

preWarmWorld

BMBF Forschung für Nachhaltige Entwicklung (FONA)

Sachbericht zum Verwendungsnachweis Teil I

Projektpartnern	Deutsches Klimarechenzentrum, Forschungszentrum Jülich, Max-Planck-Institut für Meteorologie
Vorhabenbezeichnung	01LK2106
Laufzeit des Vorhabens	01.10.2021 – 30.09.2023

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Über dieses Dokument

Dieser Bericht bezieht sich auf die Arbeit, die im Rahmen des preWarmWorld Projekt durchgeführt wurde. - Kurzbericht

Leitautor*in:

Daniel Klocke, MPI-M

Kontakt: daniel.klocke@mpimet.mpg.de

Besuchen Sie uns auf: www.warmworld.de

1 Projektrahmen

Im Vorhaben WarmWorld soll - aufbauend auf der führenden Rolle und Expertise der deutschen Klimaforschung und den signifikanten Investitionen der EU in neue Technologien - die Grundlage für ein innovatives System zur Erstellung von Klimaprojektionen geschaffen werden. Zentrale Komponente dieses Systems ist das ICON Atmosphärenmodell, welches ursprünglich vom Max-Planck-Institut für Meteorologie und dem Deutschen Wetterdienst entwickelt wurde und in einer erweiterten Partnerschaft mit dem Deutschen Klimarechenzentrum und dem Karlsruher Institut für Technologie weiter entwickelt wird. Das Modell soll global auf einem feinmaschigen Netz von 2 km gerechnet werden und ermöglicht es dadurch im Vergleich zu heutigen Modellen, die Gitterweiten um die 100 km haben, physikalische Prozesse direkt zu berechnen, anstatt sie, wie bisher, aus empirischen Annahmen abzuschätzen.

Der ICON Code ist, wie alle erprobten Klima- und Wettermodelle, über viele Jahre gewachsen und inzwischen sehr umfangreich sowie bereits auf vielen HPC-Systemen im Einsatz. Die aktuellen technischen Entwicklungen im HPC-Sektor erzwingen jedoch neue Ansätze und Programmiermethoden, um die nächste Rechnergeneration effizient nutzen zu können. Gleichzeitig soll in WarmWorld der Sprung zu einer neuen wissenschaftlichen Qualität der Modelle erfolgen. Beides erfordert ein "redesign" großer Teile des Codes, welches von den aktuellen Entwicklern parallel zu Ihren wissenschaftlichen und operationellen Aufgaben nicht leistbar ist.

Das Projekt preWarmWorld legte die Grundlagen für die Restrukturierung und Modernisierung, des ICON Modellcodes und der Lizenz und Governance-Struktur, um eine Transformation zu einer modernen Software zu starten, die agiles und skalierbares entwickeln, sowie effizientes Nutzen erlaubt. Konkret wurden Programmierkonzepte evaluiert und eine Blaupause für die Entwicklungen in WarmWorld für die Software-, Test- und Entwicklungsstrategien entwickelt.

2 Ablauf des preWarmWorld

Das zwei Jährige Projekt war in drei Arbeitspakete unterteilt, mit den folgenden Zielen:

1. Die Entwicklung einer Software Blaupause, die mit den ICON Entwicklungspartnern abgestimmt wird und die Basis für die Umstrukturierung und Modernisierung von ICON bildet.
2. Evaluierung von verschiedenen Programmiermodellen und Prototyp Implementierungen zur Bewertung der Performanz, Portability und Wartbarkeit.
3. Erstellung einer Umgebung, die eine agile und skalierbare Entwicklung der ICON Software erlaubt. Neben technischen Tools wie GIT und einer Testinfrastruktur beinhaltet dies auch die Governance und Lizenz Modell.

3 Wesentlichen Ergebnisse

Das preWarmWorld Projekt hat wesentlich dazu beigetragen, dass die ICON Entwicklung klare Governancestrukturen über die Partnerinstitute hinweg hat und dass ICON jetzt unter der BSD-3C OpenSource Lizenz veröffentlicht wird (ICON partnership (DWD, MPI-M, DKRZ, KIT and C2SM), 2024). Das offene ICON vereinfacht Kooperationen und externe Beiträge aus Forschung und Wirtschaft.

Für das Anschlussprojekt WarmWorld wurden wichtige Grundlagen in der Infrastruktur für die Entwicklung geschaffen. Dies beinhaltet vereinfachte Strukturen und Richtlinien für die Entwicklung im GIT Arbeitsablauf, Definition von Abläufen zur Restrukturierung von Programmteilen und eine integrierte Testinfrastruktur, die auf Systemen der Projektpartner (JSC und DKRZ) und darüber auf weiteren Testsystemen implementiert wurde. Dies beinhaltet die Implementierung vom gekoppelten ICON-Modell auf der "modular supercomputing architecture" am JSC.

Einzelne Teile der ICON Software wurden isoliert (Advektion und Mikrophysik) um verschiedenen Programmierparadigmen zu testen und im Bezug auf ihre Performanz, Portierbarkeit und Wartbarkeit zu evaluieren. Für die Weiterverfolgung in WarmWorld-Faster konnten AnyDSL und eDSL als nicht geeignet, bzw. noch nicht ausgereift genug, ausgeschlossen werden. GridTools wird in dem Schweizer Projekt EXCLAIM implementiert und C++ basierte Ansätze wurden als die vielversprechendsten für WarmWorld identifiziert.

Die preWarmWorld entwickelten Softwareblaupausen für ICON sind in die Konzeptionierung der Konsolidierung von ICON eingeflossen und dienen auch als Grundlage in WarmWorld-Faster.

4 Bibliografie

ICON partnership (DWD, MPI-M, DKRZ, KIT and C2SM) (2024). *ICON release 2024.01*. DOI: [10.35089/WDC/IconRelease01](https://doi.org/10.35089/WDC/IconRelease01).

preWarmWorld

BMBF Forschung für Nachhaltige Entwicklung (FONA)

Sachbericht zum Verwendungsnachweis Teil II

Projektpartnern	Deutsches Klimarechenzentrum, Forschungszentrum Jülich, Max-Planck-Institut für Meteorologie
Vorhabenbezeichnung	01LK2106
Laufzeit des Vorhabens	01.10.2021 – 30.09.2023

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Über dieses Dokument

Dieser Bericht bezieht sich auf die Arbeit, die im Rahmen des preWarmWorld Projekt durchgeführt wurde. - Eingehende Darstellung

Leitautor*in:

Daniel Klocke, MPI-M

Beitragende Autoren*innen:

Hendryk Bockelmann (DKRZ), Lars Hoffmann (FZ-Jülich), Yen-Sen Lu (FZ-Jülich)

Kontakt: daniel.klocke@mpimet.mpg.de

Inhaltsverzeichnis

1	Einleitung	4
2	Erreichung der Projektziele und wichtigste Ergebnisse	4
2.1	Meilensteine	4
2.2	Arbeitspakete	7
2.2.1	Arbeitspaket 1	7
2.2.2	Arbeitspaket 2	9
2.2.3	Arbeitspaket 3	13
3	Weitere Entwicklungen	15
4	Auswirkungen und Verbindungen zu anderen Projekten/Programmen	16
4.1	Verbundübergreifende Tätigkeiten, wie Meetings, Austausch, Workshops, etc. .	16
5	Veröffentlichungen	17
6	Bibliografie	17

1 Einleitung

Das Projekt preWarmWorld ist Teil des Vorhaben WarmWorld. Es startete früher, um die strategischen, organisatorischen und technischen Grundlagen für die Arbeiten in WarmWorld (Better, Faster, Easier) zu legen.

Im Vorhaben WarmWorld soll - aufbauend auf der führenden Rolle und Expertise der deutschen Klimaforschung und den signifikanten Investitionen der EU in neue Technologien - die Grundlage für ein innovatives System zur Erstellung von Klimaprojektionen geschaffen werden. Zentrale Komponente dieses Systems ist das ICON Atmosphärenmodell, welches ursprünglich vom Max-Planck-Institut für Meteorologie und dem Deutschen Wetterdienst entwickelt wurde und in einer erweiterten Partnerschaft mit dem Deutschen Klimarechenzentrum und dem Karlsruher Institut für Technologie weiter entwickelt wird. Das Modell soll global auf einem feinmaschigen Netz von 2 km gerechnet werden und ermöglicht es dadurch im Vergleich zu heutigen Modellen, die Gitterweiten um die 100 km haben, physikalische Prozesse direkt zu berechnen, anstatt sie, wie bisher, aus empirischen Annahmen abzuschätzen.

Der ICON Code ist, wie alle erprobten Klima- und Wettermodelle, über viele Jahre gewachsen und inzwischen sehr umfangreich sowie bereits auf vielen HPC-Systemen im Einsatz. Die aktuellen technischen Entwicklungen im HPC-Sektor erzwingen jedoch neue Ansätze und Programmiermethoden, um die nächste Rechnergeneration effizient nutzen zu können. Gleichzeitig soll in WarmWorld der Sprung zu einer neuen wissenschaftlichen Qualität der Modelle erfolgen. Beides erfordert ein "redesign" großer Teile des Codes, welches von den aktuellen Entwicklern parallel zu Ihren wissenschaftlichen und operationellen Aufgaben nicht leistbar ist.

Das Projekt preWarmWorld legte die Grundlagen für die Restrukturierung und Modernisierung, des ICON Modellcodes und der Lizenz und Governance-Struktur, um eine Transformation zu einer modernen Software zu starten, die agiles und skalierbares entwickeln, sowie effizientes Nutzen erlaubt. Konkret wurden Programmierkonzepte evaluiert und eine Blaupause für die Entwicklungen in WarmWorld für die Software-, Test- und Entwicklungsstrategien entwickelt.

2 Erreichung der Projektziele und wichtigste Ergebnisse

2.1 Meilensteine

Folgende Meilensteine wurden erfolgreich abgeschlossen:

Meilenstein 1 (Monat 9): Blaupause zum SW-Design von ICON in Version 0

- Zusammen mit der ICON-Koordinierungsgruppe wurde beschlossen, die Arbeiten aus preWarmWorld in das Gesamtkonzept von ICON-C (ICON-consolidated) einfließen zu lassen.
- Es wurden entsprechende Arbeitsgruppen zu den wesentlichen Bausteinen gebildet: modularisation/librarisation, Test-Umgebung, memory-management, repository Struktur. Ei-

ne enge Verzahnung der Arbeiten mit dem EXCLAIM Projekt, dem WarmWorld-Faster, sowie Eigenbeiträgen der Partner im ICON-Konsortium ist dadurch etabliert.

- Die standalone-Versionen der ICON-Advektion und Wolkenmikrophysik, welche in pre-WarmWorld implementiert wurden, dienten in ICON-C als Vorlage für weitere Arbeiten.

Meilenstein 1 (Monat 9): Lizenz- und Governance-Modell für ICON

- Basierend auf einem White Paper von MPI-M und DKRZ, wurde das Governance-Modell für ICON angepasst. Ein **ICON Board** (D5) wurde etabliert, in dem jede ICON-Partnerinstitution durch eine Führungskraft vertreten ist. Das ICON Board gibt die strategische Entwicklung von ICON vor. D5 beauftragt die **ICON Koordinierungsgruppe** (C5) die Strategie zu implementieren.
- Das ICON Board (in dem alle ICON Entwicklungspartner vertreten sind) hat entschieden ICON unter die OpenSource Lizenz BSD-3C zu stellen. Im Januar 2024 wurde ICON unter der OpenSource Lizenz veröffentlicht, was die Zusammenarbeit mit Partnern wesentlich erleichtert.
- C5 und eine „ICON Lizenz“ Arbeitsgruppe treibt die Entwicklung zu Open Development weiter vor ran.
- Vorarbeiten zu OpenSource Möglichkeiten und deren Implikationen auf die SW-Entwicklung wurden im Kontext von preWarmWorld durch das MPI-M und das DKRZ eingebracht. Das DKRZ stellt die entsprechende Infrastruktur (gitlab Server) zur Verfügung.

Meilenstein 2 (Monat 12): Testinfrastruktur für ICON

- Nachdem sich das ICON-Konsortium auf die gemeinschaftliche Arbeit an ICON-C geeinigt hatte, wurde die entsprechende Infrastruktur (basierend auf gitlab und buildbot) am DKRZ installiert und der ICON-community zur Verfügung gestellt.

Meilenstein 3 (Monat 20): Demonstrator für das Modularisierungskonzept läuft auf JSC und DKRZ Rechnern

Levante@DKRZ:

Bis August 2023 konnten hetjobs, d.h. modulare Experimente auf verschiedenen Partitionen, aufgrund von Problemen mit dem SLURM Job-Scheduler nicht durchgeführt werden. Dennoch wurden notwendige Vorarbeiten, z.B. das Update des YAC Kopplers auf Version 3 mit dem NVHPC Compiler, unabhängig gelöst, so dass schlussendlich die modulare Kopplung von ICON auf unterschiedlicher Hardware (CPU-GPU) erfolgreich war.

JUWELS@JSC:

Dadurch, dass JUWELS seine Module Cluster (CPU) und Booster (GPU) über eine modulare Supercomputing Architektur (MSA = Modular supercomputing architecture) (Suarez u. a., 2019) verbindet, war es notwendig, dass die 2 Binaries auf den jeweiligen Maschinen gebaut werden. Hierfür benötigt jede Maschine einen eigenen Configure Wrapper, welche zusammen mit dem JSC erstellt wurden. Die FORTRAN Version von YAML musste vom JSC zunächst als

Nutzerinstallation gebaut werden. Es gibt keinen Runscriptgenerator, welcher auf JUWELS fertige Runscripte erstellt. Stattdessen müssen Runscripte auf Levante erzeugt werden und dann, mit Hilfe von erfolgreich auf JUWELS laufenden Runscripts, angepasst werden. Die Länge (~ 1300 Zeilen) als auch die verschiedenen Varianten solcher bash-scripte erschwerten diese Aufgabe.

Meilenstein 3 (Monat 20): gekoppeltes ICON läuft auf MSA am JSC

Die wesentlichen Ergebnisse stehen in der Veröffentlichung „Earth system modeling on modular supercomputing architecture: coupled atmosphere–ocean simulations with ICON 2.6.6-rc“ (Bishnoi u. a., 2024). Für die gekoppelten Läufe (R2B9-R2B9) mussten 793 GB Inputdaten von Levante nach JUWELS kopiert werden. Parameter, wie die Anzahl an Threads oder das NPROMA, mussten an die Auflösung des Experimentes und die Maschine angepasst werden, um eine gute Performance zu erreichen.

Folgender Meilensteine konnten nur bedingt erfüllt werden:

Meilenstein 2 (Monat 12): geeignete Back-Ends aus Aufgabe 2 in AP2 sind in erster Version auf den Systemen der Partner lauffähig

- Die Arbeiten an Arbeitspaket 1 wurden im Laufe des Berichtszeitraums als notwendige Voraussetzung für die Evaluierung der Back-Ends identifiziert. Eine ursprünglich geplante parallele Arbeit an den Aufgaben hätte das Bild verzerren können, falls man keine passenden ICON-Module zur Bewertung genommen hätte.
- Dennoch konnten die möglichen Optionen für geeignete Programmieransätze in ICON-C reduziert werden, da sich nur wenige Kandidaten entsprechend der Anforderungen von ICON weiterentwickelt haben.
- Gridtools/gt4py wird im Projekt EXCLAIM verwendet, dass über Arbeiten innerhalb von ICON-C indirekt mit unterstützt wurde. Eine eigene Gridtool Evaluierung wurde zurückgestellt, da z.B. Fragen nach Performance und Vollständigkeit für ICON zunächst von EXCLAIM beantwortet werden müssen. Eine positive Bewertung dieser Punkte wäre Voraussetzung für einen Ansatz mit verbesserter Usability, falls das Python Frontend sich für die Community als untauglich erweist.
- Kokkos/SYCL wird im Rahmen von WarmWorld-Faster weiter betrachtet. Hierzu ist eine explizite Testumgebung basiert auf einem Mikrophysik Submodule entwickelt worden (siehe Milestone 1).
- Das End-of-life des Backends (Dawn) der ESCAPE2-Toolchain wurde erreicht. Da das embedded C++ DSL Front-end jedoch stark abhängig ist von den speziellen Backend Eigenschaften (Programmiermodell), müsste das Programmiermodell zunächst erweitert bzw. neu designed werden. Dies bedeutet einen erheblichen Aufwand, welche bei geringer Aussicht auf weitere Nutzung nicht gerechtfertigt erschien.
- Der Parflow eDSL Ansatz wurde zusammen mit dem JSC evaluiert; es ergab sich jedoch keine direkte/effiziente Verwendung im Rahmen von ICON.

Die ursprüngliche Arbeitsplanung konnte in den Arbeitspaketen 1 und 3 aufrechterhalten wer-

den. Im Arbeitspaket 2 sollten verschiedene Ansätze zur Hardware-unabhängigen Programmierung evaluiert und miteinander verglichen werden. Hierbei hat sich eine Veränderung der Arbeitsplanung ergeben, da sich verschiedene Annahmen aus der ursprünglichen Planungen änderten. Auf diese wird in den einzelnen Unterpunkten eingegangen.

Anstatt eine Vielzahl von komplexen Abstraktionen zur Performance Portabilität zu betrachten, wurde im ICON-C Kontext an Modularisierungskonzepten (Aufbrechen tiefer Datenstrukturen) und Support (Kommunikation & Daten-Serialisierung) für standalone-Granules gearbeitet. Dies ermöglicht im Folgeprojekt WarmWorld-Faster eine zielgerichtete Evaluierung neuer Ansätze mit geringerem Aufwand.

2.2 Arbeitspakete

2.2.1 Arbeitspaket 1

Aufgabe 1 (Monat 1-9): Blaupause v0 erstellen

Es wurden zwei exemplarische Module aus ICON als eigenständige Arbeitsgrundlagen gewählt: Advektion und Mikrophysik. Hierzu sind Referenzimplementierungen in Fortran vorhanden, welche die Erprobung anderer Programmierparadigma zulassen. Diese Module können demnach eigenständig (von Dritten) weiterentwickelt werden und lassen eine spätere Rückintegration in ICON zu.

Diese konzeptionellen Arbeiten hierzu wurden auch mit ähnlichen Ambitionen im Projekt EXCLAIM der Schweizer Kollegen synchronisiert, welche ebenfalls eine Modularisierung von ICON anstreben. Entsprechende Arbeitsgruppen wurden unter der Zusammenarbeit in ICON-C gebildet und befinden sich im regen Austausch.

Aufgabe 2 (Monat 7-18): Spezifikation von Schnittstellen und Standards

Als wesentliche Punkte zur Neudefinition von sprachunabhängigen Programmieransätzen wurde Arbeiten an folgenden zentralen Infrastrukturkomponenten begonnen: Memory-Management, external interfaces, single I/O layer, single communication layer, testing.

Die hierbei durchzuführenden Arbeiten basieren auf den in Aufgabe 1 definierten exemplarischen Modulen bzw. auf abstrakten/analytischen Tests (Hadley-Test, Bubble-Test), welche eine Beurteilung der Implementierung unabhängig von den wissenschaftlichen Kolleg:innen im Projekt macht. Auf diese Weise wird erreicht, dass auch neue Software-Ingenieure schnell mit dem Code arbeiten können und zukünftige Entwicklungen effizienter umgesetzt werden können.

Im Rahmen von preWarmWorld konnten grundlegende Arbeiten zu folgenden Reorganisationen im ICON Code gestartet werden:

- Modularisierung: Konzept zum Aufbrechen tiefer Datenstrukturen, um Modulabhängigkeiten zu kontrollieren/reduzieren. Exemplarisch umgesetzt für `t_patch` Daten im granule-support sowie in der Advektion.
- tiny-comm: eine vollständige standalone Implementierung der ICON Kommunikation für

Haloaustausch und gather/scatter patterns. Dies ist Voraussetzung für die standalone Ausführung aller Granules mit horizontaler Datenabhängigkeit.

- Memory manager: zur Implementierung von Submodulen in verschiedenen Programmiersprachen und Programmierparadigmen (z.B. CPU/GPU) war es notwendig, die Datenstrukturen zur Speicherung von Modellvariablen explizit durch eine Ebene im userspace zu handhaben. Eine automatische Behandlung durch den Compiler bzw. die runtime Umgebung ermöglicht hierbei nicht die angestrebte Leistungsfähigkeit.

Die definierten Schnittstellen und Standards werden allesamt in WarmWorld-Faster fortgesetzt und im Rahmen von ICON-C in den Hauptentwicklungszweig von ICON zurückgespielt.

Aufgabe 3 (Monat 17-22): Demonstrator für modulares SW-System

Die ursprüngliche Idee, einen expliziten und eigenständigen Prototyp für ICON im Rahmen von WarmWorld zu entwickeln, wurde nach Rücksprache mit den ICON Hauptentwicklern am MPI-M und DWD verworfen. Stattdessen werden die Entwicklungen direkt unter der ICON-C Initiative in das Haupt-ICON Repository einfließen. Dadurch wird sichergestellt, dass die Entwicklungen von der gesamten ICON Entwicklergruppe akzeptiert werden und z.B. die neuen Libraries direkt aktiv verwendet werden können.

Aufgabe 4 (Monat 21-24): blueprint v1 erstellen Die v0 Version des blueprint (ICON SW Strukturanalyse und Möglichkeiten der Designentscheidungen) war durch zwei wesentliche Aspekte geprägt:

- application-specific complex datatypes, welche zwar einfache, kurze Interfaces der einzelnen Submodule ermöglichen, jedoch die Modularisierung erschweren, da jeweils komplexe Infrastrukturen auch für kleine Testfälle benötigt werden,
- generic primitive datatypes, welche zu umfänglichen Interfaces führen, jedoch eine direkte Abbildung auf Standardtypen erlauben. Dies ermöglicht zwar die Modularisierung in kleinere Submodules ohne die Notwendigkeit zusätzlicher Infrastrukturen, jedoch wird der gesamte ICON code dadurch unverhältnismäßig erweitert.

Im Rahmen der fortlaufenden ICON-C Arbeiten haben sich beide Aspekte als zu „extrem“ erwiesen und wurden nicht in das allgemeine Design von ICON übernommen. Stattdessen ist die v1 Version entwickelt worden, welche sich durch die folgenden Kompromisse auszeichnet:

- portables domain-uid argument & complex/primitive getter in der Applikation, wodurch eine minimale Abstraktion der komplexen Gitterstrukturen erreicht wird, jedoch ohne die Datenstrukturen zu überladen,
- reconstructed complex data types in jeder Zielsprache (z.Z. Fortran, C, C++ -> keine getters notwendig), wodurch deutlich schlankere Interfaces entstehen, die jedoch sprachabhängig implementiert werden müssen,
- type-erasure Konzept für geeignete Komponenten innerhalb tiefer Datentypen, so dass die Modulabhängigkeiten verringert werden können.

In WP1 war das JSC dafür verantwortlich, sicherzustellen, dass die modulare Codestruktur

und die Datenabläufe des ICON-Modells mit dem Konzept der Modularen Supercomputing-Architektur (MSA) vereinbar sind. ICON RC Version 2.6.6 wurde verwendet, um MSA-basierte Simulationen auf dem JUWELS-Cluster und Booster am JSC zu entwickeln und zu testen. ICON wurde mit Unterstützung des MPI-M sowohl für CPUs als auch für GPUs kompiliert, ohne eine zusätzliche Granularität zu schaffen, bereiteten die Daten für die Tests vor und testeten die Codes durch die Durchführung von Simulationen. Die Ergebnisse der Simulationsleistung, einschließlich der arithmetischen Intensität, der Speicherzugriffsmuster und der Skalierung von ICON bei Verwendung des MSA, werden in Bishnoi u. a., 2024 diskutiert.

2.2.2 Arbeitspaket 2

Aufgabe 1 (Monat 1-2): Kriterien für die Evaluierung verschiedener Programmierparadigmen erstellen

Ein Satz von Bewertungskriterien für die zu untersuchenden Programmieransätze wurde auf Basis von vier zentralen Blöcken entwickelt:

- Portabilität (wieviele verschiedene HPC-Systeme können adressiert werden? Wie umfassend ist der Programmieransatz bzw. können alle notwendigen Methoden aus ICON damit abgedeckt werden?)
- Performance (Welche Leistung kann auf welchen HPC-Systemen erreicht werden? Wie verhält sich diese Leistung zu einer nativen Programmierung für das jeweilige System?)
- Nutzbarkeit/Wartbarkeit (Ist der Code im entsprechenden Programmieransatz für Klimawissenschaftler lesbar? Wie hoch ist der Aufwand bei der Programmierung neuer Elemente? Müssen spezielle Infrastrukturen (preprocessor, compiler, etc.) bestehen, um den Programmieransatz anwenden zu können?)
- Ökosystem hinter dem Ansatz (Wie groß ist die Nutzergruppe des Programmieransatzes? Sind kommerzielle Interessen vertreten oder handelt es sich um akademische Projekte? Ist man auf Hilfe Dritter angewiesen, falls der Ansatz scheitert?)

Aufgabe 2 (Monat 1-20): Evaluierung verschiedener Programmierparadigmen

Diese Aufgabe konnte nur teilweise erfüllt werden, da manche Ansätze verworfen wurden und für andere Ansätze weitergehende Vorbereitungen notwendig waren. Grund dafür sind die länger als geplant andauernden Aufgaben in Arbeitspaket 1, da hier eine enge Abstimmung mit den anderen Projekten im Rahmen von ICON-C erfolgen mussten. Von den Vorarbeiten profitiert das WarmWorld-Faster Projekt, in dem jetzt verschiedene Programmierparadigmen implementiert werden.

Der Stand zu den einzelnen Teilschritten zum Ende von preWarmWorld ist wie folgt:

- GridTools: Die Entwickler dieses Frameworks haben den Kolleg:innen im WarmWorld-Projekt erst Anfang 2023 Zugang zu entsprechenden Ressourcen geben können, da die Entwicklung auf deren Seite noch nicht ausgereift genug war. Dementsprechend konnte im preWarmWorld Projekt noch keine Evaluierung an realem Code stattfinden.

- AnyDSL: Eine eingehende Evaluierung des Ansatzes wurde aufgrund mangelnden Umfangs für den ICON-Code und zu geringer Nutzergemeinde des Werkzeuges verworfen. Es ist nicht abzusehen, dass dieser Ansatz während der Laufzeit von WarmWorld (bzw. über einen noch längeren Zeitraum) eine nutz- und wartbare Codebasis für ICON liefern kann.
- eDSL: JSC hat beigetragen zur Evaluierung des Ansatzes der eingebetteten domänenspezifischen Sprache (eDSL) für die ICON-Codebasis bei. Diese Arbeit basierte auf früheren Erfahrungen mit der Anwendung des eDSL-Ansatzes auf das hydrologische Modell ParFlow in Jülich. Die wichtigsten Ergebnisse und die damit verbundenen Arbeiten sind in einem Papier zusammengefasst, das bei Geoscientific Model Development eingereicht wurde (Piotrowski u. a., 2023) und in einem öffentlich zugänglichen Projektbericht (Lu u. a., 2024).
- Kokkos: Da ICON im Wesentlichen in Fortran geschrieben ist, muss für die Evaluierung von Kokkos zunächst eine repräsentative Teilmenge der Module in C/C++ umgeschrieben werden. Diese Arbeit konnte erst nach Abschluss von Aufgabe 1 in Arbeitspaket 1 starten, wodurch ein Start erst in 2023 erfolgte.

Bereits im Zwischenbericht 2022 wurde die eingehende Evaluierung von AnyDSL verworfen. Die Analyse von GridTools (bzw. gt4py als Nachfolger) wird fortlaufend in Zusammenarbeit mit dem EXCLAIM Projekt vorgenommen. Aufgrund der unerwartet hohen Komplexität – insbesondere falls größere Teile von ICON, wie der dynamische Kern, untersucht werden sollen – war es dabei notwendig die Ressourcen aus dem Folgeprojekt WarmWorld-Faster abzuwarten. Im Rahmen von preWarmWorld konnten jedoch wesentliche Vorarbeiten geleistet werden (z.B. Bereitstellung von modularen Tests). Ebenso wurde im Rahmen von ICON-C entschieden, dass sich am DKRZ eher auf die Evaluierung von C++ basierten Ansätzen (wie Kokkos oder SYCL) konzentriert wird.

Die ursprüngliche Verallgemeinerung der Ansätze durch Einführung eines Front-Ends auf Basis der im ESCAPE-2 Projekt geleisteten Vorarbeiten wurde verworfen. Ähnliche Arbeiten an einer Toolchain im Rahmen von gt4py haben gezeigt, dass dadurch ein erheblicher zusätzlicher Aufwand betrieben werden muss. Dieser war aufgrund der knappen Ressourcen in preWarmWorld nicht verhältnismäßig.

Die Parflow eDSL wurde insb. vom Projektpartner JSC evaluiert. Die Ergebnisse sind publiziert (<https://eartharxiv.org/repository/view/6767/>) und wurden den ICON Entwicklern präsentiert. Das JSC evaluierte auch die Möglichkeit, den eDSL-Ansatz für die ICON-Codebasis zu verwenden, indem es zwei Kernel überprüfte und potenzielle Vorteile und Grenzen der Anwendung des eDSL-Ansatzes auf ICON feststellte (Lu u. a., 2024). Eine Ausweitung dieser Arbeit erfordert jedoch einen größeren personellen Aufwand, als hier verfügbar war. Die Anpassung des ICON-Codes an den eDSL-Ansatz erfordert eine klare und konkrete Software-Strategie mit einer gemeinsamen Entwicklungsarbeit zwischen den ICON-Hauptentwicklern und HPC/GPU-Experten. Zusammenfassend lässt sich sagen, dass der Ansatz – ähnlich wie bei Gridtools/gt4py – noch eine starke Limitierung auf reguläre Gitter bzw. einfache Schleifenkonstrukte enthält, welche den Einsatz für ICON Datenstrukturen ineffizient macht. Die wichtigsten Ergebnisse und die damit verbundenen Arbeiten sind in einem Papier zusammengefasst,

das bei Geoscientific Model Development eingereicht wurde (Piotrowski u. a., 2023) und in einem öffentlich zugänglichen Projektbericht (Lu u. a., 2024).

ParFlow ist in C geschrieben, während das ICON-Modell in Fortran geschrieben ist. Daher wurde zunächst ein Granulat des Eulag-Modells (Ziemiański et al., 2021) verwendet, um das Konzept des eDSL-Ansatzes für einen numerischen Wettervorhersage- und Klimacode zu testen. Der eDSL-Ansatz wurde erfolgreich in einem Granulat des Eulag-Modells implementiert, wobei gezeigt wurde, wie der eDSL-Ansatz als Abstraktionsschicht für GPU-Offloading in dessen Advektionscode implementiert werden kann. Über die Leistung und Entwicklung des eDSL-Ansatzes wird in Piotrowski u. a., 2023 berichtet.

Am Ende des preWarmWorld-Projekts wurden zwei granulare Modelle von ICON erstellt, eines für das Advektionsschema und das andere für das Wolkenmikrophysikschema. Das Advektionsschema verfügte zum gegebenen Zeitpunkt nicht über GPU-Unterstützung und konnte daher nicht verwendet werden, um das Konzept von eDSL weiter zu testen. Die Evaluierung des Wolkenmikrophysik-Schemas zeigte, dass der eDSL-Ansatz für den Fortran-Code nicht einfach implementiert werden kann, solange sich Operatoren im Schleifenkörper befinden. Es wurde geschätzt, dass der Aufwand für die Implementierung von eDSL selbst für das ausgewählte Code-Granulat die Ressourcen, die in diesem Projekt für diese Aufgabe bereitgestellt werden, bei weitem übersteigen würde. Darüber hinaus gibt es zusätzliche Herausforderungen bei der Anwendung des eDSL-Ansatzes im Zusammenhang mit der Unterstützung unterschiedlicher Hardware-Architekturen. Mit fortschrittlicheren Konzepten wie Kokkos können Entwickler die Schwierigkeiten bei der Implementierung von Backends von Codes besser abmildern, wenn sie in der Sprache C/C++ geschrieben sind.

Dennoch wird der eDSL-Ansatz nach wie vor als nützlich angesehen, um in bestimmten Fällen die Auslagerung auf GPUs zu ermöglichen. Der Projektbericht, der die Herausforderungen bei der Implementierung der eDSL für ICON zusammenfasst (Lu u. a., 2024), weist darauf hin, dass die eDSL bereits für andere geowissenschaftliche Codes eine portable Leistung auf GPUs ermöglicht. Außerdem kann die portable Leistung des eDSL-Ansatzes die Herausforderungen der Portierung des Codes auf neue Hardware überwinden, solange die Programmiersprache von den verfügbaren Compilern unterstützt wird. Obwohl die Implementierung der eDSL in ICON derzeit nicht favorisiert wird, scheint sie dennoch machbar zu sein. Sie würde eine sorgfältige Planung und eine Strategie zur Bewältigung der Komplexität der sehr großen Codebasis erfordern.

Aufgabe 3 (Monat 15-24): Auswertung und Dokumentation der Evaluierung

Während einige Programmieransätze gemäß Kriterien aus Aufgabe 1 bereits als unpassend für ICON identifiziert werden konnten, muss die Evaluierung bei anderen stetig fortgeführt werden.

Der ursprüngliche Ansatz war die Tracer Advection aus dem ICON Modell herauszulösen und diesen Code dann als Standalone Testumgebung für die verschiedenen Portabilitätsansätze zu verwenden. Leider hat sich das aber als komplexer als erwartet herausgestellt, sodass wir in preWarmWorld stattdessen in Zusammenarbeit von DKRZ und MPI-M eine eigenständige Implementierung des Mikrophysik-Schemas von ICON erstellt und über ein gitlab repo verteilt haben. In diesem Umfeld können nun verschiedene Ansätze – auch über Programmiersprachen und Programmierparadigmen hinweg – weiter analysiert werden. Dies hat aber zu Verzögerun-

gen geführt und die eigentliche Evaluierung wird nun in WarmWorld-Faster stattfinden. Diese prototyp Implementierung der Mikrophysik dient auch als Basis für die Zusammenarbeit mit anderen Instituten.

Aktuell umfasst die Testumgebung Programmieransätze zur Nutzung von GPUs auf Basis von Fortran+OpenACC, C++ native, C++/OpenMP, SYCL, Kokkos. Geplant sind Erweiterungen für Fortran+OpenMP. Diese Auswahl wird sowohl in Hinblick auf Leistungsfähigkeit als auch Nutzbarkeit für wissenschaftliche Programmierer eine umfassende Auswertung erlauben, auf Basis deren dann neue SW-Konzepte für den ICON Code beschlossen werden können.

Das JSC trug zur Bewertung des eDSL-Ansatzes bei, der von den Modellen ParFlow (in C geschrieben) und Eulag (in Fortran geschrieben) verwendet wurde (Piotrowski u. a., 2023). Diese Analyse ergab eine gute Leistung bei der Verwendung des eDSL-Ansatzes für alte Codes, um die Qualität und den Aufwand für diese Aufgabe zu demonstrieren. Das JSC evaluierte auch die Möglichkeit, den eDSL-Ansatz für die ICON-Codebasis zu verwenden, indem es zwei Kernel überprüfte und potenzielle Vorteile und Grenzen der Anwendung des eDSL-Ansatzes auf ICON feststellte (Lu u. a., 2024). Eine Ausweitung dieser Arbeit erfordert jedoch einen größeren persönlichen Aufwand, als hier verfügbar war. Die Anpassung des ICON-Codes an den eDSL-Ansatz erfordert eine klare und konkrete Software-Strategie mit einer gemeinsamen Entwicklungsarbeit zwischen den ICON-Hauptentwicklern und HPC/GPU-Experten.

Aufgabe 4 (Monat 13-24): MSA (modular supercomputing architecture) Prototyp

Der Prototyp wurde am JSC mit Unterstützung des MPI-M und DKRZ erstellt und durch Meilenstein 3 (gekoppeltes ICON läuft auf MSA am JSC) abgeschlossen. Das JSC war für die Prüfung des gekoppelten ICON-Modells auf JUWELS im Hinblick auf die modulare Supercomputing-Architektur (MSA) verantwortlich. Die Arbeit wurde erfolgreich abgeschlossen und ist in Bishnoi u. a., 2024 beschrieben. MSA-Läufe auf JUWELS können mit dem bestehenden Slurm-Scheduler-Setup durchgeführt werden, um gekoppelte Modellsimulationen auszuführen, bei denen die Atmosphäre auf GPUs und der Ozean und die Datei-I/O auf CPUs ausgeführt werden. Wir haben gezeigt, dass gekoppelte Simulationen mit der aktuellen Architektur und der ICON-Code-Struktur durchgeführt werden können, ohne dass eine zusätzliche mittlere Code-Schicht entwickelt oder zusätzliche Änderungen am aktuellen Code vorgenommen werden müssen.

Das ICON-Modell wurde portiert und optimiert, um auf dem JUWELS-MSA-System zu laufen, das aus JUWELS-Cluster und JUWELS-Booster mit CPU- bzw. GPU-Knoten besteht. Die Atmosphären-Komponente läuft auf den GPU-Knoten, während die Ozean-Komponente auf den CPU-Knoten läuft. Die Testsimulationen basierten auf einer horizontalen Gitterauflösung von 5 km, was der konvektiven Auflösung nahe kommt. ICON wurde mit asynchroner E/A betrieben, um 0,5 bis 3 stündliche Ausgaben von 14 dreidimensionalen atmosphärischen Variablen und 125 zweidimensionalen Variablen zu erzeugen. 17 CPU-Knoten wurden für I/O verwendet, um während des Laufs insgesamt 2,2 TB an Daten zu erzeugen. Weitere Einzelheiten sind in Bishnoi u. a., 2024 zu finden.

Ein wesentlicher Schritt, um den MSA-Ansatz zu ermöglichen und von ihm zu profitieren, bestand darin, den Sweet Spot der möglichen Kombinationen von CPU- und GPU-Knoten für die Ozean- bzw. Atmosphären-Komponenten zu finden. Bei einem Ungleichgewicht würde eine Komponente laufen, während die andere im Leerlauf wäre, wodurch Rechenressourcen ver-

schwendet würden. Daher kann die Ermittlung eines „Sweet Spot“ für die Aufteilung zwischen CPU- und GPU-Knoten die Leerlaufzeit verringern und die Berechnungseffizienz erheblich steigern. Das Ergebnis zeigt, dass durch die Verwendung von 84 GPU-Knoten und 63 CPU-Knoten für das gegebene ICON-Setup die Gesamtlaufzeit minimiert werden konnte.

In einem zweiten Experiment haben wir reine CPU-Simulationen mit dem MSA-Setup aus dem vorherigen Experiment verglichen. Bei der reinen CPU-Simulation wurden bis zu 860 CPU-Knoten verwendet, jetzt einschließlich Atmosphäre, Ozean und E/A-Knoten. Während die Nicht-MSA- und MSA-Setups ungefähr die gleiche Laufzeit haben, kann das MSA-Setup den Energieverbrauch auf 45 % reduzieren (basierend auf Schätzungen der thermischen Designleistung), was dem dringenden Bedarf an Energieeinsparungen bei groß angelegten Simulationen entspricht, wie in Schulthess u. a., 2018 angesprochen.

2.2.3 Arbeitspaket 3

Aufgabe 1 (Monat 1-3): GIT repo erstellen

In enger Abstimmung mit der Umstrukturierung zu ICON-C wurde ein entsprechendes GIT Repository erstellt und den Projektmitarbeitenden offen gemacht. Die Entwicklungen aus diesem preWarmWorld Zweig (und später dann auch aus WarmWorld-Faster) werden zunächst in Form von kleinen „sandbox“ Beispielen entwickelt und dann in den Hauptzweig der ICON-C Entwicklung gemerged. Auf diese Weise sollen disruptive Eingriffe in den Code von der wissenschaftlichen Weiterentwicklung getrennt werden.

Aufgabe 2 (Monat 4-15): Test-Infrastruktur bereitstellen

Die bestehende ICON Test-Infrastruktur (gitlab.dkrz.de und buildbot) wurde auf den Entwicklungsbereich von preWarmWorld erweitert. Neben den HPC-Ressourcen am DKRZ können Codeentwicklungen so auch auf z.B. JSC und CSCS-Systemen getestet werden. Vom MPI-M definierte Tests wurden durch sogenannte code-coverage analysen Erweitert, um aussagen über die Codeabdeckung der Tests treffen zu können.

Aufgabe 3 (Monat 3-14): Management und Dokumentationswerkzeuge bereitstellen

Die am DKRZ gehosteten Tools des gitlab.dkrz.de Servers wurden für die Entwicklung und Planung von ICON-C der gesamten ICON-Community offen gemacht. Als wesentliche Werkzeuge der strategischen Weiterentwicklung von ICON dienen hierbei Issue-Boards und der Austausch von Meeting-Ergebnissen. Die Dokumentation von konkreten Implementierungen zur Neustrukturierung in ICON erfolgt in einem mehrstufigen Prozess: Darstellung der Projektidee gegenüber der ICON-Koordinationsgruppe, Offenlegung der geplanten Abläufe für alle ICON Entwickler und Bitte um Rückmeldung/Kritik, Implementierung erster Prototypen, Evaluierung der Ergebnisse, ggf. Verbesserung der Projektidee, finaler merge in den Hauptentwicklungszweig von ICON-C.

Aufgabe 4 (Monat 13-24): Tutorials und Nutzerunterstützung

Die Arbeiten basierend auf gitlab, buildbot und einem Fortran src-code analyser (siehe 1) wurden fortlaufend mit der ICON Community auf technical ICON-C Meetings geteilt. Es wurde ne-

ben der direkten Unterstützung insb. die Nutzung des buildbot und gitlab-CI zur verbesserten ICON Codeentwicklung präsentiert und im internen ICON-Wiki dokumentiert.

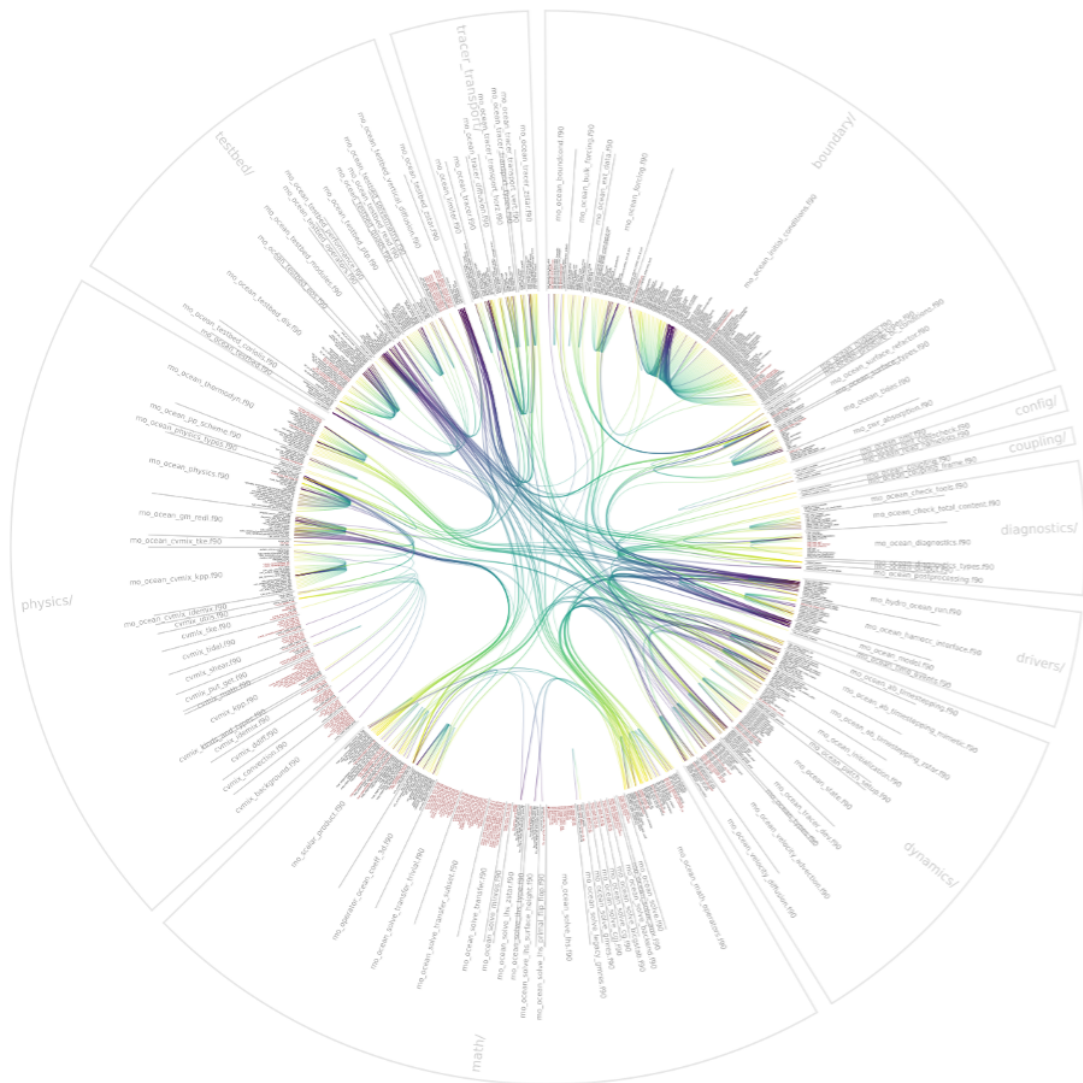


Abbildung 1: Statische Analyse der Abhängigkeiten im ICON Code zur Identifikation potenzieller Schnittstellen und Modularisierung

Aufgabe 5 (Monat 16-22): Teststrategie auf großen Systemen

Die Test-Infrastruktur von ICON konnte erfolgreich um wichtige Systeme erweitert werden: Als erstes System mit AMD-GPUs wurde LUMI integriert, was die Stabilität des Entwicklungs- und Produktions-Betriebs entscheidend verbessert hat. Die zukünftige Schweizer HPC-Infrastruktur ALPS ist mit einem Test-System ebenfalls angebunden und wird das der-zeitige Produktivsystem DAINTE bis Mitte 2024 vollständig ersetzen. Das aktuelle MSA-System des JSC konnte ebenfalls als automatische Testplattform für ICON etabliert werden, was beispielsweise die Ergebnisse und Erfahrungen aus Arbeitspaket 3 diesbezüglich verstetigt. Insbesondere stellt dies eine Vorbereitung für die Integration des kommenden Exascale System JUPITER des JSC dar.

Als Erweiterung der Testinfrastruktur von ICON wurde die "continuous integration" der gitlab service platform so erweitert, dass direkter Zugriff auf die HPC-Ressourcen vom DKRZ möglich ist.

Einerseits erleichtert dies den Testvorgang aus Sicht der ICON-Entwickler. Andererseits ermöglicht es die Zusammenarbeit mit anderen HPC-Betreibern, die ebenfalls auf diese Technologie umsteigen wollen. Im Gegensatz zur etablierten “buildbot“ Plattform ermöglicht sie i.d.R. kürzer Testzyklen, was laufenden und zukünftigen Restrukturierungen am ICON Code im Rahmen von WarmWorld-Faster zugutekommt.

Aufgabe 6 (Monat 1-12): Evaluate licence options and governance models to provide technologies, SW components and ideally full model releases as open source

Für ICON beschlossen die Entwicklungspartner von ICON, unter der Führung vom MPI-M, die BSD-3-Lizenz zu verwenden, um die Open-Source-Version von ICON der Öffentlichkeit zugänglich zu machen. Die Open-Source-Version hat sich bereits als vorteilhaft für neue Projekte erwiesen, z.B. für die BMBF-Projekte SCALEXA oder WarmWorld, und so können Mitarbeiter von externen Gruppen an der Optimierung der Leistung von ICON arbeiten, ohne um Erlaubnis für die Nutzung des Codes fragen zu müssen. Diese Wahl der Lizenz wurde auch von JSC favorisiert.

3 Weitere Entwicklungen

Wie ursprünglich geplant, können die Erkenntnisse und Implementierungen im Rahmen von preWarmWorld nahtlos in das Projekt WarmWorld-Faster übernommen werden. Beide Projekte arbeiten eng zusammen und teilen sich dieselbe Entwicklungsplattform.

Eine direkte Anschlussfähigkeit ist durch das im November 2022 gestartete Projekt WarmWorld-Faster eingetreten. Der vorbereitende Charakter des Projektes preWarmWorld hat dabei die Startphase deutlich beschleunigt – mit Ausnahme der Tatsache, dass weitere wissenschaftliche Programmierer:innen nur schwer auf dem Markt zu finden sind.

Eine unmittelbare wirtschaftliche Anschlussfähigkeit ist aufgrund des Charakters der Grundlagenforschung bzw. Entwicklung von OpenSource Software nicht abzusehen.

Wie im Zwischenbericht für 2022 bereits dargelegt, konnte der gt4py Ansatz nicht vollumfänglich getestet werden, da notwendige Implementierungen für ICON noch nicht vorlagen und auch das backend noch zu wenige Systeme (insb. keine AMD-GPUs) unterstützt. Dies untermauerte die Entscheidung sich auf Ansätze einer breiteren Community zu konzentrieren und zunächst C++ basierte Lösungen (Kokkos, SYCL) zu evaluieren.

Wie ursprünglich geplant wurden die Erkenntnisse und Implementierungen im Rahmen von preWarmWorld direkt im Projekt WarmWorld-Faster fortgeführt. Die entwickelten Blaupausen für modulares Design (hierbei insb. microphysics) wurden erfolgreich im Rahmen einer Masterthesis und der ISC Student cluster competition eingesetzt, was die Nutzbarkeit des Ansatzes außerhalb von ICON belegt.

Eine direkte Anschlussfähigkeit ist durch das im November 2022 gestartete Projekt WarmWorld-Faster eingetreten. Der vorbereitende Charakter des Projektes preWarmWorld hat dabei die Startphase deutlich vereinfacht. Eine unmittelbare wirtschaftliche Anschlussfähigkeit ist aufgrund des Charakters der Grundlagenforschung bzw. Entwicklung von OpenSource Software

nicht abzusehen.

4 Auswirkungen und Verbindungen zu anderen Projekten/Programmen

Im Rahmen der Beantragung eines Projektes für den BMBF-call „SCALEXA“ wurde deutlich, dass der Programmieransatz „AnyDSL“ nicht für die Bedürfnisse des ICON Modells passt bzw. ein sehr hoher Aufwand betrieben werden muss, um diesen vollumfänglich nutzbar zu machen. Es wurde daher entschieden, diesen Ansatz nicht weiter in Aufgabe 2b des Arbeitspaket 2 zu evaluieren.

In verschiedenen Präsentationen der Entwickler von GridTools (nunmehr gt4py genannt) wurde erwähnt, dass die Implementierung noch nicht robust genug sei, um gute Performance für wesentliche stencil-Operation in ICON zu erreichen. Ebenso sind Tests nur für alte Plattformen von Nvidia GPUs (P100 und V100) durchgeführt worden, während aktuelle ICON-Versionen schon die aktuelle Generation der Hardware (A100 bzw. sogar H100) unterstützt. Auch AMD-GPUs werden noch nicht unterstützt. Damit ist eine umfassende Testung dieses Ansatzes nicht möglich gewesen.

4.1 Verbundübergreifende Tätigkeiten, wie Meetings, Austausch, Workshops, etc.

- Die Mitglieder des preWarmWorld Projektes haben durchweg an allen ICON-C Treffen teilgenommen.
- Seit Ende 2022 sind die preWarmWorld Verbundtreffen übergegangen in Treffen des gesamten WarmWorld-Faster Verbundes.
- „Librarysation-workshop“ 2022 in Karlsruhe: Teilnehmer des ICON-Konsortiums, des WarmWorld und EXCLAIM Projektes trafen sich zu Austausch, Design und erster Implementierung von Modularisierungsansätzen in ICON-C; auf diese Weise wurde eine Blaupause für das SW-Design weiter verfeinert.
- Die Arbeiten in preWarmWorld sind nahtlos in den WarmWorld-Faster Verbund übergegangen – insbesondere beteiligen sich die SW-Entwickler an den bi-weekly Meetings zur Besprechung eines angepassten Designs des ICON Codes. Unter dem Titel ICON-C (ICON consolidated) wurden in 2023 regelmäßige Treffen abgehalten, welche den aktuellen Stand des SW-engineering beleuchteten und fortlaufend eine Roadmap aktualisierten: 10. März 2023 online; 08.-10. Mai 2023 Hamburg; 21.-22. November 2023 Karlsruhe.

5 Veröffentlichungen

1. A. Bishnoi u. a. (2024). “Earth system modeling on modular supercomputing architecture: coupled atmosphere–ocean simulations with ICON 2.6.6-rc”. In: *Geoscientific Model Development* 17.1, S. 261–273. DOI: [10.5194/gmd-17-261-2024](https://doi.org/10.5194/gmd-17-261-2024)
2. ICON partnership (DWD and MPI-M and DKRZ and KIT and C2SM) (2024). *ICON release 2024.01*. DOI: [10.35089/WDCC/IconRelease01](https://doi.org/10.35089/WDCC/IconRelease01)
3. Y.-S. Lu u. a. (2024). *Challenges of applying an embedded domain specific language for performance portability to Earth system models*

6 Bibliografie

- Bishnoi, A., O. Stein, C. I. Meyer, R. Redler, N. Eicker, H. Haak, L. Hoffmann, D. Klocke, L. Kornblueh und E. Suarez (2024). “Earth system modeling on modular supercomputing architecture: coupled atmosphere–ocean simulations with ICON 2.6.6-rc”. In: *Geoscientific Model Development* 17.1, S. 261–273. DOI: [10.5194/gmd-17-261-2024](https://doi.org/10.5194/gmd-17-261-2024).
- ICON partnership (DWD and MPI-M and DKRZ and KIT and C2SM) (2024). *ICON release 2024.01*. DOI: [10.35089/WDCC/IconRelease01](https://doi.org/10.35089/WDCC/IconRelease01).
- Lu, Y.-S., D. Caviedes-Voullième, O. Stein und L. Hoffmann (2024). *Challenges of applying an embedded domain specific language for performance portability to Earth system models*.
- Piotrowski, Z. P., J. Hokkanen, D. Caviedes-Voullième, O. Stein und S. Kollet (2023). “Parflow 3.9: development of lightweight embedded DSLs for geoscientific models”. In: *EGUsphere* 2023, S. 1–21.
- Schulthess, T. C., P. Bauer, N. Wedi, O. Fuhrer, T. Hoefler und C. Schär (2018). “Reflecting on the goal and baseline for exascale computing: a roadmap based on weather and climate simulations”. In: *Computing in Science & Engineering* 21.1, S. 30–41.
- Suarez, E., N. Eicker und T. Lippert (2019). “Modular Supercomputing Architecture: from Idea to Production; 3rd”. In: *Contemporary High Performance Computing: From Petascale toward Exascale, Volume 3*. Bd. 3. FL, USA: CRC Press, S. 223–251. ISBN: 9781138487079.