Computational & Information Systems Laboratory

CISL

Update on many-core optimization of CESM

John Dennis, Chris Kerr, Youngsung Kim, Raghu Kumar, Sheri Mickelson

Application Scalability and Performance Group, CISL

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

NSF

Motivation

- CESM is expensive to run!
- Maximize amount of science by reducing cost of CESM
- Many-core architectures providing large number of on-node threads
  - KNL:
    - 64-72 cores/ 4-way threading
    - High-bandwidth on-package memory (5x DDR)

Group/Team

- Rich Loft, Division Director (NCAR)
- John Dennis, Scientist (NCAR)
- Chris Kerr, Software Engineer, contractor
- Youngsung Kim, Software Engineer (NCAR) /Graduate Student (CU)
- Raghu Raj Prasanna Kumar, Associate Scientist (NCAR)
- Sheri Mickelson, Software Engineer (NCAR) / Grarduate Student (CSU)
- Ravi Nanjundiah, Professor (IISc)

Related/Collaborative Activities

- Funding from Intel Parallel Computing Center (IPCC-WACS)
- NESAP (NERSC Exascale Science Application Program)
  - Bi-weekly: NERSC-Cray-NCAR telecon on CESM & HOMME performance (Feb 2015)
- Weekly Intel-TACC-NREL-NERSC-NCAR telecon
  - Concall focused on CESM/HOMME KNC performance
- Strategic Parallel Optimization of Computing
  - NCAR effort focused on Xeon architectures
  - SPOC currently focused on MPAS

Current optimization focus

- Xeon and Xeon Phi based platforms
  - Sandybridge (SNB) [i.e. Yellowstone]
  - Ivybridge (IVB) [i.e. Edison]
  - Haswell (HSW) [i.e. Cori Phase I]
  - Broadwell (BDW) [i.e. Cheyenne)
  - Knights Landing (KNL) [i.e. Cori Phase II]

Optimization Projects

- Completed in CAM code base (~14% reduction in CAM)
  - Morrison Gettelman micro-physics scheme
  - Random number generator
  - Planetary Boundary Layer Utility module
  - Bilinear interpolation module
  - Heterogenous freezing module from MAM4
- Underway
  - Implicit Chemistry solver (WACCM) [6-10% reduction in WACCM]
  - Eulerian Advection [5-10 % reduction in CAM-SE, 40% reduction in WACCM-SE]
  - CICE boundary Exchange
  - Dropmix
  - CLUBB

Optimization Projects

- Completed in CAM code base (~14% reduction in CAM)
  - Morrison Gettelman micro-physics scheme
  - Random number generatory
  - Planetary Boundary Layer Utility module
  - Bilinear interpolation module
  - Heterogenous freezing module from MAM4
- Underway
  - Implicit Chemistry solver (WACCM) [6-10% reduction in WACCM]
  - Eulerian Advection [5-10 % reduction in CAM-SE, 40% reduction in WACCM-SE]
  - CICE boundary Exchange
  - Dropmix
  - CLUBB

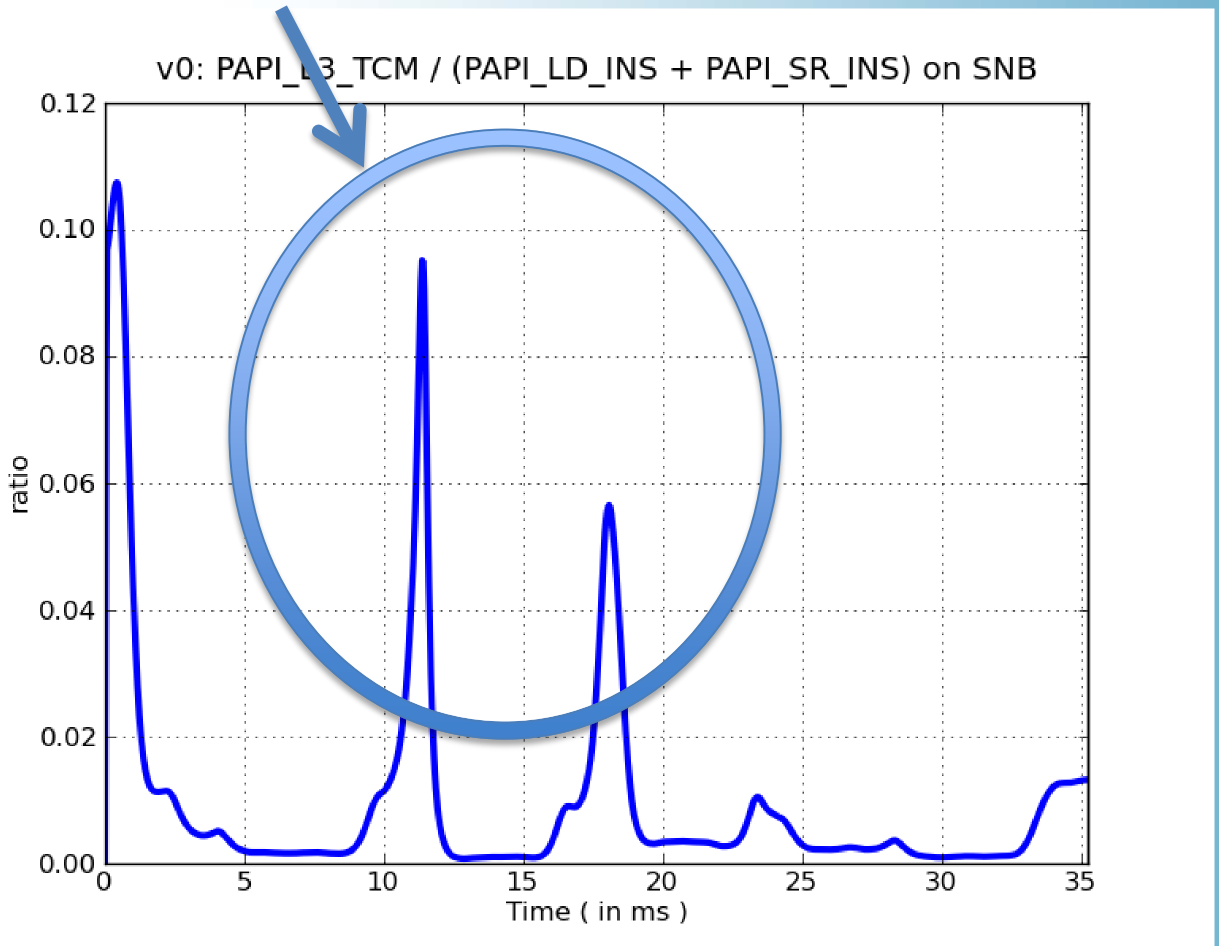
HOMME: Eulerian Advection

HOMME overview

- High Order Methods Modeling Environment (HOMME)
  - CAM: 35% of time (plev=30, qsize=25)
  - WACCM: 88% of time (plev=70, qsize=135)
- Default advection algorithm in HOMME
  - 2nd order Runge-Kutta
  - Memory hierarchy bandwidth limited
  - Expensive:
    - CAM: 23% of time
    - WACCM: 81% of time

HOMME advection (CAM-like): L3 miss rates

Why are we missing so badly in L3?



Computational & Information Systems Laboratory

CISL

NCAR

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

NSF

Optimization steps

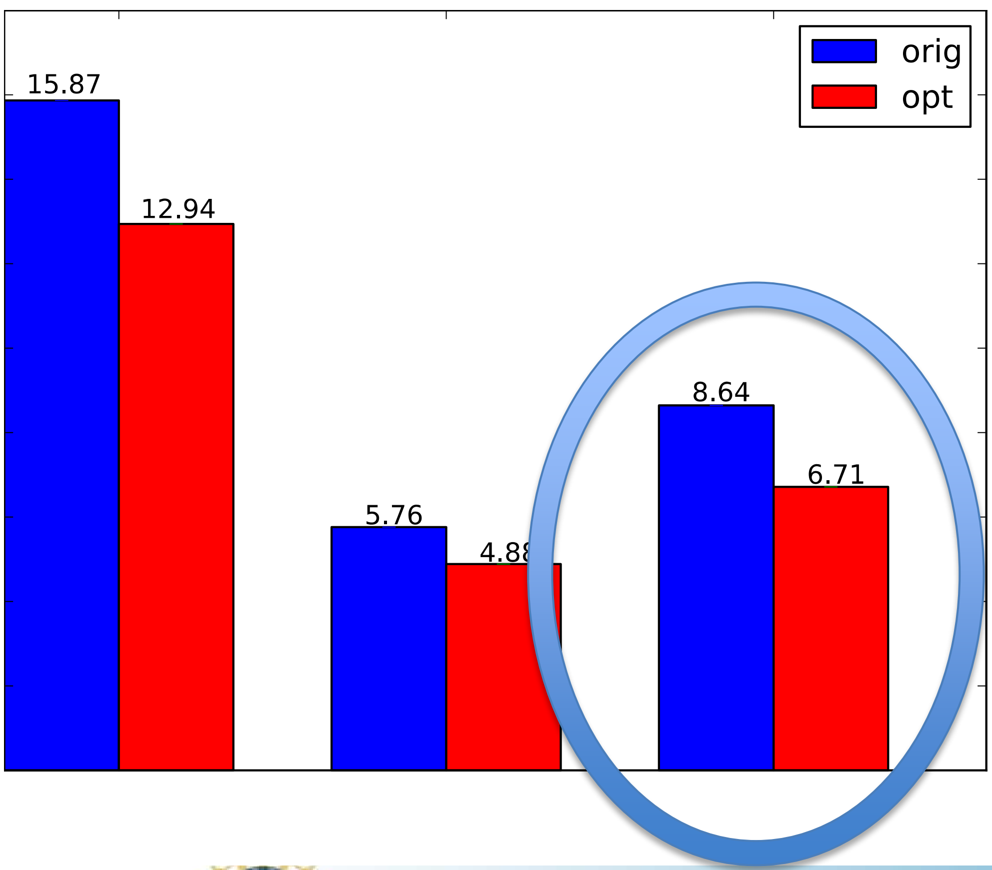
1. Threading memory copy in boundary exchange [Jamroz]
2. Restructure data-structures for vectorization [Vadlamani & Dennis]
3. Rewrite message passing library/ specialized comm ops [Dennis]
4. Rearrange calculations in euler_step for cache reuse [Dennis]
5. Reduced # of divides [Dennis]
6. Restructured/alignment for better vectorization [Kerr]
7. Rewrote and optimized limiter [Demeshko & Kerr]
8. Redesign of OpenMP threading [Kerr & Dennis]

CISL Computational & Information Systems Laboratory

NCAR NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

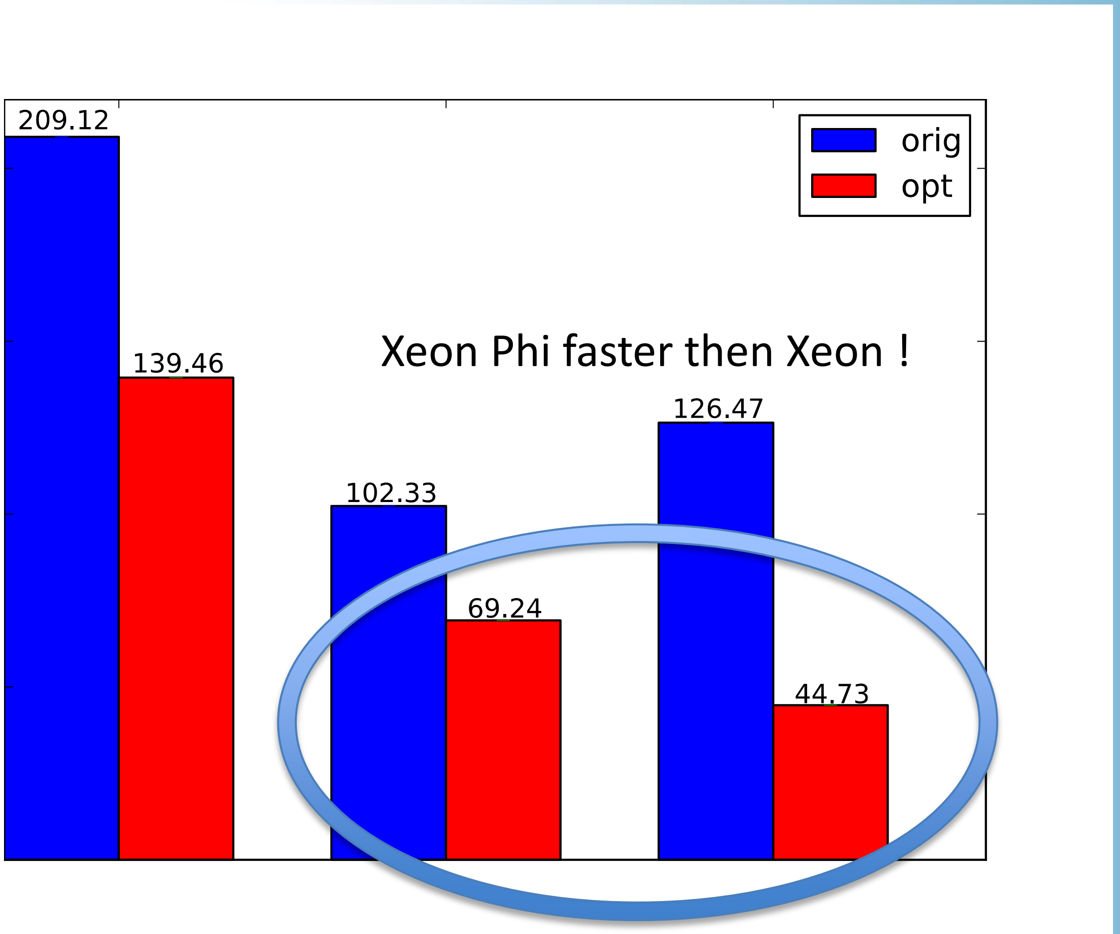
NSF

HOMME (NE=8, PLEV=26, qsize=25) [single node]



15-22% reduction in cost!

HOMME (NE=8, PLEV=70, qsize=135)[single node]



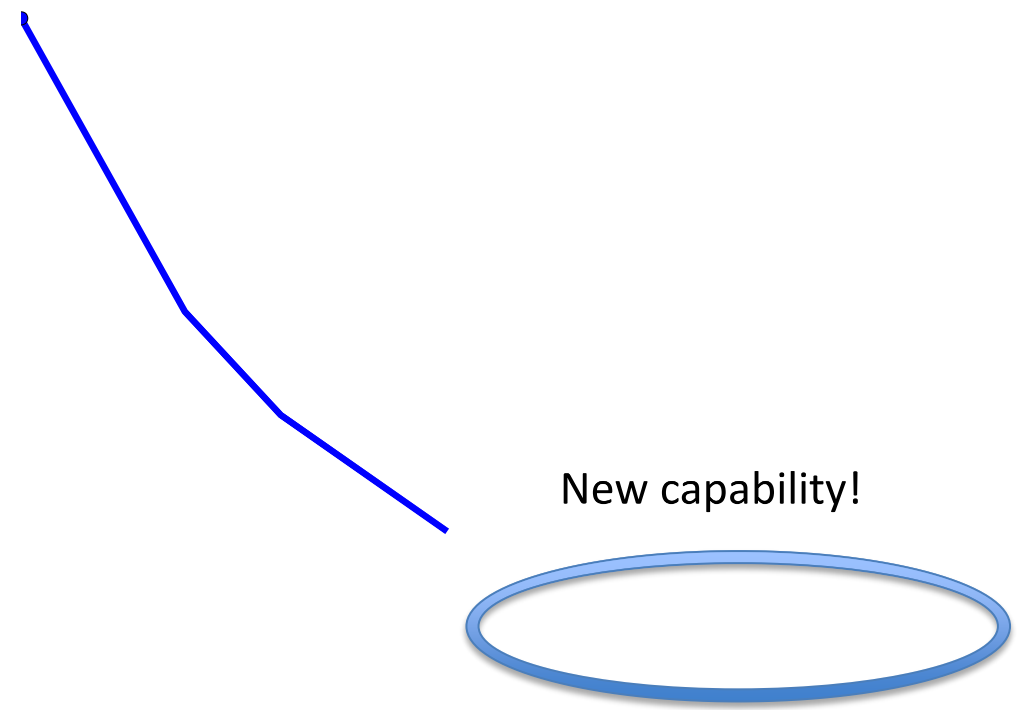
Computational & Information Systems Laboratory

CISL

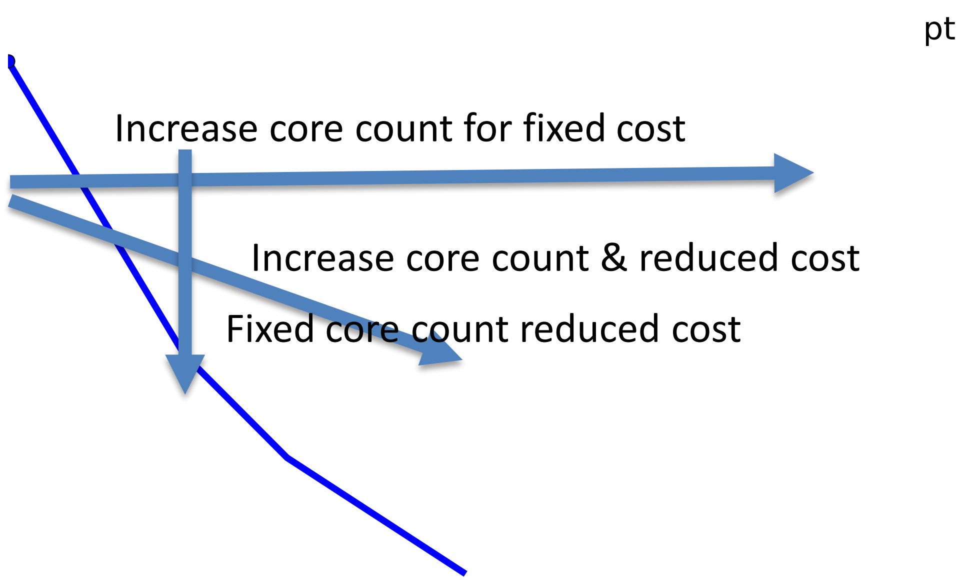
NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

32-64% reduction in cost!

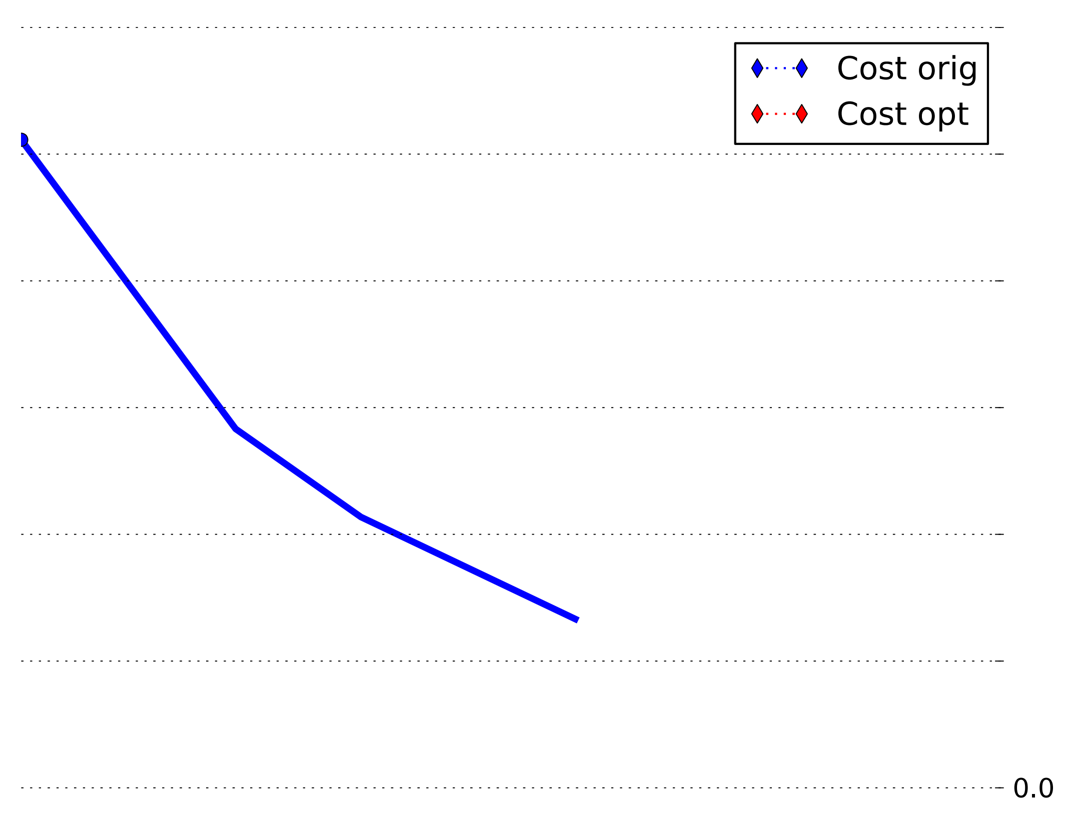
HOMME (NE=8, PLEV=70, qsize=135)



HOMME (NE=8, PLEV=70, qsize=135)



HOMME (NE=8, PLEV=26, qsize=25)



~30% reduction in cost!

Computational & Information Systems Laboratory

CISL

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

NSF

Recommendations for optimization

- Eliminated use of elemental functions
- Pushed vector loop into lower into call stack
- Eliminated divides where possible
- Remove initialization of variables that are overwritten
- Avoid redundant initial computations
- Eliminate use of automatic arrays
- Rearrange loops to allow alignment
- Use more aggressive compiler optimizations

Lessons from Optimization

- Need for multiple different performance analysis tools
  - General identification (GPTL)
  - Diagnose performance issue (extrae, vTune)
  - Optimization phase (gprof, vprof, system_clock)
- Optimizations involve many phases
  - Fixing one issue reveals another
  - Keep going until you are sick of it! ☺

Conclusions/Future work

- Concerted effort reduced cost of CAM by 14%
- A number of transformations are relatively easy
- Community code → Needs to be a community effort to improve performance
- New threading capability in HOMME allows greater parallelism.

Questions?

dennis@ucar.edu

Long-wave rad. in PORT and PSrad (sequential - one thread)

PORT*

- Configuration
  - NX=144, NY=96, NLEV=26, NCOL=16

- SUBROUTINE rad_rrtmg_lw()

```
call mcica_subcol_lw()
do iplon = 1, ncol
        call inatm()       \
        call cldprmc()      |
        call setcoef()      |  rrtmg_lw()
        call taumol()       |
        call rtrnmc()      /
end do
```

* : Parallel Offline Radiative Transfer

PSrad

- Configuration
  - LAT=96, LON=192, NLEV=47, KPROMA=16

- SUBROUTINE lrtm()

```
call sample_cld_state()
call lrtm_coeffs()
do ig = 1, n_gpts_ts
        do jl = 1, kproma
                call gas_optics_lw()
        end do
end do
do ig = 1, nbndlw
        planckFunction()
end do
do ig = 1, n_gpts_ts
        call lrtm_solver()
end do
```

What code look like before and after changes

After changes:

```
function wv_sat_svp_to_qsat(es, p, mgncol) result(qs)
  integer,                             intent(in) :: mgncol
  real(r8), dimension(mgncol), intent(in) :: es  ! SVP
  real(r8), dimension(mgncol), intent(in) :: p   !
Current pressure.
  real(r8), dimension(mgncol) :: qs
  integer :: i
  do i=1,mgncol
  ! If pressure is less than SVP, set qs to maximum of
1.
  if ( (p(i) - es(i)) <= 0._r8 ) then
     qs(i) = 1.0_r8
  else
     qs(i) = epsilo*es(i) / (p(i) - omeps*es(i))
  end if
  enddo
end function wv_sat_svp_to_qsat
```

What code look like before and after changes

Before changes:

```
elemental function wv_sat_svp_to_qsat(es, p)
result(qs)

  real(r8), intent(in) :: es   ! SVP
  real(r8), intent(in) :: p    ! Current pressure.
  real(r8) :: qs

  ! If pressure is less than SVP, set qs to maximum
of 1.
  if ( (p - es) <= 0._r8 ) then
     qs = 1.0_r8
  else
     qs = epsilo*es / (p - omeps*es)
  end if

end function wv_sat_svp_to_qsat
```