Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement des Innern EDI
Bundesamt für Meteorologie und Klimatologie MeteoSchweiz

Overview and status of ICON @ GPU within IMPACT project

Carlos Osuna, Xavier Lapillonne, Remo Dietlicher, Will Sawyer, and IMPACT

WG6, COSMO General Meeting, 3rd Sept 2020

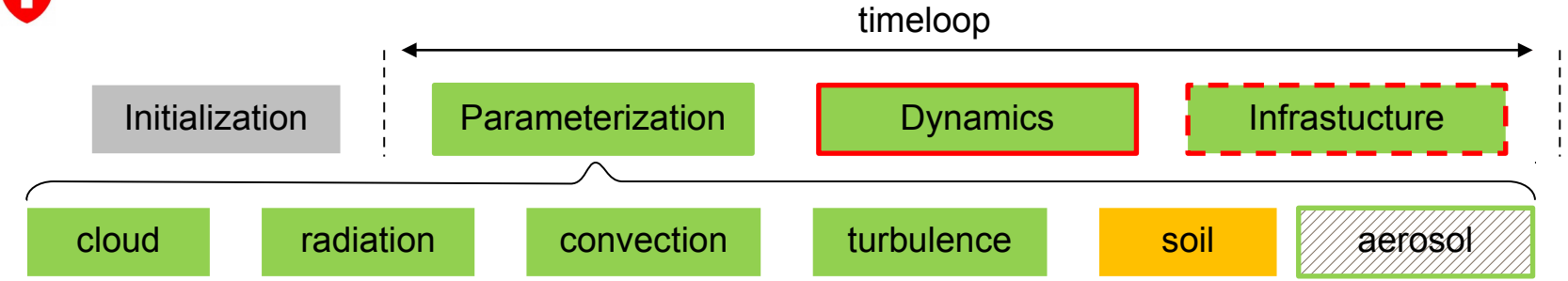
Enabling ICON climate on GPU

- The porting work on the ICON model is part of multiple projects (ENIAC, IMPACT)
- Goal: Adapt the global weather and climate model ICON to run on accelerators
  - Base line GPU port with OpenACC compiler directives
- Achieving a high degree of performance portability by:
  - using source-to-source translation tool : CLAW
- Operational with QUBICC (MPI-M) @2.8 km

Model configuration

- Dynamics/transport on a 2.8 km grid (R2B10) with ca. 200 layers up to ca. 80 km.
- Radiation, Vertical diffusion implicitly coupled to JSBACH-lite"Graupel" cloud microphysics & saturation adjustment

Porting Strategy : full GPU port

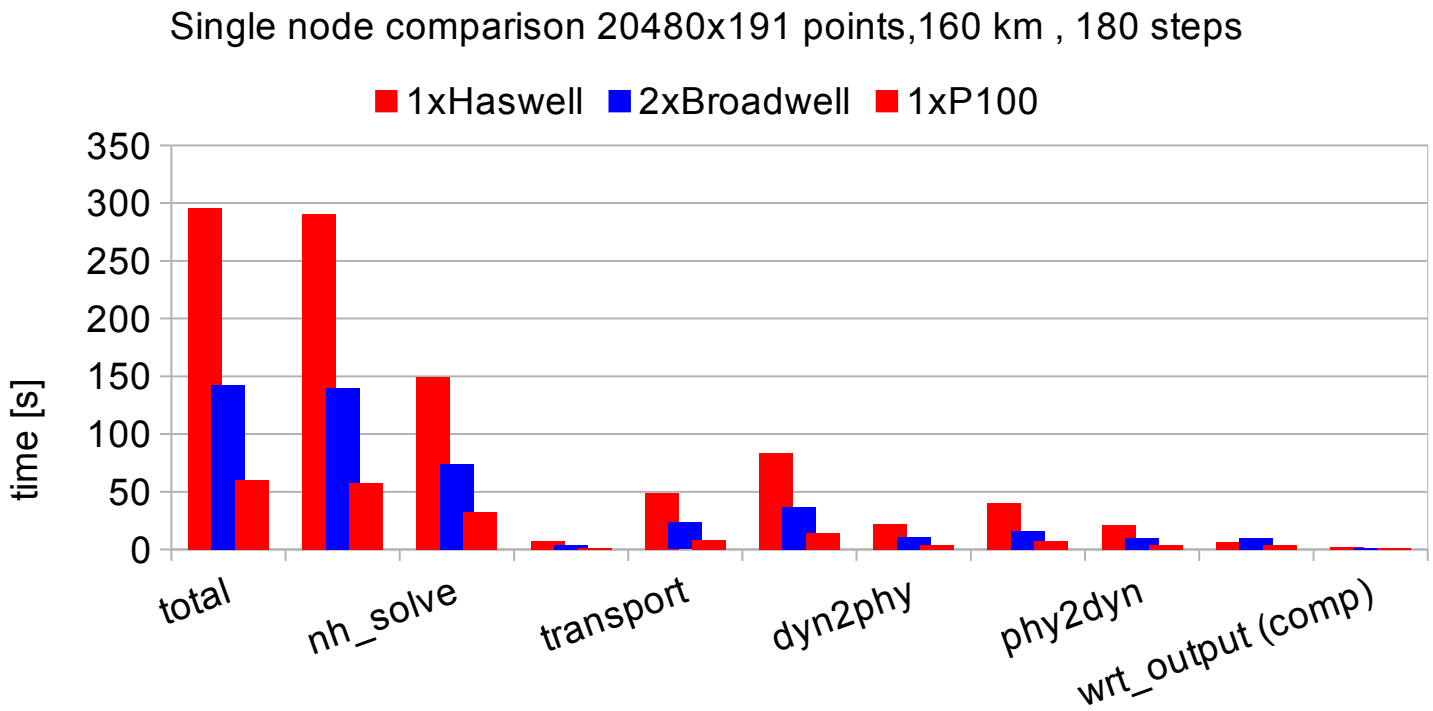


- First base version OpenACC:
  - Dynamics + Climate physics + infrastucture:
  - JSBACH (soil model) :
  - Dynamics and part of the infrastructure already implemented by CSCS (W. Sawyer) prior to ENIAC

OpenACC

claw-dsl

- Required code adaptation and changes
- Only supported compiler for GPU : PGI (requirement for OpenACC 2.6)
- Implement new tolerance base testing, integrated in regular automatic ICON testing infrastructure

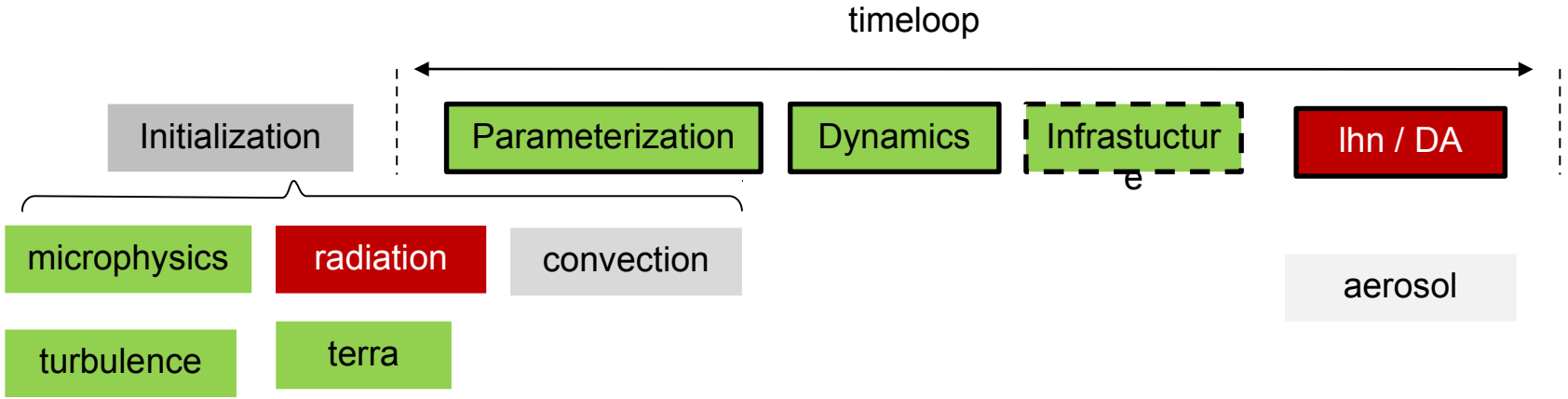


Single socket speedup, 1 Haswell CPU vs1 P100 GPU: 4.9x

Source :W. Sawyer CSCS

4

NWP ICON on GPU



- Good progress on physical parametrization support with OpenACC: micropyhsics (graupel), turbulence, terra, flake integrated back in main ICON repository
- Ihn/ DA will start soon (september 2020)
- Radiation is a blocker (see next slides)

Structure of ecRad solver

Current code:

```
subroutine solver_lw

    horizontal loop
      initialize fields...
      vertical loop                          <---- 60-80 iterations
        call calc_gamma                      <---- contains wavelength loop
        call calc_transmittance              <---- contains wavelength loop
      end vertical loop

      call compute_fluxes                    <---- contains wavelength loop
      some direct calculations…
    end horizontal loop

end subroutine solver_lw

...

subroutine calc_gamma

    wavelength loop                          <---- 112 iterations
      some direct calculations…
    end wavelength loop

end subroutine calc_gamma
```

Structure of ecRad solver

Current code:

```
subroutine solver_lw

    horizontal loop
      initialize fields...
      vertical loop
        call calc_gamma
        call calc_transmittance
      end vertical loop

      call compute_fluxes
      some direct calculations…
    end horizontal loop

end subroutine solver_lw

...

subroutine calc_gamma

    wavelength loop
      some direct calculations…
    end wavelength loop

end subroutine calc_gamma
```

<---- 60-80 iterations
<---- contains wavelength loop
<---- contains wavelength loop

<---- contains wavelength loop

<---- 112 iterations

Needed for OpenACC:

```
subroutine solver_lw

        horizontal loop
            initialize fields…
        end horizontal loop
        call calc_gamma
        call calc_transmittance

        call compute_fluxes
        horizontal loop
            some direct calculations…
        end horizontal loop

end subroutine solver_lw

...

subroutine calc_gamma

        horizontal loop
        wavelength loop
        vertical loop
            some direct calculations…
        end vertical loop
        end wavelength loop
        end horizontal loop

end subroutine calc_gamma
```

Structure of ecRad solver

Current code:

```
subroutine solver_lw

    horizontal loop
      initialize fields...
      vertical loop
        call calc_gamma
        call calc_transmittance
      end vertical loop

      call compute_fluxes
      some direct calculations
```

<---- 60-80 iterations
<---- contains wavelength loop
<---- contains wavelength loop

<---- contains wavelength loop

Problem:

Many fields in the solver of size:

(number of levels)*(number of wavelengths)

This is 112 times what other parametrizations need.

To fit this into GPU memory, we would need to decrease the block size by a factor of ~112.

→ not enough parallelism to keep GPU occupied

Solution:

Harness wavelength parallelism

<---- 112 iterations

Needed for OpenACC:

```
subroutine solver_lw

        horizontal loop
            initialize fields…
        end horizontal loop
        call calc_gamma
        call calc_transmittance

        call compute_fluxes
        horizontal loop
            some direct calculations…
        end horizontal loop

end subroutine solver_lw

...
```

Problem:

Radiation code must be frozen because of large refactoring

Porting involves refactoring CPU code *and* writing OpenACC code

Solution:

Automate refactoring + OpenACC, but...

Ecrad on GPU

- An OpenACC port of ecrad requires a significant refactoring (loop structures) which would lead to source code duplication
- Not acceptable for maintenance reasons
- Ecrad @ICON needs to stay close to the main developments @ECMWF
- Exploring a solution with CLAW source-to-source transformation:
  - tool is not yet ready for the radiation structure and transformations required
  - investigating funding solutions to continue developments of CLAW

Conclusions

- Good progress on GPU port of OpenACC for both climate & NWP mode
- Good progress on DSL (see previous talk) for portability of dynamical core of ICON
- Currently radiation (ecrad) is a blocker to enable NWP on GPU
- Exploring funding solutions to develop CLAW for ecrad to retain a single source code