

Offshore-Windparks

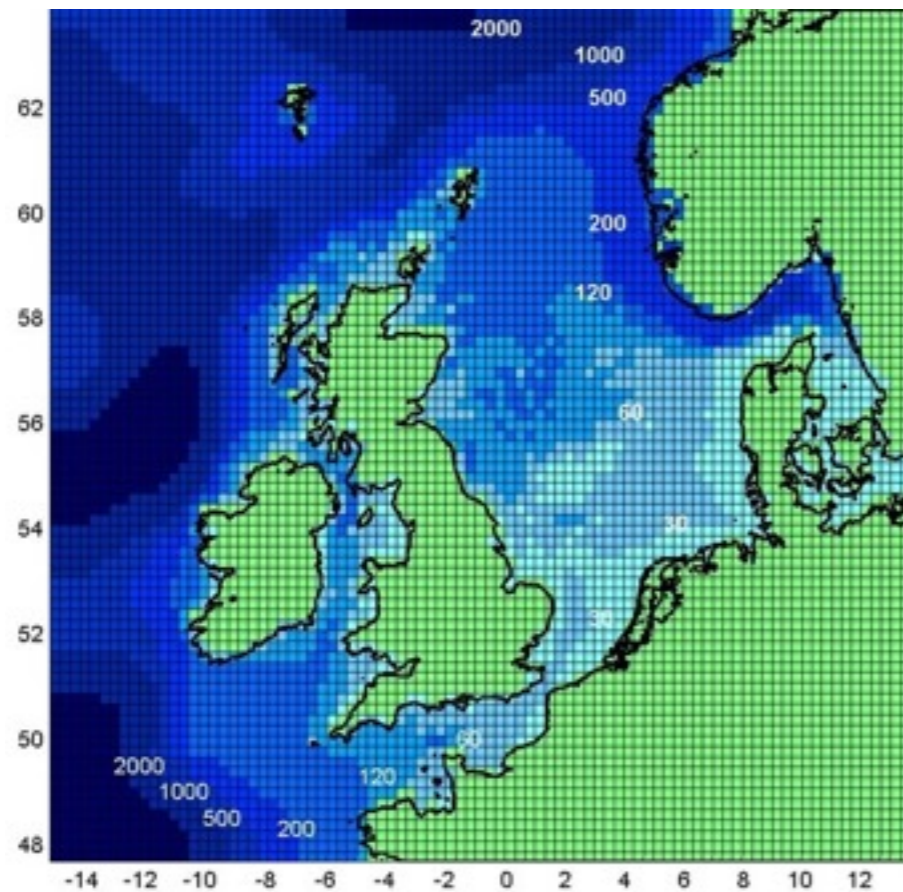
Parallelisierung von Analysewerkzeugen von
Ökosystem-Langzeitläufen in Fortran



Aufgabe

- Untersuchung der Sauerstoffkonzentration in der Nordsee
- Zeitraum von 36 Jahren (1970 - 2006)
- Daten aus Simulation
- Parallelisierung in Fortran mithilfe MPI
- Ergebnisse zusammenfassen und aufbereiten

ECOHAM4



- Simulationszeitraum:
1970 - 2006
- Schichttiefe: 5 - 3500 m
- Länge einer Gitterzelle:
ca. 20 km

Einleseprogramm

- Einleseprogramm liest NetCDF-Dateien des Modells

```
program read_data
  use mod_netcdf_read
  implicit none

  character (len = *), parameter :: filename = "D093_o2o_2000.nc"
  character (len = *), parameter :: varname="o2o"
  real, dimension(jmax, imax, kmax) :: var
  integer :: i, j, irec, ierr

  do irec=1,1
    call read_data_ncdf(filename,varname,irec,ierr,array3d=var)
    if (ierr.ne.0) print *, "ERROR reading netcdf data for ", varname
    write(*, '(88i5)') ((int(var(j,i,1)),j=1,jmax),i=imax,1,-1)
  enddo
end program read_data
```

Hilfsprogramm

- Ermittelt die letzte Meeresschicht für jeden Punkt

```
call read_data_ncdf("/home/lenhart/Projekt/D093_o2o_1984.nc", "o2o", day, ierr, array3d=var)
if (ierr.ne.0) print *, "ERROR reading netcdf data for ", "o2o"
```

```
longiloop: do longitude=1, jmax
  latiloop: do latitude=1, imax
    layerloop: do layer=1, kmax
      if(var(longitude, latitude, layer) < -9000 .AND. layer == 1) then
        array(longitude, latitude) = 0
        exit layerloop
      else if(var(longitude, latitude, layer) < -9000) then
        array(longitude, latitude) = layer-1
        exit layerloop
      else if (var(longitude, latitude, layer) > 0 .AND. layer == 24) then
        array(longitude, latitude) = 24
        exit layerloop
      end if
    end do layerloop
  end do latiloop
end do longiloop
```

Hauptprogramm

- Zählt für jeden Punkt die Anzahl der Tage mit Sauerstoffdefizit (< 6 mg/l)

```
time: do day=1,365
  call read_data_ncdf("/home/lenhart/Projekt/D093_o2o_"/year_char/" .nc", "o2o", day, ierr, array3d=var)
  if (ierr.ne.0) print *, "ERROR reading netcdf data for ", "o2o"

  longiloop: do longitude=1,jmax
    latiloop: do latitude=1,imax
      if (input(longitude, latitude) > 0) then
        if (var(longitude,latitude,input(longitude, latitude)) < 187.5) then
          output(longitude,latitude) = (output(longitude,latitude) + 1)
        end if
      else
        output(longitude,latitude) = 9999
      end if
    end do latiloop
  end do longiloop
end do time
```


Message Passing Interface

- Standard zum Nachrichtenaustausch bei parallelen Programmen auf verteilten Systeme
- Drei grundlegende Befehle, die zum Einsatz kommen:

```
call MPI_SEND(year, 4, MPI_CHAR, 1, 42, MPI_COMM_WORLD, ierr)
```

```
call MPI_RECV(year, 4, MPI_CHAR, 0, 42, MPI_COMM_WORLD, status, ierr)
```

```
call MPI_BCAST(input, imax*jmax, MPI_INTEGER, 0, MPI_COMM_WORLD, ierr)
```


Programmablauf

H Broadcast Info über letzte Meeresschicht

H Lineare Verteilung der ersten Jahrgänge

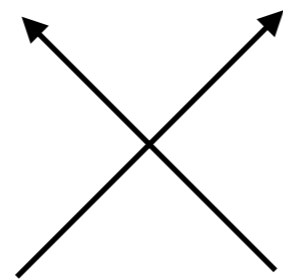
S Bearbeiten des Jahrgangs

S Ergebnisse verschicken

H Warten auf Ergebnisse

H Nächstes Jahr verschicken (bis fertig)

H Zusammenfassen der Ergebnisse

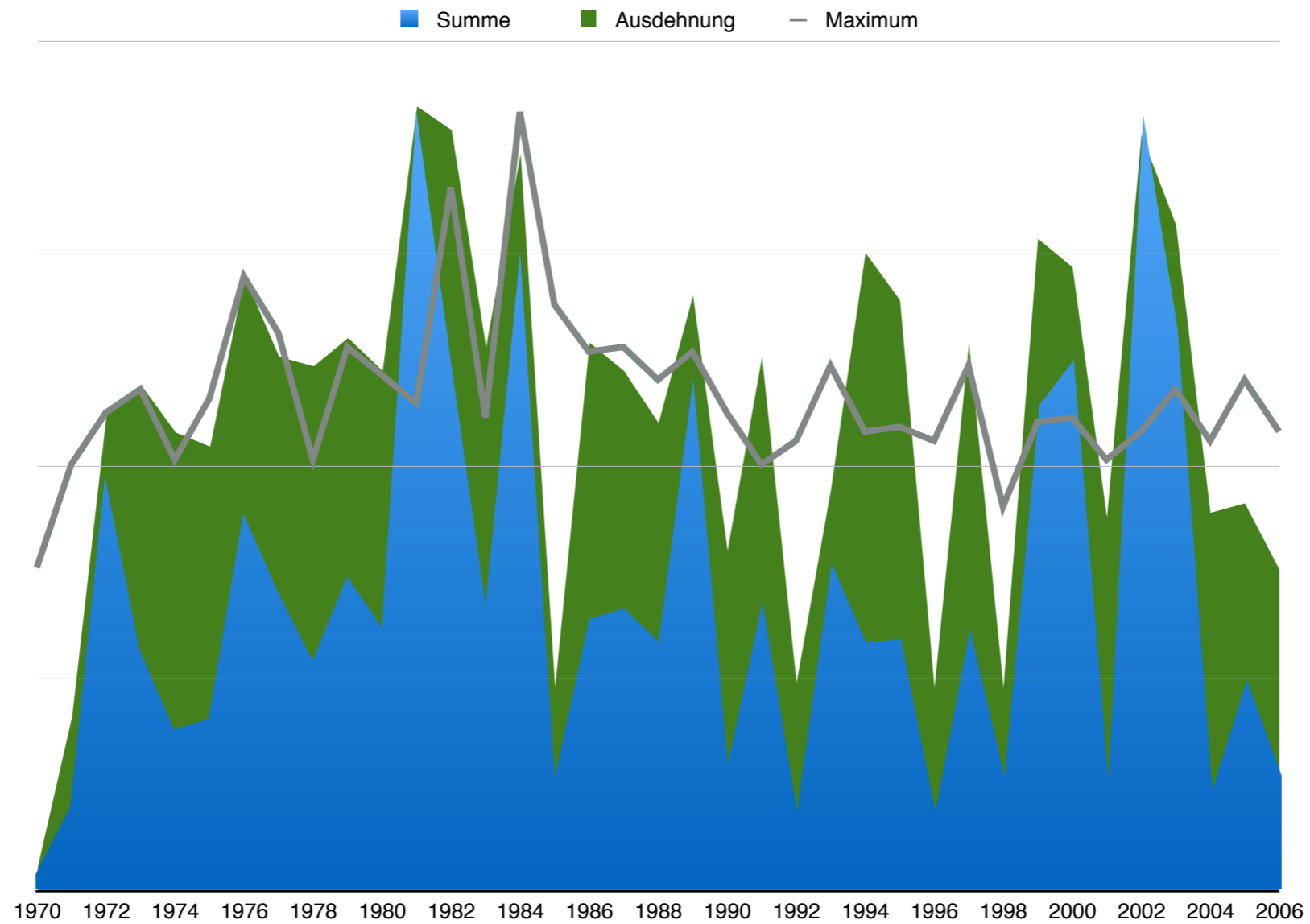


Zusammengefasste Werte

- Summe: Anzahl der Tage mit Sauerstoffdefizit von allen Punkten
- Ausdehnung: Anzahl der Punkte, die mindestens einen Tag von Sauerstoffdefizit betroffen sind
- Maximum: Maximale Anzahl der Tage mit Sauerstoffdefizit an einem Punkt

```
sum(results(year-min_year+1, :, :), results(year-min_year+1, :, :) < 9999)  
count(results(year-min_year+1, :, :) > 0 .AND. results(year-min_year+1, :, :) < 9999)  
maxval(results(year-min_year+1, :, :), results(year-min_year+1, :, :) < 9999)
```

Zusammengefasste Werte



Schwierigkeiten

- Fortran
 - Makefile
- Message Passing Interface
 - Syntax
 - Anordnung der Programmteile

```
if (rank .eq. 0) then
    ! Haupt-Anweisung 1
end if
if (rank .ne. 0) then
    ! Subanweisung 1
end if
if (rank .eq. 0) then
    ! Haupt-Anweisung 2 wird nicht erreicht
end if
```

Vielen Dank für Eure
Aufmerksamkeit