# Schnelles Netzwerken

## Wie bekomme ich Informationen von A nach B

Lars Thoms

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

2018-02-01

Betreuer: Dr. Michael Kuhn

# Fahrplan

# HPC

## Definition

*engl.: high-performance computing* für Hochleistungsrechnen
Berechnung eines komplexen Problems, das nicht mehr durch herkömmliche Computer gelöst werden kann.
Dafür muss das Problem parallelisiert werden, damit es auf einem verteilten Cluster berechnet werden kann.

# Probleme des HPC

- Programme müssen parallelisiert werden
  - Lässt sich das Problem sinnvoll partitionieren?
  - Ist eine vollständige Sicht auf die Daten essentiell?
  - Überschneiden sich Zuständigkeitsbereiche?
  - Muss (viel) kommuniziert werden?

# Probleme des HPC

- Programme müssen parallelisiert werden
  - Lässt sich das Problem sinnvoll partitionieren?
  - Ist eine vollständige Sicht auf die Daten essentiell?
  - Überschneiden sich Zuständigkeitsbereiche?
  - Muss (viel) kommuniziert werden?
- Cluster müssen nicht immer in einem Raum stehen
  - Verbund mit mehreren Universitäten möglich
  - Was für eine Verbindung gibt es zwischen den Rechenzentren?

# Kriterien einer kommunikativen HPC Anwendung

- Häufigkeit (Intervall)

# Kriterien einer kommunikativen HPC Anwendung

- Häufigkeit (Intervall)
- Regelmäßigkeit vorhanden?

# Kriterien einer kommunikativen HPC Anwendung

- Häufigkeit (Intervall)
- Regelmäßigkeit vorhanden?
- Hohe Latenz ein Problem?

# Kriterien einer kommunikativen HPC Anwendung

- Häufigkeit (Intervall)
- Regelmäßigkeit vorhanden?
- Hohe Latenz ein Problem?
- Synchronizität wichtig? (Wartezeiten)

# Kriterien einer kommunikativen HPC Anwendung

- Häufigkeit (Intervall)
- Regelmäßigkeit vorhanden?
- Hohe Latenz ein Problem?
- Synchronizität wichtig? (Wartezeiten)
- Menge der übermittelnden Daten

# Einfache Netztopologien

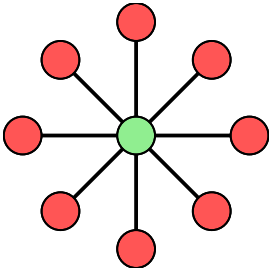

Abbildung: Star *(Stern)*

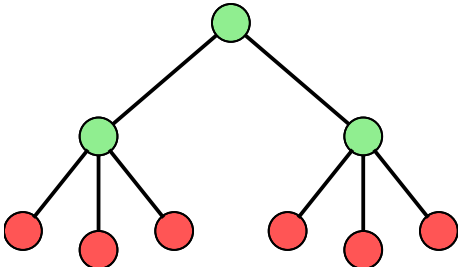

Abbildung: Tree *(Baum)*

Rote Knoten sind Rechner – grüne sind Switches

# Klassische HPC Netztopologie

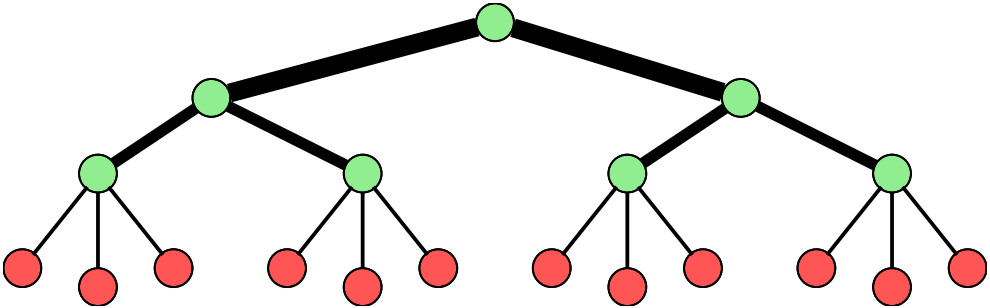

Abbildung: FatTree

Die Bandbreite zum Coreswitch (oberste Knoten) nimmt zu

# Besondere HPC Netztopologien (1/2)

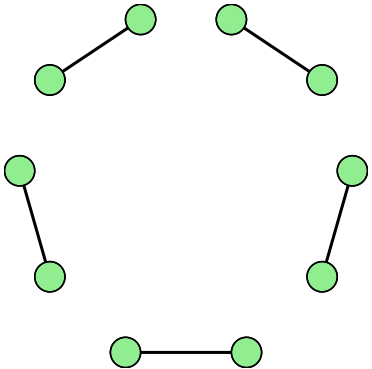

Abbildung: Dragonfly

# Besondere HPC Netztopologien (1/2)

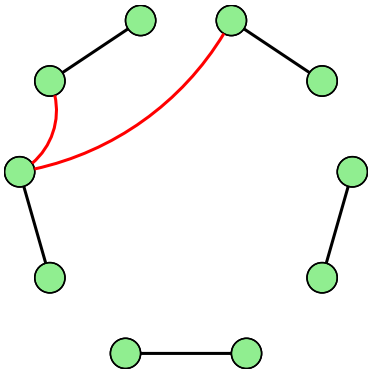

Abbildung: Dragonfly

# Besondere HPC Netztopologien (1/2)

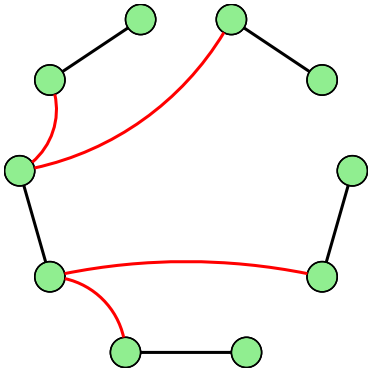

Abbildung: Dragonfly

# Besondere HPC Netztopologien (1/2)

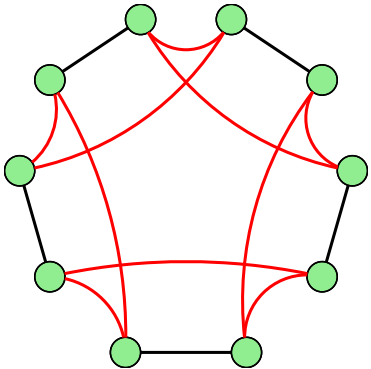

Abbildung: Dragonfly

# Besondere HPC Netztopologien (2/2)

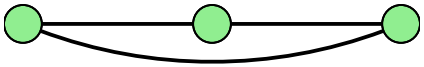

Abbildung: 1D-Torus

# Besondere HPC Netztopologien (2/2)

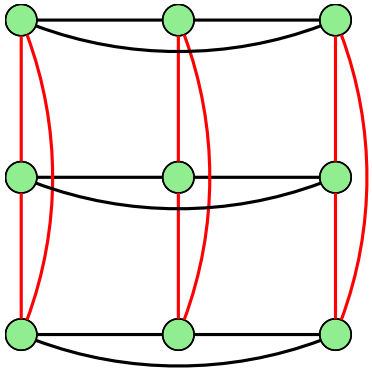

Abbildung: 2D-Torus

# Besondere HPC Netztopologien (2/2)

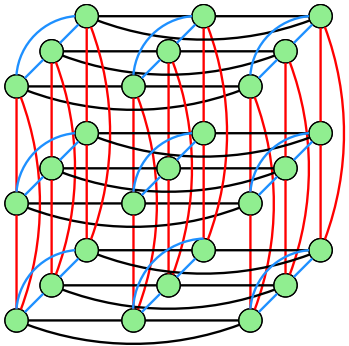

Abbildung: 3D-Torus

# Nutzungsarten von Netzwerken

- Daten-/Speicherzugriff (SAN)
  - Niedrige Latenz und
  - Hochverfügbarkeit notwendig?
  - Hohe Bandbreite ist sehr sinnvoll!

# Nutzungsarten von Netzwerken

- Daten-/Speicherzugriff (SAN)
  - Niedrige Latenz und
  - Hochverfügbarkeit notwendig?
  - Hohe Bandbreite ist sehr sinnvoll!
- Managementzugriff (SSH/IPMI/PXE)
  - Niedrige Latenz,
  - Hochverfügbarkeit und
  - hohe Bandbreite nicht notwendig

# Nutzungsarten von Netzwerken

- Daten-/Speicherzugriff (SAN)
  - Niedrige Latenz und
  - Hochverfügbarkeit notwendig?
  - Hohe Bandbreite ist sehr sinnvoll!
- Managementzugriff (SSH/IPMI/PXE)
  - Niedrige Latenz,
  - Hochverfügbarkeit und
  - hohe Bandbreite nicht notwendig
- Kommunikation während Berechnungen zw. Knoten
  - Sehr niedrige Latenz, um Wartezeiten zu vermeiden!
  - Hochverfügbarkeit!
  - Hohe Bandbreite notwendig?

# Was gibt es an Hardware?

- Ethernet
- Myrinet
- InfiniBand
- Intel OmniPath

# Was gibt es an Hardware?

- Ethernet
- InfiniBand
- Intel OmniPath

# Kerndaten von Ethernet

- Übertragungsweg sowohl elektrisch via Kupferkabel oder optisch via Glasfaser
- Bandbreiten: $1\,\mathrm{Gbit\,s^{-1}}$, $10\,\mathrm{Gbit\,s^{-1}}$, $40\,\mathrm{Gbit\,s^{-1}}$ und $100\,\mathrm{Gbit\,s^{-1}}$
- $200\,\mathrm{Gbit\,s^{-1}}$ und $400\,\mathrm{Gbit\,s^{-1}}$ für optische Übertragung spezifiziert

# Schichten, Schichten, Schichten ...

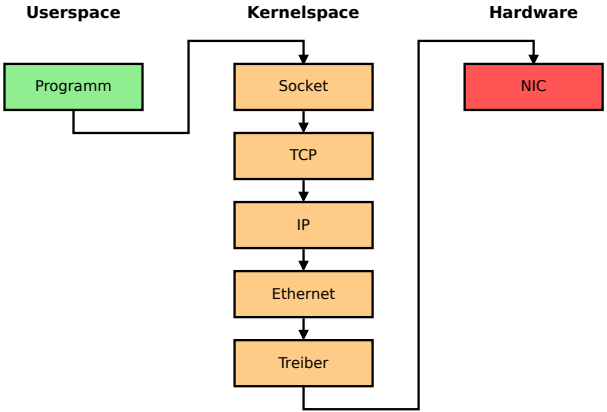


Abbildung: Netzwerkstack

# Puffer, wir brauchen mehr Puffer!

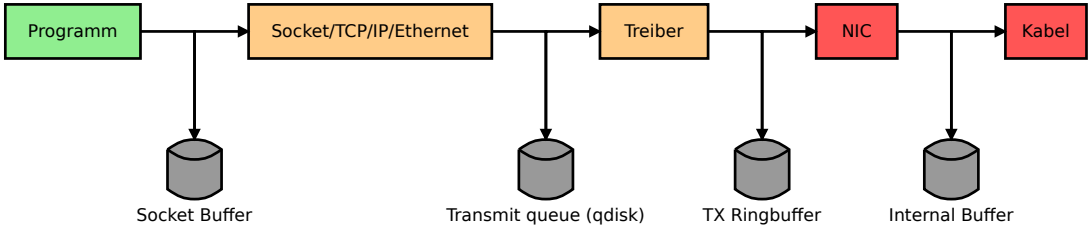


Abbildung: Standard Puffersystem im Linux Netzwerkstack

# Zero-Copy Verfahren

- Netzwerkpuffer werden nicht herum kopiert

# Zero-Copy Verfahren

- Netzwerkpuffer werden nicht herum kopiert
- Bei modernen Maschinen: Vectored I/O *(Scatter Gather)*
  - Netzwerkhardware unterstützt DMA *(Direct Memory Access)*
  - Der Vektor fungiert als Pufferdeskriptor
  - Bestehend aus: Datenpointer, Länge und den nächsten Deskriptor

# Zero-Copy Verfahren

- Netzwerkpuffer werden nicht herum kopiert
- Bei modernen Maschinen: Vectored I/O *(Scatter Gather)*
  - Netzwerkhardware unterstützt DMA *(Direct Memory Access)*
  - Der Vektor fungiert als Pufferdeskriptor
  - Bestehend aus: Datenpointer, Länge und den nächsten Deskriptor
- Nutzdaten und Header (TCP/IP- und Ethernetframes) können getrennt werden

# Kernel Bypass

- Zero-Copy wird unter Umständen genutzt
- Es gibt verschiedene Techniken und Frameworks
  - `PACKET_MMAP`
  - `PF_RING`
  - Snabbswitch
  - DPDK
  - Netmap

# Kernel Bypass

- Zero-Copy wird unter Umständen genutzt
- Es gibt verschiedene Techniken und Frameworks
  - `PACKET_MMAP`
  - `PF_RING`
  - Snabbswitch
  - DPDK
  - Netmap
- Mehr bei Cloudflare: `https://blog.cloudflare.com/kernel-bypass/`

# Kerndaten InfiniBand

- Offener Standard
- Übertragungsmedium ist ebenfalls elektrisch via Kupfer oder optisch mit Glasfasern
- Latenzarm: selten über 3 µs
- Der HCA *(Host Channel Adapter)* wird über PCIe angesteckt
- Bandbreiten: 10 Gbit $s^{-1}$, 25 Gbit $s^{-1}$, 40 Gbit $s^{-1}$, 50 Gbit $s^{-1}$, 56 Gbit $s^{-1}$, 100 Gbit $s^{-1}$ und 200 Gbit $s^{-1}$

# RDMA

- Verringerung der Latenz durch *Remote Direct Memory Access*
- Daten werden vom eigenen in den Zielarbeitsspeicher geschrieben

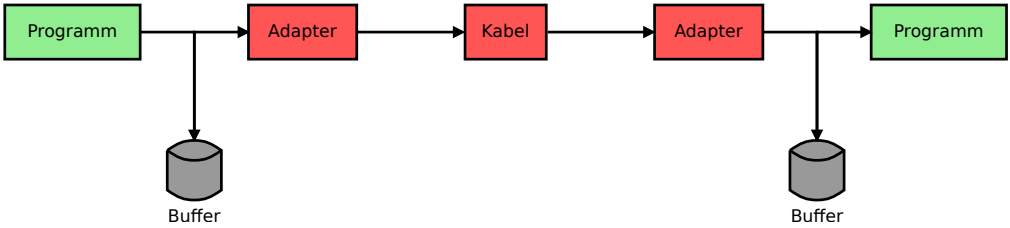


Abbildung: Remote Direct Memory Access bei Infiniband

# Kerndaten Intel OmniPath

- Proprietäre Lösung von Intel
- Benutzt optische Übertragungsmedien
- Bandbreite von $100\,\mathrm{Gbit\,s^{-1}}$
- Protokoll ähnlich zu InfiniBand

# Unterschied zu InfiniBand

- *Forward Error Correction* wird simplifiziert durch einen einfachen 14 bit CRC
- Schneller, aber im Fehlerfall Wiederholung der Transaktion

# Unterschied zu InfiniBand

- *Forward Error Correction* wird simplifiziert durch einen einfachen 14 bit CRC
- Schneller, aber im Fehlerfall Wiederholung der Transaktion
- *Traffic Flow Optimization*: Mehrere Flits *(Flow Control Units)* werden in eine Transaktion unifiziert

# Unterschied zu InfiniBand

- *Forward Error Correction* wird simplifiziert durch einen einfachen 14 bit CRC
- Schneller, aber im Fehlerfall Wiederholung der Transaktion
- *Traffic Flow Optimization*: Mehrere Flits *(Flow Control Units)* werden in eine Transaktion unifiziert
- Priorisierte Transaktionen können große Transaktionen unterbrechen

# Was haben wir heute gelernt?

- Wir müssen wissen, was das Programm macht!
- Dadurch entscheidet sich
  - die Topologie,
  - das Übertragungsprotkoll und
  - die eingesetzte Hardware.