CLOUD COMPUTING EFFORTS AT THE CMAS CENTER

SARAVANAN ARUNACHALAM

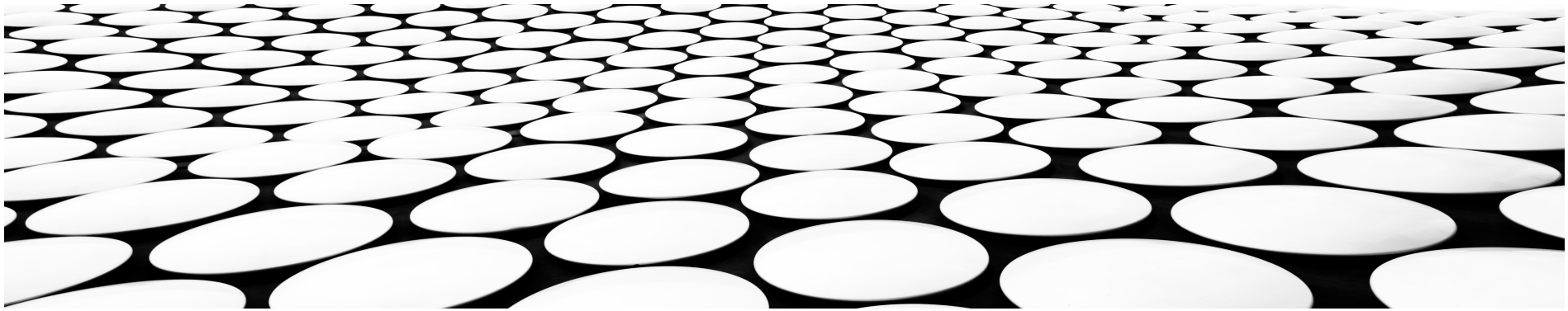
PROJECT TEAM:

ELIZABETH ADAMS, CARLIE COATS, CHRISTOS EFSTATHIOU

UNC INSTITUTE FOR THE ENVIRONMENT (UNC-IE)

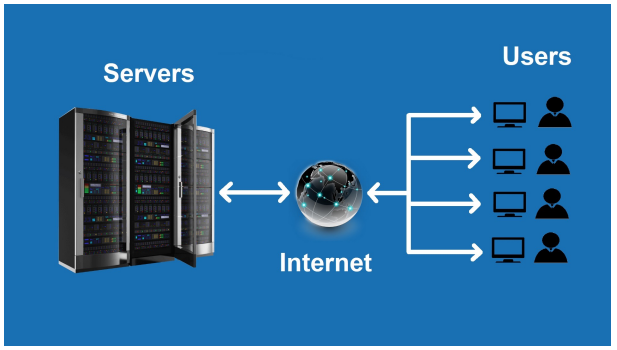
ROBERT ZELT, MARK REED

UNC INFORMATION TECHNOLOGY SERVICES (UNC-ITS)



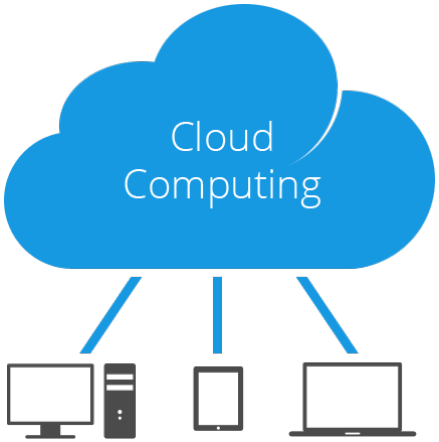
February 17, 2022

CURRENT



Move from Infrastructure on user-end to the Cloud Infrastructure as a Service (IaaS)

PROPOSED



Popular IaaS Providers

1) Amazon Web Services (AWS)
2) Microsoft Azure
3) Google Compute Engine (GCE)

CMAQ ON THE "CLOUD"

Motivation: Remove burden on user-end on two aspects

a) Maintaining hardware resources
b) Streamline process of building and running models in a "standard" computing environment

Goal: Implement and test the Community Multiscale Air Quality (CMAQ) model for on-demand access to a large remote pool of computing and data resources offered through commercial "cloud" vendors

- Address computing, software, and data issues collectively and make recommendations
- Develop summary of best practices for the CMAS User Community

Computing: Start with 2 commercial cloud computing platforms

- Amazon Web Services (AWS)
- Microsoft Azure

Software: Use of Virtual Machines, container or csh scripted native build approach to replicate existing system and software environment, so researchers avoid configuring from scratch (time & difficulty)

Data: Large volumes of data can be quickly shared and processed in the cloud, saving time from downloading locally and keeping redundant copies

DEFINING THE STEPS

Approach:

- Establish a set of benchmark cases representative of the needs of the CMAQ user community
- Review and enhance existing methods to connect to share common input datasets, store and distribute output
- Keep in mind other factors such as visualization tools, debugging, etc.

With an aim to:

- Develop methods to build, install, and run CMAQ as a Singularity container, Docker container or scripted native build
- Test performance and scalability on single node cloud computing environments
- Test multi-node high performance computing on the cloud using ***Amazon ParallelCluster*** and ***Microsoft Cyclecloud***
  - Ways to provision and manage HPC workloads using any scheduler to run MPI jobs
- Develop recommendations on provisioning resources, accurately forecasting CMAQ model run time with optimal configuration, storage requirements, and ultimately create reliable cost estimates for performing CMAQ simulations on the cloud for operational work

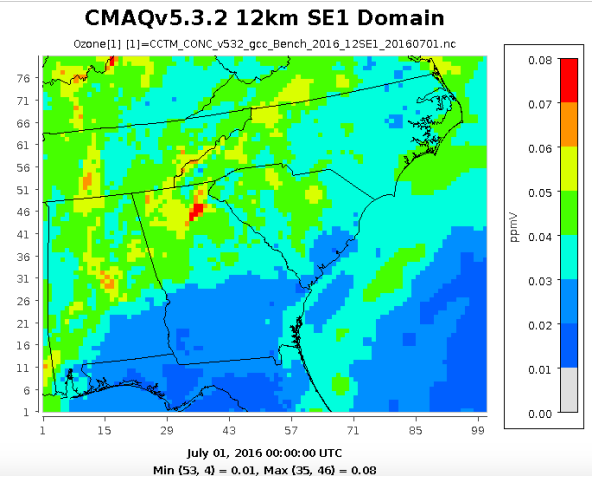
CMAS Center has been offering hands-on SMOKE and CMAQ training on AWS to a global audience for past several years

THE CMAQ BENCHMARK SUITE

Hardware configurations depend on the domain size, grid resolution, the number of variables and layers saved to the output

Typical requirements for two different 2-Day Benchmark Cases, both using a 12x12-km horizontal grid resolution

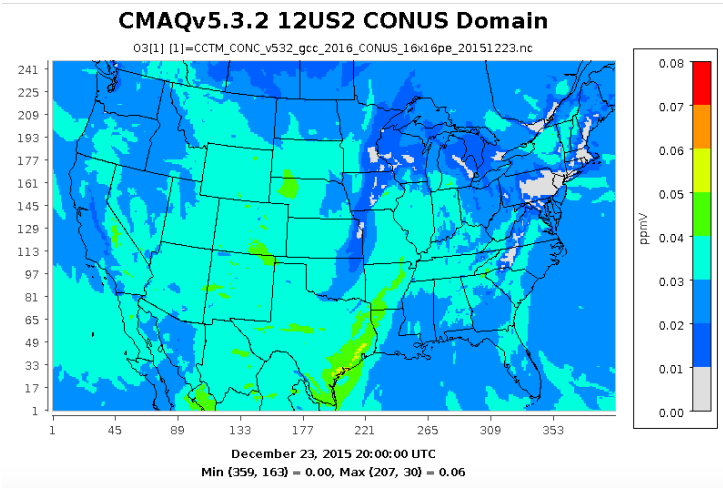
- Domain 1: Distributed 12SE1 Benchmark Case (NCOLS= 100, NROWS=80, NLAYS=35)
- Domain 2: CONUS (NCOLS= 396, NROWS=246, NLAYS=35)



Storage Requirements

Input:
23GB

Output:
15GB (full)
1.2GB (12 vars, 1 layer)



Storage Requirements

Input:
44GB

Output:
172GB (full)
17.7 GB (12vars, 1 layer)

PRELIMINARY FINDINGS AS OF FALL '21

- Benefits to working on the cloud
  - Quickly set-up and run - no waiting in job queue
  - Compute resource and storage is reliable, resizable, and quickly configurable
  - Flexible pricing – on-demand versus spot-pricing
  - AWS Parallel Cluster and Azure CycleCloud provides access to hundreds or thousands of compute cores, a job scheduler for keeping them fed, a shared file system that's tuned for throughput or IOPS (or both), updated compilers and libraries, and a fast network.
  - AWS Parallel Cluster "*infrastructure as code*" - single shell command can create a complex HPC cluster, ***and*** a Lustre file system, ***and*** a visualization studio, Azure Cycle Cloud provides some of this functionality but requires additional steps and assistance from UNC IT Support to provision.
  - AWS Share input and output data on s3 buckets with very fast download speeds using S3 API
  - AWS Parallel Cluster and Azure CycleCloud were difficult to configure run MPI across nodes within a container, but both supported building the libraries and compiling CMAQv5.3.2 natively and running using OpenMPI and the SLURM Job Scheduler.
  - Cost comparisons between similarly configured AWS Parallel Cluster and Azure Cycle Cloud couldn't be made directly, due to differences in how the clusters were provisioned and tested.

ONGOING WORK

- Phase 1: Create Minimum Viable Product (MVP) for CMAS Community
  - Ability to build (native build and not in container) and install CCTM (from CMAQ v5.3.3) on the cloud for the ConUS case (2016 12km case) on 2 environments - AWS (pCluster) and Azure (Cycle Cloud) for 2 days
  - Post-processing (combine, etc.) to reduce the data volume for potential future egress out of AWS/Azure
  - Ability to access input datasets from EPA's S3-bucket using EPA's 2016 modeling platform
  - Include robust testing (using varying # of cores and hardware configs as may be available from each vendor)
  - Documentation of entire procedure (install, run and data retrieval), with step-by-step tutorial for the user community to emulate for their own applications
  - Timings and costing info from
    - AWS: pCluster using c5n.18xlarge
    - Azure: Cycle Cloud using the HBv3 series
- Schedule for Phase 1
  - February 28, 2022: CMAQ Tutorial on AWS for EPA Review
  - March 31, 2022: CMAQ Tutorial on AWS for Cloud Computing Workgroup
  - April 30, 2022: CMAQ Tutorial on Azure for EPA Review
  - May 31, 2022: CMAQ Tutorial on Azure for Cloud Computing Workgroup
  - June 30, 2022: Comparisons of CMAQ Performance on AWS vs Azure

FUTURE WORK

Phase 2: Expanded capabilities of MVP (Specifics TBD)

- Move CMAQ to package management software (Yum or other)
- Change scripts from C-shell to Bash
- Enable running containers on both AWS pCluster and Cycle Cloud
- Access long-term simulation datasets from CMAS Data Warehouse (e.g., EQUATES datasets or other)
  - Currently on CMAS Center's Google Drive
  - Optionally move entire data to AWS Open Data Program, and provide access to compute nodes from here
- Add additional post-processing tools (e.g., AMET, VERDI or other python tools)
- Software as a "Service" (SaaS)?
- Other?

Schedule for Phase 2 TBD

REFERENCES

- Singularity Container Build Method and Documentation by Dr. Carlie Coats
  - https://cjcoats.github.io/CMAQ-singularity/
- Singularity Container Build Method by Ed Anderson
- Amazon AWS Parallel Cluster
  - https://aws.amazon.com/hpc/parallelcluster/
  - https://aws.amazon.com/blogs/storage/building-an-hpc-cluster-with-aws-parallelcluster-and-amazon-fsx-for-lustre/
- Azure Cycle Cloud
  - https://azure.microsoft.com/en-us/features/azure-cyclecloud/#overview
  - https://docs.microsoft.com/en-us/azure/cyclecloud/how-to/hb-hc-best-practices?view=cyclecloud-8#centos-76-hpc-marketplace-image

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.
- John McGee, UNC-ITS and Microsoft Azure for Cloud Credits
- We thank the following staff at the EPA for several inspiring and helpful discussions and contributions to this work
  - Kristen Foley
  - Fahim Sidi
  - Steve Fine
  - Ed Anderson
  - Tom Pierce