

Interview Luis Kornblüh/ICON

Seminar Softwareentwicklung in der Wissenschaft

Fin Töter

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

27. August 2018



informatik
die zukunft

Gliederung (Agenda)

1 Das MPI

2 ICON

3 Softwareentwicklung beim MPI

4 Kollaboration

5 Blick in die Zukunft

Geschichte

- 1974 gegründet zur Erforschung der Fragestellungen der langfristigen Klimaerwärmungen. [4]
- Zum Zeitpunkt der Gründung zwei Arbeitsbereiche
 - Atmosphäre im Erdsystem
 - Ozean im Erdsystem
- 1986 neue Abteilung: Land im Erdsystem [4]

SCLAB

Gruppe zur technischen Unterstützung der Hauptgruppen

Aufgeteilt in :

- CIS - Central IT Services
- CIMD - Computational Infrastructure and Model Development
- SIP - Strategic IT Partnerships

ICON-ESM

- Numerisches Wettervorhersagemodell
- Zusammen entwickelt von DWD und MPI
- Zwei Teile: ICON-A und ICON-O bilden zusammen ICON-ESM
- Multiphysikmodell

ICON Auflösung

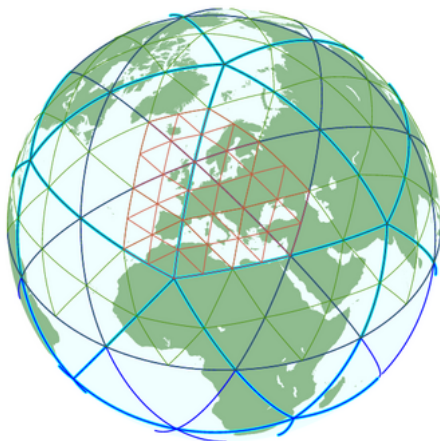


Abbildung: Example for a regionally refined grid. [1]

ICON Auflösung

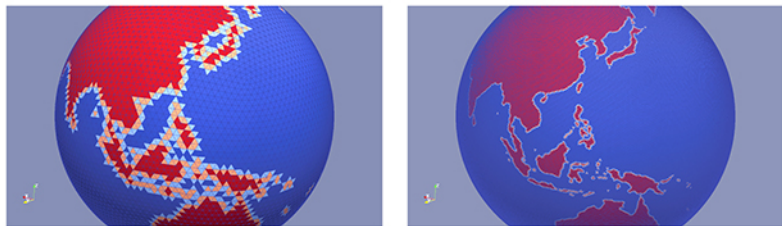


Abbildung: Typical grid configuration for the Indonesian region in the 160 km ICON-A (left) and the 40 km ICON-O (right) that are coupled to form ICON-ESM.[3]

ICON

- Validation und Projektion:
 - 1. Versuchen nachzurechnen und mit dem Modell zu bestätigen
- Validierung des physikalischen Verständnis →
 - 2. Grundvertrauen für Hypothesen in der Zukunft
- Experimente die in der echten Welt nicht möglich: z.B. Eiswelt, Solareinstrahlung wegnehmen etc.
- Sprachen:
 - mehrere 100k Zeilen C
 - mehrere 100k Zeilen Fortran
 - ein wenig Assembler ;-)

YAC-Coupler

YAC → Yet Another Coupler

Verbindet ICON-A und ICON-O (Skaliert die Werte)

Am DKRZ entwickelt

Zuständig für den Austausch von:

- Energie
- Bewegung
- Wasser
- Wichtige Gase

ICON

ICON - ESM

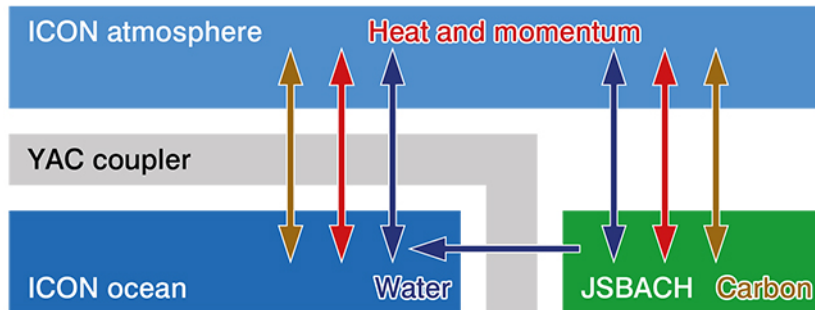


Abbildung: ICON Funktionsweise[2]

Luis Kornblüh

- Studium an der TH Darmstadt → Fokus auf Optimierung
- Arbeit am GKSS heute Forschungszentrum Geesthacht → Modellentwicklung
- Satellitenentwicklung und Entwicklung auf Fortran Standard
- Ab 1992 bis 1997 Promotion an der Universität Hamburg mit Fokus auf Modellentwicklung
- Ab März 1997 CIMD Gruppe am MPI → Entwicklung von ICON
- Begeisterung am Job → dauernd neue Probleme 90% interessant und 10% ätzend

Verwendete Software

- Numerische Modelle
- Datenformate: netcdf, hdf5, crypt2
- Datenbanken: PostgreSQL
- Tools zum verteilten Zugriff auf Datenbanken
- MPI
- kryptographische Tools
- Compiler
- Shells etc.

Verwendete Sprachen

- C
- Fortran
- C++
- Python
- Bash
- Assembler

Verwendete Hardware

- 99,98% Verwendung Mistral
- 0,02% Verwendung Bürorechner
- 100% Verwendung **CPU** keine GPUs → Mistral pure CPU Maschine

Verwendete Hardware

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,282,544	122,300.0	187,659.3	8,806
2	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
3	Sierra - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/NNSA/LLNL United States	1,572,480	71,610.0	119,193.6	
4	Tianhe-2A - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 , NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
5	AI Bridging Cloud Infrastructure (ABCI) - PRIMERGY CX2550 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR , Fujitsu National Institute of Advanced Industrial Science and Technology (AIST) Japan	391,680	19,880.0	32,576.6	1,649

Abbildung: Top 5 Supercomputers[5]

Was ist SE?

- Planung
- Kommunikation
- Validation
- **Programmieren nicht Teil von SE**

Klassisch vs. Wissenschaftlich

- Assembler für konstante Geschwindigkeit und nicht für schnelleren Code
- Subroutinen für Strukturierung und nicht für Wiederverwertung
- Software-Life-Cycle: 30 Jahre eher weiterentwickeln statt neu schreiben

Qualitätsmanagament

- ISO's (z.B. ISO/IEC 9126) egal, werden zwar vermutlich erfüllt aber testet niemand
- Programmierkonventionen sind festgelegt
- Strikte Regeln in welchen Teilen des Codes was verwendet werden darf. z.B. im Quality-of-Life-Teil der Software geht alles unter $O(n^4)$. Im Kern dann sehr viel strikter.

Testing

- Mehrere Stufen des Testings
- CI Testings
- Build-bot tests
- Für die inhaltlichen Tests monatelang Modell laufen lassen
- Bei jedem Release:
 - AMIP (Athmosphere Model Intercomparison Project) Lauf → 2-3 Tage
 - OMIP (Ocean Model Intercomparison Project) Lauf → 2 Tage
 - CMIP (Coupled Model Intercomparison Project) Lauf → 6 Wochen

Benchmarking

- Verwendung der SCT Library (hausintern)
- Ermöglicht Einbau von Timern in Code sowie Export in HDF5 Dateien
- Anzeige möglich durch hauseigenes Frontend (<http://perf.mpimet.mpg.de/>)

Releases

- Zweimal jährlich volles Release.
- Jede Gruppe kann bei sich selber noch kleiner Sub-releases machen

Datenmanagement

- Code:
 - GIT (Development und Verteilung)
- Daten:
 - Eingabedaten → 100GB
 - Ausgabe → 120PB das ist das 1,2 **Millionfache**
 - In Realität auf ca. 1 PB "echte" Daten zurückgeschrumpft
 - Daten werden ausgewertet und dann ins DKRZ-Archiv,
 - Danach theoretisch Löschung

Bottleneck

Frage an Euch

Was glaubt ihr was nach Luis Ansicht das größte Bottleneck bei der Entwicklung ist?

Bottleneck

Luis Antwort

Manpower

Gruppenintern

- KANBAN
- Tür zu Tür
- Telefonate
- Ab und an ein größeres Meeting

DWD

- 2 mal jährliches Entwicklertreffen in Offenbach
- Regelmäßige Videokonferenzen
- Telefonate
- Slack
- Sprint Sessions

Konferenzen

- Kompression von Daten
- Vorteile von Formaten
- Vorteile von Compilern
- Gedanken über I/O Subsysteme
- Parallelität in Programmen erklären
- Optimierungen auf verschiedenen Ebenen

IT-Unterstützung-innerhalb

- Bearbeitet von CIS
- Soweit relativ gut
- Nachinstallieren von Sachen sehr dröge
- Daher nach einiger Zeit ein 500 Zeilen Skript gebaut, welches die Sachen im \$Home installiert.

IT-Unterstützung-DKRZ

- Sehr persönliches Verhältnis
- Im Großen und Ganzen sehr guter Service
- I/O Operationen auf der Mistral so einfach wie “mkdir“ oder “cp “
- Selbe Probleme wie IT innerhalb
 - Langsame Installation → Skript benutzen

Zukunft Modellentwicklung

- Cloud?
 - Nicht preislich kompetitiv bei den Datenmengen
- Neuerungen eher in der eigenen Entwicklung
- Skalierung von KI aktuell noch nicht sonderlich gut.
- Großes Gap zwischen dem was sich Informatiker ausdenken und was sich Wissenschaftler benötigen
- Mehr Rechenpower wäre nett
- Nicht lineare Modelle sind empfindlich auf Bedingungen → schwere Validierung

Zusammenfassung

- ICON ist zweiteiliges Klimamodell
- Programmierung ist kein Teil von Softwareentwicklung
- Assembler für konstante Zeit nicht schnellere Zeit
- Bottleneck Manpower
- Kommunikation ist das wichtigste
- Zukunft liegt eher im Selbstgebauten

Danke für eure Aufmerksamkeit! 😊

Quellen I

- [1] Max Planck Institut für Meteorologie. *ICON Development*. [Online; accessed June 27, 2018]. ? URL: <https://www.mpimet.mpg.de/en/communication/news/focus-on-overview/icon-development/>.
- [2] Max Planck Institut für Meteorologie. *ICON Development*. [Online; accessed June 27, 2018]. ? URL: <https://www.mpimet.mpg.de/en/communication/news/focus-on-overview/icon-development/>.
- [3] Max Planck Institut für Meteorologie. *ICON Earth System Model*. [Online; accessed June 27, 2018]. ? URL: <https://www.mpimet.mpg.de/en/communication/news/focus-on-overview/icon-earth-system-model/>.

Quellen II

- [4] MPG. *Institutsgeschichte Meteorologie*.
http://pubman.mpdl.mpg.de/pubman/item/escidoc:2375806:4/component/escidoc:2375805/MPG_Handbuch_zur_Institutsgeschichte_Meteorologie.pdf. 2013.
- [5] Top 500 ORG. *Top 500 Hune 2018*. [Online; accessed Juli 9, 2018]. ? URL: <https://www.top500.org/lists/2018/06/>.