# Clouds in the Cloud*

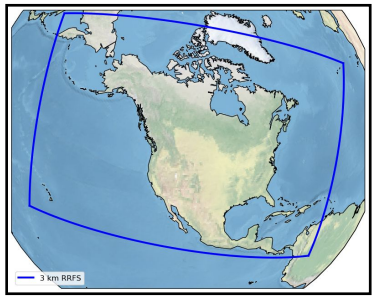
## Developing NOAA's Next Generation Convection-Allowing Prediction System with Cloud HPC

**Jacob R. Carley**[1], Raj Panda[1,2], James Abeles[1,3], Christina Holt[4,5], Christopher Harrop[4,5], Daniel Abdi[4,5], Jili Dong[1,3], Matthew E. Pyle[1], and Arun Chawla[1]

[1]NOAA/EMC, [2]Axiom Consultants, [3]IMSG, [4]NOAA/GSL, [5]CIRES

*With acknowledgement to Molthan, et al. (BAMS, 2015) who were the first to the eye catching title

Molthan, A. L., and Coauthors, 2015: Clouds in the Cloud: Weather Forecasts and Applications within Cloud Computing Environments. *Bulletin of the American Meteorological Society*, **96,** 1369-1379.



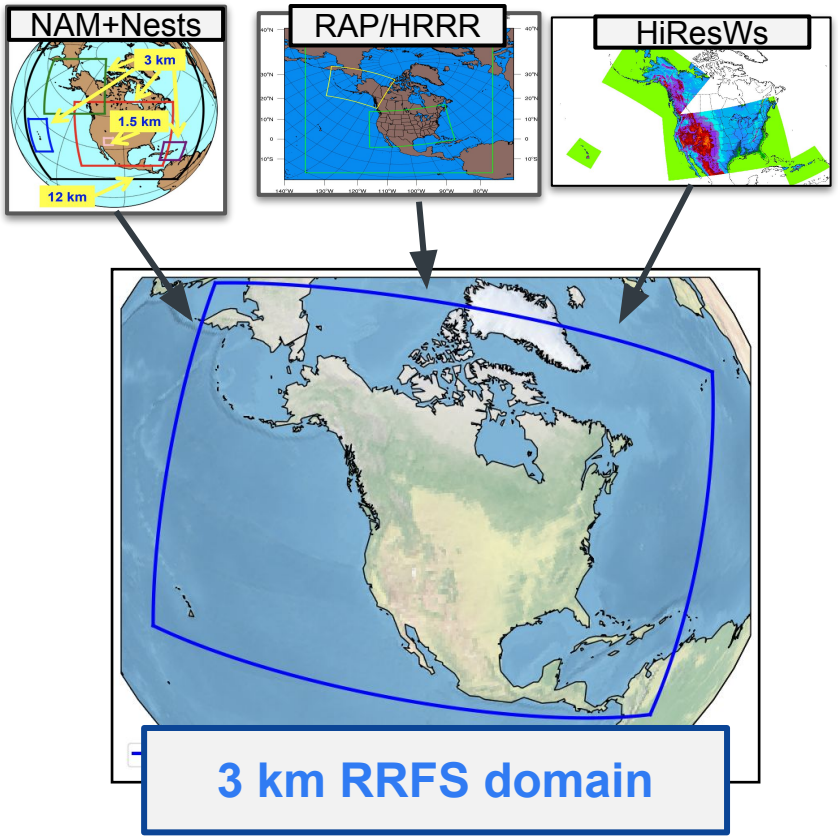

UFS
UNIFIED FORECAST SYSTEM

# Outline

- RRFS development - resource challenges
  - And some HPC + NWP history
- Using the Cloud for development
- Real-time prototype RRFS ensemble in the Cloud
- Lessons learned, next steps, and some thoughts

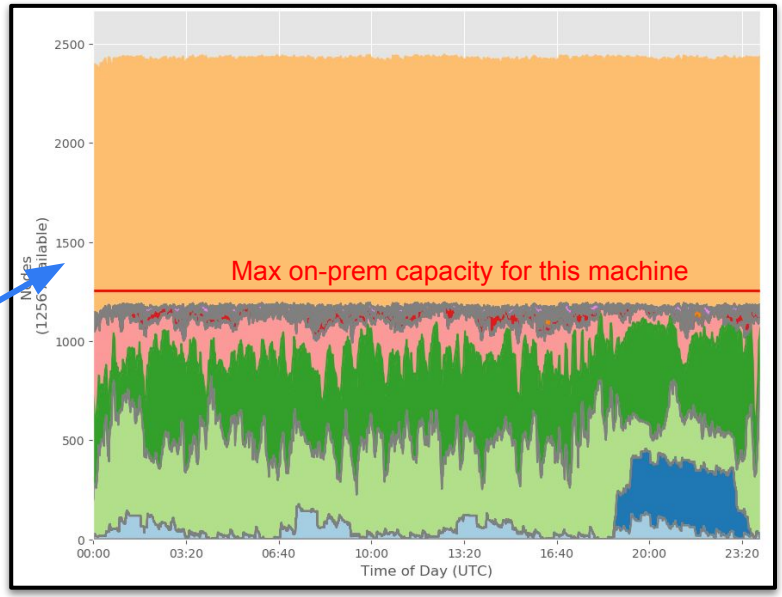
# RRFS - details

- Rapid Refresh Forecast System (RRFS)
  - Based on the FV3-Limited Area Model (LAM) from Black et al. (*JAMES*, 2021)
  - Rapidly updated
  - Convection-allowing (~3 km)
  - 65 vertical layers
  - Hybrid EnVar assimilation (est. 36 members)
  - Ensemble forecasts (~9 members)
    - *Stochastic and multiphysics suite*
    - 18h+ hourly
    - 60h every 6 hours



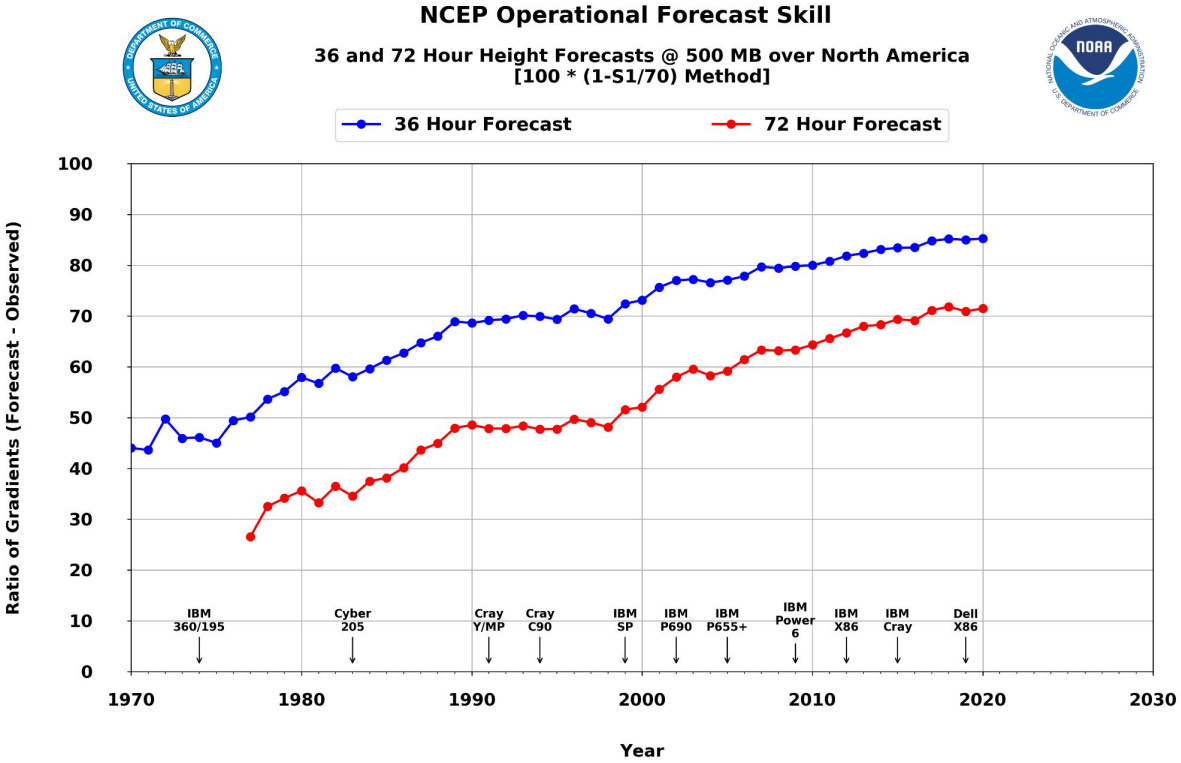
# Computing Needs for Development

- RRFS is Designed to run on NOAA's next operational supercomputing system (WCOSS2)
  - 12.1 PF machine (theoretical peak)
- Slated for implementation in late 2023
- Development & testing requires compute resources well beyond the availability of on-premises NOAA HPC systems
- **Can cloud computing fill the gap?**
  - Objective → on-time delivery in late 2023 of a well-tested system




The resources needed for development & testing of RRFS on a NOAA on-prem system is shown in **light orange**

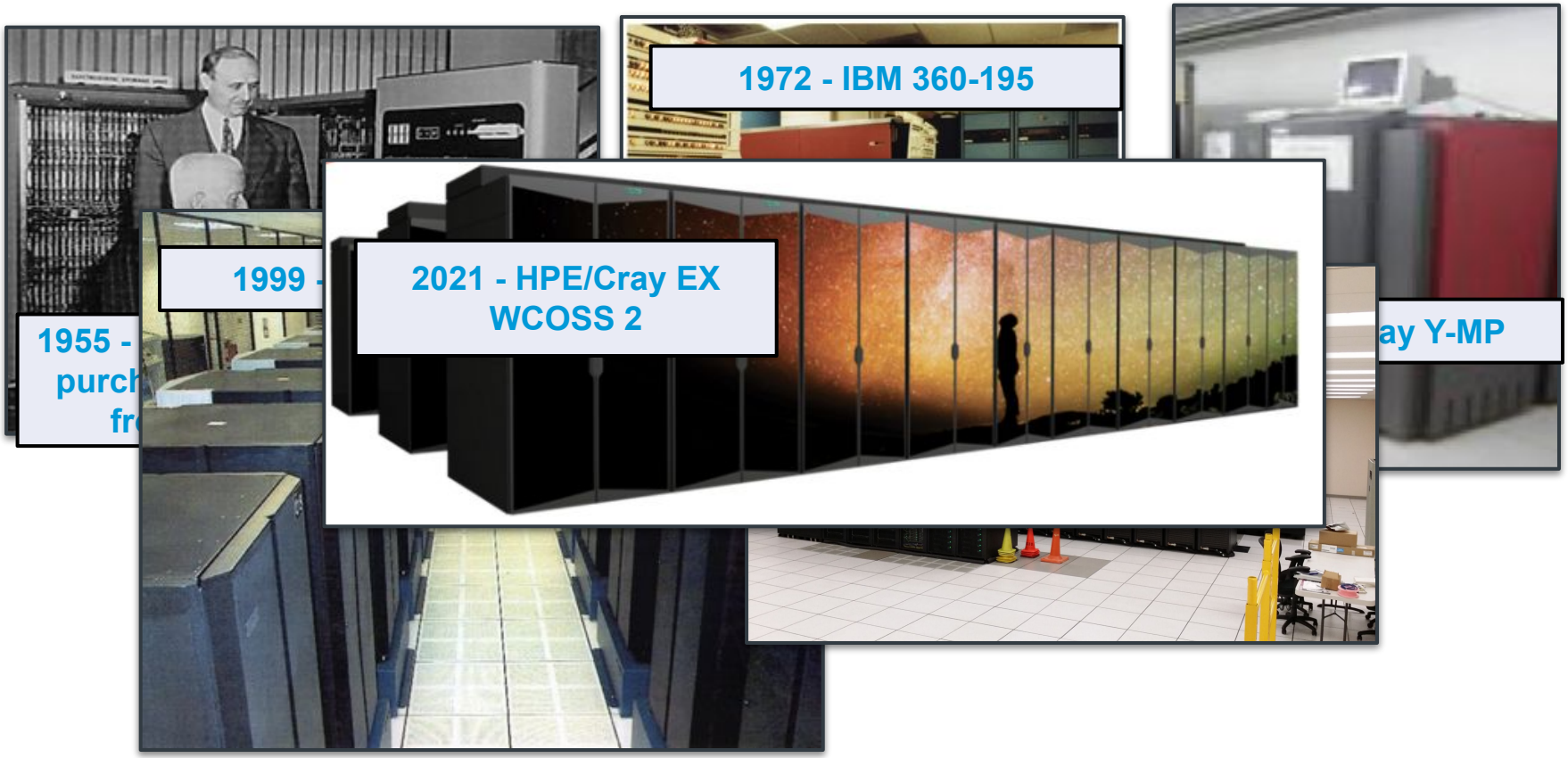
# Some HPC and NWP History




As computational performance increases, forecast skill improves

* Thanks to Mallory Row for the updated skill figure *

# A Few Highlights of HPC in Operational NWP






Much of this material was borrowed from Geoff DiMego's 2010 presentation, *50+ Years of NWP: Supercomputers and the Models that Consumed Them: A sordid tale of codependence*

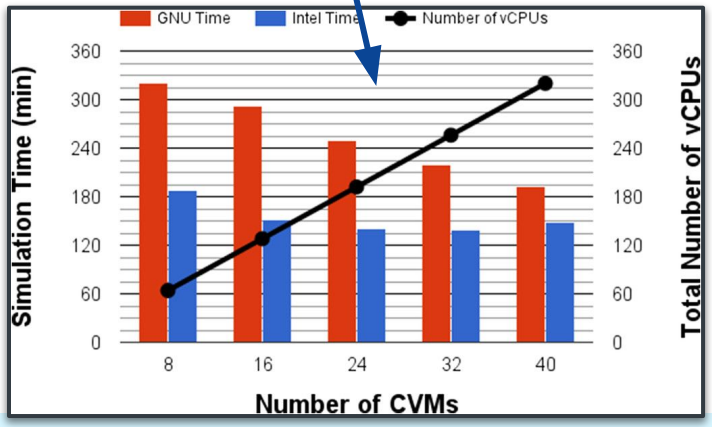
# Cloud Computing

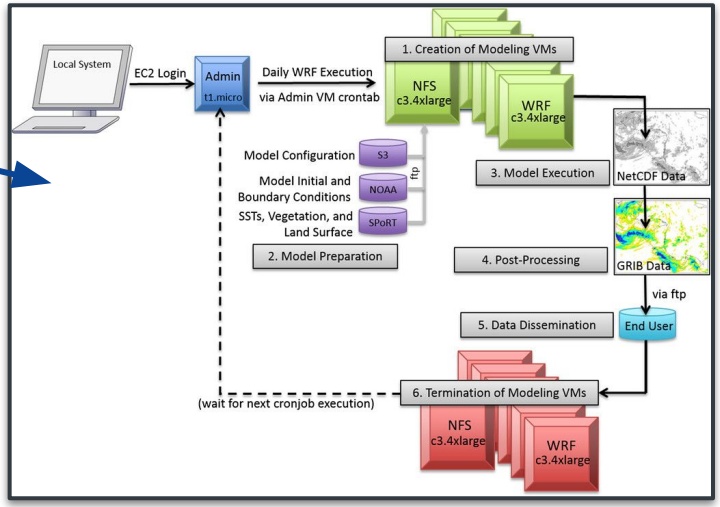
- Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. *Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis.**
- Has been around for well over a decade - we use it everyday though we may not be aware
  - Google suite of services, app stores, streaming services, big data firms, etc.
- Up until ~2015 Cloud was not very suitable for NWP HPC applications
  - NWP is a large, distributed memory task that requires fast interconnects and efficient I/O
  - As of 2010 cloud computing was 20x slower than modern (at the time) HPC, with interconnect being a significant issue (Jackson et al., 2010)

*From https://aws.amazon.com/what-is-cloud-computing/

# Cloud Computing - 2015 to 2019

- Molthan et al. (*BAMS*, 2015) demonstrated NWP with cloud HPC as feasible
- Siuta et al. (*WaF*, 2016) further evaluated capabilities across compilers and examined scalability




- Chiu et al (*JTECH*, 2019) Provide an analysis of ways to optimize cost when running NWP in cloud (data compression, spot instances, etc.)



From https://aws.amazon.com/what-is-cloud-computing/

# Today

## June 2021 Top 500

All cloud instances

| Rank | System | Cores | Rmax (TFlop/s) | Rpeak (TFlop/s) | Power (kW) |
|---|---|---|---|---|---|
| 26 | **Pioneer-EUS** - NDv4 cluster, AMD EPYC 7V12 48C 2.45GHz, NVIDIA A100, Infiniband HDR, Microsoft Azure<br>Azure East US<br>United States | 157,440 | 16,590.0 | 24,557.2 | |
| 27 | **Pioneer-SCUS** - NDv4 cluster, AMD EPYC 7V12 48C 2.45GHz, NVIDIA A100, Infiniband HDR, Microsoft Azure<br>Azure South Central US<br>United States | 157,440 | 16,590.0 | 24,557.2 | |
| 28 | **Pioneer-WEU** - NDv4 cluster, AMD EPYC 7V12 48C 2.45GHz, NVIDIA A100, Infiniband HDR, Microsoft Azure<br>Azure West Europe<br>Netherlands | 157,440 | 16,590.0 | 24,557.2 | |
| 29 | **Pioneer-WUS2** - NDv4 cluster, AMD EPYC 7V12 48C 2.45GHz, NVIDIA A100, Infiniband HDR, Microsoft Azure<br>Azure West US 2<br>United States | 157,440 | 16,590.0 | 24,557.2 | |
| 40 | **Amazon EC2 Instance Cluster us-east-1a** - Amazon EC2 r5.24xlarge, Xeon Platinum 8260 24C 2.4GHz, 25G Ethernet, Amazon Web Services<br>Descartes Labs<br>United States | 172,692 | 9,950.3 | 15,106.5 | |

# FY19 Disaster Supplemental IFHFW* Portfolio Project to Accelerate RRFS Development

- Port, deploy, and test RRFS prototype(s) in the cloud
- Key application/library requirements
  - Porting – easy to run existing binaries or rebuild from scratch
  - Nodes (instances) – need highest performance processors and adequate memory/node
  - Interconnect – support parallel scaling on 100+ nodes
- Other requirements
  - Availability of resources
  - System reliability
- We began testing in *January 2021*

**Goal**: Deploy Prototype real-time RRFS Ensemble Forecast system for evaluation in two of NOAA's flagship testbeds → the 2021 HWT Spring Forecast Experiment and the HMT Flash Flood and Intense Rainfall Experiment

# Our Hybrid Approach

- We developed a methodology to leverage the on-prem systems for porting, performance & debugging in support of the Cloud-based simulations. Key benefits include
  - Cost savings
  - Leverage the expertise of on-prem user community (SMEs, computational scientists, etc.) in debugging porting & performance issues
- Porting
  - Use on-prem binaries and file system layout, etc. – rapidly identify key issues
    - Building from scratch on cloud was also done, but does not improve performance
  - For rebuilding on Cloud, use same compilers, MPI, etc. for easy debugging
  - Final step is rebuilding of libraries and application binaries
- Performance (and cost)
  - On Cloud, a new dimension of cost is added
  - Application benchmarking is critical to determine the right type of instance to optimize both application performance and cost
  - Benchmarking needs to be an ongoing activity as new types of processors are added to the Cloud system

# Building the Capability on Amazon Web Services (AWS)

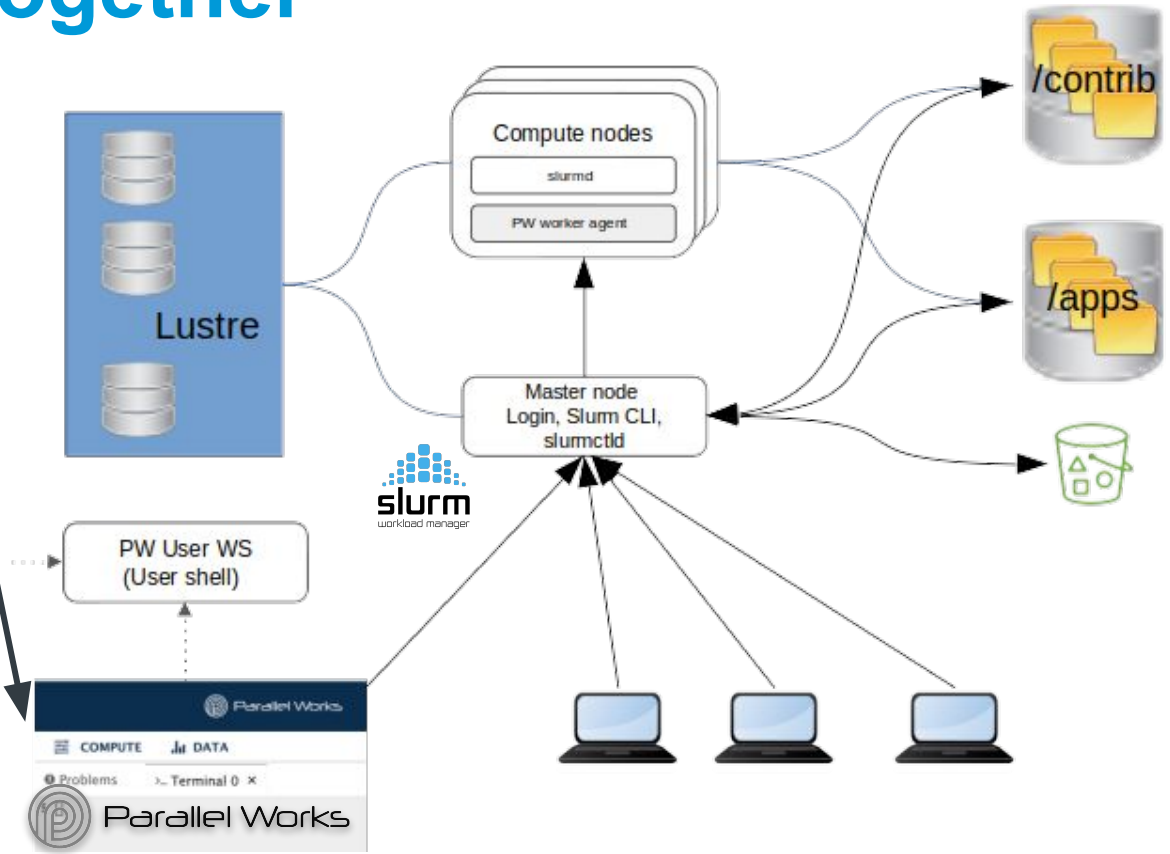
- Instances (i.e. nodes)
  - c5n.18xlarge → Used for the ensemble forecast step
    - Intel Xeon Skylake chips @ 3.5 GHz
    - 36 cpus per instance (72 w/ hyperthread)
    - 192G memory per instance
    - Elastic Fabric Adapter (i.e. the interconnect) 100 Gbps
  - r5.24xlarge → Less compute and MPI bound steps (pre-processing and graphics)
    - Intel Xeon Skylake chips @ 3.1 GHz
    - 48 cores per instance (96 w/ hyperthread)
    - 768G memory per instance
    - 25 Gbps network
- *On-demand* or *spot* instances?
  - *On demand* → You get it when you ask for it, typically more expensive
  - *Spot* → Leverages unused instances for reduced price (you bid for), suitable for fault tolerant work. Jobs may be terminated based on spot market and capacity from *On demand*
  - For big MPI jobs needing many instances, *On demand* is the best option

# Filesystems

- Amazon FSx for Lustre
  - High performance file system (transient) needed for good performance on the forecast step
    - Transient → it shuts down when we shut down the cluster
- Amazon EFS (Elastic File System)
  - Persistent, shared space used for libraries & user data
  - e.g., HPC stack
- AWS Simple Storage Service (S3)
  - Cost effective, long term storage)
  - Project bucket – source codes, binaries, fixed data, etc.
  - BDP (Big Data Project) bucket – post processed and plot files

# Putting it all together

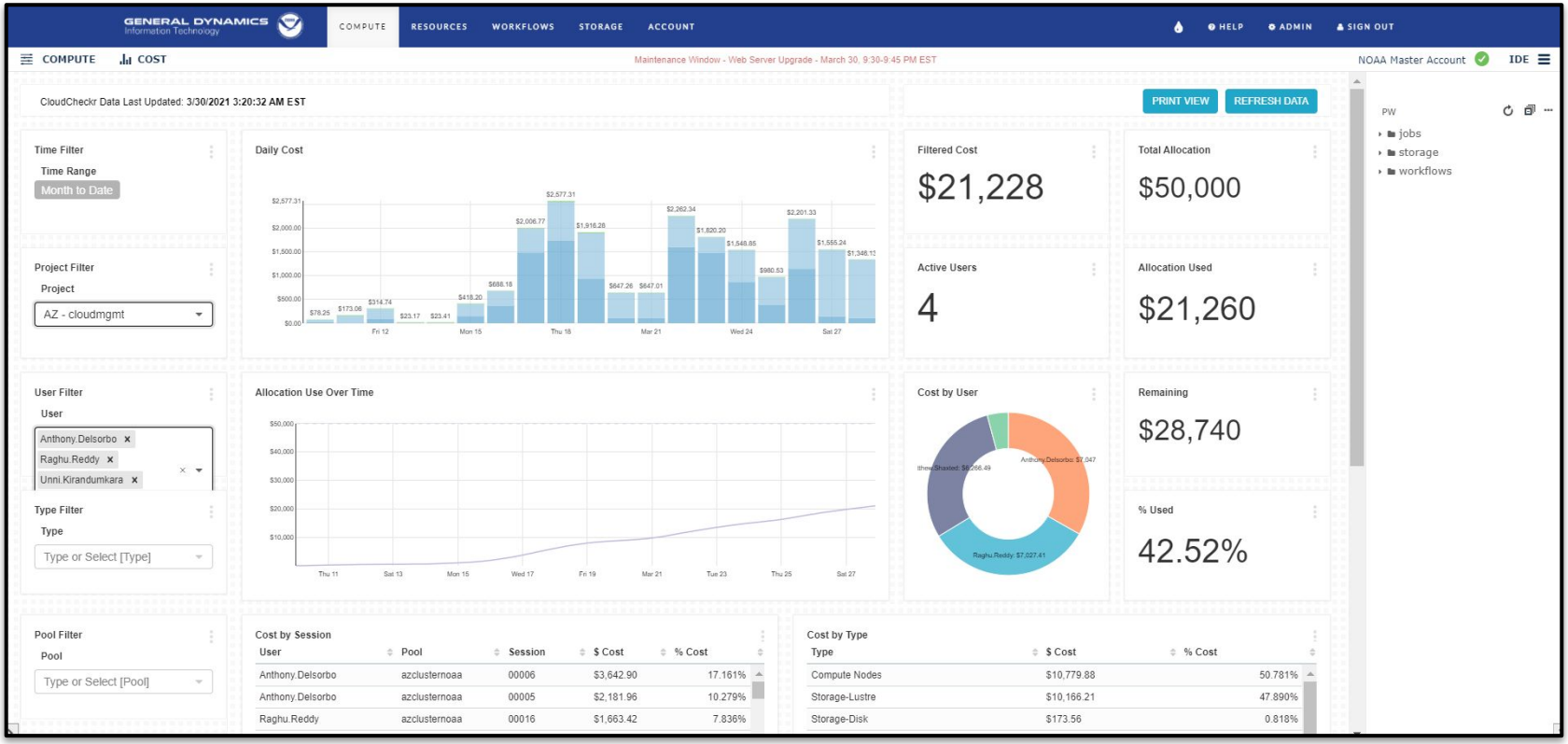
- GDIT/Parallel Works platform
  - Access to AWS, GCP and Azure HPC transient clusters
  - Dashboard for near real time cost/resource monitoring
  - Shared group persistent /contrib space
- Use AWS Parallel Cluster
  - Transient Lustre Filesystem
  - Shared group S3 bucket
  - Slurm batch queueing






For developers → Has the look/feel of on-prem HPC

# GDIT/Parallel Works: Resource Cost Tracking

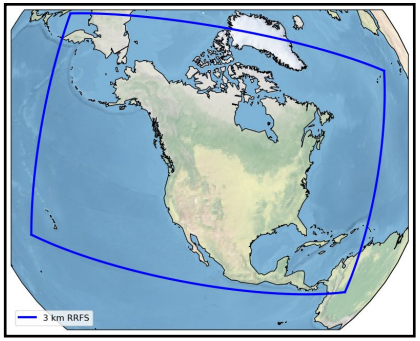





* Acknowledgement to Unni Kirandumkara for this material *

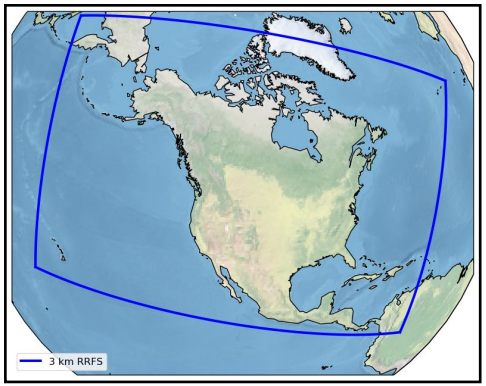
# Optimizing Resources and Runtime

- We ran 9 members once per day for the 00Z cycle to 60 hours with a goal delivery time of the morning of the testbed experiments (HWT and FFaIR)
- This gave us *more* time to run, so could use fewer instances → saving cost
- We benchmarked this with NOAA on-prem HPC system for member1
  - AWS is ~15% faster than Hera




| System | Number of instances or nodes | Wall time for FFAIR 60hr forecast (hrs) |
|---|---|---|
| AWS | 80 (c5n.18xlarge) | 3.96 |
| NOAA HPC System (Hera) | 80 | 4.68 |

# Prototype *Forecast* Ensemble




| Ens member ID | IC/LBCs | Physics Suite | Stochastic Schemes |
|---|---|---|---|
| Member 1 | 6H forecast from 18ZGFS | PBL: MYNN MP: Thompson | none |
| Member 2 | 6H forecast from 18ZGEFS - mem1 | | SPPT |
| Member 3 | 6H forecast from 18ZGEFS - mem2 | | SPPT/SHUM/SKEB |
| Member 4 | 6H forecast from 18ZGFS | PBL: TKE-EDMF MP: GFDL | none |
| Member 5 | 6H forecast from 18ZGEFS - mem1 | | SPPT |
| Member 6 | 6H forecast from 18ZGEFS - mem2 | | SPPT/SHUM/SKEB |
| Member 7 | 6H forecast from 18ZGFS | PBL: Hybrid-EDMF MP: NSSL | none |
| Member 8 | 6H forecast from 18ZGEFS - mem1 | | SPPT |
| Member 9 | 6H forecast from 18ZGEFS - mem2 | | SPPT/SHUM/SKEB |

Stochastic physics options came via synergies with UFS-R2O as well as the HIWT project, *Implementation and testing of stochastic perturbations within an FV3-Limited Area Model (LAM) ensemble using the Common Community Physics Package (CCPP)", but "regional FV3 ensemble (PIs J. Beck and J Wolff)*

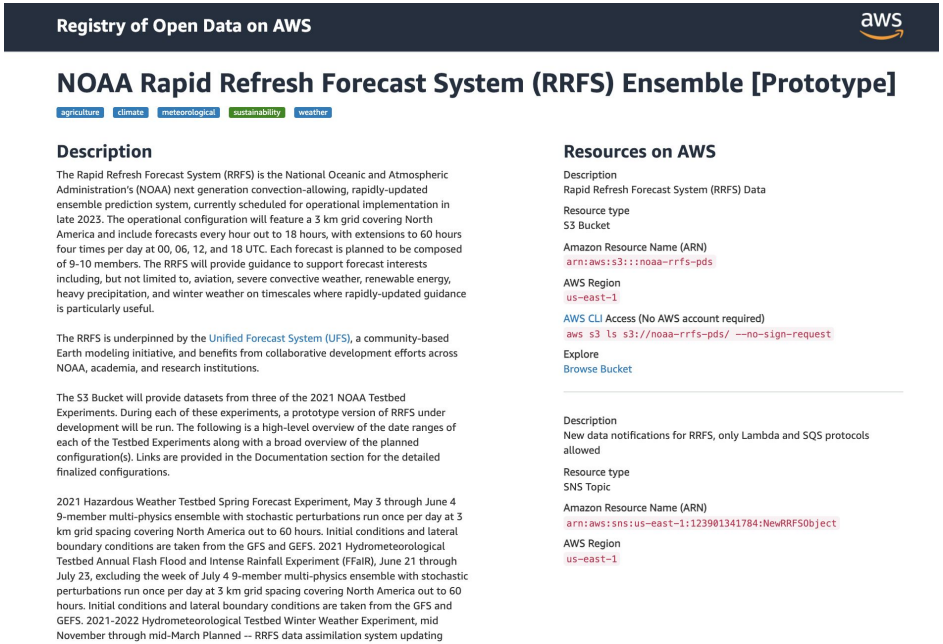


* Acknowledgement to Ted Mansell (NSSL), Larissa Reames (CIMMS / NSSL), and Tim Supinie (OU/CAPS) for assistance with NSSL 2-moment MP *

# NOAA's Big Data Program

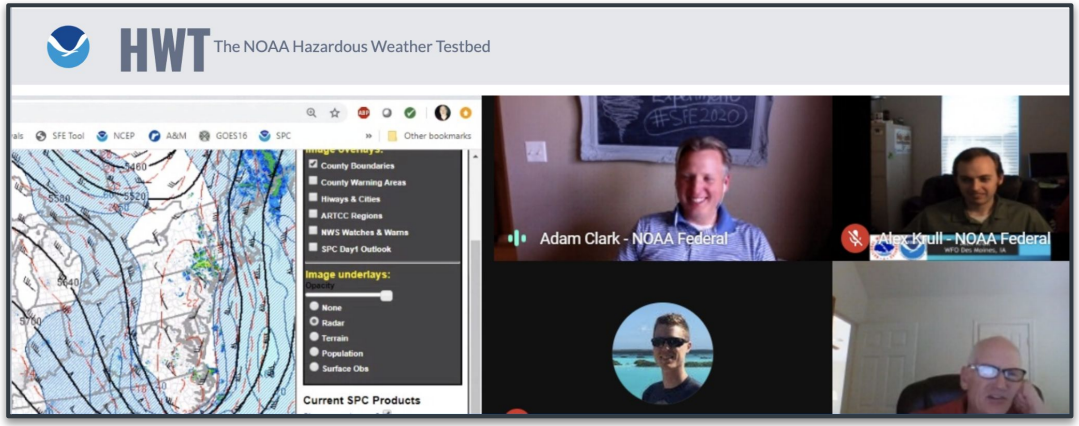
## The RRFS Ensemble Generates Considerable Data Making it Accessible is Critical

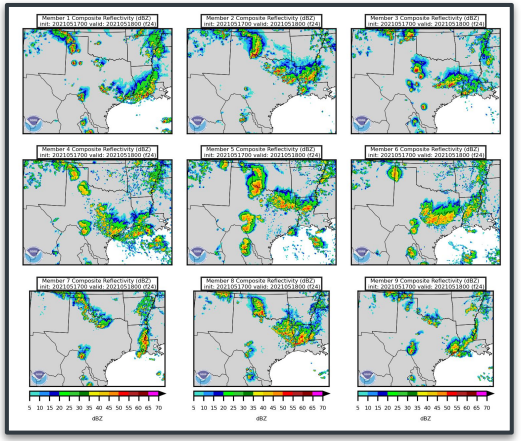
- We partnered with the NOAA NOAA Big Data Program (BDP) to host the outputs of the RRFS ensemble
  - Eliminates storage/egress fees for our project
- RRFS prototype output is a part of the Registry of Open Data on AWS
- Easy to access *and* use
- Easy for the community to work with




https://registry.opendata.aws/noaa-rrfs/

# The 2021 HWT Spring Forecast Experiment



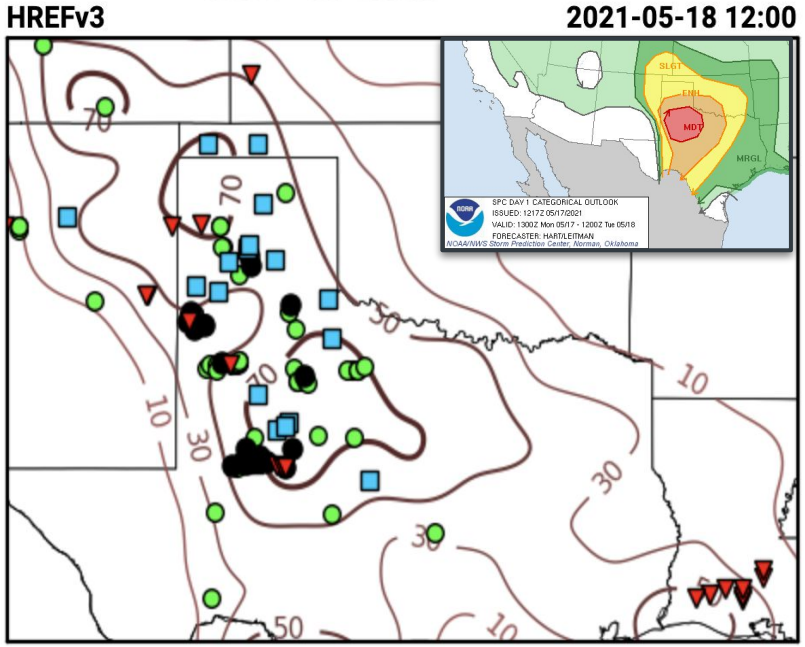



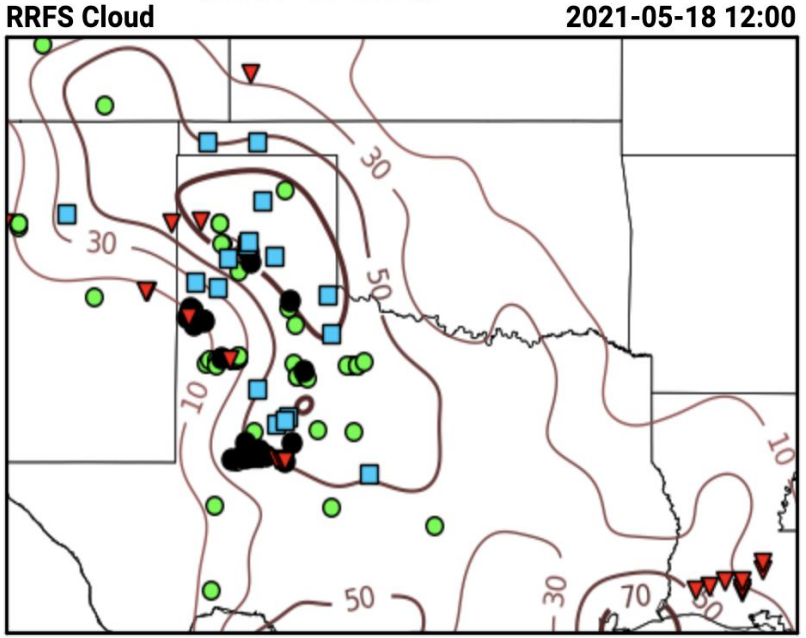
- May 3 - June 4
- The SFE is a yearly experiment that investigates the use of convection-allowing model forecasts as guidance for the prediction of hazardous convective weather
- Comparisons made with HREFv3 as well as other collaborator-provided experimental ensembles



HWT SFE info and figure obtained from https://hwt.nssl.noaa.gov/sfe/2021/

# The 2021 HWT Spring Forecast Experiment

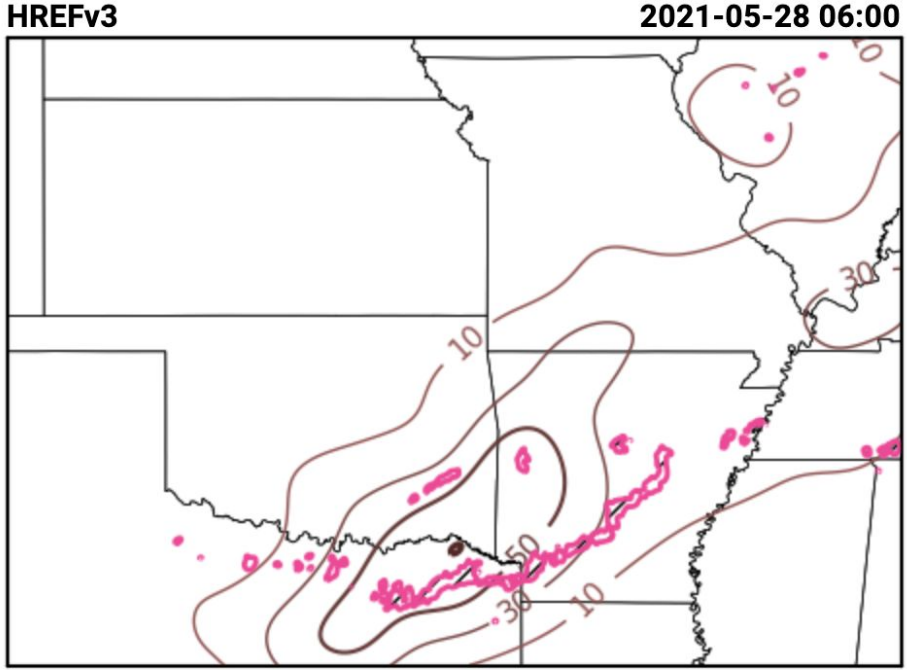


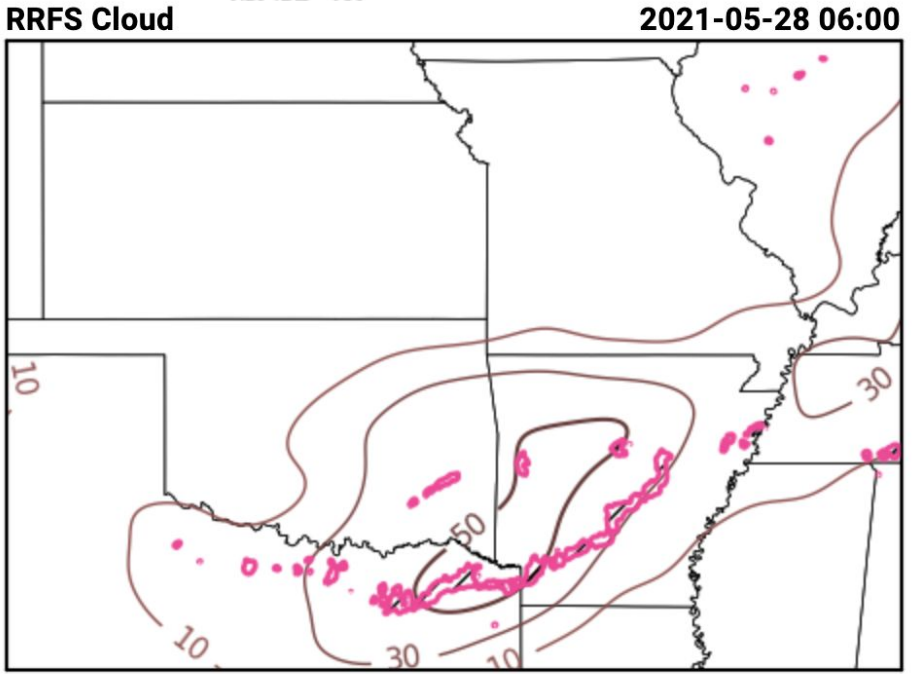




24hr neighborhood (r=40km) maximum probabilities of updraft helicity > 99.85 pctile
00Z initializations covering 12Z May 17th through 12Z May 18th

Figures from the HWT's excellent model SFE comparison page:
https://hwt.nssl.noaa.gov/sfe_viewer/2021/model_comparisons/

# The 2021 HWT Spring Forecast Experiment

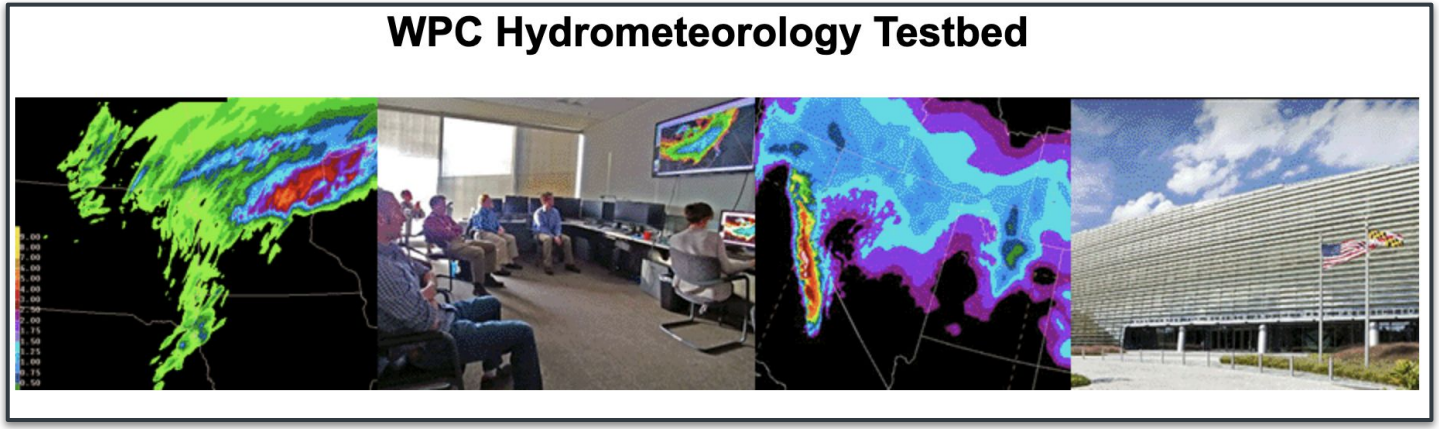



Neighborhood (r=20km) probabilities of column max reflectivity > 40 dBZ
MRMS Observed column max reflectivity > 40 dBZ in pink
30 hr forecasts valid May 28th at 06Z

Figures from the HWT's excellent model SFE comparison page:
https://hwt.nssl.noaa.gov/sfe_viewer/2021/model_comparisons/

# The 2021 HMT Flash Flood and Intense Rainfall Experiment



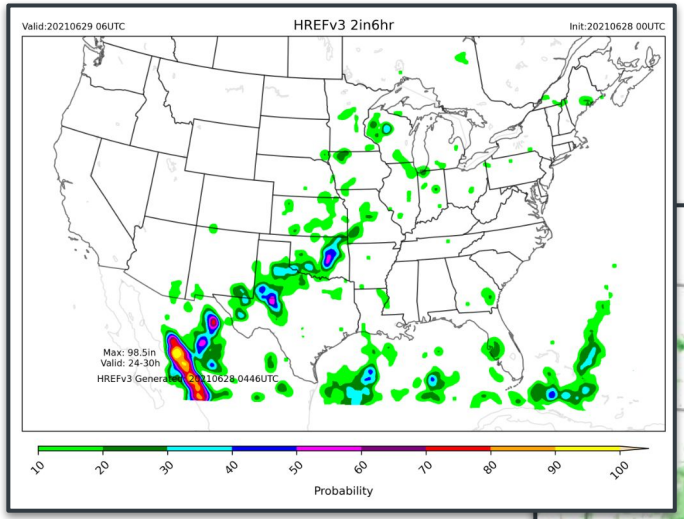

- June 21 - July 23
- FFaIR focuses on improving short term QPF and flash flood forecasts through the use of high resolution models and ensembles and rapidly updating hydrologic information
- For FFaIR we added new ensemble products, including probabilities of exceeding flash flood guidance and recurrence intervals



HMT FFaIR info and figure obtained from https://www.wpc.ncep.noaa.gov/hmt/

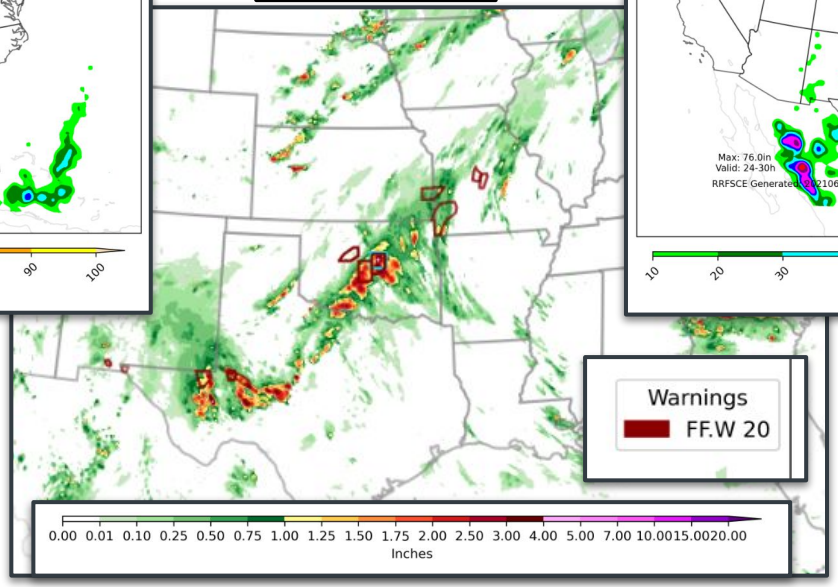
# The 2021 HMT Flash Flood and Intense Rainfall Experiment

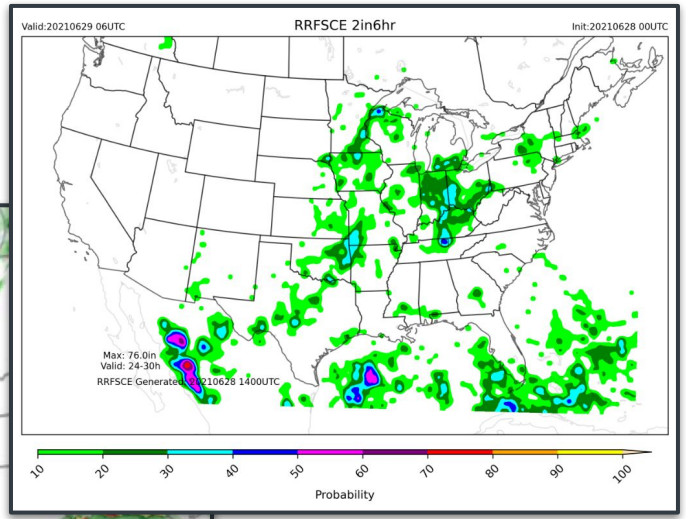
Neighborhood (r=40km) maximum probabilities of QPF > 2 in. in 6 hours
00Z 28 June initializations covering 00Z-06Z June 29th




HREFv3

MRMS QPE



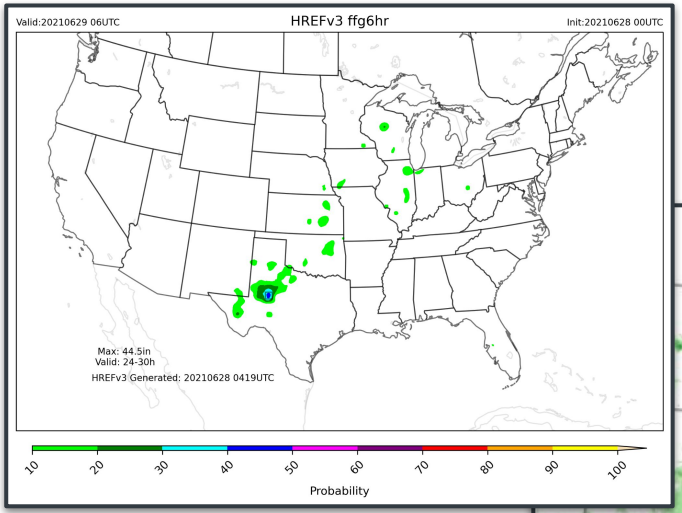

RRFS Ens



* Acknowledgement to the HMT's Sarah Trojniak for the material on this slide *

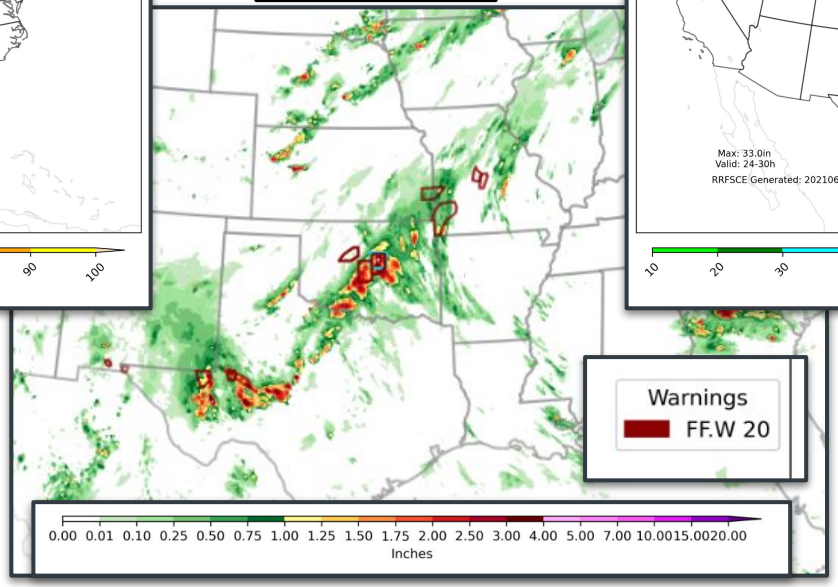
# The 2021 HMT Flash Flood and Intense Rainfall Experiment

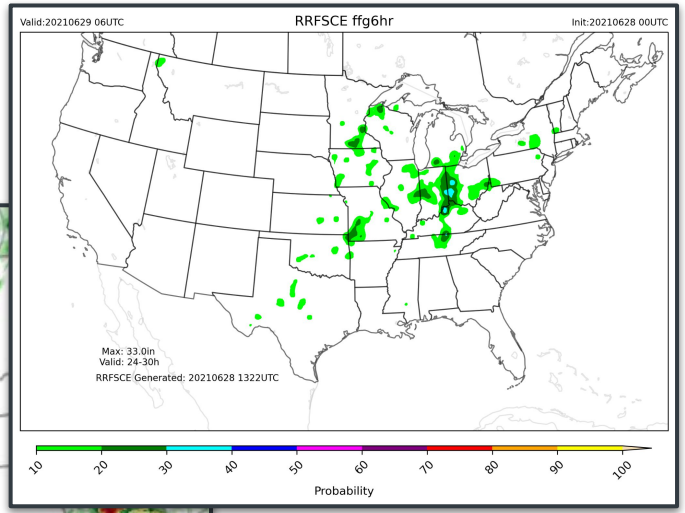
## Probability of exceeding Flash Flood Guidance in 6 hours
## 00Z 28 June initializations covering 00Z-06Z June 29th




**HREFv3**

**MRMS QPE**




**RRFS Ens**



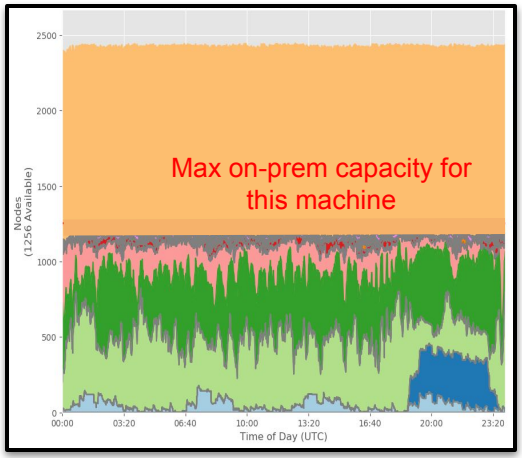

* Acknowledgement to the HMT's Sarah Trojniak for the material on this slide *

# A Developer's Perspective on Cloud & On-prem

- Resource capacity
  - Cloud: 100s of instances readily available – a huge advantage for test & development of parallel applications
  - On-prem: batch queueing prioritization (e.g. FairShare)
- Processor availability
  - Cloud: many types of processors including the latest generation of hardware (Intel, AMD, ARM, GPUs)
  - On-prem: same type of processor in the system which is used for 3-5 yrs
- Parallel File System
  - Cloud: Lustre file system
  - On-prem: WCOSS IBM Spectrum Scale (GPFS) and Lustre (Hera,Orion,Jet)
- Throughput
  - Cloud: On-demand (unless spot)
  - On-prem: batch queueing system prioritization
- Up time
  - Cloud: no maintenance windows, no production switches, 100% up time
  - On-prem: regular maintenance down time, production switches

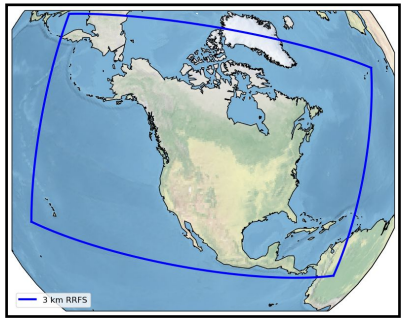
# Where Cloud has Been a Breakthrough

- RRFS is big
- We have access to a few on-prem HPCs
  - Presently none have enough space for the full RRFS
    - WCOSS2 certainly will
  - We cannot run 1-2 members on each HPC
  - We can (and do) make the domain smaller, run a smaller ensemble, and make other adjustments
    - Reasonable actions to conserve resources
    - But it is *not* the full RRFS
- The cloud → *Dynamically scales to accommodate the resources we need*



# Closing Items

- Critical RRFS development is made possible by access to cloud HPC
- Interested in accessing output from early RRFS testing?
  - BDP: https://noaa-rrfs-pds.s3.amazonaws.com/index.html
- DA components are being added in preparation for the Winter Weather Experiment
  - Statistical analysis for HWT and FFaIR is underway
- RRFS planned implementation in late FY23
- As a part of this effort, the UFS-SRW App was also ported to work in the cloud
  - https://github.com/ufs-community/ufs-srweather-app
- Does cloud computing have a role in future operational NWP? It would appear so!




UFS
UNIFIED FORECAST SYSTEM

*Acknowledgement to Unni Kirandumkara (GDIT) and Matt Shaxted (Parallel Works) for their exceptional support in this project!*

# Talks and materials referenced in this presentation


Black, T. L., and Coauthors, 2021: A Limited Area Modeling Capability for the Finite-Volume Cubed-Sphere (FV3) Dynamical Core and Comparison With a Global Two-Way Nest. Journal of Advances in Modeling Earth Systems, 13, e2021MS002483. https://doi.org/10.1029/2021MS002483.

Chui, T. C. Y., D. Siuta, G. West, H. Modzelewski, R. Schigas, and R. Stull, 2019: On Producing Reliable and Affordable Numerical Weather Forecasts on Public Cloud-Computing Infrastructure. *Journal of Atmospheric and Oceanic Technology*, **36,** 491-509. 10.1175/jtech-d-18-0142.1.

Jackson, K. R., and Coauthors, 2010: Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud. *Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science*, IEEE Computer Society, 159–168.

Molthan, A. L., and Coauthors, 2015: Clouds in the Cloud: Weather Forecasts and Applications within Cloud Computing Environments. *Bulletin of the American Meteorological Society*, **96,** 1369-1379. 10.1175/bams-d-14-00013.1.

Siuta, D., G. West, H. Modzelewski, R. Schigas, and R. Stull, 2016: Viability of Cloud Computing for Real-Time Numerical Weather Prediction. *Weather and Forecasting*, **31,** 1985-1996. 10.1175/waf-d-16-0075.1.