

# **Lastverwaltung in parallelen und verteilten Systemen**

**Prof. Dr. Thomas Ludwig**

Ruprecht-Karls-Universität Heidelberg

Institut für Informatik

Abteilung für parallele und verteilte Systeme

Email : [t.ludwig@computer.org](mailto:t.ludwig@computer.org)

# Problemstellung

## Dynamisch variierende Lastungleichheiten

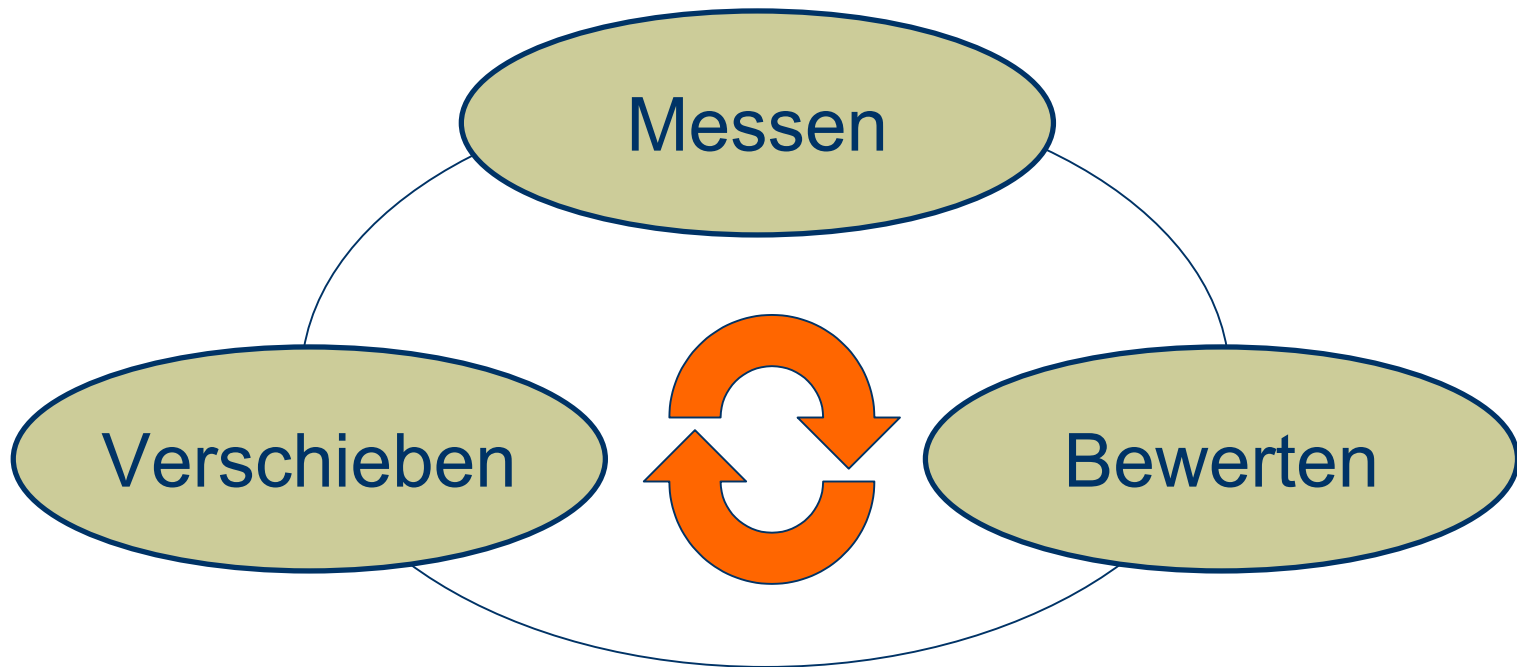
- Typisch für viele parallele und verteilte Anwendungen
- Statisch nicht ausregelbar; datenabhängig
- Reduziert Durchsatz/Auslastung des Systems

## Interessierende Ressourcen

- 99%: Prozessor
- Selten: Hauptspeicher, Netz, *Platten*

# Grundmechanismus

## Regelkreis Lastverwaltung



# Fragestellungen

- ◆ Integrationsort

Anwendungsintegriert vs. systemintegriert



SILR

- ◆ Lasterfassung

Schnell und rückwirkungsfrei

- ◆ Lastbewertung

Wie ist das Systemverhalten in der Zukunft?

- ◆ Lastverschiebung

Profitabel

# Lastbewertung

## Einfach

- Einzelnes Maß (CPU), einzelner Prozessor

## Komplex

- Systemkenngrößen, Anwendungskenngrößen
- Gewichtet mit neuronalem Netz
- Globale Übersicht
- Metawissen

Problem: beschreibt Vergangenheit

# Lastverschiebung

## Gegenstände der Verschiebung

- Einfach (anwendungsintegriert, feingranular)  
Datenpakete, Anfragen
- Komplex (systemintegriert, grobgranular)  
Prozesse, Objekte, Dateien

## Problematik

- Veränderung des Ausführungsortes
- Beibehaltung aller Querbeziehungen

# Beispiel Prozeßmigration

- Wie hält man den laufenden Prozeß an? Wie migriert man ihn? Was migriert man?
- Wie geht man mit offenen Dateien um?
- Was wird mit Signalen an den Prozeß?
  
- Was passiert mit Nachrichten während der Migration?
- Was passiert mit Nachrichten nach der Migration? Wie finden sich die Komm´partner?

# Systeme und Anwendungen

1987

2001



Parallelrechner  
dediziert

- Strömungs-  
mechanik

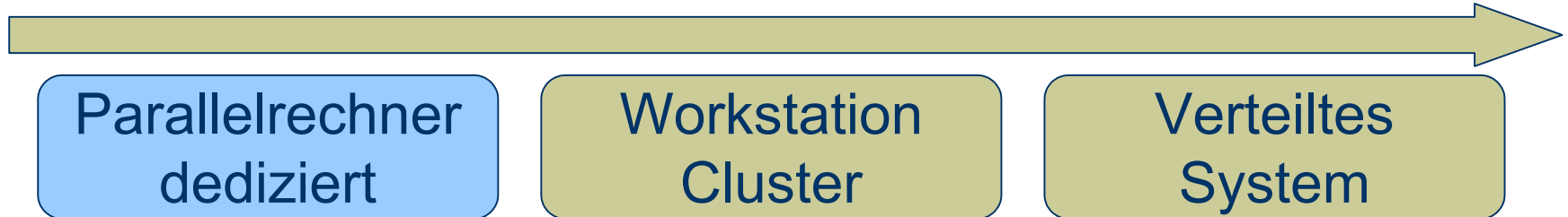
Workstation  
Cluster

- Computer-  
tomographie
- Crash-  
Simulation

Verteiltes  
System

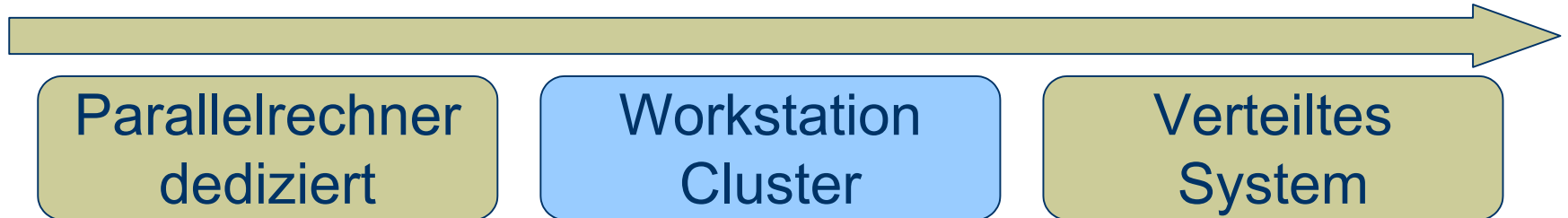
- Computer-  
tomographie
- Verkehrs-  
simulation

# Lastmessung und Lastverschiebung



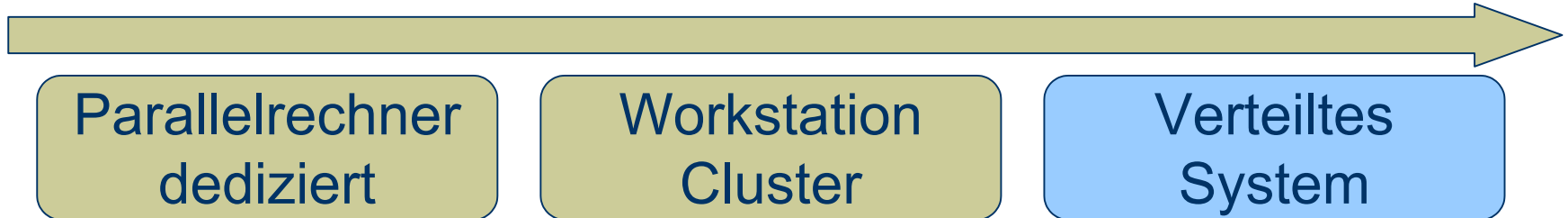
- ◆ Knotenorientierte Lastmessung
  - In das Betriebssystem integriert
- ◆ Prozeßmigration
  - Präemptiv
  - Seitenorientiert, Paging über das Netzwerk
  - In das Betriebssystem integriert

# Lastmessung und Lastverschiebung



- ◆ Node Status Reporter (NSR)
  - Konfigurierbarer Daemon-Prozeß
- ◆ Checkpoint-Mechanismus (CoCheck)
  - Prozeßverschiebung durch Verlagerung des Sicherungspunktes auf Zielknoten
  - Aufsatz auf das Betriebssystem

# Lastmessung und Lastverschiebung

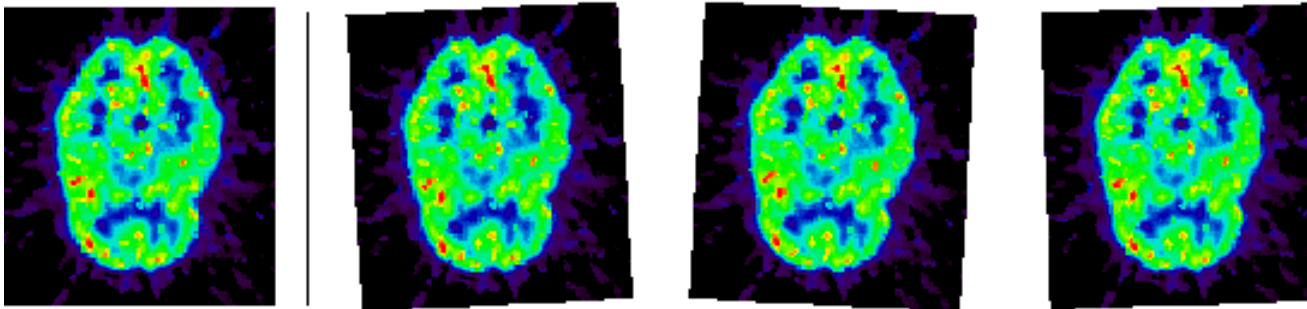


- ◆ Lastmessung mittels SNMP
- ◆ Lastverschiebung objektbasiert
  - Feinere Granularität
  - Objektplatzierung, -migration und –replikation
  - Integration in ORB und POA

# Beispiel Computertomographie

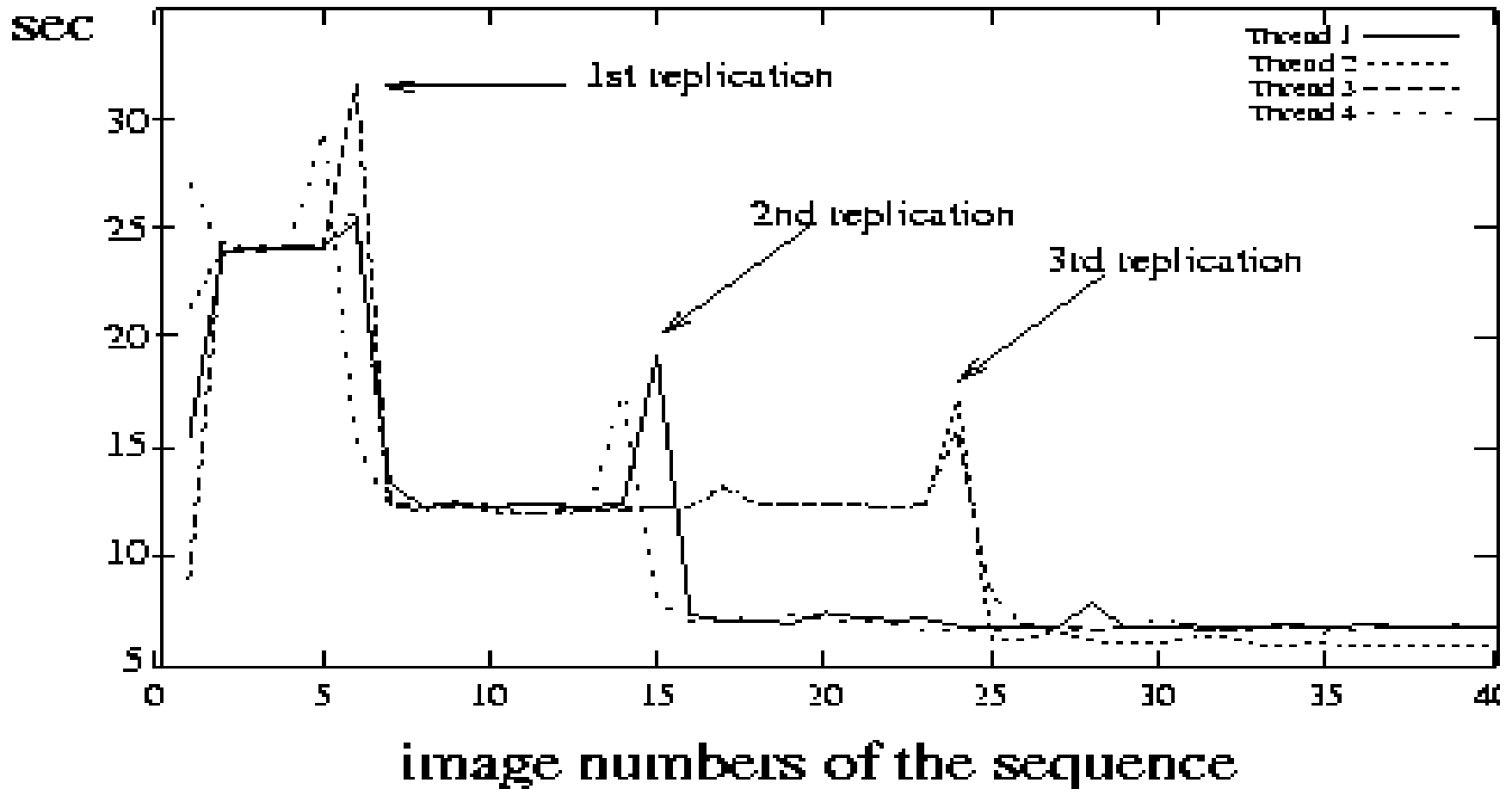
## Realignment (Ausrichtung)

- Gegeben: Bilderserie 1...n
- Problem: Patientenbewegungen



- Ansatz: Referenzdaten für erstes Bild
- Transformationsdaten für Bild 2...n

# Beispiel Computertomographie



# Stand der Lastverwaltung

## Status:

- Verfahren zur effizienten Lasterfassung
- Verschiebeverfahren für Prozesse und Objekte
- Verfahren zur Systemintegration und Anwendungsintegration der Lastverwaltung

## Übergang zu neuen Architekturen

- Grid-Umgebungen

# Künftig: Betrachtung von E/A

Ein-/Ausgabe bisher bei der Lastverwaltung vernachlässigt!

Hochleistungsrechnen:

- E/A-Phasen sehr lange; große Datenmengen
- E/A stark ortsgebunden
- Dateisysteme mit Lastausgleich

# Künftig: Interoperabilität

## Zusammenspiel der Lastverschiebung mit anderen Werkzeugen

- Konfliktfreies Zusammenspiel mit anderen Werkzeugen: Synergie erwünscht
- Z.B. mit Leistungsanalysator, Visualisierer
- Langjährige Arbeiten bei Monitoring-Konzepten (siehe Vortrag Wismüller 😊)
- Erste Adaption an CORBA erfolgreich

# Künftig: Sicherheit

## Sicherheit des Systembetreibers

- Prozesse/Objekte dürfen nicht spionieren

## Sicherheit des Anwenders

- Prozesse/Objekte dürfen nicht ausspioniert werden

**CCGrid2002 Workshop Security**

Besonders wichtig in GRID-Umgebungen