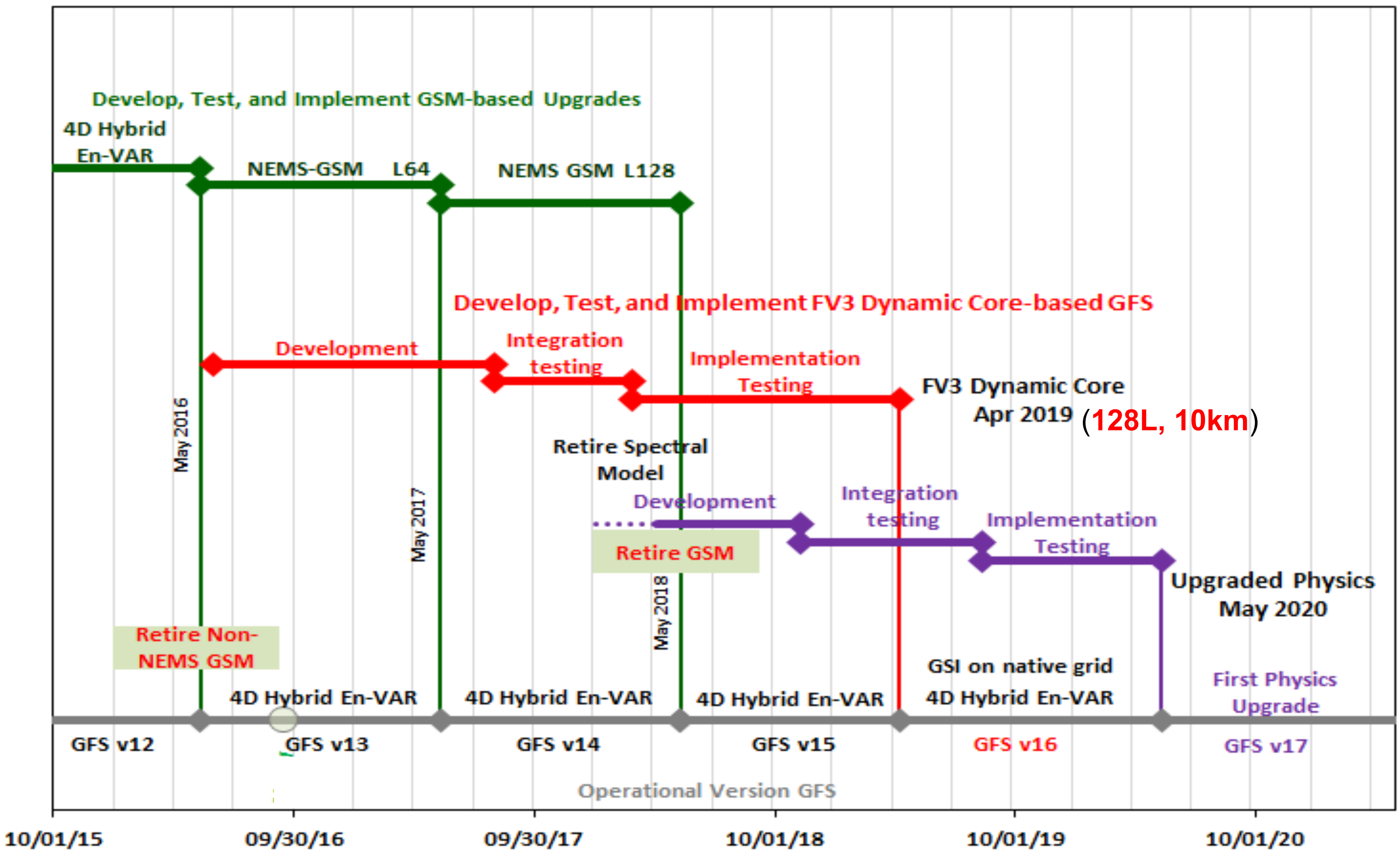
Acknowledgments: GFDL FV3 Team, Jun Wang, Hang Lei, Rahul Mahajan, John Derber, Jeffry Whitaker, Phil Pegion, George Gayno



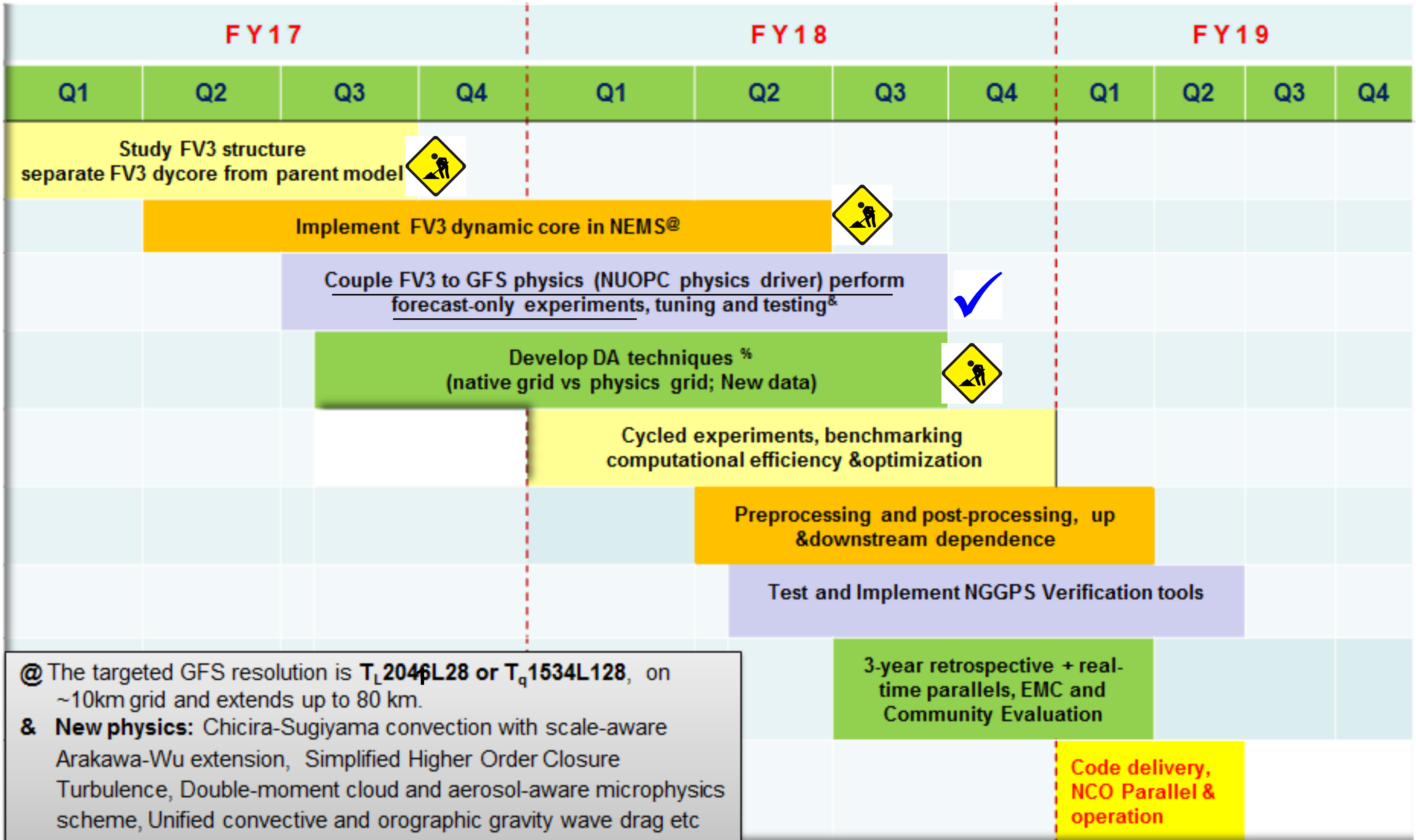
Outline

- Implementation plan of FV3
- EMC-GFDL FV3GFS technical training workshop
- Installing and running FV3GFS that was used for phase-II dycore comparison on WCOSS Cray
- Building FV3 in NEMS: Infrastructure and NUOPC Physics Driver
- Data Assimilation

GFS Upgrade Strategy FY17-FY20



Implementation Plan of FV3 Dynamic Core in NEMS GFS (FY17-FY19)



@ The targeted GFS resolution is **T_L2046L28 or T_q1534L128**, on ~10km grid and extends up to 80 km.

& **New physics:** Chicira-Sugiyama convection with scale-aware Arakawa-Wu extension, Simplified Higher Order Closure Turbulence, Double-moment cloud and aerosol-aware microphysics scheme, Unified convective and orographic gravity wave drag etc

% T_L678L128 4D-EnVAR data assimilation

Outline

- Implementation plan of FV3.
- **EMC-GFDL FV3GFS technical training workshop**
- Installing and running on FV3GFS that was used for phase-II dycore comparison on WCOSS Cray.
- Building FV3 in NEMS: Infrastructure and NUOPC Physics Driver.
- Data Assimilation.

EMC-GFDL held a joint workshop on the FV3 modeling system at NCWCP on October 18-19, 2016. More than 60 scientists joined this meeting.

GFDL scientists presented six talks on FV3 dynamical core, code structure, dynamical-physics coupling, nesting and regional application, FMS modeling environment, and supporting utilities and tools.

EMC scientists reported their progress of porting FV3GFS to WCOSS Cray, inserting FV3 into the NEMS system, and the plan of developing the data assimilation system for FV3GFS.

Both sides also discussed how to use FV3 in the EMC meso-scale convection-allowing modeling system for the 2017 NOAA Hazardous Weather Testbed Spring Forecasting Experiment.

Dr. Lin from GFDL also presented a One-NOAA Science Seminar titled An in depth look at the Finite Volume Cubed-Sphere Dynamical Core (FV3) .

The global branch is planning to hold a weekly meeting to discuss FV3 implementation technical issues with collaborators from GFDL and ESRL.

Outline

- Implementation plan of FV3.
- **EMC-GFDL FV3GFS technical training workshop**
- **Installing and running on FV3GFS that was used for phase-II dycore comparison on WCOSS Cray.**
- Building FV3 in NEMS: Infrastructure and NUOPC Physics Driver.
- Data Assimilation.

GFDL provided EMC the source code of FV3 dycore coupled with GFS physics, which was used for NGGPS Phase-2 Atmospheric Dynamic Core Evaluation.

GFDL also provided programs and scripts for preparing orography fields, creating initial conditions, and processing forecast files.

The package has been installed on WCOSS Cray. The forecast model can be run at either 32-bit or 64-bit precision.

The GFDL package has restructured to follow NCEP NCO operational standard. Scripts and utilities are developed to run FV3GFS on WCOSS Cray.

Restructured FV3GFS on WCOSS-CRAY

fix/	**C1152/ C192/ C3072/ C384/ C48/ C768/ C96/ fix_am/ parm/ topo/**
jobs/	RUN_fv3_gfs.**sh** submit_fv3.sh
lib/	build_sfcio.sh sfcio_v1.0.0/
sorc/	build_chgres.sh build_orog.sh modulefile fre-nctools/ **fv3gfs**/ orog.fd/ global_chgres.fd/ fv3nc2nemsio.fd/
ush/	**driver_fv3gfs_chgres.sh** run_chgres.sh **driver_fv3gfs_grid.sh** make_grid.sh make_orog.sh filter_topo.sh run_remap.sh create_weights.sh fv3nc2nemsio.sh
exec/	

Resolution, Physics Grid, and Run-time on Cray
10-d forecast, 6-hourly output, 3.75-minute time step
C768, 13km, 3,538,944 points

Hydro/ non-hydro	precision	threads	nodes	CPU (min/ 10day)
Non-hydro	32-bit	2	16	323
Non-hydro	64-bit	2	16	360
Non-hydro	32-bit	2	64	89
Non-hydro	64-bit	2	64	137
Non-hydro	**64-bit**	**2**	**144**	**69**
Non-hydro	**64-bit**	**4 Hyper-threading**	**64**	**135**
hydro	**64-bit**	**4 Hyper-threading**	**64**	**95**
hydro	64-bit	4 Hyper-threading	144	51

T1534 NEMS GFS (~13 km, 3072x1536), 61 nodes, 73 minutes

FV3GFS is about 1.5 ~2 times slower than NEMS GFS

Resolution, Physics Grid, and Run-time on Cray

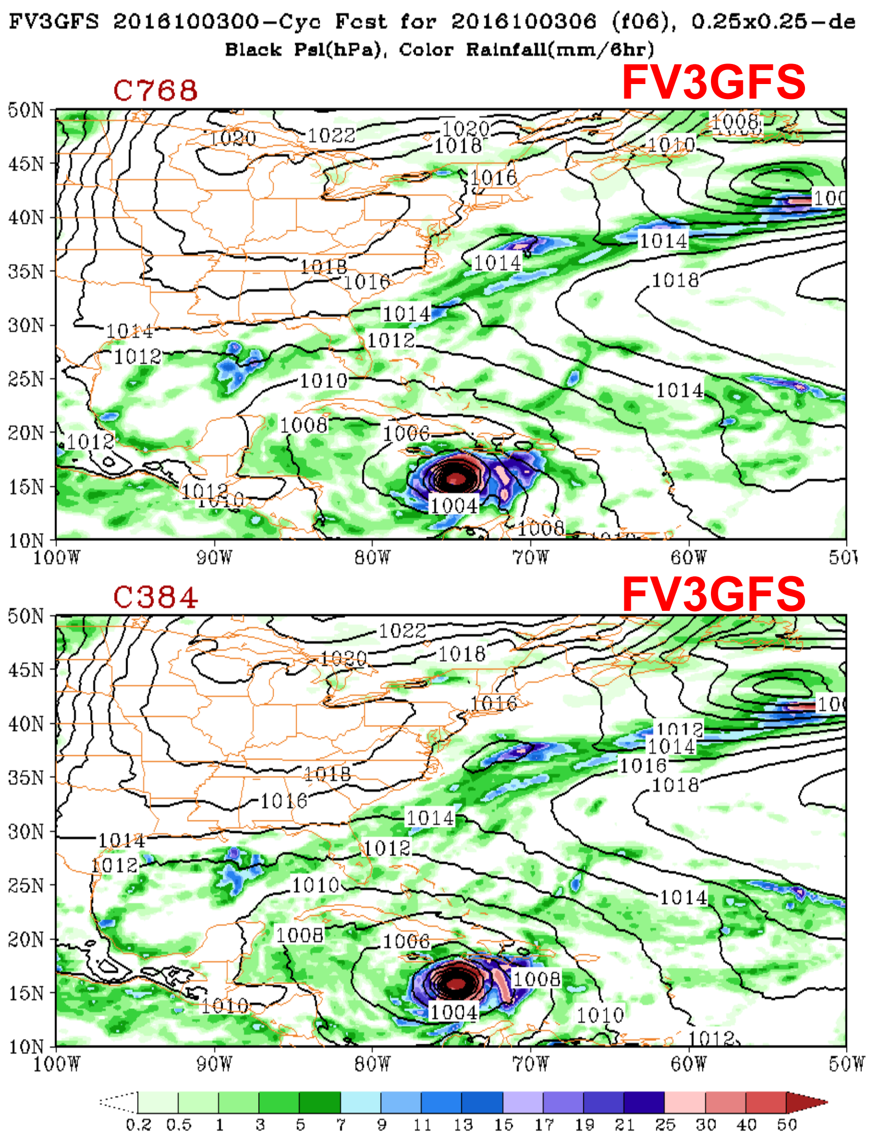
10-d forecast, 6-hourly output, non-hydrostatic, 3.75-minute time step,

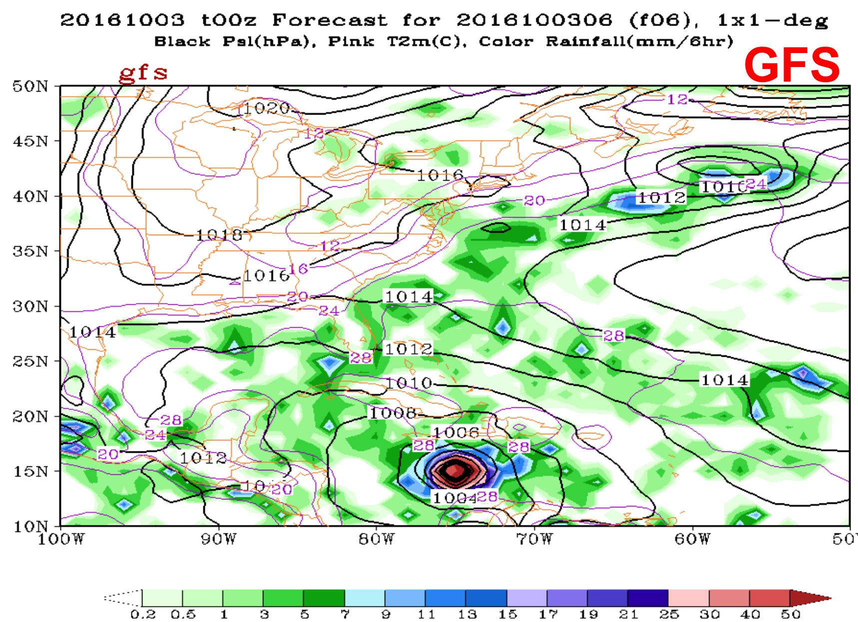
C#L63	Res	#Physics Grid (6xCxC)	Runtime (minutes) 2threads
C48	200km	13,824	**32-bit, 16 nodes: 3.6**
C96	100km	55,296	**32-bit, 16 nodes: 7.2**
C192	50km	221,184	**32-bit, 16 nodes: 21.0**
C384	25km	884,736	**32-bit, 16 nodes: 75.7**
C768	**13km**	**3,538,944**	**32-bit 16 nodes: 323**
C1152	8km	6,291,456	
C3072	3.5 km	9,830,400	

T1534 NEMS GFS (~13 km, 3072x1536), 61 nodes, 73 minutes

Thumb of rule : 10,000 / C# is grid size

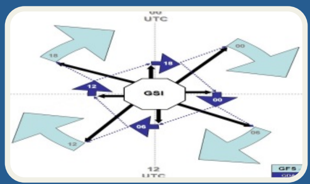
Hurricane Mathew (2016100300 IC)





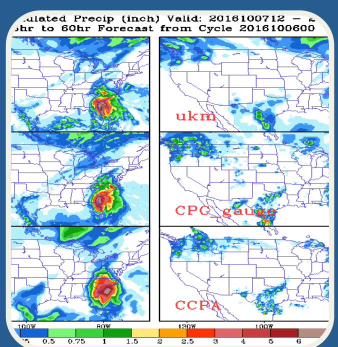
All initialized with GFS analyses

Near Term Goal: Running Real-time FV3GFS Forecast with current operational GFS initial conditions on WCOSS Cray



Workflow:

operational GFS sigio ICs → fv3 6-tile NetCDF ICs → Forecasts → convert 6-tile netCDF output to NEMSIO → run NCEP_POST for post processing → verification.



Work in progress:

- Update NCEP_POST to process FV3 NEMSIO output on lat-lon grid
- Code optimization to speed up the forecast model
- Update CHGRES to reduce memory usage for producing C1152 and C3072 initial conditions, and to convert NEMS GFS ICs to FV3GFS format.
- Set up real-time workflow scripts and cron jobs.
- Side-by-side comparison between ops GFS and FV3GFS
- To be completed in November 2016

This system can be used for NGGPS physics development before the FV3 dycore is implemented in NEMS.

fv3GFS Code Repository and SVN Trac Page

http://svnemc.ncep.noaa.gov/projects/fv3gfs

http://svnemc.ncep.noaa.gov/trac/fv3gfs

Access has been given to FV3 developers in EMC, GFDL and ESRL

Outline

- Implementation plan of FV3.
- **EMC-GFDL FV3GFS technical training workshop**
- **Installing and running on FV3GFS that was used for phase-II dycore comparison on WCOSS Cray.**
- **Building FV3 in NEMS: Infrastructure and NUOPC Physics Driver.**
- Data Assimilation.

Steps to build FV3 in NEMS

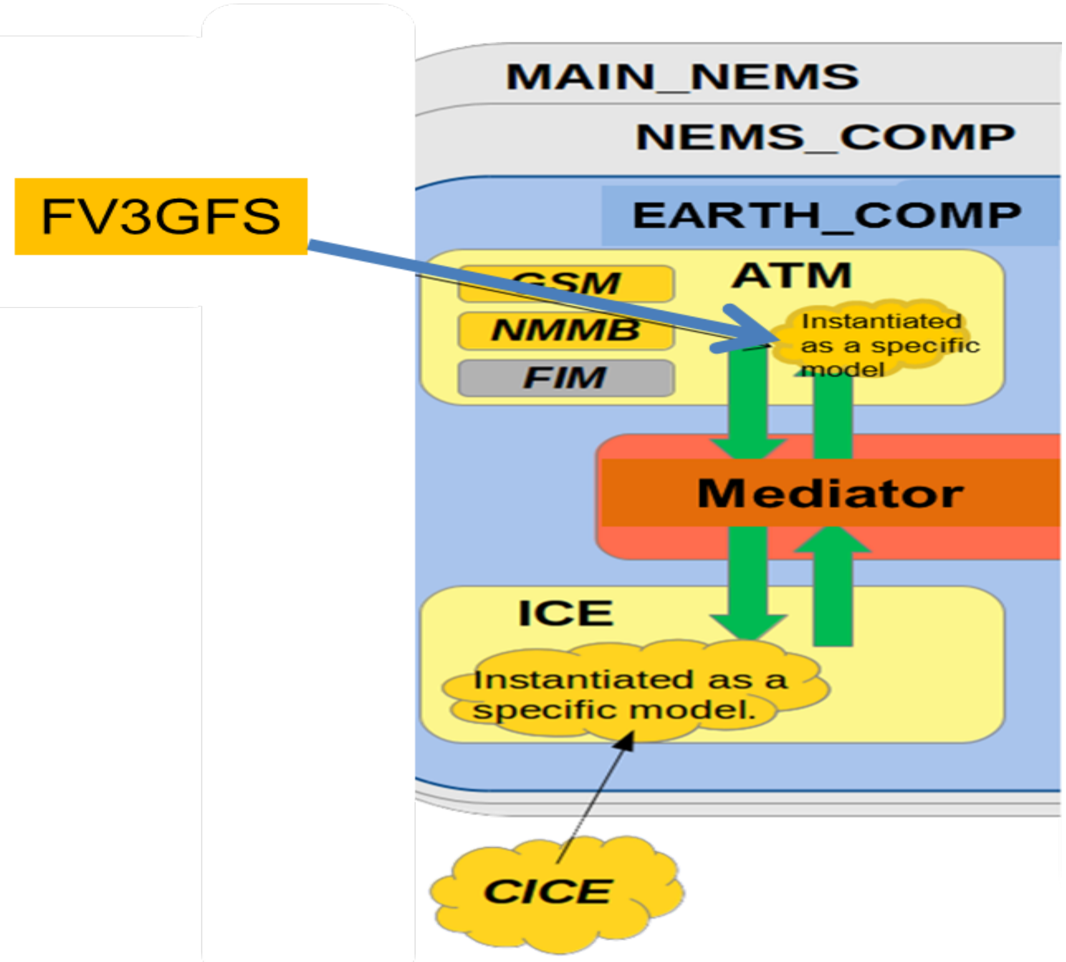
- **Build standalone fv3 as independent atmosphere grid component in NEMS, in which dyn and Phys are combined – FV3 CAP (in progress)**

- **Update physics driver – NUOPC Physics Driver Version 3 (in progress)**

- **Develop separate dyn and physics grid components**
- **Develop write grid component and inline POST for FV3**
- **Build coupled system with FV3 atmosphere grid component**
- **Set up atmosphere run phases for dynamics and physics**

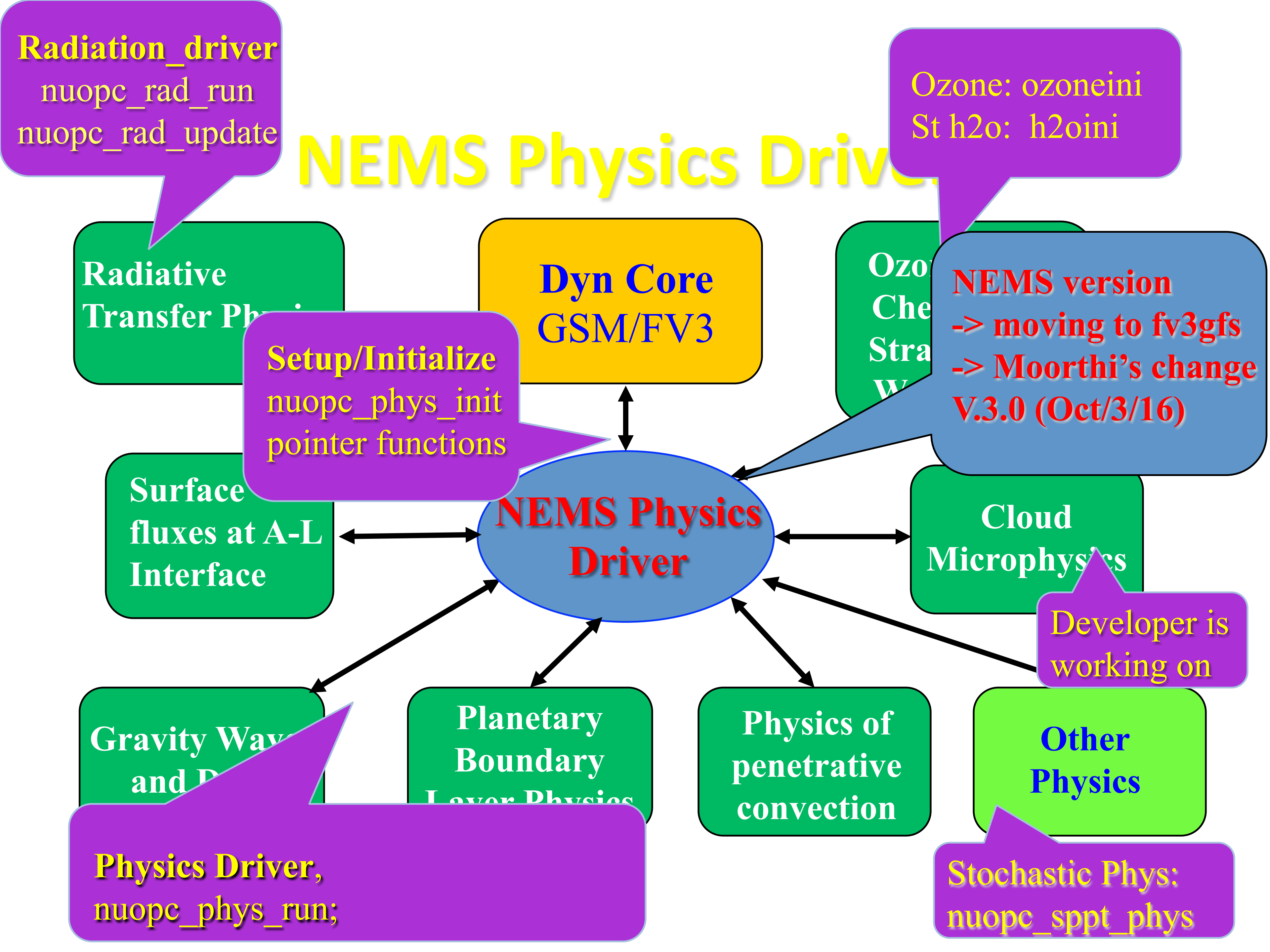
Build standalone FV3 in NEMS

- FV3GFS is implemented as an instantiation of atmosphere model
- An atmosphere CAP fv3gfs_cap.F90 will be generated and added to fv3gfs code



Write grid component and inline POST

- Write grid component will be set up for scalable IO through sets of quilt PEs
- Regridding method will be called on write grid component to output history files in grids other than fv3 native cubed-sphere grid
- Inline post will be set up on write grid component



NEMS Physics Driver
Radiation_driver
nuopc_rad_run
nuopc_rad_update
Ozone: ozoneini
St h2o: h2oini
Radiative Transfer Ph
Dyn Core
GSM/FV3
Setup/Initialize
nuopc_phys_init
pointer functions
NEMS version
-> moving to fv3gfs
-> Moorthi's change
V.3.0 (Oct/3/16)
Surface fluxes at A-L Interface
NEMS Physics Driver
Cloud Microphysics
Developer is working on
Gravity Wav and D
Planetary Boundary
Physics of penetrative convection
Other Physics
Physics Driver,
nuopc_phys_run;
Stochastic Phys:
nuopc_sppt_phys

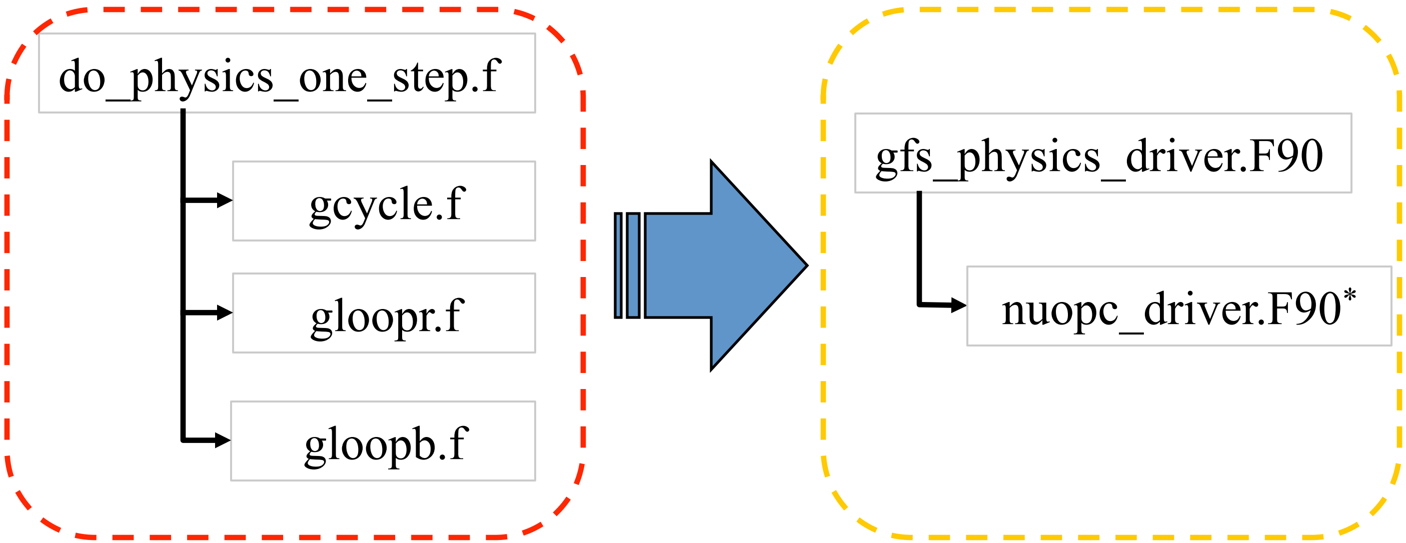
Move physics driver and physics to FV3

- Step 1: merge fv3gfs updates into physics driver 3.0

  add procedures to pre-allocate variables within physics containers

  “block” data structure to hold transform from (i,j,k) to (ix,k)

- Step2: reconnect latest physics: wrapper change.



- Step3: adjust and reorganize doxygen in the code.

  added doxygen in driver and physics with fv3gfs

* *- to be renamed to gfs_ipd_driver.F90 to reduce confusion with NUOPC caps*

Outline

- Implementation plan of FV3.
- **EMC-GFDL FV3GFS technical training workshop**
- **Installing and running on FV3GFS that was used for phase-II dycore comparison on WCOSS Cray.**
- **Building FV3 in NEMS: Infrastructure and NUOPC Physics Driver.**
- **Data Assimilation**

Requirements and Steps

- FV^3 model grid to NEMSIO lat-lon and vice-versa
- Stochastic Physics in NEMS + FV^3
- Static Background Error Covariance
- IAU functionality in FV^3 + NEMS
- Re-tuning of the hybrid DA system
- Update to *global_cycle*
- Scripting and Workflow
- Storm relocation
- Inclusion of non-hydrostatic components in the GSI
- Unified Forward Operator (UFO) and JEDI

Regridding -- Offline: *fregrid* and *global_chgres;* Online: NEMS + ESMF similar to GMAO (MAPL + ESMF)

random pattern generator -- in the GSM currently uses spherical harmonics and have spatial and temporal correlations.

- Option 1: Offline database of patterns that can be interpolated to model native grid and added during model integration
- Option 2: Separate NEMS component that would generate the patterns on the fly and utilize the ESMF regridding capability to map to model native grid. This NEMS component can be used by GSM and FV3 both.

Error Covariance --The hybrid system currently uses ***12.5%*** of static background error covariance, thus it will be necessary to build a static background error covariance model. Options:

- Use the forecast error differences from the current GFS
- Convert initial conditions from GFS into FV3, run these to generate the 48h - 24h forecast error differences, and re-create the static background error covariance.
- Use a cycling EnKF-based data assimilation system to generate a time-series of ensemble based background error covariances, and then average them over the many many samples.

IAU -- Recent research (UKMet, Env. Canada, ESRL) has shown that IAU has clear advantages in initializing the forecast model, particularly with microphysics.


Thank you

Integration of new NGGPS Dycore in NEMS: Initial Development Phase

- **Form a dedicated team of scientists at EMC to integrate the new GFDL FV3 dycore into NEMS in close collaboration with model developers (and NOAA labs)**
  - Create a specific project plan for adoption into NEMS
  - Train EMC staff on all aspects of the new system

- **Benchmark and prepare for initial implementation**
  - Configure the system to match the operational GFS configuration
  - Create and execute a test plan to reproduce operational GFS results
  - Include FY18 upgrade components and conduct parallel evaluation
  - Create FY19 implementation plan with comprehensive evaluation strategy

- **EMC global branch will maintain the operational code**
  - Master version on local repository
  - Agile development for NCEP needs

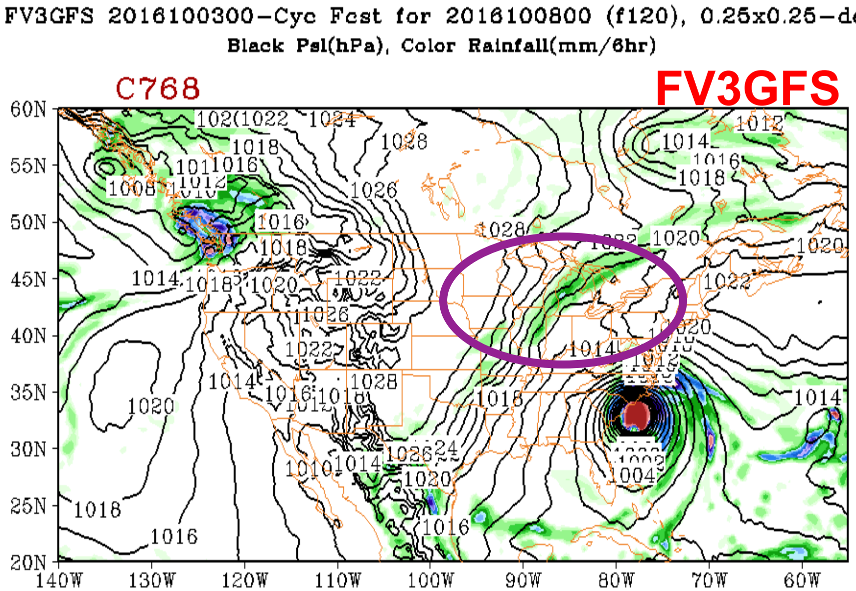
Physics: Two-Stream Strategy

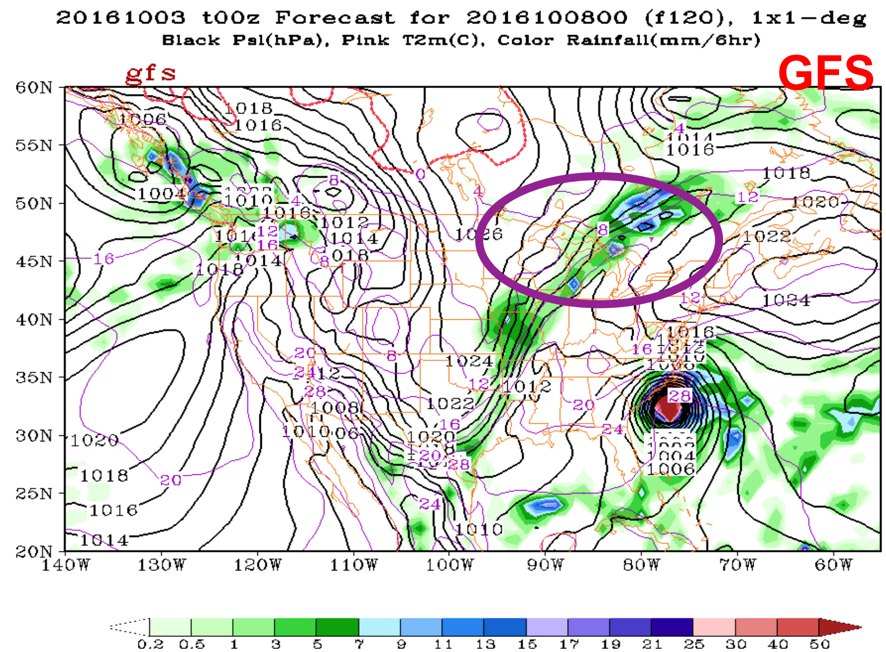
NUOPC Physics Driver in NEMS using Community Common Physics Package (EMC, GFDL, ESRL, GMTB)

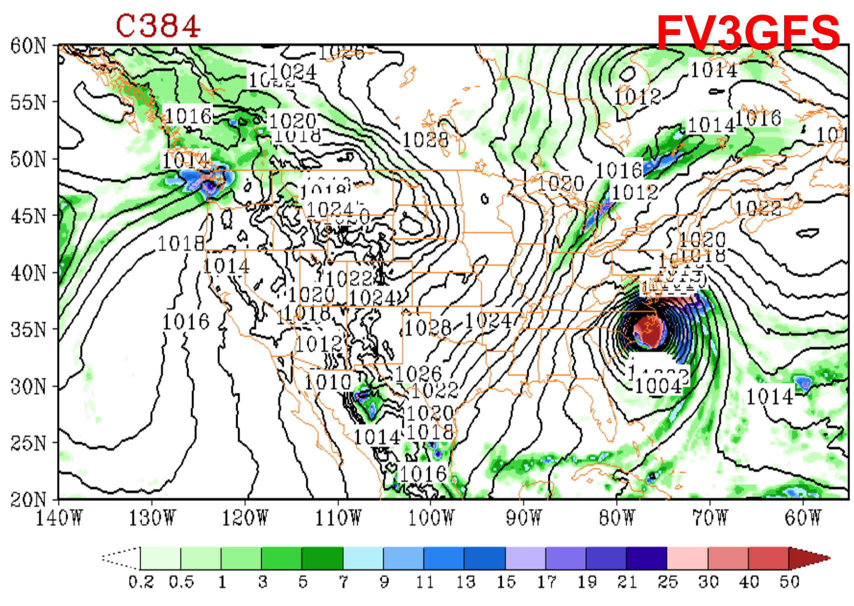
Physical Processes	Operational Physics (Evolved)	Advanced Physics* (CCPP)
Radiation	RRTMG	RRTMG (scale and aerosol aware, w/sub-grid scale clouds)
Penetrative convection and Shallow convection	SAS RAS	Scale-aware Chikira-Sugiyama & Arakawa-Wu; Grell-Freitas
Turbulent transport (PBL)	Hybrid EDMF	CS+SHOC (unified convection & turbulence)
Cloud microphysics	Zhao-Carr WSM-6	Double Moment scheme (Morrison, Thompson Barahona)
Gravity wave drag	Orographic GWD Stationary convective GWD	Unified representation of GWD
Ozone physics	NRL simplified scheme	Modified NRL scheme
Land surface model (LSM)	Noah	Noah and LIS
SST	Reynolds/RTG SST	NSST

***Includes aerosol chemistry (NGAC) module**

Hurricane Mathew, 2016100300IC, 120-hr Fcst







All initialized with GFS analyses

Packages from GFDL
Model, Compiler, Running script

- **fv3_gfsphysics.tar**
  - atmos_cubed_sphere/ avec_timer/ simple_coupler/ atmos_drivers/ gfs_physics/ shared/
- **fv3_gfs_build.tar**
  - DCMIP/ **FV3GFS/** PP/ SUPERCELL/ c2l/ **site**/

FV3GFS/:

BUILD/ **DA/** **RUN**/

BUILD/:

COMP_ALL_avx*
COMP_DEV_avx*
mk_make*
mk_makefile*
mk_paths*
nceplibs/

RUN/:

AMIP/ **PERF/ VARRES/**
RETRO/

RETRO/:

GAEA/ JET/ data_table
diag_table diag_table_no3d
field_table

RUN_fv3_gfs_c768_hyperthreads.csh

site/:

env.cshrc*
intel.mk*
intel.mk.impi*
intel.mk.mv2*
intel.mk.mv2.Jet
list_paths* **mkmf**

Packages from GFDL
Toolkit and Post-Processing

- **fv3_gfs_preproc.tar**
  - FREGRID_scripts:
    - create_weights.csh   run_remap.csh
  - GFS_tools:
    - NCEPlibs/   fre-nctools/   global_chgres/   orog/

      filter_topo/ fregrid/ make_hgrid/ make_solo_mosaic/ shared
  - IC_scripts:
    - DRIVER_CHGRES.csh   submit_chgres.csh   run_chgres.csh
    - DRIVER_GRID.csh   submit_grid.csh   make_grid.csh   filter_topo.csh make_orog.csh

      ENV.GAEA
  - input:
    - gtopo30_gg.fine   landcover30.fixed   thirty.second.antarctic.new.bin

Restructured FV3GFS on WCOSS-CRAY – fix

fix_am/ -- **CO2, aerosol, solar constant etc**

parm/ -- **data_table diag_table diag_table_no3d field_table**

topo/ -- gtopo30_gg.fine landcover30.fixed thirty.second.antarctic.new.bin

C1152/ C192/ C3072/ C384/ C48/ C768/ C96/

C768_grid.tile1.nc	**C768_grid.tile3.nc**	**C768_grid.tile5.nc**
C768_grid.tile2.nc	**C768_grid.tile4.nc**	**C768_grid.tile6.nc**
C768_oro_data.tile1.nc	**C768_oro_data.tile3.nc**	**C768_oro_data.tile5.nc**
C768_oro_data.tile2.nc	**C768_oro_data.tile4.nc**	**C768_oro_data.tile6.nc**

C768_mosaic.nc

remap_weights_C768_0p25deg.nc **remap_weights_C768_0p5deg.nc**
remap_weights_C768_1deg.nc

Samples of Initial Conditions and Restart Files

gfs_data.tile1.nc gfs_data.tile3.nc gfs_data.tile5.nc gfs_ctrl.nc
gfs_data.tile2.nc gfs_data.tile4.nc gfs_data.tile6.nc

sfc_data.tile1.nc sfc_data.tile3.nc sfc_data.tile5.nc
sfc_data.tile2.nc sfc_data.tile4.nc sfc_data.tile6.nc sfc_ctrl.nc

RESTART

coupler.res fv_core.res.nc
fv_core.res.tile1.nc fv_core.res.tile3.nc fv_core.res.tile5.nc
fv_core.res.tile2.nc fv_core.res.tile4.nc fv_core.res.tile6.nc

fv_srf_wnd.res.tile1.nc fv_srf_wnd.res.tile3.nc fv_srf_wnd.res.tile5.nc
fv_srf_wnd.res.tile2.nc fv_srf_wnd.res.tile4.nc fv_srf_wnd.res.tile6.nc

fv_tracer.res.tile1.nc fv_tracer.res.tile3.nc fv_tracer.res.tile5.nc
fv_tracer.res.tile2.nc fv_tracer.res.tile4.nc fv_tracer.res.tile6.nc

sfc_data.tile1.nc sfc_data.tile3.nc sfc_data.tile5.nc
sfc_data.tile2.nc sfc_data.tile4.nc sfc_data.tile6.nc

What's in ICs

```
ncdump -c gfs_data.tile1.nc |grep float
        float lon(lon) ;
        float lat(lat) ;
        float ps(lat, lon) ;
        float t(lev, lat, lon) ;
        float w(lev, lat, lon) ;
        float zh(levp, lat, lon) ;
        float sphum(lev, lat, lon) ;
        float o3mr(lev, lat, lon) ;
        float liq_wat(lev, lat, lon) ;
        float u_w(lev, lat, lonp) ;
        float v_w(lev, lat, lonp) ;
        float u_s(lev, latp, lon) ;
        float v_s(lev, latp, lon) ;
```

```
ncdump -c sfc_data.tile1.nc |grep float
        float lon(lon) ;
        float lat(lat) ;
        float lsoil(lsoil) ;
        float geolon(lat, lon) ;
        float geolat(lat, lon) ;
        float slmsk(lat, lon) ;
        float tsea(lat, lon) ;
        float sheleg(lat, lon) ;
        float tg3(lat, lon) ;
        float zorl(lat, lon) ;
        float alvsf(lat, lon) ;
            ……………………………..
        float shdmax(lat, lon) ;
        float slope(lat, lon) ;
        float snoalb(lat, lon) ;
        float stc(lsoil, lat, lon) ;
        float smc(lsoil, lat, lon) ;
        float slc(lsoil, lat, lon) ;
```

What's in RESTART files

```
ncdump -c fv_core.res.tile1.nc |grep float
	float xaxis_1(xaxis_1) ;
	float xaxis_2(xaxis_2) ;
	float yaxis_1(yaxis_1) ;
	float yaxis_2(yaxis_2) ;
	float zaxis_1(zaxis_1) ;
	float Time(Time) ;
	float u(Time, zaxis_1, yaxis_1, xaxis_1) ;
	float v(Time, zaxis_1, yaxis_2, xaxis_2) ;
	float W(Time, zaxis_1, yaxis_2, xaxis_1) ;
	float DZ(Time, zaxis_1, yaxis_2, xaxis_1) ;
	float T(Time, zaxis_1, yaxis_2, xaxis_1) ;
	float delp(Time, zaxis_1, yaxis_2, xaxis_1) ;
	float phis(Time, yaxis_2, xaxis_1) ;
```

```
ncdump -c fv_tracer.res.tile4.nc |grep float
…………
	float sphum(Time, zaxis_1, yaxis_1, xaxis_1) ;
	float liq_wat(Time, zaxis_1, yaxis_1, xaxis_1) ;
	float o3mr(Time, zaxis_1, yaxis_1, xaxis_1) ;
```

```
ncdump -c fv_srf_wnd.res.tile3.nc |grep float
	float u_srf(Time, yaxis_1, xaxis_1) ;
	float v_srf(Time, yaxis_1, xaxis_1) ;
```

```
ncdump -c sfc_data.tile4.nc |grep double
	double slmsk(Time, yaxis_1, xaxis_1) ;
	double tsea(Time, yaxis_1, xaxis_1) ;
	double sheleg(Time, yaxis_1, xaxis_1) ;
	double tg3(Time, yaxis_1, xaxis_1) ;
	double zorl(Time, yaxis_1, xaxis_1) ;
	double alvsf(Time, yaxis_1, xaxis_1) ;
…………
	double stc(Time, zaxis_1, yaxis_1,…) ;
	double smc(Time, zaxis_1, yaxis_1, xaxis_1) ;
	double slc(Time, zaxis_1, yaxis_1, xaxis_1) ;
	double phy_f2d(Time, zaxis_2, yaxis_1,
xaxis_1) ;
	double phy_f3d_01(Time, zaxis_3, yaxis_1,
xaxis_1) ;
	double phy_f3d_02(Time, zaxis_3, yaxis_1,
xaxis_1) ;
	double phy_f3d_03(Time, zaxis_3, yaxis_1,
xaxis_1) ;
	double phy_f3d_04(Time, zaxis_3, yaxis_1,
xaxis_1) ;
```

Output

- All in netCDF format
- Divided by tiles
- All 10-day forecasts (6-hourly) in one file

20150801000000.atmos_4xdaily.tile1.nc
20150801000000.atmos_4xdaily.tile2.nc
20150801000000.atmos_4xdaily.tile3.nc
20150801000000.atmos_4xdaily.tile4.nc
20150801000000.atmos_4xdaily.tile5.nc
20150801000000.atmos_4xdaily.tile6.nc
20150801000000.atmos_daily.tile1.nc
20150801000000.atmos_daily.tile2.nc
20150801000000.atmos_daily.tile3.nc
20150801000000.atmos_daily.tile4.nc
20150801000000.atmos_daily.tile5.nc
20150801000000.atmos_daily.tile6.nc
20150801000000.atmos_sos.tile1.nc
20150801000000.atmos_sos.tile2.nc
20150801000000.atmos_sos.tile3.nc
20150801000000.atmos_sos.tile4.nc
20150801000000.atmos_sos.tile5.nc
20150801000000.atmos_sos.tile6.nc
20150801000000.nggps2d.tile1.nc
20150801000000.nggps2d.tile2.nc
20150801000000.nggps2d.tile3.nc
20150801000000.nggps2d.tile4.nc
20150801000000.nggps2d.tile5.nc
20150801000000.nggps2d.tile6.nc
20150801000000.nggps3d_4xdaily.tile1.nc
20150801000000.nggps3d_4xdaily.tile2.nc
20150801000000.nggps3d_4xdaily.tile3.nc
20150801000000.nggps3d_4xdaily.tile4.nc
20150801000000.nggps3d_4xdaily.tile5.nc
20150801000000.nggps3d_4xdaily.tile6.nc

fv3gfs for coupled model

- **fv3gfs grid component is an instantiation of atmosphere model**
- **The import and export states of fv3gfs will be set up and filled with ESMF fields required for coupling system**
- **fv3gfs NUOPC cap will be updated to advertise and realize those fields**
- **Fv3gfs fields passed in NEMS are on cubed-sphere grid without going through exchange field. Regridding will be done on NEMS mediator/connector.**

Fv3gfs for coupled model cont.

- **For land model coupling, fv3gfs grid component run will have two phases:**
  - **Phase 1: dyn run**
  - **Phase 2: phys run**
  - **The nems application run sequence can directly call the two atmosphere run phases in order to be coupled with land model (land model is a grid component inside NEMS)**