5.3 **Prozessorvergabe auf Anwendungsebene (user level threads)**

Bellosa, F.: Three Dimensions of Scheduling. Arbeitsberichte des IMMD, Band 31, Nummer 12, Dezember 1998, 192 Seiten.

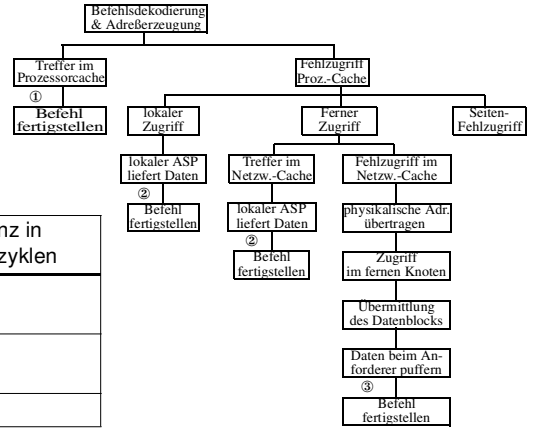
Prozessorvergabe auf Anwendungsebene besser an Aufgabenstellung anpaßbar

- **Alle Schedulingoperationen im gleichen Adreßraum**
  ➡ **Reduktion von Fehlzugriffen**
- **Vergabealgorithmen können an Aufgabe angepaßt werden, z. B. Verzicht auf verdrängende Zuordnung**
- **Bessere Anpaßbarkeit der Datenstrukturen für Verwaltungsdaten**

❑ **Besondere Berücksichtigung der Fähigkeiten von NUMA-Architekturen möglich**

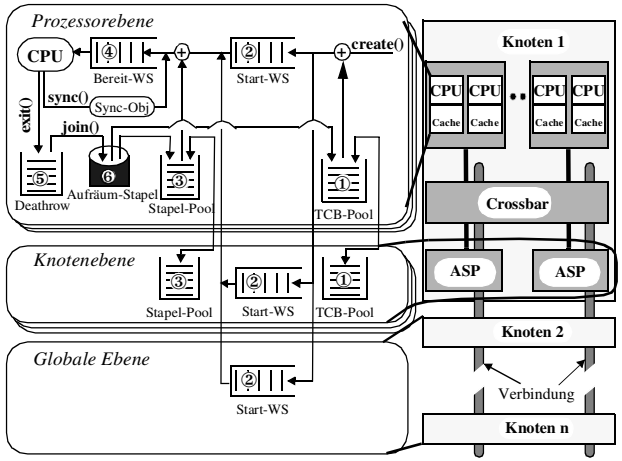
- **Möglichst keine globalen Datenstrukturen (Warteschlangen, Koordinierungsobjekte) wegen der Gefahr von Engpässen**
- **Zuordnung zu Prozessor mit gutem Zugang zu den benötigten Daten**
- **Vorausschauendes Puffern zur besseren Überlappung von Berechnung und Transport**

❑ **Beispielhafte Realisierung für Convex SPP 1000 (Architektur sehr ähnlich HP 9000 V2500):**

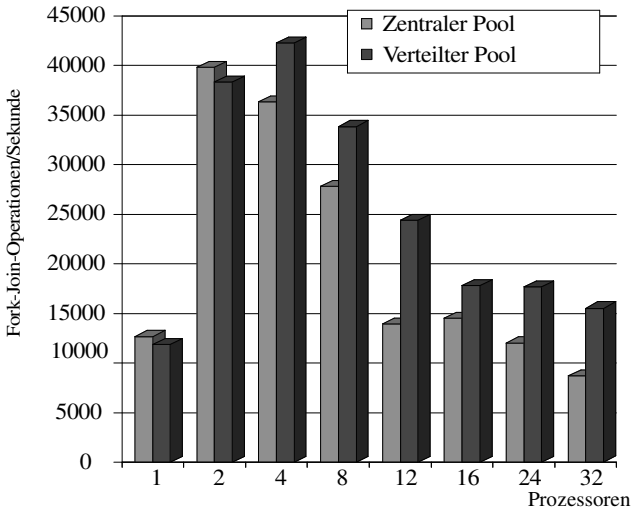


	Speicherhierarchie	Latenz in Taktzyklen
①	Prozessor-Cache	1
②	Knotenlokaler ASP	50
③	Ferner ASP	200

❑ **Architektur der Datenstrukturen in Anlehnung an die Hardware-Architektur**



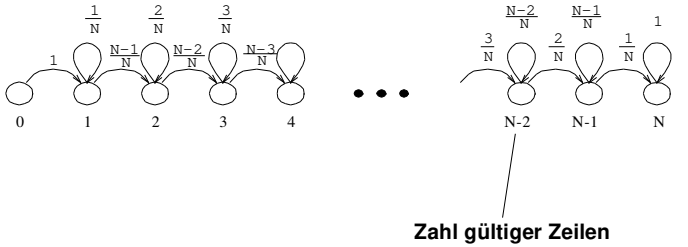
❑ **Vergleich zentraler und verteilter Haltung des Pools für Kontrollblöcke**



- **Schnelle Umschaltung**
  - **Durch Prozeßumschalter-Faden**
  - **Vorausschauend**
    - **Vergleich von Umschaltzeiten**

Operation	Takt Zyklen
Kontextumschaltung zwischen Fäden mit den zugehörigen Informationen im Cache	153
Kontextumschaltung zwischen Fäden im gleichen Knoten	1122
Kontextumschaltung zwischen Fäden in unterschiedlichen Knoten	1805

- ***Reload Transient Model***
  - **Bei Zuordnung Berücksichtigung der Arbeitsmenge eines Fadens, die sich im Pufferspeicher befindet; im weiteren als "Fußabdruck" (footprint) bezeichnet**
  - **Wie kann die Größe des Fußabdrucks ermittelt werden?**
  - **Abschätzung anhand eines Markovmodells**
    - **Laufende Fäden steigern die Zahl der gültigen Pufferspeicherzeilen (alle N Pufferspeicherzeilen gleich häufig angesprochen)**



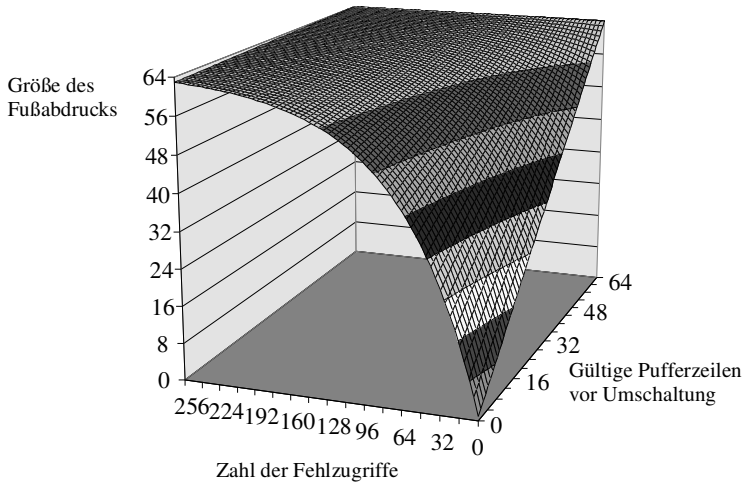
Zugehörige Übergangsmatrix

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{N} & \frac{N-1}{N} & 0 & 0 & \dots & 0 \\ 0 & 0 & \frac{2}{N} & \frac{N-2}{N} & 0 & \dots & 0 \\ \vdots & & & \ddots & & & \vdots \\ & & & & & & 0 \\ 0 & & & \dots & 0 & \frac{N-1}{N} & \frac{1}{N} \\ 0 & & & & \dots & 0 & 1 \end{bmatrix}$$

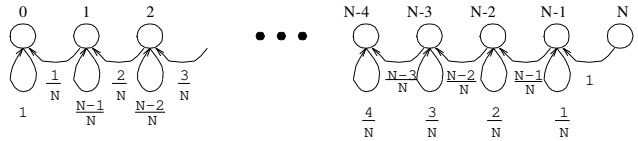
➡ **Mittlere Größe** $E[F|V = v, M = n]$ **des Fußabdrucks, wenn zunächst v**

Pufferzeilen gültig sind nach n Fehlzugriffen: $F_v^n = \sum_{i=0}^{N} j \cdot P^n[v, j]$

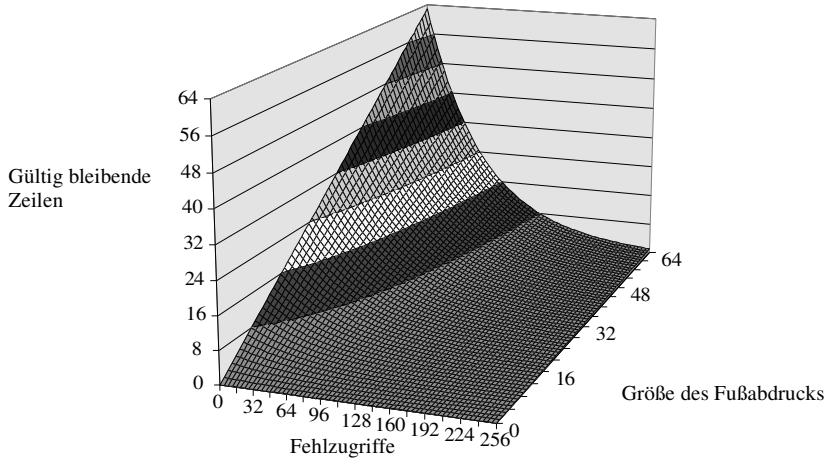
- **Direkt abgebildeter Pufferspeicher, 64 Zeilen (ähnlich bei anderer Pufferzeilenzahl**



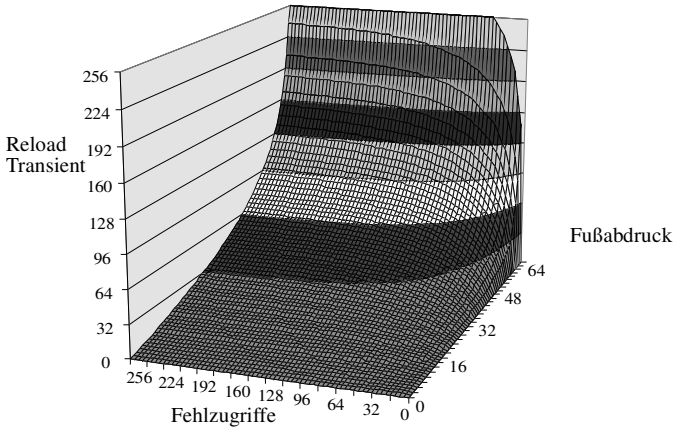
- **Verlust gültiger Cachezeilen**
  - **Blockierte Fäden verlieren Cachezeilen auf Grund von Fehlzugriffen anderer Fäden**



Mittlere Zahl $E[V|F = f, M = n]$ gültig bleibender Zeilen, wenn zunächst v Pufferzeilen gültig sind, nach n Fehlzugriffen

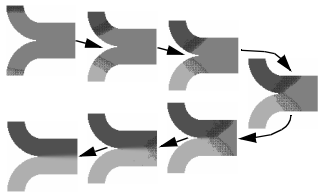


- **Schätzung der erwarteten Fehlzugriffe beim Wiederanlauf**
  - **Wiederanlauf des Threads mit minimalem Reload Transient**

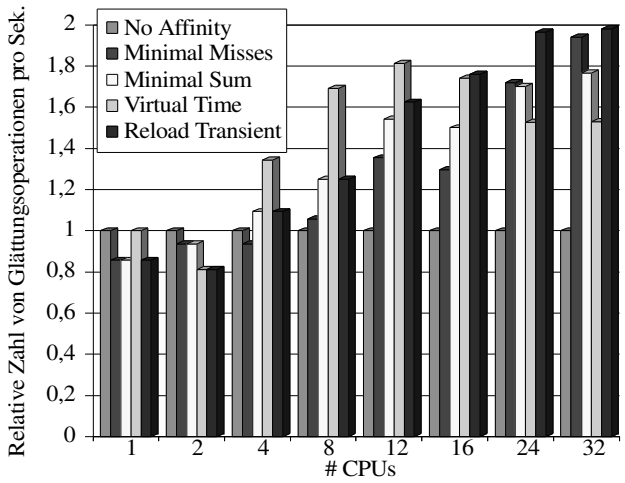


- **Auswahl des Fadens mit dem geringsten Aufwand zur Wiederherstellung seines letzten Fußabdruck**

- **Messungen**
  - **Verglichene Strategien**
    1. *No Affinity:* **Zuordnung nach LIFO**
    2. *Virtual Time:* **Zuordnung des jüngst zugeordneten**
    3. *Minimal Misses:* **Zuordnung des Fadens, der während letzter Bearbeitung die wenigsten Fehlzugriffe verursachte**
    4. *Minimal Sum:* **Zuordnung des Fadens, der bislang die wenigsten Fehlzugriffe verursachte**
    5. *Reload Transient:* **Zuordnung nach minimalem geschätztem Wiederherstellungsaufwand für seinen Fußabdruck**
  - **Beispiel:**



Active Threads Smoother



Universität Erlangen-Nürnberg, Lehrstuhl für Informatik 4 (Verteilte Systeme und Betriebssysteme), F. Hofmann
Reproduktion jeder Art oder Verwendung dieser Unterlage zu Lehrzwecken außerhalb der Universität Erlangen-Nürnberg
ist ohne Genehmigung des Autors unzulässig