ASGS - Building A Storm Surge Forecasting System That Saves Lives

Brett Estrade, MS
adcirc.live

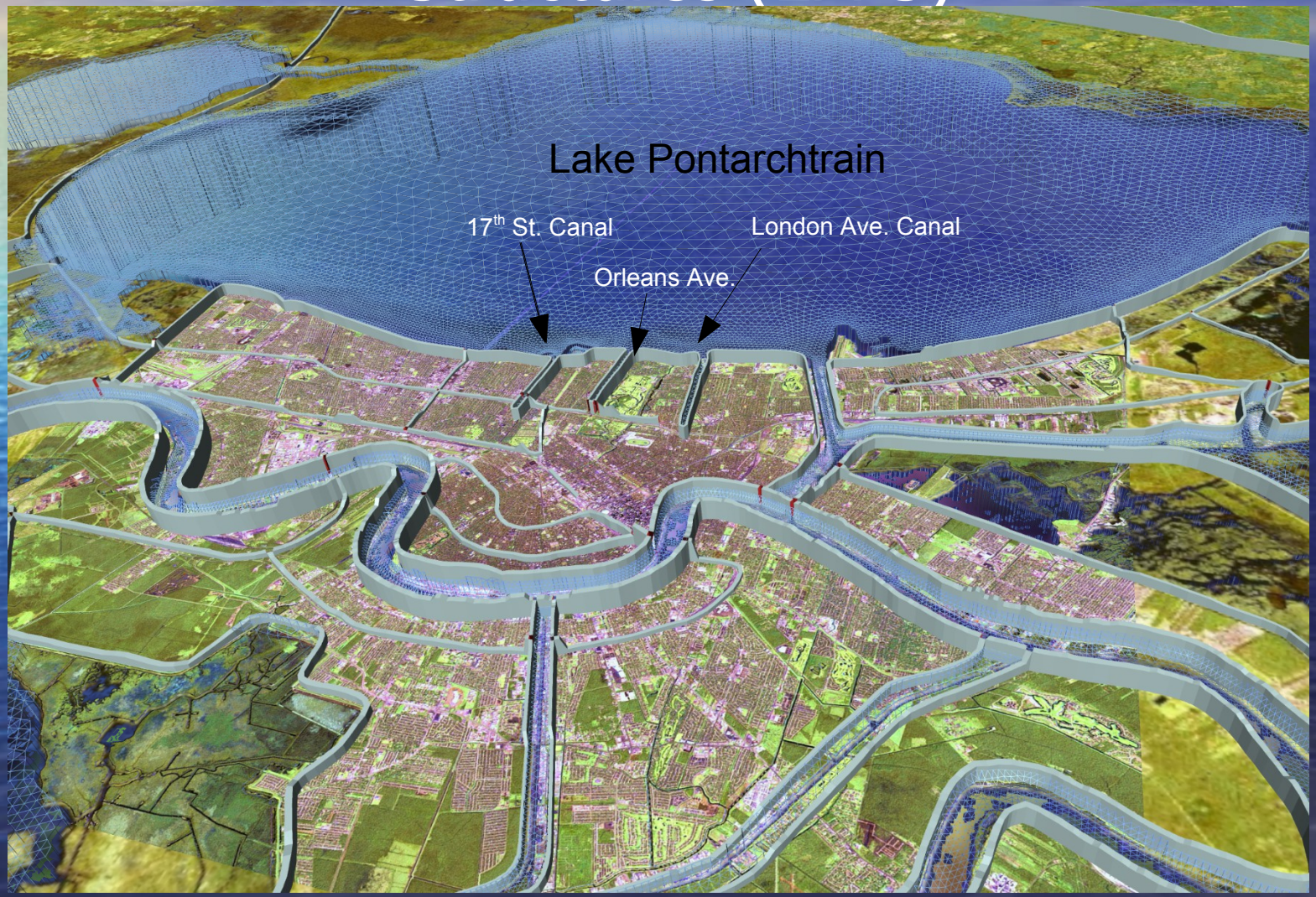
Jason G. Fleming, PhD
adcirc.live

ASGS' Guiding Principle

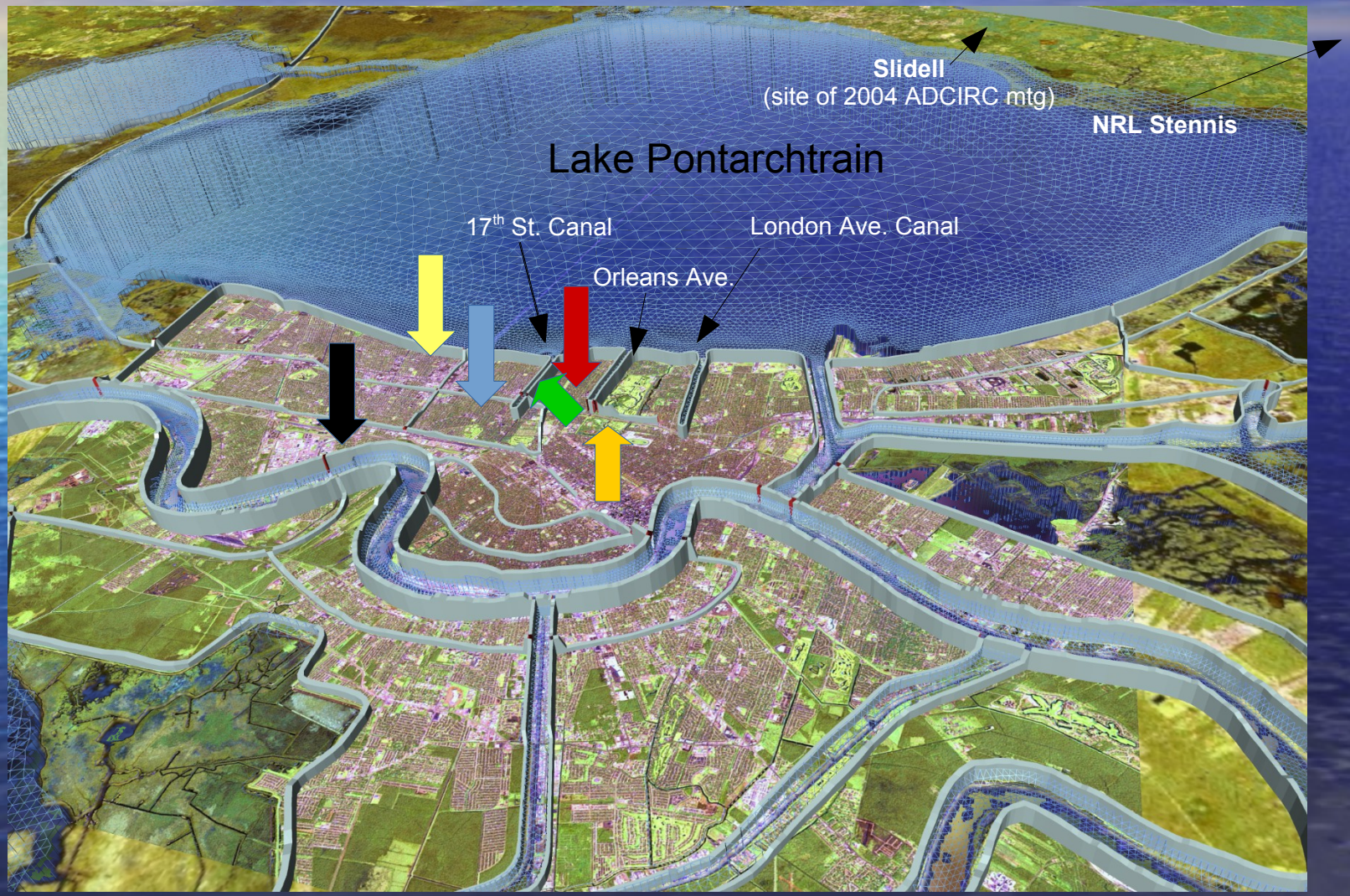
(ADCIRC+SWAN) + ASGS + CERA

- **ADCIRC** is the numerical storm surge model
- **SWAN** is a numerical wave model often coupled with ADCIRC
- **ASGS** automates ADCIRC during the storm
- **CERA** is how people view the surge forecasts online

Original Motivation: Post-Katrina, USACE Interim Gated Structures (LPFS)



Added Motivation: This is personal.

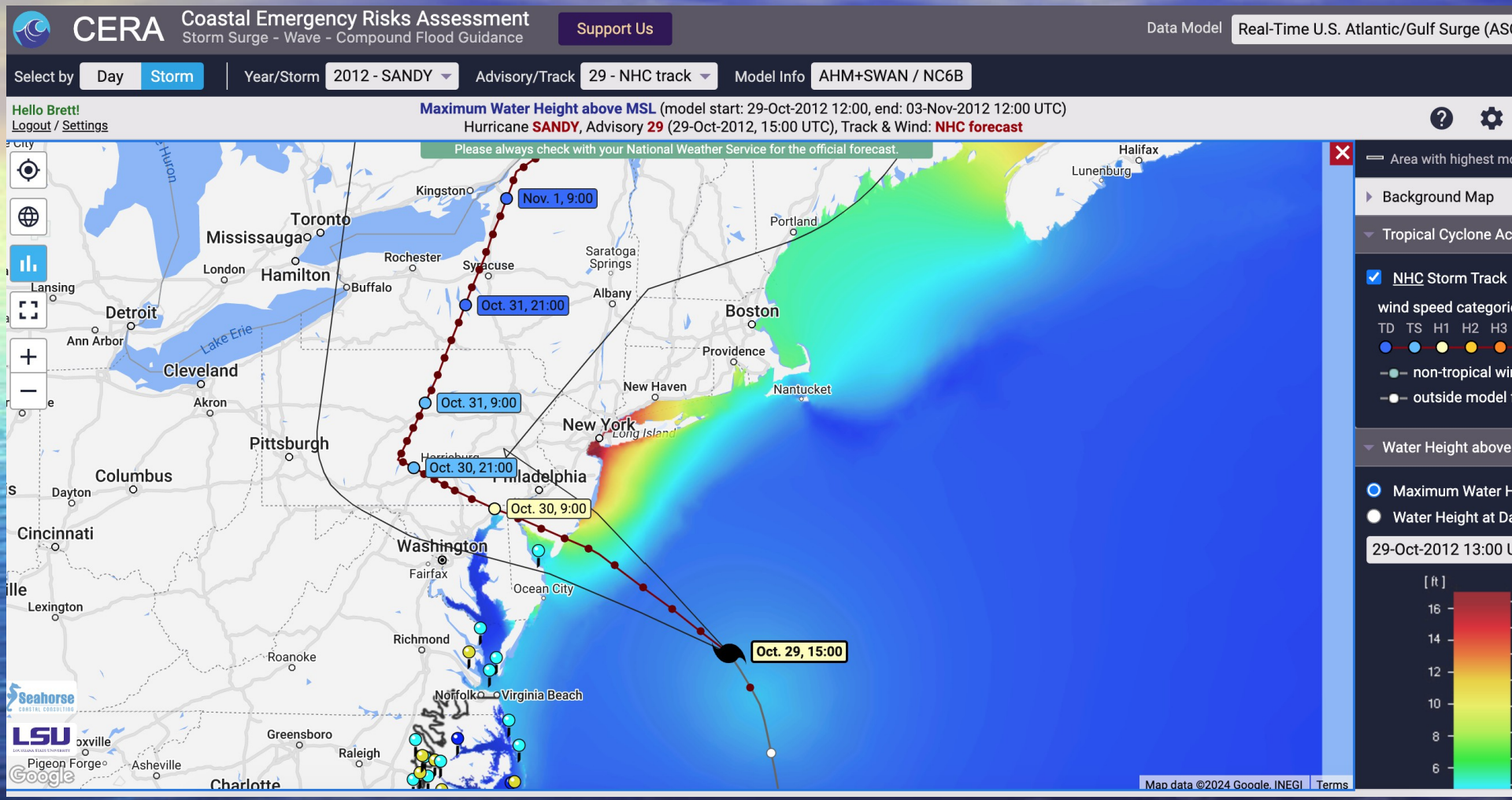


The Reality of Real Time

- Hurricanes don't have a pause button
  Guaranteed:
  - cost billions of dollars in damage
  - cause havoc on actual humans – people DO die
  - decision makers don't know what they need until they need it

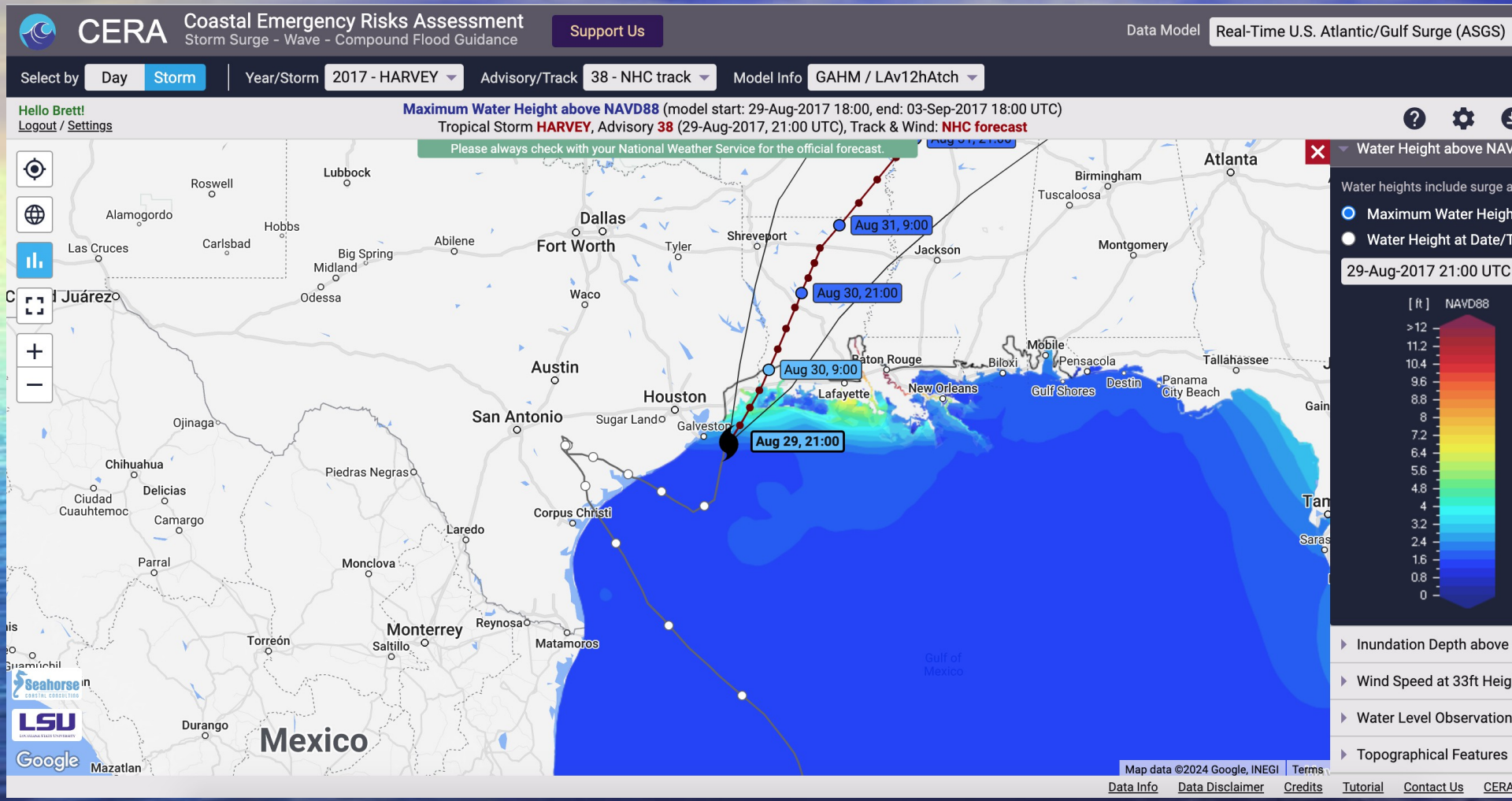
- And *they* need it 24/7, to:
  - Stage supplies and boots on the ground
  - Relocate valuable assets and command centers
  - Accurately call for targeted evacuations areas

Hurricane Sandy (2012)



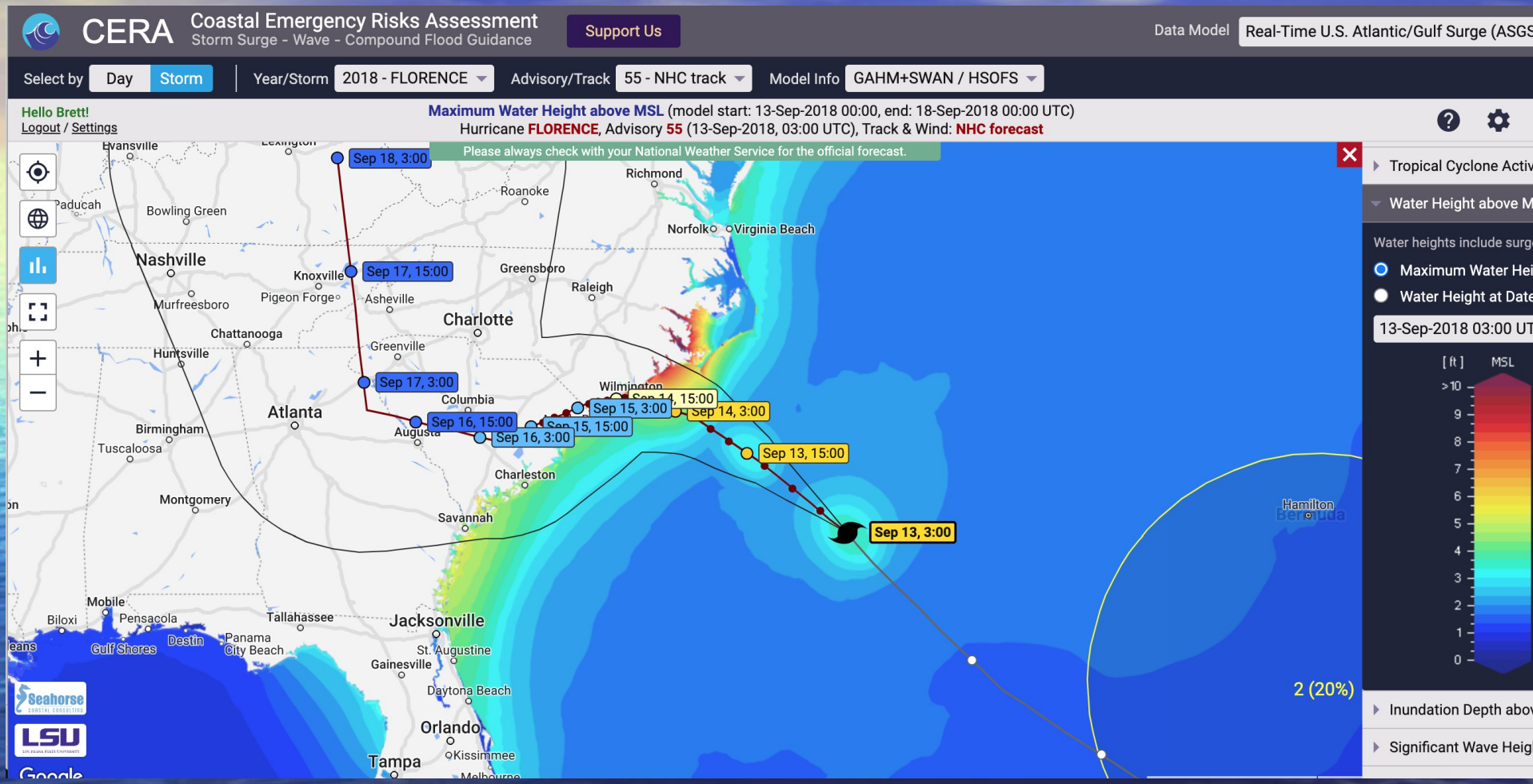
https://cera.coastalrisk.live/s/a2cf

Hurricane Harvey (2017)



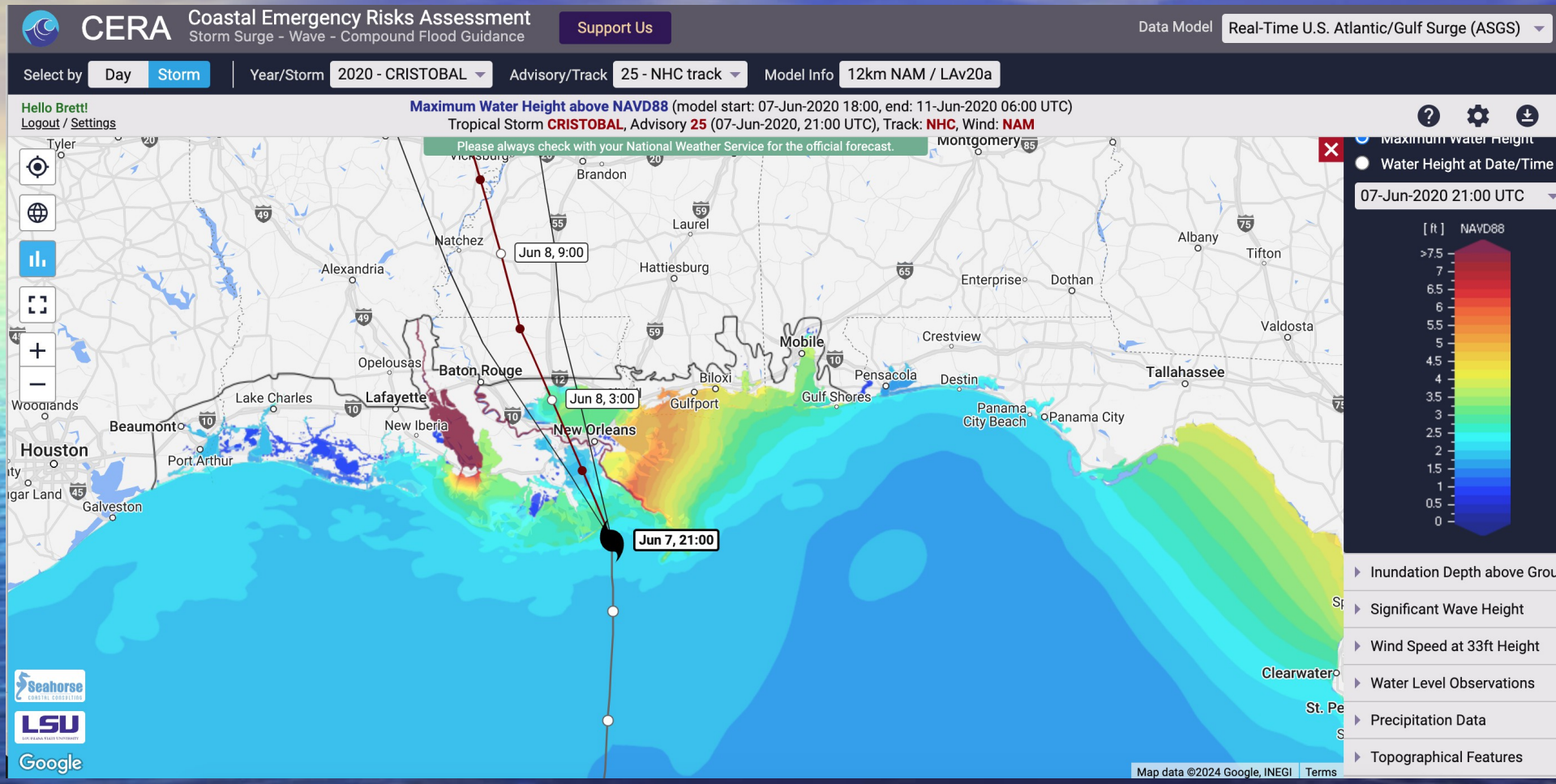
https://cera.coastalrisk.live/s/8c6b

Hurricane Florence (2018)



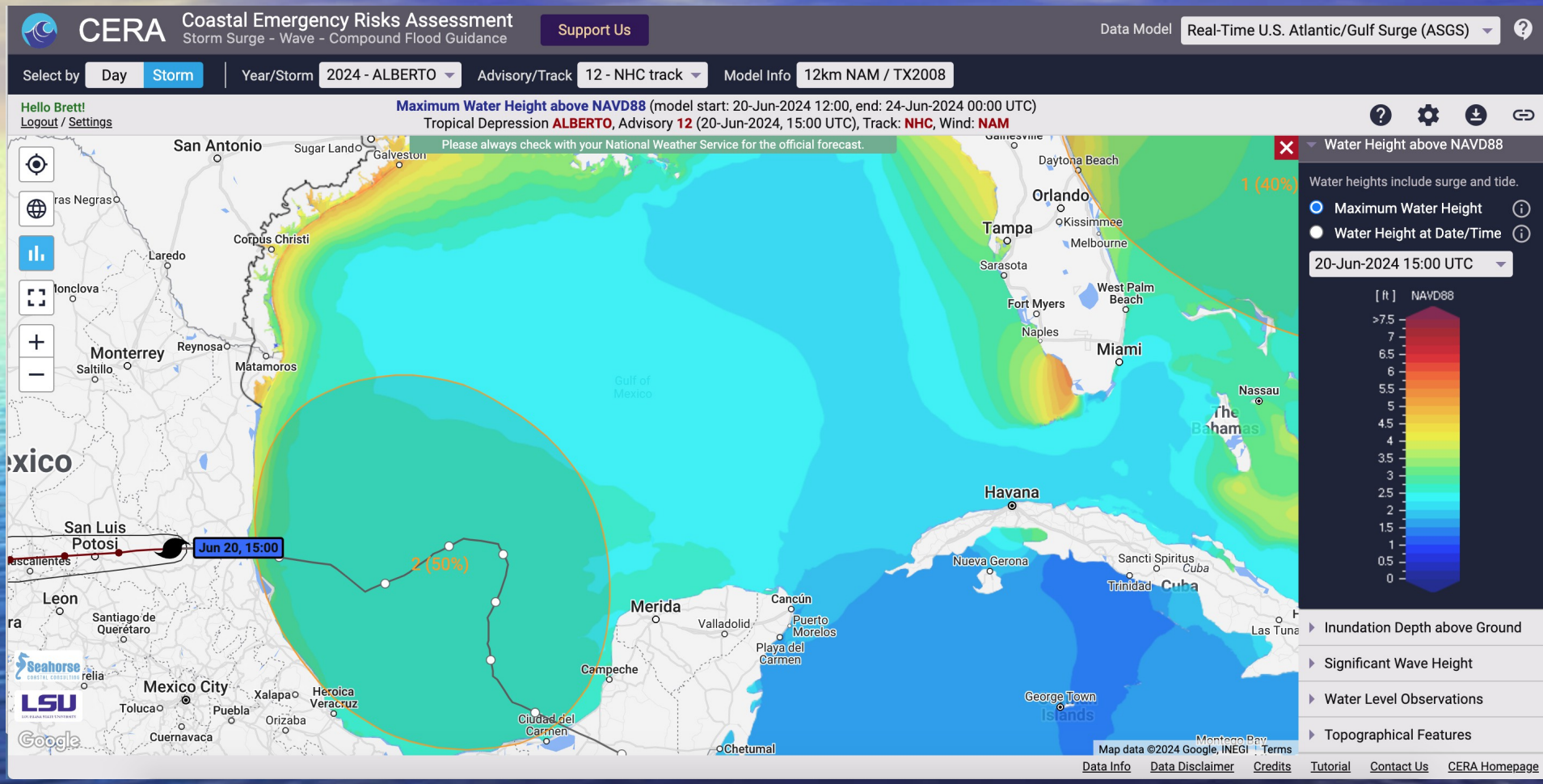
https://cera.coastalrisk.live/s/46a5

Hurricane Cristobal (2020)



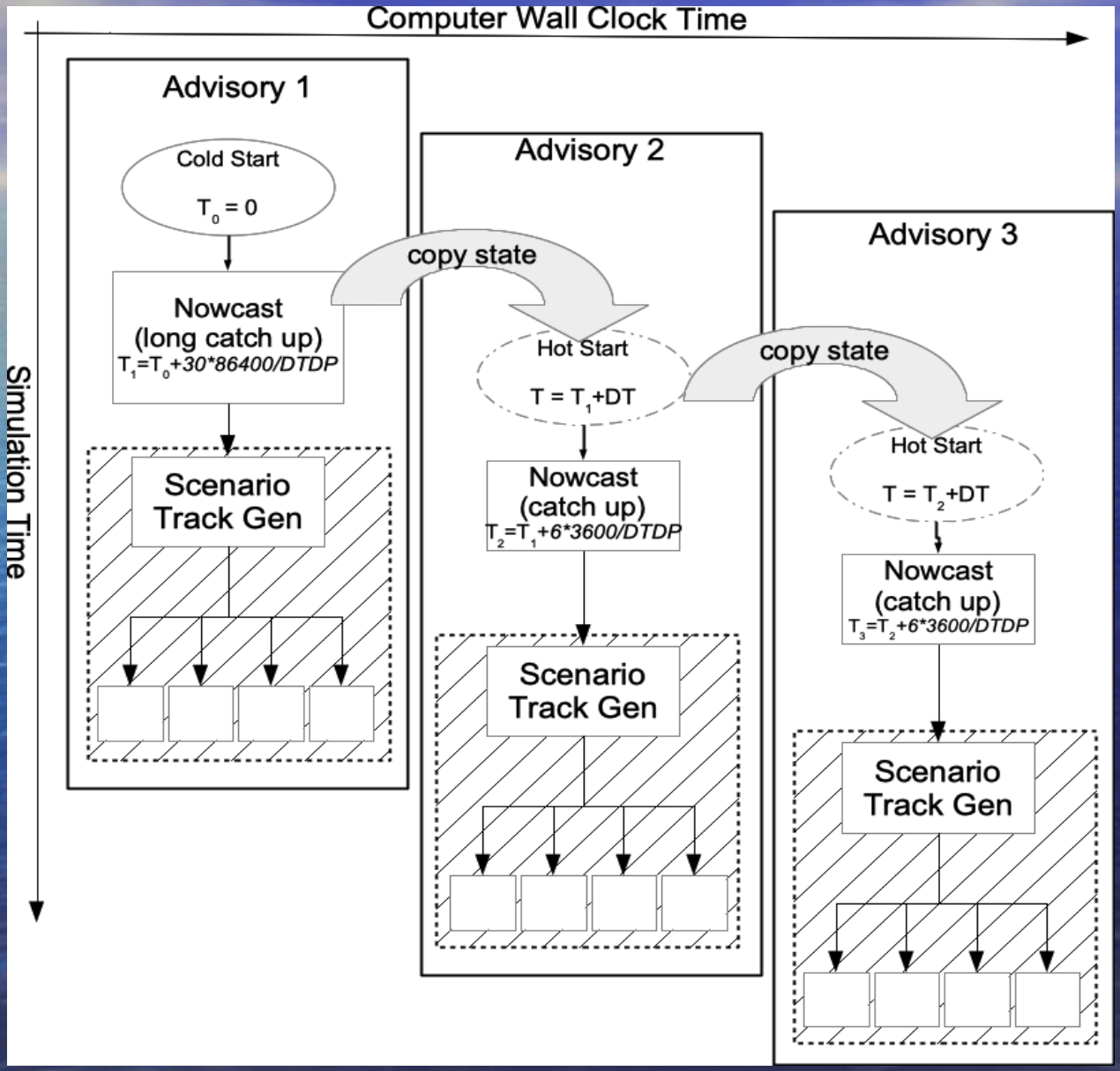
https://cera.coastalrisk.live/s/2532

Tropical Storm Alberto (2024)



https://cera.coastalrisk.live/s/7eb3

Example Continuous Workflow



More ADCIRC Automation History

- 2001 - NRL - benchmarking automation
  - **bash, Perl, Fortran**
- 2003 - NRL / OIF (operational)
  - **bash, Perl, Fortran**, Matlab
- 2005 - Katrina surprises *mostly* everyone
- 2006 - LPFS (LSU HPC/CCT demo/research project)
  - **bash, Perl, Fortran**, 5 LSU HPCs, LSU/CCT HARC scheduler (Java), SMS texts, AOL alerts, email alerts
- 2008 - Gustav (LSU HPC/LONI/CCT - operational)
  - **bash, Perl, Fortran**, 1 LSU/LONI HPC, email
- 2009 - 2024+ (operational)
  - **bash, Perl, Fortran**, redundant HPCs, email

Bash, Perl, Fortran - Why?

- **Bash** - Unix process automation, HPC (DoD) submit scripts were written in it
- **Perl** - text file munging: fort.15, queue scripts, model input and output data data, metadata
- **Fortran** - it's what the domain experts know and use, many existing utilities
- Helper OSS utils that are easy to build from source
- All of the above produce *long living* programs
- Known (by the dev/Ops), trusted, proven
- And these core technologies allow us to accomplish the operational needs that are described in these slides

Why not Others?

- Truest reason: not in the areas of competency of the primary developers & operators
- MatLab is cumbersome for automation,technicalities of licensing requirements is difficult to manage
- Octave (OSS Matlab-workalike) is also hard to add to automation
- Python requires too much maintenance (version churn, modules disappearing – e.g., Python 2.7), it's virtual environments(s) are complicated

Bash - ASGS Shell Environment

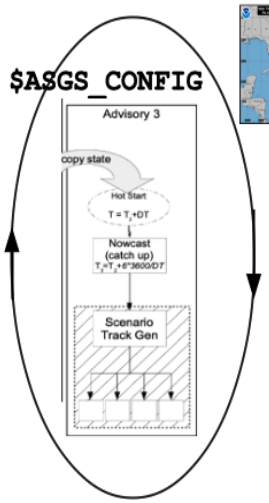
User's Login Shell Environment

ASGS Shell Environment

```
export ADCIRC_GIT_REPO="adcirc"
export ASGS_MESH_DEFAULTS="/ddnB/work/USER/QBD/asgs-daily/config/mesh_defaults.sh"
export ADCIRC_COMPILER="intel"
export ASGS_MAKEJOBS="1"
export LD_RUN_PATH="/work/USER/QBD/asgs-daily/opt/lib"
export ADCIRC_PROFILE_NAME="v53release-intel-queenbeeD"
export PERLBREW_ROOT="/work/USER/QBD/asgs-daily/opt/perl5/perlbrew"
export ASGS_META_DIR="/work/USER/QBD/asgs-daily/profiles"
export SCRIPTDIR="/ddnB/work/USER/QBD/asgs-daily"
export ASGS_INSTALL_PATH="/work/USER/QBD/asgs-daily/opt"
export LD_LIBRARY_PATH="/work/USER/QBD/asgs-daily/opt/lib:..."
export ASGS_TMPDIR="/work/USER/QBD/asgs-daily/tmp"
export PERLBREW_HOME="/work/USER/QBD/asgs-daily/opt/perl5/perlbrew"
export LD_INCLUDE_PATH="/work/USER/QBD/asgs-daily/opt/include"
export ASGS_CONFIG="/work/USER/QBD/asgs-daily/asgs-config/asgs_config_Katrina.sh"
```

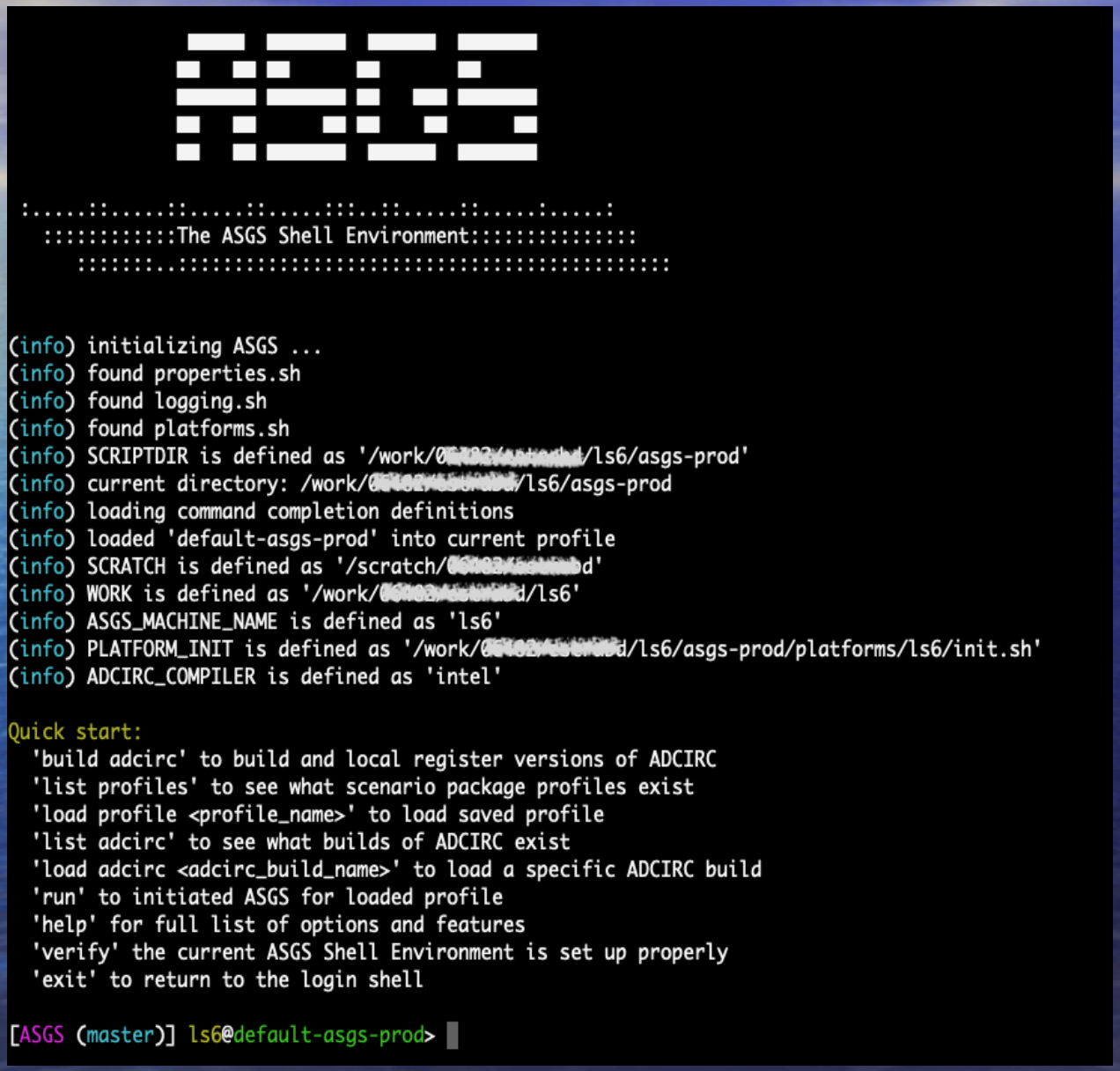
ASGS_CONFIG file

source **$ASGS_CONFIG**



./asgs_main.sh
Looping Forecast Process

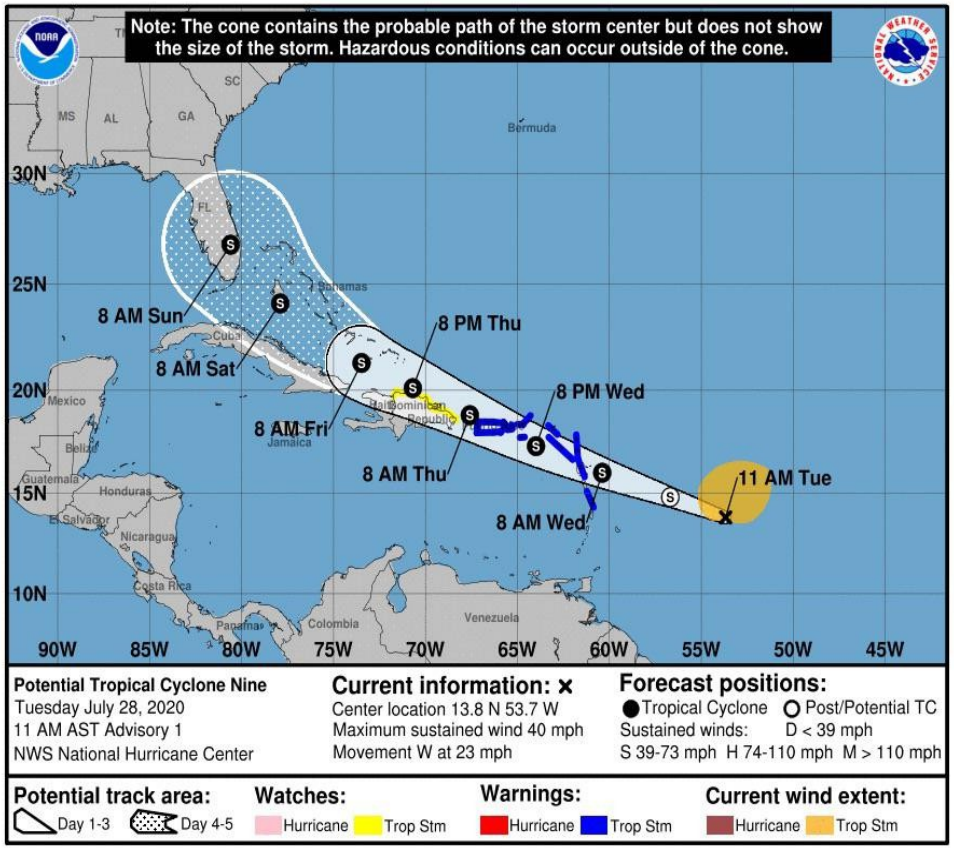
Bash - ASGS Shell Environment



Perl - Triggering on NHC Advisory

Example - Tropical Cyclone Forecast/Advisory

```
ZCZC MIATCMAT3 ALL
TTAA00 KNHC DDHHMM

HURRICANE LEE FORECAST/ADVISORY NUMBER  22
NWS NATIONAL HURRICANE CENTER MIAMI FL       AL132023
2100 UTC SUN SEP 10 2023

HURRICANE CENTER LOCATED NEAR 22.1N  61.7W AT 10/2100Z
POSITION ACCURATE WITHIN  15 NM

PRESENT MOVEMENT TOWARD THE WEST-NORTHWEST OR 300 DEGREES AT   7 KT

ESTIMATED MINIMUM CENTRAL PRESSURE  954 MB
EYE DIAMETER  20 NM
MAX SUSTAINED WINDS 105 KT WITH GUSTS TO 120 KT.
64 KT....... 40NE  35SE  30SW  40NW.
50 KT....... 90NE  70SE  50SW  80NW.
34 KT.......150NE 140SE 100SW 140NW.
12 FT SEAS..300NE 180SE 240SW 300NW.
WINDS AND SEAS VARY GREATLY IN EACH QUADRANT.  RADII IN NAUTICAL
MILES ARE THE LARGEST RADII EXPECTED ANYWHERE IN THAT QUADRANT.

REPEAT...CENTER LOCATED NEAR 22.1N  61.7W AT 10/2100Z
AT 10/1800Z CENTER WAS LOCATED NEAR 21.9N  61.4W

FORECAST VALID 11/0600Z 22.7N  62.7W
MAX WIND 115 KT...GUSTS 140 KT.
64 KT... 50NE  40SE  35SW  50NW.
50 KT... 90NE  80SE  50SW  80NW.
34 KT...150NE 140SE 100SW 140NW.

FORECAST VALID 11/1800Z 23.3N  63.9W
```



Perl - Parsing NHC Advisories,
Converting to ATCF Data

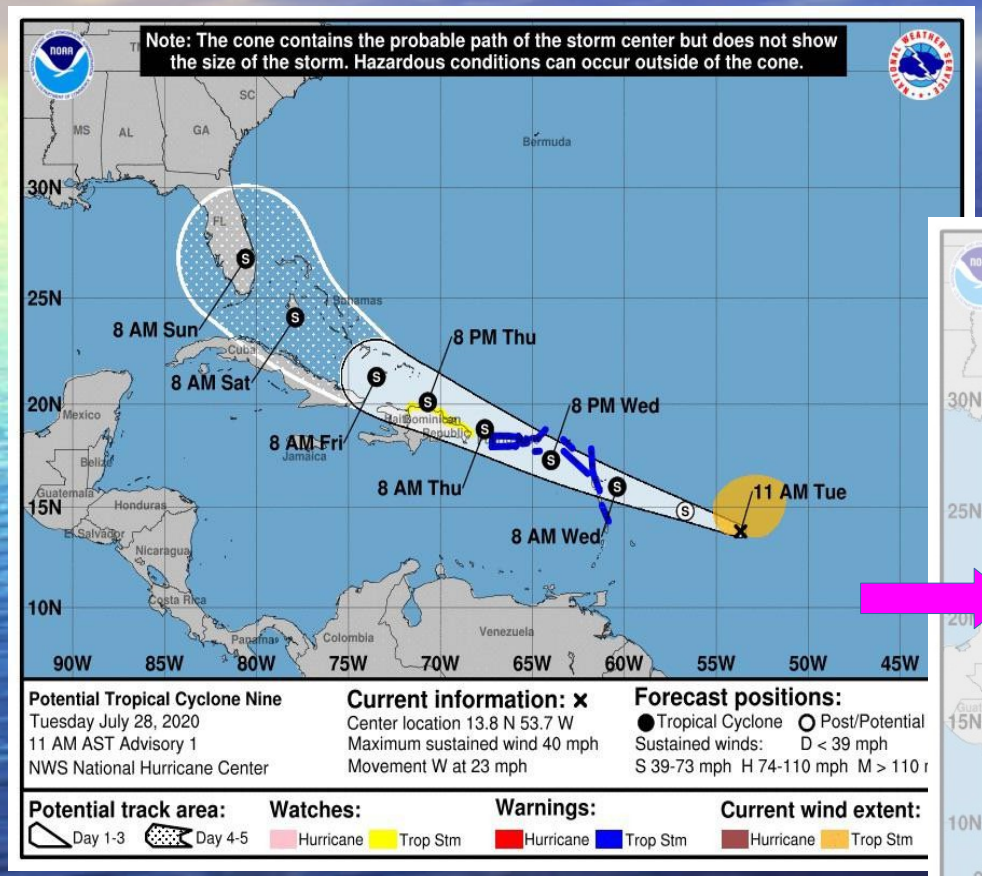
Forecasted Pressures and Center of Pressure (the "eye")

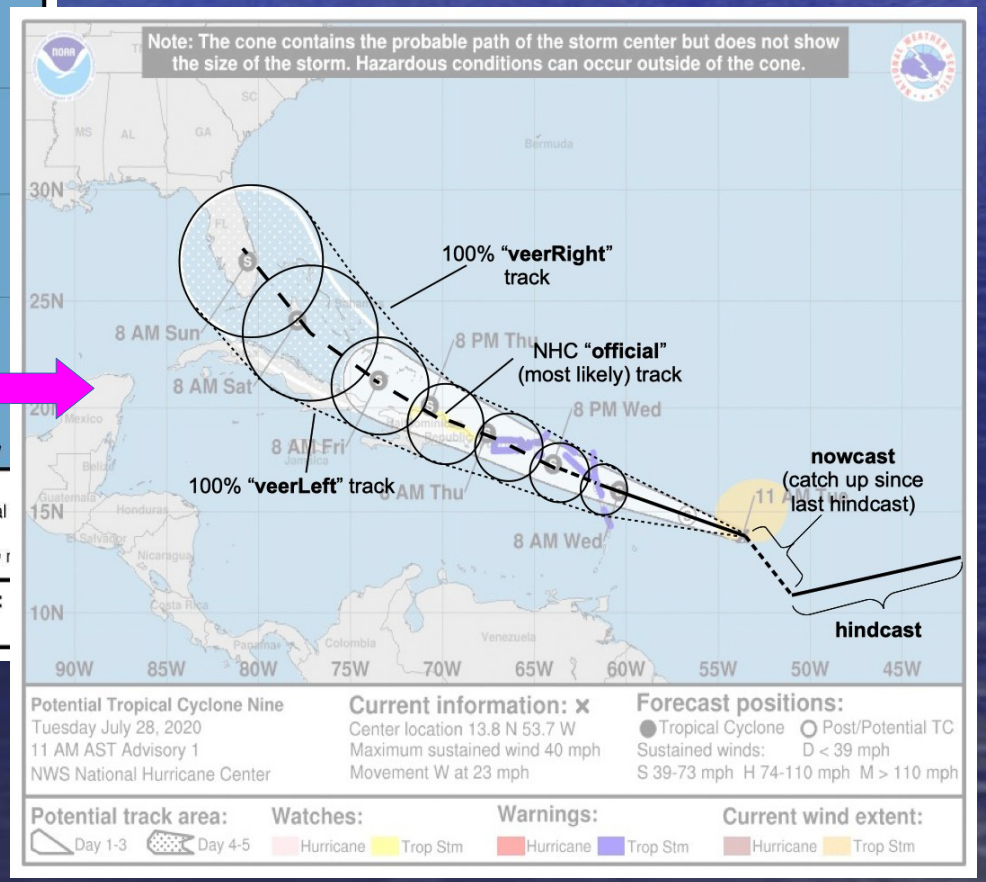
```
HURRICANE CENTER LOCATED NEAR 24.6N  83.6W AT 27/0300Z
POSITION ACCURATE WITHIN  15 NM

PRESENT MOVEMENT TOWARD THE WEST-SOUTHWEST OR 250 DEGREES AT   7 KT

ESTIMATED MINIMUM CENTRAL PRESSURE  965 MB
EYE DIAMETER  10 NM
MAX SUSTAINED WINDS  90 KT WITH GUSTS TO 110 KT.
64 KT....... 20NE  20SE  15SW  10NW.
50 KT....... 60NE  60SE  35SW  20NW.
34 KT....... 75NE  75SE  55SW  35NW.
12 FT SEAS.. 80NE  80SE  70SW 100NW.
WINDS AND SEAS VARY GREATLY IN EACH QUADRANT.  RADII IN NAUTICAL
MILES ARE THE LARGEST RADII EXPECTED ANYWHERE IN THAT QUADRANT.
```

ATCF Data Format

```
AL, 12, 2005082612,   , BEST,   0, 251N,  820W,  75,  979, HU,  34, NEQ,   75,   75,   45,   25, 1011,  150,  20,  90,   0,   L,   0,   ,   0,   0,   KATRINA, D,
AL, 12, 2005082612,   , BEST,   0, 251N,  820W,  75,  979, HU,  50, NEQ,   60,   60,   25,   20, 1011,  150,  20,  90,   0,   L,   0,   ,   0,   0,   KATRINA, D,
AL, 12, 2005082612,   , BEST,   0, 251N,  820W,  75,  979, HU,  64, NEQ,   20,   20,   10,   10, 1011,  150,  20,  90,   0,   L,   0,   ,   0,   0,   KATRINA, D,
AL, 12, 2005082618,   , BEST,   0, 249N,  826W,  85,  968, HU,  34, NEQ,   75,   75,   55,   35, 1011,  150,  15, 105,   0,   L,   0,   ,   0,   0,   KATRINA, D,
AL, 12, 2005082618,   , BEST,   0, 249N,  826W,  85,  968, HU,  50, NEQ,   60,   60,   35,   20, 1011,  150,  15, 105,   0,   L,   0,   ,   0,   0,   KATRINA, D,
AL, 12, 2005082618,   , BEST,   0, 249N,  826W,  85,  968, HU,  64, NEQ,   20,   20,   15,   10, 1011,  150,  15, 105,   0,   L,   0,   ,   0,   0,   KATRINA, D,
AL, 12, 2005082700,   , BEST,   0, 246N,  833W,  90,  959, HU,  34, NEQ,   90,   75,   75,   75, 1011,  150,  15, 105,   0,   L,   0,   ,   0,   0,   KATRINA, D,
AL, 12, 2005082700,   , BEST,   0, 246N,  833W,  90,  959, HU,  50, NEQ,   60,   60,   40,   30, 1011,  150,  15, 105,   0,   L,   0,   ,   0,   0,   KATRINA, D,
AL, 12, 2005082700,   , BEST,   0, 246N,  833W,  90,  959, HU,  64, NEQ,   25,   25,   20,   15, 1011,  150,  15, 105,   0,   L,   0,   ,   0,   0,   KATRINA, D,
```

Perl - Generating Alternate Tracks





Perl - Generating Settings Files

```
ASGS %StormName%
%EnsembleID%
1           ! NFOVER - NONFATAL ERROR OVERRIDE OPTION
0           ! NABOUT - ABREVIATED OUTPUT OPTION PARAMETER
%NSCREEN%   ! output frequency (in time steps) to STDOUT or adcirc.log
%IHOT%      ! IHOT - HOT START OPTION PARAMETER
2           ! ICS - COORDINATE SYSTEM OPTION PARAMETER
111112      ! IM - MODEL RUN TYPE: 0,10,20,30 = 2DDI, 1,11,21,31 = 3D(VS)
1           ! NOLIBF - NONLINEAR BOTTOM FRICTION OPTION
2           ! NOLIFA - OPTION TO INCLUDE FINITE AMPLITUDE TERMS
1           ! NOLICA - OPTION TO INCLUDE CONVECTIVE ACCELERATION TERMS
1           ! NOLICAT - OPTION TO CONSIDER TIME DERIVATIVE OF CONV ACC TERMS
5           ! NWP - Number of nodal attributes.
mannings_n_at_sea_floor
primitive_weighting_in_continuity_equation
sea_surface_height_above_geoid
surface_canopy_coefficient
surface_directional_effective_roughness_length
1           ! NCOR - VARIABLE CORIOLIS IN SPACE OPTION PARAMETER
1           ! NTIP - TIDAL POTENTIAL OPTION PARAMETER
%NWS%       ! NWS - WIND STRESS AND BAROMETRIC PRESSURE OPTION PARAMETER
1           ! NRAMP - RAMP FUNCTION OPTION
9.80665     ! G - ACCELERATION DUE TO GRAVITY - DETERMINES UNITS
-3.0        ! TAU0 - WEIGHTING FACTOR IN GWCE
1.0000      ! DT - TIME STEP (IN SECONDS)
0.00000     ! STATIM - STARTING SIMULATION TIME IN DAYS
0.00000     ! REFTIME - REFERENCE TIME (IN DAYS) FOR NODAL FACTORS AND EQUILIBRIUM
%WTIMINC%   ! YYYY,MM,DD,HH24,StormNumber,BLAdj
%RNDAY%     ! RNDAY - TOTAL LENGTH OF SIMULATION (IN DAYS)
5           ! DRAMP - DURATION OF RAMP FUNCTION (IN DAYS)
...
```

Perl - Queue Batch Script Generation

```
#!/bin/bash
#-------------------------------------------------------------------------------
#           Q U E U E   S Y S T E M   D I R E C T I V E S
#-------------------------------------------------------------------------------
#PBS -N %jobtype%.%scenario%
#PBS -l walltime=%walltime%
#PBS -l nodes=%nnodes%:ppn=%ppn%
#PBS -q %queuename%
#PBS -A %account%
#PBS -o %advisdir%/%scenario%/%jobtype%.out
#PBS -V
#PBS -j oe
#PBS -m a
#PBS -M %notifyuser%
#SBATCH --job-name="%jobtype%.%scenario%"
#SBATCH --time=%walltime%
#SBATCH --ntasks-per-node=%ppn%
#SBATCH --ntasks=%totalcpu%
#SBATCH --nodes=%nnodes%
#SBATCH --partition=%queuename%
#SBATCH --reservation=%reservation%
#SBATCH --constraint=%constraint%
#SBATCH --account=%account%
#SBATCH --qos=%qos%
#SBATCH --output=%advisdir%/%scenario%/%jobtype%.out
#SBATCH --mail-type=FAIL,TIME_LIMIT
#SBATCH --mail-user=%notifyuser%
echo "-------------------------------------------------------------------------"
#
#-------------------------------------------------------------------------------
#  I N I T I A L I Z E   D I R E C T O R Y   A N D   F I L E   N A M E S
#-------------------------------------------------------------------------------
THIS=%jobtype%.%queuesyslc%  # name of this script for use in log messages
SCRIPTDIR=%scriptdir%
SYSLOG=%syslog%
CYCLEDIR=%advisdir%
CYCLE=$(basename %advisdir%)
CYCLELOG=$CYCLEDIR/cycle.log
SCENARIO=%scenario%
SCENARIODIR=$CYCLEDIR/$SCENARIO
SCENARIOLOG=$SCENARIODIR/scenario.log
```

Perl - Static File Validation

```
# Fundamental

INSTANCENAME=TXLA22a_nam_be  # "name" of this ASGS process
ASGSADMIN="asgsnotify@someemail.tld"
ASGSADMIN_ID=be

# Input files and templates

GRIDNAME=TXLA22a
parameterPackage=default   # <-----<<
createWind10mLayer="yes"   # <-----<<
source $SCRIPTDIR/config/mesh_defaults.sh

#NSCREEN=-1000
# Physical forcing (defaults set in config/forcing_defaults)

TIDEFAC=on              # tide factor recalc
HINDCASTLENGTH=30.0     # length of initial hindcast, from cold (days)
BACKGROUNDMET=on        # NAM download/forcing
FORECASTCYCLE="06" # UTC time of meteorological forecast
TROPICALCYCLONE=off     # tropical cyclone forcing
STORM=09                # storm number, e.g. 05=ernesto in 2006
YEAR=2021               # year of the storm
WAVES=on                # wave forcing
REINITIALIZESWAN=no     # used to bounce the wave solution
VARFLUX=off             # variable river flux forcing
CYCLETIMELIMIT="99:00:00"

# Computational Resources (related defaults set in platforms.sh)

NCPU=959                # number of compute CPUs for all simulations
NUMWRITERS=1
NCPUCAPACITY=9999

# Post processing and publication
EMAILNOTIFY=yes
INTENDEDAUDIENCE=general    # can also be "developers-only" or "professional"
OPENDAPPOST=opendap_post2.sh
POSTPROCESS=( includeWind10m.sh createOPeNDAPFileList.sh $OPENDAPPOST )
```

Real-Time Personnel Requirements

- Reliability is everything
- Reliability derives from people
- On call support 24/7 from the key developers
- Commitment to be willing to break/fix in realtime
- Developer empathy for real time Operators
- People who are going to do what it takes to deliver

Real-Time Ops Requirements

- Software and systems that works
- People who know how to use and automate the software
- People who are willing to break/fix in realtime
- People able to engineer the entire software stack that makes it easy for the people babysitting to be successful
- People who are going to do what it takes to deliver
- The remainder of this talk is mostly about the *software*

Realtime Ops Requirements

- **Reliable** – whole stack is well understood, tested, uses reliable components
- **Flexible** – can be easily adapted in way that adds capabilities
- **Portable** – necessarily must run on many systems, from desktops to HPCs
- **Pluggable** – additional tools (or commands) are easily added
- **Scalable** – a single op can manage multiple machines and scenario packages
- **Configurable** – allows for stakeholders to update requests between NHC advisories
- **Relocatable** – redundancy is critical to reliability
- **Compatible** – tools and data follow consistent principles and should work together
- **Maintainable** – limit need for specialized knowledge, minimize setup needed for *dev*
- **Field Serviceable** – should be able to quickly diagnose and fix the issue; otherwise
- **Replaceable** – easy to rebuild, this is how *many* "weird" issues are fixed in practice
- **Enable Operator Efficiency** – …

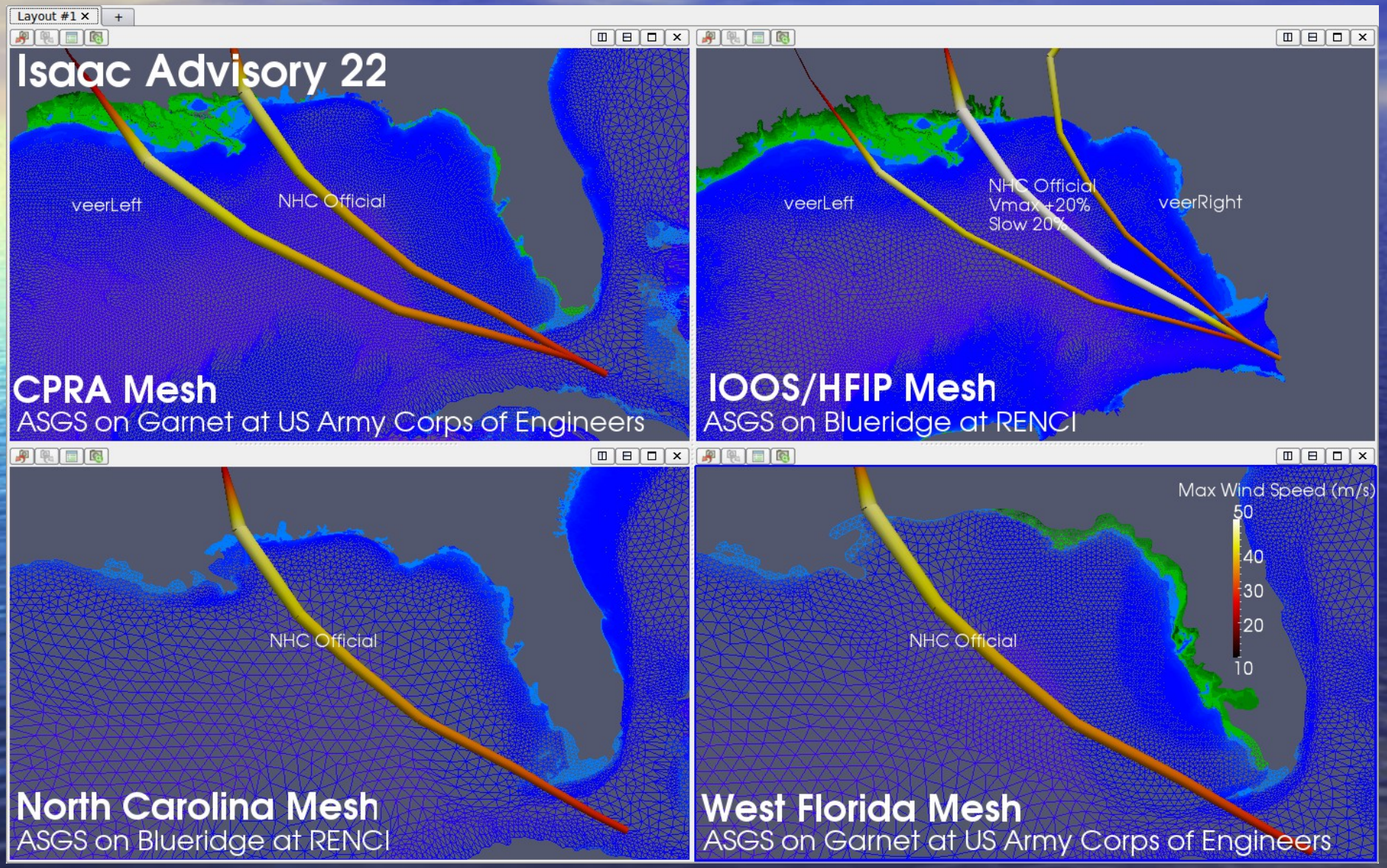
Developer Empathy for Operators

- Design decisions should minimize cognitive load on the operator
- Better still, they should be force multipliers – allow the op to do more than before *and* while thinking less in the moment
- Examples,
  - Fully automated system (re)installation
  - Built in support for multiple ADCIRC versions
  - Switch configurations on-the-fly using *profiles*
  - Share hotstarts across HPC sites, systems, and Operators
  - Minimize number of choices Operators need to make

Operator Efficiency Factors

- Scalability achieved by minimizing number of operators managing maxiumum number of
  - HPC systems
  - ASGS Installations and Instances
  - Number of scenarios

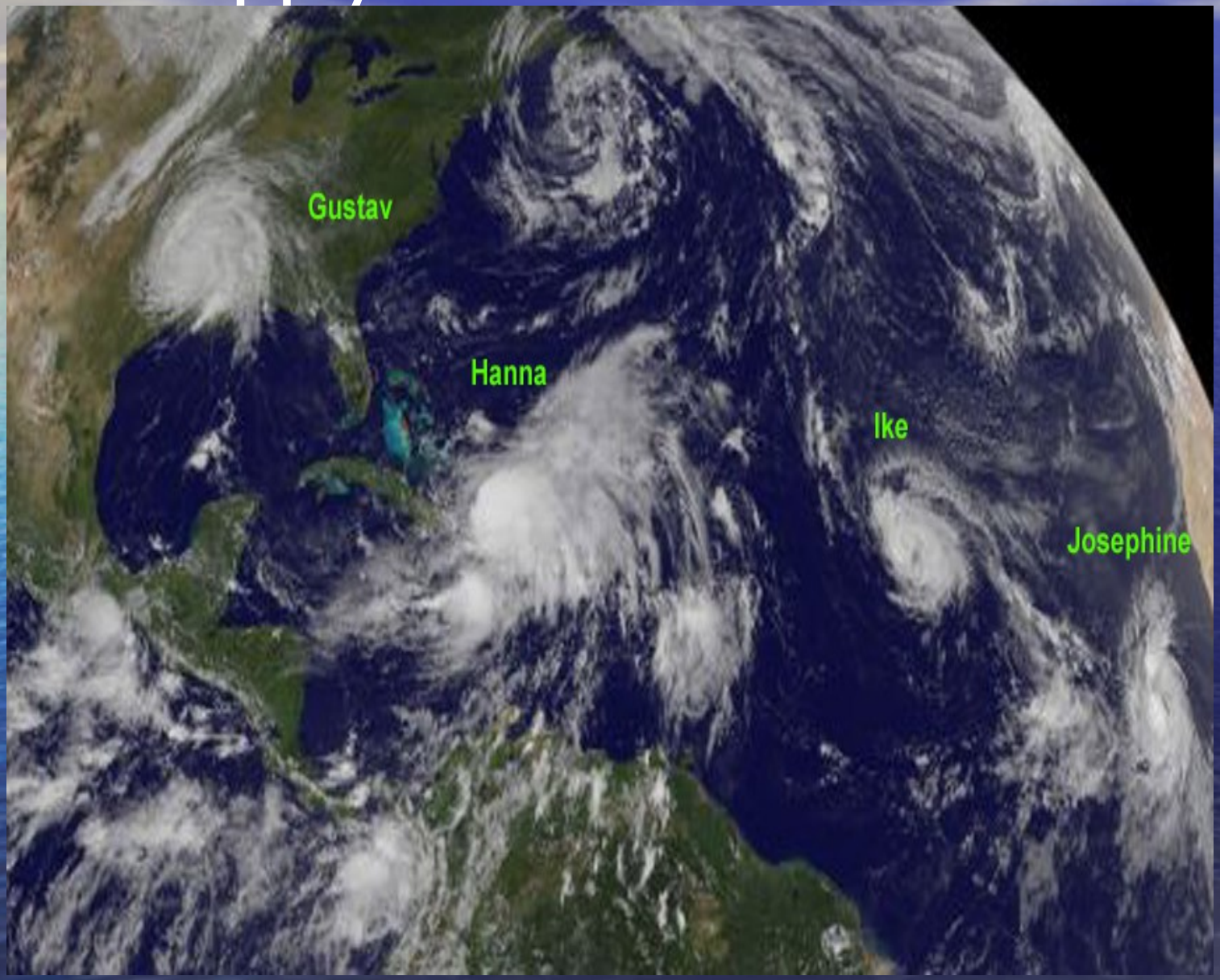
Operator Efficiency Example



Operator Scalability

- 1 Experienced Operator can
  - Can deploy on 5-6 HPC (LSU, TACC)
  - each with multiple run profiles
  - each run profile with many scenarios, configs updatable before the next forecast
  - deliver to multiple remote targets (ssh)
  - with the ability to run nearly hands-free
  - able to be updated quickly to deal with upstream data and system problems
- Now add another operator, and they can handle MANY simultaneous instances on MANY systems

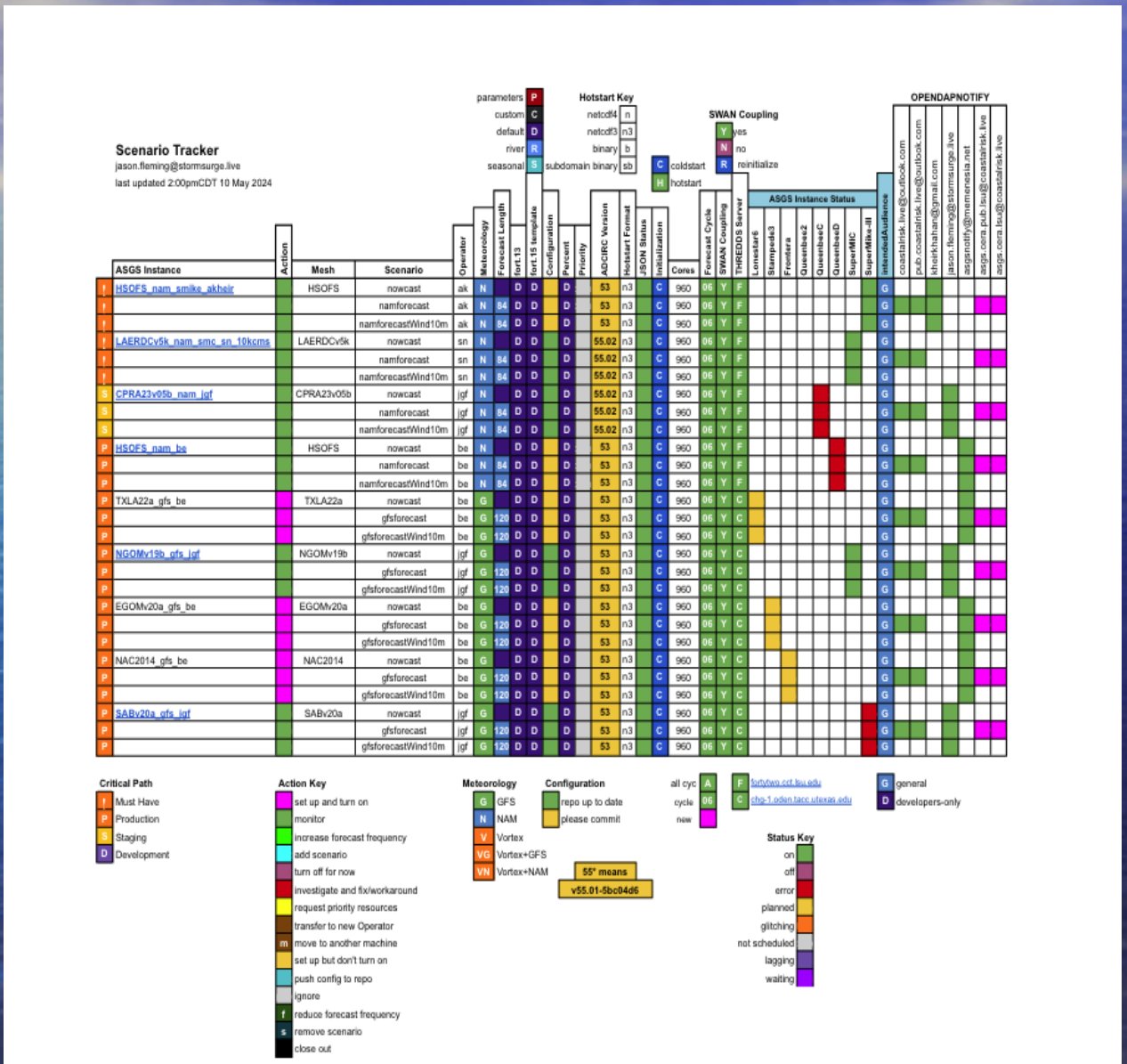
Supply Creates Demand



QB & Battle Rhythm

- The QB executes the game plan, which is centered around NHC forecasts -
- Warm Up,
  - NHC lights an area of interest
  - QB says, get to your battle stations and sends out a preliminary plan or "scenario package" (which could be *meteorological* based)
- QB is in comms with stakeholder, who asks for an initial deliverable
- Things go hot when NHC starts issuing advisories
  - QB sends out updated scenario package, head of next advisory
  - QB in comms with all ops about their status
  - QB meets with stakeholders, answers questions and adjusts scenario packages as needed
  - Repeat until landfall
- Hindcasts, Postmortem

The Battle Plan



Bash is an Essential Tool

Why?

- Stable, long living scripts
- Integration with OS shell environment
- Integration into OS process management
- Rapid *prototyping* is actually rapid *doing*
- Iterative refactoring is always occurring
- Easy system debugging

Perl is an Essential Tool

Why?

- Stable, long living scripts
- +Text processing capabilities
- +Elegant, efficient syntax gets out of the for quickly
- Integration into OS process management
- +Multi/mixed paradigm is very powerful
- Rapid *prototyping* is actually rapid *doing*
- Iterative refactoring is always occurring
- Easy system debugging

Thank You

BINARY.NET

ADCIRC Live

Professional training, paid support, online tools, and ADCIRC data.

CERA Coastal Emergency Risks Assessment

Storm Surge - Wave - Compound Flood Guidance

Seahorse COASTAL CONSULTING

LSU High Performance Computing

Louisiana State University

Coastal Computing Services, LLC

LONI