# CMAQv5.4 on AWS

**Elizabeth Adams**[1], Christos Efstathiou[2], Carlie Coats[1], Robert Zelt[3], Mark Reed[3], John McGee[3], Kristen M. Foley[4], Fahim I. Sidi[4], David C. Wong [4], Steven Fine[5], Saravanan Arunachalam[1]

*[1]Institute for the Environment, The University of North Carolina at Chapel Hill, NC, USA*
*[2] Formerly with the Institute for the Environment, now at PSE Healthy Energy*
*[3] Research Computing, Information Technology Services, The University of North Carolina at Chapel Hill, NC, USA*
*[4] Center for Environmental Measurement and Modeling, Office of Research and Development, U.S. Environmental Protection Agency, Research Triangle Park, NC, USA*
*[5] Formerly with the Office of Air and Radiation, U.S. Environmental Protection Agency, Washington, DC, USA*

**Presented Online**
**August 2023**

UNC | INSTITUTE FOR THE ENVIRONMENT

# Tutorial Objectives


- Enable use of high-performance computing (HPC) on the cloud for current and future CMAQ releases
  - Provide guidance for setting up, evaluating, and optimizing model performance.
  - Provide Benchmark/QA Suite for “target end-users” to take advantage of different AWS offerings, while demonstrating ways of controlling the costs.

Definitions:

- Single VM is a single Linux virtual machine created using Amazon Web Services (AWS) Elastic Compute Cloud (EC2) run CMAQ in the cloud (virtual machines or 'instances').
- AWS ParallelCluster is an AWS supported open source cluster management tool that helps you to deploy and manage high performance computing (HPC) clusters in the AWS Cloud. It automatically sets up the required compute resources, scheduler, and shared file system to run CMAQ.

UNC INSTITUTE FOR THE ENVIRONMENT

# Why the Cloud is ideal for running CMAQ!

## Train new users

- Software and input data are pre-installed, and output data for verification is provided on the CMAS Registry on the AWS Open Data Program.
- EC2 instances are provisioned within five minutes.
- Avoid wait times in queue encountered on shared system.
- Avoid costly computer purchases.
- Learn how to use performance metrics to improve hardware and software.

## Right size compute resource utilization/spending

- Based on domain size/resolution, required time to solution and budget.
- Use preconfigured CMAQ resources (in the form of an Amazon Machine Image [AMI] with EC2 instance of different sizes within same family (AMD - c6a).
- **One thread per core**: Disable multi-threading by specifying a single thread per core when the VM is created. This is recommended for high performance computing (HPC) workloads. AWS EC2 instance types use two threads per core virtual CPU (vCPU), unless hyperthreading is disabled.

| CMAQ Benchmark | Grid Domain | EC2 (cores, vcpu) | Time (hr) | $ (Spot/OnDemand) |
|---|---|---|---|---|
| 12LISTOS-Training | (25x25x35) | c6a.2xlarge (4, 8) | .14 | .04 / .29 |
| 12NE3 | (100x100x35) | c6a.8xlarge (16, 32) | .279 | .279 / .341 |
| 12NE3 | (100x100x35) | c6a.48xlarge (96, 192) | .065 | .366 / .48 |

# Why the Cloud is ideal for running CMAQ! (continued)

UNC INSTITUTE FOR THE ENVIRONMENT

**"Accelerate time to results for compute-heavy workloads" ***

- Autoscaling capability of ParallelCluster (add VMs to run on more cores)
  - Expand the extent of domains and use higher resolution with access to more compute cores, or add more cores to get faster completion time.
  - Quickly obtain resources for even the most demanding simulations, such as two-way coupled WRF-CMAQ, CMAQ-ISAM, CMAQ-DDM3D.
- Benchmark CMAQ on different EC2 instances (Intel, AMD, ARM, Nvidia), Operating Systems, and Compilers (Intel, GCC), and MPI implementations (OpenMPI, Intel MPI)
  - Quickly install and build libraries and CMAQ using csh scripts in the developer guide.
  - Modify run scripts to match the cores available.
  - Newer EC2 instances are faster, less costly, and more energy efficient.
- Expandability of low-cost data storage
  - Facilitates sharing data on S3 Storage Buckets.
  - Members of the CMAS Community are encouraged to contribute publicly available data via the CMAS Data Warehouse on AWS.
- Infrastructure as code (IaC)
  - Extend FAIR (Findability, Accessibility, Interoperability, and Reuse of digital data) principles to research computing infrastructure.
  - Barker, M., Chue Hong, N.P., Katz, D.S. *et al.* Introducing the FAIR Principles for research software. *Sci Data* **9**, 622 (2022). https://www.nature.com/articles/s41597-022-01710-x

# CMAQ Data Warehouse on AWS

UNC INSTITUTE FOR THE ENVIRONMENT

CMAS Data Warehouse on AWS collects and disseminates 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

# Tutorial: CMAQ on AWS for End-Users

UNC INSTITUTE FOR THE ENVIRONMENT

**Online tutorial with background info, hands-on instructions, performance and cost analysis, etc:**

- CMAQ on AWS Tutorial:

https://pcluster-cmaq.readthedocs.io/en/latest/index.html

**Target end-users (not mutually exclusive):**

- Novice users, early-stage model testing.
- Advanced users, who need to fully harness the HPC potential of CMAQ through annual/high resolution simulations, operational use.

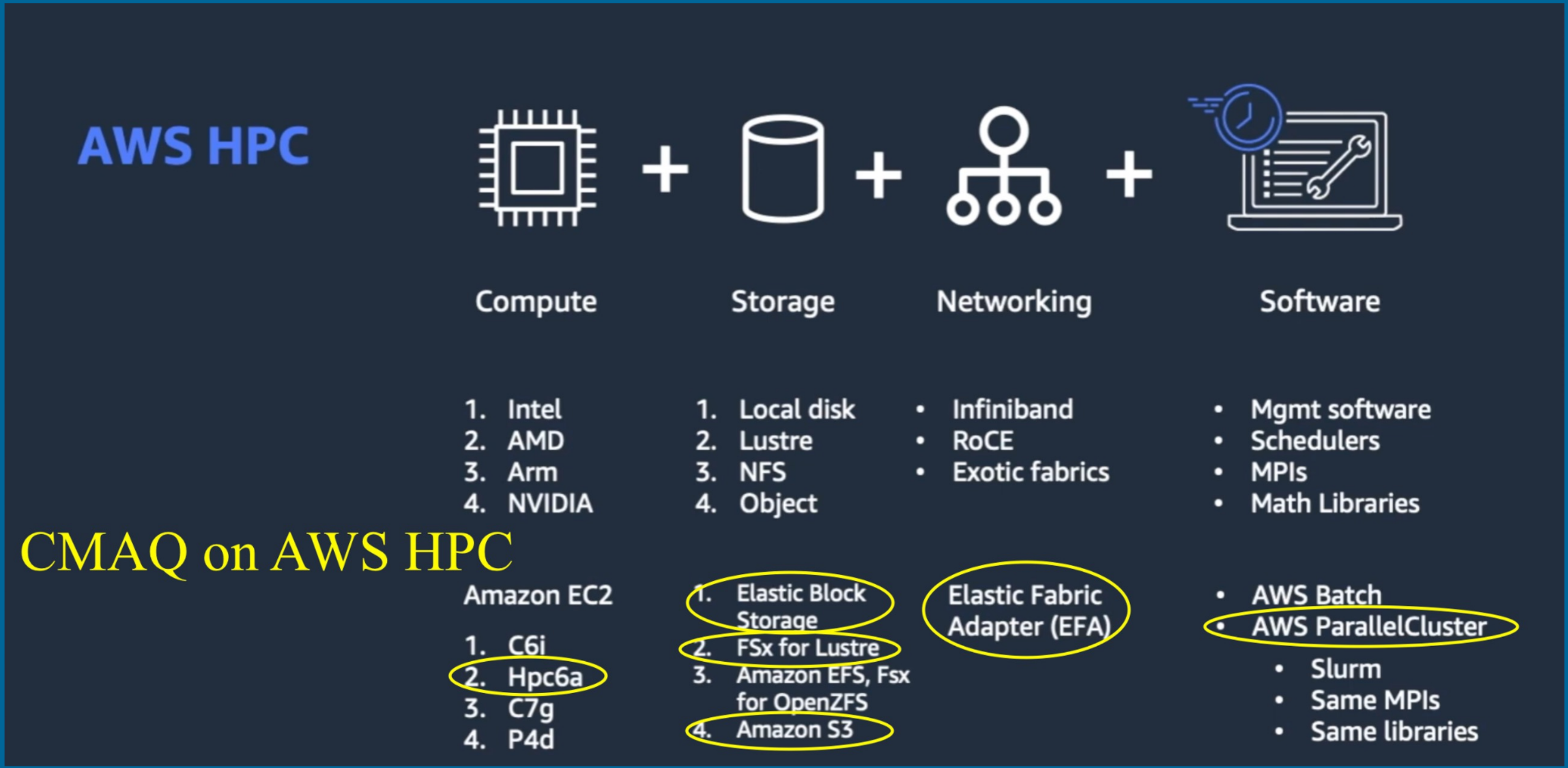
Focus of Webinar: highlight update/changes to CMAQ on AWS Tutorial

- EPA OAR and ORD staff currently use a development environment that is restricted to using the AWS us-east-1 region due to firewall requirements, so the examples will use AWS c6a resources, as hpc6a is only available in AWS us-east-2b.
- Cloud Computing Workgroup members may not have access to the AWS Management Console due to internal IT restrictions, so examples will use the AWS Command Line.
- Updates to the tutorial since the previous CMAS Conference (Oct. 2022) include updating code: CMAQv5.4+ and adding benchmarks: 12US1, 12NE3, and 12LISTOS-Training.

# AWS HPC Resources Used in Webinar



AWS Elastic Compute Cloud (EC2) Instance Types are comprised of varying combinations of CPU, memory, storage, and networking capacity




AWS HPC Tech Short Video

https://www.youtube.com/watch?v=KHx22oJSNso

* Image from above video, but was annotated to highlight resources used in webinar in yellow.

# EC2 Instance Type Used in Webinar



c6a.48xlarge (192 vcpu) - Third generation AMD EPYC (available everywhere)

- CPU option: disable hyperthreading (Use 96 cores/instance)
- Market pricing (Spot)
  - When you use Spot Instances, you must be prepared for interruptions. Amazon EC2 can interrupt your Spot Instance when the demand for Spot Instances rises or when the supply of Spot Instances decreases.
  - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances-history.html
- On-demand pricing
  - Let's you pay for compute capacity by the hour or second (minimum of 60 seconds) with no long-term commitments.
  - https://aws.amazon.com/ec2/pricing/on-demand/

https://aws.amazon.com/blogs/aws/new-amazon-ec2-c6a-instances-powered-by-3rd-gen-amd-epyc-processors-for-compute-intensive-workloads/

# Storage Used in Webinar



## 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.

- gp3 (supports selecting maximum of 16,000 IOPS, and 1000 MiB/s throughput) 3,000 IOPS free and $0.005/provisioned IOPS-month over 3,000 (default for Parallel Cluster /shared volume).
- 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

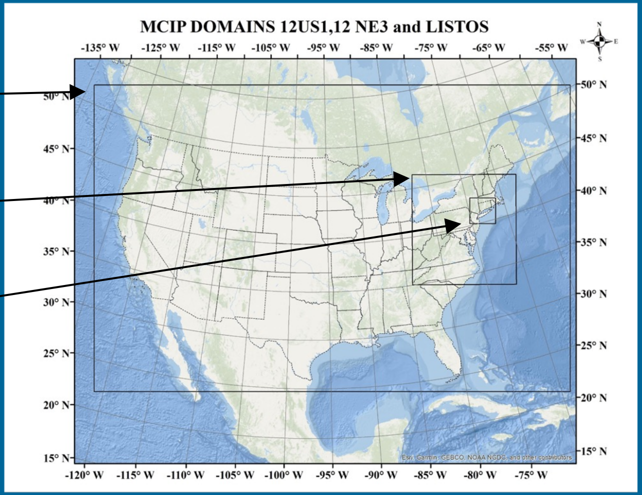
# Network Options used in Webinar

- Elastic Fabric Adapter enabled
- Placement group enabled
- 50 Gbps for network bandwidth
- High throughput, low-latency network

# CMAQ Benchmark Cases Used in Webinar

CMAQv5.4 (cb6r5_ae7_aq)

- 2-day 12x12 km 12US1 (459x299x35), 85 GB
  - compressed netCDF4 converted to classic netCDF3
- 1-day 12x12 km 12NE3 (100x105x35), 56 GB
  - classic netCDF3
- 3-day 12x12 km 12LISTOS-Training (25x25x35), 1.9 GB
  - classic netCDF3




## Benchmark data available on CMAS Data Warehouse

- Ingress and egress fees to S3 Bucket are waived as part of the CMAS Center Registry on AWS Open Data Program.
- Scripts are provided in the tutorial to load the benchmark data from the S3 bucket to the EBS /shared volume or to use “lazy loading” technique available on the Lustre FSx filesystem.

# Read-the-docs Tutorial Structure

- Single VM
  - Configure VM using AWS Console using "CMAQ AMI"
  - Configure VM using AWS CLI using "CMAQ AMI"
- Create a Parallel Cluster (many VMs) using "CMAQ Snapshot"
- Performance Cost and Optimization
- Developers Guide (How to install CMAQ software and libraries)
- Post-processing and saving to S3 Bucket
- Logout and Terminate Cluster
- Additional Resources
- Future Work
- How to contribute to this work

* Chapters highlighted in yellow are demonstrated in this Webinar

# Notes on accounts & hardware availability – AWS

UNC INSTITUTE FOR THE ENVIRONMENT

- The AWS Management Console is used to establish administrative and user-level permissions.
- To reproduce these instructions, the AWS administrator needs to create policies to allow the use of c6a.48xlarge EC2 instances in the east-us-1 region, both On-Demand and Spot instances, and the EBS and Lustre FSx filesystems.
- Recommend requesting a limit increase for access to the hpc6a.48xlarge (AWS) instances in the us-east-2 region. This instance type is only available in limited regions, but it is priced at 60% of the cost of the c6a.48xlarge instance that is available in all regions.
- Command line options may differ depending on how your system is configured to connect to AWS resources.
- This webinar uses the c6a.48xlarge ec2 instance, but the CMAQ on AWS Tutorial provides instructions for hpc7g.16xlarge as those instance types are less expensive.

# CMAQ on AWS HPC Tutorial includes:

UNC INSTITUTE FOR THE ENVIRONMENT

- C-shell scripts to install and compile libraries and CMAQ
- Use of environment modules to load OpenMPI and other libraries
  - Custom environment modules can be created to load libraries necessary to run CMAQ using either netcdf3 classic (*.nc) or netCDF4 (*.nc4) compressed files (this is under development).
- Run scripts provided for scaling tests
  - Scaling tests are needed because as more cores are added for a given CMAQ domain, the amount of work done per core decreases and the communication between cores increases. MPI communication time to compute time ratio increases, and the overall performance decreases. In summary, at some point, the model run time won't decrease much as number of cores increases.
- Run scripts for each compute node instance type
  - NPCOL, NPROW, and Slurm sbatch settings must be set to the number of cores available on each EC2 instance. c6a.48xlarge has 96 cores, to run CMAQ using 192 cores with hyperthreading disabled, set NPCOL x NPROW and --nodes=2, --ntasks-per-node=96, cores = nodes x ntasks-per-node. The maximum number of cores on a single VM is 96.

Run script examples:

| | |
|---|---|
| ## Single VM c6a.48xlarge (96 cores) | ## Parallel Cluster c6a.48xlarge (2x96=192 cores) |
| ## Run interactively - no Slurm | #SBATCH --nodes=2 |
| | #SBATCH --ntasks-per-node=96 |
| @NPCOL = 12; @ NPROW = 8 | @ NPCOL = 16; @ NPROW = 12 |

- Run scripts for each benchmark case and CMAQ version

# AWS HPC on Single VM or Parallel Cluster

Tutorial provides two command line interface (CLI) methods to deploy cloud resources on AWS.

- **Single Virtual Machine (VM)**
  - The AWS 3.0 CLI and a JSON Template are used to create, login, and run CMAQ on a single EC2 instance.
- **ParallelCluster (many VMs)**
  - A YAML file specifies the configuration of the cluster.
  - AWS 3.0 CLI is used to create and login to the head node of the Parallel Cluster.
  - Slurm workload manager is used to run CMAQ on the EC2 instances that were specified as compute nodes.

Benchmarking

Tutorial provides CMAQ benchmarks and results for different EC2 instance types, networking, and storage options.

To obtain the scripts provided in the tutorial use:

```
git clone -b main https://github.com/CMASCenter/pcluster-cmaq
cd pcluster-cmaq
```

# AWS Single Virtual Machine Workflow

UNC INSTITUTE FOR THE ENVIRONMENT

Step 1. Install AWS 3.0 CLI on your local machine

Step 2. Configure a VM to run CMAQv.5.4 with JavaScript Object Notation (JSON) file.

Step 3. Use AWS CLI to create the instance. (highlighted items are unique to your account)

```
aws ec2 run-instances --debug --key-name cmaqv5.4 --security-group-ids launch-wizard-179 --region
us-east-1 --ebs-optimized --cpu-options CoreCount=4,ThreadsPerCore=1 --cli-input-json
file://json/runinstances-config.gp3.c6a.2xlarge.json
```

Step 4. Obtain IP address.

```
aws ec2 describe-instances --region=us-east-1 --filters "Name=image-id,Values=ami-051ba52c157e4070c" | grep
PublicIpAddress
```

```
or (for EPA Sandbox): grep PrivateIPAddress
```

Step 5. Login to c6a.2xlarge EC2 Instance.

```
ssh -v -Y -i ~/downloads/cmaqv5.4.pem ubuntu@XX.XXX.XX
or (for EPA Sandbox) use putty to login to private IP address
```

Step 6. Load Environment Modules

```
module load ioapi-3.2/gcc-11.3.0-netcdf  mpi/openmpi-4.1.2  netcdf-4.8.1/gcc-11.3
```

Step 7. Pre-warm data on /shared disk (obtain data from snapshot for large model domains)

Step 8. Review run script, and edit to run for 1 day

```
cd /shared/build/openmpi_gcc/CMAQ_v54+/CCTM/scripts/
vi run_cctm_2018_12US1_listos.csh
```

Step 9. Run CMAQ interactively from the command line using 4 pes (NPCOLxNPROW=2x2)

```
./run_cctm_2018_12US1_listos.csh |& tee ./run_cctm_2018_12US1_listos.c6a.2xlarge.4pe.log
```

Step 10. After run, shutdown the Virtual Machine.

```
  aws ec2 describe-instances --region=us-east-1 | grep InstanceId
  aws ec2 terminate-instances --region=us-east-1 --instance-ids i-xxxxxxxxxxkj
```

# AWS Single VM JSON File



JSON file

```
{
        "DryRun": false,
        "MaxCount": 1,
        "MinCount": 1,
        "InstanceType": "c6a.2xlarge",
                                   # InstanceType specifies an EC2 instance
        "ImageId": "ami-051ba52c157e4070c",
                                   # ImageID specifies a public AWS AMI with CMAQv5.4 installed

        "InstanceMarketOptions": {
        "MarketType": "spot"            # MarketType specifies the price: spot or ondemand
        },
        "TagSpecifications": [
        {
        "ResourceType": "instance",
        "Tags": [
                {
                "Key": "Name",
                "Value": "EC2c6a2xlargeCMAQv54gp3"
                }
        ]
        }
        ]
}
```

# Use htop to view performance

UNC INSTITUTE FOR THE ENVIRONMENT

htop

```
  0[||||||||||||||||||||100.0%]    1[||||||||||||||||||||100.0%]    2[||||||||||||||||||||100.0%]    3[||||||||||||||||||||100.0%]
Mem[||||||||||||                     832M/15.3G]   Tasks: 45, 57 thr; 4 running
Swp[                                     0K/0K]   Load average: 3.85 1.70 0.71
                                                  Uptime: 01:07:03

  PID USER      PRI  NI  VIRT   RES   SHR S CPU%▽MEM%   TIME+  Command
10780 ubuntu     20   0  620M  166M 21644 R 100.  1.1  2:36.21 /shared/build/openmpi_gcc/CMAQ_v54+/CCTM/scripts/BLD_CCTM_v54_gcc/CCTM_v54.ex
10781 ubuntu     20   0  618M  164M 21676 R 100.  1.1  2:36.38 /shared/build/openmpi_gcc/CMAQ_v54+/CCTM/scripts/BLD_CCTM_v54_gcc/CCTM_v54.ex
10782 ubuntu     20   0  616M  162M 21664 R 99.7  1.0  2:36.16 /shared/build/openmpi_gcc/CMAQ_v54+/CCTM/scripts/BLD_CCTM_v54_gcc/CCTM_v54.ex
10779 ubuntu     20   0  639M  185M 21592 R 99.0  1.2  2:35.87 /shared/build/openmpi_gcc/CMAQ_v54+/CCTM/scripts/BLD_CCTM_v54_gcc/CCTM_v54.ex
    1 root       20   0  163M 13064  8384 S  0.0  0.1  0:03.09 /sbin/init
  185 root       19  -1 64192 15964 14868 S  0.0  0.1  0:00.15 /lib/systemd/systemd-journald
  228 root       RT   0  282M 27100  9072 S  0.0  0.2  0:00.18 /sbin/multipathd -d -s
  230 root       20   0 23424  6904  4740 S  0.0  0.0  0:00.15 /lib/systemd/systemd-udevd
  232 root       20   0  282M 27100  9072 S  0.0  0.2  0:00.00 /sbin/multipathd -d -s
  257 root       RT   0  282M 27100  9072 S  0.0  0.2  0:00.00 /sbin/multipathd -d -s
  258 root       RT   0  282M 27100  9072 S  0.0  0.2  0:00.00 /sbin/multipathd -d -s
  259 root       RT   0  282M 27100  9072 S  0.0  0.2  0:00.00 /sbin/multipathd -d -s
  260 root       RT   0  282M 27100  9072 S  0.0  0.2  0:00.11 /sbin/multipathd -d -s
  261 root       RT   0  282M 27100  9072 S  0.0  0.2  0:00.00 /sbin/multipathd -d -s
  444 systemd-n  20   0 16248  8176  7172 S  0.0  0.1  0:00.04 /lib/systemd/systemd-networkd
  446 systemd-r  20   0 25528 12788  8596 S  0.0  0.1  0:00.05 /lib/systemd/systemd-resolved
  490 root       20   0  2812  1152  1068 S  0.0  0.0  0:00.00 /usr/sbin/acpid
  495 root       20   0  7284  2844  2620 S  0.0  0.0  0:00.00 /usr/sbin/cron -f -P
```

If the CPU usage is not ~ 100%, there may be an I/O issue that can be solved by pre-warming the volume.

## After run is complete, view timing:

```
tail -n 15 run_cctm_2018_12US1_listos.c6a.2xlarge.4pe.1day.log
```

```
   ***** CMAQ TIMING REPORT *****
==================================
Start Day: 2018-08-05
End Day:   2018-08-05
Number of Simulation Days: 1
Domain Name:               2018_12Listos
Number of Grid Cells:      21875  (ROW x COL x LAY)
Number of Layers:          35
Number of Processes:       4
   All times are in seconds.

Num  Day                   Wall Time
01   2018-08-05   228.4
            Total Time = 228.40
             Avg. Time = 228.40
```

# AWS ParallelCluster Workflow



Step 1. Create and activate a virtual environment to activate the ParallelCluster CLI.

```
source ~/apc-ve/bin/activate
source ~/.nvm/nvm.sh
```

Step 2. Configure cluster using AWS 3.0 CLI.

```
pcluster configure -config your-cmaq.yaml
```

Step 3. Edit YAML file to use snapshot with pre-loaded software and lustre filesystem.

Step 4. Create ParallelCluster and login to head node using AWS 3.0 CLI.

```
pcluster create-cluster --cluster-configuration c6a.large-
48xlarge.ebs_unencrypted_installed_public_ubuntu2004.fsx_import_ondemand.y
aml --cluster-name cmaq --region us-east-1
```

Step 5. Check the status of the cluster

```
pcluster describe-cluster --region=us-east-1 --cluster-name cmaq
```

Step 6. Log in using pcluster command line.

```
pcluster ssh -v -Y -i ~/cmas.pem --region=us-east-1 --cluster-name cmaq
```

Step 6. Verify the CMAQ software on /shared and preload input data on lustre /fsx.

Step 7. Run CMAQ by submitting to Slurm queue.

Step 8. Shut down the ParallelCluster.

```
pcluster delete-cluster --region=us-east-1 --cluster-name cmaq
```

# AWS YAML File


```yaml
Region: us-east-1
Image:
 Os: ubuntu2004
HeadNode:
 InstanceType: c6a.xlarge
 Networking:
  SubnetId: subnet-018fdd14871ed2b0d
 DisableSimultaneousMultithreading: true
 Ssh:
  KeyName: cmas-east-1
 LocalStorage:
  RootVolume:
   Encrypted: true
Scheduling:
 Scheduler: slurm
 SlurmSettings:
  ScaledownIdletime: 5
 SlurmQueues:
  - Name: queue1
   CapacityType: ONDEMAND
   Networking:
    SubnetIds:
     - subnet-018fdd14871ed2b0d
    PlacementGroup:
     Enabled: true
   ComputeResources:
    - Name: compute-resource-1
     InstanceType: c6a.48xlarge
     MinCount: 0
     MaxCount: 10
     Efa:
      Enabled: true
      GdrSupport: false
SharedStorage:
 - MountDir: /shared
  Name: ebs-shared
  StorageType: Ebs
  EbsSettings:
   Encrypted: false
   SnapshotId: snap-05a36eeec1f5267bd
 - MountDir: /fsx
  Name: name2
  StorageType: FsxLustre
  FsxLustreSettings:
   StorageCapacity: 1200
   ImportPath: s3://cmas-cmaq
```


Region: us-east-1

HeadNode:
InstanceType: c6a.large

ComputeResources:
InstanceType: c6a.48xlarge
MaxCount: 10

StorageType:
EbsSettings:
Encrypted: false
SnapshotId: snap-05a36eeec1f5267bd (CMAQ Software)

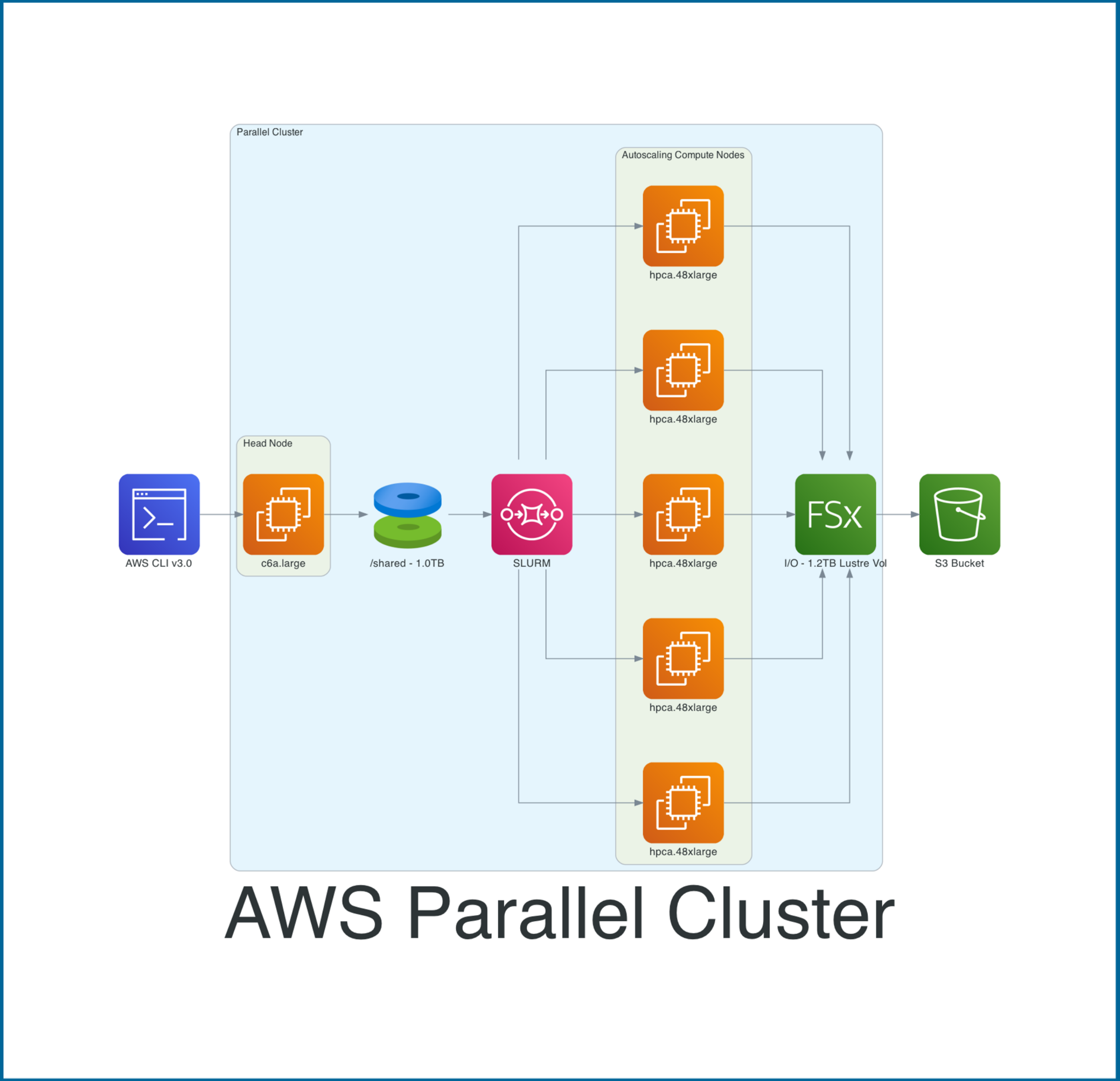
FsxLustre

FsxLustreSettings:

StorageCapacity: 1200 (Input/Output Data)

ImportPath: s3://cmas-cmaq

# HPC ParallelCluster Diagram



# HPC Autoscaling on AWS

UNC INSTITUTE FOR THE ENVIRONMENT

- SLURM auto-scaling on AWS ParallelCluster allows CMAQ to be run on any number of cores, and/or to run multiple CMAQ runs (e.g. base and zero-out), by submitting jobs to the queue.

- Users submit Slurm job scripts, and the required number of compute nodes are provisioned and ready to start the job within a few minutes. Compute nodes only incur costs when running.

- If the cluster remains idle for more than a preset time, the cluster will automatically shut down the compute nodes after job completion.

- The head node remains running and available to start jobs when the user uses sbatch to submit to the Slurm queue. The Lustre file system (/fsx) accumulates charges as long as the head node is up. To reduce costs, terminate the cluster after the workload is complete. (cost is $384 if you leave it on for a month)

# Creating ParallelCluster

## Use AWS CLI 3.0 commands

1. Activate the virtual environment to use the ParallelCluster command line.

```
source ~/apc-ve/bin/activate
source ~/.nvm/nvm.sh
```

2. Upgrade to get the latest version of ParallelCluster.

```
python3 -m pip install --upgrade "aws-parallelcluster"
```

3. Verify that the ParallelCluster AWS CLI is installed by checking the version.

```
pcluster version
{
  "version": "3.6.0"
}
```

4. Use YAML file to create cluster.

```
pcluster create-cluster --cluster-configuration c6a.xlarge-48xlarge.ebs_unencrypted_installed_public_ubuntu2004.fsx_import_ondemand.yaml --cluster-name cmaq --region us-east-1
```

5. Check status of ParallelCluster.

```
pcluster describe-cluster --region=us-east-1 --cluster-name cmaq
```

# Connecting to ParallelCluster

## Use ParallelCluster AWS CLI 3.0 commands

1. List clusters

```
pcluster list-clusters --region=us-east-1
```

Output:


```
{
"clusters": [
 {
        "clusterName": "cmaq",
        "cloudformationStackStatus":
"UPDATE_COMPLETE",
       "region": "us-east-1",
       "version": "3.6.0",
       "clusterStatus": "UPDATE_COMPLETE"
}
 ]
}
```


2. Login to cluster in region us-east-1

```
pcluster ssh -v -Y -i ~/cmas.pem --cluster-name cmaq
--region=us-east-1
```

# Configure shell and input data

1. Change shell

```
sudo usermod -s /bin/tcsh ubuntu
exit
```

2. Re-login

```
pcluster ssh -v -Y -i ~/cmas.pem --cluster-name cmaq --region=us-east-1
```

3. Copy file to .cshrc, and source

```
cp /shared/pcluster-cmaq/install/dot.cshrc.pcluster ~/.cshrc
csh
```

2. Load environment modules

```
module load openmpi libfabric-aws openmpi ioapi-3.2 netcdf-4.8.1
```

3. Preload input data on /fsx

```
nohup find /fsx/ -type f -print0 | xargs -0 -n 1 sudo lfs hsm_restore &
```

4. Create a directory and link input files

```
mkdir -p /fsx/data/CMAQ_Modeling_Platform_2018/
cd /fsx/data/CMAQ_Modeling_Platform_2018/
ln -s /fsx/CMAQv5.4_2018_12US1_Benchmark_2Day_Input/2018_12US1/ .
```

# Run CMAQ on ParallelCluster



1. Change directory to scripts location

```
cd /shared/build/openmpi_gcc/CMAQ_v54+/CCTM/scripts
```

2. Edit script to run for 1 day

```
vi run_cctm_2018_12US1_v54_cb6r5_ae6.20171222.3x96.ncclassic.csh
```

3. Submit job to Slurm queue

```
sinfo
sbatch run_cctm_2018_12US1_v54_cb6r5_ae6.20171222.3x96.ncclassic.csh
```

4. View status of job using squeue

```
squeue
```

Output:

```
JOBID PARTITION NAME      USER   ST      TIME      NODES NODELIST(REASON)
41     queue1    CMAQ      ubuntu CF       0:01      3 queue1-dy-compute-resource-1-[1-3]
```

Note that 3 nodes were provisioned, each with 96 pes per node. After configuring, the job will begin running, and the status changes from CF to R

5. Login to the compute node and run htop to view performance

```
ssh -Y queue1-dy-compute-resource-1-1
htop
```

6. After run is complete exit cluster and return to your local machine

```
exit
```

6. Delete Cluster

```
pcluster delete-cluster --region=us-east-1 --cluster-name cmaq
```

# Cost of 12US1 Benchmark

Cost for a two-day Benchmark of 12US1 in us-east-1 region

On 96 cores of c6a.48xlarge (1 node)
$7.344/hr *6313.40 sec/2day benchmark/3600 sec * 1 node
$7.344/hr * 1.75 hr * 1 = $13 (benchmark completes in ~ 2 hrs)

On 192 cores of c6a.48xlarge (2 nodes)
$7.344/hr * 3888.50sec/2day benchmark/3600 sec * 2 nodes =
$7.344/hr * 1.08 hr * 2= $16
(benchmark completed in ~ 1 hr)

Cost of the hpc6a.48xlarge node in us-east-2 region is only $2.88/hr and has the same performance at 60% cost savings.
*Requires access to us-east-2 region

*Using on-demand pricing for comparison because hpc6a is only available as on-demand

If you need higher core density, and more memory per node, latest release of AMD instances:

hpc7a.96xlarge (1 node) 192 cores, 768 GiB Memory $7.33 /hr

# CMAQv5.4 on AWS: The Big Picture

UNC INSTITUTE FOR THE ENVIRONMENT

**Summary:**

CMAQv5.4 workloads scale on AWS High Performance Computers (HPC). AWS enables sharing of software (public snapshots and AMIs) and Data (public S3 Buckets) to allow groups to quickly start 12US1 annual workloads from pre-installed, pre-set HPC configurations.

Single Virtual Machine (VM)

- 2-day CMAQv5.4 12US1 Domain on 96 cores completes in 2 hours for $13 using c6a.48xlarge.

Parallel Cluster (many VMs)

- 2-day CMAQv5.4 12US1 Domain on 192 cores (2 nodes) completes in 1 hour for $16 using c6a.48xlarge.

ParallelCluster gives users the flexibility to run CMAQ on many VMs and to design slurm queues using different EC2 sizes (c6a.4xlarge, c6a.48xlarge) to meet the demands of their entire workflow (pre- and post-processing).

Increasing the number of cores used to run CMAQv5.4 will generally result in faster time to completion, but the cost will go up as CMAQ does not scale linearly. Use 1 or 2 day scaling benchmarks to find balance between model run time and cost.

# Help Accelerate the use of Cloud Computing!

Join our Cloud Computing Workgroup:

https://github.com/CMASCenter/modeling-in-the-cloud/wiki/Modeling-in-the-Cloud-Workgroup-Charter

Answer our poll to help us understand your needs:

https://forms.gle/gbWqHWAeNoFUeYuK6

Please try the CMAQ on AWS tutorial!

https://pcluster-cmaq.readthedocs.io/en/latest/index.html#

Need help? Create new issue on the CMAS Center Forum.

https://forum.cmascenter.org/c/cloud-computing/95

# CMAS Cloud Computing Announcements

## CMAQ on AWS Workshop - Last day of the CMAS Conference (8/18/2023)

Register here:

Registration for the conference and this event preferred by Sunday, October 1st. Laptop requ

In-person only. Using ParallelCluster UI (web

Workshop limited to 80 attendees.

## Zoom Help Sessions - Provided the first Friday of each month from 2pm-3pm ET, hosted by Liz Adams of the CMAS Center

Register here:

Every month on the first Friday until December 1, 2023. 2pm - 3pm EST

- September 8, 2023 02:00 PM (adjusted date due to labor day)
- October 6, 2023 02:00 PM
- November 3, 2023 02:00 PM
- December 1, 2023 02:00 PM

# Acknowledgments


- The U.S. EPA, through its Office of Research and Development, partially funded and collaborated in the research described here under Contract EP-W-16-014 to the Institute for the Environment at the University of North Carolina at Chapel Hill.
- We thank the following AWS team for their help with running and optimizing in their systems and for the AWS Open Data Warehouse:
  - Timothy Brown
  - Tommy Johnston
  - Chris Stoner
- We thank the following staff at the EPA for several inspiring and helpful discussions and contributions to this work:
  - Chris Misenis
  - Ed Anderson
  - Tom Pierce - retired
- We thank the leaders of the Modeling in the Cloud Computing Workgroup:
  - Zac Adelman (LADCO)
  - Steve Fine (EPA) - retired
  - Fahim Sidi (EPA)
- We thank the members of the UNC Institute of the Environment for their contributions:
  - Manish Soni for creating the domain plots showing the extent of the CMAQv5.4 Benchmark cases using ArcGIS
  - Erin Valentine for helping with the webinar development