A Language-Interoperable C++-Based Memory-Manager for the ICON Climate and Weather Prediction Model

Claudius Holeksa[1], Hartwig Anzt[1], Jörg Behrens[2], Christopher Bignamini[5], Yen-Chen Chen[1], Terry Cojean[1], Claudia Frauen[2], Daniel Klocke[3], Luis Kornblueh[3], Sergey Kosukhin[3], Xavier Lapillonne[6], Ralf Müller[2], Florian Prill[4], Will Sawyer[5]
[1]KIT, [2]DKRZ, [3]MPI-M, [4]DWD, [5]CSCS, [6]MeteoSwiss

SCAN ME

Find out more: https://warmworld.de

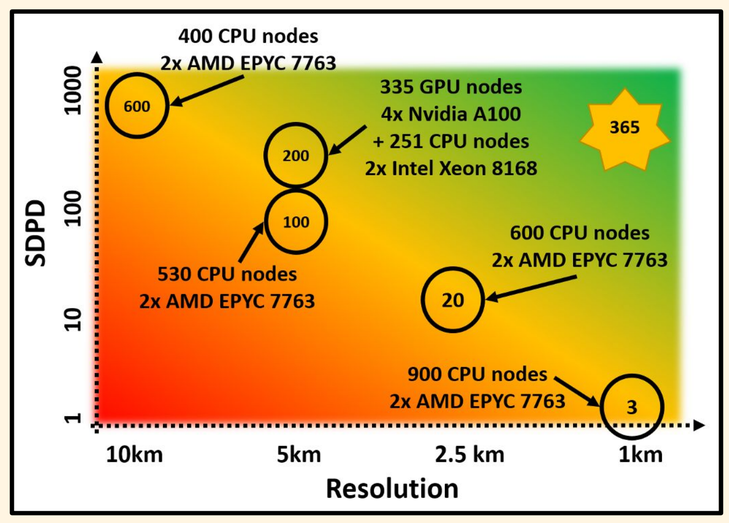
Climate modeling in the exascale era - New possibilities and challenges

- **Exascale supercomputers**'s high performance allows climate simulations to target **high-resolution simulations** (e.g. 2.5 km) for longer simulation times
- But **heterogeneous hardware architectures** (CPU+GPU, vector systems, …) are a challenge for scientific software development
- **WarmWorld** is a German national project that aims to use **advances in information technology to compute and evaluate climate warming trajectories**.

WarmWorld Faster

Target Global Objectives

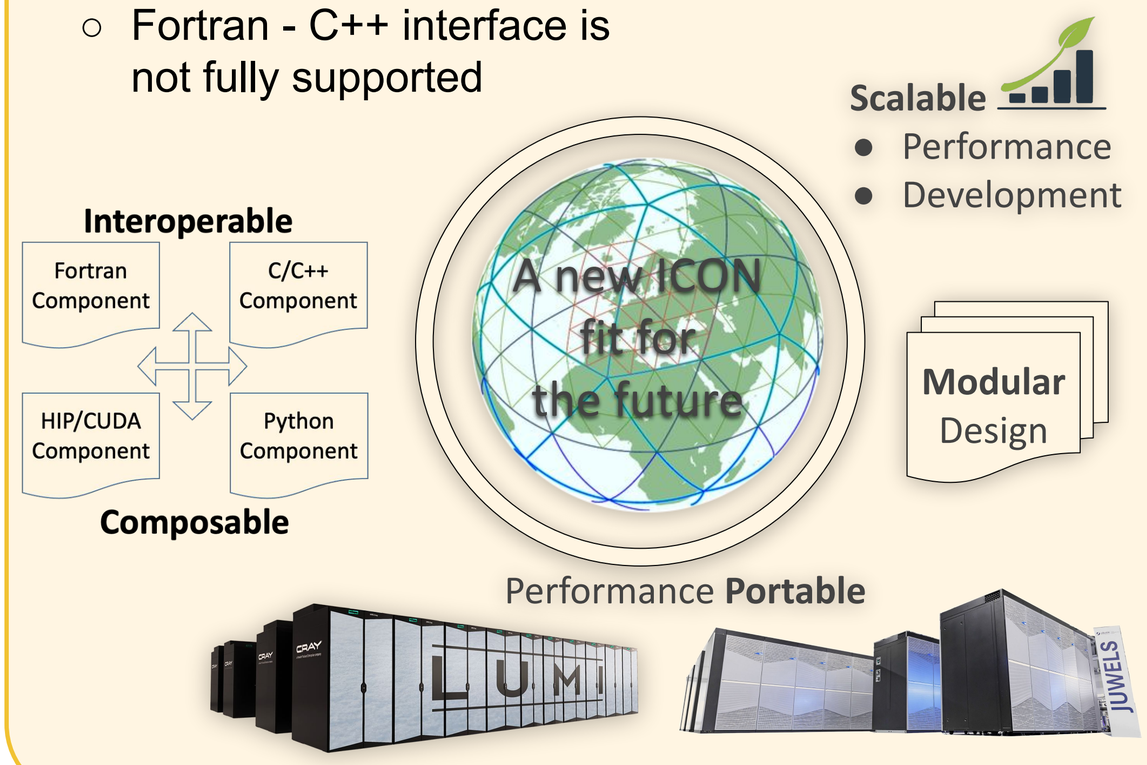
1. Enable portable performance improvement with scalable development
2. Free, Open Source, refactored ICON for scalable development
3. Over 0.5 simulated years per day (SYPD) on ≤ 2.5km grid



A language-interoperable and platform portable memory-manager

Challenge of Language Interoperability

- Challenges
  - The memory manager has to register various fields
  - Many ICON components touches the fields
  - Most vendor are C++ based while ICON is Fortran based (Kokkos, RAJA,...)
  - Fortran - C++ interface is not fully supported
- Opportunities
  - Modularize the code
  - Easy extension
  - Favor relying on existing building blocks
  - Better GPU support for a C++ front end than the legacy fortran code



Memory Manager Design

Idiomatic interface, e.g. Fortran

```
interface
  function register_real32(ctx, desc_c, &
                           & num_elem) &
              & result(err) bind(c)
    …
  end function register_real32
end interface
contains
function register_real(ctx_id, desc, &
                       & kind, num_elem) &
            & result(err)
  …
  select case(kind)
    …
    err = register_real32(get_ctx(ctx_id), &
                          & to_c_desc(desc), &
                          & int(num_elem, &
                              & kind=c_int))
    …
  end select
end function
```

C for ABI

```
struct ctx;
struct var_descriptor;

int register_real32(
  struct ctx* device,
  struct var_descriptor* desc,
  const size_t num_elem);
```

C++ based core backend written in a generic fashion

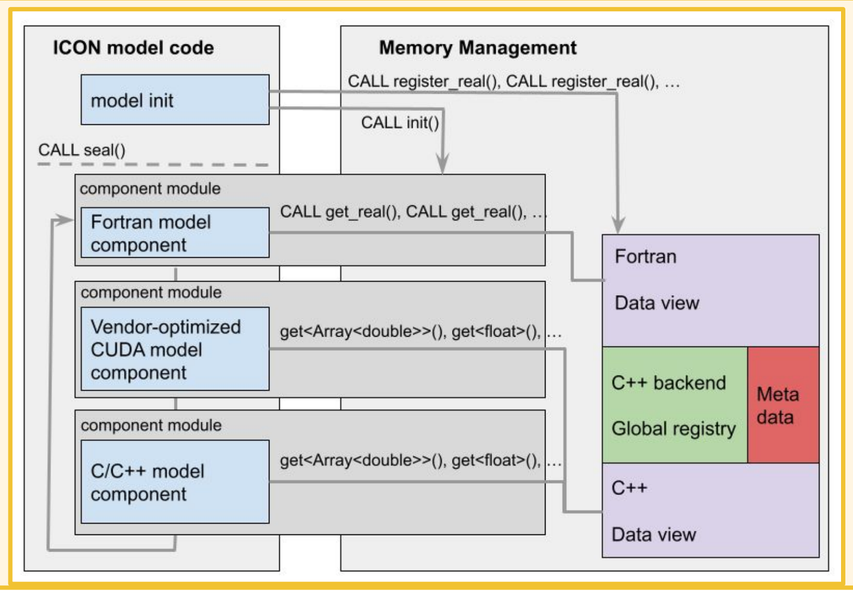
```
template<typename Storage, typename T>
uuid_t Registry::register_var(T&& value);

template<typename K, typename… T>
class KeyedRegistries;

template<typename K, typename… Store, typename… T>
class KeyedRegistries<K, Registry<Store,T>…> {
  std::unordered_map<K, uuid_t> index;
  std::variant<Registry<Store,T>…> regs;
public:
  // Access / Insertion / Manipulation / Deferred Allocation
};
```

Towards sustainable software development of the ICON Climate and Weather Prediction Model

Leveraging the memory-manager for sustainable development

Integrating the memory-manager within the ICON climate and weather prediction model allows to bring sustainable development and to introduce further optimizations:

1. Iterative modularization to generate independent model components
2. Well-defined interfaces allows to replace components by e.g. vendor-optimized code or third-party components
3. Allows easy serialisation of data, eases component testing
4. Abstraction allows reuse by other projects
5. Towards a data-driven control flow of the ICON code base



SPONSORED BY THE
Federal Ministry of Education and Research
ICON
Karlsruher Institut für Technologie
Deutscher Wetterdienst
Wetter und Klima aus einer Hand
DWD
DKRZ
MeteoSwiss
MAX-PLANCK-INSTITUT FÜR METEOROLOGIE
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre