

# Reduced precision microphysics parameterizations in CAM

*John Dennis,*  
*NCAR CISL/TDD/ASAP, Scientist*

June 19, 2019



**Do all calculations in CAM/CESM need to be performed in 64-bit floating point?**



# History/Motivation

- Historically:
  - Climate models: 64-bit floating-point
    - On Cray vector systems -> single precision is 64-bit floating-point
    - Small per MPI rank problem size means less cache pressure
    - 4-byte calculations cost same as 8-bytes
    - Needed for conservation
    - Simpler
  - Weather models: 32-bit floating-point
    - Large per MPI rank problem size
    - Lower precision floating reduces cache pressure
- Now:
  - Vector instruction sets means that 32-bit floating-point is potentially 2x that of 64-bit floating-point

# **What are the implications of the use of 32-bit in CAM/CESM?**



# Potential implications of 32-bit floating-point

- Is correctness maintained?
- Does it reduce code execution time?
- Does it negatively impact maintainability?



# Approach

- Previous results:
  - WACCM implicit solver [kernel] -> 1.97x speedup
  - Solver was trivial to vectorize
  - Virtually no 'if' tests in computational kernel
- Want something more challenging!
  - Morrison Gettelman Microphysics version 2
  - Relatively expensive: ~5% of total CAM cost
  - Complex code with lots of 'if' tests
  - Extensive experience optimizing code base
  - Willing collaborator (KEY)

# Optimization approach: vectorize everything

```
real, intent(in) :: t ! Temperature in Kelvin  
real, intent(out) :: es ! SVP in Pa
```

```
! uncertain below -70 C  
es = 10.**(-7.90298*(tboil/t-1.)+ &  
5.02808*log10(tboil/t)- &  
1.3816e-7*(10.**((11.344*(1.-t/tboil))-1.)+ &  
8.1328e-3*(10.**(-3.49149*(tboil/t-1.))-1.)+ &  
log10(1013.246))*100.
```

1 single-precision result  
1 double-precision result

```
integer, intent(in) :: vlen  
real, intent(in) :: t(vlen) ! Temperature in Kelvin  
real, intent(out) :: es(vlen) ! SVP in Pa  
integer :: i  
! uncertain below -70 C  
do i=1,vlen  
  es(i) = 10.**(-7.90298*(tboil/t(i)-1.)+ &  
5.02808*log10(tboil/t(i))- &  
1.3816e-7*(10.**((11.344*(1.-t(i)/tboil))-1.)+ &  
8.1328e-3*(10.**(-3.49149*(tboil/t(i)-1.))-1.)+ &  
log10(1013.246))*100.  
enddo
```

8 single-precision results  
4 double-precision results

# Is correctness maintained?

- Did not pass CESM verification test
  - The changes are statistically distinguishable from natural variability
- Does appear to pass initial evaluation by Andrew.



# Does it reduce execution time?

- MG2 calculation only
  - Cheyenne
    - Kernel: R4  $\rightarrow$  1.35x speedup versus R8
    - In CAM: R4  $\rightarrow$  1.22x speedup versus R8
- Current R8 $\rightarrow$ R4 speedup is equivalent of Broadwell to Skylake speedup.
- Variation across different MPI ranks:
  - 2x speedup on a few execution paths
  - Additional execution paths could be optimized?
- Overall impact on CAM:  $\sim 0.5\%$ 
  - Very large overhead in actually calling MG2 from CAM
  - Other parameterizations in CAM are significantly more expensive (CLUBB)

# Does it negatively impact maintainability?

- Single point to switch from 8-byte to 4-byte calculations 😊
- Multiple entry points into modified code
  - Certain MG2 utility routines are called outside main subroutine
  - Saturation vapor pressure calculations called from multiple locations in CAM
  - Need to include both 4-byte, 8-byte, vector and scalar versions of numerous subroutines 😞
- Constants: Maintain separate 8-byte and 4-byte versions or type conversion of 8-byte constant?

# Recommendation for new parameterizations

- Simplified support for reduced precision will be in next version of CESM
- Develop new parameterization that can be switch between single and double precision
  - Focus on 4-byte version
  - Scientific justification for 8-byte
- Think of calculation on groups of points not single model grid-point



# Conclusions/Future work

- Use of 32-bit floating-point does not break correctness
- Achieves speedup comparable to next generation of processor
- Does currently impact code maintainability due to call structure in CAM
- Should future parameterizations be 32-bit floating-point?

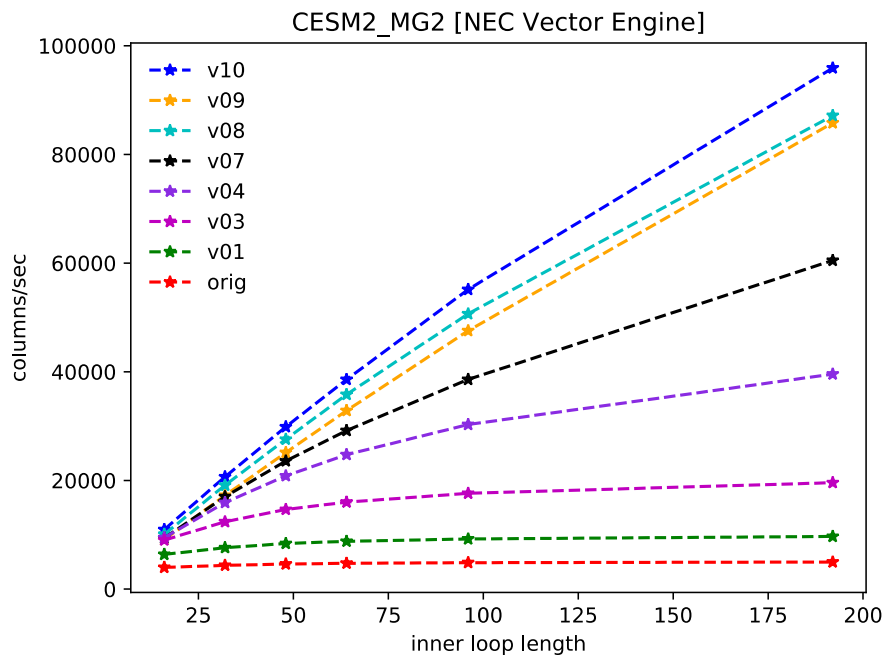


# Questions?

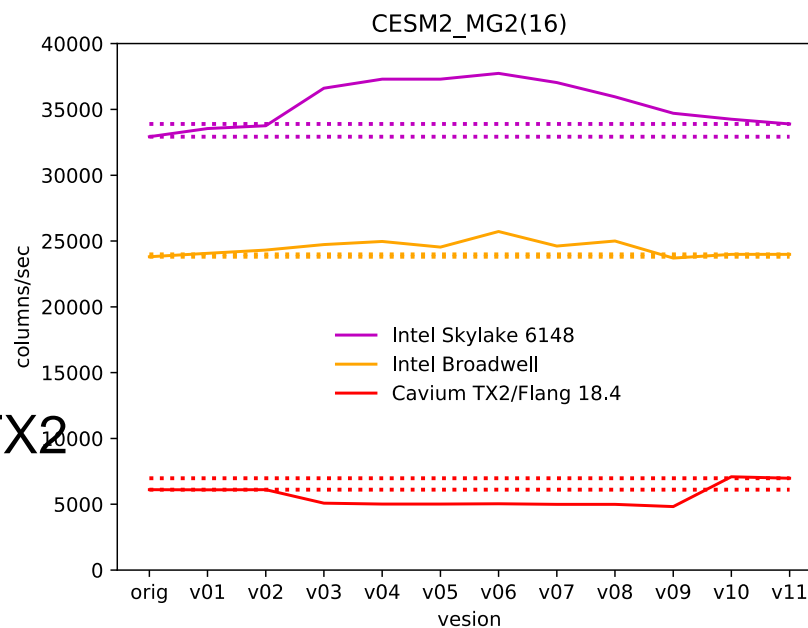
- Contact info: [dennis@ucar.edu](mailto:dennis@ucar.edu)
- Collaborators:
  - Dan Milroy
  - Andrew Gettelman
  - Allison Baker



# Extreme vectorization of the CESM2\_MG2 kernel



~20x speedup on NEC VE



Performance neutral for Xeon and TX2