

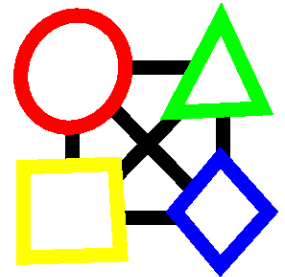
Optimierung des Kommunikationsverhaltens auf Metacomputern

Carsten Clauss

carsten@lfbs.rwth-aachen.de
<http://www.mp-mpich.de>



Lehrstuhl für Betriebssysteme
RWTH Aachen





Übersicht

Teil 1 – Metacomputing / gekoppelte Cluster

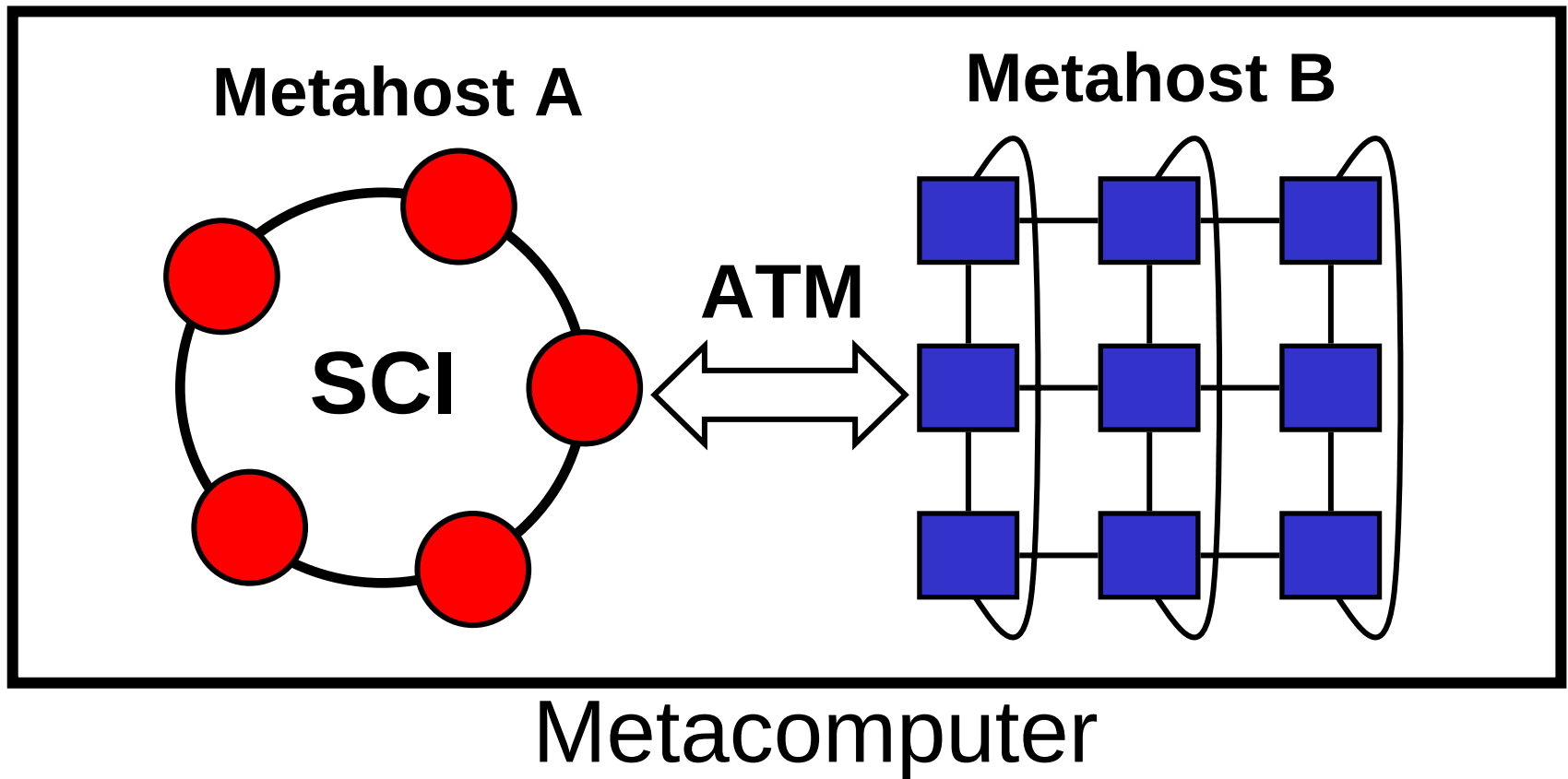
- Was ist ein Metacomputer?
- MetaMPICH - Bibliothek
- Vor- und Nachteile von Metacomputing

Teil 2 – Optimierung von MPI-Applikationen

- Randwertprobleme auf strukturierten Gittern
- Last-Balancierung auf Metacomputern
- Zu Evaluationszwecken: SmartPart / MetaComm
- Wann lohnt sich hier Metacomputing?

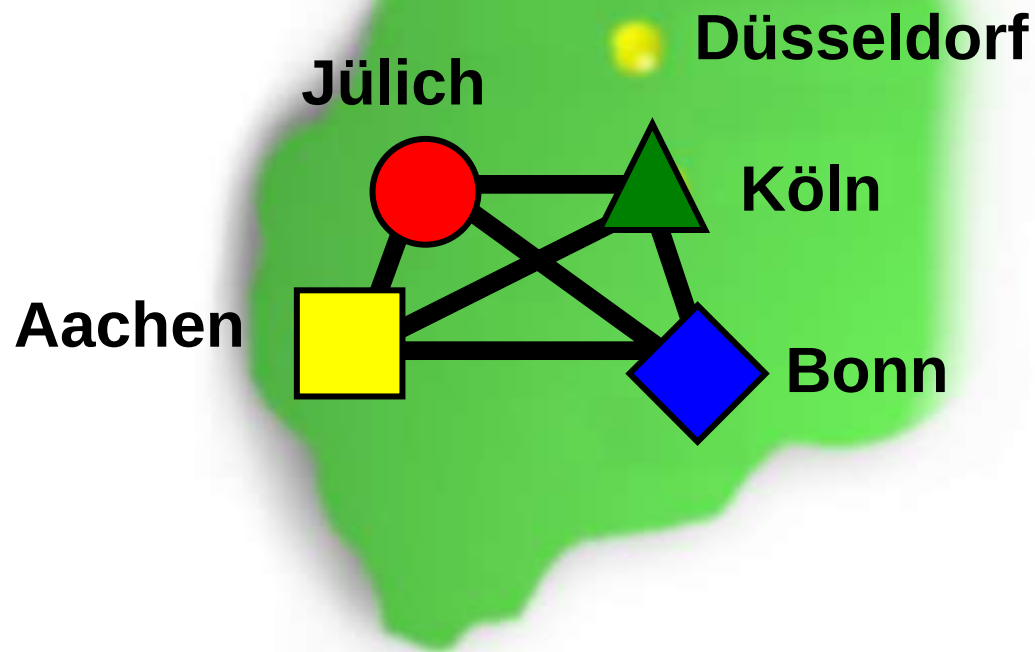


Gekoppelte Cluster





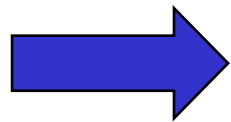
Ein möglicher Einsatz?





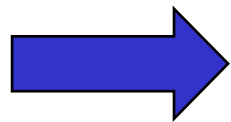
MetaMPICH / MP-MPICH

MetaMPICH[★]



Bibliothek zur *transparenten* Nutzung von MPI-Applikationen auf Metacomputern

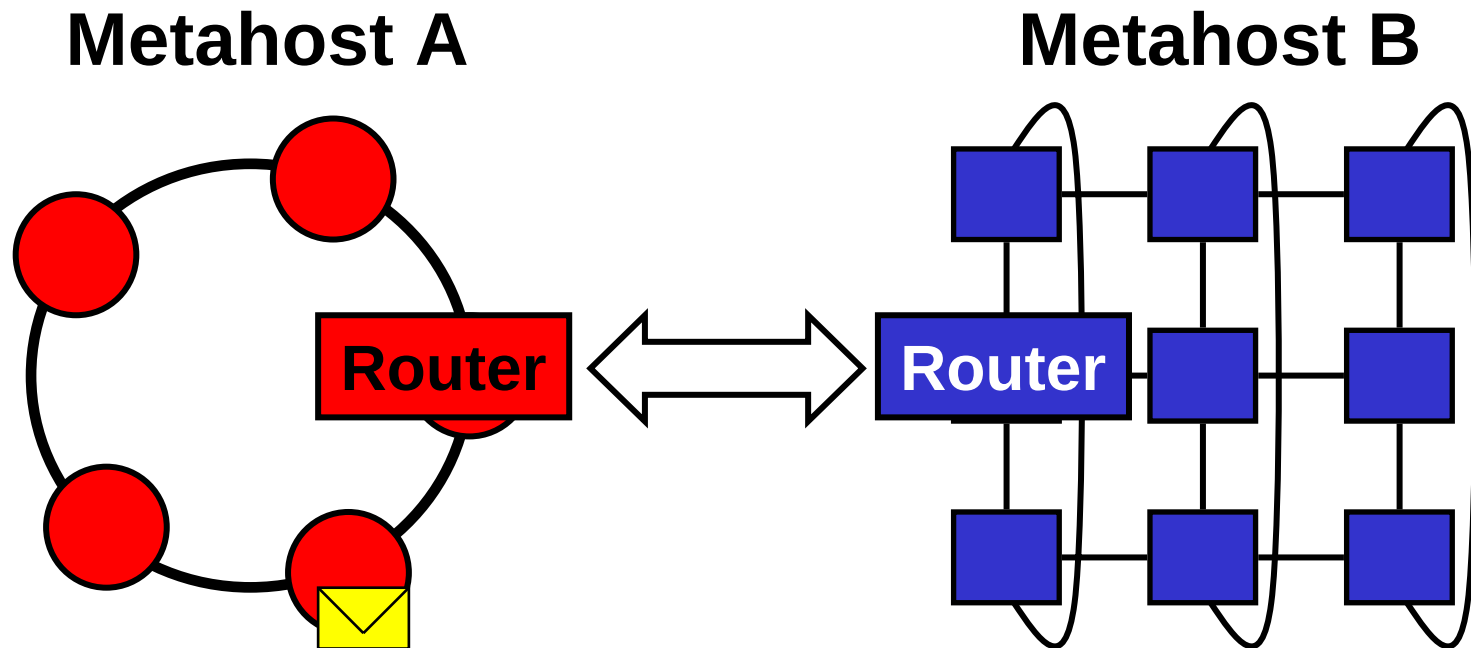
MP-MPICH



- Windows 2000/XP, Linux, Solaris
- SCI, TCP, SHMEM **internal**
- TCP and AAL5 **external**

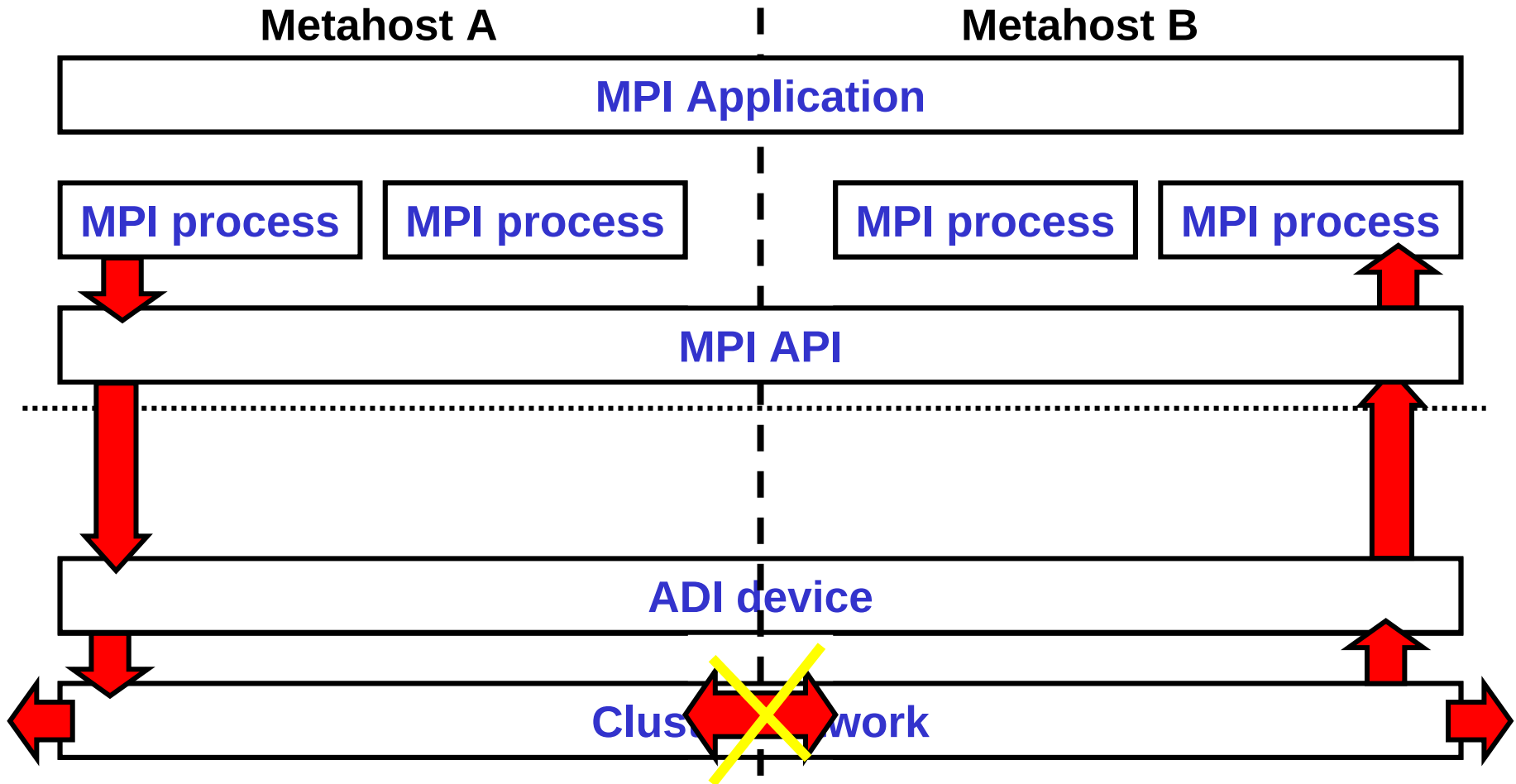


Architektur von MetaMPICH



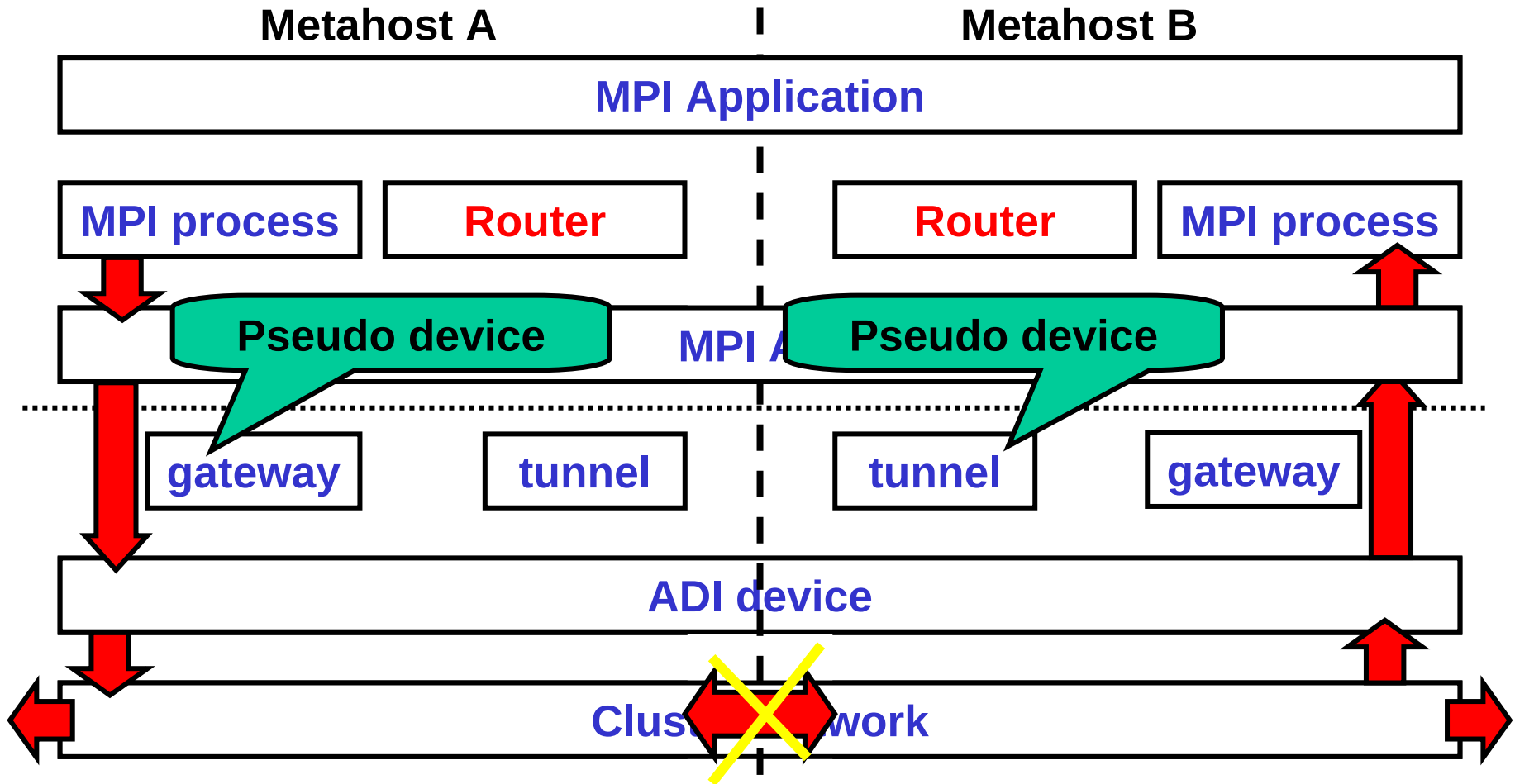


Architektur von MetaMPICH



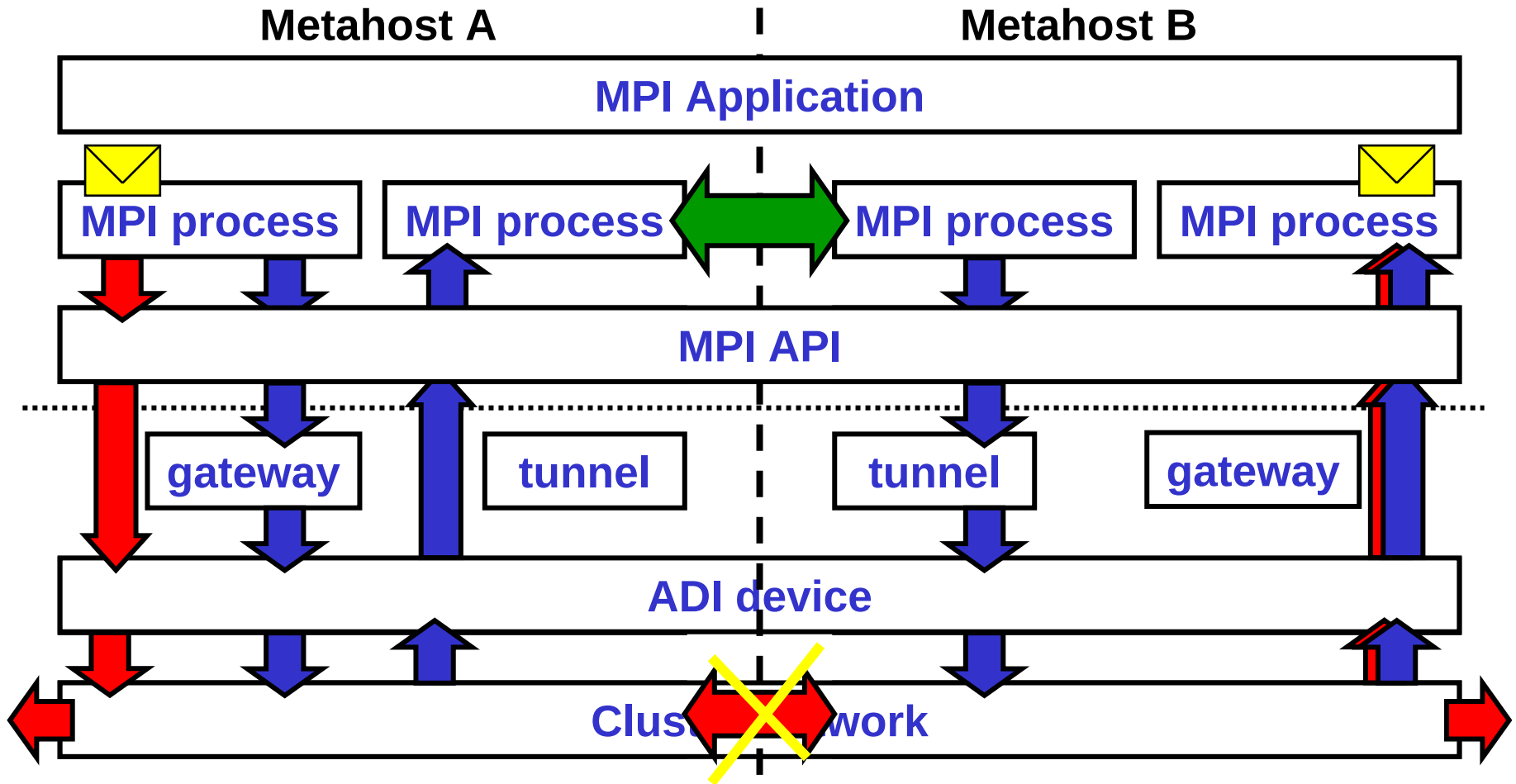


Architektur von MetaMPICH



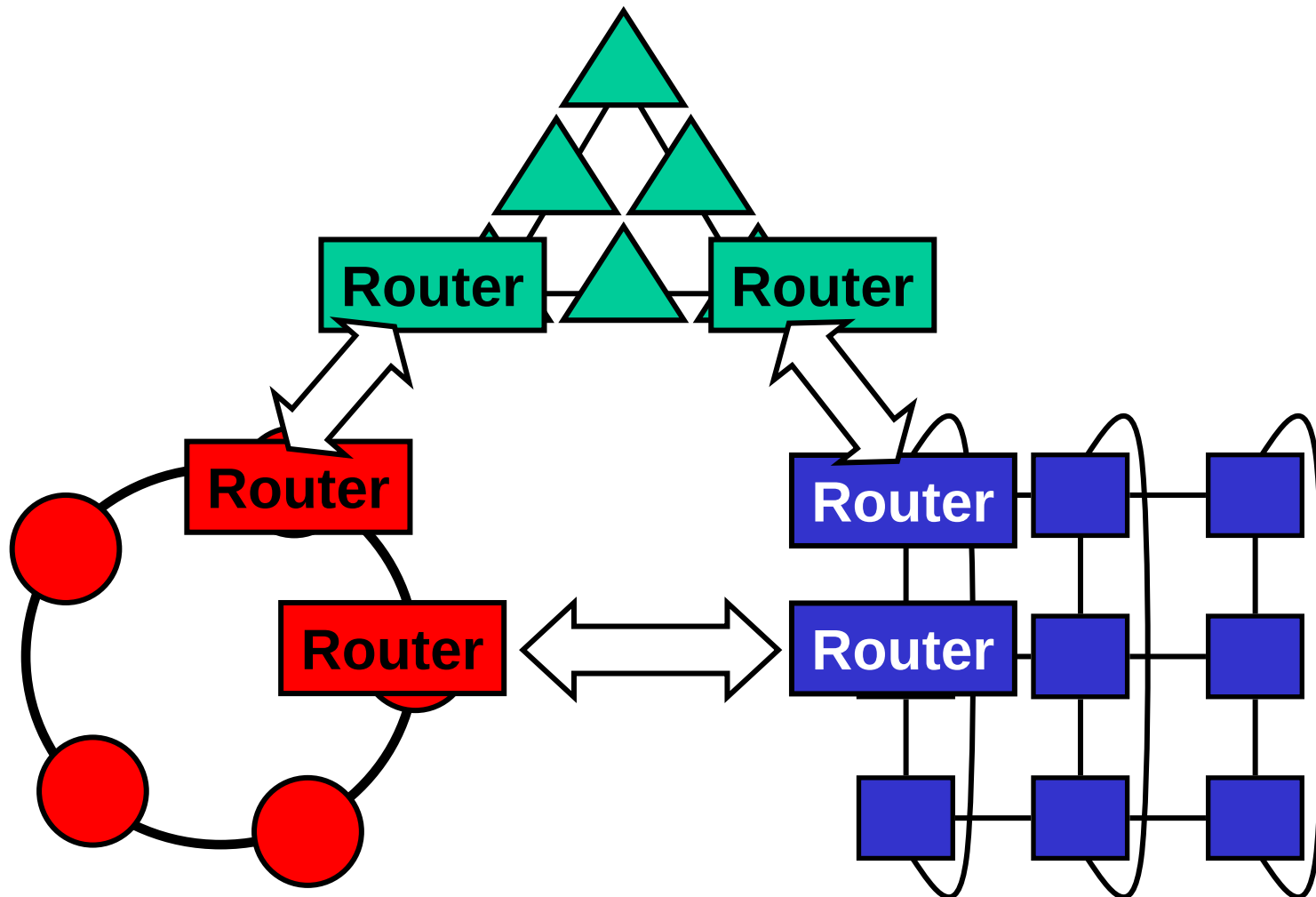


Architektur von MetaMPICH





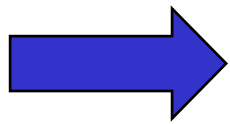
Architektur von MetaMPICH





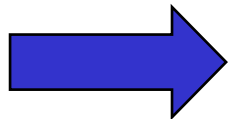
Metacomputing

Vorteile:



Viel **mehr MPI Prozesse** durch Zugriff auf die verteilte Rechenleistung

Nachteile:



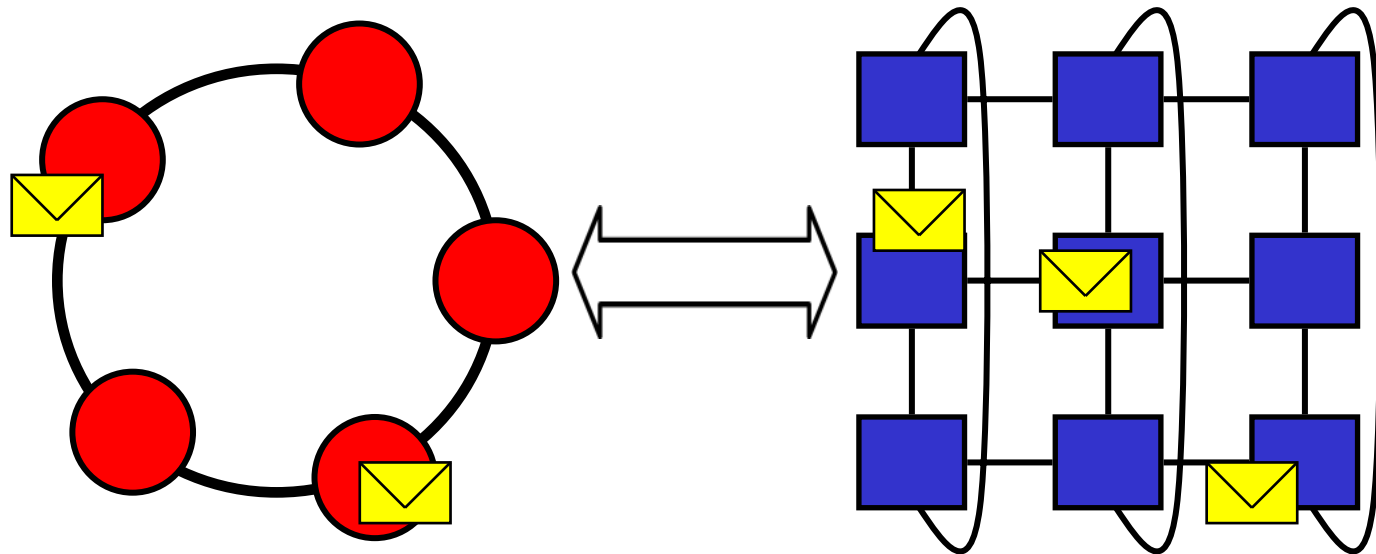
Die **Inter-Metahost-Kommunikation** ist der **Flaschenhals** des Systems

Die Applikationen können **nicht** von den **schnellen internen Netzwerken profitieren**



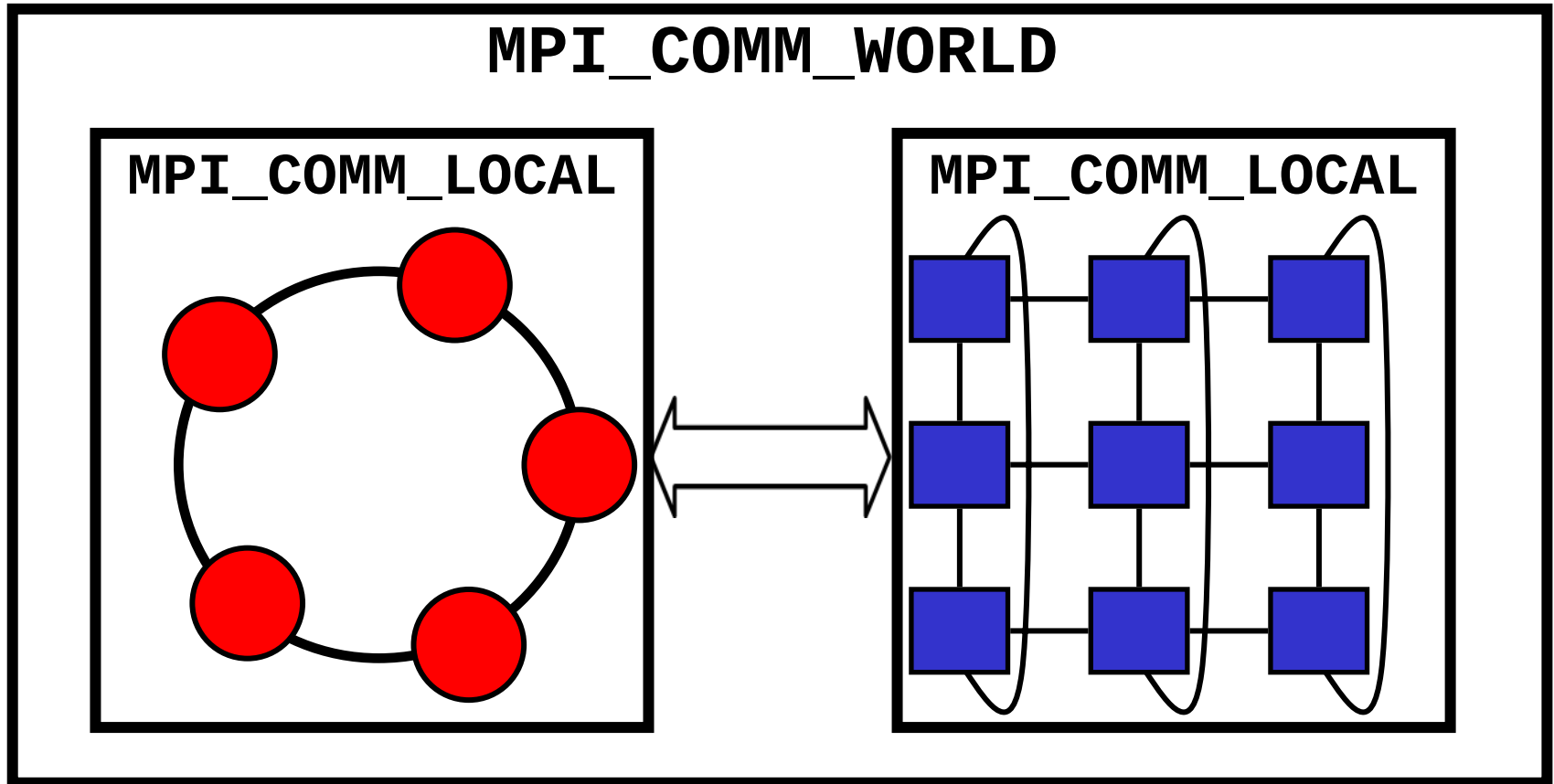
Metacomputing

Gruppierung der Prozesse!





Metacomputing



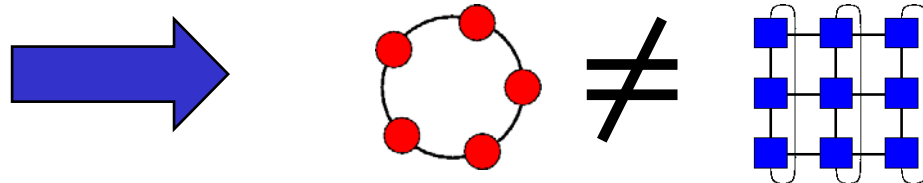


Metacomputing

Wichtig: Last-Balancierung !!!

➔ **Metahosts müssen NICHT identisch sein**

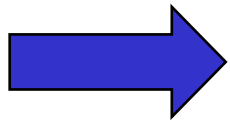
- ➔
- **Unterschiedliche CPU Leistung**
 - **Unterschiedliche Anzahl an Knoten**
 - **Unterschiedliche interne Netzwerke**



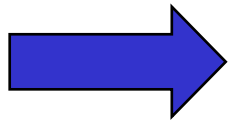


Metacomputing

Was ist zu berücksichtigen?



Flaschenhals:
Inter-Metahost-**Kommunikation**



Heterogenes System:
Last-Balance !!!



Übersicht

Teil 1 – Metacomputing / gekoppelte Cluster

- Was ist ein Metacomputer?
- MetaMPICH - Bibliothek
- Vor- und Nachteile von Metacomputing

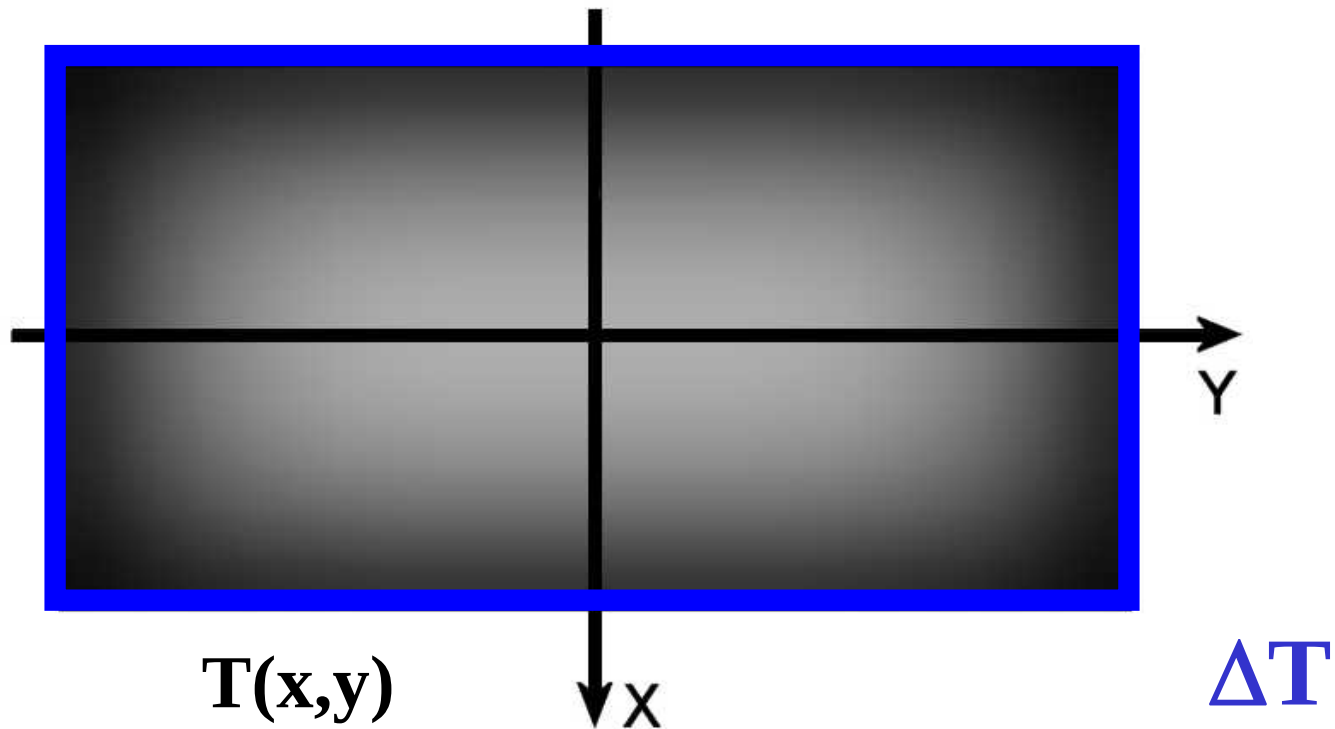
Teil 2 – Optimierung von MPI-Applikationen

- Randwertprobleme auf strukturierten Gittern
- Last-Balancierung auf Metacomputern
- Zu Evaluationszwecken: SmartPart / MetaComm
- Wann lohnt sich hier Metacomputing?



Randwertprobleme

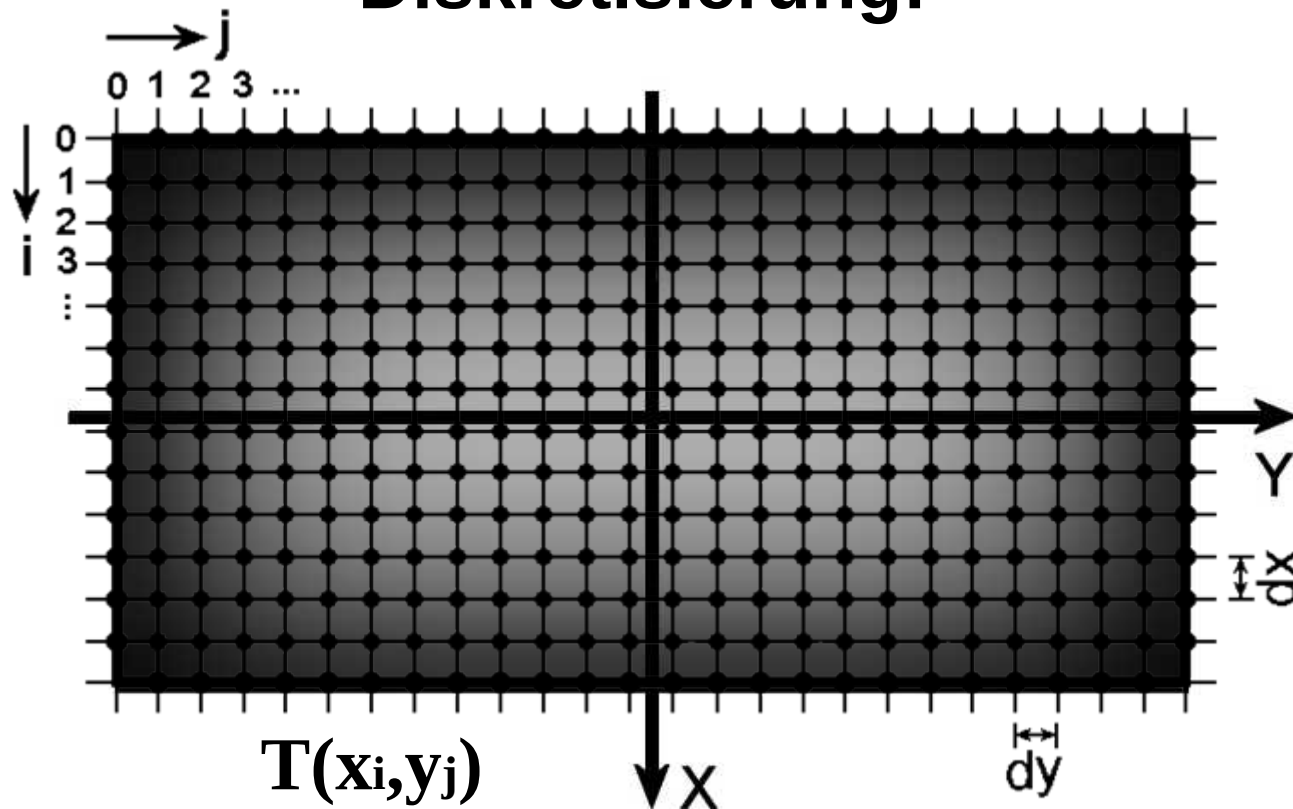
Temperaturverteilung in einer Platte:





Randwertprobleme

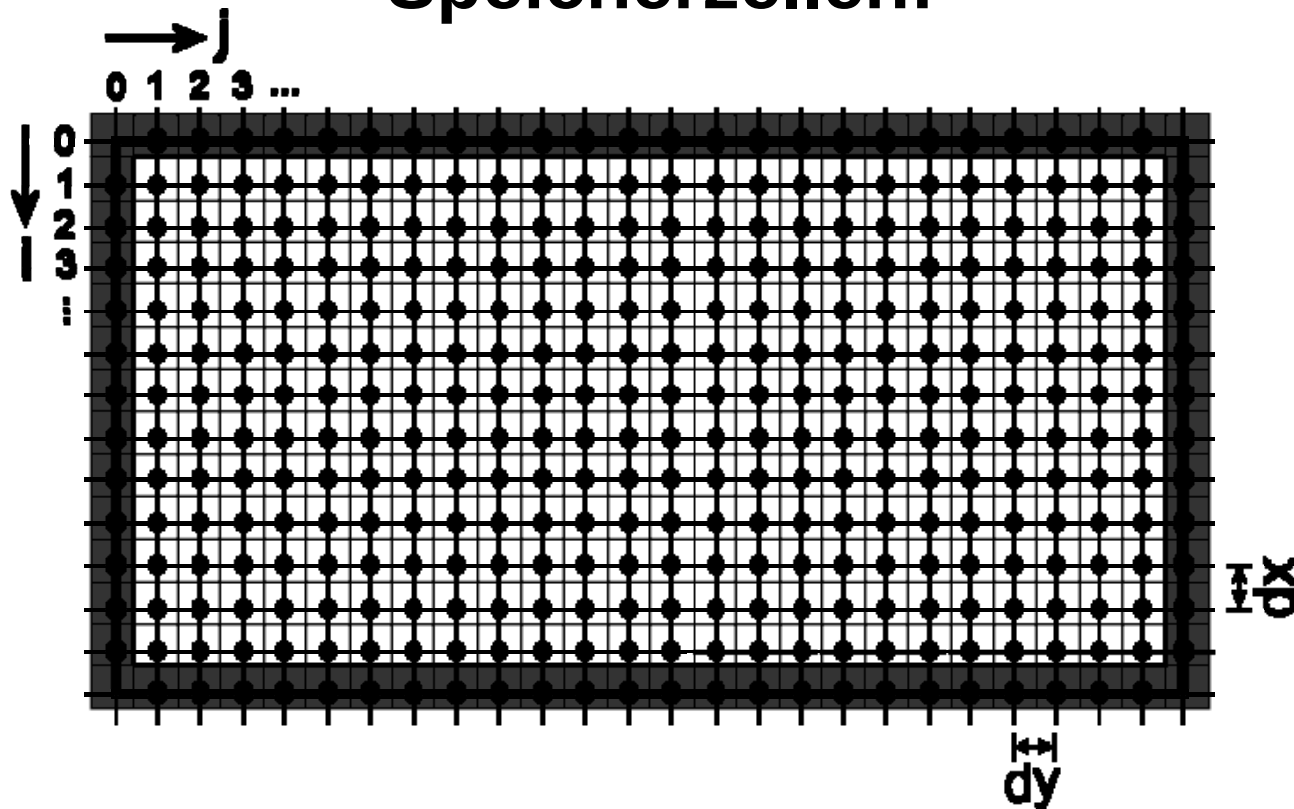
Diskretisierung:





Randwertprobleme

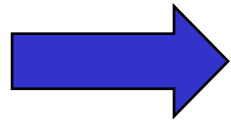
Speicherzellen:





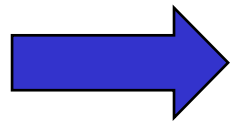
Randwertprobleme

Diskretisierung:



große und dünnbesetzte
lineare Gleichungssysteme

Iterative Gleichungslöser:

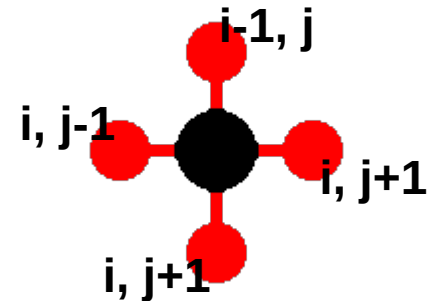


einfachste Vertreter: **Jacobi-**
oder **Gauss-Seidel** Verfahren



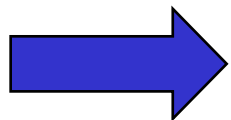
Randwertprobleme

Iterationsvorschrift:



$$T[i][j] = 0.25 (T[i-1][j] + T[i+1][j] + T[i][j-1] + T[i][j+1])$$

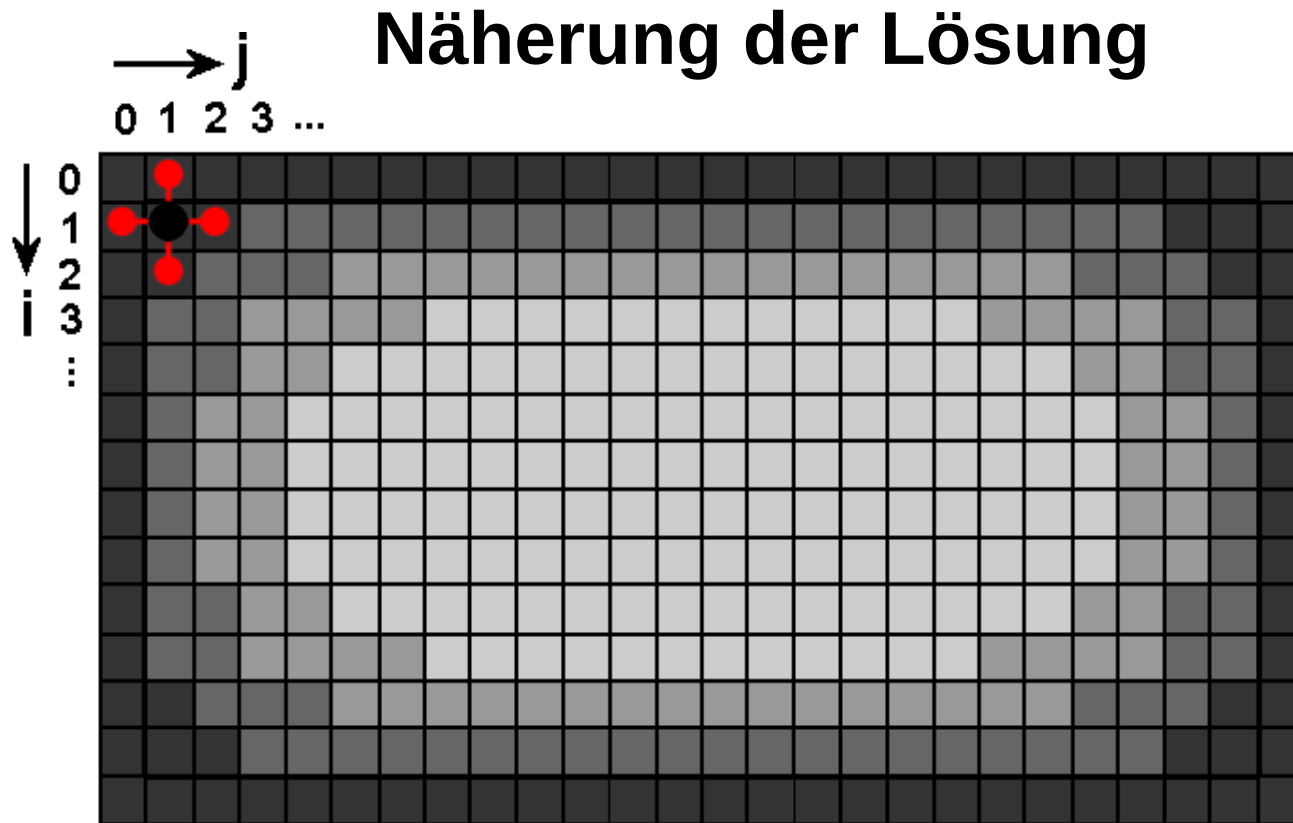
Große Anzahl an Iterationen über alle i, j :



Näherung für die exakte **diskrete Lösung**



Randwertprobleme





Parallelisierung

Aufteilung des Gebietes:

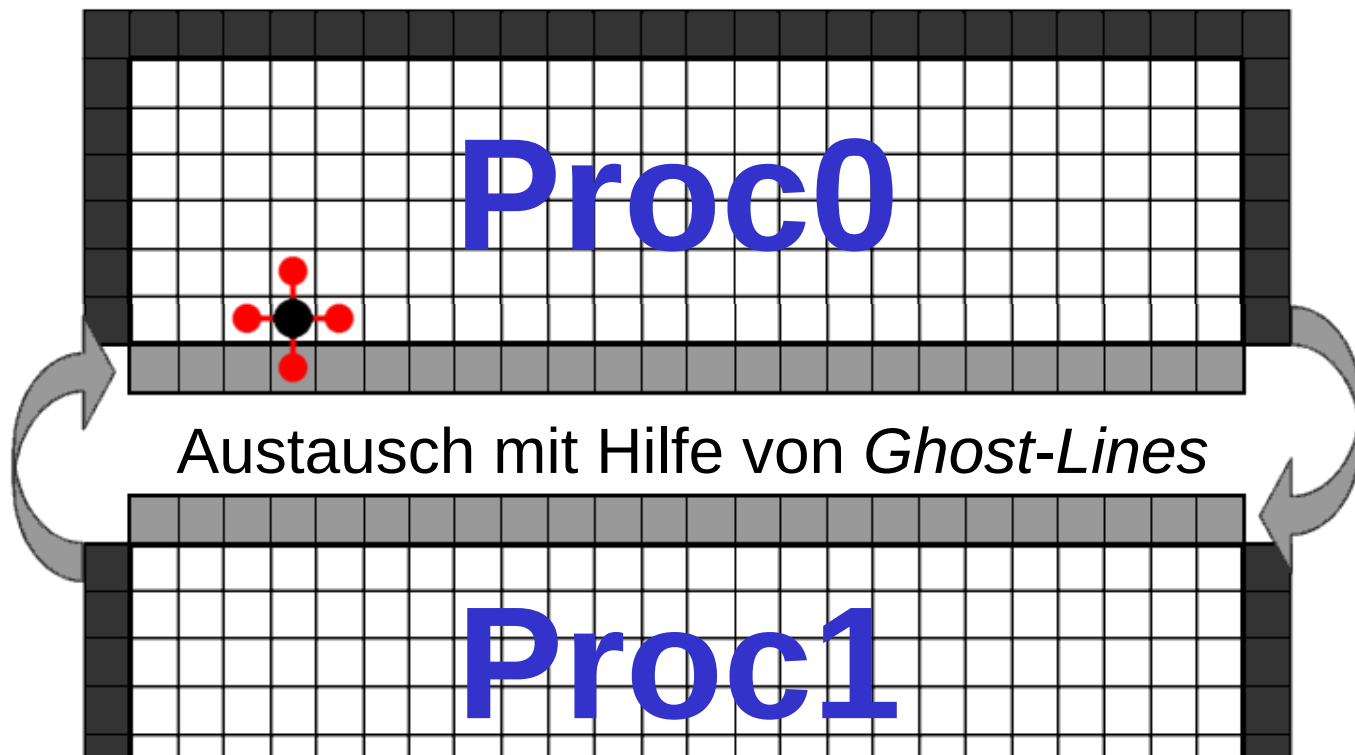




Abbildung auf Metacomputer

Gruppierung der Prozesse möglich?

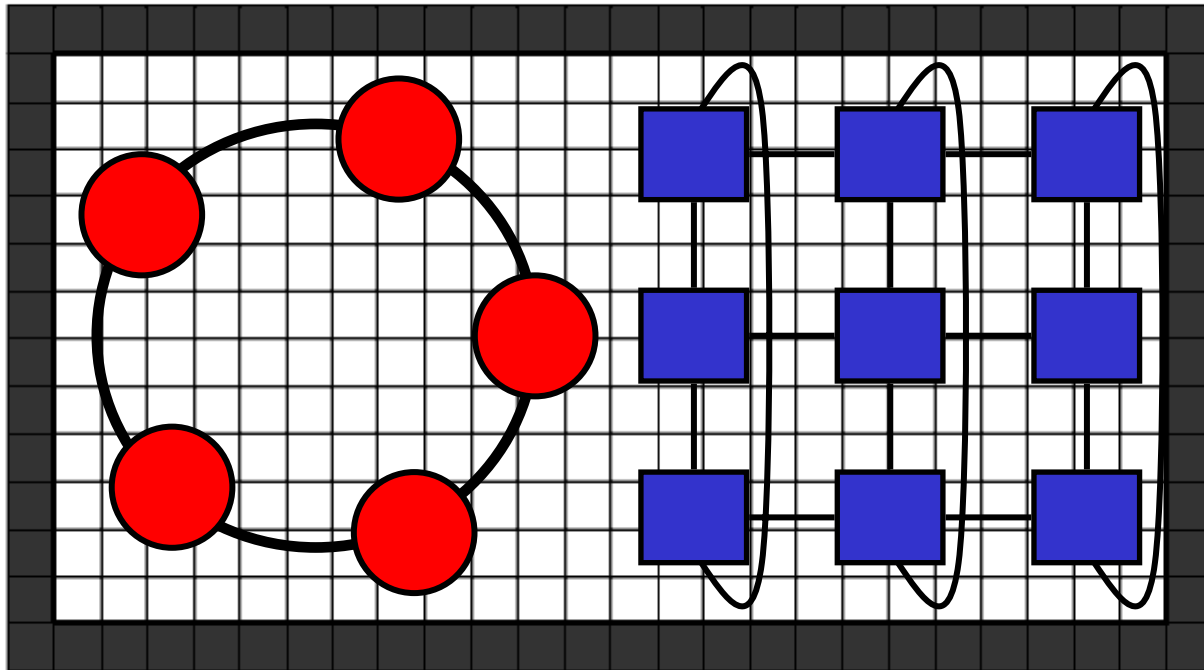




Abbildung auf Metacomputer

Innere Randwerte:

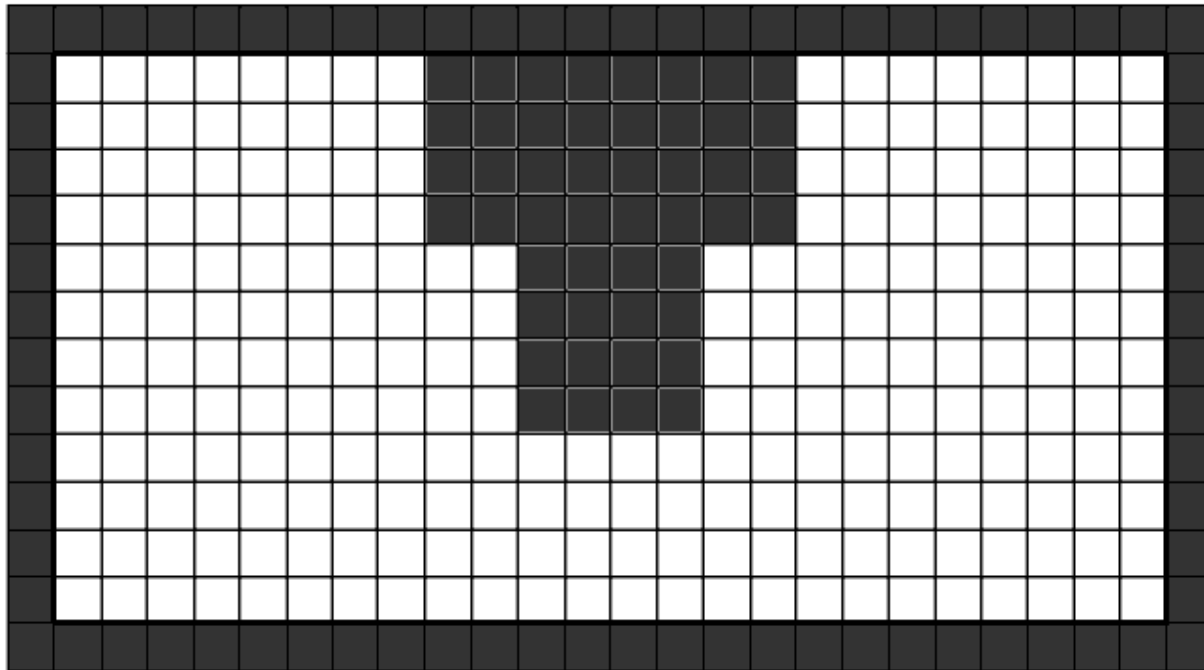




Abbildung auf Metacomputer

Reduzierte Kommunikation:

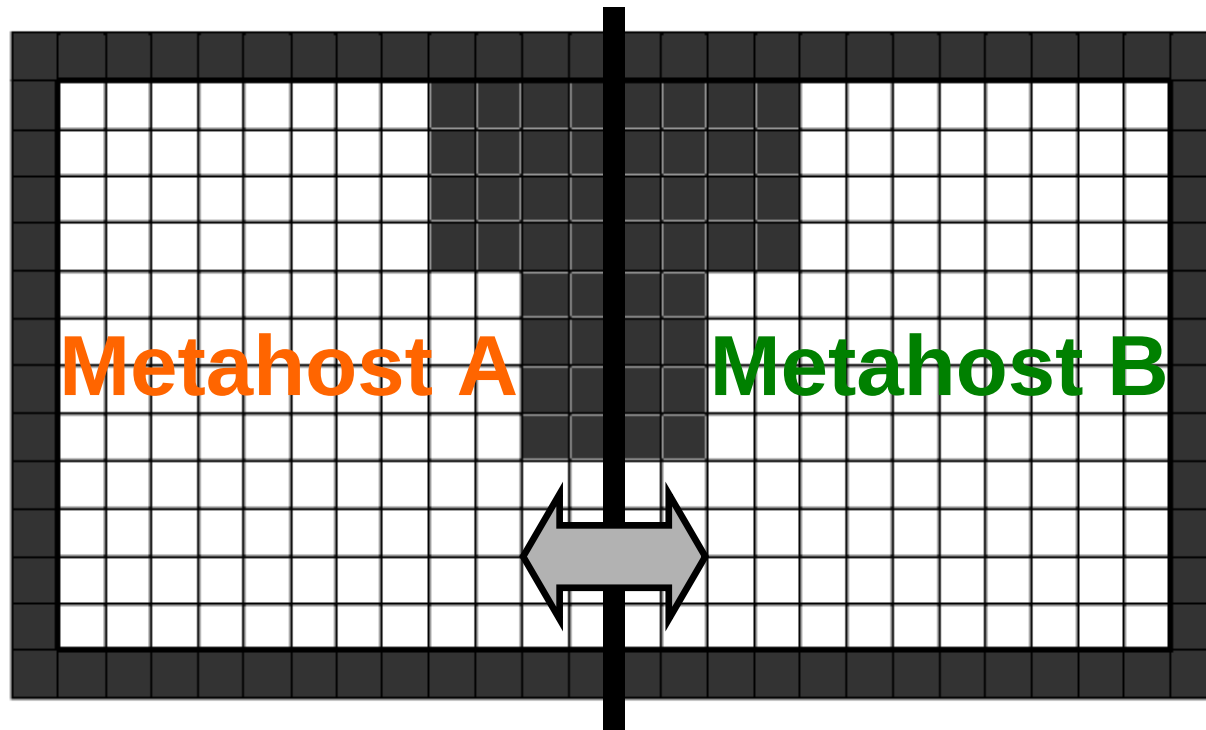




Abbildung auf Metacomputer

Last-Balance?

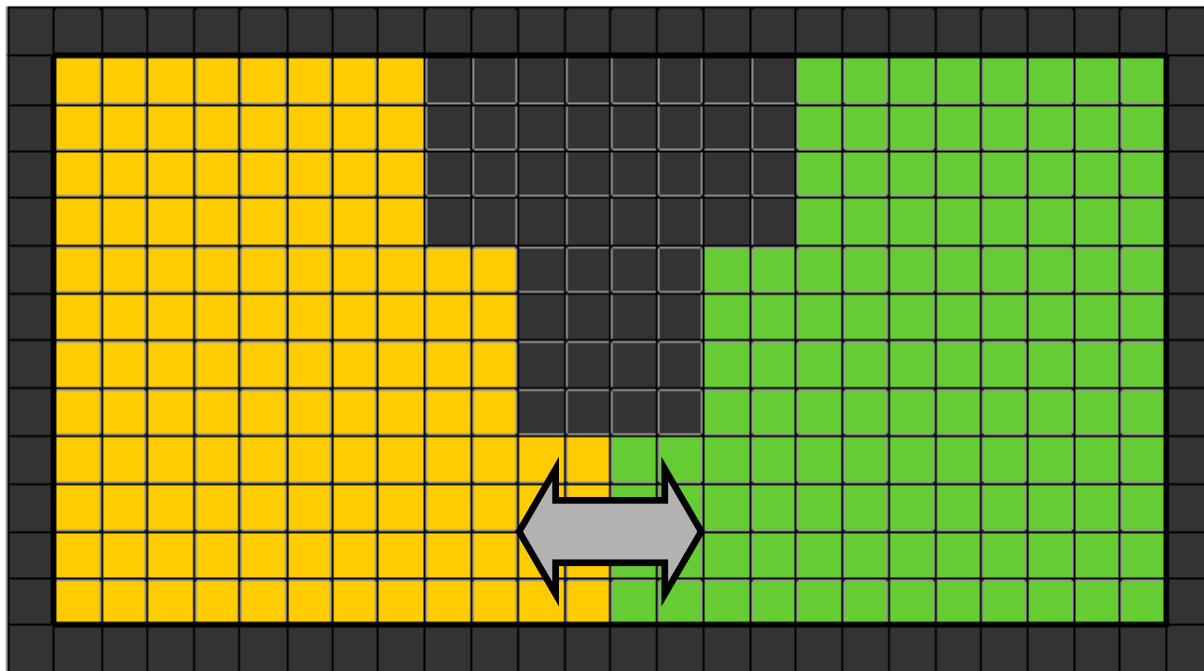




Abbildung auf Metacomputer

Weiterverteilung auf die Prozesse:

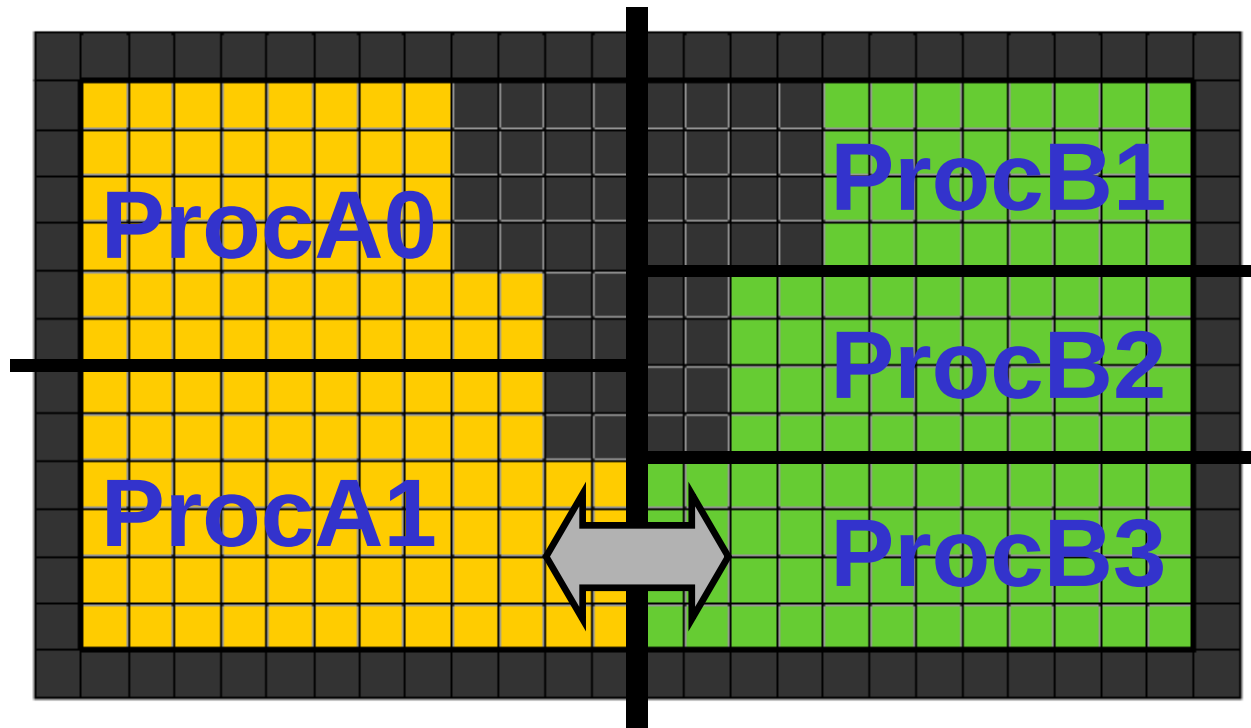
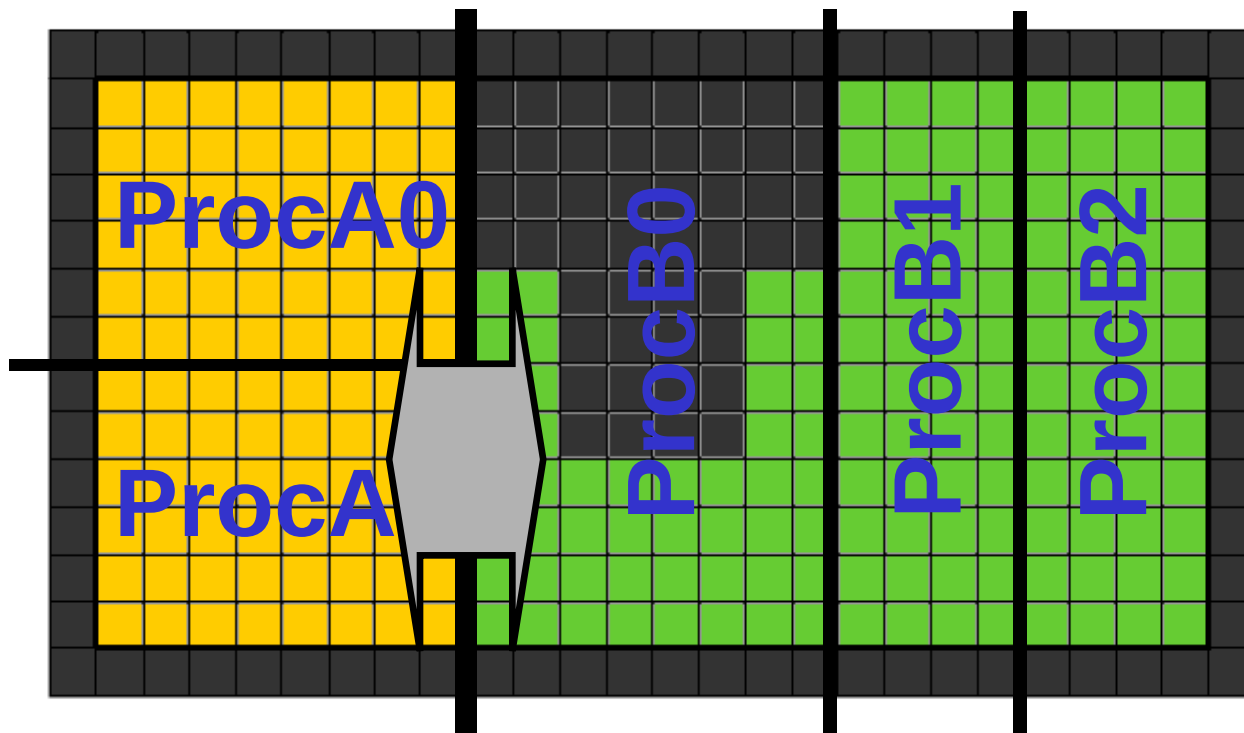




Abbildung auf Metacomputer

Lastverteilung VS Kommunikation





Smart Partitioner

Eine *intelligente* Gebietsverteilung umfasst:

➔ **Last-Balance *und* reduzierte Kommunikation** (falls möglich)

Eine *intelligente* Gebietsverteilung hängt ab von:

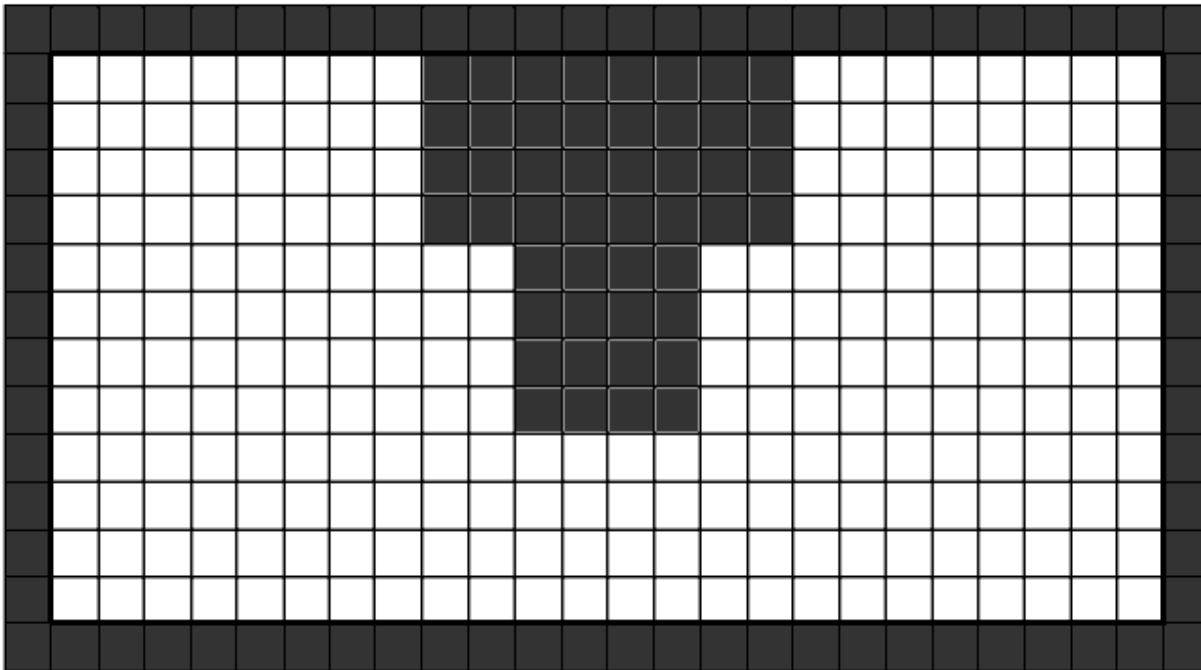
➔ - dem gegebenen **Randwertproblem**
- den **Strukturen** des **Metacomputers**

➔ **Smart Partitioner (SmartPart)** ✨



Smart Partitioner

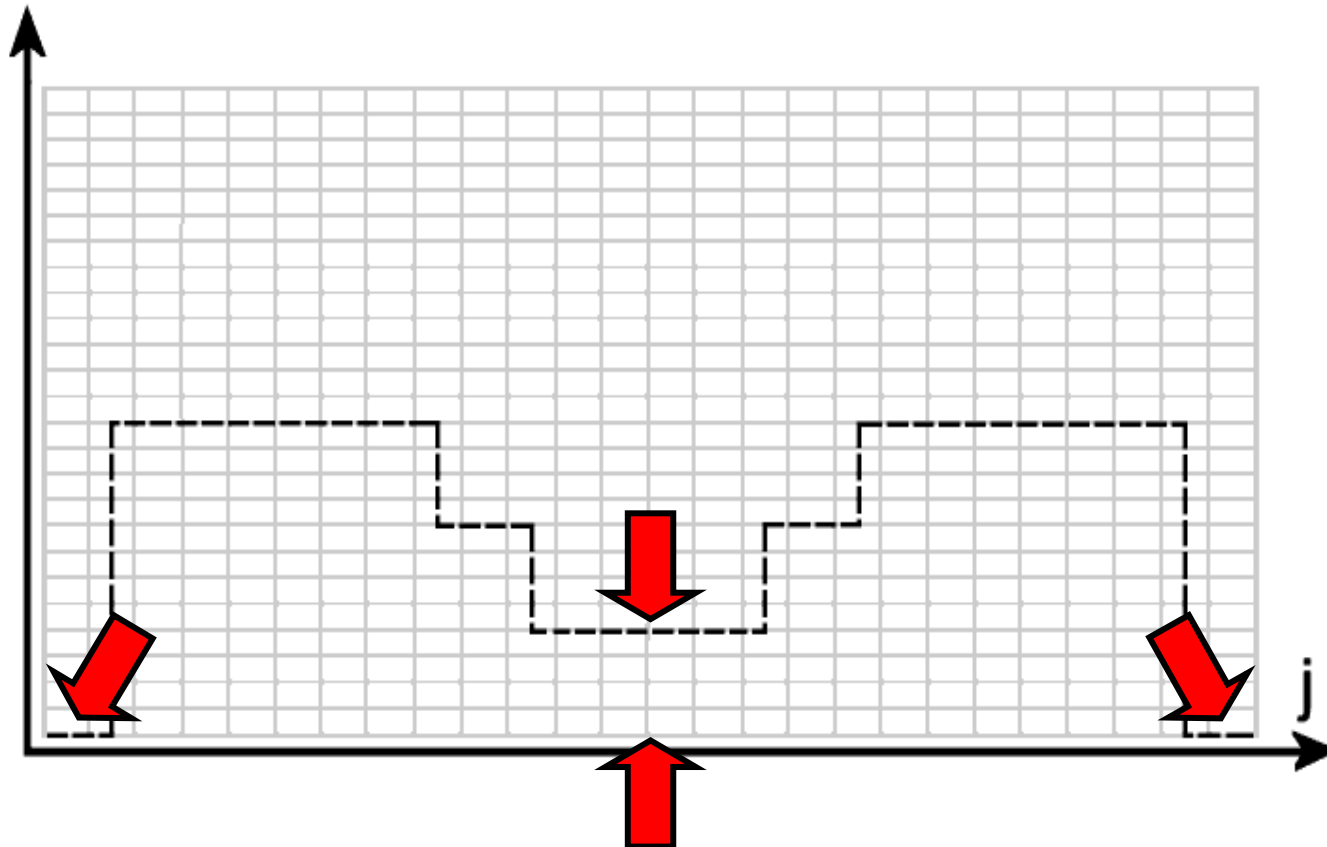
Bestimmung der Schnittmetrik:





Smart Partitioner

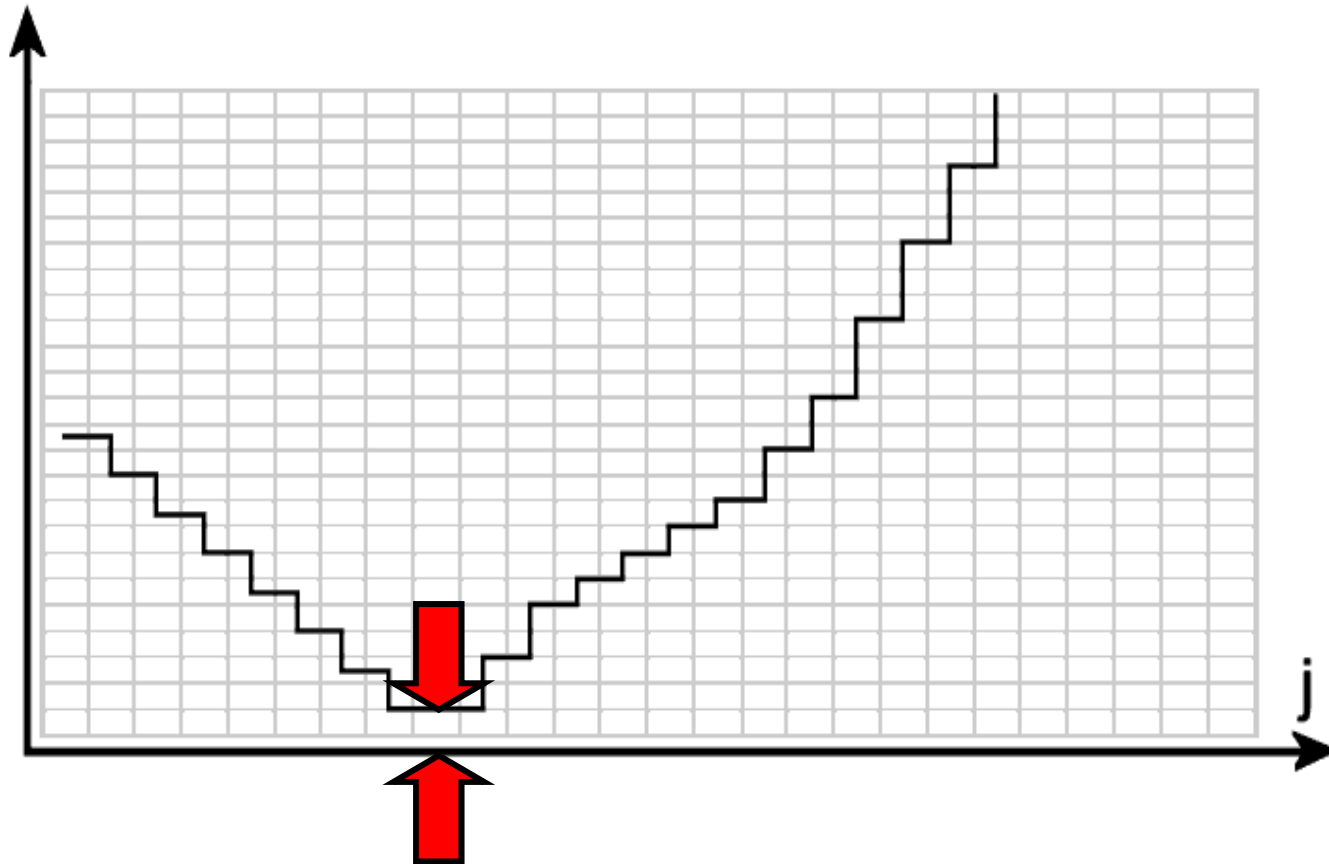
Kommunikationsmetrik:





Smart Partitioner

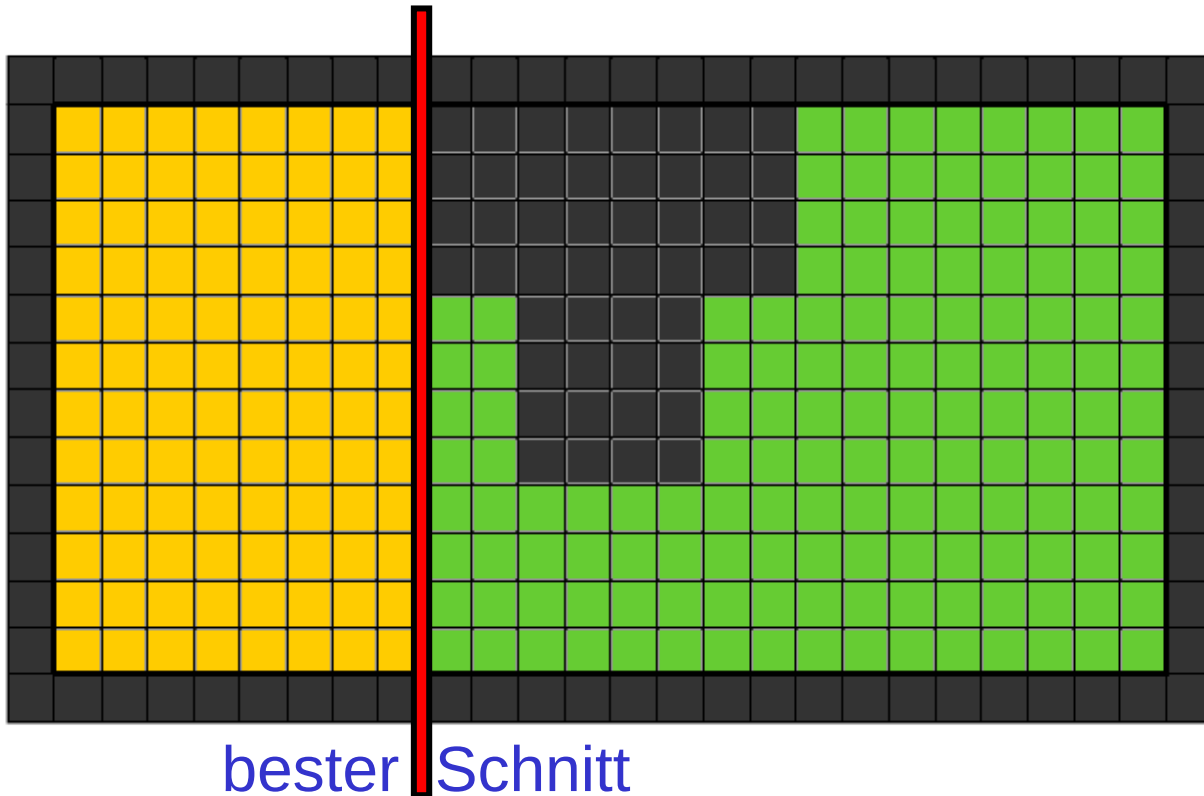
Last-Balance-Metrik:





Smart Partitioner

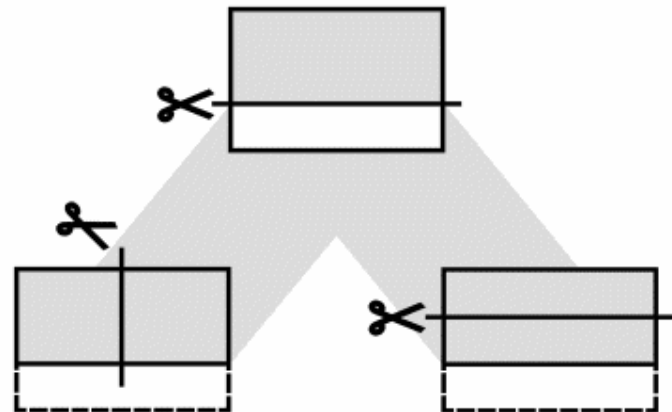
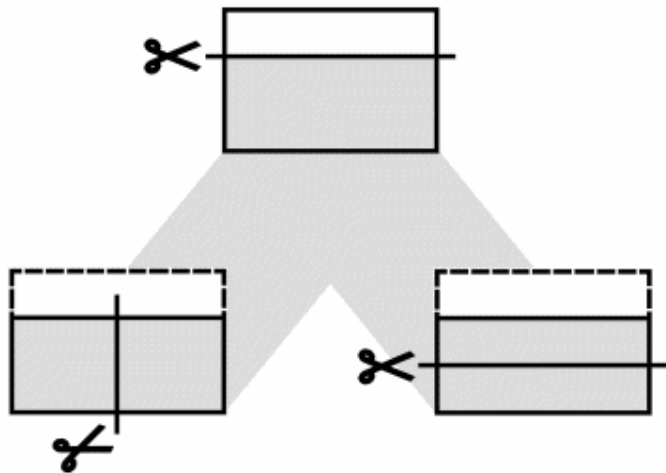
Superposition:





Smart Partitioner

Verteilungsmuster:

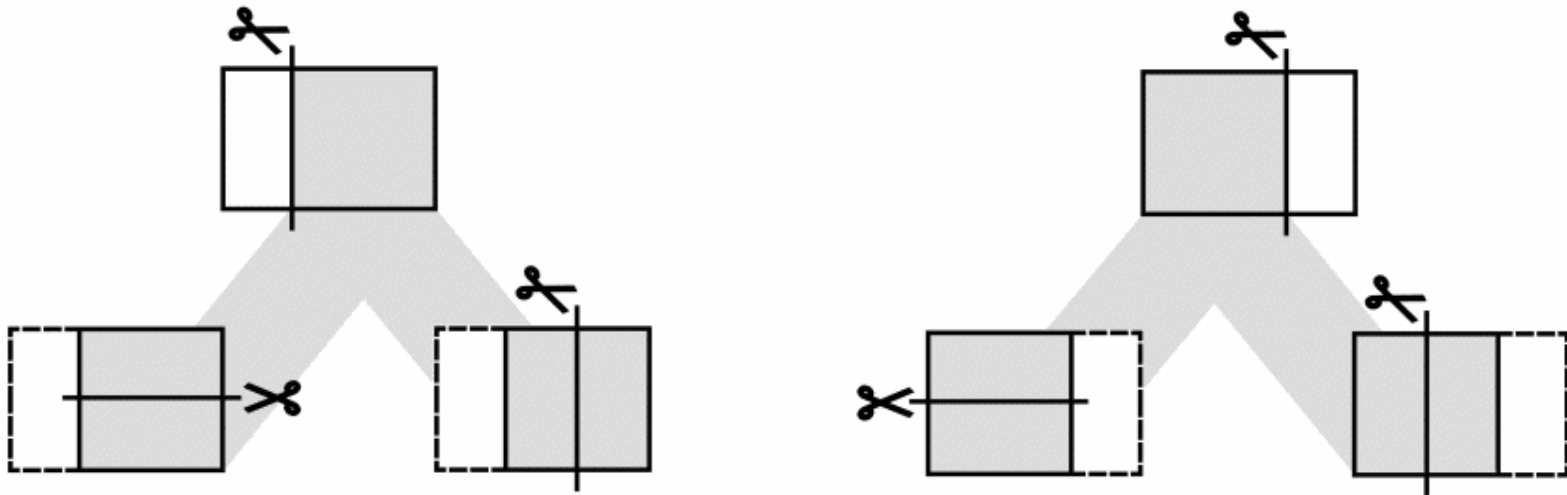


drei Metahosts / erster Schnitt: **horizontal**



Smart Partitioner

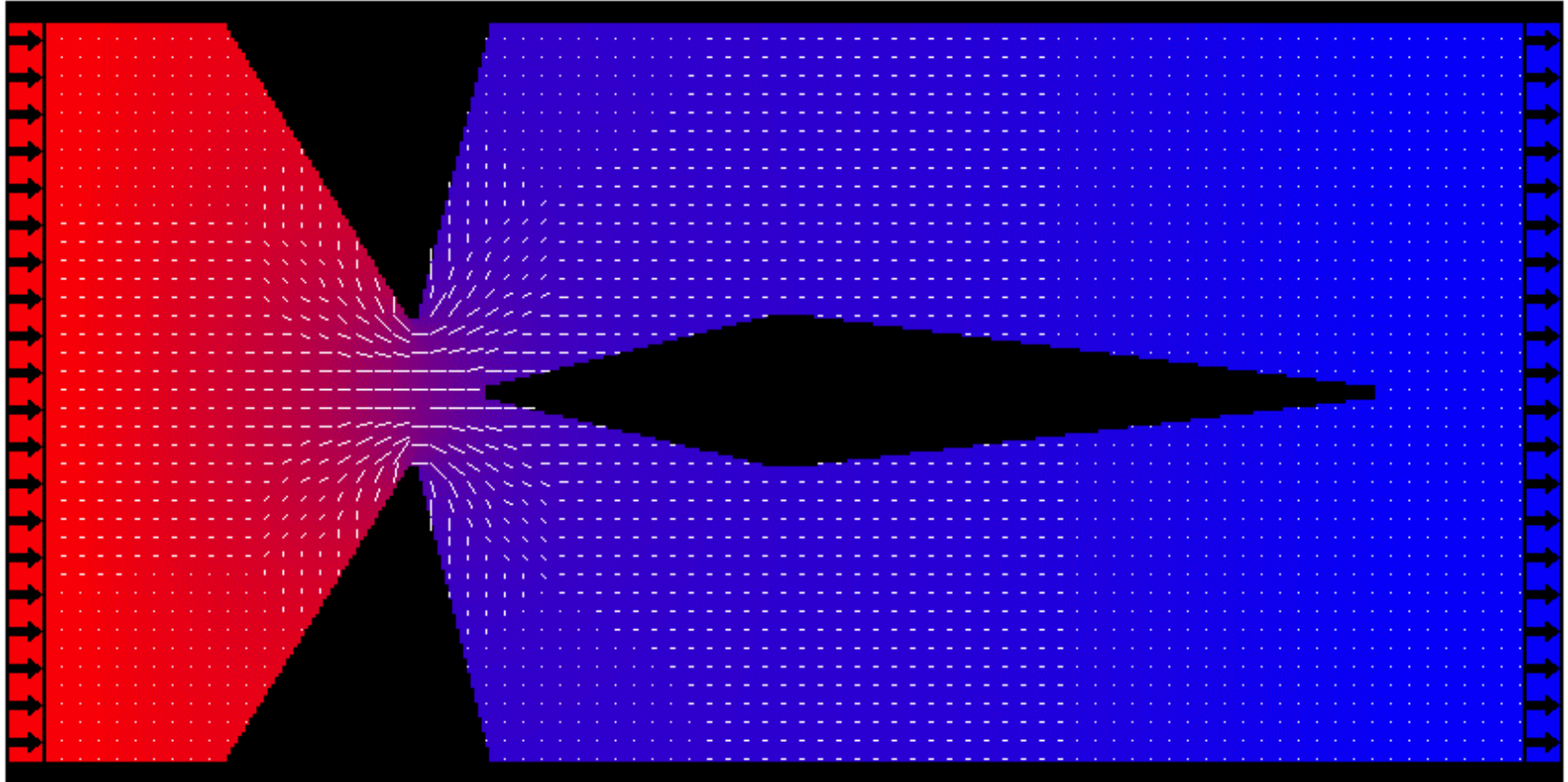
Verteilungsmuster:



drei Metahosts / erster Schnitt: vertikal

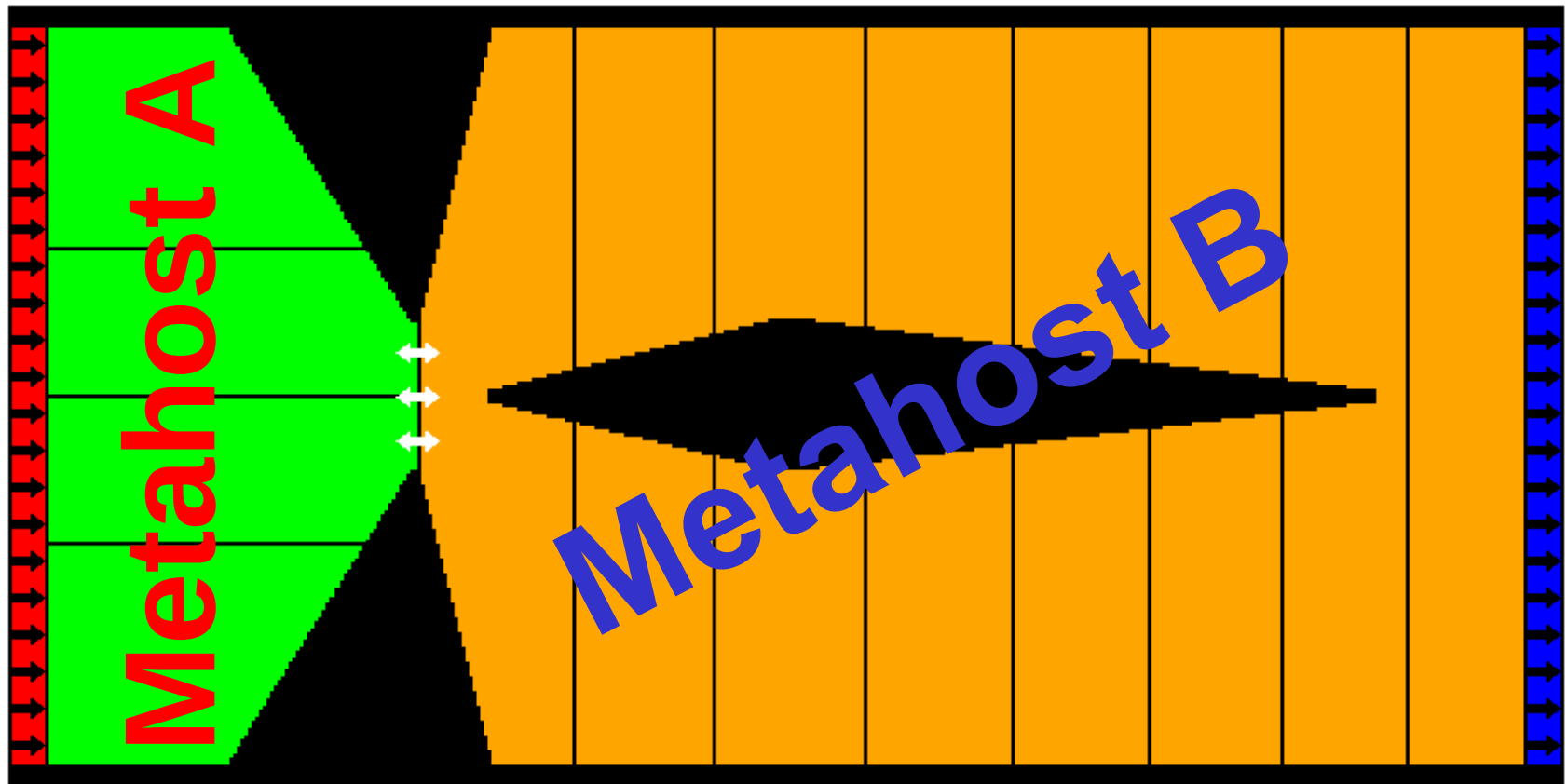


Beispiel: CFD Simulation



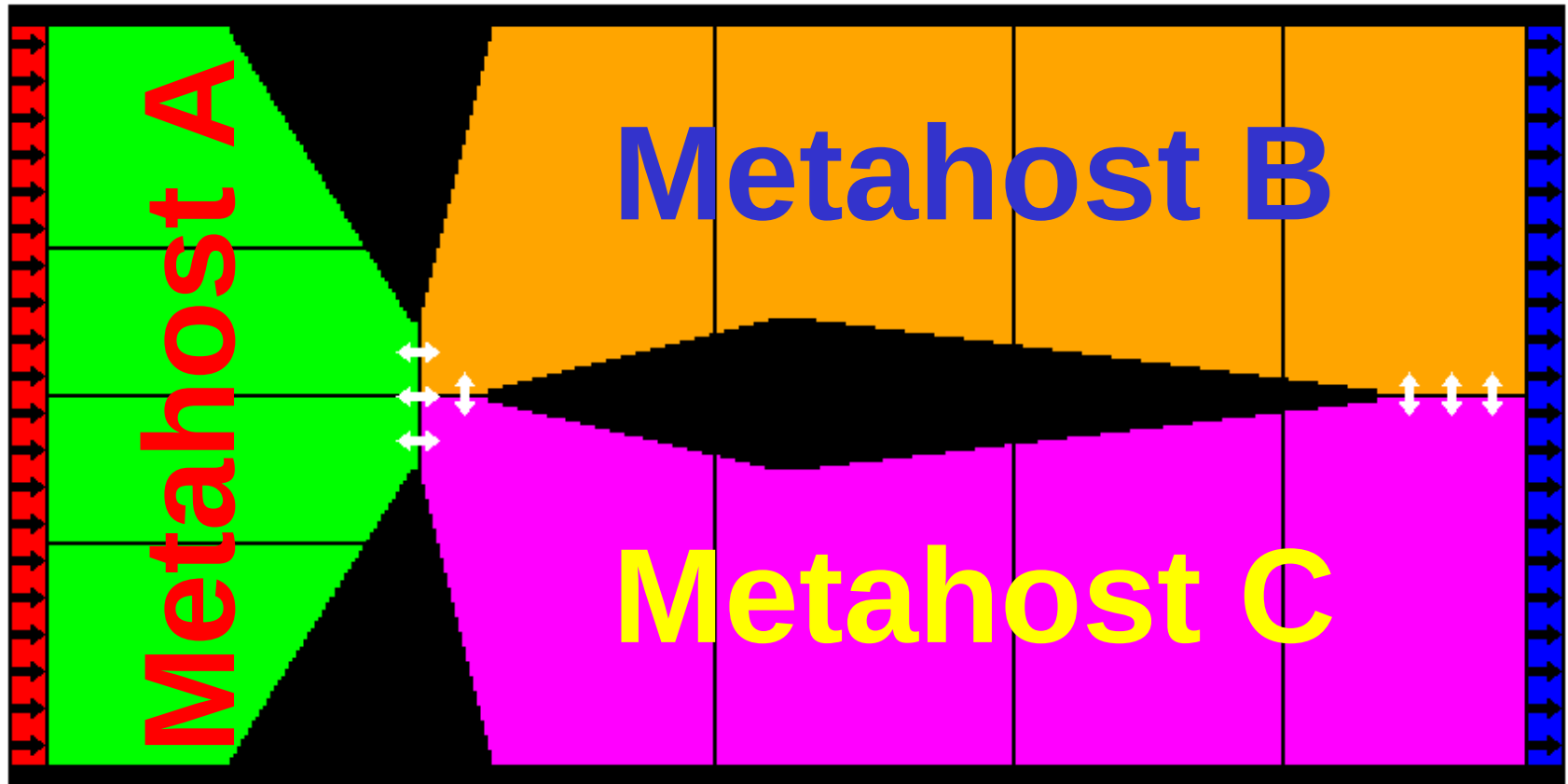


Beispiel: CFD Simulation





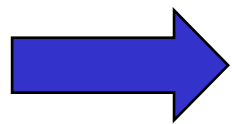
Beispiel: CFD Simulation





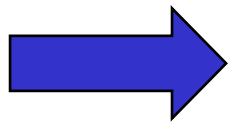
Kommunikations-Bibliothek

Smart Partitioner



Optimales Verteilungsschema für ein Problem auf einem Metacomputer

Einfache Umsetzung in Applikationen?



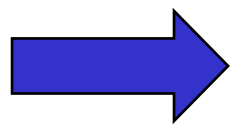
Kommunikations-Bibliothek basierend auf MetaMPICH

(MetaComm) 



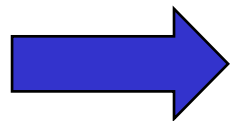
MetaComm

Funktionen zum Austausch von Randdaten:



Reduktion basiert auf einer
intelligenten Gebietsverteilung

Optimierte kollektive Operationen:

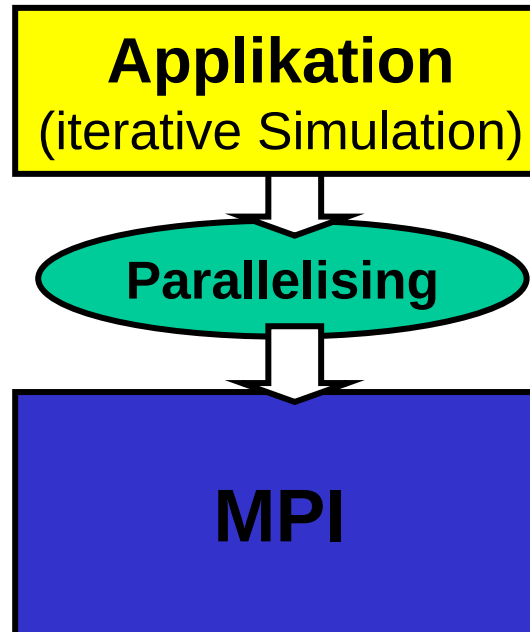


Ausnutzung **lokaler** Kommunikation
Bsp: **Meta_Allreduce()**



Kommunikations-Bibliothek

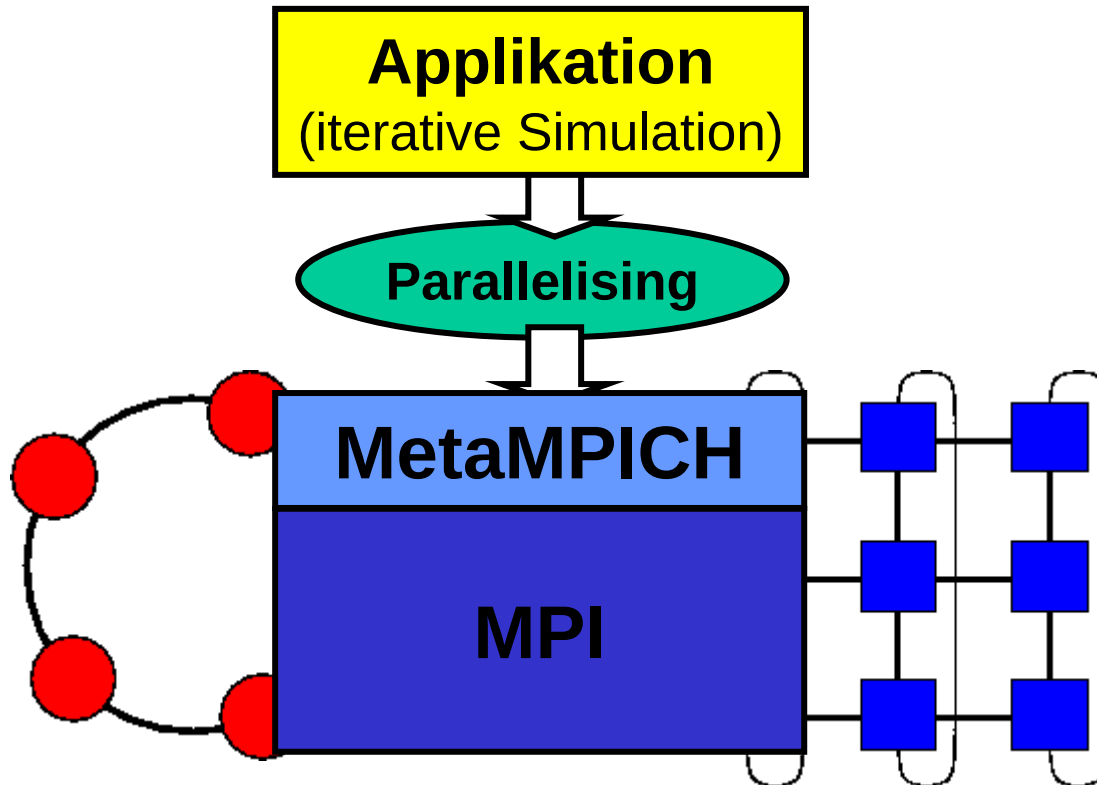
normaler Fall:





Kommunikations-Bibliothek

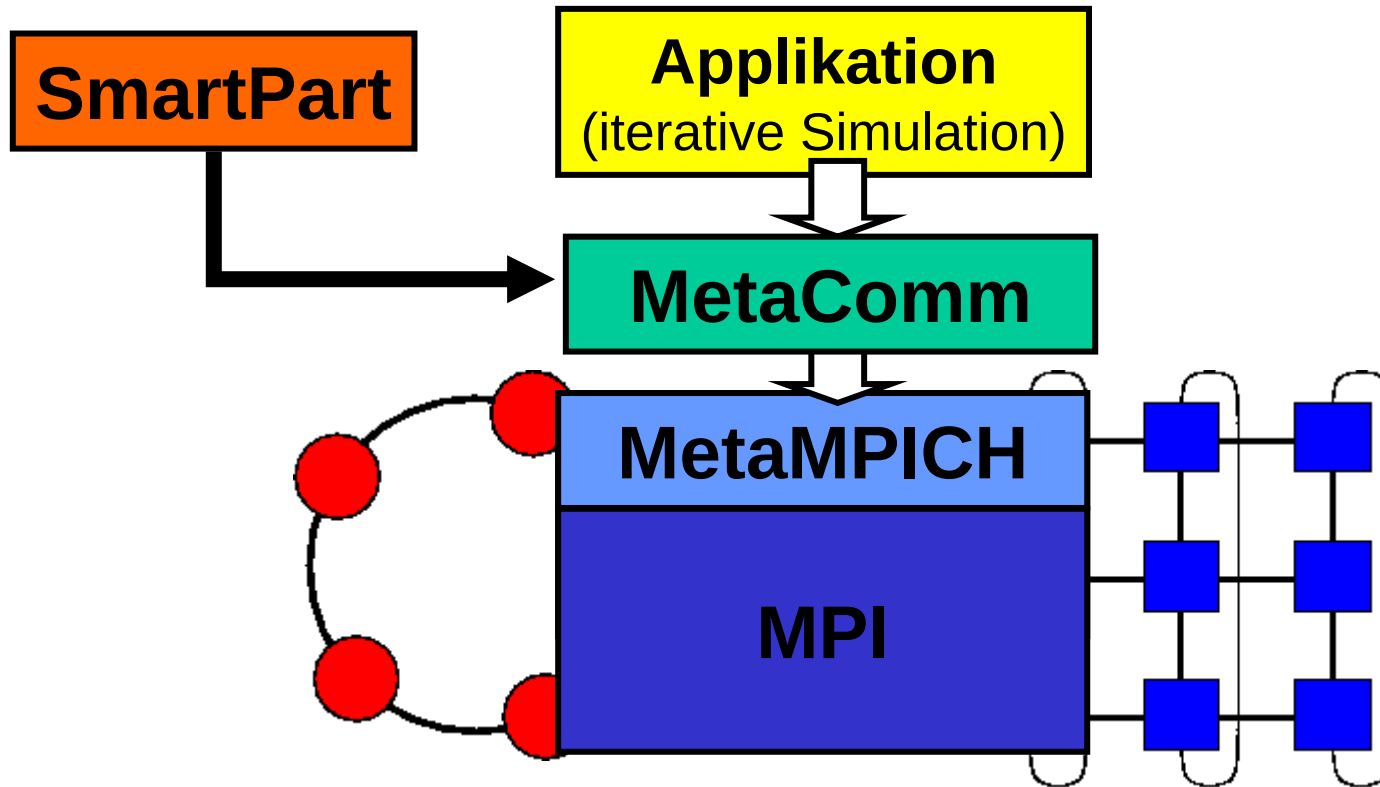
Metacomputer:





Kommunikations-Bibliothek

Metacomputer:



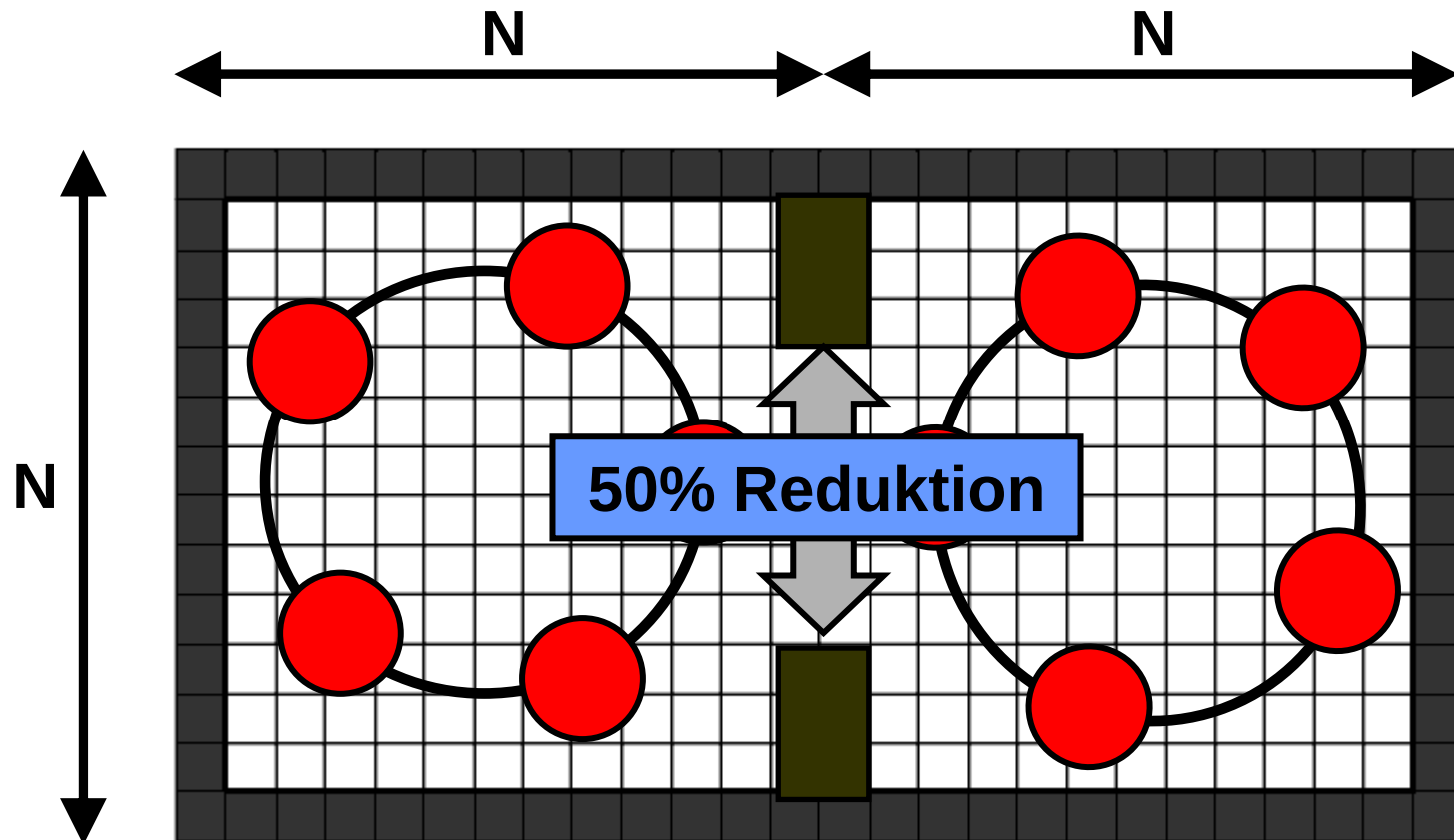


Metacomputing?

Wann lohnt sich Metacomputing?

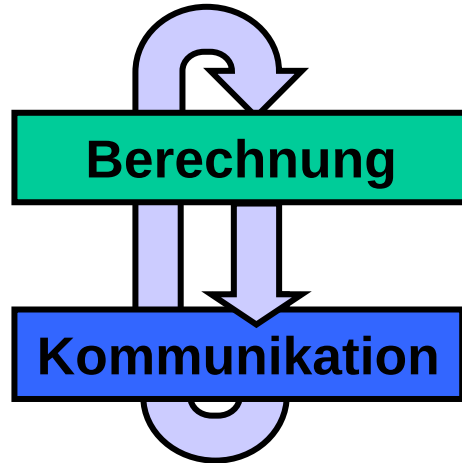


Metacomputing?





Metacomputing?



$$T_{\text{calc}}(N) = N^2 t_{\text{calc}}$$

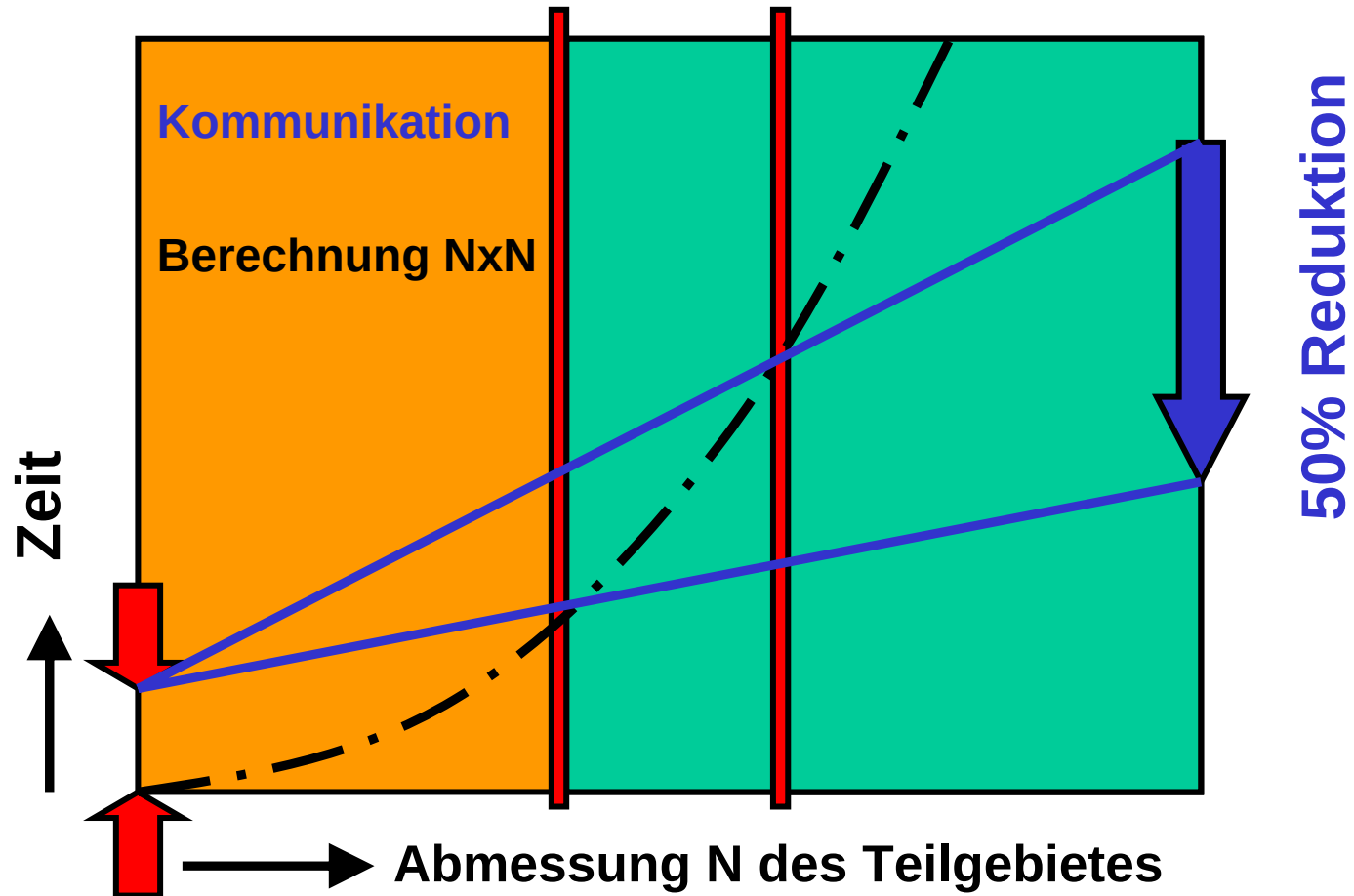
$$T_{\text{comm}}(N) = t_{\text{startup}} + \alpha N t_{\text{comm}}$$

Speedup:

$$S = \frac{T(1 \text{ Metahost})}{T(2 \text{ Metahosts})} = \frac{T_{\text{calc}}(2 N)}{T_{\text{calc}}(N) + T_{\text{comm}}(N)}$$

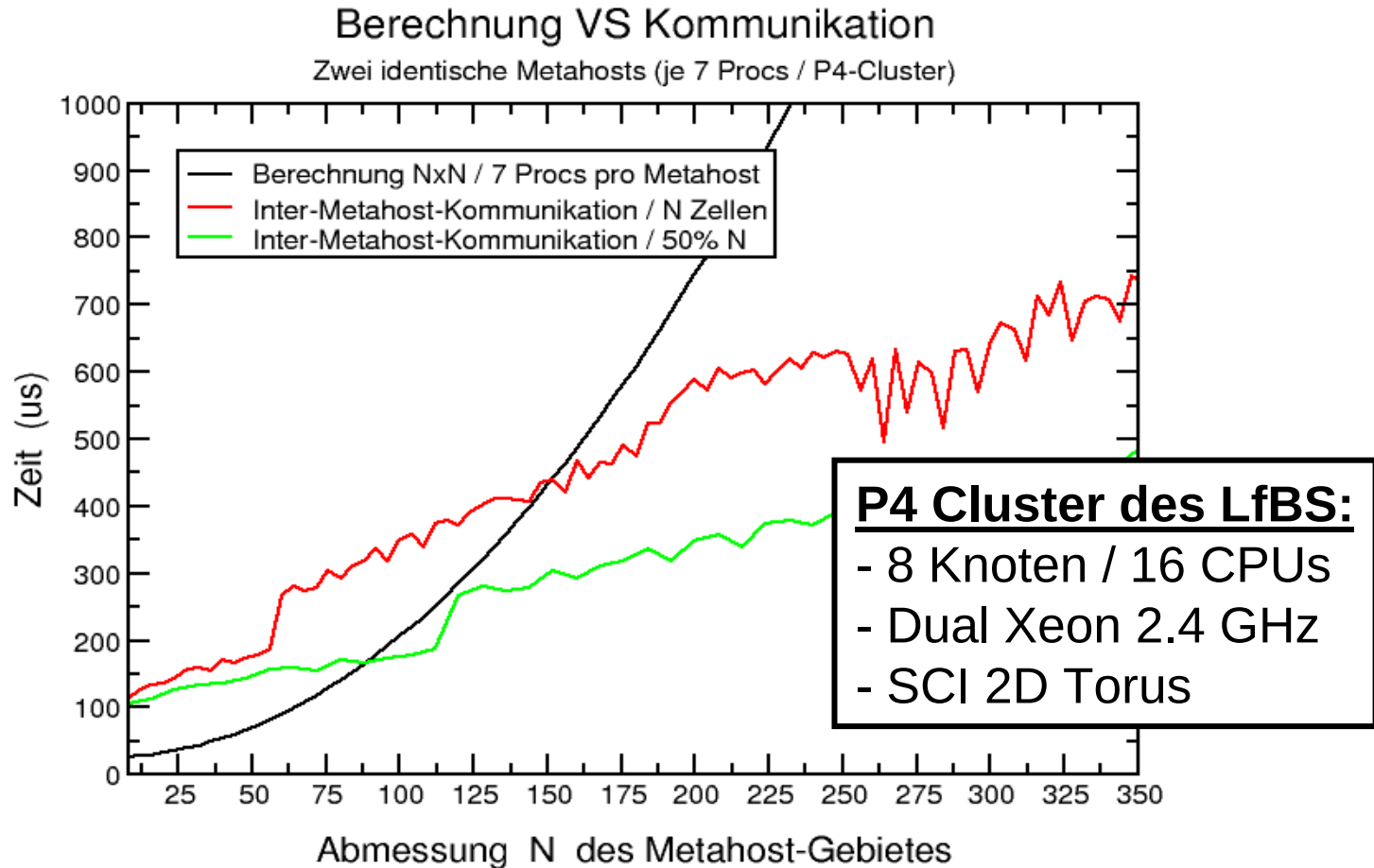


Metacomputing?



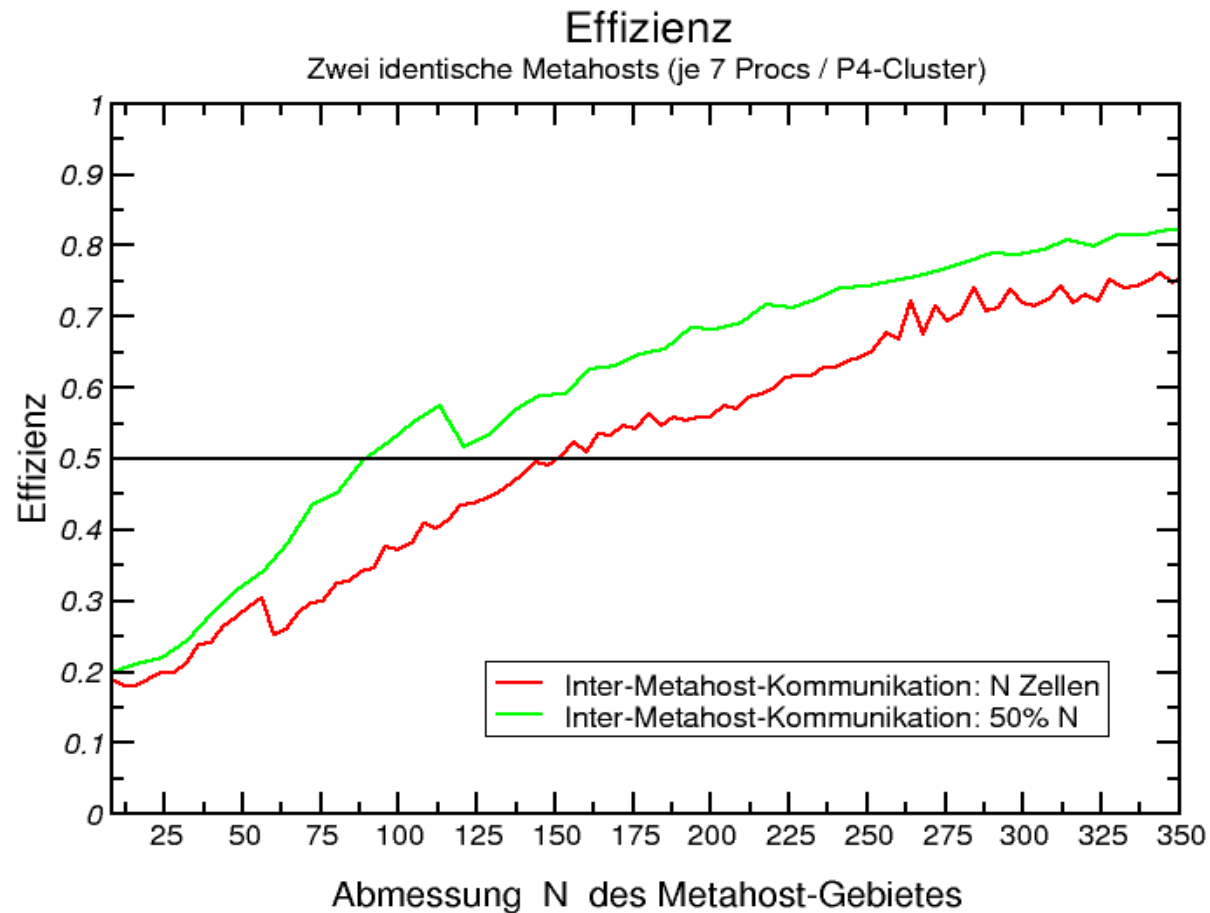


Metacomputing?





Metacomputing?

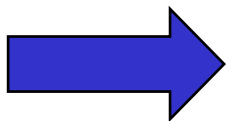




Metacomputing?

Metacomputing *kann* sinnvoll sein...

Ein Metacomputer ist immer ein
heterogenes System !!!



Anpassung der Applikationen
in der Regel unerlässlich !!!

**Vielen Dank für Ihre
Aufmerksamkeit!**



Lehrstuhl für Betriebssysteme
RWTH Aachen

