On the Potential of Concurrency for Scaling W&C Applications in the ExaSCALE Era

Rupert Ford, STFC Ukri

Reinhard Budich, MPI-M

Julia Duras, DKRZ

MAX-PLANCK-INSTITUT FÜR METEOROLOGIE

esiwace CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Why this Presentation?

- ➡ Attempt in ESiWACE to tackle the topic „Concurrency“
- ➡ People had radiation and OBGC in mind, and may be others
- ➡ More activity in that direction as well

- ➡ Somehow we got stuck: Why?
- ➡ Report* had to be written, the base for this talk

* Available on request from Reinhard

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

Concurrency: Which One?

- Rob Pike is a Canadian programmer and author: Go, Bell Labs, now at google.
- From Wikipedia:
  „According to Rob Pike, concurrency is the composition of independently executing computations, [2] and concurrency is not parallelism:
  - **Concurrency** is about dealing with lots of things at once but **parallelism** is about doing lots of things at once.
  - **Concurrency** is about structure, **parallelism** is about execution,
  - Concurrency provides a way to structure a solution to solve a problem that may be (but is not necessarily) parallelizable.[3]"

(2) „Go Concurrency Patterns". https://talks.golang.org/2012/concurrency.slide#6 talks.golang.org. Retrieved 2021-04-08.

(3) "Go Concurrency Patterns". https://talks.golang.org/2012/waza.slide#8 talks.golang.org. Retrieved 2021-04-08.

Concurrency: Which One?

- Definitions for C&W computing problems / type of concurrency
  - a) ensembles / external
  - b) model components / coupling
  - c) comm. and comp. / I/O servers
  - d) data structures / nodes or threads
- Here
  - e) functions / intra-component ➡ **Functional Concurrency**

MAX-PLANCK-INSTITUT FÜR METEOROLOGIE

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

esiwace CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

This Talk

- ➡ Intro
- ➡ Ansatz for a definition of functional concurrency
- ➡ Findings about experiences in the community
- ➡ The potential of functional concurrency
- ➡ Why it is so hard
- ➡ Potential Solutions & Outlook

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

Functional Concurrency: Ansatz for a Definition

- The report describes it as:
  „Model components can often also be further split into finer-grain components with some of these components having the potential to run in parallel with each other. This intra-component functional parallelism is currently typically not exploited by model components ...“
- Examples include, but are not limited to
  - atm - rad
  - oce - obgc
  - lnd - rivers
  - ...

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

Findings about Experiences in the Community

- Examples
  - MPAS
  - COSMO
  - LFRic
  - ICON

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

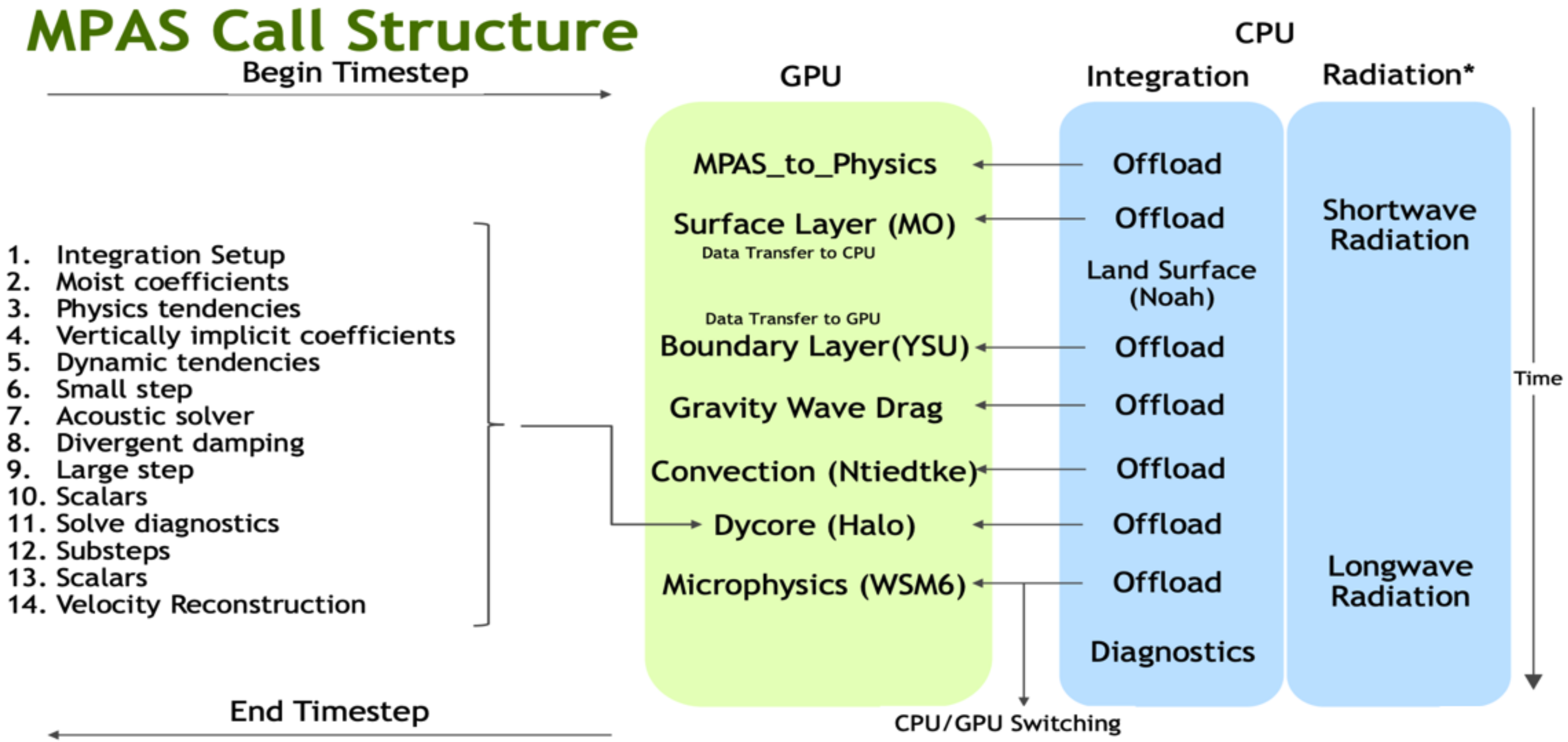
MPAS: Courtesy Rich Loft

(Inter-component task concurrency in ESM, not considered here)

- Intra-component task concurrency
    - Look at cost, refactoring ease, expected acceleration, potential for software reuse
    - Rad and lnd are on CPU, rest on GPU
    - cpu as „Co-processor“

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

MPAS: Courtesy Rich Loft



Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

MPAS: Courtesy Rich Loft, Remarks

- Radiation
  - To GPU
    - Rad RRTMG was not yet ported to GPU, as it is now the case.
    - Rad was 1/3 of the LoC of MPAS-A: Major task!
    - Many lookup-tables problematic for memory access
    - Not clear if spreading the task to many GPUs would scale and perform
  - On CPU
    - Would rad@cpu be fast enough for the rest? Also depending on ratio between
      - rad and dycore time step!
      - Performance of cpus and gpus

MPAS: Courtesy Rich Loft, Conclusions

- It depends ...
  - Please refer to the report: Tremendous work!
- ... on implementation of the concurrency in the component(s)

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

COSMO - Preliminaries: Courtesy Carlos Osuna

- NWP code of MeteoSuisse parallelized with
  - OpenACC for Driver, parametrization, assimilation
  - GridTools for the dycore
- Observation
  - Strong scalability saturates at 40 k grid points per GPU device
  - GPUs can not be fully occupied, adding GPUs does not help
- Idea
  - Run dycore and rad (every 60^{th} time-step) in parallel as GPU streams, OpenMP and OpenACC applied (async queues)
- Hypothesis
  - This way GPUs can be utilized in a way nearer to saturation

COSMO - Results: Courtesy Carlos Osuna

- Results
  - No bit-identical results, but differences should be small from theoretical considerations - full meteorological evaluation was not performed, though.
  - Quote from report: „Even though we observed the desired task parallelism in the radiation scheme, we could not measure a significant improvement in performance corresponding to it.„
- Reasons
  - (Too) Large kernels in the rad scheme use all resources, and block other kernels
  - Controlling launch times for kernels would help here, but was not possible from OpenACC
- My take
  - Smart strategy, but disappointing results
  - Strategy dropped for the time being
  - Dependent on implementation of the concurrency in the component(s)

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

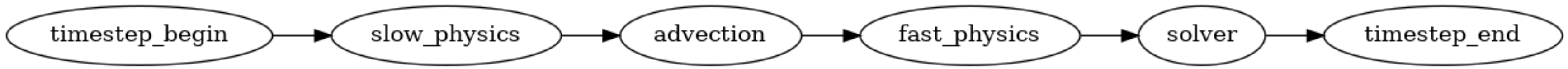
LFRic: Courtesy Rupert Ford

- W&C atmosphere code of MetOffice to come
  - Not yet complete, not yet with the desirable performance, but useable for prototypical case studies
  - Concurrency for components and data treated as usual, also with DSL
    (for functional and task low level Conc. on loop level)
- Approaches: High level concurrency
  - for slow physics, within a time-step
    - processes are independent of each other within a tilmestep
    - (Fast physics are not considered and executed sequentially)
  - for radiation, across time-step
    - Use physics consideration
    - Rad status is not (very) dependent on parameters in the same time-step, information can be used from earlier time-steps

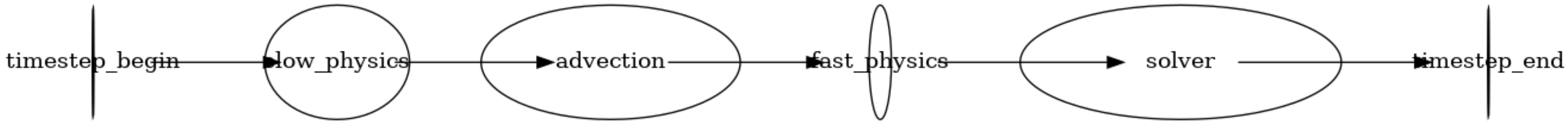
Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

LFRic, slow physics: Courtesy Rupert Ford

- Time-step in LFRIC as DAG:



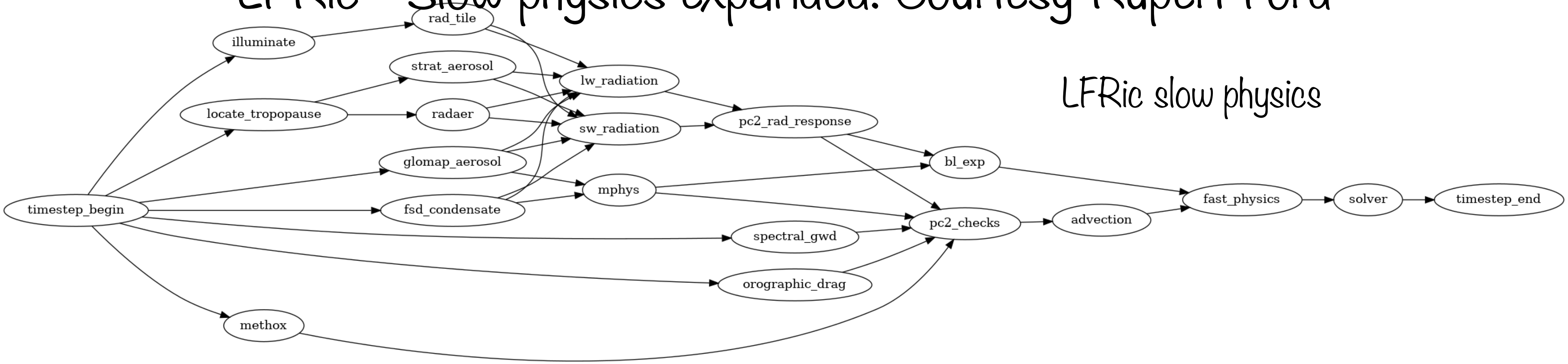
- Add timing information, TS weighted



MAX-PLANCK-INSTITUT FÜR METEOROLOGIE

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

LFRic - Slow physics expanded: Courtesy Rupert Ford

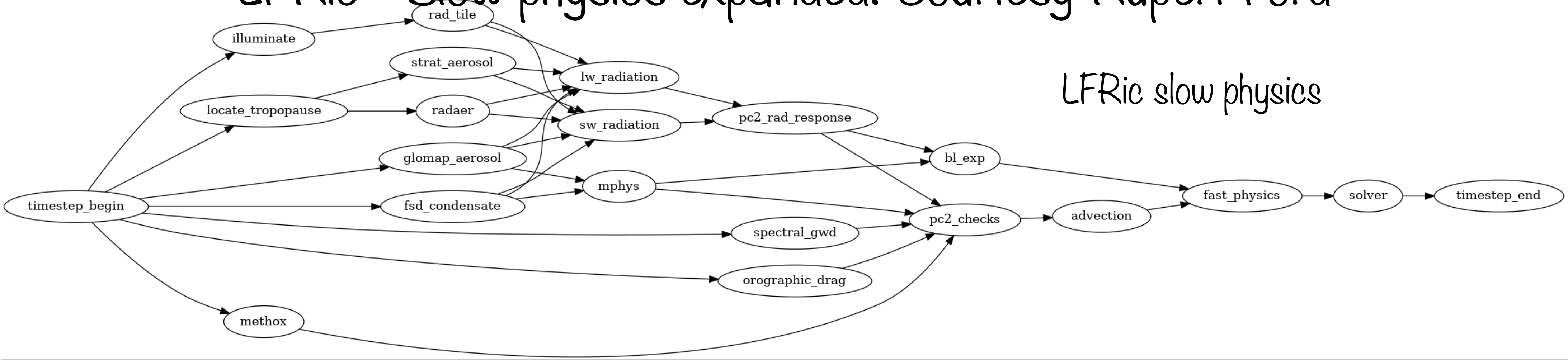


LFRic slow physics

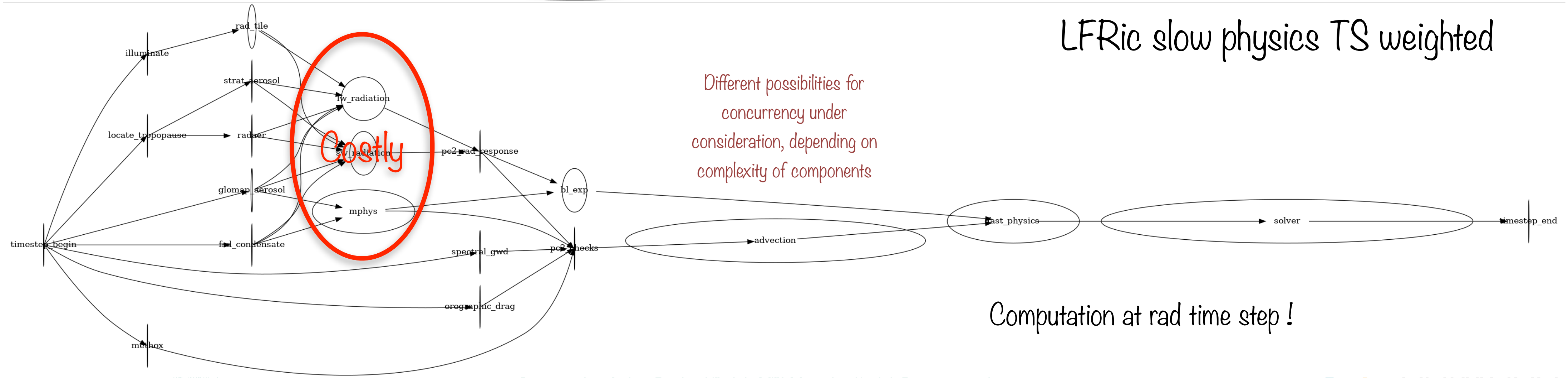
Computation at rad time step !

LFRic - Slow physics expanded: Courtesy Rupert Ford

LFRic slow physics



LFRic slow physics TS weighted



Different possibilities for concurrency under consideration, depending on complexity of components

Computation at rad time step !

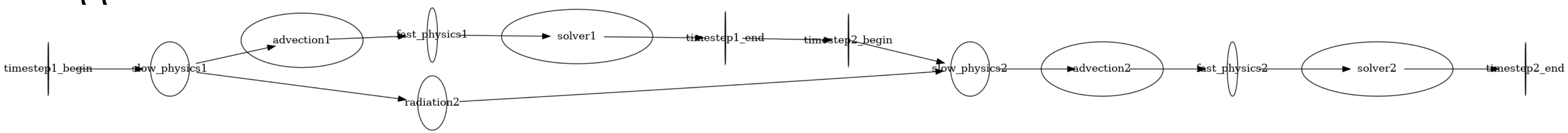
MAX-PLANCK-INSTITUT FÜR METEOROLOGIE

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

LFRic, concurrent rad: Courtesy Rupert Ford

- Approach:



- Results:
  - It depends
    - on physical assumptions, and cost benefit considerations by scientists
    - on the exact balance of costs for the different components, depending on their complexity and the frequency with which they are computed
    - on implementation of the concurrency in the component(s)

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

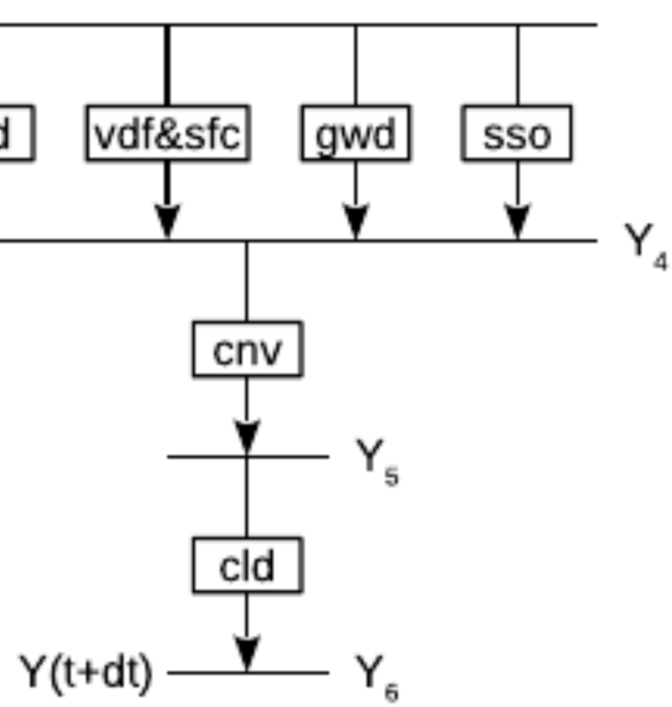
ICON: C. Many Iconists - Marco Giorgetta, Leonidas Linardakis, Julia Duras ...

- Complex coupled weather and climate model of DKRZ, DWD, KIT and MPI-M
    - Atm, Lnd, Oce
    - Provides ample concurrency in all dimensions
        - Coupling of the atmosphere and the ocean via the coupler library YAC
        - I/O-Server library CDI-PIO, enabling concurrent execution of I/O
        - Data parallelism within model components employing OpenMP and MPI
        - functional concurrency in ICON-A, ICON-O
- OBGC (HAMOCC)
    - provides a concurrent ocean case
    - Described elsewhere
    - It runs well (T. Ilyina, pers. comm.)
- Look at rad in ICON-A here

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

ICON-A: C. Many Iconists - Marco Giorgetta, Julia Duras ...

- Process splitting for slow physics in ICON-A (M. Giorgetta)
  - Coarse res / long time-steps can lead to higher tracer concentrations then the resource (neg. concentrations) -> physically not reasonable, numerically unstable
  - High res / short TS configurations allow to reduce number of processes (rad, diff, cloud µ-physics)
    - Sequential coupling currently
    - But considered for complete parallel coupling (some remarks next slide)
    - So far, partial parallel coupling was considered



Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

Remarks on Complete Parallel Coupling in ICON-A

- Parallel coupling and computing of more than one "slow physics" process and dynamics not yet been done
  - It is a tremendous work, for which sustainability is not guaranteed
  - Parallel coupling scheme
    - Two synchronizations necessary:
    - Combine with some monitoring and regularization scheme to prevent rare exceptions

- Distribute the common input state to all processes computed in parallel
- Collect and process their output within the same time step

Remarks on Complete Parallel Coupling in ICON-A

- Parallel coupling and computing of more than one "slow physics" process and dynamics not yet been done
  - It is a tremendous work, for which sustainability is not guaranteed
  - Parallel coupling scheme
    - Two synchronizations necessary:
    - Combine with some monitoring and regularization scheme to prevent rare exceptions
- Scientifically parallel coupling scheme could be developed independently:
  - Once an accepted coupling scheme for a certain experiment is found, working with acceptable time steps (i.e. computing costs), such a parallel computing scheme could be developed and employed
  - Sustainability of the implementation questionable

- Make input data and the tasks available
- Get work done as fast as possible
- Collect results from all processes
- Combining them ready for the next phase of computation

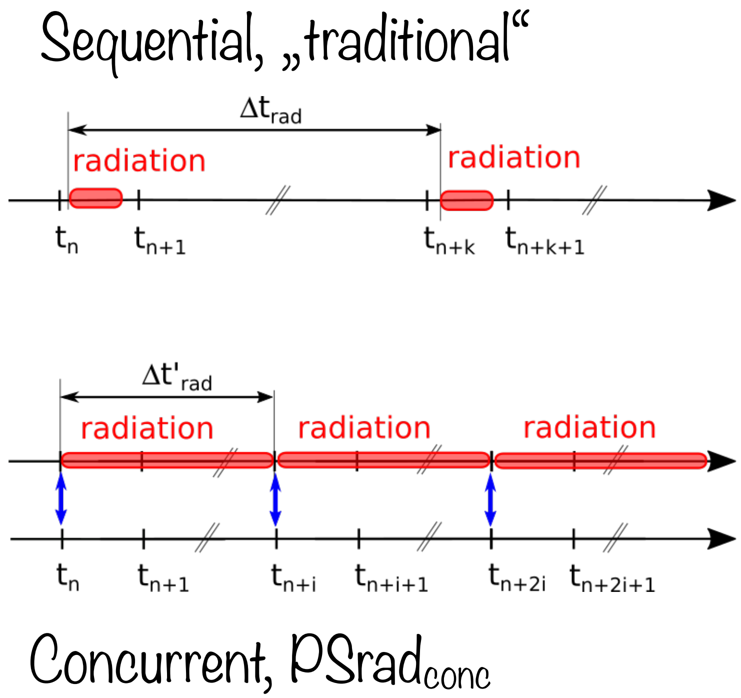
Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

Remarks on Complete Parallel Coupling in ICON-A

- Parallel coupling and computing of more than one "slow physics" process and dynamics not yet been done
  - It is a tremendous work, for which sustainability is not guaranteed
  - Parallel coupling scheme
    - Two synchronizations necessary:
    - Combine with some monitoring and regularization scheme to prevent rare exceptions
- Scientifically parallel coupling scheme could be developed independently:
  - Once an accepted coupling scheme for a certain experiment is found, working with acceptable time steps (i.e. computing costs), such a parallel computing scheme could be developed and employed
  - Sustainability of the implementation questionable
- With a completely parallelized computation of processes, distribution of resources to the tasks can be embedded in the model, invisible to the user
- Dynamical process scheduling might be an option

Partial parallel coupling in ICON-A

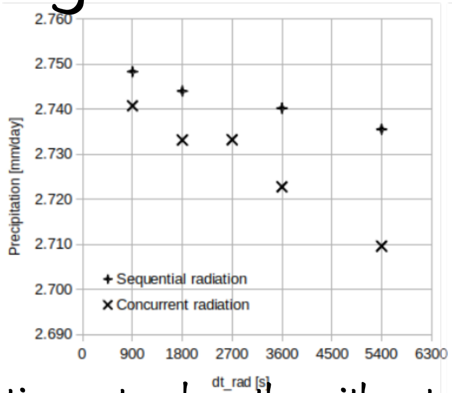
- Concurrent computation of radiation process
  - Use of longer rad time-step:
    $\Delta t_{rad} = k \cdot \Delta t,\ k > 1$
    - The larger k, the more the radiation will deviate from „true“ solution - affordable
  - Concurrent PSrad was developed
    - Own processes

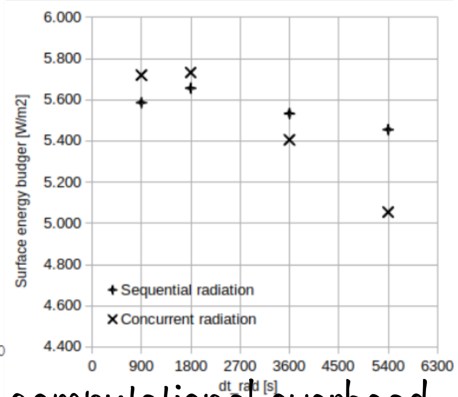


- different coupling scheme, different causality, different results

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

Partial parallel coupling in ICON-A





- Results:
  - 25 quantities checked in 9 yr AMIP-like runs
  - Deviation is in the range of a few percent
  - Radiation accuracy can be increased by a reduced time-step length, without computational overhead
  - Computational gain ok, though not terrific:
    - integration of the whole model gets faster by about the amount needed for the radiation computed sequentially
  - Very much depending on resource distribution between PSrad$_{conc}$ and „rest of the model"
  - Practical complications and intense testing necessary <- Very costly, not affordable in our case!
  - And: PSrad was dropped for other reasons

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

ICON: Conclusion

- ICON-A:
  - More work to be done
  - Re-structuring and -engineering in full swing (DestinE, Warmworld, others)
  - For functional concurrency, extra project funding applied for
- Dependent on implementation of the concurrency in the component(s)
- Seems to have worked for ICON-O/HAMOCC (OBGC)

The Potential of Functional Concurrency

➡ Dependent on implementation of the concurrency in the component(s)

The Potential of Functional Concurrency

- Dependent on implementation of the concurrency in the component(s)
- No generic approach visible, so case- and model-dependent
- AND: It is depending on intense scientific evaluation & cooperation
- But the only potential candidate for more parallelization
- Huge cost involved

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

Why it is so hard

- Radiation is special case since Δt_{rad} >> $\Delta t_{rest\ slow\ physics}$
- Other (sub-time-step) components need to be **modified scientifically** before they can be algorithmically and numerically altered in order to achieve concurrency ☛ ***High effort, high risk***
- However, having the flexibility in the software to be able to test whether it is beneficial to run sub-components concurrently or not and being able to choose a different solution depending on the required configuration and underlying architecture is something that would be beneficial for scientists.
- In the report different scenarios concerning properties of the sub-components and the resource availability are discussed which would benefit from such flexibility
    - Flexibility in ordering of SC, or choosing different resolution, rates or precision might also help - discussed in the report
- Conclusion of this report: A well defined interface (of which a subroutine boundary with all data being passed by argument is considered sufficient) is necessary in order to be able to maintain a more flexibility for concurrency; two of four models have this already
- Depending on machine architecture, a mix of MPI&OpenMP&OpenACC seems most appropriate - also hard!
- Buffer space needs to be allocated to store information from former time steps, also depending on the scientific set-up
- Complexity seems to grow overwhelmingly

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

Potential Solutions

- Ideas/recommendations from the report for further investigation:
- Coupling systems and frameworks show potential
  - ESMF
  - DSL
  - Combination of library code, code-generation and -modification (CCPP@NCAR)
    - Lib between component and sub-component as a SC-wrapper

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

Outlook

- Technology changes
  - NIVIDIA'S Multi-Instance GPU for better load balancing
- Further problems:
  - Proliferating programming codes
  - Heterogeneity
- Big projects plus pressure for more parallelism might help ...

- We need to use the concept of functional concurrency to structure the problems in a way to parallelize execution of taylor-made codes of certain physical processes

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de

Thanks!

➡ Questions

ESiWACE2 has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 823988

Corresponding Author: Reinhard Budich, MPI-M, reinhard.budich@mpimet.mpg.de