



TECHNISCHE UNIVERSITÄT  
CHEMNITZ

Fakultät für Informatik  
Professur Praktische Informatik

# Proseminar „Parallele Programmierung“ BSP-Programmiermodell

Danny Hofmann

22.11.2016

# Parallel Computing

- Moderne Probleme Lösen durch Verteilung auf viele CPUs
- Pixar benutzte 1999 zum Rendern von ToyStory 2 einen Parallelrechner mit rund 1400 Prozessoren



# Paralleles Programmieren

- Wieso benötigt man parallele Ansätze in der Informatik?
  - Single Processor Usage: Existieren physikalisch /technische Grenzen  
→ Leistungsverlust
  - Parallel Processor Usage: Leistungssteigerung!  
→ Problem: Synchronisation und Nutzung Gemeinsamer Daten
- Was wurde bisher erforscht?
  - Hardware → Entwerfen von geeigneten Plattformen für parallele Modelle  
→ **Gelöst**
  - Programmiermodelle: → Entwerfen von Programmiermodellen  
→ Seit den letzten gut 30 Jahren immer wieder Mittelpunkt von Forschungsarbeiten

# Programmiermodelle

## Wieso?

- Modell zur Abstrahierung von der Hardware
  - CPUs verarbeiten Befehle schnell
  - Wenig Befehle für den Programmierer, viele für die CPU
- Sequentiell: Loops
- Parallel: Konzept welches Last auf alle Prozessoren verteilt

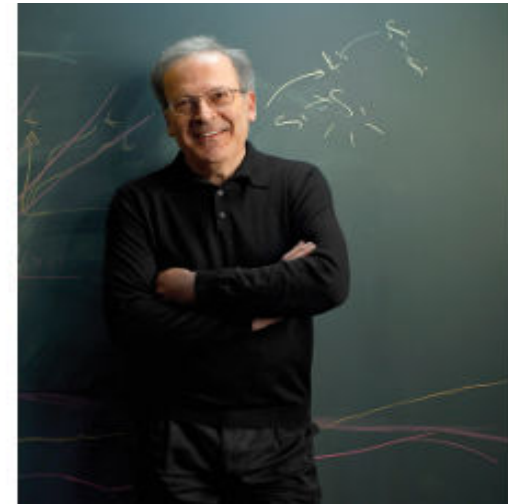
# Programmiermodelle

## Anforderung

- Übertragbarkeit
- Effizienz
- Einfachheit
- Berechenbare Performance

# BSP - Bulk Synchronous Parallel

- Leslie Valiant (\* 28. März 1949)
- Britischer Informatiker
- Entwickelt um 1980er Jahren den MIMD Ansatz BSP
- Veröffentlichte seine Arbeit 1990
- 1997 McColl Oxford BSPlib Standard

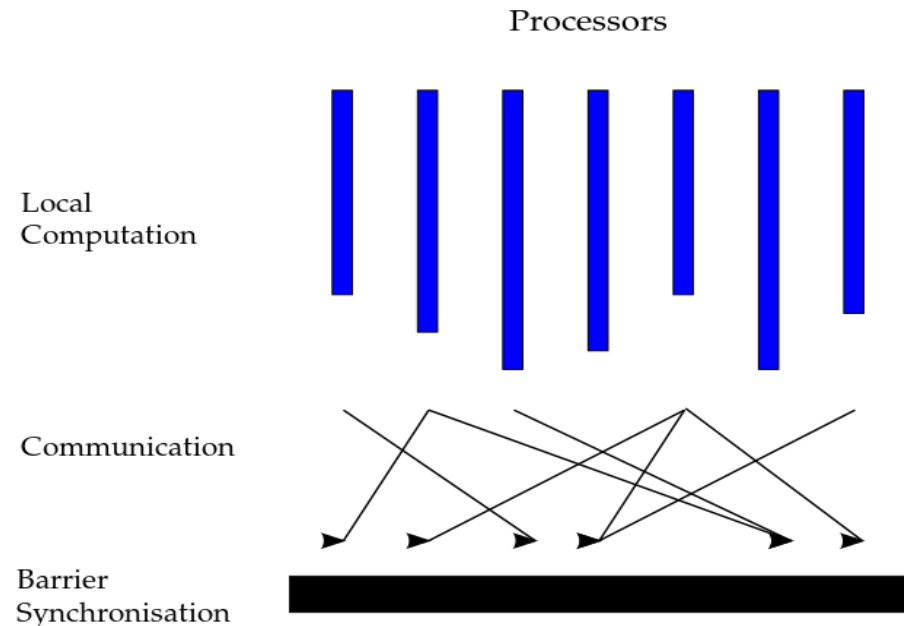


# Aufbau BSP abstrakter Computer

- Prozessoren und deren Speicher
- Netzwerk zur Nachrichten Übermittlung
- Einheiten zur Globalen Synchronisation mit Hilfe einer „Barrier“

# BSP Algorithmen

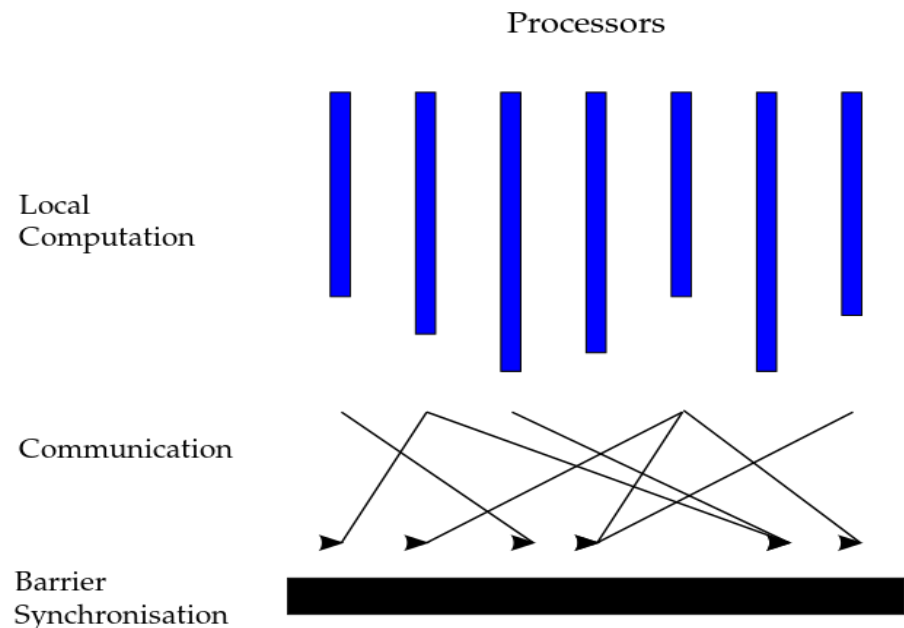
- Bestehen aus mehreren Supersteps
  - Jeder Prozessor für sich
  - Eigene Lokale Daten
- Local Computation
  - Jeder Prozessor für sich
  - Eigene Lokale Daten





# Kommunikation und Synchronisation

- Unabhängig von einander
  - Löst message passing Problem  
→ Überlastung
  - Barrier  
→ Setzt Globale Daten



# BSP Optimierungen

- Optimierungen der Performance zielen auf das Programm.
- Im Fokus Kommunikation
  - Message packing
  - Destination Scheduling

# Message Packing

- BSP setzt voraus:
    - Kosten der Übertragung 1000 1-byte Nachrichten == 1 1000-byte Nachricht
  - BSP „packt“ Nachrichten zusammen
- + Reduziert Overhead

# Destination Scheduling

- Neuordnung der Sendereihenfolge
  - via Zufälligen Auswahlverfahren
- Verhindern von busy-wait, Kollision von Nachrichten  
+Performance Verbesserung

# Anwendung von BSP

- Cluster
  - Zusammenschluss vieler Rechneinheiten
  - Möglichkeit eines großen BSP Systems
- Komplexe Geometrische Aufgaben
- Molekulardynamische Simulationen
- Simulation für Elektromagnetische Felder

# Zusammenfassung

- Struktur
- Netzwerk Optimierungen
- Berechenbare Performance
- Anwendungsmöglichkeiten

## Quellen

- Fotos:  
Leslie Vailant <http://cacm.acm.org/magazines/2011/6/108643-beauty-and-elegance/fulltext>  
Superstep <https://upload.wikimedia.org/wikipedia/en/thumb/e/ee/Bsp.wiki.fig1.svg/768px-Bsp.wiki.fig1.svg.png>  
<http://harlemtoys.com/wp-content/uploads/2015/10/toy-story-1-full-movie.jpg>

### Literatur:

R.Correa et al (eds), Models for Parallel and Distributed Computation. Theory, Algorithmic Techniques and Applications 85-115  
<https://www.mcs.anl.gov/~itf/dbpp/text/node9.html>  
<http://www.cse.unt.edu/~tarau/teaching/parpro/papers/Bulk%20synchronous%20parallel.pdfh>  
[http://www.computingreviews.com/hottopic/hottopic\\_essay.cfm?htname=BSP](http://www.computingreviews.com/hottopic/hottopic_essay.cfm?htname=BSP)  
Mattson, Timothy G., Berna Massingill, and Beverly A Sanders. Patterns for Parallel Programming. Boston: Addison-Wesley, (2005):  
Porter, Tom, and Galyn Susman. "On Site: Creating Lifelike Characters in Pixar Movies". Commun. ACM, 43.1 (2000): 25.  
Gerbessiotis, Alexandras V., and Leslie G Valiant. Direct Bulk-synchronous Parallel Algorithms. Algorithm Theory – SWAT '92(1992): 1 - 18.

•

Vielen Dank für die Aufmerksamkeit