Clouds, Containers & CESM

Brian Dobbins
NCAR CISL/TDD/ASAP
June 19th, 2019
Overview

• What are we *actually* trying to understand?
• What does the cloud offer CESM?
• What do containers offer CESM?
• Making a containerized CESM RTE
• Additional benefits
• Conclusions
What are we *actually* trying to understand?

- People always ask two questions:
  - What *is* the cloud?
  - What *are* containers?

- These aren’t the right questions to ask! Too broad!
  - “What’s the answer to life, the universe and everything?”

- What we want to know is:
  - How can these technologies help scientists *run CESM*?
Successful Projects

• We’ve *already* used the AWS cloud for projects:
  • AGU & AMS hands-on tutorials
  • University-based science runs on AWS (August 2019)

• And recently containers too:
  • SCAM container – CESM & Jupyter Lab fused together
  • Same environment runs on Mac/Windows/Linux!

• *These experiences paved the way for the next steps.*
What’s needed to run CESM?

• At the highest level, we need resources:
  • Compute
  • Storage

• We also need some software:
  • Compilers, libraries, queuing system, etc.

• And we (very often) need expertise:
  • Installing CESM is non-trivial.
  • (Training is also important, but isn’t covered here!)
Overview

- What are we *actually* trying to understand?
- **What does the cloud offer CESM?**
- What do containers offer CESM?
- Making a containerized CESM RTE
- Additional benefits
- Conclusions
What does the Cloud offer CESM?

- At the highest level?
  - Compute & Storage
  - We can also pre-configure the software, but with notable limitations.

- This sounds great! What’s the catch?!
  - Performance at scale
  - Cloud complexity
  - Price
  - Portability
Cloud Limitations - Price

- On-demand pricing is expensive!
  - C5N 18xLarge in US-East-1 = $3.88/hr ($34K/yr)

- Spot pricing helps:
  - C5N 18xLarge in US-East-1 = $1.1659/hr ($10.2K/yr)
  - (But ‘spot’ pricing fluctuates, and jobs can be killed!)

- Price also depends on time-to-solution:
  - Intel Fortran can be ~2x faster than gfortran
  (But we can’t distribute the compiler!)
Cloud Limitations - Portability

• No compatibility between different providers:
  • Need to maintain master image *per provider*
  • Images aren’t compatible with local clusters either

• Each *region* needs its own copy of an image
  • AWS alone has 16 regions!

• Big maintenance issue for us!
Overview

• What are we *actually* trying to understand?
• What does the cloud offer CESM?
• **What do containers offer CESM?**
• Making a containerized CESM RTE
• Additional benefits
• Conclusions
What do containers offer CESM?

• An application environment that is:
  • Portable
  • Pre-configured

• Well, this *really* sounds great! What’s the catch?
  • MPI performance & portability
  • Development vs. Run-time Environments
Containers - MPI Performance & Portability

- MPICH ABI Compatibility Initiative
  - Software guarantees *run-time* library compatibility
    (Includes Intel MPI, Cray MPI, MVAPICH, etc.)

- Example: Build CESM in container w/ MPICH, then:
  - Run with vendor-tuned MPI & get native performance!
  - Works, but still not automatic (except: Shifter on Cray)
Containers – Development Environments

- CESM 2.1 requires a development environment:
  - Need to compile POP, CICE for specific PE counts
  - Source mods

- Problems:
  - Dev environments increase container size
  - License issues for the Intel compiler
  - *Need to modify & save container per case* (Not allowed at some centers.)
Containers – Runtime Environments (RTE)

• Everything needed to run is provided:
  • No compilation
  • No issues with compiler licenses
  • Smaller size
  • Easier to use
Overview

• What are we *actually* trying to understand?
• What does the cloud offer CESM?
• What do containers offer CESM?
• **Making a containerized CESM RTE**
• Additional benefits
• Conclusions
CESM RTE – POP & CICE

- Compile-time dependencies in POP:
  - POP -> MOM, which doesn’t have these dependencies
  - Alternatively, modify POP?

- Compile-time dependencies in CICE:
  - Modify the code – what’s the performance hit?
What if we treat all components as ‘plug-ins’?
  - Build *shared libraries*, not static ones
  - Coupler *dynamically loads* components at start-up

Development environment ensures compatibility
  - Intel & GCC are compatible
  - Plugins can be *shared*
CESM RTE – What would it look like?

• cesm.exe:
  • Runs on any processor counts
  • Dynamic load of precompiled components

• Out of the box, a single executable runs *all* cases:
  • We build CESM just once
  • Intel-built objects can be shared - better performance!
Overview

- What are we *actually* trying to understand?
- What does the cloud offer CESM?
- What do containers offer CESM?
- Making a containerized CESM RTE
- **Additional benefits**
- Conclusions
Additional Benefits

- Reproducibility:
  - Single compiler, flags and math libraries
  - (Caveats: MPI collective orderings & CPU optimizations)

- Debugging:
  - Fewer compiler / library / OS issues to solve
  - The exact environment can be reproduced anywhere

(And with additional work, *run-time* load balancing.)
Overview

- What are we *actually* trying to understand?
- What does the cloud offer CESM?
- What do containers offer CESM?
- Making a containerized CESM RTE
- Additional benefits
- **Conclusions**
Conclusions

- The cloud lets us reach more scientists with CESM:
  - Hardware & storage on-demand
  - Expertise via a preconfigured environment

- Containers offer a portable, preconfigured CESM
  - But development environments are a poor fit

- A CESM RTE container gives us:
  - Portable, performant CESM
  - Can run on multiple clouds and university systems
  - Easier maintenance / user support
Questions?

Brian Dobbins
bdobbins@ucar.edu

(Also, in talks with CU on a Cloud/Container meet-up for research computing.)
MPICH ABI Compatibility on Cheyenne

Relative times for OSU 'Reduce' MPI Benchmark
(16 nodes, 1 rank/node)

- Blue: MPICH Container vs MPICH Native
- Red: MPT Native vs MPICH Native

Percent Change

Message Size (bytes)