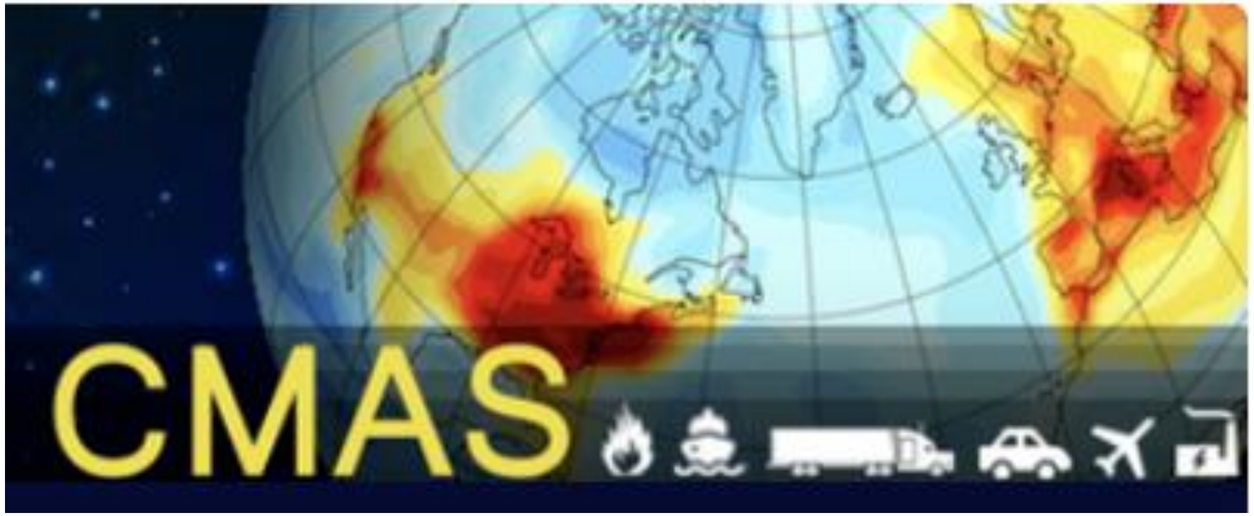
Community Modeling and Analyses (CMAS) Center Webinar Series



25 years of Serving a Global Community

CMAS Center Trainings

- Spring 2026
  - SMOKE Online, April 7 – 10
  - CMAQ Online, April 21 – 24
- Fall 2026
  - SMOKE Online, October 13 – 16
  - CMAQ Online, October 27 – 30

**** Seats available. Register soon ****

https://cmascenter.org/training/classes.cfm

25th Annual CMAS Conference

CMAS
CONFERENCE
2026

Chapel Hill, NC

October 19–21

Look out for Call for Abstracts and announcements coming soon!

https://cmascenter.org/conference/2026/index.cfm

CMAQ on the Cloud Help Sessions

CMAQ on the Cloud – Monthly Zoom Help Session
Hosted by: CMAS Center @ UNC
Audience: Modelers exploring or running **CMAQ on the Cloud**
Goal: Provide **one-on-one** support to members of the CMAS community
Topics Covered

- Troubleshooting AWS or Azure cloud **HPC resource provisioning**
- **Benchmarking CMAQ v5.4** (2018 12US1 2-day domain) on cloud HPC clusters
- Accessing input data & archiving outputs using **S3 copy or Lustre *lazy loading*** from the **CMAS Data Warehouse (AWS Open Data Registry)**
- Using **BeeGFS on Microsoft Azure CycleCloud**

When: First Thursday of every month, **3 – 4 PM ET**
Upcoming Sessions: Apr 2 • May 7 • Jun 4 • Jul 2

Upcoming CMAS Webinar: Exploring AMET Web on AWS

Save the Date: Thursday May 14 1:00 – 2:30 ET

- **Next CMAS Webinar on May 14 to demonstrate AMET Web on single VM**
  - **Interactive web-based approach to use AMET for model evaluation on the Cloud (Will include a Live Demo)**
- Documentation:
  - AMET AQ Website: https://pcluster-cmaq.readthedocs.io/en/latest/user_guide_pcluster/amet/ami_website.html#create-air-quality-plots-using-the-amet-aq-website
  - AMET MET Website: https://pcluster-cmaq.readthedocs.io/en/latest/user_guide_pcluster/amet/ami_website.html#create-met-plots-using-the-amet-met-website

03/19/2026

Lessons Learned for Performing Long-Term CMAQ Production Simulations for the 12US1 and 36US3 Domains on AWS Parallel Cluster

Manish Soni, Elizabeth Adams and Saravanan Arunachalam

UNC Institute for the Environment | CMAS Webinar

The University of North Carolina at Chapel Hill

Outline

- Motivation
- CMAQ Modeling Framework and Study Domains
- AWS ParallelCluster Architecture
- Benchmark Setup
- Performance Results
- Storage and Cost Optimization
- Lessons Learned
- Recommended Configuration
- Future Work

AWS - Amazon Web Service
CMAQ- Community Multi-scale Air quality Modeling System

Motivation

- Long-term CMAQ simulations require **significant HPC resources**
  - Air quality metrics for regulatory applications or quantifying health impacts and climate interactions need annual or even multi-annual simulations
- Cloud computing enables **scalable HPC environments**
- AWS ParallelCluster simplifies **cluster deployment**

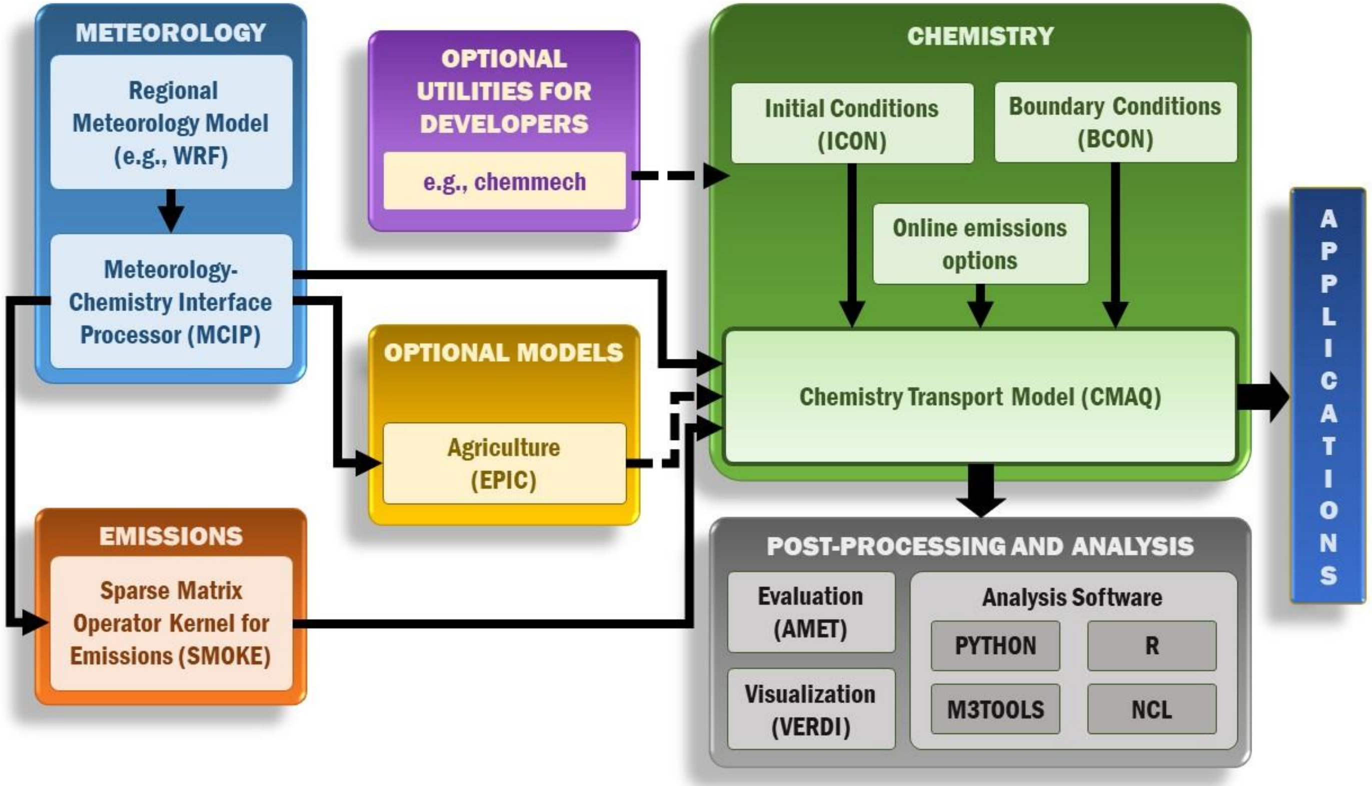
Problem Statement

- Traditional HPC clusters have **limited availability and fixed capacity**
- Cloud computing provides **on-demand scalable resources**
- However, **cost and performance tradeoffs must be evaluated**

Objectives

- Identify **cost-efficient AWS configuration for CMAQ production runs**
- Perform **long-term CMAQ simulations** on AWS Parallel Cluster and provide guidance for CMAS community

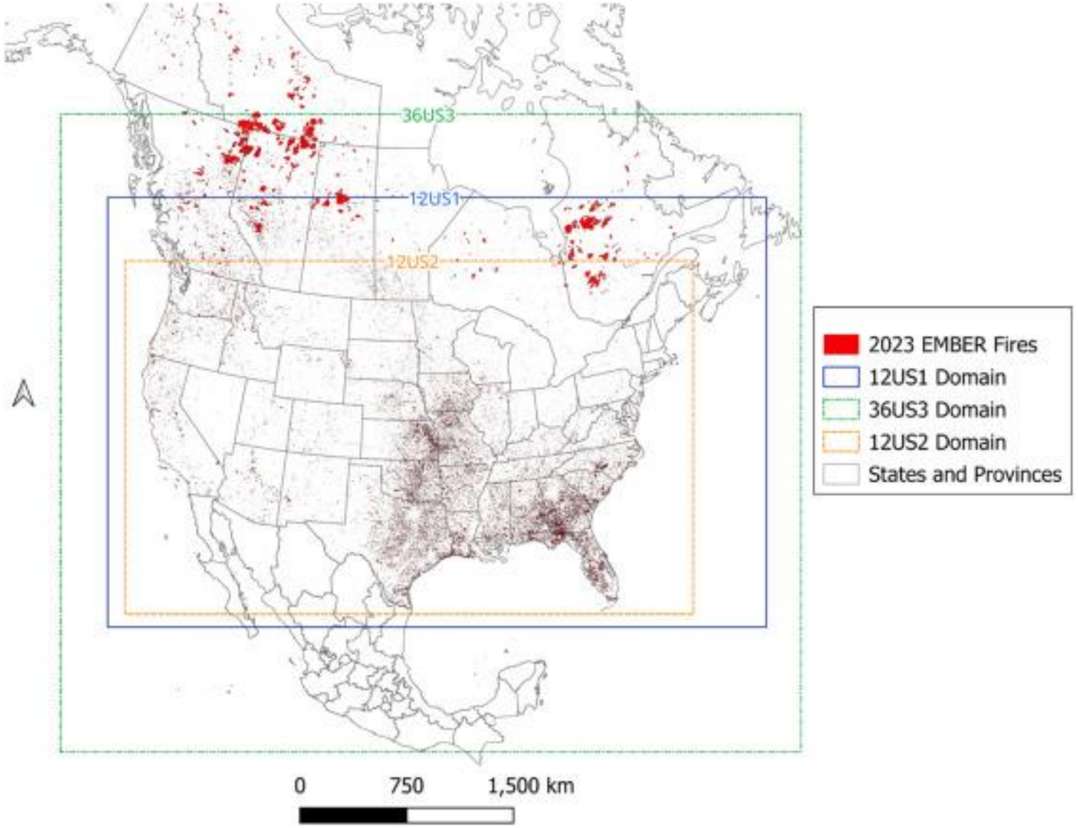
CMAQ Modeling System Framework



- Community Multiscale Air Quality Model (CMAQ) simulates O_3, $PM_{2.5}$ and NO_2, etc. for different air quality applications
- Requires large-scale HPC resources
- Typical production runs require annual simulations

Source : https://www.epa.gov/cmaq/cmaqs-purpose

Domains Used in Study



- **12US1 domain:** 499 × 259 (columns × rows)
- **36US3 domain:** 172 × 148
- Used for **benchmarking computational** costs
- **Different compute and storage** requirements

Source: Simon, H et al., (2025). Expedited modeling of burn events results (EMBER): A screening-level dataset of 2023 ozone fire impacts in the US. *Data in Brief*, *58*, 111208.

Running on Single Virtual Machine (VM) vs Parallel Cluster

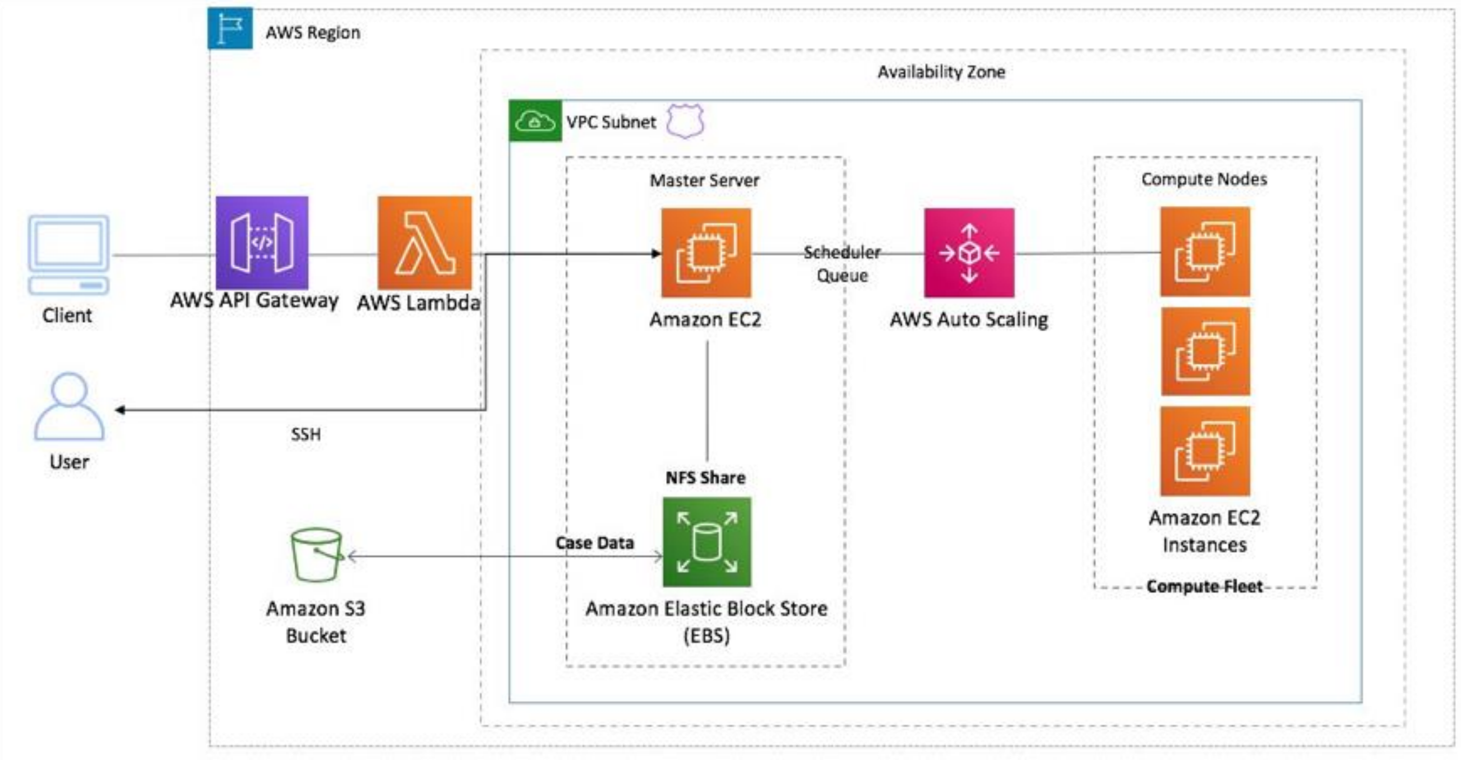
- **Single VM:** A single virtual machine with fixed CPU, memory, and storage where tasks run on one node. It is simple to set up but limited in scalability and performance
- **ParallelCluster:** A group of multiple interconnected machines (nodes) that work together to execute tasks simultaneously. It provides higher performance, scalability, and is suitable for large-scale simulations or HPC workloads
- Example
  - **Single VM:** Running a CMAQ simulation on one EC2 instance
  - **Parallel Cluster:** Running CMAQ across multiple nodes using MPI with a fast interconnect like **AWS Elastic Fabric Adapter (**network interface for faster intercommunication within nodes)

Prerequisite for AWS parallel Cluster

- **AWS Account** with billing enabled and access to the AWS Management Console
- **IAM permissions** to create and manage AWS resources (EC2, VPC, FSx, S3, Auto Scaling)
- **AWS CLI installed and configured** to interact with AWS services from the command line
- **Python environment (Python 3.8 or later)** to install and run ParallelCluster tools
- **EC2 Key Pair** for secure SSH access to the cluster head node
- **VPC networking setup** including subnets, security groups, and internet gateway
- **HPC EC2 instances** (e.g., **hpc7g** or **hpc6a**) for high-performance parallel workloads
- **Shared storage configuration** such as **FSx for Lustre** for fast parallel file access
- **Amazon S3 storage** for input data, output results, and backups (optional)

AWS - Amazon Web Services IAM - Identity and Access Management EC2 - Elastic Compute Cloud VPC - Virtual Private Cloud FSx - Amazon File System x S3 - Simple Storage Service CLI - Command Line Interface SSH - Secure Shell HPC - High Performance Computing

AWS ParallelCluster Architecture



- Open-source tool for deploying HPC clusters on AWS
- Supports SLURM workload manager
- Automates cluster creation and scaling
- SLURM dynamically provisioned the compute nodes and deregistered them once the job is complete, ensuring that compute costs are only incurring costs during job execution
- Supports FSx for Lustre and EFA

Source :https://aws.amazon.com/blogs/compute/using-aws-parallelcluster-with-a-serverless-api/

EFA - Elastic Fabric Adapter
SLURM - Simple Linux Utility for Resource Management

AWS Parallel Cluster Storage and Networking

Elastic Block Storage (EBS)

Configure to use maximum Input/Output Operations per Second (IOPS) and throughput. EBS allows users to provision performance independent of storage capacity

- High performance, low-latency disk

Lustre/FSx (used on ParallelCluster)

- Process massive datasets at up to hundreds of gigabytes per second of throughput, millions of IOPS, and sub-millisecond latencies
- Capable of "*lazy loading*" data (loading data when it is first used) from S3 bucket

Simple Storage Service (S3) (used on ParallelCluster)

Archive data at the lowest cost

- Run cloud-native applications by *lazy loading* from the s3 bucket to /fsx

Network Options used

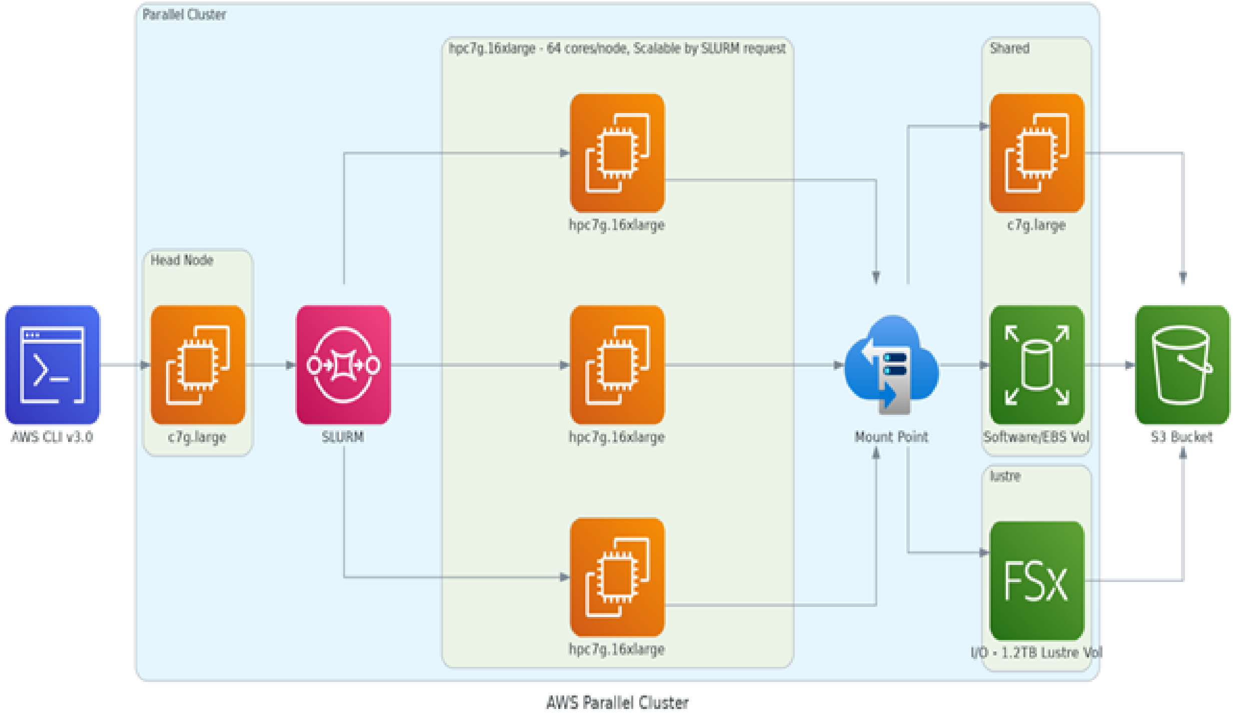
- Elastic Fabric Adapter (EFA) enabled
- High throughput

AWS Parellel Cluster Deployment

- Clusters deployed in US-East regions
- ARM64 cluster using Graviton3E processors
- x86_64 cluster using AMD EPYC processors
- Separate build environments created for benchmarking
- ARM64 does not support intel compilers
- Compilers installed on x86_64: intel 25.0.4 and gcc 11.4.1
- Compilers installed on ARM64: gcc 11.4.1
- Operating System: ubuntu
- Parallel cluster Version: 3.12

ARM – Advanced RISC Machine
RISC – Reduced Instruction Set Computer
AMD – Advanced Micro Devices

AWS ParallelCluster Architecture for CMAQ



Why HPC EC2 instances are required for CMAQ

- CMAQ requires **highly parallel HPC compute resources**
- Use **AWS HPC instances (hpc7g, hpc6a)**
- Built on **AWS Nitro System for** low-latency internode communication
  - **Up to 200 Gbps network bandwidth** for tightly coupled workloads
- Enables **highly scalable CMAQ simulations**

Source: Efstathiou et al., (2024). Enabling high-performance cloud computing for the Community Multiscale Air Quality Model (CMAQ) version 5.3. 3: performance evaluation and benefits for the user community. *Geoscientific model development*, *17*(18), 7001-7027.

Architecture and Processor Selection

Component	x86_64 Cluster	ARM64 Cluster
Login Node	c6a.xlarge 4 cores 8GB RAM $0.15/hour	c7g.large 2 cores 4GB RAM $0.07/hour
Compute Node	hpc6a.48xlarge 96 cores 384 GB RAM $2.88/hour	hpc7g.16xlarge 64 cores 128 GB RAM $1.68/hour
Architecture	x86_64	ARM64
Region	us-east-2 (Ohio)	us-east-1 (N. Virginia)

https://aws.amazon.com/ec2/instance-types/hpc7g/
https://aws.amazon.com/ec2/instance-types/hpc6a/

Benchmark Methodology

- Initial tests conducted using **12US1 domain**
- Performance metric: Average **CPU-sec/day**
- **Different processor counts** evaluated
- **Shared** storage and **Lustre** storage tested
- CMAQ version : **5.4+**
- Mechanism : **cb6r5_ae7**
- Compilers : **intel/gcc**

AWS Parallel Cluster Workflow

1. Activate environment (python CLI) and Install Parallel cluster v 3.12
   source ~/hpc-ve/bin/activate
2. Configure the cluster using with AWS CLI 3.0
   pcluster configure –config hpc7g.16xlarge.yaml
3. Modify YAML to use CMAQ snapshot and Lustre storage
4. Create the ParallelCluster
   pcluster create-cluster --cluster-configuration hpc7g.16xlarge.yaml --cluster-name cmaq --region us-east-1
5. Check cluster status
   pcluster describe-cluster --region=us-east-1 --cluster-name cmaq
6. Connect to head node
   pcluster ssh -v -Y -i ~/cmas.pem --region=us-east-1 --cluster-name cmaq
7. Run CMAQ simulations
   Submit jobs to SLURM queue and use Lustre for output
8. Delete Cluster
   pcluster delete-cluster --region=us-east-1 --cluster-name cmaq

Complete Step by step method how to install and create Single VM and parallel cluster is available here:
https://pcluster-cmaq.readthedocs.io/en/latest/user_guide_pcluster/cmaq-vm/quick-start.html
Git repository for preconfigured yaml examples and run scripts to use for running CMAQ v5.4
https://github.com/CMASCenter/pcluster-cmaq

Yaml – Yet Another Markup Language

CMAS Data Warehouse

CMAS Data Warehouse on AWS is a CMAS-Center repository to disseminate meteorology, emissions, and air quality model input and output for Community Multiscale Air Quality (CMAQ) Model Applications

https://registry.opendata.aws/cmas-data-warehouse/

- This repository is available as part of the AWS Open Data Program; therefore, egress fees are not charged to either the host or the person downloading the data
- These public S3 buckets are hosted by the CMAS Center on behalf of the U.S. EPA's Office of Research and Development, and the U.S. EPA's Office of Air and Radiation
- Metadata for the datasets in the CMAS Data Warehouse are available from the CMAS Dataverse site: https://dataverse.unc.edu/dataverse/cmascenter

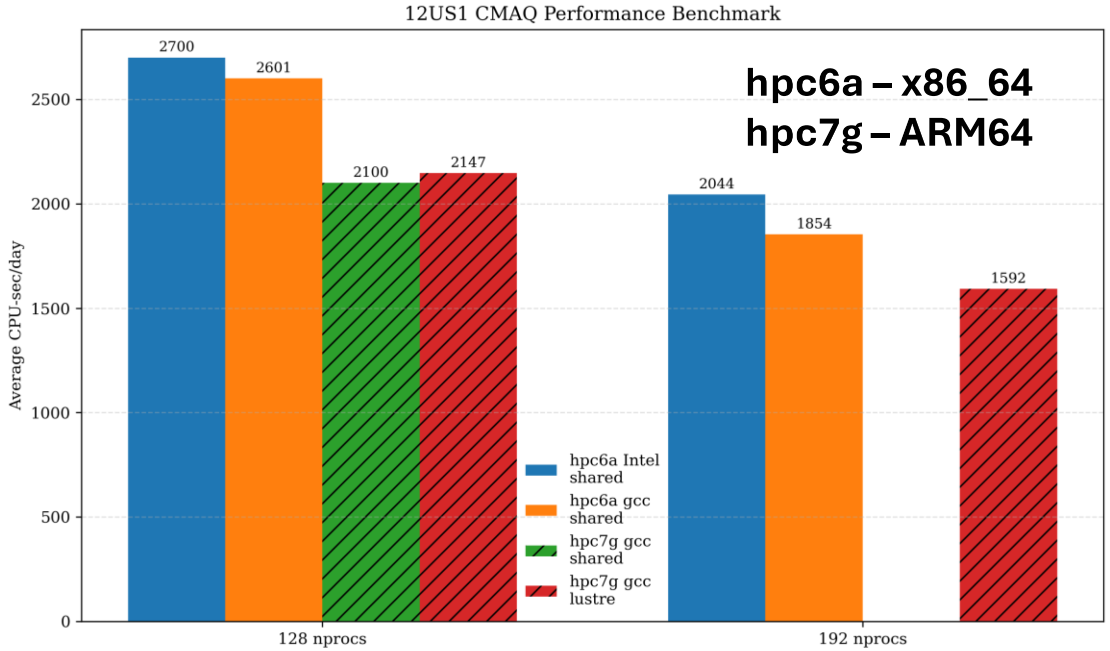
CMAS Data Warehouse

S3 Bucket	ncdump -k	Storage method	Extension	Size
CMAQ Input Data				
CMAQ Release Benchmark Data for Easy Download	tar.gz	archive	tar.gz	24.2 GiB
CMAQ CONUS-2 Benchmark Data	netcdf-3 classic	intact directory structure	.ncf	461.8 GiB
CMAQ Benchmark Data	netcdf-3 classic	intact directory structure	.nc	142.1 GiB (2018 12US1 data)
CMAS WWLLN Lightning Data	netcdf-3 classic	intact directory structure	.ioapi	31.4 GiB
EQUATES EPA's Air QUAlity TimE Series Project Data	netcdf-4 compressed	intact directory structure	.ncf	4.3 TiB
CMAQ 2018 Modeling Platform	netcdf-4 compressed	intact directory structure	.nc4	6.1 TiB
AMET Observational Input Data				
AMET Data	netcdf-3 classic	intact directory structure	.nc	834.6 GiB
SMOKE Emissions Platforms Data				
2019 Modeling Platform	netcdf-3 classic	intact directory structure	ncf.gz	2.0 TiB
2016v3 Modeling Platform	netcdf-3 classic	mix	.camx/.ncf/zip	35.2 TiB
SMOKE 2016 Modeling Platform	netcdf-3 classic	mix	.ncf/zip	392.6 GiB

2023 36US3 Input data can be obtained from here : https://zenodo.org/records/13737754

Source : Simon, H et al., (2025). Expedited modeling of burn events results (EMBER): A screening-level dataset of 2023 ozone fire impacts in the US. *Data in Brief*, *58*, 111208.

Benchmark Results (12US1)



Nodes	Instance Type	Cost ($/hour)
2 nodes (128 nprocs)	hpc6a.48xlarge	**$5.91**
2 nodes (128 nprocs)	hpc7g.16xlarge	**$3.44**
3 nodes (192 nprocs)	hpc7g.16xlarge	**$5.12**

- Increasing **nprocs from 128 → 192** improves runtime for all configurations
- **hpc7g + Lustre** gives the best performance among the Graviton(ARM64) runs
- **Gcc on hpc6a (x86_64)** is faster against intel, though **hpc7g (ARM64) performs better for both 128 and 192 nprocs**
- Annual simulation runtime ranges from **~7–11 days for the 12US1 domain** depending on processor count and cluster configuration

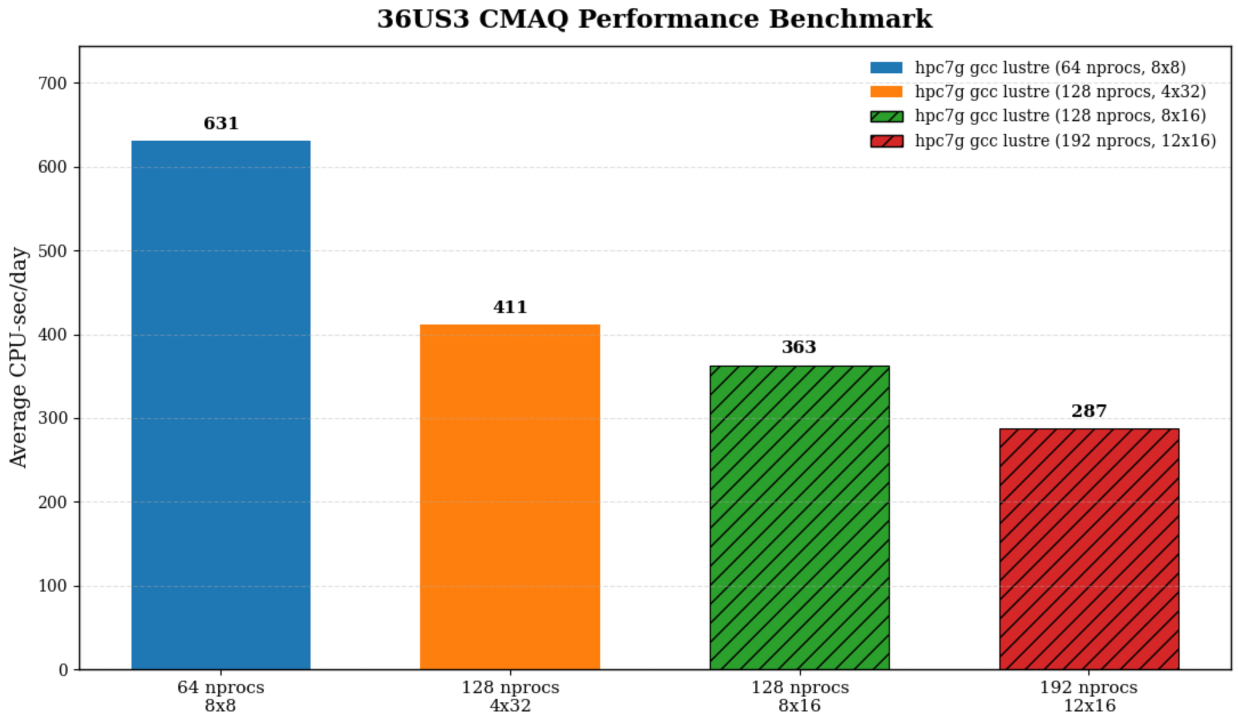
Note: Here cost includes login node + compute nodes

The 12US1 benchmarks demonstrate good parallel scaling, where increasing processor counts improves runtime and the hpc7g + Lustre configuration provides the best performance among the tested ARM-based setups.

Key Findings

- GCC **outperformed** Intel compilers on x86_64 **by 4-10%** for 12US1
- GCC on ARM64 was **~24% faster than GCC** on x86_64 for 12US1
- Hpc7g (ARM64) instances offered **strong cost-performance**
- ARM processors reduce CMAQ compute cost by **~50% compared with x86_64 configurations**
- **ARM64 architecture** selected for **production runs for 36US3**
- FSx Lustre improved **parallel I/O performance**

Benchmark Results (36US3 on ARM64)



Nodes	Instance Type	Cost ($/hour)
2 nodes (128 nprocs)	hpc7g.16xlarge	**$3.44**
3 nodes (192 nprocs)	hpc7g.16xlarge	**$5.12***

***Increasing the configuration from 2 nodes ($3.44/hr) to 3 nodes ($5.12/hr) increases compute cost by $1.68/hr (~49%), while improving runtime by only ~26%.**

- Increasing **nprocs from 64 → 192 significantly** improves runtime
- 128 nprocs already provides **strong performance improvements compared to 64**
- Best performance is **achieved with 192 nprocs** (3 nodes), reducing runtime from 3 to 1 day
- 36US3 domain covers a larger area but due to **coarse resolution,** runs are **much faster and cheaper than 12US1**
- Annual simulation runtime ranges **~1–3 days for the 36US3 domain**, depending on processor count and cluster configuration

Storage Requirements

Domain	Daily Storage	Annual Storage
36US3	~7 GB/day	~3 TB
12US1	~90 GB/day	~36 TB

Note : Daily Storage includes inputs (meteorology, emissions , initial conditions, boundary Conditions) and outputs (2D and 3D)

FSx Lustre Storage Costs (Annual Run)

Domain	Provisioned Size	Cost
36US3	2,400 GB	~$350
12US1	19,200 GB	~$3,000

FSx for Lustre Storage for 12US1 (1 Month)

- Storage capacity: **36 TB × 1024 = 36,864 GB**
- Data compression (50%): **18,432 GB saved**
- Effective storage: **18,432 GB**
- **Billed storage (rounded): 19,200 GB**

Cost Calculation (12US1)

- Price: **$0.145 per GB-month**
- Storage cost: **19,200 × 0.145 ≈ $3,000**
- **Total monthly cost: ≈ $3,000**

The cost estimate represents an annual CMAQ run, assuming the Lustre storage is provisioned for one month and includes cluster setup, benchmarking, simulation runtime, post-processing, and data transfer to Amazon S3 or local storage

Source: https://calculator.aws/#/createCalculator/FSxForLustre

Total Estimated Cost per Annual Run

- 12US1: ~$800 –$1,600 compute + ~$3,000 storage

(cost range is based upon architecture, processor, domain decomposition and compiler used)

- 36US3: ~$120 –$220 compute + ~$350 storage
- Storage corresponds to ~1 month Lustre usage
- Actual Lustre usage time typically shorter, based on user needs

Final Cost Summary

Domain	Compute Cost	Storage Cost	Total
12US1	$800 – $1,600	**$3,000**	$3,800 – $4,600
36US3	$120 – $220	**$350**	$470 – $570

Storage Strategy

- FSx for Lustre used during active simulations
- Provides high-performance parallel filesystem
- In our study, since we had multiple emissions scenarios that shared the same input data (meteorology, IC/BC), we created a shared drive. Once the simulation was completed, we copied onto shared and then post-processed and then archived to s3 bucket
- **Recommendation:**
  - **Store active simulation outputs in Amazon S3 Standard for fast access, while long-term archived data can be moved to S3 Glacier storage to significantly reduce storage costs (s3 costs in next slide)**

S3 bucket costing

Storage Class	Typical Use Case	Retrieval Speed	Cost (USD / GB / Month)
S3 Standard	Frequently accessed simulation outputs and active analysis data	Milliseconds	**$0.023**
S3 Standard – Infrequent Access	Data accessed occasionally but requiring fast retrieval	Milliseconds	**$0.0125**
S3 Glacier Instant Retrieval	Archive data accessed occasionally (few times per year)	Milliseconds	**$0.004**
S3 Glacier Flexible Retrieval	Long-term backups and archives	Minutes–Hours	**$0.0036**
S3 Glacier Deep Archive	Very long-term archival (rarely accessed)	Up to 12 hours	**$0.00099**

Using S3 Glacier Deep Archive can reduce storage cost by over 95% compared to S3 Standard for long-term archival data.

Source : https://aws.amazon.com/s3/pricing/

36US3 Study Workflow on AWS (hpc7g)

Study design: 6 months simulated (April–September) for 3 years, each with 3 emissions scenarios | Total = 9 CMAQ runs

1. Environment Setup
- Python virtual env
- AWS CLI
- AWS ParallelCluster

2. Create ParallelCluster
- ARM64 or x86_64
- c7g.large and hpc7g
- SLURM + EFA
- FSx for Lustre as needed

3. Copy Inputs from local
- Stage CMAQ

4. Run 6-Month CMAQ
- Submit job to SLURM
- Execute simulation

5. Copy Raw Outputs to S3
- Archive outputs
- Post process

6. Download COMBINE and hr2day post processed outputs
- Transfer results locally
- Prepare for AMET

7. QA / Evaluation
- Load outputs into AMET
- Review model performance

8. Clean Storage
- Remove FSx if needed

9. Repeat
- Next scenario
- Continue to other runs

3 scenarios x 3 years = 9 runs

Lesson Learned (How to save costs)

- **Estimated costs ~2x lower** (~$2,000 vs ~$4,000 actual), as earlier costs increased due to CMAQ version changes, repeated runs, and no Lustre compression
- **Storage costs reduced** using compressed, cost-efficient Lustre (≈4x cheaper than 50 MBps/TiB) with **Persistent (125 MBps/TiB)** or **Scratch (200 MBps/TiB)** throughput
- Use **small root and /shared volumes**, keep all **I/O on Lustre**, and **disable Lustre when not in use** to save costs
- **Cost calculator example** shown for **Persistent Lustre** with **50 and 125 MBps/TiB** configurations

FSx for Lustre Storage Cost (Monthly)
Storage capacity: 3 TB × 1024 = **3,072 GB**
Data compression saving (50%): 3,072 × 0.5 = **1,536 GB saved**
Effective storage: 3,072 − 1,536 = **1,536 GB**
Billed Storage : 2400 GB (either 1.2 TB or chunks of 2.4 TB)

50 Mbps/TiB
Price: **$0.60 per GB-month**
Storage cost: 2,400 GB × $0.60 = ~**$1500**

125 Mbps/TiB
Price: **$0.145 per GB-month**
Storage cost: 2,400 GB × $0.145 = ~**$350**

Cost and Performance Advantages of ARM64 (Graviton 3E)

- Better price to performance ratio
  - ARM64 was **24% faster than x86_64** in CMAQ 12US1 benchmarking using the **gcc compiler**
- Lower compute and login node cost
  - Over **50% savings for login nodes** and **40% savings for compute nodes** compared to Intel x86_64 instances
- Improved Efficiency for CMAQ
  - Provides faster simulations with **lower overall cloud computing costs**

Recommended Configuration

- Compute: hpc7g instances (Also depends upon your application)
- Compiler: GCC
- Storage: FSx for Lustre
- Delete Lustre after simulation
  - move raw CMAQ outputs to s3 glacial

Advantages of AWS ParallelCluster

- Scalable HPC environment
- Automated cluster management
- Integration with AWS storage services
- Suitable for research workflows

Implications for CMAQ Community

- Cloud HPC enables flexible research infrastructure
- ARM processors reduce simulation cost
- Optimized storage strategies minimize expenses
- Useful for multiple long-term simulations

Future Work

- Benchmark CMAQ-DDM with multiple sensitivity variables
- Evaluate additional new compute instances and storage options
  - As AWS releases new compute nodes (e.g., hpc8a)
  - As EPA releases new versions of CMAQ
- Automate CMAQ workflows in cloud
  - Provisioning compute nodes, pull input data from s3bucket, submitting jobs, post processing, archive outputs to to s3 bucket or local drive, removal of resources after job completion
- Improve cost optimization strategies
  - Efficient storage options, choosing right size compute resources, use HPC instances, auto scaling and shut down idle resources, optimize data transfer (compression),cost allocation tags

References

- CMAQ Parallel Cluster Tutorial
  - https://pcluster-cmaq.readthedocs.io/en/latest/
  - https://github.com/CMASCenter/pcluster-cmaq
- CMAQ on AWS workshop
  - https://catalog.workshops.aws/cmaq-tutorial/en-US/
- Efstathiou et al., (2024). Enabling high-performance cloud computing for the Community Multiscale Air Quality Model (CMAQ) version 5.3. 3: performance evaluation and benefits for the user community. *Geoscientific model development*, *17*(18), 7001-7027.
- CMAQ Documentation
  - https://github.com/USEPA/CMAQ
- AWS ParallelCluster Documentation
  - https://github.com/aws/aws-parallelcluster
- AWS FSx for Lustre Pricing
  - https://aws.amazon.com/fsx/lustre/pricing/

Acknowledgements

- The U.S. EPA, through its Office of Research and Development, partially funded and collaborated in the research described here under Contract 68HERD21A002 to UNC Chapel Hill
- We acknowledge Amazon Web Services (AWS) for support of the CMAS Center Data Warehouse through its Open Data Program
- The U.S. EPA CMAQ Development Team

Questions

- Thank you
- Questions?

Additional Slides

Instructions to add and remove lustre drive

Updating the CMAQ Cluster

1. Stop the compute fleet before updating the cluster

pcluster update-compute-fleet --cluster-name cmaq --status STOP_REQUESTED

2. Update the cluster configuration to add FSx for Lustre storage

pcluster update-cluster -n cmaq -c hpc7g.16xlarge.fsx.yaml

3. Check the compute fleet status

pcluster describe-compute-fleet --cluster-name cmaq

Proceed once the status indicates the update is complete.

4. After the simulation and data transfer are complete, remove the FSx for Lustre storage

pcluster update-cluster -n cmaq -c hpc7g.16xlarge.nofsx.yaml

5. Restart the compute fleet after the cluster update

pcluster update-compute-fleet --cluster-name cmaq --status START_REQUESTED

Example hpc7g.16xlarge.fsx.yaml file

```yaml
Region: us-east-1
Image:
  Os: ubuntu2204
HeadNode:
  InstanceType: c7g.large
  Networking:
    SubnetId: subnet-07e412a0ca36ad7fa
  DisableSimultaneousMultithreading: true
  Ssh:
    KeyName: ubuntu
  LocalStorage:
    RootVolume:
      Size: 200
      Encrypted: false
      VolumeType: gp3
      DeleteOnTermination: false
Scheduling:
  Scheduler: slurm
  SlurmSettings:
    ScaledownIdletime: 3
  SlurmQueues:
    - Name: queue1
      CapacityType: ONDEMAND
      Networking:
        SubnetIds:
          - subnet-07e412a0ca36ad7fa
        PlacementGroup:
          Enabled: true
      ComputeResources:
        - Name: compute-resource-1
          InstanceType: hpc7g.16xlarge
          MinCount: 0
          MaxCount: 12
          DisableSimultaneousMultithreading: true
          Efa:
            Enabled: true
            GdrSupport: false
SharedStorage:
  - MountDir: /shared
    Name: ebs-shared
    StorageType: Ebs
    EbsSettings:
      Encrypted: false
      Size: 1200
  - MountDir: /fsx
    Name: name2
    StorageType: FsxLustre
    FsxLustreSettings:
      StorageCapacity: 1200
~
```

Delete this and save it as other yaml with the steps from previous slide save it with nofsx tag

Settings for compression and choosing persistent drive 2 to save cost

- SharedStorage:
```
  - Name: FsxLustre0
    StorageType: FsxLustre
    MountDir: /fsx
    FsxLustreSettings:
      DeletionPolicy: Retain
      StorageCapacity: 1200
      DeploymentType: PERSISTENT_2
      PerUnitStorageThroughput: 125
      DataCompressionType: LZ4
```