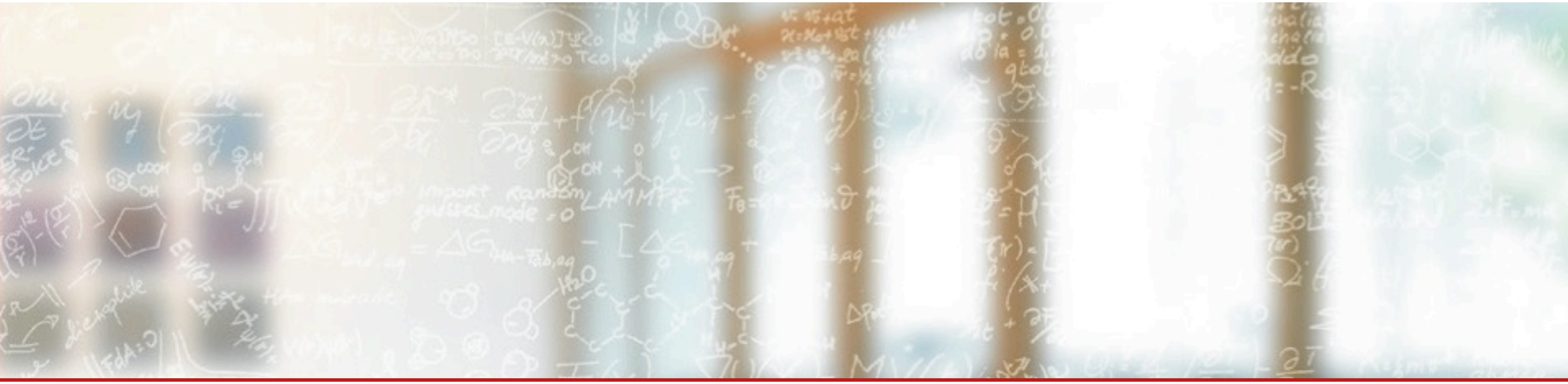
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Status of ICON climate model port on GPUs

Event: ENES Workshop, Toulouse, France

Author: Will Sawyer (CSCS), Markus Wetzstein (CSCS), **Carlos Osuna (MCH)**

Apr. 7, 2016

Status: ICON GPU port with OpenACC directives

- Dycore: Running, on icon-gpu development branch (next slide)
- Advection: Several horizontal/vertical schemes implemented
    - *upwind_vflux_up:* implemented, tested
    - *upwind_hflux_up:* implemented, tested
    - *upwind_vflux_ppm:* implemented, tested
    - *upwind_hflux_miura:* implemented, except not yet semi-Lagrangian option
- Currently: incremental merge into icon-dev
- Albeit: *uses non-standard features of Cray compiler (full deep copy). Cray's long-term support of OpenACC is questionable*

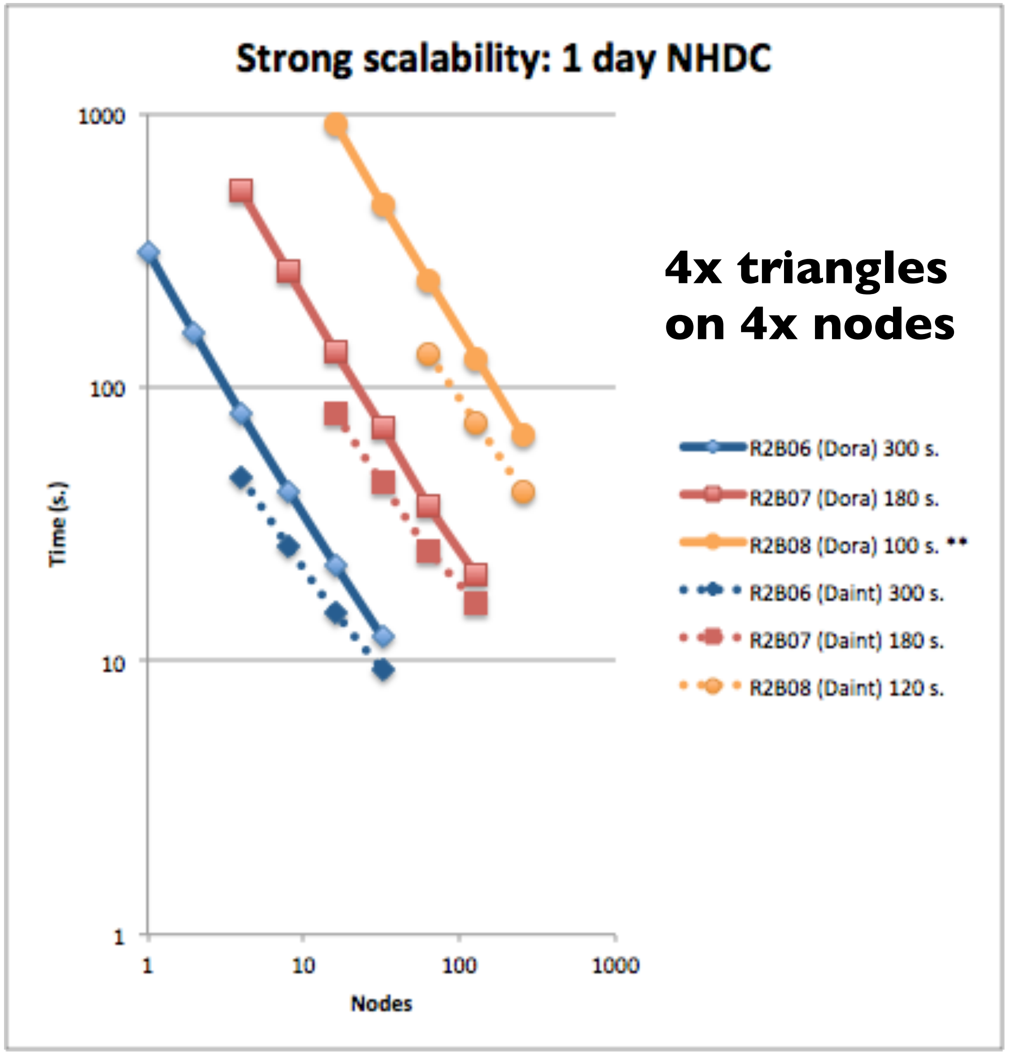
Issue: OpenACC Deep Copy

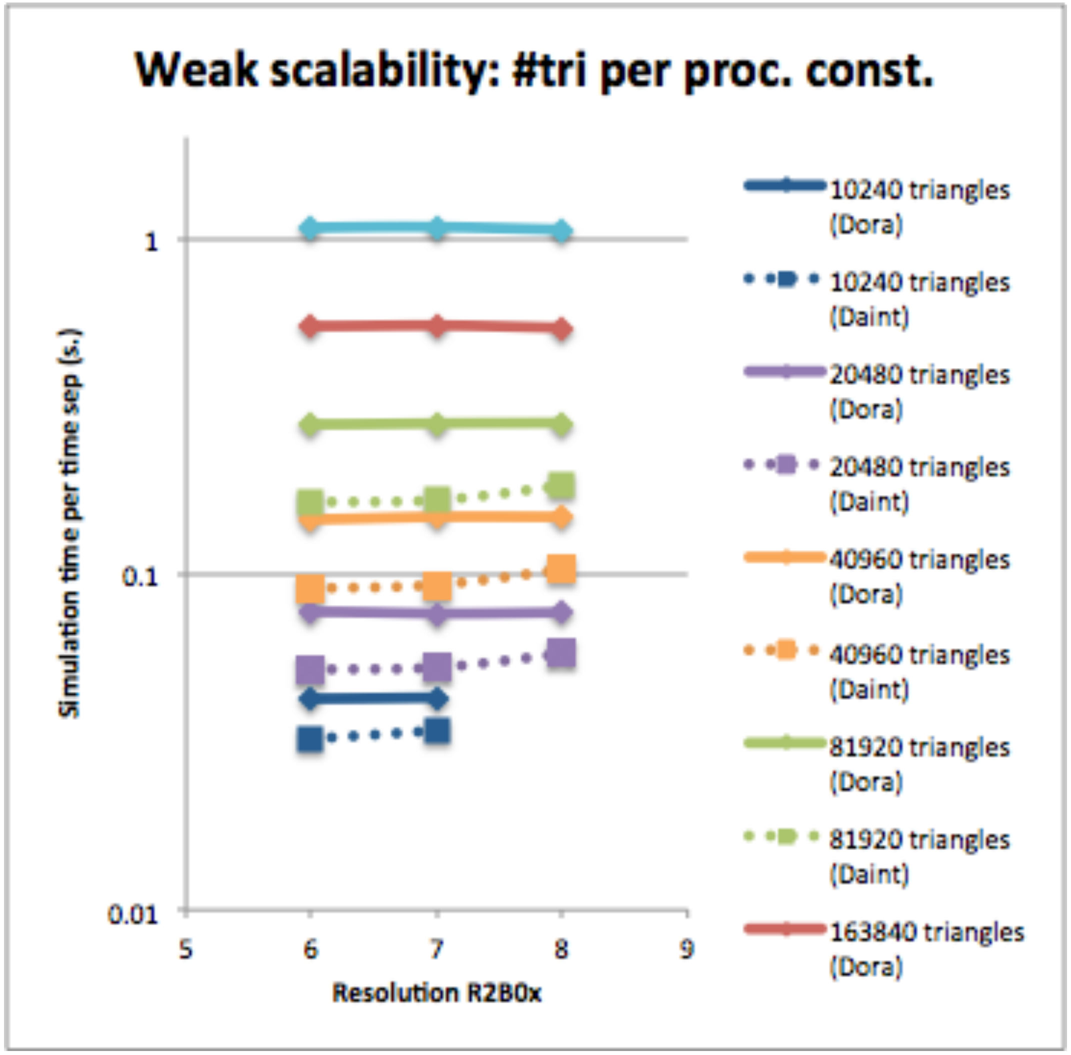
```
#if defined( _OPENACC )
  CALL init_gpu_variables( )
  CALL save_convenience_pointers( )
!$ACC DATA COPYIN( p_int_state, p_patch, p_nh_state, prep_adv ), IF ( i_am_accel_node )
  CALL refresh_convenience_pointers( )
#endif
  TIME_LOOP: DO jstep = (jstep0+jstep_shift+1), (jstep0+nsteps)
   :
  ENDDO TIME_LOOP
#if defined( _OPENACC )
  CALL save_convenience_pointers( )
!$ACC END DATA
  CALL refresh_convenience_pointers( )
  CALL finalize_gpu_variables( )
#endif
```

- Cray extension to standard
- Was expected to be included in 2014 (not accepted by committee)
- Cray now leaning to OpenMP-accelerated (but implementations immature)
- PGI/NVIDIA offers “unified memory” and now addressing deep copy for OpenACC 3.0 (16Q4)
- *ICON needs to compile/run with PGI*

ICON non-hydrostatic dynamical core scaling

Compare ICON Trunk Piz Dora (2x Haswell sockets) vs. Piz Daint nodes w/ K20x (both with Cray CCE)





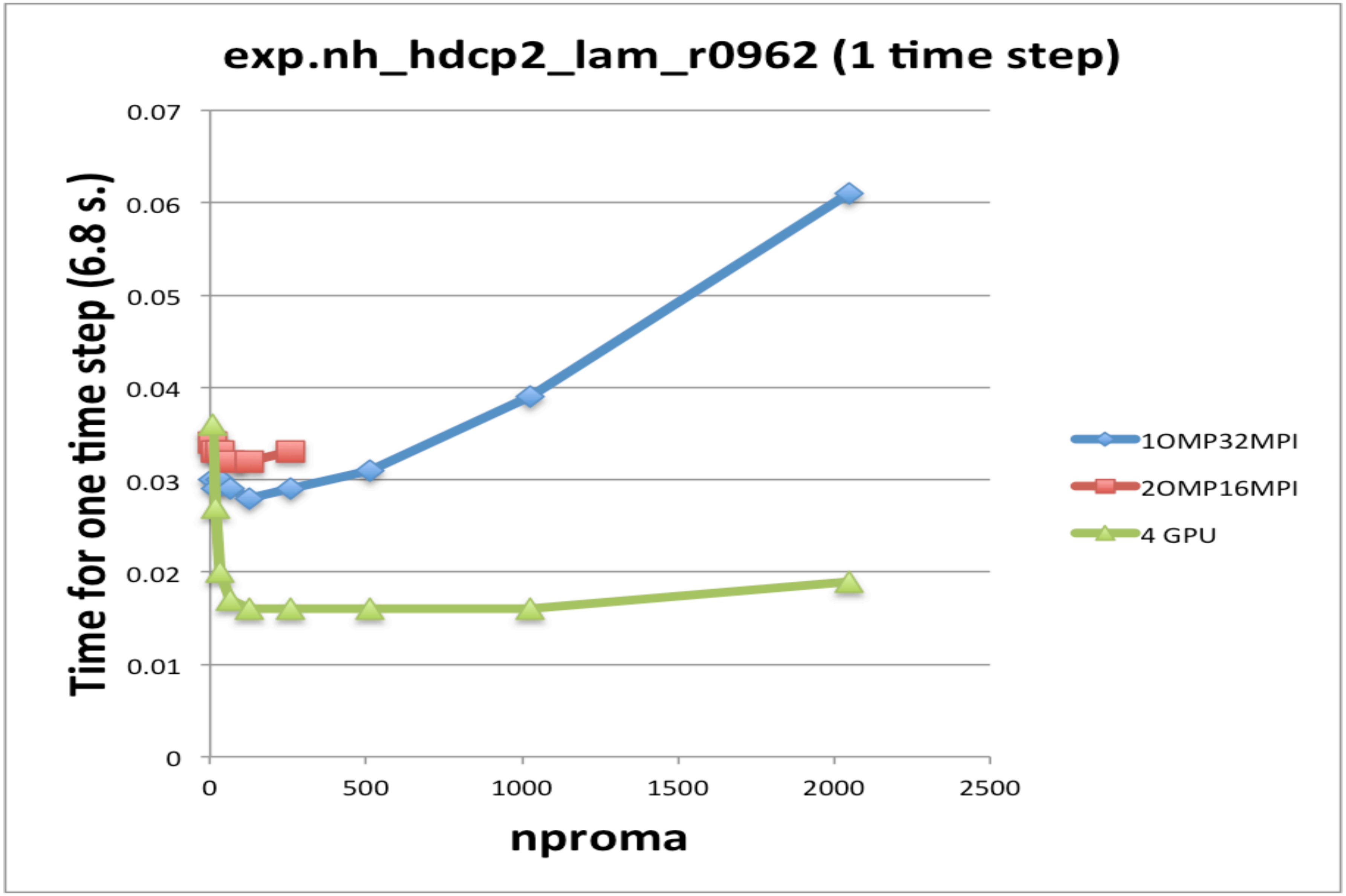
$$\frac{[\text{R2B08 Dora-64 (s)}]}{[\text{R2B08 Daint-64 (s)}]} = 1.45 \qquad \frac{[\text{BW Daint node (GB/s)}]}{[\text{BW Dora node (GB/s)}]} = \frac{165}{110} = 1.5$$

Status: ICON Physics

- *Radiation:* RRTMGP prototype (Robert Pincus, NOAA), more later
- *ECHAM saturation adjustment* (Xiaolin): OpenACC version running and tested
- *NWP saturation adjustment* (Will): OpenACC version running/tested
- *LES Turbulence* (Markus with Anurag, MPI-M): early design of interface and 1-2 routines ported, but *stopped work due to code restructuring*
- *Cloud microphysics* (Markus with Axel, DWD and Thomas, DKRZ): extensive refactoring (w/ development conflicts) was needed; version very close to hdcp2_refactor version (DKRZ) with some OpenACC additions. Nearing completion

Saturation Adjustment: Performance Sandybridge vs. K20x

NWP SatAdj (4 nodes)



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CSCS
Swiss National Supercomputing Centre

Cloud microphysics port: Hybrid OpenMP/OpenACC

Effective collaboration between DKRZ (Thomas), DWD (Axel) and CSCS (Markus)

```
!$OMP PARALLEL
!$OMP DO PRIVATE(jb,jc,i_startidx,i_endidx,zncn,qnc,qnc_s) ICON_OMP_GUIDED_SCHEDULE
      DO jb = i_startblk, i_endblk
        CALL get_indices_c(p_patch, jb, i_startblk, i_endblk, &
          &                 i_startidx, i_endidx, i_rlstart, i_rlend)
                     :
        CALL two_moment_mcrph(                             &
                       isize  = nproma,                     &!in: array size
                             :
                       qv     = p_prog_rcf%tracer (:,:,jb,iqv), &!inout:sp humidity
                       qc     = p_prog_rcf%tracer (:,:,jb,iqc), &!inout:cloud water
        )
```

Many layers down:

```
SUBROUTINE clouds_twomoment(ik_slice, dt, use_prog_in, atmo, &
       cloud, rain, ice, snow, graupel, hail, n_inact, n_cn, n_inpot)
!$ACC ROUTINE VECTOR
        :
!$ACC LOOP COLLAPSE(2) VECTOR IF(i_am_accel_node .AND. acc_on)
      DO k=kstart,kend
        DO i=istart,iend
          cloud%n(i,k) = MAX(cloud%n(i,k), cloud%q(i,k) / cloud%x_max)
          cloud%n(i,k) = MIN(cloud%n(i,k), cloud%q(i,k) / cloud%x_min)
        END DO
      END DO
```

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CSCS
Swiss National Supercomputing Centre

ICON-GPU effort: next steps

- *Dynamics/Advection:* check in OpenACC code for dycore/advection. *No implications for normal CPU operation*
- *Cloud Microphysics:*
  - Consolidate/merge DKRZ/DWD/CSCS work on branches/icon-gpu/icon-gpu-cscs-phys
  - Incorporate Markus's OpenACC implementation
- *Turbulence:* restart OpenACC effort (contact person at MPI-M?)
- *Saturation Adjustment:* optimize OpenACC implementations (NWP & ECHAM), check into icon-dev
- *Radiation LW:* collaboration with NOAA/CU/AER to implement OpenACC RRTMGP LW prototype (next slide)
- *Radiation SW:* Investigate solutions, e.g., RRTMGPU_SW

RRTMGP: new radiation code (AER/CU/NOAA, Robert Pincus)

- RRTMGP is an evolution of PSrad, itself based on RRTMG, aimed at increasing *efficiency* and *flexibility*
- *Flexibility* comes through Fortran 2003 classes: data and procedures that operate on them
- Current version computes clear sky fluxes, soon also all-sky
- Underlying spectroscopic data will not be available until Fall 2016, thus RRTMGP can't be used in production currently

Well-structured code for fast prototyping, e.g., OpenACC implementation:

- Nov. 2015: Andre Wehe, lead software engineer, @ CSCS with WS
- *Fortran 2003 classes in OpenACC not properly supported by CCE or PGI*
- Andre backed out Fortran 2003 code; manual inlining. *Worst-case scenario!*
- *Validated OpenACC version of gas_optics_core with CCE;* PGI not validated
- PGI promised help for compiler bugs, *https://github.com/eth-cscs/openacc*

RRTMGP rewrite: collaboration of MPI-M/CSCS/C2SM/DKRZ/DWD

- As RRTMGP is a new code which succinctly expresses LW radiation
- The LW solver is minimal, about 200 lines of code
- LW solver is a good candidate for a full rewrite (like COSMO dycore) for performance-portability (CPU / GPU / Xeon Phi)

Currently analyzing possible approaches:

- CUDA/C++, OpenCL/C, continued OpenACC
- Domain specific languages (possibly embedded in C++)
  - GridTools (yesterday's talk)
  - CLAW (Jon Rood's subsequent talk)
  - ICONFOR (MPI-M, Leonidas Linardakis)

Next step: decide on approach(es), then pass off to programming team

Serialization of results for model validation

pp_ser.py source-to-source translator expands !$ser directives to serialize check-point output to compare with reference baseline: crucial for validation of OpenACC code. Also used for COSMO (MCH)

```
! !serialize every input
    !$ser savepoint cuadjtq.DoStep-in iteration=test_counter
    !$ser mode my_test_mode
    !$ser data pt=pt(:,kk) pq=pq(:,kk)
    !$ser data ldcnt=ldcnt createpointer
    !$ser data pp=pp(:) createpointer
    !$ser data ldidx=ldidx createpointer
      :
    CALL timer_start(timer_cuadjtq)
!$ACC PARALLEL &
!$ACC PRESENT( ldidx, pt(:,kk), pq(:,kk), ua_idx, ua_zalpha, pp, zppi, ua, dua, uc, ub, ncond, zcond)

       :

!$ACC END PARALLEL
CALL timer_stop(timer_cuadjtq)
    !$ser savepoint cuadjtq.DoStep-out iteration=test_counter
    !$ser mode write
    !$ser data pt=pt(:,kk) pq=pq(:,kk)
```

Requires preprocessing stage in build framework

Credit: Xiaolin Guo, (CSE Master' s student ETH)

ICON-GPU: take-home messages

- Dynamics and advection is good shape; performance as anticipated from memory-bandwidth; being incorporated into ICON trunk;
- OpenACC 2.5 and implementations have key deficiencies
- Some progress on ICON Physics; *much more needs to be done*
- Looking into other paradigms for porting ICON physical parameterizations, such as RRTMGP (e.g., CLAW talk of Jon Rood)
- Many uses for source translation, such as incorporation of directives for code validation on new architectures (e.g., GPU)
- CSCS (T. Schulthess) has made a commitment for development and maintenance of ICON-GPU until 2018 (at least)

Thanks for your attention!