

# MPI Datentypen

Seminar Effiziente Programmierung

Michel Böker  
Universität Hamburg  
Fachbereich Informatik

11. Januar 2021

# Gliederung

## 1. Grundlagen

- Was und wofür ist MPI?
- MPI-Routinen
- Demo

## 2. Arten der Kommunikation

- Point-To-Point
- Kollektive Kommunikation

## 3. Datentypen

- Vordefinierte
- Abgeleitete

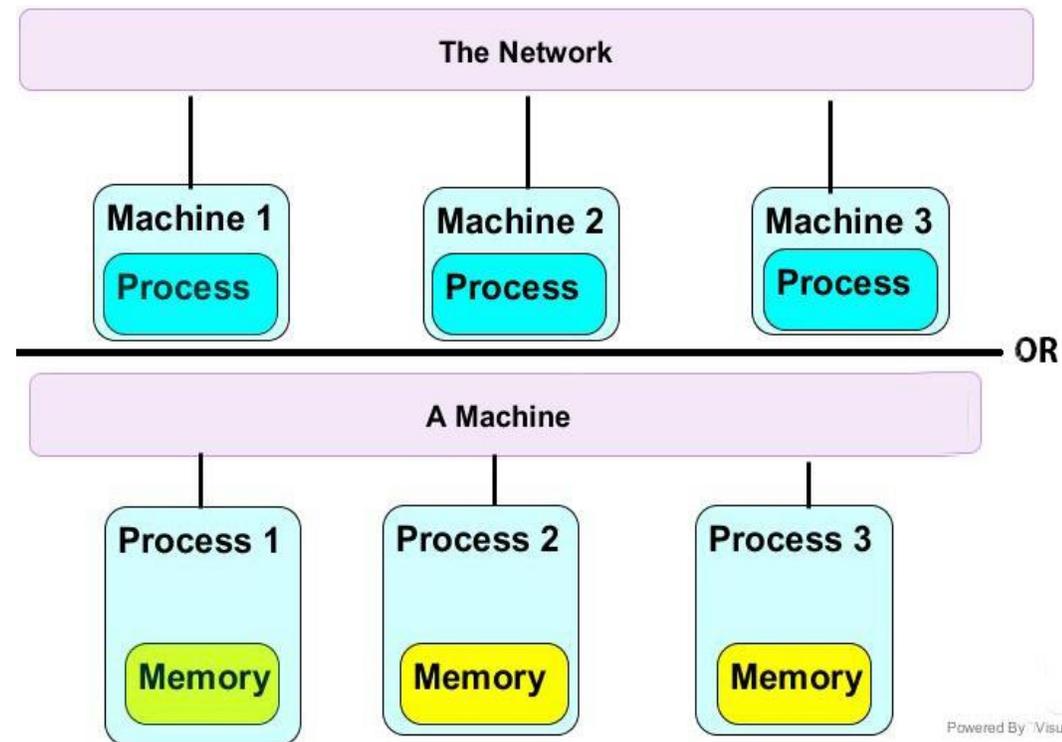
# Grundlagen

## 1. Grundlagen

- Was und wofür ist MPI?
  - MPI-Routinen
  - Demo
- ## 2. Arten der Kommunikation
- ## 3. Datentypen

# Message Passing Interface

- Basiert auf dem Message-Passing-Programmiermodell
- Prozess kann nicht auf Speicher eines anderen Prozesses zugreifen
- Berechnung  $\longleftrightarrow$  Übertragung



Powered By Visual Paradigm Community Edition

## 1. Grundlagen

- Was und wofür ist MPI?
  - MPI-Routinen
  - Demo
2. Arten der Kommunikation
3. Datentypen

# Message Passing Interface

- Standard-Library für den Nachrichtenaustausch bei parallelen Berechnungen auf verteilten Systemen
- legt eine Programmierschnittstelle fest
- Allgegenwärtig im HPC
- Zwei große MPI-Anbieter



**MPICH**

## 1. Grundlagen

- Was und wofür ist MPI?
- MPI-Routinen
- Demo

## 2. Arten der Kommunikation

## 3. Datentypen

# Message Passing Interface

Vorteile von MPI:

- ✓ standardisiert
- ✓ portabel
- ✓ weit verbreitet
- ✓ skalierbar

# MPI Routinen

*Fortran*

## 1. Grundlagen

- Was und wofür ist MPI?
  - **MPI-Routinen**
  - Demo
- ## 2. Arten der Kommunikation
- ## 3. Datentypen

## **MPI\_Init** ( ierror )

- Muss als erste MPI-Funktion aufgerufen werden

## **MPI\_Finalize** ( ierror )

- Muss als letzte MPI-Funktion aufgerufen werden

## **MPI\_Comm\_size** ( comm, size, ierror )

- Liefert Anzahl der Prozesse in der MPI-Umgebung

## **MPI\_Comm\_rank** ( comm, rank, ierror )

- Liefert den Rang des aktuellen Prozesses in der MPI-Umgebung

# Aufbau MPI Programm

## 1. Grundlagen

- Was und wofür ist MPI?
- MPI-Routinen
- **Demo**

2. Arten der Kommunikation

3. Datentypen

Demo

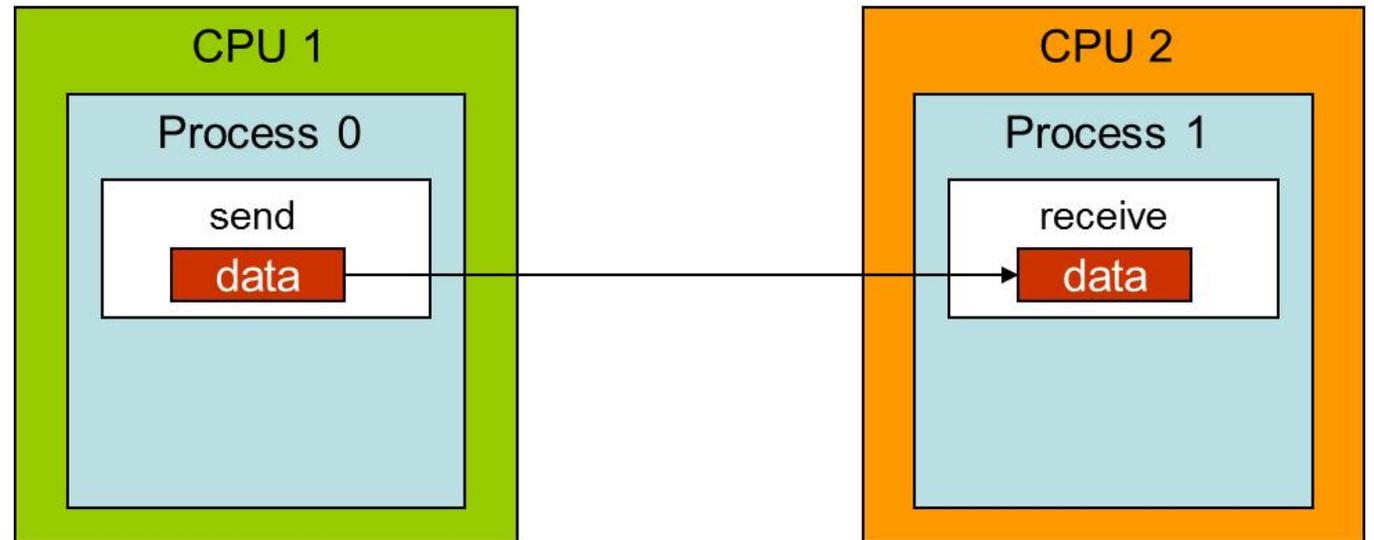
# Arten der Kommunikation

# Arten der Kommunikation

- Point-To-Point
- Kollektive Kommunikation

# Point-To-Point Kommunikation

- Genau zwei Prozesse
- Sender und Empfänger
- Identifiziert durch ihren Rang im Kommunikator



- 1. Grundlagen
- 2. Arten der Kommunikation
  - Point-To-Point
  - Kollektive
- 3. Datentypen

# Point-To-Point Kommunikation

Blockierend

**Fortran**

**MPI\_SEND** ( buf, count, datatype, dest, stag, comm, ierror )

Input	Beschreibung
<b>buf</b>	Adresse des Sendepuffers
<b>count</b>	Anzahl der zu versendenden Elemente
<b>datatype</b>	Typ der zu versendenden Daten
<b>dest</b>	Rang des Empfängers im Kommunikator
<b>stag</b>	Etikett zur Unterscheidung der Nachrichten
<b>comm</b>	Kommunikator

- 1. Grundlagen
- 2. Arten der Kommunikation
  - Point-To-Point
  - Kollektive
- 3. Datentypen

# Point-To-Point Kommunikation

Blockierend

**Fortran**

**MPI\_RECV** ( buf, count, datatype, src, rtag, comm, status, ierror )

Input	Beschreibung
<b>count</b>	Anzahl der zu empfangenden Elemente
<b>datatype</b>	Typ der zu empfangenden Daten
<b>src</b>	Rang des Senders im Kommunikator
<b>rtag</b>	Tag zur Unterscheidung der Nachrichten
<b>comm</b>	Kommunikator
Output	Beschreibung
buf	Adresse des Empfangspuffers
status	Information über empfangene Nachricht

# Point-To-Point Kommunikation

## Phasen

1. Daten aus Sendepuffer kopieren & Nachricht erstellen
2. Nachricht zum Empfänger senden
3. Daten aus Nachricht lesen und im Empfangspuffer auspacken

## Datentypen

- Daten im Sendepuffer zu *MPI\_DATATYPE* in Send
- Gleicher *MPI\_DATATYPE* in Send und Recv
- Daten im Empfangspuffer zu *MPI\_DATATYPE* in Recv

# Point-To-Point Kommunikation

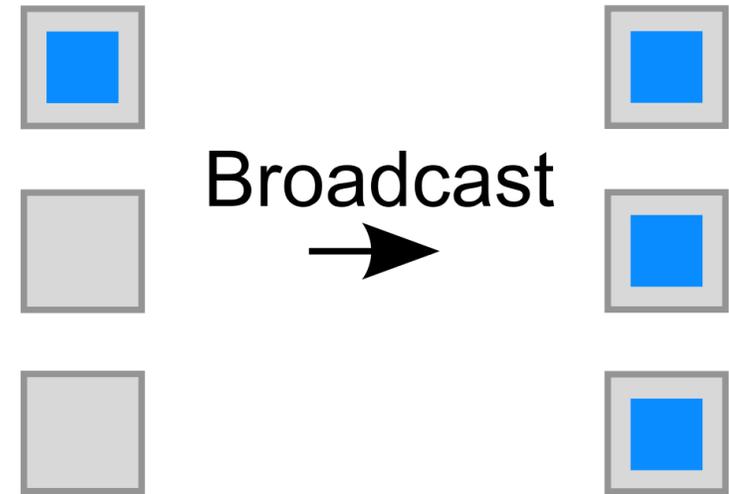
1. Grundlagen
2. Arten der Kommunikation
  - **Point-To-Point**
  - Kollektive
3. Datentypen

Demo

- 1. Grundlagen
- 2. Arten der Kommunikation
  - Point-To-Point
  - **Kollektive**
- 3. Datentypen

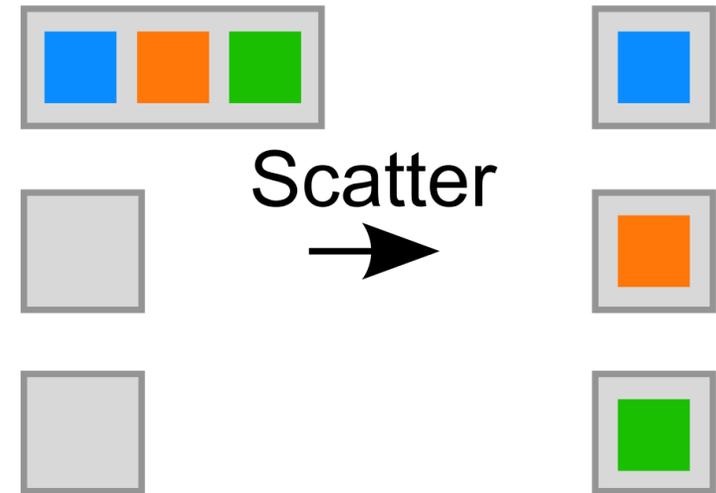
# Kollektive Kommunikation

- Alle Prozesse innerhalb des Kommunikators beteiligt



# Kollektive Kommunikation

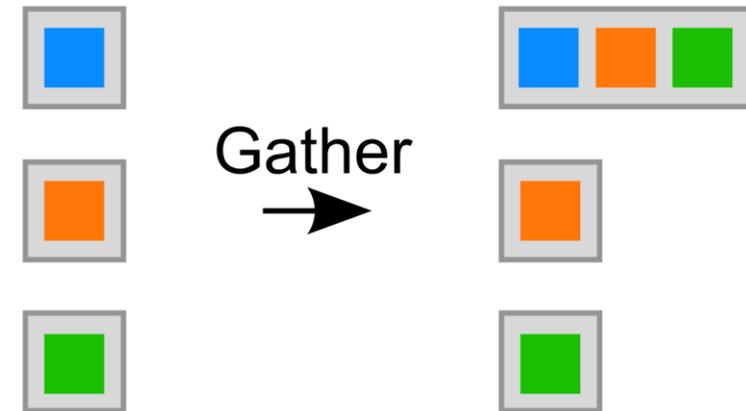
- Alle Prozesse innerhalb des Kommunikators beteiligt



- 1. Grundlagen
- 2. Arten der Kommunikation
  - Point-To-Point
  - **Kollektive**
- 3. Datentypen

# Kollektive Kommunikation

- Alle Prozesse innerhalb des Kommunikators beteiligt



# Datentypen

# Datentypen

- Vordefinierte
- Abgeleitete

- 1. Grundlagen
- 2. Arten der Kommunikation
- 3. Datentypen
  - **Vordefinierte**
  - Abgeleitete

# Vordefinierte Datentypen

MPI Datentyp	Fortran Datentyp
MPI_INTEGER	INTEGER
MPI_REAL	REAL
MPI_DOUBLE_PRECISION	DOUBLE_PRECISION
MPI_COMPLEX	COMPLEX
MPI_LOGICAL	LOGICAL
MPI_CHARACTER	CHARACTER(1)
MPI_BYTE	
MPI_PACKED	

# Abgeleitete Datentypen

## Wozu?

- Kommunikation verursacht Kosten



Beachte:

- Moderner Parallelrechner schafft ca. 3 Mrd. floating point operations / sec
- Nachrichtenaustausch aber nur 10 Mio. Wörter / sec

**Faktor 300 !**

Besser: Eine große Nachricht als viele kleine

# Abgeleitete Datentypen

## Wie?

- Konstruiere den neuen Datentyp mithilfe der dazugehörigen MPI-Routine:

`MPI_TYPE_CONTIGUOUS`

`MPI_TYPE_VECTOR`

`MPI_TYPE_INDEXED`

`MPI_TYPE_CREATE_STRUCT`

- Übergebe den neuen Datentyp:

`MPI_TYPE_COMMIT`

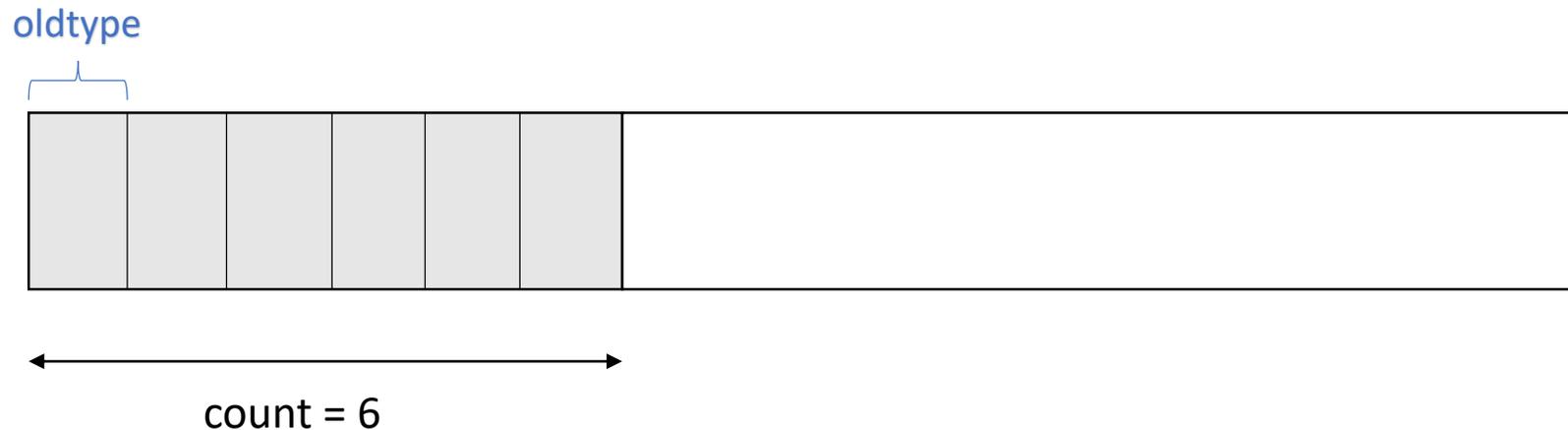
- Benutze den neuen Datentyp in send/receive etc.

# Datentyp Contiguous

**Fortran**

**MPI\_TYPE\_CONTIGUOUS** ( count, oldtype, newtype, ierror )

- Einfachste Art einen Datentyp abzuleiten
- Besteht aus einer bestimmten Anzahl gleichartiger Element



# Datentyp Vector

*Fortran*

**MPI\_TYPE\_VECTOR** ( count, blocklength, stride, oldtype, newtype, ierror )

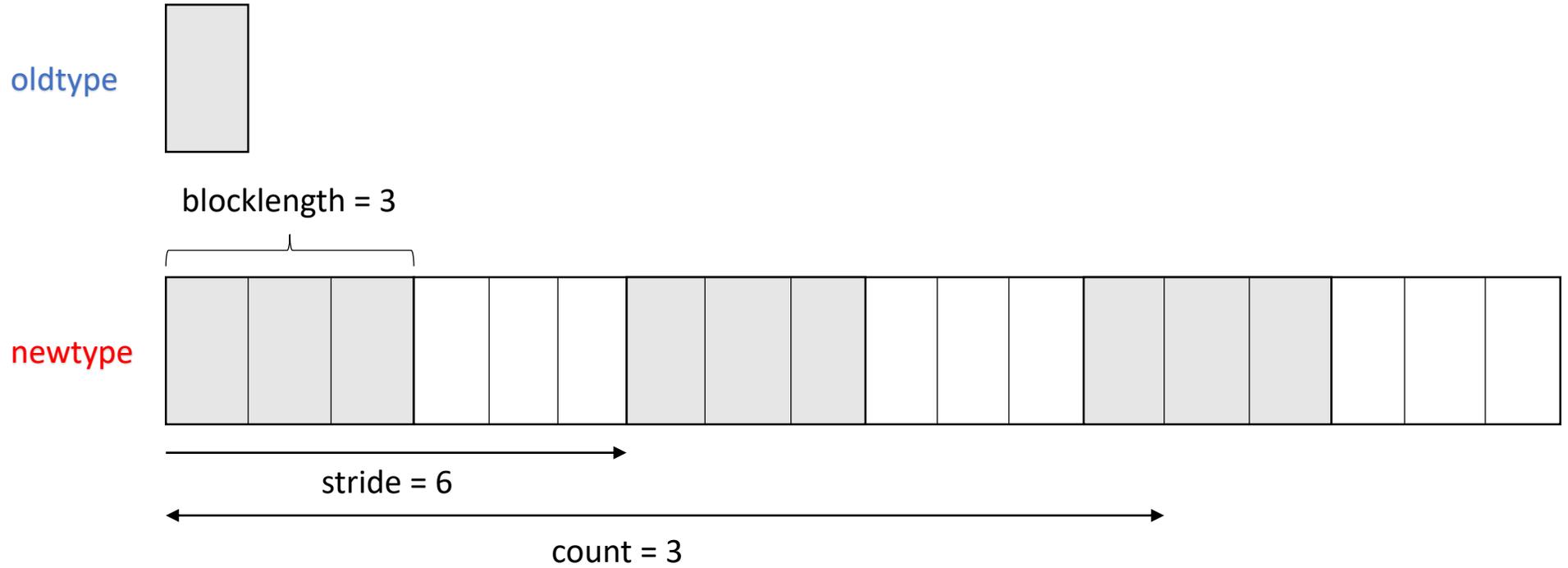
- Replikation eines Datentyps in mehrere gleichgroße Blöcke
- Abstand zwischen den Blöcken

- 1. Grundlagen
- 2. Arten der Kommunikation
- 3. Datentypen
  - Vordefinierte
  - Abgeleitete

# Datentyp Vector

**Fortran**

**MPI\_TYPE\_VECTOR** ( count, blocklength, stride, oldtype, newtype, ierror )



# Datentyp Indexed

*Fortran*

**MPI\_TYPE\_INDEXED** ( count, blocklengths, displs, oldtype, newtype, ierror )

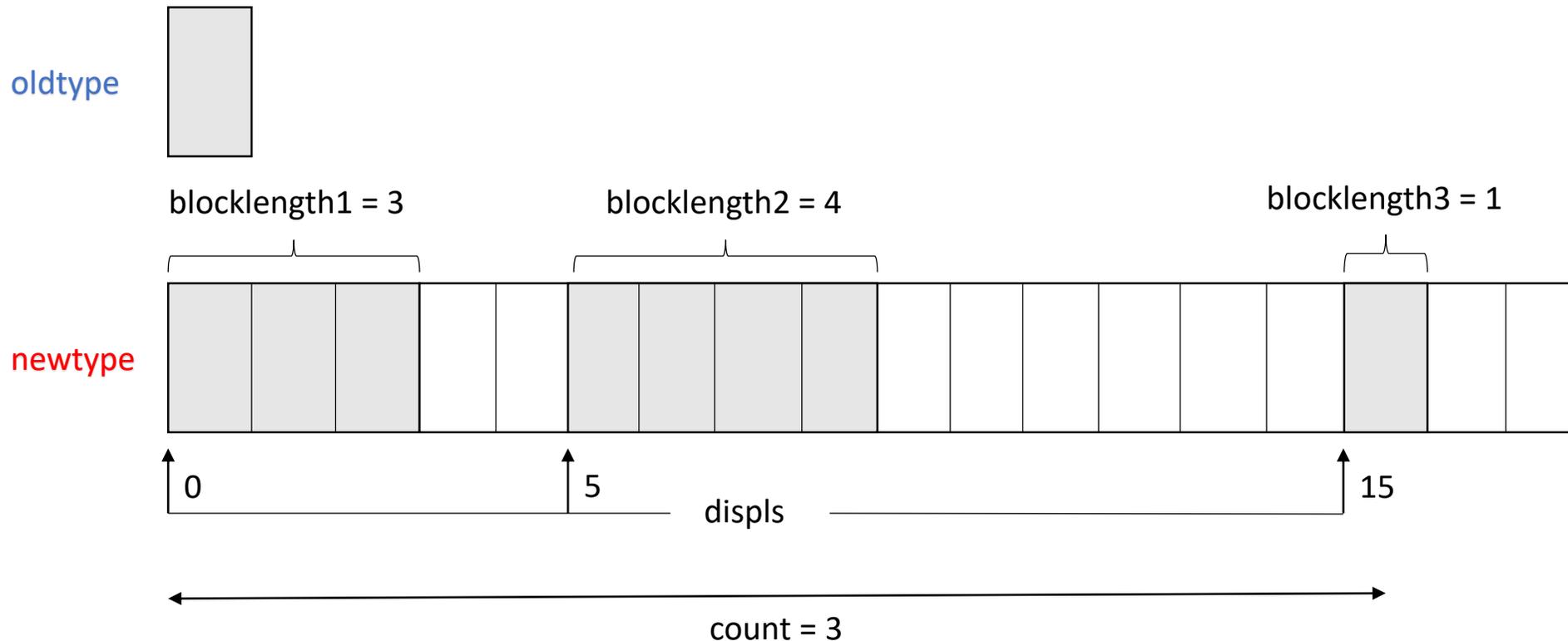
- Allgemeiner als Datentyp Vector
- Jeder Block kann eine unterschiedliche Anzahl Elemente enthalten
- Abstand zwischen Blöcken kann ebenfalls unterschiedlich sein

- 1. Grundlagen
- 2. Arten der Kommunikation
- 3. Datentypen
  - Vordefinierte
  - Abgeleitete

# Datentyp Indexed

**Fortran**

**MPI\_TYPE\_INDEXED** ( count, blocklengths, displs, oldtype, newtype, ierror )



# Datentyp Struct

**MPI\_TYPE\_CREATE\_STRUCT** ( count, blocklengths, displs, oldtypes, newtype, ierror )

**Fortran**

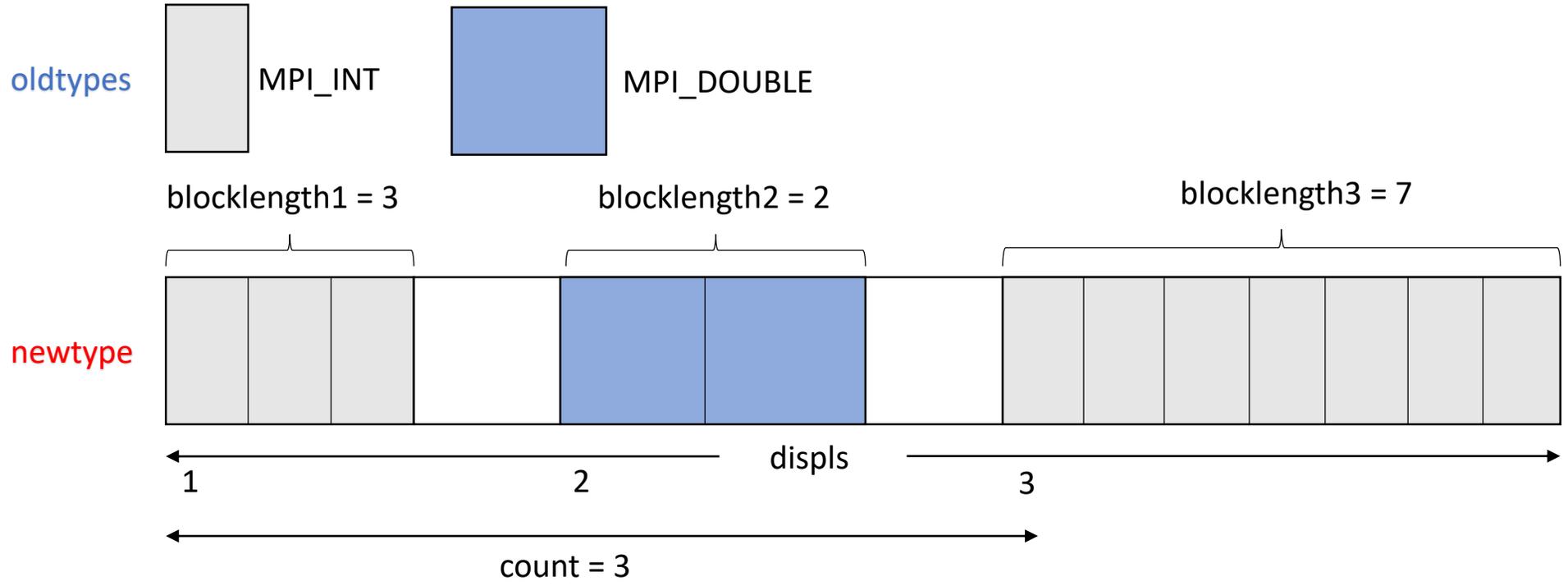
- Allgemeinste Form eines Typ-Konstruktors
- Jeder Block enthält Elemente eines bestimmten Datentyps
- Die verschiedenen Blöcken können sich in Datentyp, Länge und Blockanfang unterscheiden

- 1. Grundlagen
- 2. Arten der Kommunikation
- 3. Datentypen
  - Vordefinierte
  - Abgeleitete

# Datentyp Struct

**MPI\_TYPE\_CREATE\_STRUCT** ( count, blocklengths, displs, oldtypes, newtype, ierror )

**Fortran**



# Abgeleitete Datentypen

- 1. Grundlagen
- 2. Arten der Kommunikation
- 3. Datentypen
  - Vordefinierte
  - **Abgeleitete**

Demo

# Zusammenfassung

- MPI ist ein Standard für die parallele Berechnung auf verteilten Systemen
- Point-To-Point Kommunikation für Nachrichtenaustausch zwischen genau zwei Prozessen
- Vordefinierte Datentypen können unterschiedlich abgeleitet werden
- Abgeleitete Datentypen für die effizientere Kommunikation

# Literatur

1. MPI-Forum Docs, <https://www.mpi-forum.org/docs/>, letzter Zugriff: 02.01.2021
2. MPICH Docs, <https://www.mpich.org/static/docs/v3.3>, letzter Zugriff: 02.01.2021
3. Marc Snir, Steve Otto, Steven Huss Lederman, David Walker, Jack Dongarra: MPI The Complete Reference: The MIT Press
4. The Shared Memory and Message Passing Models of Interprocess Communication, [http://www.umsl.edu/~siegelj/CS4740\\_5740/Overview/MPSM.html](http://www.umsl.edu/~siegelj/CS4740_5740/Overview/MPSM.html), letzter Zugriff: 11.01.2021
5. Message Passing Interface Wikipedia, [https://de.wikipedia.org/wiki/Message\\_Passing\\_Interface](https://de.wikipedia.org/wiki/Message_Passing_Interface), letzter Zugriff: 09.01.2021
6. MPI Instructions for Windows, [https://abhilashreddy.com/writing/3/mpi\\_instructions.html](https://abhilashreddy.com/writing/3/mpi_instructions.html), letzter Zugriff: 09.01.2021
7. Using MPI with Fortran, <https://curc.readthedocs.io/en/latest/programming/MPI-Fortran.html>, letzter Zugriff: 11.01.2021