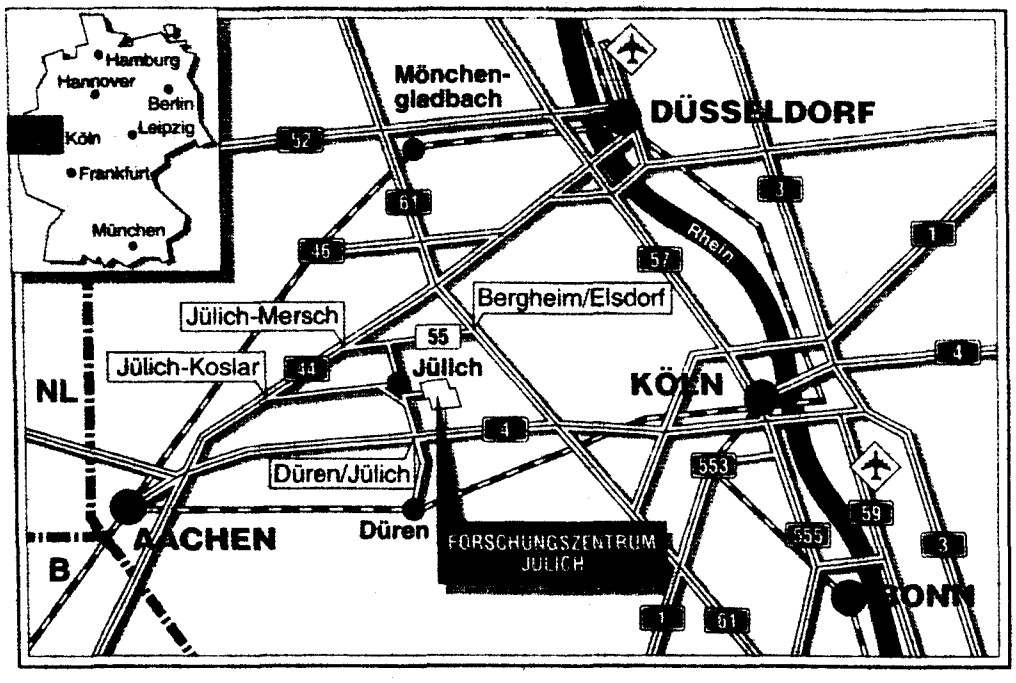
FORSCHUNGSZENTRUM JÜLICH GmbH

Zentralinstitut für Angewandte Mathematik

Visualisierung der Speicheraktivitäten von parallelen Programmen in Systemen mit virtuell gemeinsamem Speicher

Christof M. Müllender



Berichte des Forschungszentrums Jülich ; 2911
ISSN 0944-2952
Zentralinstitut für Angewandte Mathematik Jül-2911

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek
D-52425 Jülich · Bundesrepublik Deutschland
Telefon: 02461/61-6102 · Telefax: 02461/61-6103 · Telex: 833556-70 kfa d

Visualisierung der Speicheraktivitäten von parallelen Programmen in Systemen mit virtuell gemeinsamem Speicher

Christof M. Müllender

Zusammenfassung

Das Konzept des *virtuell gemeinsamen Speichers (Virtual Shared Memory, VSM)* wurde entwickelt, um die Programmierung moderner Parallelrechner mit physikalisch verteiltem Speicher zu vereinfachen. Die bei derartigen Architekturkonzepten zu beobachtenden Speicheraktivitäten sind so komplex, daß geeignete Werkzeuge zur Darstellung dieser Vorgänge notwendig sind, die das Systemverhalten visualisieren und damit sowohl zum Verständnis der dynamischen Vorgänge beitragen als auch die Basis für eine effektive Optimierung bereitstellen.

In dieser Arbeit wird ein Werkzeug vorgestellt, das die internen Vorgänge in Systemen mit virtuell gemeinsamem Speicher auf verschiedene Arten graphisch darstellt. Dabei sind Zeitpunktdarstellungen, in denen die Beziehungen zwischen den Prozessoren und den Einheiten der Adreßraumabbildung (Speicherseiten) übersichtlich dargestellt werden, ebenso vorgesehen wie Zeitachsendarstellungen, die es erlauben, das Systemverhalten vollständiger Zeiträume zu erfassen. Automatische Statistiken und eine Möglichkeit zur Darstellung dreidimensionaler Topologiemodelle zur Beobachtung der Kommunikationsaktivitäten bilden weitere, wesentliche Komponenten des Werkzeugs. Durch die Integration dieser neuen, auf das Architekturkonzept des virtuell gemeinsamen Speichers ausgerichteten Werkzeugkomponenten in die bestehende Visualisierungsumgebung PARvis (Visualisierung von Prozeßwechseln auf Parallelrechnern) ist so ein leistungsfähiges Werkzeug entstanden, das nun, aus ganz unterschiedlichen Perspektiven, ein weitgehend vollständiges Bild des Zusammenspiels von parallelem Programm und Parallelrechner zeichnen kann.

Abstract

The concept of *virtual shared memory (VSM)* was developed in order to simplify the programming of modern parallel computers with physically distributed memory. The memory activities occurring in such an architecture are so complex that we require tools to visualize and understand the dynamic processes to provide a basis for an effective debugging and optimization.

In this thesis, a tool is introduced which visualizes the virtual shared memory references in different ways. Graphical representations for specified points in time, which clearly show the communication of memory pages, are implemented as well as time line visualizations that allow the observation of the evolution of system behaviour. The system can automatically generate statistics, and a three dimensional representation of the communication topologies can be used to observe communication activities. Because these VSM components are integrated into PARvis (an existing tool for visualizing process state changes of a parallel computer), an efficient tool has been developed that, by taking into consideration different perspectives, is now able to draw a nearly complete image of the interaction between a parallel program and a parallel computer.

Inhalt

Kapitel 1

Einleitung

Da die Rechengeschwindigkeit herkömmlicher, skalarer Computersysteme an physikalische Grenzen stößt, die durch minimale Leitungslängen auf der einen Seite und maximale Signalgeschwindigkeit (Lichtgeschwindigkeit) auf der anderen Seite abgesteckt sind, werden in der heutigen Zeit zunehmend Parallelrechner zur Lösung zeitkritischer Aufgaben herangezogen.

Die prinzipiell unbegrenzte Skalierbarkeit paralleler Architekturen, die theoretisch beliebige Geschwindigkeitsgewinne erhoffen läßt, stößt aber nicht nur an technische Grenzen. Vielmehr gibt es auch grundlegende Verständnisschwierigkeiten bei der Programmierung dieser Rechner, die bereits bei der parallelen Verwendung einiger weniger Prozessoren auftreten. So ist es oft sehr schwer, eine eventuell vorhandene probleminhärente Parallelität zu erkennen und bei der Programmierung auszunutzen. Zudem treten insbesondere im komplexen, vernetzten System eines MIMD-Parallelrechners Effekte auf, die kaum vorhersehbar und praktisch nicht reproduzierbar sind, da bei solchen Architekturen die einzelnen Prozessoren in der Regel nicht zeitsynchron arbeiten.

Der Hauptspeicher ist bei massiv-parallelen Rechnern zumeist physikalisch auf die Prozessoren verteilt, was kostengünstiger zu realisieren ist und, im Vergleich zu Multiprozessorsystemen mit gemeinsamem Speicher, eine leichtere Skalierbarkeit des Systems ermöglicht. Allerdings erhöht sich durch die verteilte Speicherarchitektur die Komplexität der Programmierung, da dafür gesorgt werden muß, daß die richtigen Daten zur richtigen Zeit am richtigen Ort sind (lokaler Prozessorspeicher). Um dies sicherzustellen, müssen zudem explizite Datenübertragungsbefehle aufgerufen werden (Message-Passing-Programmierkonzept).

Das von K. Li 1986 vorgestellte und in der Folgezeit dann eingehend untersuchte Konzept des virtuell gemeinsamen Speichers bietet in diesem Sinn eine Erleichterung der Programmierung eines Rechners mit physikalisch verteiltem Speicher [11]. Dies geschieht durch die Aufteilung eines logisch gemeinsamen Adreßraums auf physikalisch verteilte Speichermodule des parallelen Rechnersystems (Virtual Shared Memory, VSM). Das Verständnis der Vorgänge in so konzipierten Rechnersystemen und

damit die Minimierung des unvermeidlichen Effizienzverlustes, der durch die Adreßraumabbildung entsteht, sind aktuelle Forschungsthemen [13]. Zur Unterstützung dieser Forschung sind leistungsfähige Werkzeuge nötig, die die internen Vorgänge in Systemen mit virtuell gemeinsamem Speicher darstellen und damit sowohl zum allgemeinen Verständnis der hochdynamischen Vorgänge beitragen als auch, zum Zweck der Bewertung und Optimierung, Parameterstudium zur effizienten Abbildung von Anwendungen auf derartige Systeme unterstützen.

Ein besonders naheliegender Ansatz zur Verdeutlichung ist die graphische Veranschaulichung. Dabei kann im Sinn einer komfortablen Benutzeroberfläche die moderne Bildschirm-Fenstertechnik gewinnbringend eingesetzt werden, die auch die gleichzeitige Verwendung von verschiedenen Werkzeugkomponenten unterstützt.

Zur Durchführung dieser Visualisierungsaufgabe müssen Konzepte entwickelt werden, die einerseits eine unkomplizierte Bedienung und andererseits eine weitgehende Variabilität in bezug auf zukünftige, veränderte Darstellungsaufgaben gewährleisten. Dazu ist es notwendig, zunächst einige grundsätzliche Überlegungen zum Thema "Visualisierung dynamischer Systeme" anzustellen, um die vorliegende Aufgabe sowohl theoretisch als auch praktisch zu systematisieren. In gleichem Maße ist ein genaues Verständnis des Darstellungsgegenstandes mit allen darin inhärent enthaltenen Eigenheiten und Problemen zur geeigneten Abbildung auf graphische Elemente nötig.

Die vorliegende Arbeit beschreibt nicht nur das Ergebnis der Entwicklungsarbeiten, sondern auch den Weg dorthin und spiegelt damit die oben genannten Aspekte wider. Daher werden in den nachfolgenden Kapiteln die Themen

- Visualisierung dynamischer Systeme (Kapitel 2),
- der virtuell gemeinsame Speicher (Kapitel 3),
- Werkzeuge zur Visualisierung von Systemen mit virtuell gemeinsamem Speicher (Kapitel 5 und 6)

behandelt. Ein Kapitel über bisherige Arbeiten auf dem Gebiet der Visualisierung der Vorgänge in Rechnersystemen (Kapitel 4) rundet die Arbeit ab.

Das im Rahmen dieser Arbeit vorgestellte Werkzeug bildet eine Erweiterung des bereits existierenden Visualisierungssystems PARvis [1], das Prozeßwechselaktivitäten in Mehrprozessorsystemen auf vielfältige Weise darstellen kann. Gerade die Kombination dieser Komponenten mit den neuen, speicherbezogenen Werkzeugen bildet eine ausgesprochen fruchtbare Synthese, da kein Systemaspekt sinnvoll ohne Berücksichtigung des anderen betrachtet werden kann und erst die aus beiden Perspektiven entstehende Gesamtsicht ein vollständiges Bild des Untersuchungsobjekts Parallelcomputer zeichnen kann.

Kapitel 2

Visualisierung dynamischer Systeme

Die Visualisierung von Zustandswechseln in dynamischen Systemen wirft eine Reihe von Problemen auf. So können z.B. die Vorgänge in einem komplexen dynamischen System in den seltensten Fällen durch eine einzige Darstellungsmethode in ihrer Vielfalt wiedergegeben werden. Daher müssen in der Regel mehrere Visualisierungswerkzeuge bereitstehen, damit durch verschiedene Perspektiven die unterschiedlichen Aspekte des Systems herausgestellt werden können. In diesem Kapitel sollen die verschiedenen Möglichkeiten der Visualisierung dynamischer Systeme auf einige Grundtypen zurückgeführt werden, die jeweils bestimmte Aspekte besonders deutlich zur Darstellung bringen. Dadurch soll eine Einordnung der behandelten Werkzeuge aus der Literatur und der mit dieser Arbeit eingeführten Hilfsmittel möglich gemacht werden.

2.1 Visualisierungsmethoden: Versuch einer Klassifizierung

Auf welche Weise kann ein dynamisches System sinnvoll betrachtet werden? Eine mögliche Klassifizierung unterscheidet die verschiedenen Alternativen danach, in welcher Weise die Zeit in einer Darstellung abgebildet wird. In dieser Arbeit wird die Einteilung in folgende Kategorien vorgeschlagen:

- *Zeitpunktdarstellungen*: dies sind Darstellungen des Systemzustandes für einen bestimmten Zeitpunkt. Dieser Zeitpunkt bildet einen grundsätzlichen Parameter der gewählten Visualisierung und muß *vor* deren Aufbau festgelegt werden. Das System selbst wird in der Ansicht auf graphische Darstellungselemente abgebildet, die es anschaulich repräsentieren sollen.

- *Animationen*: sie entstehen dadurch, daß gleichartige Darstellungen der oben erwähnten Art für eine Folge sukzessiver Zeitpunkte hintereinander zur Ansicht gebracht werden. Die natürliche zeitliche Folge von Zuständen des realen Objektes wird auf eine zeitliche Folge von *Zustandsdarstellungen* abgebildet.

- *Zeitachsendarstellungen*: hier handelt es sich um die Visualisierung von Zeiträumen in einer Darstellung. Bei dieser Art der Veranschaulichung bildet man die Zeit auf eine meist räumliche Dimension (Zeitachse) eines Koordinatensystems ab. Das System stellt man dann durch andere Dimensionen desselben Koordinatensystems wie orthogonale Achsen, Farben oder Muster dar.

- *Statistiken*: sie bilden eine besondere Form statischer Darstellung, die Aussagen über vorgegebene Zeiträume machen. Hier ist also die Zeit *implizit* in den Darstellungen enthalten. Die ermittelten Werte können auf verschiedene Arten veranschaulicht werden. Verbreitete Methoden hierfür sind Tortendiagramme, Histogramme oder Sterndiagramme.

In Abbildung 2.1 werden diese vier Kategorien noch einmal verdeutlicht.

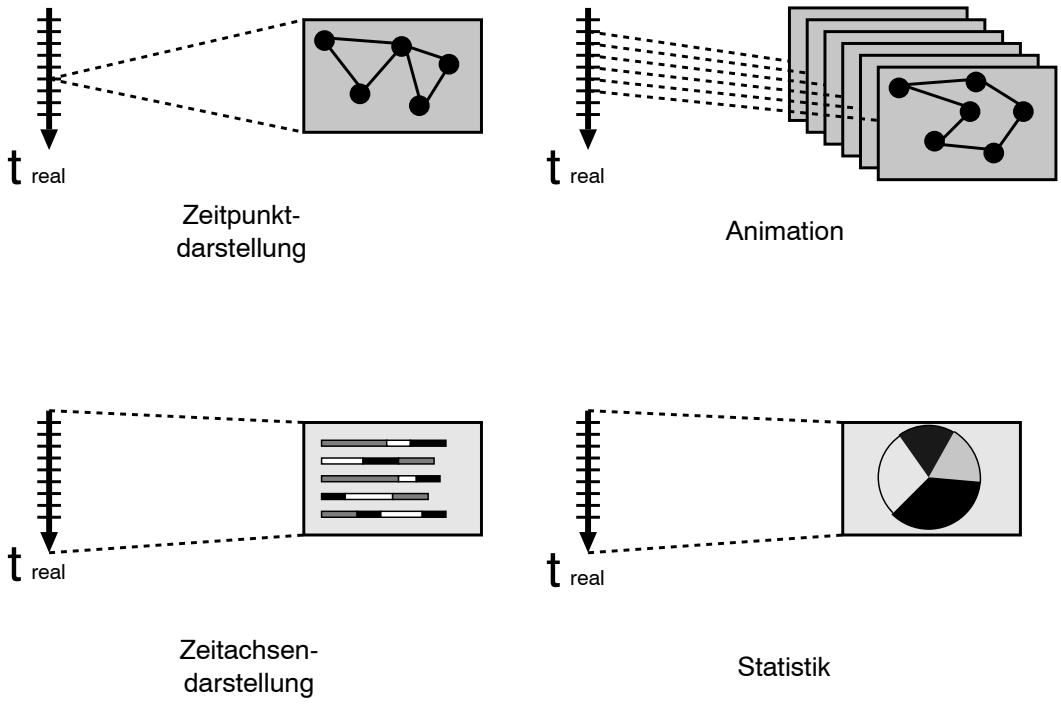


Abbildung 2.1: Zur Verdeutlichung der Visualisierungs-Kategorien

2.2 Die Post-Mortem-Analyse

Zur Darstellung von Daten, die von einem realen oder simulierten System erzeugt werden, gibt es grundsätzlich zwei Methoden:

- Eine Möglichkeit besteht darin, das System *während* seines Laufs zu betrachten (online). Hierbei müssen die zu veranschaulichenden Ereignisse direkt registriert und an das Visualisierungswerkzeug weitergegeben werden. Diese Technik erlaubt nur die Methode der animierten Darstellung (s.o.).
- Wesentlich mehr Spielraum bietet dagegen die Betrachtung *nach* dem Systemlauf (die sogenannte Post-Mortem-Analyse). Die vom System erzeugten Daten müssen hier für die spätere Analyse auf einem externen Speichermedium festgehalten, d.h. protokolliert werden. Nach Erzeugung der Protokolldatei kann diese mit geeigneten Hilfsmitteln aller oben beschriebenen Kategorien untersucht werden, die Ergebnisse der Untersuchung können zur Anpassung der Systemparameter vor einem erneuten Lauf verwendet werden.

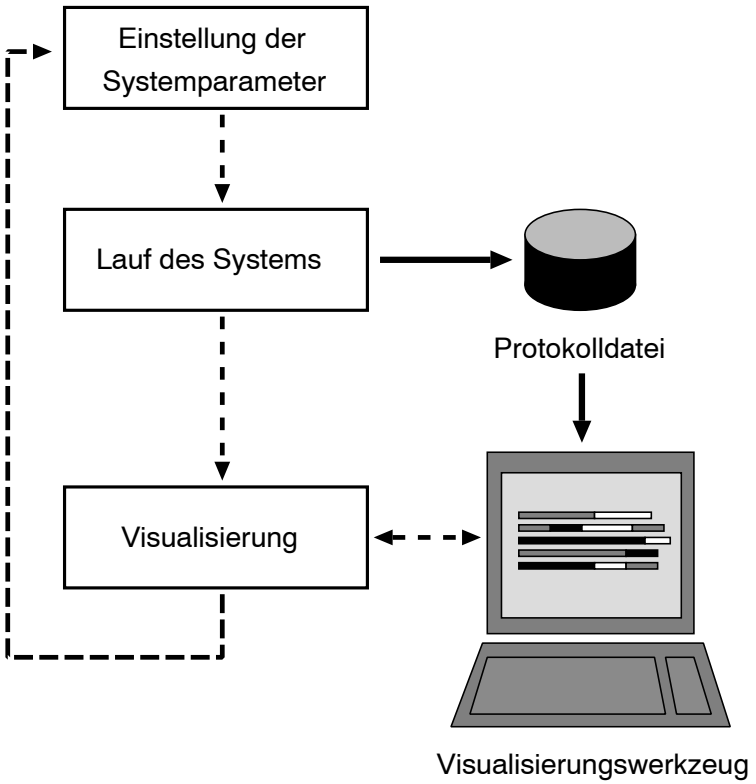


Abbildung 2.2: Prinzip der Post-Mortem-Analyse

Abbildung 2.2 zeigt das Prinzip der Post-Mortem-Analyse. Sie besitzt gegenüber der Betrachtung während des Laufs die folgenden zusätzlichen Möglichkeiten, die sich aus der Loslösung von der realen Systemzeit ergeben:

- freie Bewegung auf der Zeitachse, d.h. z.B.
  - wiederholte Betrachtung gleicher Zeitpunkte und -räume
  - Anpassung an das Aufnahmevermögen des Anwenders
- größere Vielfalt von Werkzeugen, d.h. z.B.
  - Darstellung von bestimmten Zeitpunktdarstellungen, Zeitachsendarstellungen sowie Statistiken
  - gleichzeitige Verwendung verschiedener Visualisierungswerkzeuge
  - gleichzeitige Darstellung von verschiedenen Zeitpunkten und -räumen

Aus diesen Gründen wurde bei PARvis das Post-Mortem-Prinzip verwirklicht, alle bereits vorhandenen Hilfsmittel sowie die in dieser Arbeit vorgestellten Werkzeuge dienen zur Untersuchung eines bereits abgeschlossenen, protokollierten Systemlaufs aus unterschiedlichen Blickwinkeln.

Aber es gibt auch einen bedeutsamen Nachteil dieses Visualisierungsprinzips: Der Anwender der Werkzeuge hat keine Möglichkeit, während der Analyse auf den Ablauf des Systems Einfluß zu nehmen, um beispielsweise schon gewonnene Erkenntnisse zu berücksichtigen. Diese Möglichkeit ist bei der Untersuchung während des Laufs grundsätzlich gegeben.

In beiden Fällen ist das Registrieren und die Weitergabe der für die Beobachtung wesentlichen Ereignisse und Daten erforderlich. Bei Simulationsläufen ist dies problemlos möglich, bei der Untersuchung realer Systeme jedoch besteht das Problem, daß die wichtigen Ereignisse übertragen werden müssen, ohne daß dadurch das zu untersuchende Systemverhalten wesentlich beeinflußt wird. Das ist eine Forderung, die unter Umständen nur schwer zu erfüllen ist, wenn man bedenkt, daß man es meistens mit komplexen, nichtlinearen, vernetzten Systemen zu tun hat, bei denen "isolierte Manipulationen" fast unmöglich sind. Es gibt jedoch Forschungsprojekte, in denen die erforderlichen Daten eines Rechnersystems während des Laufes durch Hardware-Komponenten aufgenommen werden, wobei das zu untersuchende System nur wenig gestört wird [8, 9].

2.3 Visualisierung von Multiprozessorsystemen

In typischen Zeitpunktdarstellungen bei der Visualisierung der internen Vorgänge in Mehrprozessorsystemen werden bestimmte Komponenten (wie z.B. Prozessoren) durch graphische Elemente wiedergegeben. In diese Elemente können dann relevante Informationen durch Text, Farbe, Muster usw. hineingebracht werden (als Beispiel siehe [1, S. 48]). Weit verbreitete Zeitpunktdarstellungen sind auch Topologie-Visualisierungen [22, 14], in denen Prozessoren in einer Anordnung dargestellt werden, die die Struktur des Verbindungsnetzwerks wiedergibt. Das Netzwerk selbst

kann dann durch Linien verdeutlicht werden, die durch ihr jeweiliges Aussehen Auskunft über die Aktivitäten des Netzes geben können. Auch Matrixdarstellungen, in denen die horizontale und die vertikale Achse mit Systemkomponenten beschriftet sind, eignen sich sehr gut zur Visualisierung von momentanen Beziehungen zwischen verschiedenen Komponenten des Systems. Animationen können dann durch die fortlaufende Wiedergabe sukzessiver Zeitpunktdarstellungen erzeugt werden.

Bei Zeitachsendarstellungen hat man die Wahl, mit welchen Komponenten oder Eigenschaften man die (neben der Zeitachse) verbleibende räumliche Dimension der Bildschirmebene identifiziert. Wählt man dafür bestimmte Komponenten aus wie z.B. Prozessoren, können Eigenschaften dieser Komponenten oder Beziehungen zu anderen Elementen des Systems durch Attribute wie Farbe, Muster oder Beschriftung wiedergegeben werden. Zum Beispiel können solche Diagramme die Beziehungen von Prozessoren (vertikale Achse) zu Prozessen (Farben, Beschriftungen) für einen Zeitraum veranschaulichen, wie es in PARvis realisiert wurde [1, S. 55].

Typische Mittel zur Darstellung von Statistiken sind u.a. Tortendiagramme, die z.B. die anteiligen Beschäftigungszeiten eines Prozessors mit verschiedenen Prozessen verdeutlichen können. Eine weitere Darstellungsmethode dieser Kategorie bilden die schon erwähnten Histogramme. Beispiele für Statistik-Fenster finden sich in [1, S. 53]. Meistens sind es Gesamtaussagen wie die Auslastung oder die Effizienz des Systems oder bestimmter Komponenten, die durch Statistik-Darstellungen verdeutlicht werden. Derartige Aussagen können einen ersten Eindruck vom Verhalten des Systems geben und dem kundigen Beobachter erste Anzeichen möglicher Problemstellen liefern.

Kapitel 3

Architekturkonzepte für Systeme mit virtuell gemeinsamem Speicher

Der virtuell gemeinsame Speicher (Virtual Shared Memory, VSM) ist ein Architekturkonzept zur Kombination von physikalisch verteiltem mit logisch gemeinsamem Speicher, das in der grundlegenden Arbeit von K. Li 1986 ausführlich vorgestellt und in der Folgezeit eingehend untersucht wurde [11]. Es dient dazu, dem Programmierer eines Mehrprozessorsystems mit auf die Prozessoren aufgeteilten, lokalen Speichereinheiten einen globalen Adreßraum zur Verfügung zu stellen und ihm damit die Perspektive eines gemeinsamen Speichers zu geben, wodurch ihm die Programmentwicklung erleichtert werden soll. Die Abbildung von dem logischen auf den physikalischen Adreßraum geschieht, ähnlich wie bei Betriebssystemen skalarer Rechenanlagen, durch die Verwendung virtueller Speichereinheiten (Seiten), die in geeigneter Weise auf physikalische Speichereinheiten (Seitenrahmen) abgebildet werden.

3.1 Das Grundkonzept

Für den Programmierer eines Systems mit virtuell gemeinsamem, physikalisch jedoch verteiltem Speicher ist von jedem einzelnen Prozessor aus der Zugriff auf einen gemeinsamen virtuellen Adreßraum möglich (Abbildung 3.1). Dieser virtuelle Adreßraum wird durch die Gesamtheit der lokalen Speichermodule aller Prozessoren des Systems realisiert. Beim Zugriff einer CPU auf eine virtuelle Speicheradresse dient ihr, ähnlich wie bei einem Cache, der eigene lokale Speicher als Puffer. Das bedeutet, daß der gewünschte Speicherinhalt aus dem Speichermodul eines anderen Prozessors über das Verbindungsnetzwerk in den eigenen lokalen Speicher geladen werden muß, falls er sich dort nicht befindet. Dies geschieht für den Programmierer völlig transparent, für ihn ist es lediglich ein Zugriff auf eine Adresse im virtuellen

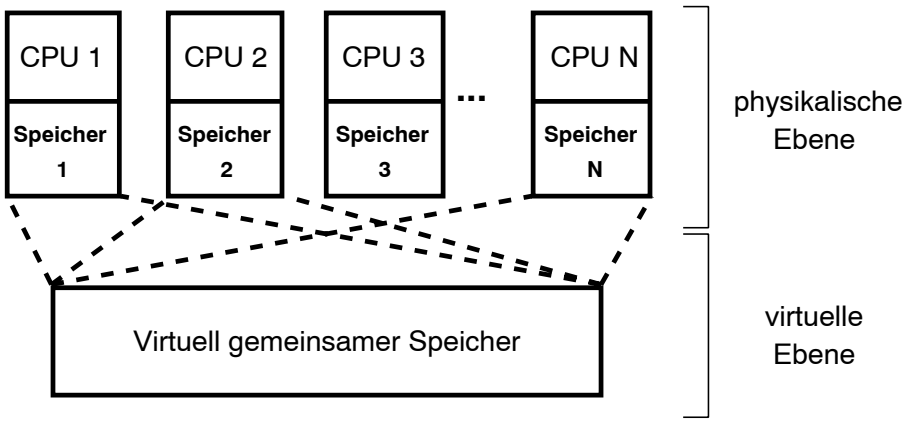


Abbildung 3.1: Das Prinzip des virtuell gemeinsamen Speichers

Adreßraum. Real ergibt sich eine Speicherhierarchie, die in folgender Weise aufgebaut ist:

- lokaler Speicher des anfragenden Prozessors
- lokale Speicher aller übrigen Prozessoren
- eventuell externer Hintergrundspeicher (z.B. Plattenspeicher)

In den hier vorgestellten Überlegungen und Werkzeugen bleibt die Ebene des externen Hintergrundspeichers unberücksichtigt.

Der gesamte virtuelle Adreßraum ist in Seiten eingeteilt, die die Grundeinheiten für die Adreßraumabbildung darstellen [11]. Die virtuellen Seiten sind physikalisch in Seitenrahmen in den lokalen Speichern der Prozessoren abgelegt (Abb. 3.2).

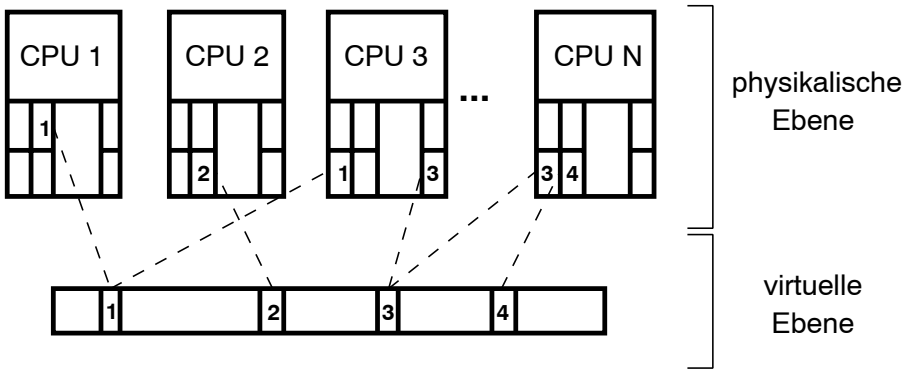


Abbildung 3.2: Die Seite als Einheit der Adreßraumabbildung

In [13] findet man einen sehr guten Überblick über Virtual-Shared-Memory-Systeme und die dort realisierten Konzepte. Diese Ausführungen bilden im wesentlichen die Grundlage für die weiteren Bemerkungen in den nächsten Abschnitten.

Da der parallele Lesezugriff von mehreren CPUs auf eine Seite möglich sein muß, können mehrere Kopien einer Seite in den realen Speicherbereichen existieren. Dabei muß sichergestellt sein, daß eine gültige Kopie einer Seite ihrem aktuellen Stand und damit der Speicherseite entspricht, die nach dem letzten Schreibbefehl eines beliebigen Prozessors auf diese Seite entstanden ist. Dies entspricht der Forderung nach der sogenannten "strikten Kohärenz". Wird nur "schwache Kohärenz" verlangt, dürfen zeitweise unterschiedliche Versionen einer virtuellen Seite existieren. Diese Versionen müssen dann zu Synchronisationszeitpunkten vereinheitlicht werden.

Es gibt verschiedene Verfahren, mit denen die Forderung nach strikter Kohärenz erfüllt werden kann:

- *Invalidation*: Bei Erteilung der Schreibberechtigung an einen Knoten auf eine Seite werden alle anderen Kopien der Seite ungültig gemacht.

- *Zurückschreiben*: Es wird gleichzeitig auf alle existierenden Kopien einer Seite geschrieben (ungeeignet für Systeme mit verteiltem Speicher wegen des hohen Kommunikationsaufwandes).

- *Write-Update-Technik*: Bei Erteilung der Schreibberechtigung an einen Knoten werden alle anderen Kopien der Seite nur vorübergehend gesperrt und nach Beendigung des Schreibvorganges aktualisiert.

- *Remote-Mapping*: Direkter Zugriff auf einzelne Speicherstellen unmittelbar im Lokalspeicher eines anderen Prozessors, ohne daß die ganze Seite vorher in den eigenen lokalen Speicher geladen wird.

Die speziellen Aspekte der Strategie der schwachen Kohärenz sollen in dieser kurzen Einführung nicht weiter behandelt werden. Es wird auf die Literatur verwiesen [13].

3.2 Zugriffsverfahren

Greift ein Prozessor auf eine Speicherstelle in einer virtuellen Seite zu, die sich nicht in seinem lokalen Speicher befindet, werden entsprechende Page-Fault-Aktionen ausgelöst. Die folgenden Ausführungen beziehen sich auf die Anwendung der Manager-Algorithmen "zentraler Manager" oder "statisch-verteilter Manager", die in [12] vorgestellt und miteinander verglichen werden.

Zwei Prozessoren spielen bei der Auflösung eines Page-Fault bezüglich einer Seite eine besondere Rolle:

- der *Besitzer* einer virtuellen Seite ist ein Prozessor, der über eine aktuelle Kopie der Seite verfügt. Das ist in der Regel derjenige, der als letzter eine Schreiboperation auf der Seite durchgeführt hat.
- der *Manager* ist ein der virtuellen Seite zugeordneter Prozessor, der die Information bereithält, wer der aktuelle Besitzer dieser Seite ist. Wird ein Besitzerwechsel beantragt, wird er hier in der Regel sofort eingetragen. Bis der Wechsel tatsächlich stattgefunden hat, entspricht die Information des Managers dann also zeitweilig nicht den wirklichen Gegebenheiten.

Somit wird der folgende Weg zur Auflösung des Page-Fault beschritten: Der anfordernde Prozessor sendet eine entsprechende Nachricht an den Manager der gewünschten Seite. Dieser beauftragt den Besitzer, eine Kopie der Seite an die anfordernde CPU zu senden. Es werden also zur Auflösung eines Page-Fault drei Nachrichten über das Kommunikationsnetz des Mehrprozessorsystems übertragen (Abbildung 3.3).

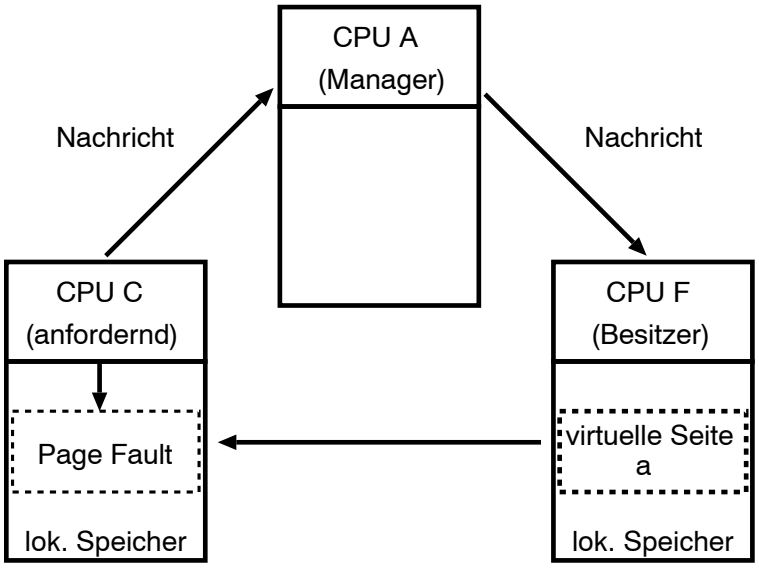


Abbildung 3.3: Auflösung von Page-Faults in VSM-Systemen

Zur Veranschaulichung der erwähnten Prinzipien soll kurz der Weg einer virtuellen Seite in einem Virtual-Shared-Memory-System mit strikter Kohärenz durch Invalidation verfolgt werden:

Beispiel (siehe auch Abbildung 3.3): Die virtuelle Seite a befinde sich im Schreibzugriff des Prozessors F. Der Prozessor C möchte die Seite a lesen. Er sendet eine

entsprechende Nachricht an den Manager der Seite, Prozessor A. Dieser übermittelt eine Anforderungsnachricht an den aktuellen Besitzer der Seite a, Prozessor F (da F Schreibzugriff besitzt). Dieser unterbricht seinen Schreibzugriff und sendet eine aktuelle Kopie von a an Prozessor C. Während dieser seinen Lesevorgang durchführt, fordert F die Seite erneut zum Schreiben an. Da sich eine aktuelle Kopie im eigenen Speicher befindet, braucht keine Anforderung an den Manager übermittelt zu werden. Prozessor F sendet vielmehr eine Invalidations-Nachricht über den Manager A an C und erklärt damit dessen Kopie für ungültig. Anschließend beschreibt Prozessor F die Seite a und so fort.

Dieses Beispiel vermittelt bereits einen Eindruck von der Komplexität der Vorgänge in einem System mit virtuell gemeinsamem Speicher. Dabei ist zu beachten, daß hier lediglich drei Zugriffsaktionen zweier Prozessoren auf eine Seite beobachtet wurden. Bei heute schon existierenden VSM-Systemen kann man es bereits mit mehreren hundert Prozessoren und Tausenden von Speicherseiten zu tun haben.

3.3 Wichtige Parameter in VSM-Systemen

Ein zentrales Objekt in den betrachteten Systemen ist die Speicherseite. Die *Seitengröße* hat einen entscheidenden Einfluß auf das Systemverhalten. Wird sie zu klein gewählt, wird der Verwaltungsaufwand beim Versenden einer Seite über das Kommunikationsnetz im Verhältnis zu groß, da dieser im allgemeinen für kleine und große Datenpakete gleich ist. Außerdem werden durch die höhere Zahl an Seiten bei gleicher Speichergröße alle seitenorientierten Systemtabellen, wie z.B. die zur Verwaltung der Zugriffsrechte, sehr umfangreich (großer Aufwand beim Zugriff). Wird die Größe der Speicherseiten jedoch zu groß gewählt, steigt die Wahrscheinlichkeit, daß mehrere Prozessoren gleichzeitig exklusiv auf eine Seite zugreifen wollen (Speicherzugriffskonflikte), was wiederum zu Verzögerungen führt. Die Größe der Seiten kann von der vorhandenen Hardware vorgegeben sein, insbesondere wenn von ihr Mechanismen zur VSM-Organisation zur Verfügung gestellt werden.

Ein weiterer wichtiger Parameter ist die Zuordnung zwischen den Speicherseiten und den Managern (*Manager-Algorithmus*). Man unterscheidet zwischen einem zentralen Manager, der alle Seiten verwaltet, und verteilten Managern, die statisch oder dynamisch zugeordnet sein können.

Da eine schwache Kohärenzstrategie den Parallelitätsgrad (beim Schreiben) erhöht, hat die *Kohärenzstrategie* (strikt oder schwach) einen direkten Einfluß auf das Systemverhalten. Ebenso prägen die Verfahren zur *Gewährleistung* der Kohärenz (s.o.) wesentlich das Verhalten des Gesamtsystems. Zum Beispiel ergibt die Technik des Zurückschreibens einen sehr hohen Kommunikationsaufwand bei Rechnern mit verteiltem Speicher. Die Write-Update-Technik kann gegenüber der Invalidationstechnik den Kopier-Aufwand reduzieren, benötigt jedoch einen stärkeren Nachrichtenaustausch [13, Kapitel 2].

Zusätzlich gibt es noch weitere wichtige Systemeigenschaften, die festzulegen sind und schon von Betriebssystemen skalarer Rechenanlagen bekannt sind. In diesem Zusammenhang sind die *Speicher-Allokations-Strategie* und die *Seitenersetzungsmechanismen* zu erwähnen.

Konstellationen dieser und anderer Parameter sind bei der Entwicklung von VSM-Systemen im Zusammenhang mit unterschiedlichen Anwendungs-Probemstellungen und verschiedenen Anfangs-Datenverteilungen zu untersuchen. Die Bewertung der Untersuchungsergebnisse ist nur mit Hilfe von Werkzeugen möglich, die genauen Einblick in das Verhalten und das Zusammenspiel der einzelnen Komponenten des VSM-Systems geben.

3.4 Der Einfluß des Netzwerks

In den Virtual-Shared-Memory-Simulationen, für die die hier vorgestellten Hilfsmittel zunächst entwickelt wurden, wird die Struktur der Prozessor-Vernetzung, die Topologie, unberücksichtigt gelassen. Das hat zur Folge, daß für Kommunikationsvorgänge, wie z.B. das Versenden einer Kopie einer virtuellen Seite, immer der gleiche Zeitbedarf angenommen wird. Natürlich entspricht das nicht der Realität. Auf zwei Arten kann die Struktur des Verbindungsnetzwerks einen entscheidenden Einfluß auf die Zeitdauer haben, die für das Versenden einer Seitenkopie erforderlich ist, sofern nicht eine vollständige Vernetzung der Prozessoren (*crossbar*) vorliegt:

- Nachrichten müssen über mehrere Etappen (CPUs) hinweg verschickt werden, wobei unterschiedlich lange Wegstrecken unter Einsatz unterschiedlich vieler Kommunikationsprozessoren (als Relais-Stationen) vorkommen können.
- Auf von mehreren Nachrichten gleichzeitig genutzten Verbindungen kann es zu lokalen Engpässen kommen, die entweder durch Sequentialisierung der verschiedenen Datenströme oder durch Verweigern der Weitergabe seitens des Kommunikationsprozessors behandelt werden können. Beide Möglichkeiten verlangsamen oder verhindern die Kommunikation über das Netz.

Weiterhin hat die Prozessortopologie einen signifikanten Einfluß auf die Fehlertoleranz des Systems bezüglich des Ausfalls eines oder mehrerer (Kommunikations-) Prozessoren. Liegt beispielsweise eine Baumstruktur vor, bedeutet der Ausfall eines Zwischenknotens die Abkopplung des gesamten darunterliegenden Teilbaums vom Rest des Netzes. Virtual-Shared-Memory-Systeme benutzen ohne direktes Eingreifen des Programmierers das Netzwerk für den Austausch von Kopien virtueller Seiten, wobei davon ausgegangen wird, daß jede CPU mit jeder anderen kommunizieren kann. Ein kompletter Ausfall der Verbindung zwischen zwei Prozessoren führt also in VSM-Systemen in der Regel zum Absturz. Für das Virtual-Shared-Memory-Konzept geeignete Prozessornetze sollten also eine ausreichende Wegeredundanz enthalten, um die Möglichkeit des Umleitens von Nachrichten im Fehler- oder Überlastungsfall zu garantieren.

In dieser Arbeit wird unter anderem ein Werkzeug vorgestellt, daß für verschiedene Netzwerktopologien Kommunikationswege und die Belastungen der einzelnen Netzverbindungen beispielhaft darstellt. Topologie und Routing-Schema sind dabei jedoch nicht den darzustellenden, das Systemverhalten beschreibenden Eingabedaten entnommen, sondern werden vom Anwender interaktiv und damit nachträglich angegeben. Somit bleiben bei der Anwendung dieses Hilfsmittels natürlich auch die Rückwirkungen der Netzbelastung auf die Kommunikationsgeschwindigkeiten unberücksichtigt. Dennoch kann mit diesem Werkzeug ein Eindruck davon vermittelt werden, wie der Austausch von Seitenkopien in einem konkreten Prozessornetz realisiert werden kann, wobei mögliche Problemstellen zutage treten können.

Kapitel 4

Bisherige Arbeiten im Bereich der Visualisierungswerkzeuge

Auf dem Gebiet der Visualisierung der Abläufe in Mehrprozessorsystemen gibt es bereits eine Reihe von Werkzeugen. Leider ist die Darstellung der Speicheraktivitäten, wenn sie überhaupt enthalten ist, meistens auf die Visualisierung von direkten Kommunikationsvorgängen (*send-receive*-Kombinationen) beschränkt. Tatsächlich ist auf der Grundlage des reinen Message-Passing-Modells auch keine weitergehende Darstellung sinnvoll, da alle übrigen Speicheraktivitäten lokale Operationen der Prozessoren auf ihren lokalen Speichern und im Hinblick auf die Probleme der parallelen Abläufe in Mehrprozessorsystemen uninteressant sind.

Die Schaffung eines virtuellen Adreßraumes, der in Speicherseiten aufgeteilt ist, ist der Grund für das Bedürfnis nach genaueren Visualisierungsmöglichkeiten, da weitere (virtuelle) Komponenten ins Spiel kommen und somit das Gesamtsystem erheblich komplexer wird. Da jede virtuelle Seite zu jeder Zeit "Gemeingut" ist, d.h. grundsätzlich allen Prozessoren des Systems zur Verfügung steht, ist selbst der lokale Zugriff auf eine Seite keine private Angelegenheit des zugreifenden Prozessors mehr. Zum Beispiel erfordert der Lesezugriff auf eine lokale Kopie einer Speicherseite die Invalidation dieser Kopie durch einen Prozessor, der Schreibzugriff auf dieselbe virtuelle Seite erhalten will.

Virtual-Shared-Memory-Systeme erfordern also, im Vergleich zur reinen Darstellung der Interprozessor-Kommunikation, aus zwei Gründen eine weitreichendere Visualisierung: Erstens wird die Komplexität des Systems durch die neuen, virtuellen Komponenten und deren Beziehungen zueinander erhöht, zweitens haben durch das virtuelle Speicherprinzip alle, auch scheinbar lokale Operationen globale Bedeutung und müssen darstellbar sein. Es folgt eine Übersicht über vorhandene Visualisierungswerkzeuge, die Betrachtungen der Prozessor- und Speicheraktivitäten erlauben.

4.1 Überblick über die Literatur

Schon bei der Untersuchung von Skalarprozessorsystemen ergab sich die Notwendigkeit, das Speicher-Zugriffsverhalten von bestimmten Algorithmen unter bestimmten Systemvoraussetzungen optisch aufbereitet darzustellen. Insbesondere die Untersuchung höherer Speichermodelle mit abstrahierter Sicht des physikalischen Speichers wie das Modell des virtuellen Speichers erforderten wegen ihrer Komplexität geeignete Beobachtungsinstrumente. In diesem Zusammenhang wurden sogenannte *Raum-Zeit-Diagramme* verwendet.

Es gibt eine ganze Reihe von Ansätzen zur Visualisierung der Abläufe in Mehrprozessorsystemen in bezug auf Prozessoraktivitäten und Prozeßwechsel. Die meisten von ihnen legen ein System mit verteiltem Speicher und das Konzept des Nachrichtenaustausches (Message Passing) zugrunde. Die Visualisierung der Speicheraktivitäten beschränkt sich damit meistens auf die Darstellung der Prozessortopologie in Verbindung mit den Aktivitäten des implementierten Kommunikationsnetzwerks. Es handelt sich also um Zeitpunktdarstellungen, die meistens in einer Animation oder für bestimmte Zeitpunkte die Übertragung einer Nachricht durch Hervorheben der entsprechenden Verbindung und der beteiligten Knoten darstellen. Im folgenden werden einige interessante Arbeiten kurz vorgestellt, die sich mit visueller Darstellung der Vorgänge in Mehrprozessorsystemen beschäftigen.

In [20] findet sich eine kurze Übersicht über einige Visualisierungswerkzeuge, derselbe Autor beschreibt in [21] unter anderem spezielle Werkzeuge zur Veranschaulichung der Speicheraktivitäten von Mehrprozessorsystemen mit gemeinsamem Speicher. Eine allgemeingültige Klassifizierung von Visualisierungswerkzeugen für parallele Programme wird in [3] durchgeführt. Es werden die Kategorien *Datenorientierte Visualisierung* (Darstellung der Ergebnisse), *Programm- oder Systemorientierte Visualisierung* (Darstellung des Ablaufs auf der Ebene des Quellcodes bzw. des Systems) und *Problem-Visualisierung* (Veranschaulichung der algorithmischen Funktionalität) eingeführt. In den folgenden Teilen der Arbeit wird das Visualisierungswerkzeug VISTOP vorgestellt, das Teil des TOPSYS-Systems ist, einer Umgebung, die mehrere Hilfsmittel zur Analyse paralleler Programme bereitstellt. Die markanteste Eigenschaft von TOPSYS ist, daß alle Werkzeuge, einschließlich der Visualisierungsumgebung, *während* des Laufs des zu untersuchenden Programms eingesetzt werden. In [7] wird gezeigt, wie objektorientierte Programme graphisch veranschaulicht werden können, wobei eine Erweiterung auf parallele, objektorientierte Programme in Aussicht gestellt wird. In [23] werden "concurrency maps" vorgestellt, die in Form von prozeßorientierten Zeitlinien Abhängigkeiten und Synchronisationspunkte mit Hilfe von Abhängigkeitspfeilen aufgezeigen. In [10] geht es um den Versuch einer Kombination des weiter unten beschriebenen Visualisierungswerkzeugs ParaGraph mit einem symbolischen Debugger. Die Autoren von [24] sind der Meinung, daß zur Visualisierung von Abläufen in massiv-parallelen Systemen ein auf den Prozessoren parallel arbeitendes Werkzeug erforderlich ist, um "Flaschenhälse" bei der Datenaquisition zu vermeiden und so beliebige Skalierbar-

keit bezüglich der Prozessorzahl zu gewährleisten. Sie präsentieren eine auf dem Client-Server-Modell basierende Lösung für ein hypothetisches, UNIX-gestütztes, massiv-paralleles Rechnermodell.

Weitere bemerkenswerte und interessante Visualisierungswerkzeuge sind in [1, Kapitel 2] beschrieben und sollen hier nicht mehr erwähnt werden. Im folgenden werden einige weitere typische wie interessante Analysehilfsmittel genauer vorgestellt, um das Spektrum der vorhandenen Möglichkeiten auszuleuchten.

4.2 Raum-Zeit-Diagramme

Die visuelle Darstellung von Speicheraktivitäten ist nicht erst seit dem Beginn der Untersuchungen von Mehrprozessorsystemen interessant. So kamen beispielsweise bei der Untersuchung des Verhaltens von Einprozessor-Systemen mit virtuellem Speicher Visualisierungswerkzeuge zum Einsatz, die in der Literatur als *Raum-Zeit-Diagramme* (strip charts) bekannt sind [15, S. 22]. Es handelt sich dabei um zweidimensionale Koordinatensysteme, bei denen eine Achse mit der Zeit und die andere Achse mit Adreßraumeinheiten beschriftet ist. Die Zeitachse kann dabei in Sekunden, Anzahl ausgeführter Instruktionen oder Anzahl der Adreßzugriffe eingeteilt werden. Die Adreßraum-Achse hingegen kann je nach Darstellungszweck mit Speicheradressen, Speicherseiten oder aber symbolischen Variablennamen beschriftet sein.

Diese Art der Darstellung ist sehr aussagekräftig. Es entstehen visuell sehr gut zu erfassende Muster, die sehr stark von der Betrachtung einzelner Zugriffe abstrahiert sind und globale Aussagen über das Zugriffsverhalten eines Algorithmus erlauben. Insbesondere Speicherlokalitäten, aber auch allgemeinere Aspekte der Zugriffsstrategien können auf diese Weise sehr einfach erfaßt werden. Allerdings können Raum-Zeit-Diagramme nicht ohne weiteres auf Mehrprozessorsysteme erweitert werden, da die Einführung einer zusätzlichen Dimension für die Prozessoren in der Darstellung erforderlich wird.

K. Mevissen verwendet Diagramme, bei denen die Zeitachse mit Adreßzugriffen und die Raumachse mit symbolischen Feldnamen beschriftet sind, zur Untersuchung von Speicherlokalitäten in Systemen mit virtuellem Speicher [15, S. 22]. Abbildung 4.1 zeigt ein derartiges Raum-Zeit-Diagramm einer Matrixmultiplikation. Die (waagerechte) Raumachse ist mit den zusammenhängenden Speicherbereichen der Eingabematrizen (A, B) und der Ergebnismatrix (C) beschriftet. Mit Hilfe derartiger Darstellungen können Algorithmen auf sehr anschauliche Weise charakterisiert werden.

Die später von J. Dongarra erschienene Arbeit greift diese Darstellung auf und beschreibt ein Post-Mortem-Werkzeug, das auf die gleiche Weise Speicherzugriffsmuster erzeugt [4]. Dieses Werkzeug dient zur Untersuchung des Verhaltens verschiedener Speicherhierarchie-Schemata auf Rechnern mit gemeinsamem Speicher.

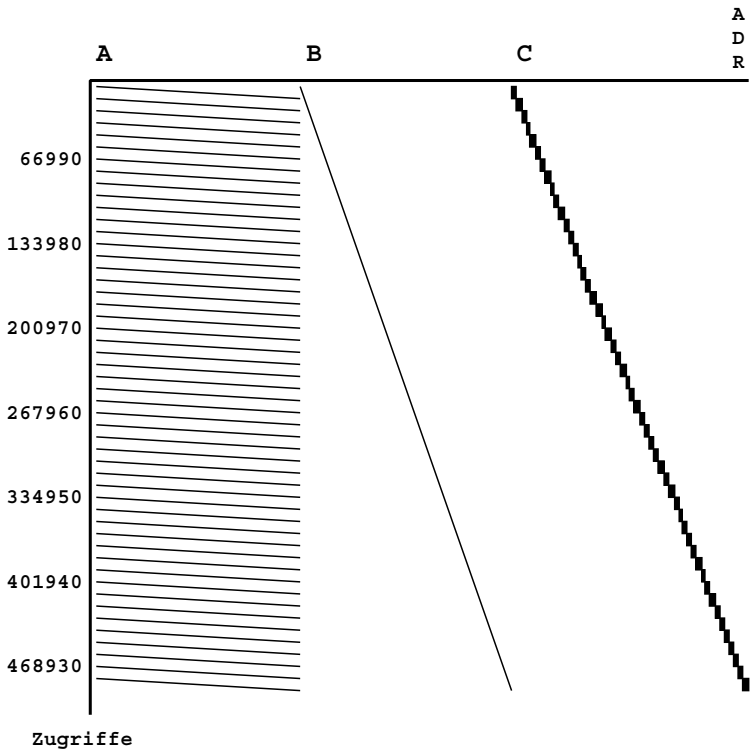


Abbildung 4.1: Typisches Raum-Zeit-Diagramm, hier: Matrixmultiplikation (nach [15])

R. Berrendorf untersucht das Zugriffsverhalten in Virtual-Shared-Memory-Systemen mit Hilfe eines Simulators und stellt seine Ergebnisse durch ähnliche Diagramme dar [2]. Er verwendet dabei Einteilungen nach der Zeit bzw. nach der Speicheradresse. Speziell zur Untersuchung von Systemen mit virtuell gemeinsamem Speicher ist die Lokalität von Zugriffen besonders wichtig, damit das System effizient arbeitet.

4.3 Hyperview: Visualisierung im Baukastensystem

Hyperview [14] ist ein Visualisierungswerkzeug, das in erster Linie für Hypercube-Topologien entwickelt wurde, da es ursprünglich von *Seecube*, einer Visualisierungsumgebung für den NCUBE-Hypercube inspiriert war. Es konzentriert sich in seinen Darstellungsmöglichkeiten stark auf die Visualisierung der verschiedenen Aspekte des Datenaustausches zwischen den Prozessoren über das Verbindungsnetzwerk, eine Tatsache, die es im Kontext dieser Arbeit besonders interessant macht. Hyperview ist ein Post-Mortem-Analysewerkzeug, das mit einer Protokolldatei arbeitet, die als Trace-Datei eines realen Systemlaufes oder als Ausgabe eines Simulationslaufes er-

zeugt werden kann. Es existieren zwei Ausbaustufen des Werkzeugs. Bei beiden wurde das Prinzip der Trennung von Datenanalyse und -visualisierung angewendet.

In der ersten Version können neben anderen Darstellungsmöglichkeiten vier Sichtweisen zur Verdeutlichung der Verbindungstopologie ausgewählt werden, in denen jeweils eine feste Prozessoranordnung vorgegeben ist:

- die Anordnung, die auf natürliche Weise die *Hypercube-Topologie* wiedergibt,
- ein sogenanntes *"Pascal Display"*, das die Einbettung logischer Bäume in den Hypercube deutlich machen kann,
- ein *"FFT-Display"*, das aus mehreren Spalten besteht, die jeweils alle Prozessoren übereinander angeordnet enthalten; in solchen Darstellungen können Topologien wie die für die Schnelle Fourier-Transformation geeignete Schmetterlings-Struktur anschaulich dargestellt werden.
- ein sogenanntes *"GRAY CODE display"*, es zeigt insbesondere die Verbindungen zwischen den beiden (n-1)-dimensionalen Teilwürfeln eines n-dimensionalen Hypercube.

Zusätzlich kann z.B. ein Histogramm erzeugt werden, das die Längen der Nachrichtenwarteschlangen für jeden Knoten als Zeitpunktdarstellung veranschaulicht.

Da die Autoren der Visualisierungsumgebung den Eindruck erhielten, daß das Werkzeug nur sehr schwer erweiterbar war (in bezug auf zusätzliche Darstellungsmöglichkeiten, mehr Möglichkeiten innerhalb der einzelnen Ansichten), wurde in einer zweiten Ausbaustufe eine gründliche Neustrukturierung des Werkzeugs vorgenommen. Gleichzeitig sollte die Flexibilität des Sytems in bezug auf verschiedene zu untersuchende Architekturen und Anwendungen wesentlich erhöht werden. Daher wurden insbesondere die einzelnen Elemente des Werkzeugs noch stärker voneinander isoliert. Die Hauptkomponenten des neuen HyperView sind:

- die *Datenfilter*, die die eingehenden Protokolldaten aufbereiten und die für die Darstellung gewünschten Daten extrahieren,
- die ganz allgemein konzipierten *Darstellungswerkzeuge* wie z.B. Balkendiagramme, Matrixdarstellungen, Zeigerinstrumente oder sogenannte "Graphen", unter denen der Autor Topologieillustrationen versteht, die aus Knoten- (Kreisen) und Verbindungselementen (Linien) bestehen,
- die *Bedienungsoberfläche*, mit der die einzelnen Filter und Darstellungselemente konfiguriert bzw. angefordert werden können sowie die Schnittstellen zwischen ihnen festgelegt werden.

Diese sehr allgemein gehaltenen Elemente garantieren ein Höchstmaß an Flexibilität in bezug auf Möglichkeiten in der Visualisierung, so daß gleichsam nach Art eines Baukastensystems speziell zugeschnittene Darstellungen erzeugt werden können.

Auch die Erweiterbarkeit des Systems ist durch klar definierte Schnittstellen zwischen den einzelnen Komponenten garantiert.

Die Filter ermöglichen z.B. die Untersuchung der Prozessorzustände, die Auslastung des Systems und des Zeitanteils ihrer Beschäftigung mit verschiedenen Prozessen. In bezug auf Speicher- bzw. Kommunikationsaktivitäten sind u.a. folgende Analysedaten darstellbar:

- Aussagen über Anzahl der Nachrichten und Größe des Nachrichtenaufkommens zwischen je zwei Prozessoren (z.B. durch matrixartige Schaubilder)
- Informationen über die Anzahl und die Gesamtgröße der gesendeten Nachrichten jedes Prozessors (z.B. durch Histogramme)
- Aussagen über das durchschnittliche Nachrichtenaufkommen des Gesamtsystems in einem gleitenden Zeitintervall (z.B. als Zeitachsendarstellungen)

Die extreme Flexibilität ist das hervorstechende Merkmal des HyperView-Systems. Jedoch kann die Vielfalt der Kombinationsmöglichkeiten für ungeübte Benutzer, die noch keine Erfahrung darüber besitzen, welche Darstellungsmethode sich für bestimmte zu visualisierende Daten eignet, leicht zu Verwirrungen führen. Dabei ist zu bedenken, daß eventuell auch unsinnige oder zumindest ungünstige Kombinationen von den darzustellenden Daten und der Darstellungsmethode erzeugt werden können.

4.4 ParaGraph: vielfältige Einblicke in die Dynamik paralleler Programme

Eine ganz andere Philosophie verfolgt die Visualisierungsumgebung ParaGraph, die ebenfalls das Verhalten paralleler Programme auf Systemen mit nachrichtengestützter Kommunikationsstruktur graphisch darstellt [6, 22]. Das Werkzeug erlaubt keine Benutzer-definierten Kombinationen von darzustellenden Daten mit Darstellungswerkzeugen in Form eines Baukastensystems, wie Hyperview (Abschnitt 4.3) es tut. Vielmehr bietet es eine Vielzahl von Darstellungsmöglichkeiten, die alle auf die Visualisierung ganz bestimmter Informationen speziell zugeschnitten und gezielt angepaßt sind. Dafür erlauben die einzelnen Ansichten jedoch keine weitreichenden Manipulationen, eine gewählte Darstellung kann kaum modifiziert werden. Falls der Benutzer eine andere Sichtweise einnehmen will, muß er auf ein anderes der zur Zeit insgesamt 25 Visualisierungsfenster übergehen. In der Tat ist die Bereitstellung vieler unterschiedlicher Darstellungshilfsmittel das wesentliche Charakteristikum und ein erheblicher Vorzug des ParaGraph-Systems; mit ihnen können verschiedene Perspektiven alternativ oder gleichzeitig ausgewählt werden. In konsequenter Fortsetzung des oben genannten Prinzips bietet ParaGraph die Möglichkeit, leicht neue

Darstellungsmethoden zu integrieren, die spezielle Sichtweisen auf Programme erlauben und dabei die besonderen Eigenheiten der an diesen Programmen interessierenden Aspekte berücksichtigen und in Kombination mit vorhandenen und/oder anderen neuen Visualisierungsmöglichkeiten neue Gesamteindrücke vermitteln.

Auf Schwierigkeiten stößt der erfahrenere Anwender des Systems dann, wenn er mit Hilfe einer spezifischen Darstellungsmethode, die sich vielleicht für seine Zwecke als besonders geeignet erwiesen hat, bestimmte Phänomene "nur ein bißchen anders" beleuchten möchte. Hier reichen dann die Möglichkeiten des speziellen Fensters unter Umständen nicht aus. So gibt es beispielsweise in den Zeitachsendarstellungen in ParaGraph keine Zooming-Möglichkeit, wodurch eine unmittelbare Detailuntersuchung nicht möglich ist. Auch die parallele Verwendung anderer visueller Perspektiven hilft an dieser Stelle nicht weiter, da das zu untersuchende Phänomen dadurch zwar aus verschiedenen Blickwinkeln betrachtet werden kann, die gewünschten Details aber dennoch nicht in der erforderlichen Art und Weise herausgestellt werden können.

Ein weiterer wichtiger Punkt in der Philosophie des ParaGraph-Systems ist die Konzentration auf dynamische Darstellungen. Damit ist gemeint, daß vorzugsweise Zeitpunktdarstellungen unterstützt werden, welche für bestimmte Zeitpunkte, als Einzelschrittfolge oder als fortlaufende Animation angezeigt werden. Die Autoren drücken diese Philosophie wie folgt aus [6, Seite 5]:

> "In designing ParaGraph, we have adopted a more dynamic approach whose conceptual basis is algorithm animation. We see the tracefile as a script to be played out, visually re-enacting the original live action of parallel program execution in order to provide insight into the program's dynamic behavior."

Hauptziel von ParaGraph ist also die Vermittlung einer intuitiven Einsicht in das dynamische Verhalten von parallelen Programmen. Auch hieraus ist zu entnehmen, daß die Extraktion von Detailinformationen nicht zu den Hauptzielen von ParaGraph gehört, sondern vielmehr die Herausstellung genereller Aspekte des Lauf- und Kommunikationsverhaltens durch sich dynamisch entwickelnde Muster. Diese Art der Ablauf- und Kommunikationsanalyse stellt natürlich hohe Anforderungen an die Phantasie und an das Vermögen, Muster und Strukturen aus den Bilderbewegungen heraus in seiner Vorstellung zu entwickeln und Deutungen abzuleiten.

ParaGraph wurde in erster Linie dazu entwickelt, Protokolldaten, die als Trace eines realen Systemlaufs erzeugt wurden, sichtbar zu machen. Die Visualisierungsumgebung benutzt dabei ein Eingabeformat, das von PICL (Portable Instrumented Communication Library) bei der Ausgabe von Trace-Informationen verwendet wird. PICL ist ein für viele Architekturen erhältliches Programm-Instrumentierungswerkzeug, mit dem signifikante Ereignisse beim Ablauf paralleler Programme protokolliert werden können. Die Verwendung dieses Datenformats garantiert eine weitgehende Portierbarkeit des Werkzeugs.

4.5 Akustische Darstellung: Sinnvolle Ergänzung zur Visualisierung?

Nun soll ein Analysehilfsmittel vorgestellt werden, das strenggenommen nicht unter das Thema dieses Kapitels fällt, da es sich nicht der optischen, sondern der akustischen Darstellung der Abläufe widmet. Es werden die Ergebnisse aus der Arbeit [5] beschrieben. Die Darstellung fällt etwas ausführlicher aus, da die beschriebenen Prinzipien in dem Zusammenhang dieses Kapitels neu und vielleicht etwas ungewohnt sind.

Ein Argument aus dieser Arbeit für die Verwendung von Geräuschen ist die Tatsache, daß Unregelmäßigkeiten in einer Klangkulisse auch beim passiven Zuhören bemerkt werden, ohne daß man konzentriert "hinhören" muß. Zum anderen können Klangmuster (wie z.B. Melodien) leicht erkannt werden. Dies gilt zwar auch für optische Muster, jedoch können verschiedene Phänomene mit Hilfe der beiden unterschiedlichen Medien verschieden deutlich zutage treten. Zum Beispiel läßt sich die Melodie einer Bach-Fuge wesentlich leichter durch Hören einer Interpretation als durch bloßes Lesen des Notentextes erfassen. Umgekehrt sind unser menschliches Auge wie auch unser Ohr in unterschiedlicher Weise in der Lage, Auslassungen in Mustern zu nivellieren, was zur Folge hat, daß solche "Fehlstellen" dem Beobachter oder Zuhörer nicht mehr bewußt auffallen. Man kann somit die Chance erkennen, durch kombinierte Anwendung beider Hilfsmittel die Wahrscheinlichkeit für die Wahrnehmung bestimmter Unregelmäßigkeiten in den Darstellungsmustern zu erhöhen.

Der Autor der o.g. Arbeit weist zusätzlich auf die natürliche Eignung von Musik zur Repräsentation paralleler Ereignisse und zeitlicher Abfolgen hin. So könnten verschiedene Ereignisse verschiedenen Tönen zugeordnet werden. Gleichzeitig auftretende Ereignisse würden dann einen Akkord ergeben, in dem sowohl die einzelnen Töne zutage treten als auch ein Gesamteindruck der parallel auftretenden Bestandteile entsteht. Selbstverständlich können Klänge in natürlicher Weise die zeitlichen Abfolgen und die absoluten Längen von Ereignissen durch Tempi und Pausen wiedergeben.

Es stehen folgende Dimensionen der Klangwelt zur Repräsentation von Systemeigenschaften zur Verfügung:

- *Tonhöhe*
- *Tondauer*
- *Intensität*
- *Klangfarbe*
- *Position* des Klangs, definiert durch die unterschiedlichen Kanalintensitäten bei der Stereo-Wiedergabe

Auch höhere Eigenschaften, die schon in den Bereich der Musik fallen, wie Takt, Rhythmus, Melodie, Harmonie usw. können verwendet werden. So kann ein harmonisches Zusammenspiel verschiedener Noten eine gültige Konstellation unterschiedlicher Ereignisse bedeuten. Drei Systemeigenschaften werden in [5] auf die oben aufgezählten Dimensionen abgebildet: die Prozessor-Kommunikation, die Prozessor-Auslastung und der Kontrollfluß.

4.5.1 Darstellung der Prozessorkommunikation

Bei dieser Analysemethode wird jedem Prozessor eine bestimmte Klangfarbe zugeordnet. Verschiedene Nachrichten, die von einem Prozessor ausgesendet werden, erklingen in verschiedenen Tonhöhen in der dem Prozessor zugeordneten Klangfarbe. Ein Nachrichten-Ton ertönt so lange, bis die Nachricht empfangen wird. Der Sende- sowie der Empfangsvorgang werden von kurzen Tonfolgen begleitet; der Sendevorgang spielt dabei eine Melodie an, der Empfangsvorgang vervollständigt sie. Auf diese Weise soll durch das Senden eine Art von psychischer Spannung aufgebaut werden, die sich durch das Empfangen wieder auflöst. Zusätzlich kann der Stereo-Effekt ausgenutzt werden, indem Sende- und Empfangsmelodien aus verschiedenen Kanälen ertönen.

4.5.2 Akustische Repräsentation der Auslastung von Prozessoren

Die grundsätzliche Idee hier ist die Erzeugung eines bestimmten Tones, der solange erklingt, wie ein Prozessor unbeschäftigt (idle) ist. Damit "normale" von unerwünschten Idle-Zeiten unterschieden werden können, ist es möglich, einen Schwellwert für anzuzeigende Leerlauf-Zeiten zu definieren. Ein unerwünschter Leerlauf wird dann durch einen speziellen Ton in der Klangfarbe des "faulen" Prozessors eingeläutet, auf den dann der oben erwähnte Dauerton folgt, dessen Lautstärke stetig zunimmt. Durch Variation der Lautstärke der Veranschaulichung kann die Feinheit der Registrierung eingestellt werden: bei hoher Lautstärke werden selbst kurze Idle-Zeiten erfaßt, bei leisem Abspielen der Darstellung nur noch sehr lange. Auch hier ist die Stereo-Technik nützlich einzusetzen: die gesamte Leerlaufzeit eines Prozessors kann dadurch deutlich gemacht werden, daß die gerade beschriebenen Klangeffekte vom linken Kanal immer weiter zum rechten wandern, je höher die aufsummierten Idle-Zeiten werden.

4.5.3 Darstellung des Kontrollflusses

Hier wird, wie weiter oben schon angedeutet wurde, jedem Prozessor eine Klangfarbe und jedem Ereignis ein bestimmter Ton zugeordnet. Der Ton ist eine Funktion des Ereignistyps und kann je nach Definition verschiedene Bedeutungen haben wie z.B.

die Identität des sendenden/empfangenden Prozessors bei Sende-/Empfangsereignissen, den Typ der Nachricht oder die Art eines vom Programmierer erzeugten Ereignisses, das Kontrollzwecken dient.

4.5.4 Bewertung des akustischen Ansatzes

Der Autor der erwähnten Arbeit schlägt vor, akustische Analysehilfsmittel *zusätzlich* zu optischen zu verwenden. In der Tat liegt hier der Hauptvorteil des Ansatzes, da es damit möglich ist, mehr Informationen gleichzeitig aufzunehmen als mit nur einer Methode. Ob das allerdings auch innerhalb der Aufnahmefähigkeit eines möglichen Anwenders liegt, ist fraglich. Es ist anzunehmen, daß zumindest sehr viel Übung nötig ist, um sich in unproblematische und problematische Fälle "hineinzuhören", genauso wie man es in der Regel nicht sofort versteht, beim Hören einer Sinfonie alle Instrumente, deren Melodien und Rhythmen zu erkennen und die verschiedenen Aspekte zueinander in Beziehung zu setzen und dennoch den Kunstgenuß des gesamten Werkes nicht zu vernachlässigen. Ein Äquivalent zur Zeitachsendarstellung in Visualisierungssystemen, bei der ganze Zeiträume gleichsam auf einen Blick erfaßt werden können, existiert in der Hörbarmachung von Systemverhaltensstrukturen nicht.

Das in [5] vorgestellte und hier umrissene System ist ein Post-Mortem-Analysewerkzeug mit allen damit verbundenen Vorteilen wie Entkopplung vom Systemlauf, beliebiger Wiederholbarkeit der Analyse mit verschiedenen Darstellungsparametern usw. Die Verwendung des PICL-Formates (Abschnitt 4.4) für die einzulesenden Protokolldateien erlaubt eine weitgehende Portierbarkeit des Werkzeugs. Die Erzeugung von MIDI[1]-Ausgabedaten zur Reproduktion der Klangmuster erlaubt die zeitliche Synchronisation mit Visualisierungswerkzeugen, ein Vorteil, der besonders ins Gewicht fällt, zumal, wie oben erwähnt, in der Kombination mit visuellen Methoden das Haupteinsatzgebiet dieses Ansatzes liegt.

4.6 Die Visualisierungsumgebung PARvis

PARvis [1, 17] wurde in erster Linie zur Visualisierung der Prozeßwechselaktivitäten paralleler Programme in Multiprozessorumgebungen entwickelt. Dieses Werkzeug wertet Protokolldaten aus, die als Trace-Informationen während eines realen Systemlaufs oder als Ausgabe eines Simulators erzeugt wurden. Die Aufrufoberfläche *PARtools* vereinigt in sich die Bedienungselemente von PARvis und der Systeme PARsim und PARbench [1, 17] (Abbildung 4.2). PARsim ist ein Simulator zur Erforschung des Prozeßwechselverhaltens von Multiprozessorsystemen mit verschiedenen Scheduling-Strategien [16]. Dieser Simulator wurde insbesondere zur Untersuchung

[1] Musical Instrument Digital Interface

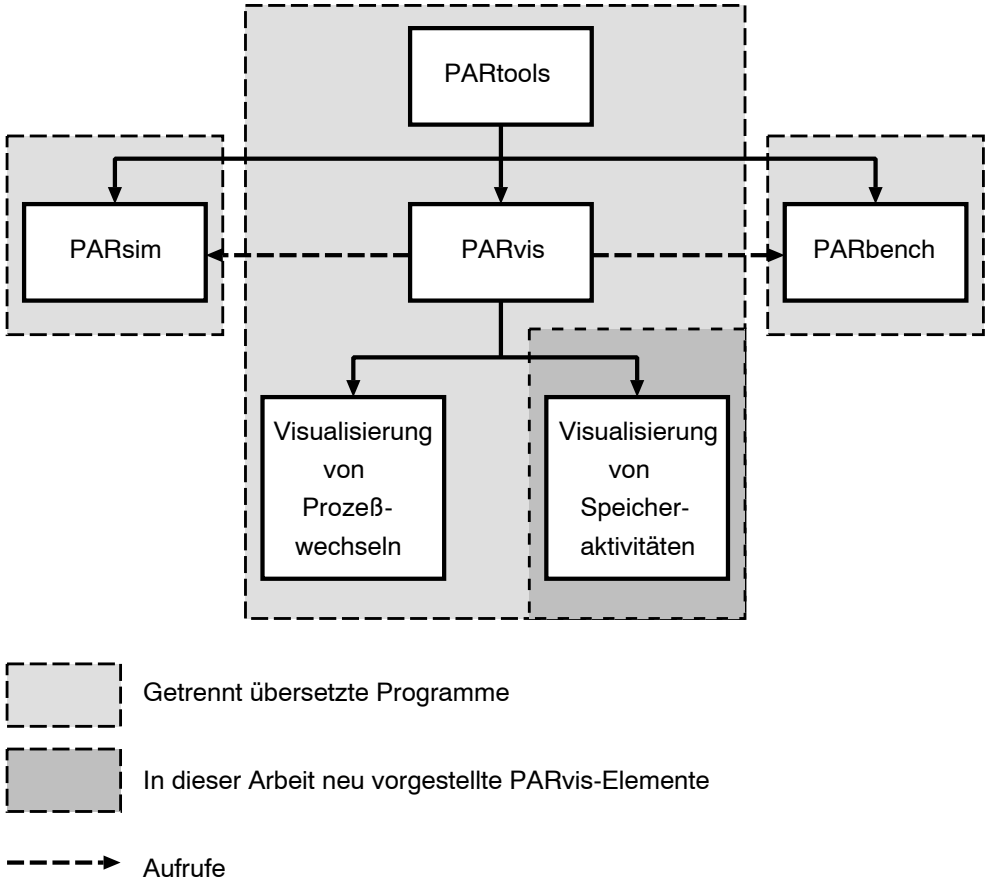


Abbildung 4.2: Überblick über die PARtools-Umgebung

der Vorteile eines kooperativen Scheduler-Konzeptes eingesetzt. PARbench hingegen ist ein Benchmark-System [19, 18], mit dem parallele Programme auf realen Multiprozessorsystemen im Rahmen einer Meßumgebung analysiert werden können.

PARvis stellt vielfältige Möglichkeiten zur Untersuchung des Ablaufs paralleler Programme zur Verfügung. Der Anwender kann sich auf einer virtuellen Zeitachse, die den Zeitraum des protokollierten Systemlaufs widerspiegelt, sowohl vorwärts (entsprechend dem natürlichen Zeitablauf) als auch rückwärts frei bewegen, wobei auch gezielte Sprünge zu bestimmten Zeitpunkten möglich sind. Für die jeweilige Stelle auf dieser Zeitachse, an der sich der Benutzer gerade befindet, wird der dazugehörige Zustand des zu untersuchenden Systems intern repräsentiert. Auf diese Weise werden dann während aller Bewegungen auf der Zeitachse alle Darstellungsfenster automatisch aktualisiert. Zu diesen Darstellungsfenstern gehören Zeitpunktdarstellungen, die bei kontinuierlicher Fortbewegung auf der virtuellen Zeitachse eine Art von animierter Simulation des Systemlaufs aus bestimmten Perspektiven heraus ermöglichen. Dazu gehören aber auch Zeitachsendarstellungen, mit denen das Verhalten des Systems mit einem Blick über einen bestimmten Zeitraum hinweg erfaßt werden

kann. Bei zeitlicher Fortbewegung können anhand des "Wachsens" von Zeitlinien besonders aufschlußreiche Einsichten gewonnen werden, die sich aus der Kombination von Überblicksperspektive und Animation ergeben. Auch die vierte Klasse der Darstellungsmethoden ist vertreten: Statistiken über Systemdaten können in unterschiedlicher Form angezeigt werden. Auch hier erlauben automatische Aktualisierungen besondere Beobachtungsmöglichkeiten in bezug auf die Dynamik der Systemdaten.

Beim Design dieser Visualisierungsumgebung wurde besonderer Wert darauf gelegt, die einzelnen Werkzeugelemente flexibel und komfortabel zu gestalten, um dem Anwender ein Höchstmaß an Einsichtmöglichkeiten mit Hilfe jedes einzelnen Werkzeugs an die Hand zu geben. Das bedeutet, daß ein erfahrener Benutzer, der eine besonders für die von ihm zu untersuchenden Phänomene geeignete Darstellungsart gefunden hat, mit dieser auch weitreichende Möglichkeiten der Betrachtung hat, insbesondere in bezug auf Extraktion von Detailinformationen. So enthalten z.B. sämtliche Zeitachsendarstellungen umfangreiche Funktionen, mit denen das betrachtete Zeitfenster beliebig den eigenen Wünschen angepaßt werden kann (z.B. Zooming- und Verschiebefunktionen).

4.6.1 Grundprinzipien und Bedienungsphilosophie

Jede Arbeit mit PARvis beginnt natürlicherweise mit dem Einlesen der Protokolldatei, wobei eine Art Vorauswertung vorgenommen wird, die weiter unten beschrieben wird. Dieses Einlesen geschieht im Hintergrund, d.h. der Anwender kann noch während des Einlesevorganges mit der Analyse der bisher eingelesenen Informationen beginnen. Bei dem Einlesevorgang wird in vom Anwender festlegbaren Abständen der Systemzustand in der Visualisierungsumgebung abgelegt. Diese Daten dienen bei der Bewegung auf der virtuellen Zeitachse als "Trittsteine". Bei wahlfreier, sprungartiger oder rückwärts gerichteter Bewegung auf der Zeitachse wird der simulierte Systemzustand auf den der letzten Stützstelle vor dem Zielzeitpunkt gesetzt. Zur genauen Herstellung des Systemzustands des gewünschten Zeitpunktes müssen dann noch die restlichen Eintragungen ab der entsprechenden Stelle aus der Protokolldatei herausgelesen und im Visualisierungssystem registriert werden.

Zur Analyse der Systemdaten stehen mehrere Darstellungsmöglichkeiten zur Verfügung, die weiter unten im Detail beschrieben werden. Alle Darstellungsfenster können natürlich - ein Vorteil, den die X Window-Umgebung liefert - gleichzeitig auf dem Bildschirm dargestellt werden, so daß eine Betrachtung von Sachverhalten gleichzeitig aus verschiedenen Perspektiven möglich ist.

Die *Zeitpunktdarstellungen* können ohne weitere Voraussetzungen für alle schon eingelesenen Protokolldaten abgerufen werden, wobei sämtliche Möglichkeiten zur Wahl der dargestellten Zeitpunkte in Anspruch genommen werden dürfen. Bevor jedoch eine *Zeitachsendarstellung* oder eine *statistische Auswertung* dargestellt werden kann, muß eine sogenannte Historienliste erzeugt werden. Das heißt, daß sämtliche Ereignisdaten eines gewünschten Zeitraumes in den Speicher des PARvis-Systems

geladen werden müssen. Dazu führt der Benutzer eine Schritt-für-Schritt-Bewegung oder eine kontinuierliche Bewegung durch das entsprechende Zeitintervall durch. Falls dabei bereits Visualisierungsfenster geöffnet sind, können diese Bewegungen dynamisch reflektiert werden, d.h. Zeitpunktdarstellungen z.B. visualisieren das zeitliche Voranschreiten durch Erneuerungen ihrer Darstellungen in Form von Animationen. Auch die Beobachtung des sukzessiven Aufbaus von Zeitachsendarstellungen während der Bewegung durch das gewünschte Zeitintervall kann wertvolle Informationen über die Dynamik des Systems liefern. Diese Bedienungsphilosophie des PARvis-Systems ist noch einmal in Abbildung 4.3 veranschaulicht.

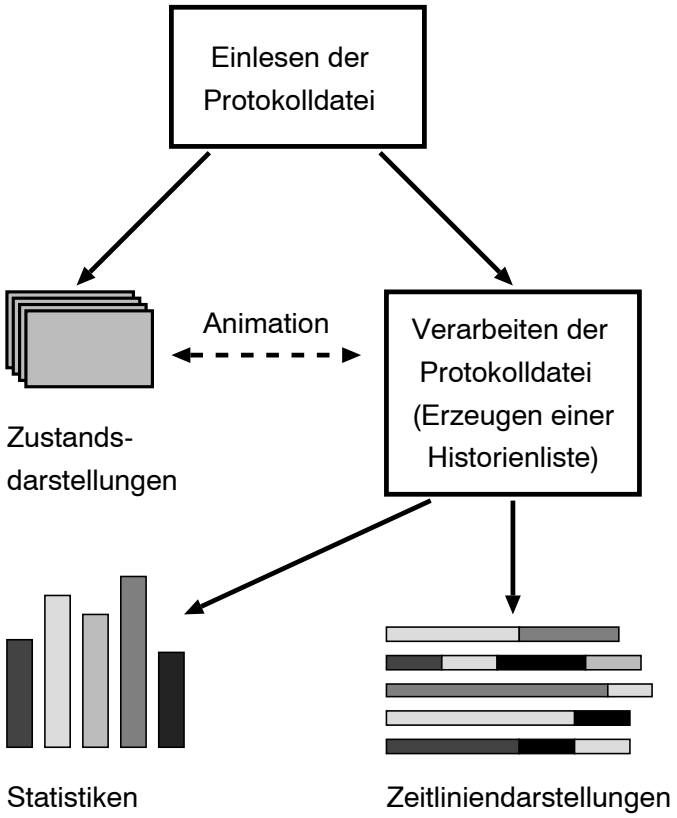


Abbildung 4.3: Die Bedienungsphilosophie der Visualisierungsumgebung PARvis

4.6.2 Darstellung von Prozessoraktivitäten

Abbildung 4.4 zeigt das Hauptfenster des PARtools-Systems, in dem die zentralen Bedienungselemente von PARtools und PARvis sowie bereits eine Darstellungsmethode enthalten sind. Die oberste Menüleiste enthält alle Bedienungselemente von PARvis, die zur Konfiguration des Systems, zur Behandlung der Eingabedaten und zur Anforderung der verschiedenen Visualisierungsmöglichkeiten erforderlich sind.

Die drei darunterliegenden Blöcke dienen zur Steuerung der unter der PARtools-Oberfläche abrufbaren Systeme PARsim und PARbench (s.o.) sowie das "Cockpit" von PARvis, mit dem die Navigation (Bewegung) auf der virtuellen Zeitachse der Visualisierungsumgebung durchgeführt wird. In der untersten Zeile des abgebildeten Fensters ist ein Rollbalken zu erkennen, der die momentane zeitliche Position innerhalb des gesamten simulierten Zeitraumes veranschaulicht.

Das Hauptfenster selbst kann bereits eine bestimmte Darstellung enthalten. In diesem Fall (Abb. 4.4) wurde die Zeitpunktdarstellung der aktuellen Prozessorzustände gewählt, in der der derzeitig vom Prozessor bearbeitete Job, der momentan behandelte Prozeß und die Auslastung bis zu diesem Zeitpunkt zu erkennen sind. Dabei wird jeder Prozessor durch einen rechteckigen Kasten symbolisiert, der Teilkästchen für die erwähnten Informationen bereithält. Alle angezeigten Werte werden zusätzlich durch zugeordnete Hintergrundfarben unterstützend repräsentiert (hier durch Graustufen dargestellt). So gilt für die Hintergrundfarbe des Auslastungs-Feldes eine Legende, die sich am rechten Rand des Fensters befindet.

Abbildung 4.5 zeigt das Hauptfenster, nachdem eine Zeitliniendarstellung, eine besondere Form der Zeitachsendarstellung, für den gesamten Simulationszeitraum ausgewählt wurde. Diese Darstellung reserviert für jeden Prozessor einen waagerechten Balken, der sich über die horizontal ausgerichtete Zeitachse erstreckt. Die Farben der Balken-Abschnitte symbolisieren die in dem entsprechenden Zeitraum bearbeiteten Jobs (siehe Legende am rechten Rand); die eingeschriebenen Zahlen hingegen geben die darin gerade sich in Bearbeitung befindenden Prozesse wieder. Solche Zeitliniendarstellungen können auch für beliebige einzelne Prozessoren in Zusatzfenstern erzeugt werden. In gleichem Maße ist die Darstellung von Zeitlinien für bestimmte Prozesse möglich. Je nach den Intentionen des Benutzers kann also die zeitliche Entwicklung der Abläufe mit einem Blick sowohl aus Prozessor- als auch aus Prozeßsicht verfolgt werden. Bei allen Zeitliniendarstellungen informiert der untere, waagerechte Rollbalken über den gerade betrachteten Ausschnitt des gesamten untersuchten Zeitraumes, der, wie schon erwähnt wurde, auf vielerlei Arten durch Verschieben und Zooming den individuellen Wünschen des Anwenders angepaßt werden kann. Um die Leistungsfähigkeit der einzelnen Fenster deutlich zu machen, sei beispielhaft eine Auswahl der Möglichkeiten aufgelistet, die dem Benutzer im oben erwähnten Prozessor-Zeitlinienfenster durch ein sogenanntes Popup-Menü zur Verfügung stehen:

- ein Meßinstrument für die Länge eines mit dem Mauszeiger markierten Ausschnitts
- Zooming durch Markieren eines Ausschnitts mit dem Mauszeiger
- Eine "undo"-Operation für den Zooming-Vorgang
- Anpassen des Darstellungsausschnittes an die vorliegenden Protokolldaten
- Verschieben des Auschnittes nach links oder rechts

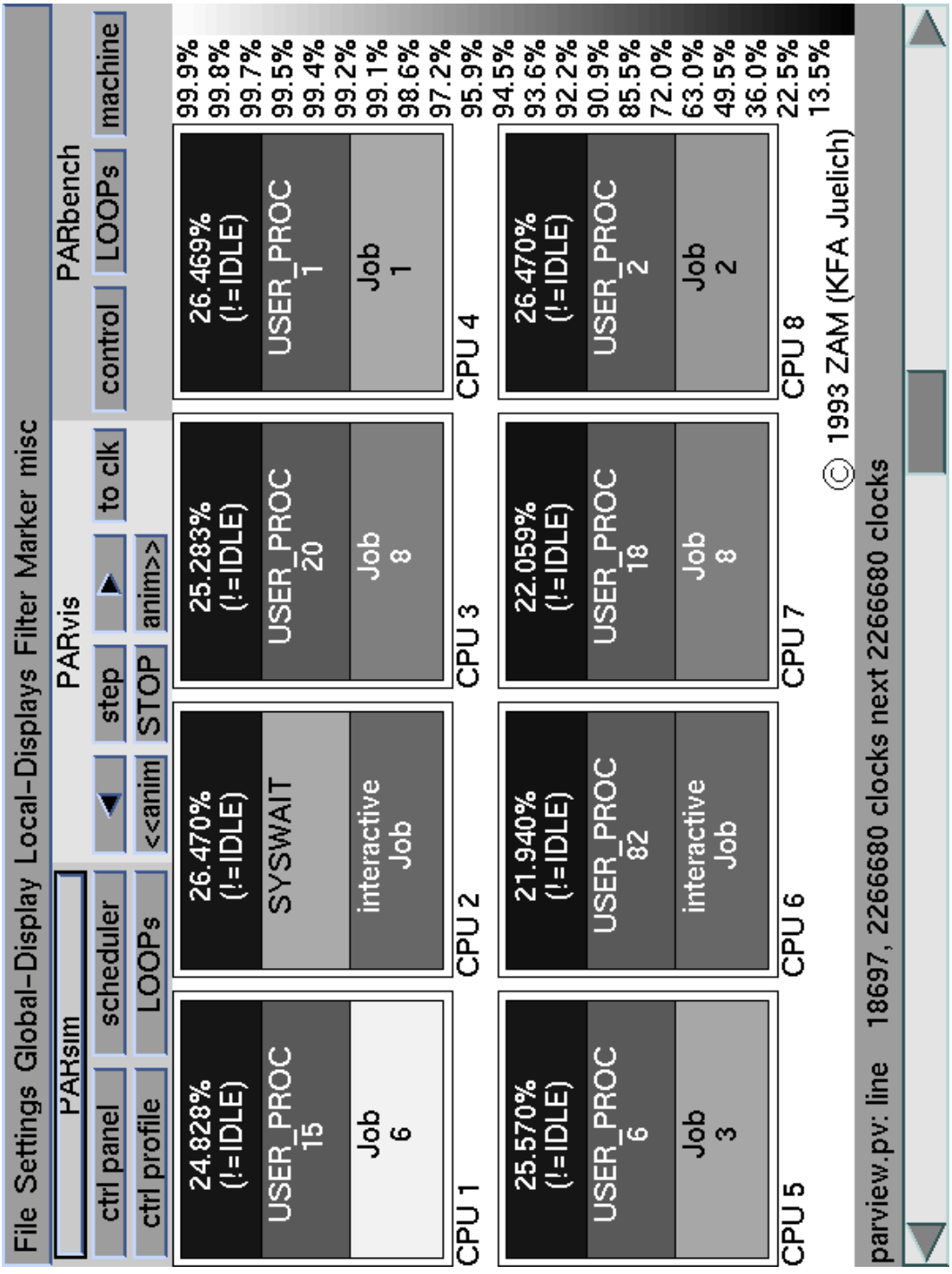


Abbildung 4.4: Hauptfenster mit PARtools-Oberfläche und Prozessor-Zeitpunktdarstellung

- Auflistung der zu einem markierten Ausschnitt gehörenden Protokolldaten in einem zusätzlichen Fenster

- Anzeige eines Graphen, der für jeden dargestellten Zeitpunkt den Grad des Parallelismus in dem betrachteten Mehrprozessorsystem veranschaulicht (Zeitachsendarstellung)

Weitere Punkte betreffen weitergehende Möglichkeiten des PARvis-Systems, die in diesem Rahmen nicht genauer behandelt werden können. Für eine detaillierte Beschreibung des Systems siehe [1]. Eine weitere Darstellungsmethode gibt in einer Zeitachsendarstellung genauere Auskunft über den Grad des Parallelismus. Hier werden auch die Belegungen gewisser Parallelitäts-Zweige (Prozessoren) mit Systemaufgaben gesondert ausgewiesen.

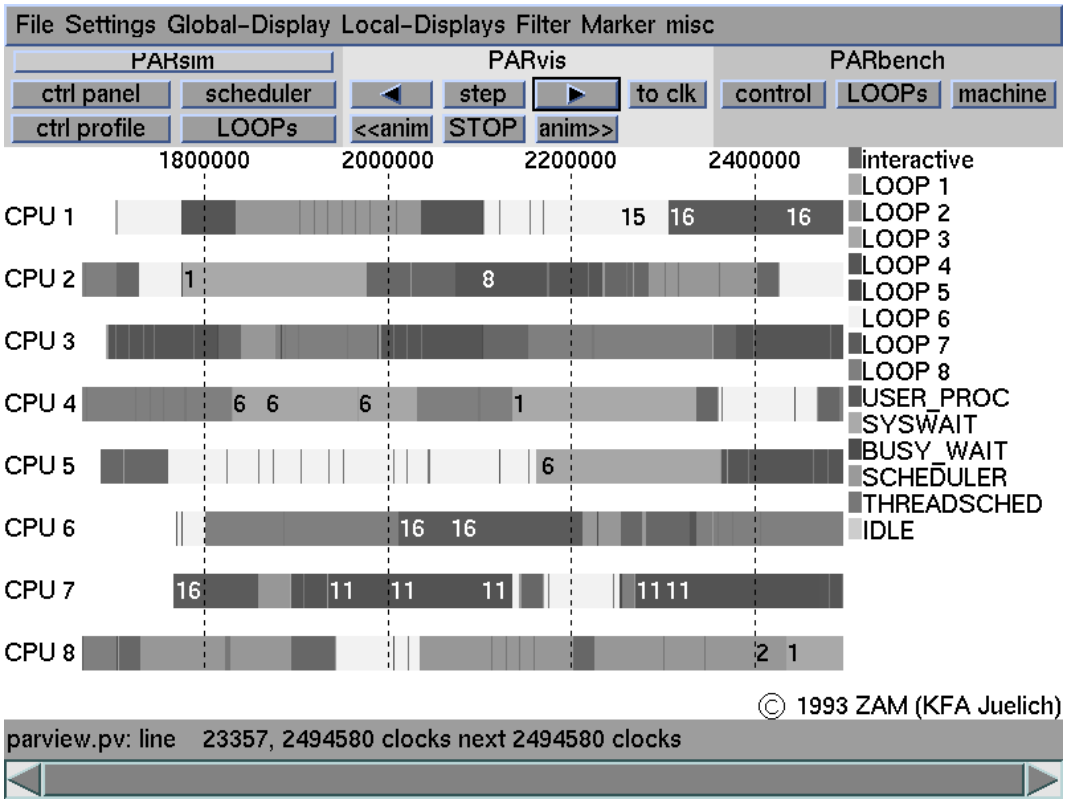


Abbildung 4.5: Hauptfenster mit Prozessor-Zeitliniendarstellung

Abbildung 4.6 zeigt eine prozentuale Aufstellung, die für einen bestimmten Prozessor die Zeitanteile der Bearbeitung verschiedener Tasks in Form eines Tortendiagramms zeigt. Es wurde hier die Möglichkeit, den jeweils maximalen Anteil auszublenden, zweimal angewendet, so daß nur noch die Anteile der Beschäftigung mit Systemaufgaben angezeigt werden. Derartige Auswertungen können auch in Form von Tabellen

oder Histogrammen dargestellt werden, ebenso für alle Prozessoren zusammen im Hauptfenster.

Drei weitere bemerkenswerte Aspekte des PARvis-Systems bedürfen noch besonderer Erwähnung: das Filterkonzept, das Auffinden von Problemstellen und das Setzen von Markierungen. In einem speziellen Dialogfenster kann die *Filterung* aller darzustellenden Informationen bezüglich bestimmter auszuwählender Prozessoren vorgenommen werden. So können z.B. nicht interessierende CPUs ausgeblendet werden. Auf die gleiche Weise kann auch eine Auswahl bezüglich bestimmter Jobs vorgenommen werden.

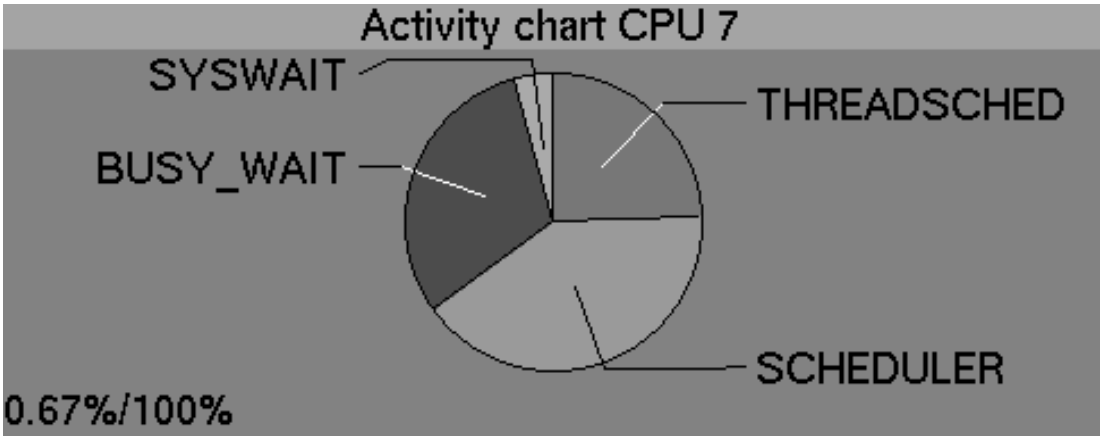


Abbildung 4.6: Hilfsfenster mit Auslastungsstatistik für Prozessor 7 (Zwei Maxima ausgeblendet)

In einem anderen Dialogfenster können bestimmte Bedingungen definiert werden, unter denen PARvis sogenannte *Problemstellen* aufspüren soll. Diese werden dann mit *Markierungen* versehen, welche ein weiteres, grundlegendes Konzept von PARvis darstellen. Markierungen können in Form einer Bewegung auf der Zeitachse direkt angesprungen, aber auch in Zeitliniendarstellungen graphisch verzeichnet werden. Es ist auch möglich, Markierungen "von Hand" zu setzen, auf diese Weise kann der Anwender selbst interessante Stellen kennzeichnen und später dorthin zurückkehren.

Die PARvis-Visualisierungsumgebung erlaubt eine vielfältige Einflußnahme auf das Aussehen der Bedienungselemente und auf die Darstellungsmethoden. So können z.B. sämtliche Farbzuordnungen verändert werden. Auch eine freie Wahl von Beschriftungsgrößen und Zeichensätzen ist ohne weiteres möglich. Für eine eingehende Beschreibung dieser Aspekte und der vielen hier unerwähnten weiteren Konfigurations- und Manipulationsmöglichkeiten muß wiederum auf [1] verwiesen werden.

4.6.3 Anknüpfungspunkte zur Visualisierung der Speicheraktivitäten

In dem bisher beschriebenen Umfang ist noch keine Berücksichtigung von Speicheraktivitäten enthalten. Sofern man das Konzept des gemeinsamen Speichers (Shared Memory) zugrunde legt, beschränken sich die Probleme, die mit dem Speicher in Verbindung stehen, auf gleichzeitige Zugriffe verschiedener Prozessoren auf gleiche Speicherstellen. Wählt man das Konzept des Nachrichtenaustauschs (Message-Passing), wird man nur mit solchen Phänomenen konfrontiert, die mit dem Senden, Empfangen und Weiterleiten von Nachrichten in Verbindung stehen. Die lokalen Operationen der Prozessoren auf ihren lokalen Speichern sind für die Untersuchung der speziellen Probleme paralleler Systeme nicht relevant.

Wenn der gemeinsame Speicher auf der logischen, virtuellen Seite und der verteilte Speicher auf der realen, physikalischen Seite miteinander kombiniert werden, ergibt sich das Phänomen der wesentlich erhöhten Komplexität und der Globalisierung der Bedeutung jeder Speicheroperation. Auf dieser Stufe ist es unumgänglich, geeignete Werkzeuge zu benutzen, um solche Systeme zu verstehen. Damit reicht die Betrachtung der reinen Prozessor- und Prozeßaktivitäten nicht mehr aus, selbst wenn nur scheinbar isolierte Probleme aus diesem Bereich untersucht werden sollen. Die Effekte der Speicheraktivitäten sind von sehr signifikantem Einfluß auf den gesamten Ablauf in dem parallelen System, so daß auch die spezielle Untersuchung der Prozessoraktivitäten ohne ihre Berücksichtigung nicht sinnvoll erfolgen kann. Wie in jedem komplexen, dynamischen, vernetzten System kann keiner der Teilaspekte isoliert von den anderen betrachtet werden.

Gegenstand dieser Arbeit ist die Erweiterung des PARvis-Systems um Werkzeuge, die die Vorgänge um den Speicher eines Mehrprozessorsystems in angemessener und geeigneter Weise veranschaulichen. Dabei wurde eine Virtual-Shared-Memory-Architektur zu Grunde gelegt.

Offensichtlich müssen, damit diese Werkzeuge zum Einsatz kommen können, die einzulesenden Protokolldaten Informationen über entsprechende Ereignisse enthalten. Dazu zählen z.B. Einträge über die Anforderung einer Speicherseite, Warten auf eine Seite, Senden oder Empfangen einer Seite usw.

Die gesamte, oben beschriebene Philosophie der Visualisierungsumgebung wurde für diese Hilfsmittel übernommen, da neue Datenstrukturen, die Informationen über die jeweiligen Speicherzustände enthalten, in die vorhandenen eingefügt bzw. parallel mitgeführt werden. Auf diese Weise entsteht kein logischer Bruch zwischen den einzelnen Visualisierungsmethoden, und Werkzeuge beider System-Generationen können parallel und verzahnt miteinander verwendet werden.

Die Hilfsmittel zur Darstellung der Speicheraktivitäten können ebenso wie die zur Visualisierung der Vorgänge in bezug auf Prozessoren und Prozesse per Knopfdruck aus sogenannten Pulldown-Menüs angefordert werden.

Kapitel 5

Visualisierung von Speicheraktivitäten

Die in dieser Arbeit entwickelten Konzepte zur Darstellung von Speicheraktivitäten und die daraus resultierenden neuen Werkzeugkomponenten tragen ganz wesentlich zum Verständnis dynamischer Vorgänge in Systemen mit virtuell gemeinsamem Speicher bei. In diesem Kapitel werden zunächst die Konzepte der neuen Komponenten vorgestellt, das darauf folgende Kapitel stellt die Anwendung der Werkzeugkomponenten in den Mittelpunkt.

5.1 Grundlegende Werkzeugkonzepte

Bei der Darstellung dynamischer Vorgänge in Systemen mit virtuell gemeinsamem Speicher sind Rahmenbedingungen zu beachten, die sich einerseits aus der vorhandenen Werkzeugumgebung ergeben und andererseits durch die grundsätzlichen Visualisierungsanforderungen bestimmt sind. Daher soll kurz auf die prinzipiellen Ziele der Darstellungshilfsmittel und ihre Einbettung in die Philosophie des PARvis-Systems eingegangen werden.

5.1.1 Visualisierungsziele

Die Darstellung von Speicheraktivitäten in einem Mehrprozessorsystem mit virtuell gemeinsamem Speicher kann auf verschiedenen Abstraktionsstufen erfolgen. Im Mittelpunkt der Anforderungen an die hier vorgestellten Hilfsmittel steht die Speicherseite selbst, wobei umfangreiche mikroskopische Einblicke in das Verhalten einzelner Seiten bzw. in die Aktionen einzelner Prozessoren auf Speicherseiten möglich gemacht werden sollen. Dies soll sowohl in Form von statischen Darstellungen mit der Möglichkeit zur Animation als auch durch Zeitliniendarstellungen erfolgen. Globale Übersichten, in denen größere Einheiten von Speicherseiten und Prozessoren

zur Ansicht gebracht werden, sind dabei ebenso vorzusehen wie Filtermöglichkeiten, mit denen Darstellungen auf bestimmte Teilmengen dieser Komponenten reduziert werden können. In den folgenden Abschnitten sollen die einzelnen Werkzeuge in bezug auf ihre grundlegenden Ideen und Konzepte vorgestellt und ihre möglichen Einsatzgebiete kurz umrissen werden.

5.1.2 Einheitlichkeit

Die neu zu entwickelnden Werkzeuge zur Darstellung der Speicheraktivitäten sollen eine konsequente Fortsetzung des bisherigen PARvis-Systems bilden. Das bedeutet, daß die Philosophie und die Prinzipien der Bedienung der einzelnen Werkzeuge soweit wie möglich beibehalten werden sollten, damit sich dem Benutzer in jeder Situation eine einheitliche Oberfläche eines als "in einem Guß" konzipierten Hilfsmittels präsentiert. Einige der global geltenden Bedienungsprinzipien sind:

- *Einheitliche Oberfläche*: Alle Darstellungsmittel sind aus einer einzigen Menüleiste heraus anzufordern.
- Es wurde das Prinzip beibehalten, in jedem Fenster umfangreiche Möglichkeiten zur *Präzisierung* der Darstellungswünsche und zur *Fokussierung* auf spezielle Teilinformationen vorzusehen.
- *Einheitliches Erscheinungsbild*: Alle Werkzeuge erscheinen als eigene Fenster, die dadurch beliebig miteinander kombinierbar sind. Ausgewählte Darstellungsparameter wie Hintergrundfarben und Zeichensätze gelten für alle Werkzeuge gleichermaßen.
- *Einheitliche Bedienung*: Der Umgang mit den unterschiedlichen Darstellungsmitteln ist, soweit es ihre Konzepte zulassen, ähnlich gestaltet worden. Ein Beispiel: In allen Zeitlinienfenstern können Zooming-Funktionen auf die gleiche Weise ausgeführt werden, um detaillierte Ausschnittvergrößerungen herzustellen.
- *Zeitsynchronisierung*: Bei der Anzeige zeitlich aufeinander folgender Zeitpunktdarstellungen (Animationen) werden alle Fenster zeitsynchron erneuert. Auch die Wahl von Zeitausschnitten für Zeitliniendarstellungen und Statistiken wirkt sich auf alle derartigen Hilfsmittel aus.

Dennoch war es nicht zu vermeiden, daß sich die neu entwickelten Werkzeuge im Erscheinungsbild und in der Handhabung in einigen Punkten von den bisher implementierten unterscheiden. Dies liegt an den anderen Visualisierungsaufgaben dieser neuen Methoden, die zum Teil ganz andere Darstellungsprinzipien verfolgen. Insbesondere das Werkzeug zur Verdeutlichung der Kommunikationsvorgänge im Prozessornetzwerk fällt in Konzeption und Realisierung aus dem Rahmen der übrigen Komponenten des PARvis-Systems. Dies ist jedoch aufgrund der vollständig anderen Anforderungen für die Visualisierungsaufgabe unumgänglich.

5.2 Darstellungsmethoden

Die einzelnen Hilfsmittel zur Visualisierung der dynamischen Vorgänge in Systemen mit virtuell gemeinsamem Speicher unterscheiden sich hinsichtlich des Standpunktes und der Perspektive, die sie einem Beobachter bieten. Brauchbare Einblicke entstehen auch hier in den meisten Fällen erst durch die Kombination mehrerer dieser Werkzeuge. In den folgenden Abschnitten werden die einzelnen Werkzeugkomponenten und ihre Grundideen vorgestellt. Abbildung 5.1 gibt einen Überblick über die Struktur der in dieser Arbeit vorgestellten Visualisierungshilfsmittel.

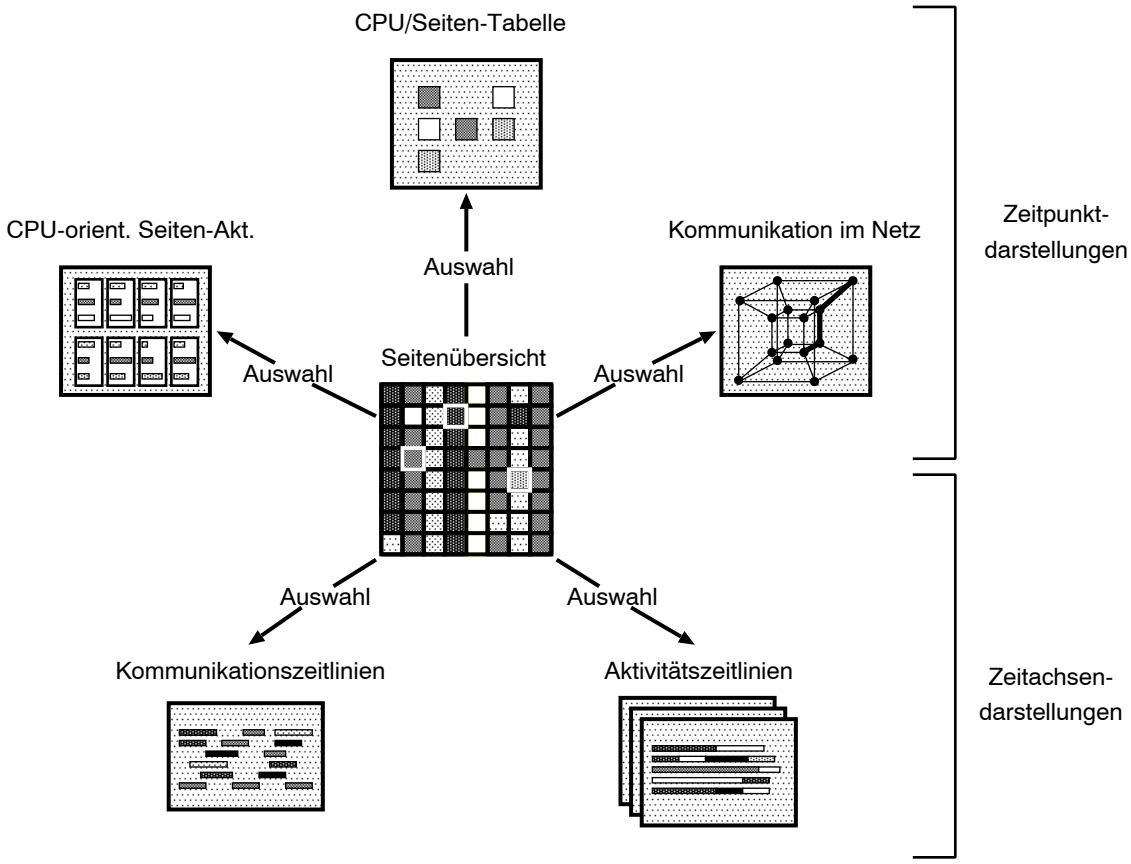


Abbildung 5.1: Die Struktur der VSM-Visualisierungswerkzeuge

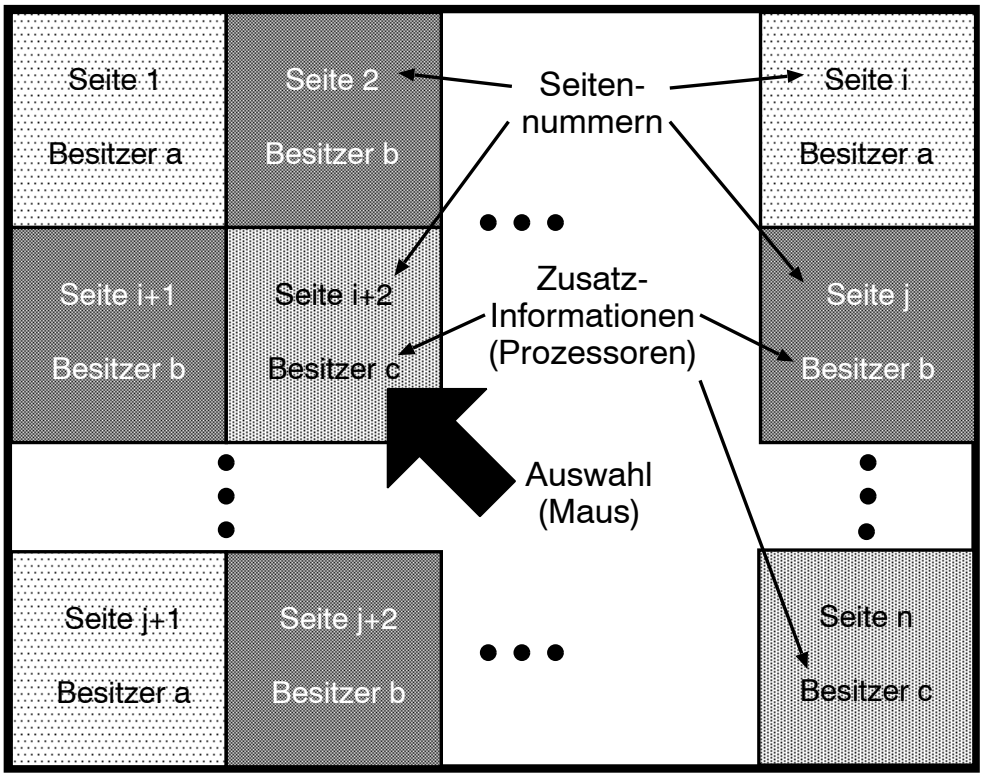


Abbildung 5.2: Die Speicherseitenübersicht mit Auswahlmöglichkeiten

5.2.1 Die Speicherseitenübersicht

Zunächst soll ein Hilfsmittel vorgestellt werden, das eine Gesamtübersicht über alle im System vorhandenen, virtuellen Speicherseiten bietet. Von dieser Übersicht aus ist die Auswahl einer oder mehrerer Seiten für weitere, spezielle Visualisierungshilfsmittel leicht möglich. Die Abbildung 5.2 zeigt das Grundprinzip dieser Darstellungsmethode.

5.2.1.1 Die Darstellung der Informationen

Das vorliegende Werkzeug bietet eine Seitendarstellung, in der jede Speicherseite durch ein Rechteck repräsentiert wird. Jedes Rechteck enthält zwei Inschriften: zum einen die Seitennummer, zum anderen den Inhalt eines der den Zustand einer Seite beschreibenden Felder, wie sie im Abschnitt über die Struktur der Eingabedaten beschrieben sind (Abschnitt 5.3.1). Bezüglich der letzteren Information liegt hier also eine *Zeitpunktdarstellung* vor. Es kommen dafür jedoch nur solche Felder in Frage, die nur jeweils einen einzigen Eintrag enthalten, da die Darstellung von Prozessorlisten im Rahmen dieses Darstellungsmittels zu komplex wäre. So kann z.B. die Nummer des Prozessors mit Schreibberechtigung auf die Speicherseite angezeigt werden. Die dargestellte Zusatzinformation wird zudem durch die

Einfärbung des entsprechenden Rechteckes wiedergegeben, jede Prozessornummer oder -anzahl entspricht dabei einer bestimmte Farbe, die Farbzuordnung kann einer Legende entnommen werden. Mit Hilfe der farblichen Repräsentation der Zusatzinformationen können eventuell in ihr enthaltene Muster und Regelmäßigkeiten schneller und gleichsam mit einem Blick erfaßt werden. Welches der skalaren Felder, die bis auf eines eine CPU-Nummer enthalten (Ausnahme: Feld mit der *Anzahl* der Prozessoren mit Leseberechtigung), angezeigt wird, kann der Benutzer mit Hilfe der Maus über ein sogenanntes Popup-Menü bestimmen.

5.2.1.2 Visualisierung des Kommunikationsanteils

Die implementierte Speicherseitenübersicht bietet darüber hinaus jedoch auch noch eine weitere Anzeigemöglichkeit. Diese setzt jedoch voraus, daß durch Vorwärtsbewegung durch die Protokolldatei eine Historienliste erzeugt wurde, wie sie auch für Zeitliniendarstellungen erforderlich ist. Tortendiagramme, die in die Rechtecke eingebracht werden, welche die Seiten repräsentieren, verdeutlichen den Kommunikationsanteil der Seite an der Gesamtzeit. Das ist der Zeitanteil, in dem die Seite Gegenstand von Kommunikationsvorgängen ist. Die Gesamtzeit ist dabei der durch die Historienliste definierte Zeitraum. Der so ablesbare, eventuell vermeidbare Kommunikationsaufwand entsteht in der Regel durch mehrere CPUs, die die Seite ausschließlich nutzen (beschreiben) wollen und dadurch miteinander in Konkurrenz treten. Die Ursache für derartige Konkurrenzsituationen ist im allgemeinen eine mangelnde Datenlokalität der in der Seite abgelegten Informationen. Daher soll diese Darstellung ein Auswahlkriterium an die Hand geben, das es ermöglicht, solche eventuell problematischen und damit für die Visualisierung interessanten Seiten zu erkennen und für weitere Darstellungsmethoden auszuwählen.

5.2.1.3 Die Auswahl von Seiten

Die Auswahl von Seiten kann in jedem Fall durch einfache Mausoperationen innerhalb des Anzeigebereiches erfolgen. So ist es möglich, einzelne Seiten durch Anklicken mit der Maus zu wählen oder aber wieder abzuwählen. Wahl oder Abwahl sind auch für ganze Seitenbereiche möglich, deren Repräsentanten dafür mit einem durch den Mauszeiger erzeugten Selektionsrechteck überdeckt werden müssen. Ausgewählte Seiten werden durch eine weiße Umrandung der sie repräsentierenden Rechtecke kenntlich gemacht.

5.2.2 Die Tabelle der Prozessor-Seiten-Beziehungen

Mit diesem Visualisierungswerkzeug ist es möglich, das Verhältnis einer über die oben beschriebene Übersicht definierten Auswahl von Seiten zu den Prozessoren für einen *Zeitpunkt* darzustellen. Die dabei zu berücksichtigenden CPUs können mit

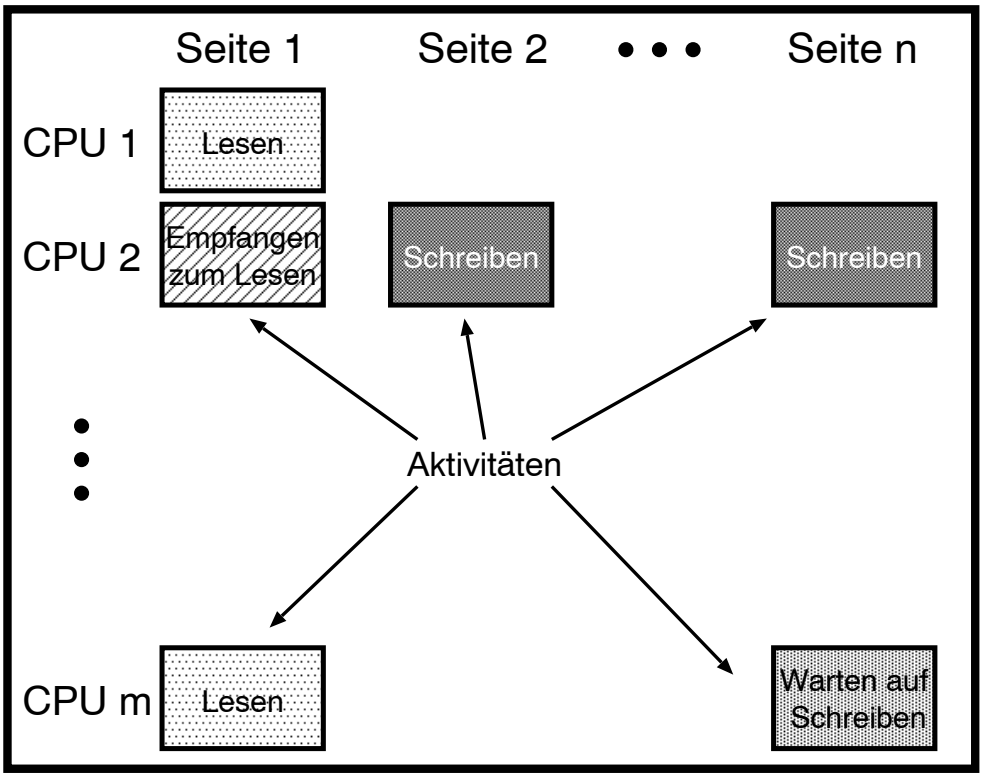


Abbildung 5.3: Die Tabelle der Prozessor-Seiten-Beziehungen

Hilfe eines separaten Dialogfensters ausgewählt werden, das global für alle Speicherdarstellungen Einstellmöglichkeiten bereithält und in Kapitel 6 näher beschrieben wird. Die Visualisierung erfolgt in einer matrixartigen Anordnung, in der für jede ausgewählte Seite eine Spalte und für jede CPU eine Zeile vorhanden ist. Die Abbildung 5.3 zeigt das Schema dieser Darstellung. Am Kreuzungspunkt zwischen der einer Seite zugeordneten Spalte und der zu einer CPU gehörenden Zeile befindet sich ein farbiges Rechteck, falls es eine logische Beziehung zwischen beiden Komponenten zum für die Anzeige gültigen Zeitpunkt gibt. Die Farbe des Rechteckes gibt die Art der Beziehung wieder, die entsprechende Zuordnung kann anhand einer Legende abgelesen werden. Hier wie auch in der im folgenden beschriebenen Zeitliniendarstellung werden nur ausgewählte, wesentliche CPU-Seitenbeziehungen berücksichtigt:

- das Bestehen einer *Leseberechtigung*,
- das Bestehen einer *Schreibberechtigung*,
- das Bestehen einer Berechtigung zum *schwach-kohärenten Schreiben*,
- der *Wartezustand* bezüglich der Aktivitäten Lesen, Schreiben, schwach-kohärentes Schreiben, Empfang einer Kopie vom Kommunikationsprozessor des aktuellen Besitzers sowie Beendigung des Besitzerwechselvorganges,

- das Bestehen des *Changing-Zustandes*: in diesem Fall ist die Lese- oder Schreibberechtigung erteilt, und es wird auf die Beendigung des Empfangsvorganges für eine aktuelle Kopie der Seite gewartet, oder aber eine CPU mit Schreibberechtigung ist mit der Invalidation der anderen Kopien der Seite beschäftigt.

Die zuletzt genannte Beziehung Changing-Zustand wird durch eine Schraffur verdeutlicht, die je nach Bedeutung mit der Farbe für den Zustand der Lese- oder Schreibberechtigung kombiniert wird.

Dieses Werkzeug ist zur genauen, mikroskopischen Untersuchung eines Zustandes zu einem Zeitpunkt konzipiert worden. Um diesen Zeitpunkt zu ermitteln, ist eine schrittweise Fokussierung mit Hilfe anderer Werkzeuge wie z.B. der nachstehend beschriebenen Zeitliniendarstellung der Seitenaktivitäten nötig.

5.2.3 Darstellung der CPU-orientierten Seitenaktivitäten

Eine weitere *Zeitpunktdarstellung* zeigt, auf welchen Seiten die Prozessoren des Systems Aktivitäten durchführen. Dieses *Prozessor*-orientierte Hilfsmittel berücksichtigt im Gegensatz zu den zuvor beschriebenen Werkzeugkomponenten *alle* möglichen Aktivitäten. Die Abbildung 5.4 zeigt das Schema des entsprechenden Visualisierungsfensters. Es wird, ähnlich dem Hauptfenster des PARvis-Systems, für jede CPU des betrachteten Systemlaufs ein rechteckförmiger Fensterausschnitt bereitgehalten. Darin wird für jede Aktivität eine Zeile bereitgestellt. Jede dieser Zeilen kann zwei unterschiedliche Zustände einnehmen, zwischen denen der Anwender umschalten kann (Abbildung 5.4).

5.2.3.1 Die Anzeige der Seitenanzahl

In diesem Modus wird die Anzahl der Seiten, auf denen der Prozessor die entsprechende Aktivität ausübt, als Zahl angezeigt. Gleichzeitig veranschaulicht ein waagerechter Balken diesen Wert als relative Anzahl bezüglich der Gesamtzahl der in dem Systemlauf vorhandenen Speicherseiten. Das bedeutet, daß ein Balken, der sich bis zur rechten Kante des die CPU repräsentierenden Rechtecks erstreckt, die entsprechende Aktivität auf *allen* Seiten symbolisiert. Die Balken besitzen eine Farbe, die der Aktivität zugeordnet ist und mit Hilfe des in Kapitel 6 beschriebenen Einstellfensters für Speicherdarstellungen ausgewählt werden kann. Diese Art der Darstellung ist nützlich, wenn die einzelnen Speicherseiten weniger interessant sind und es vielmehr um die Bedeutung eines Prozessors im System bezüglich einer Aktivität geht.

5.2.3.2 Die Darstellung der einzelnen Seiten

Wird dieser Modus gewählt, erscheint in der Aktivitätszeile ein kleines Fenster, in dem alle der Aktivität unterliegenden Speicherseiten nebeneinander dargestellt wer-

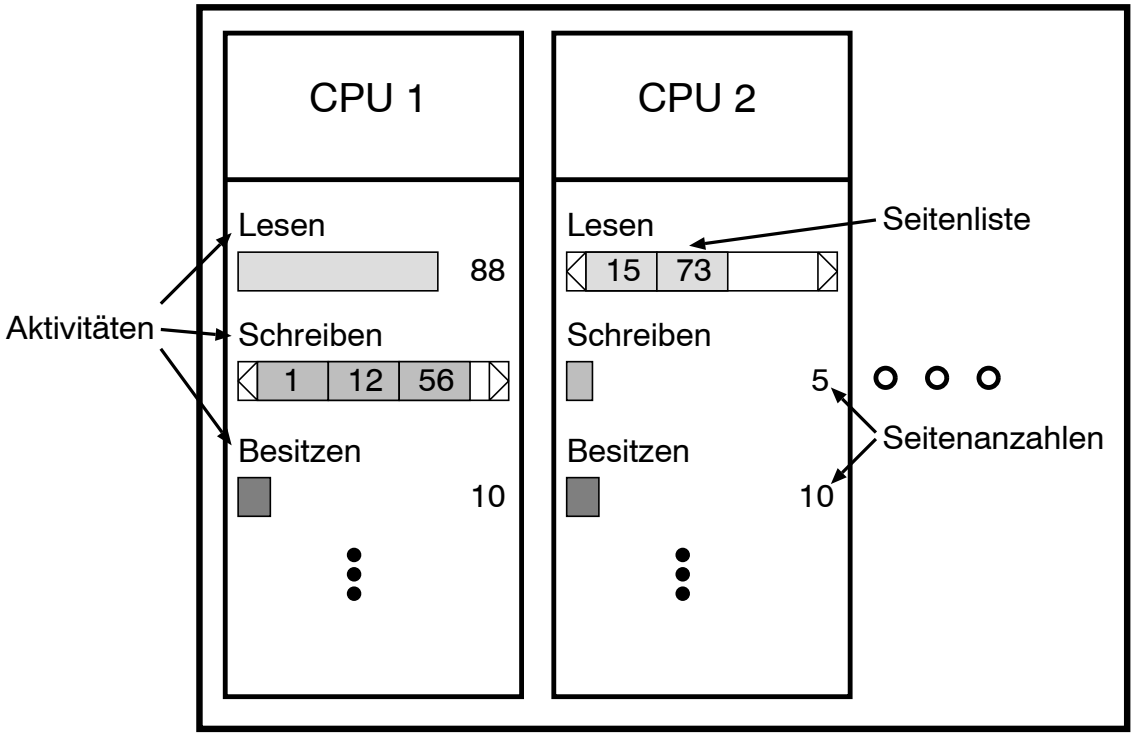


Abbildung 5.4: Die Darstellung der CPU-orientierten Seitenaktivitäten

den. Dies geschieht in Form von kleinen Rechtecken, die mit der Seitennummer beschriftet sind. Auch die Rechtecke besitzen die der Aktivität zugeordnete und mit Hilfe des globalen Konfigurationsdialogs einstellbare Farbe. Werden die Speicherseiten angezeigt, die sich im Changing-Zustand befinden, so entspricht die Farbe der beabsichtigten Aktivität (z.B. Lesen, Schreiben usw.). In der Regel passen bei weitem nicht alle darzustellenden Seiten in das der Aktivität zugeordnete Fenster. In diesem Fall kann sich der Anwender durch Anklicken von Pfeilen, die rechts und links des kleinen Fensters angeordnet sind, durch die gesamte Liste der Seiten hindurchbewegen. Es entsteht dabei der Eindruck eines wandernden Betrachtungsausschnitts aus einer Kette aneinandergereihter Speicherseiten. Dieser Darstellungsart ist für die detaillierte Verfolgung der Aktivitäten bestimmter Prozessoren auf Seiten geeignet.

5.2.4 Die Zeitliniendarstellung der Seitenaktivitäten

Zeitachsendarstellungen besitzen in PARvis eine Besonderheit: Obwohl das Visualisierungs-System sich stets in einem Zustand befindet, der einem definierten Zeitpunkt des Systemlaufs entspricht, müssen für solche Darstellungen Informationen über einen kompletten Zeitraum aufbereitet werden. Daher soll kurz auf die interne

Realisierung der Historienliste eingegangen werden, in der die Speicheraktivitäten für definierte Zeitabschnitte abgelegt werden und die damit die Eingabedaten für Zeitachsendarstellungen liefert. Danach werden die auf spezielle Speicherseiten bezogenen Zeitliniendarstellungen vorgestellt.

5.2.4.1 Die Historienliste der Speicherzustände

Wie schon bei der Beschreibung der PARvis-Philosophie erörtert, ist vor der Nutzung einer Zeitachsendarstellung die Erzeugung einer Historienliste erforderlich, die alle Ereignisdaten des ausgewählten Zeitraumes im Speicher des PARvis-Systems bereithält. Die entsprechende Liste der Speicheraktivitäten wird separat verwaltet und parallel zur Historienliste der Prozeßwechselaktivitäten geführt. Da die Historienlisten in PARvis einem zeitlichen Rückblick fester Länge vom momentanen Zeitpunkt aus entsprechen, entstehen sie bei ihrer Erzeugung durch zeitliche Vorwärtsbewegung als eine Art gleitender Betrachtungsausschnitt. Das heißt, daß bei dieser Bewegung in der Zeit neue, aktuelle Daten an den Kopf der Liste gestellt und die ältesten Informationen außerhalb des nun betrachteten Zeitfensters automatisch herausgelöscht werden. In diesem Kontext bot sich für die Historienliste der Speicheraktivitäten eine verkettete Liste als geeignete Datenstruktur an, da bei ihr Löschungen und Hinzufügungen am Anfang und am Ende dynamisch möglich sind. Die Abbildung 5.5 zeigt die Grundstruktur der Speicher-Historienliste. Jedes ihrer Elemente wird erstellt, sobald ein protokolliertes Ereignis einen Seitenzustand verändert, und repräsentiert dann den neuen Zustand dieser Seite, wobei der Zeitpunkt, ab dem dieser gültig ist, mit abgelegt wird.

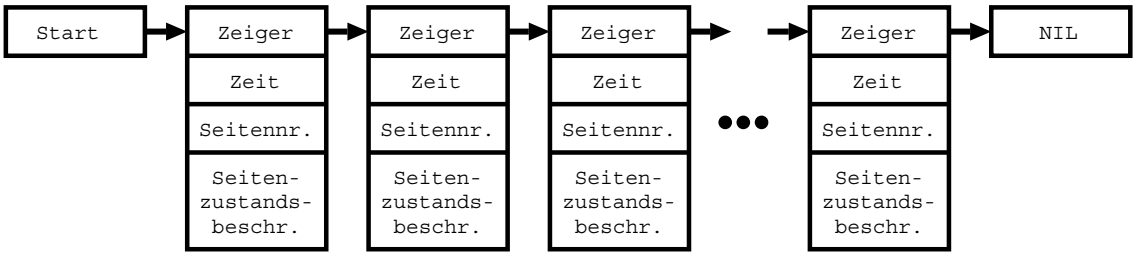


Abbildung 5.5: Die Struktur der Speicher-Historienliste

5.2.4.2 Aufruf und Aufbau der Aktivitäts-Zeitlinien

Zur Beobachtung der Aktivitäten der Prozessoren auf *einer* Speicherseite über einem ausgewählten Zeitraum steht die im weiteren beschriebene Zeitliniendarstellung der Seitenaktivitäten zur Verfügung. Die Auswahl der durch dieses Werkzeug zu untersuchenden Seiten geschieht mit Hilfe der oben beschriebenen Seitenübersicht. Für

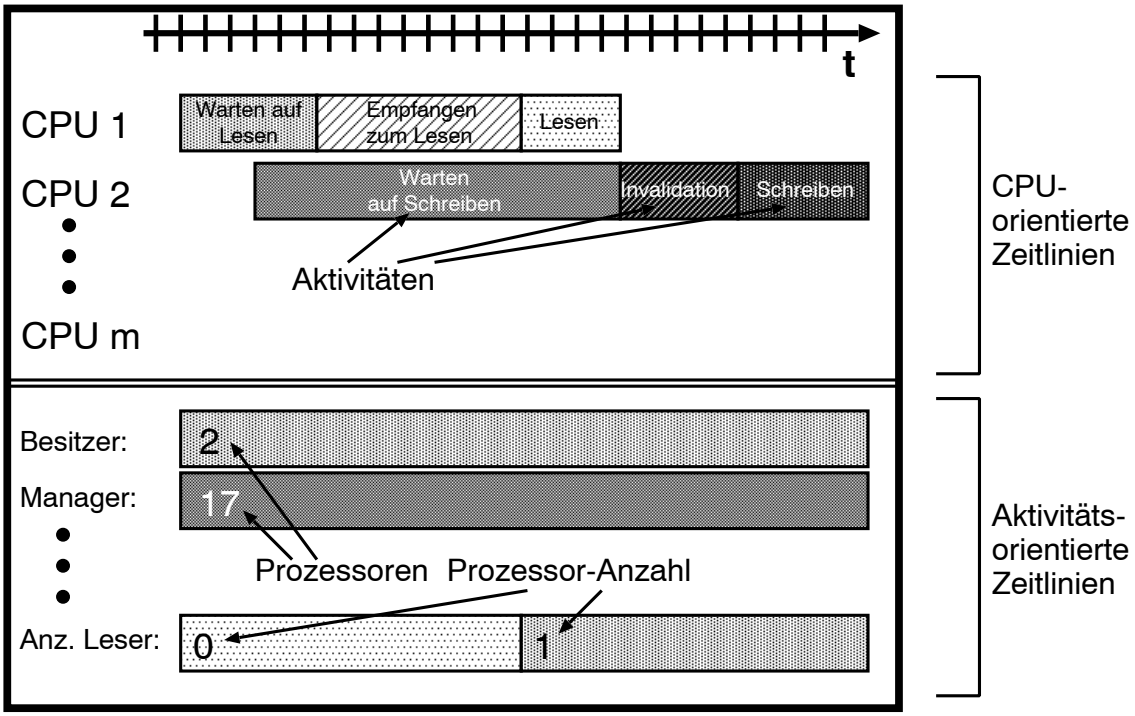


Abbildung 5.6: Die Zeitliniendarstellung der Seitenaktivitäten

jede ausgewählte Speicherseite wird bei Aufruf des Hilfsmittels ein eigenes Fenster erzeugt. Abbildung 5.6 zeigt die Grundstruktur eines solchen Fensters. Darin ist die Zeit auf einer waagerechten Achse aufgetragen. Die Einheit der Zeitachsenbeschriftung (Maschinenzyklen oder Sekunden) kann durch ein zentrales Menü des PARvis-Systems ausgewählt werden.

Unter der Zeitskala befinden sich zwei getrennte Bereiche des Fensters, die Informationen in Form von Balkendarstellungen bereithalten.

5.2.4.3 CPU-orientierte Aktivitäts-Zeitlinien

In der Abbildung 5.6 (oben) werden die *Prozessoraktivitäten* auf der Speicherseite wiedergegeben. Für jede dargestellte *CPU* wird eine eigene Zeitlinie erzeugt, auf der Balkenabschnitte die jeweiligen Aktivitäten der CPU in den überdeckten Zeiträumen repräsentieren. Diese Balkenabschnitte geben die verschiedenen Aktivitäten durch unterschiedliche Färbung bzw. Schraffur wieder. Es werden dabei dieselben Aktivitäten wiedergegeben, wie sie in der Beschreibung der Prozessor-Seiten-Tabelle erläutert wurden. Zudem wurde dabei dieselbe Farbkodierung verwendet, die auch hier an einer Legende abgelesen werden kann. Es werden nur diejenigen Prozessoren berücksichtigt, die in dem oben bereits erwähnten globalen Konfigurationsdialog für Speicherdarstellungen ausgewählt wurden.

5.2.4.4 Aktivitäts-orientierte Zeitlinien

In einem zweiten Bereich des Fensters (siehe Abbildung 5.6 unten) können weitere Zeitlinien dargestellt werden, die sich auf die Parameter

- Verwalter (Manager) einer Seite,
- Besitzer einer aktuellen Kopie der Seite nach der Information des Managers,
- tatsächlicher Besitzer einer aktuellen Kopie der Seite und
- Anzahl der Prozessoren mit Leseberechtigung

beziehen. Für jede der aufgezählten *Aktivitäten* wird eine eigene Zeitlinie angezeigt, die Farben der eingefügten Balken entsprechen den Prozessoren, die in den überdeckten Zeiträumen die entsprechende Aktivität auf der dem Fenster zugeordneten Seite ausführen. Es wird dieselbe Farbzuordnung wie in der Speicherseitenübersicht angewendet und, sofern Platz vorhanden ist, die repräsentierte CPU-Nummer zusätzlich in den Balken hineingeschrieben. Dieser Teil des Fensters bietet ergänzende Informationen, die einige Phänomene, die im ersten Teil zutage treten, leichter verständlich machen können.

5.2.4.5 Anwendung der Aktivitäts-Zeitlinien

Mehrere Funktionen zur Wahl von Darstellungsausschnitten wie Verschieben des Betrachtungsausschnitts oder Zooming erlauben weitreichende Möglichkeiten zur Variation der Beobachtungsperspektive. Dadurch ist dieses Werkzeug zum Aufspüren von interessanten Zeitpunkten oder -räumen geeignet, die dann auch mit anderen Hilfsmitteln weiter untersucht werden können. Weiterhin ist es mit diesem Werkzeug möglich, die Entwicklung der Aktivitäten über einen Zeitraum hinweg zu beobachten, ohne daß der Anwender, wie bei den Zeitpunktdarstellungen, Zustände zurückliegender Zeitpunkte im Gedächtnis behalten muß. Zeitliche Aktivitätswechsel treten sehr deutlich zutage. Bei Schritt-für-Schritt-Bewegungen oder kontinuierlichen Bewegungen durch ein Zeitintervall kann man auch den sukzessiven Aufbau der Zeitliniendarstellungen verfolgen (ähnlich einer Animation), was einen zusätzlichen Eindruck von der Dynamik der Aktivitäten verschaffen kann.

Von jedem Zeitlinienfenster aus ist die Aktivierung eines Modus möglich, in dem durch Anklicken einer beliebigen Position innerhalb eines solchen Fensters der dazugehörende Zeitpunkt mit einer senkrechten Linie markiert wird und sämtliche Zeitpunktdarstellungen den Systemzustand wiedergeben, der diesem Zeitpunkt entspricht. Dieser Modus besteht global für alle Speicher-Zeitliniendarstellungen und erlaubt schnelles, punktuelles Erforschen des Systemzustandes an interessant erscheinenden zeitlichen Positionen innerhalb der Zeitliniendarstellungen.

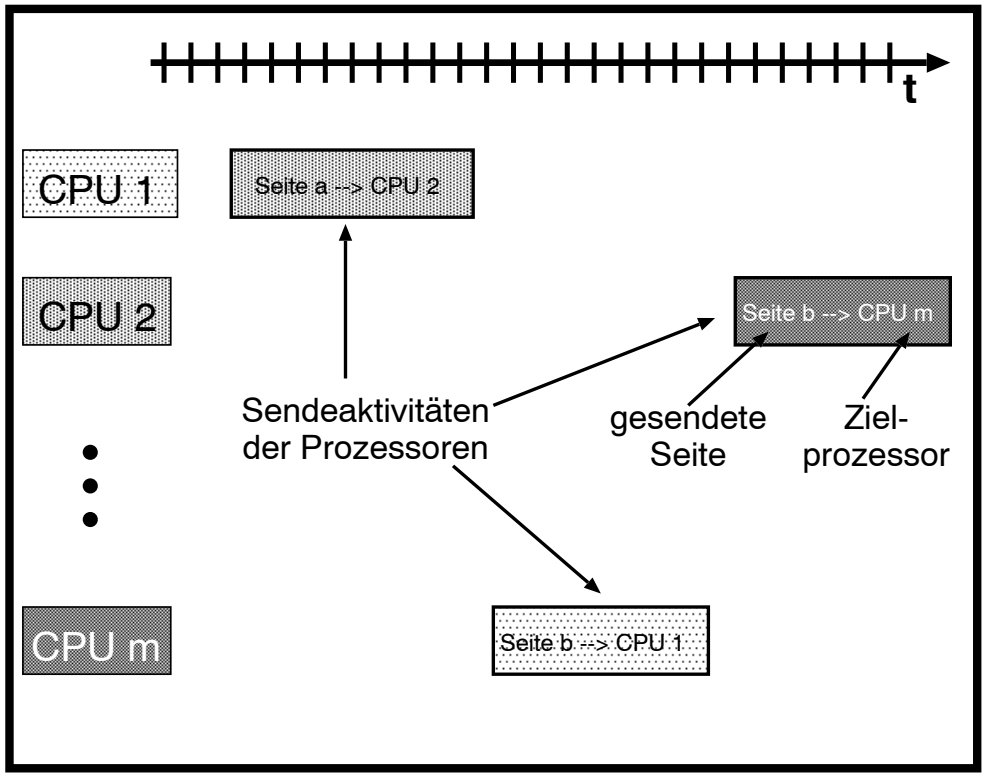


Abbildung 5.7: Die Zeitliniendarstellung der Prozessorkommunikation

5.2.5 Zeitliniendarstellung der Prozessorkommunikation

In einer weiteren *Zeitliniendarstellung* können die Kommunikationsvorgänge zwischen den Prozessoren für einen Zeitraum deutlich gemacht werden. Die Abbildung 5.7 zeigt das Schema einer solchen Darstellung. Der Benutzer hat die Möglichkeit, alle oder nur die z.B. mit Hilfe der Seitenübersicht ausgewählten Seiten berücksichtigen zu lassen. Auch hier gibt es für jeden dargestellten Prozessor eine eigene Zeitlinie, auf der die Zeitabschnitte markiert sind, in denen der Prozessor Seiten an andere Prozessoren versendet. Die Markierung dieser Sendezeiten geschieht durch farbige Balkenabschnitte. Die gesendete Seite sowie der Zielprozessor sind durch in diese Balkenabschnitte eingebrachte Inschriften deutlich gemacht. Zusätzlich wird der Zielprozessor durch die Farbe des Abschnittes repräsentiert. Dabei wird die Zuordnung der Farben zu den CPUs dem Anwender durch farbige Hinterlegung der Prozessornummern mitgeteilt, die als Zeitlinienbeschriftung dienen. Außerdem ist es möglich, Linien zur besseren Verdeutlichung der Übertragungsverbindungen einzeichnen zu lassen. Diese erstrecken sich vom Anfangspunkt eines Sendebalkens bis zu seinem *zeitlichen* Endpunkt in der Zeile des *Zielprozessors*.

Bei diesem Werkzeug stehen dieselben Möglichkeiten zur Wahl des Darstellungsausschnittes zur Verfügung wie bei allen anderen Zeitliniendarstellungen. Zusätzlich kann das Fenster an die Zeitliniendarstellung der Prozessoraktivitäten angekoppelt

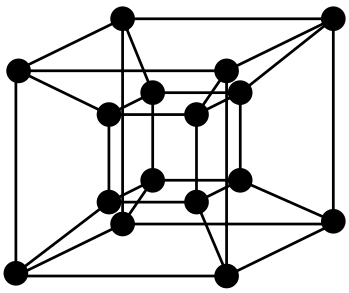
werden. Damit stimmen die Maßstäbe nach manueller Ausrichtung der linken Fensterkanten beider Darstellungen überein, und die Skalen sind deckungsgleich. Es werden auch hier nur diejenigen Prozessoren berücksichtigt, die in dem erwähnten globalen Konfigurationsdialog für Speicherdarstellungen ausgewählt wurden.

Dieses Fenster gibt einen recht guten Überblick über das Kommunikationsverhalten ausgewählter Prozessoren und Seiten. Durch die Ankopplungsmöglichkeiten eignet es sich sehr gut zur kooperativen Nutzung mit der Zeitliniendarstellung der Prozessoraktivitäten. Auch von diesem Fenster aus ist die Aktivierung des im vorigen Abschnitt beschriebenen Modus zur zeitlich-punktuellen Erforschung von Systemzuständen möglich.

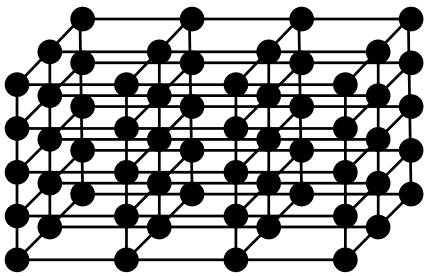
5.2.6 Visualisierung der Prozessorkommunikation im Netzwerk

Hierbei handelt es sich wiederum um eine *Zeitpunktdarstellung*, die in entsprechenden Funktionen einiger bereits vorhandener Visualisierungswerkzeuge wie z.B. ParaGraph [6] ihr Vorbild hat. Hier kam es jedoch auf zwei wesentliche Eigenschaften an, die in der in ParaGraph vorhandenen Kommunikationsdarstellung in dieser Weise nicht vorhanden sind. Zum ersten ist dies die spezielle Berücksichtigung *seitenorientierter* Kommunikation, da bei den hier vorgestellten Hilfsmitteln das Konzept des virtuell gemeinsamen Speichers zugrunde gelegt wurde. Zum anderen sollten Interaktionsmöglichkeiten zur Darstellung unterschiedlicher Detailinformationen und zur flexiblen Anpassung der Darstellung vorgesehen werden, weil dies der Philosophie der PARvis-Visualisierungsumgebung entspricht. Wie in anderen, ähnlichen Werkzeugen, sollte die Auswahl aus verschiedenen Topologien möglich sein (Abbildung 5.8).

Die Abbildung 5.9 zeigt das Grundprinzip der im Rahmen dieser Arbeit entwickel-



4-dim. Hypercube



3-dim. Gitter

Abbildung 5.8: Zwei Beispiele für Netztypen, die mit PARvis dargestellt werden können

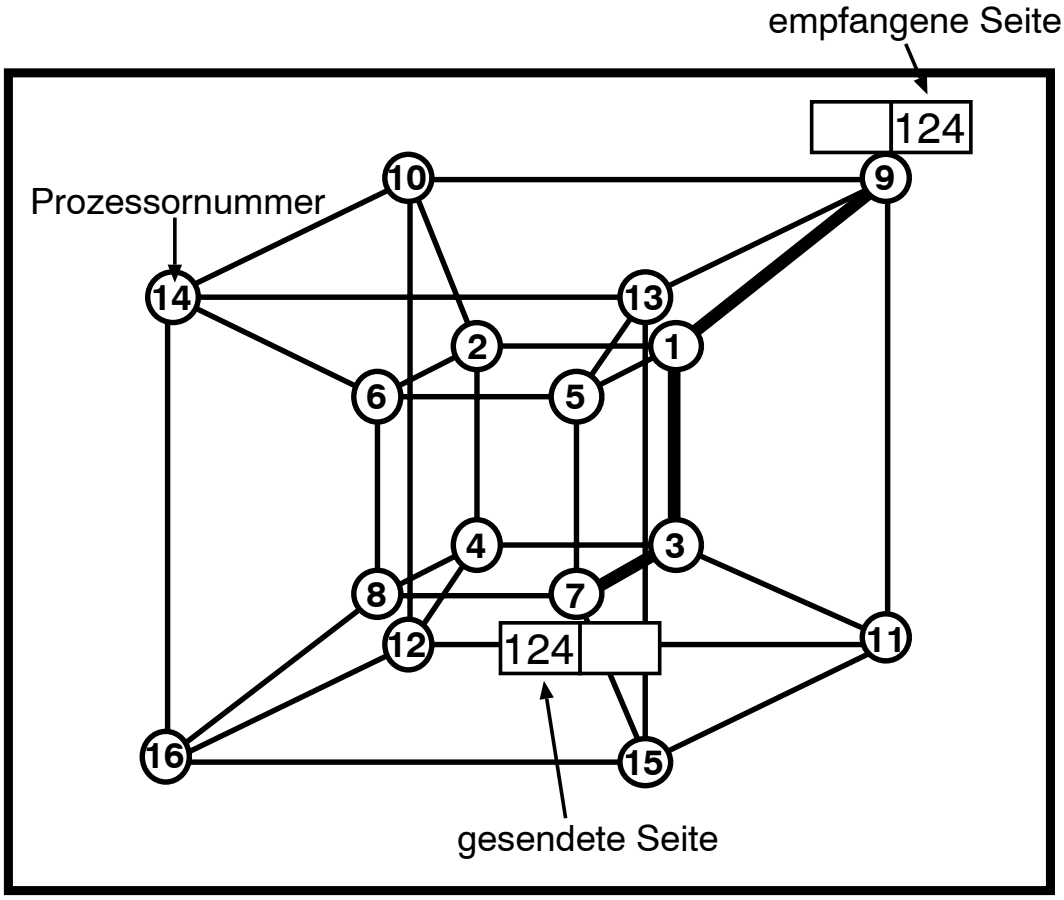


Abbildung 5.9: Visualisierung der Kommunikation im Prozessornetz

ten Werkzeugkomponente. Das Werkzeug bietet eine einfache dreidimensionale Darstellung des Netzwerkes in Form einer Parallelprojektion. Die Darstellung des Netzwerkmodells und weiterer Einzelinformationen kann über ein separates Dialogfenster in vielfältiger Weise angefordert und verändert werden. Zusammen mit einigen Funktionsaufrufen, die über ein Popup-Menü erfolgen können, ergeben sich folgende Möglichkeiten der Interaktion:

- Auswahl aus verschiedenen Netzwerkmodellen (Topologien), zum Teil mit einstellbarer Größe (Prozessorzahl) (Beispiele: siehe Abbildung 5.8)
- Darstellung von bestimmten Teilen des Netzes; es wird für jedes Modell eine Auswahl von Teilnetzen vorgegeben, aus der der Benutzer eine beliebige Kombination auswählen kann
- Drehung des Netzwerkmodells um drei Raumachsen (Veränderung der räumlichen Perspektive)
- Auswahl der Größe der Darstellung (Zooming)

- Freie Wahl der Zuordnung zwischen dargestellten Knoten und den logischen Prozessoren aus der Protokolldatei

Folgende Manipulationsmöglichkeiten haben einen direkten Einfluß auf die Art und den Umfang der dargestellten Informationen:

- Auf das Antippen eines Knotens hin werden in zwei verschiedenen Farben die momentan gesendete und die momentan empfangene Speicherseite angezeigt. Auf diese Weise können Sende- und Empfangsaktivitäten ausgewählter *Prozessoren* verfolgt werden.

- Durch Wahl einer speziellen Option werden alle Kommunikationsvorgänge für ausgewählte Seiten dargestellt, wobei der Start- und der Zielknoten mit der entsprechend eingefärbten Seite markiert werden. Auf diese Weise können Sende- und Empfangsaktivitäten ausgewählter *Seiten* verfolgt werden. Die Auswahl kann entweder mit Hilfe der Seitenübersicht (s.o.) geschehen oder durch Antippen einer Seite in der Darstellung, die im vorstehenden Listenparagraph erwähnt wurde. In diesem Zusammenhang bietet die Möglichkeit einer anderen Option dieses Fensters, direkt *alle* Speicherseiten auszuwählen, globalere Einsichten in das Kommunikationsverhalten. Das Werkzeug zeigt damit sämtliche Kommunikationsvorgänge einschließlich der Angabe der (verschieden eingefärbten) Speicherseite am Start- und am Zielknoten an.

Die durch Linien dargestellten Verbindungen zwischen den Knoten veranschaulichen durch ihre einem Farbspektrum entnommene Färbung die Menge der zum Beobachtungszeitpunkt über die Verbindung übertragenen Information.

Falls die Zahl der im dargestellten Systemlauf vorhandenen Prozessoren kleiner ist als die Zahl der Knoten im Netzwerkmodell, werden überflüssige Knoten durch eine geeignete Farbe und das Fehlen einer Prozessornummer kenntlich gemacht. Durch die oben beschriebene Möglichkeit zur freien Wahl der Zuordnung zwischen logischen Prozessoren und dargestellten Knoten können diese Komponenten jedoch jederzeit zur Repräsentation von logischen CPUs herangezogen werden. Falls die Anzahl der Knoten des Netzwerkmodells die Zahl der Prozessoren im visualisierten Systemlauf jedoch nicht erreicht, werden nur so viele CPUs wie aufgrund der Knotenzahl möglich, in der Reihenfolge aufsteigender Prozessornummern dargestellt. Aber auch hier hilft die beschriebene Möglichkeit zur freien Wahl der Zuordnung zwischen logischen Prozessoren und dargestellten Knoten weiter, indem hierdurch auch andere CPUs aus dem Systemlauf den Prozessoren des dargestellten Netzes zugeordnet werden können.

Darstellungen wie die soeben beschriebene eignen sich zur Beobachtung der Kommunikationsmuster und der Netzbelastung anhand beispielhafter Prozessortopologien. Anhand solcher Beobachtungen können Engpässe im Netz aufgespürt und gegebenenfalls beseitigt werden. Mit dem beschriebenen Werkzeug lassen sich ebenso konkrete, detaillierte Kommunikationsvorgänge nachvollziehen und ihre Wege über

das Prozessornetz nachverfolgen. Dadurch können interessante Einzelfälle im Detail untersucht werden, um die Problemursache zu analysieren und im Ursprungssystem zu korrigieren.

5.3 Eingabe und technische Voraussetzungen

Zu den fest vorgegebenen Spezifikationen eines Programmsystems gehört in der Regel der Aufbau der Eingabedaten, der im folgenden Abschnitt beschrieben wird. Zusätzlich soll die in den hier vorgestellten Werkzeugkomponenten verwendete Datenstruktur zur Speicherung dieser Eingabedaten beschrieben werden. Da es sich bei den Entwicklungsarbeiten im Rahmen dieser Arbeit um Erweiterungen eines bereits vorhandenen Programmsystems handelt, waren auch die weiter unten erwähnten Elemente der Entwicklungsumgebung vorgegeben.

5.3.1 Die Protokolldatei

Das PARvis-System wurde zunächst als Hilfsmittel zur Visualisierung von Prozeßwechselaktivitäten entwickelt [1]. Da PARvis ein *integriertes* Werkzeug ist, existiert auch nur eine einzige, gemeinsame Schnittstelle zum Einlesen der Protokolldaten [1, Abschnitt 4.3]. Über diese Schnittstelle können sowohl Prozeßwechsel- als auch Seiteninformationen eingelesen werden, allein oder aber in beliebiger Mischung, so daß beide Aspekte an einer komplexen Protokolldatei studiert werden können. Da man es bei den Untersuchungsobjekten mit komplexen, vernetzten Systemen zu tun hat, ist eine isolierte Betrachtung des einen oder anderen Aspekts ohnehin nur selten sinnvoll. Zudem geben Werkzeuge des einen Komplexes auch in bezug auf Fragestellungen des anderen Komplexes oft wertvolle zusätzliche Aufschlüsse.

Die Eingabedateien müssen im ASCII-Format vorliegen. Das hat den Vorteil, daß sie auch direkt mit Hilfe eines Editors eingesehen und notfalls verändert werden können. Außerdem ist so eine problemlose Portierbarkeit zwischen verschiedenen Rechnersystemen gegeben. Im folgenden soll der Aufbau der seitenbezogenen Informationen in einer von PARvis akzeptierten Protokolldatei erläutert werden:

Der zentrale Begriff in Systemen mit virtuell gemeinsamem Speicher ist die *Speicherseite*; es sind ihre Manipulationen, die die Speicheraktivitäten eines derartigen Systems kennzeichnen. Aus diesem Grunde liegen die Speicherinformationen in der einzulesenden Protokolldatei seitenorientiert vor, d.h. ein Datensatz beschreibt *ein* Ereignis auf *einer* Speicherseite. Die Reihenfolge der Datensätze entspricht dabei der chronologischen Abfolge der ihnen zugrundeliegenden Ereignisse. Alle Elemente in einem Datensatz, d.h. Schlüsselwörter und numerische Felder, sind durch ein oder mehrere Leerzeichen voneinander getrennt. Jeder der Datensätze muß zu Beginn das Schlüsselwort PAGE# enthalten, damit das PARvis-System daran einen Satz mit Speicherseiteninformationen erkennen kann. Darauf folgt die Nummer der Seite, auf die sich der im folgenden gekennzeichnete Zustand oder die Zustandsänderung

bezieht. Anschließend folgen in beliebiger Reihenfolge verschiedene Datenfelder, die jeweils von einem Schlüsselwort, bestehend aus zwei Großbuchstaben, eingeleitet werden. Hinter jeder dieser Einleitungen stehen, je nach Art des Schlüsselworts, eine oder mehrere Prozessornummern, die in der durch das Schlüsselwort gekennzeichneten Beziehung zu der vorab genannten Seite stehen. Die möglichen Datenfelder mit ihren Schlüsselwörtern sind:

- *Das Manager-Feld (MF)*: hier steht die Nummer des Prozessors, der der Manager für diese Seite ist.
- Im *Besitzer-Feld (OF)* befindet sich die Nummer des Prozessors, der nach der Information des Managers der aktuelle Besitzer der Seite ist (siehe auch Feld RB).
- Nach dem Schlüsselwort *RT (Read Table)* folgt eine Liste derjenigen Prozessoren, die über eine Leseberechtigung auf die Seite verfügen.
- Das *Schreiber-Feld (WF)* enthält den Prozessor, der über eine Schreibberechtigung auf diese Seite verfügt.
- Die Tabelle der *Schreiber bei schwach-kohärentem Zugriff (WT)* enthält eine Liste der Knoten, die gleichzeitig auf lokalen Kopien der Seite asynchron schreiben dürfen.
- In der ebenfalls als Liste aufgebauten *Changing-Tabelle (CT)* befinden sich die Prozessoren, die schon in der Read Table oder einem Schreiber-Feld vorhanden sind, jedoch diese Operationen mit der Seite noch nicht ausführen können, da sie noch auf das Ende des Zusendevorganges an sie warten (Empfangsstatus) oder aber noch damit beschäftigt sind, alle anderen Kopien der Seite zu invalidieren.
- Im *Sender-Feld (RB)* befindet sich die Nummer des tatsächlichen Seitenbesitzers, d.h. des Prozessors, der über eine aktuelle Kopie der Seite verfügt und dafür zuständig ist, eine solche an anfragende Prozessoren zu versenden (siehe auch Feld OF).
- Das *Zählerfeld* der *lesenden Knoten (RC)* enthält die Anzahl der Einträge in der Lesetabelle.
- In einer speziellen *Warteliste (QU)* befinden sich diejenigen Prozessoren, die über den Manager eine Seitenkopie bei einem *neuen* Besitzer angefordert haben und darauf warten, daß der Besitzerwechsel auch tatsächlich vollzogen wird (also bei Diskrepanz zwischen den Feldern OF und RB).

Bis hierher wurden Datenfelder behandelt, die bestimmte Aspekte des *Zustands* einer Seite beschreiben. Damit ist gemeint, daß jede Information der oben angesprochenen Kategorien alle vorhergehenden Informationen derselben Kategorie ersetzt.

So bedeutet z.B. ein Datenfeld RT 5 7 12, daß die Prozessoren mit den Nummern 5, 7 und 12 Leseberechtigung auf die entsprechende Seite haben und kein Prozessor sonst, was immer in vorangegangenen Datensätzen zu dieser Seite enthalten gewesen sein mag. Die im folgenden beschriebenen sogenannten Wartefelder sind jedoch von anderer Art. Die in ihnen aufgelisteten Prozessornummern bedeuten eine Aufnahme in die entsprechende Menge der wartenden Prozessoren. Damit handelt es sich hier nicht wie oben um *Zustandsinformationen*, sondern um Aussagen über eine *Zustandsänderung*. Die Elimination eines Prozessors aus einer Liste von wartenden CPUs geschieht dadurch, daß die entsprechende Prozessornummer im Feld der dazugehörigen, erwarteten Aktivität auftaucht, so führt z.B. das Erscheinen eines Prozessors in der Lesetabelle zu seiner Streichung aus der Liste der auf das Lesen dieser Seite wartenden Prozessoren. In gleicher Weise führt auch der Beginn des Wartezustands bezüglich der Zusendung einer Seite (*WC*, s.u.) zur Beendigung aller anderen, möglichen Wartezustände. Es werden vier verschiedene Wartefelder akzeptiert:

- *WR*: Liste der auf das Lesen der Seite wartenden Prozessoren
- *WW*: Liste der auf das Schreiben der Seite wartenden Prozessoren
- *WK*: Liste der auf das schwach-kohärente Schreiben der Seite wartenden Prozessoren
- *WC*: Liste der auf das Zusenden der Seite wartenden Prozessoren

Dies ist ein Beispiel für eine mögliche Eingabezeile:

```
32436 PAGE#   35  OF 5  RB 5  WF 0  RT 2 5  WR 1  CT 2
```

Hier erkennt man, daß sich Seite Nr. 35 nach 32436 Zyklen seit Beginn des Systemlaufs in folgendem Zustand befindet: Der Besitzer (mit Schreibzugriff) ist CPU 5 (*OF 5 RB 5*). Dieser Prozessor sendet die Seite an CPU 2, die zwar schon über die Leseberechtigung verfügt (*RT 2 ..*), aber noch mit dem Empfang der Seite beschäftigt ist (*CT 2*). Auch Prozessor 5 (*RT .. 5*) verfügt als Besitzer natürlich über die Leseberechtigung. CPU 1 wartet auf die Erteilung der Berechtigung zum Lesen (*WR 1*). An diesem Beispiel erkennt man auch, daß nicht alle Felder in einem PAGE-Datensatz vorkommen müssen. Unerwähnte Felder bedeuten, daß die entsprechenden Zustandsparameter unverändert bleiben. So weist der obige Datensatz nicht auf ein Wechseln des Managers hin, da das Feld mit dem Schlüsselwort *MF* fehlt.

5.3.2 Interne Darstellung der Seitenzustände

Die einzulesenden Protokolldaten liegen, wie oben dargelegt, seitenorientiert vor, d.h. ein Datensatz steht für einen neuen Zustand einer bestimmten virtuellen Speicherseite. Vor der Aufbereitung dieser Informationen zum Zweck der Darstellung auf

```
typedef struct
{
  LongInt  ManagerField;        /* Manager (MF)                       */
  LongInt  OwnerField;          /* Besitzer (Inform. des Man.) (OF)   */
  LongInt *ReadTable;           /* lesende CPUs (RT)                  */
  LongInt  WriteField;          /* schreibende CPU (WF)               */
  LongInt *WeakTable;           /* schw. koh. schreibende CPUs (WT)   */
  LongInt *ChangingTable;       /* empf. oder inval. CPUs (CT)        */
  LongInt  ReceivedBy;          /* tatsächl. Besitzer, Sender (RB)    */
  LongInt  ReadCount;           /* Anz. lesender CPUs (RC)            */
  LongInt *WaitingTableRead;    /* auf Lesezugr. wart. CPUs (WR)      */
  LongInt *WaitingTableWrite;   /* auf Schreibzugr. wart. CPUs (WW)   */
  LongInt *WaitingTableWeak;    /* auf schw.koh.Schr.wart.CPUs (WK)   */
  LongInt *WaitingTableC;       /* auf Zusendung wart. CPUs (WC)      */
  LongInt *WaitingQueue;        /* auf tats. Bes. wart. CPUs (QU)     */
} PageState;
```

Abbildung 5.10: C-Datenstruktur zur internen Darstellung der Seitenzustände (einige Felder, die nur programmtechnische Verwendung finden, wurden nicht aufgeführt)

die eine oder andere Weise müssen diese geeignet im Visualisierungssystem abgelegt werden. Hierzu benötigt man eine Technik, die es erlaubt, daß zu jedem Zeitpunkt der Gesamtzustand jeder beliebigen Seite abrufbar ist. Aus diesem Grund wurde ein eindimensionales Feld gewählt, das mit den Nummern der Speicherseiten indiziert ist und dessen Elemente Instanzen eines Strukturtyps sind, der den Zustand einer Speicherseite repräsentiert. Dieser Strukturtyp enthält Elemente, die den in Abschnitt 5.3.1 erwähnten Datenfeldern entsprechen. Er hat das in Abbildung 5.10 gezeigte Aussehen. LongInt ist der intern in PARvis verwendete Typ, der Integer-Zahlen (32 Bit) beschreibt und in Abhängigkeit vom zugrunde liegenden Rechnersystem definiert werden muß. Für jedes skalare Datenfeld (mit nur einer Prozessornummer als Eintrag) gibt es in der Struktur eine solche Integer-Variable. Für jedes vektorielle Datenfeld (mit einem oder mehreren Prozessornummereinträgen) hingegen wurde ein weiteres eindimensionales Feld in die Datenstruktur eingefügt (erkennbar am Zeigertyp, der durch einen Stern vor dem Variablennamen gekennzeichnet ist). Da diese Felder aufgrund der Eigenarten der Eingabedaten verschieden lang sein können (z.B. ist die Anzahl der Prozessoren mit Leseberechtigung variabel), enthalten sie als ersten Eintrag die Gesamtlänge des Feldes, die übrigen Einträge entsprechen dann den Prozessoren, die in der entsprechenden Beziehung zu der Seite stehen. Da die verwendete Programmiersprache C das dynamische Anlegen und Freigeben von Speicherplatz auch für Felder zuläßt, können auf diese Weise die relevanten Informationen bezüglich einer Speicherseite sehr effizient ohne Vergeudung von Speicherplatz abgelegt werden. Zwei andere mögliche Alternativen, die für die Speicherung solcher Prozessorlisten innerhalb einer Zustandsbeschreibung für eine Seite in Frage kommen, sollen hier noch kurz bewertet werden. Die klassische dynamische Datenstruktur für einen solchen Fall variabel langer Tabellen, die verkettete Liste, ergäbe hier ein sehr schlechtes Verhältnis von wirklich genutztem Speicherplatz zu dem, der nur zu Verwaltungszwecken benötigt wird. Jedes Element enthielte nämlich neben dem Zeiger auf die nächste Instanz nur einen einzigen Eintrag etwa gleicher Größe

für die Nummer des Prozessors. Eine Bitkette, die für jede in Frage kommende CPU ein Bit enthält, das gleich eins ist, falls der entsprechende Prozessor in der Liste vorkommt, würde eine Prozessorliste sehr speichereffizient repräsentieren, falls die Anzahl der Prozessoren nicht zu hoch wird. Hier wird jedoch für jede Liste eine feste Menge an Speicherplatz benötigt, die von der Gesamtzahl der in dem System enthaltenen Prozessoren abhängt. Daher ist diese Speichermethode für höhere Prozessorzahlen gegenüber der zuerst beschriebenen und implementierten Methode mit variabel langen Feldern bezüglich Speicherökonomie schlechter, da in der Regel nur eine kleinere, begrenzte Zahl von Prozessoren in solchen Listen vorkommt, eine weitaus geringere als die Gesamtzahl.

Die Möglichkeit der dynamischen Größenanpassung von Feldern wurde auch für das Gesamtfeld aller Speicherseiten ausgenutzt, indem die Gesamtheit der in dem System enthaltenen Seiten nicht vorab in der Protokolldatei ähnlich einer Deklaration angegeben werden muß. Das Feld vergrößert sich automatisch, sobald eine Speicherseite in der Eingabedatei vorkommt, deren Nummer größer ist als die bisher größte Seitennummer. Da die Nummern der Seiten als Index für das Feld verwendet werden, müssen leere Elemente angelegt werden, wenn eine neue, höhere Seitennummer eingelesen wird, die um mehr als eins von der bisherigen höchsten Seitennummer differiert. Derartige neu eingefügte Elemente werden jedoch solange als unbenutzt markiert und von jeglicher Darstellung ausgenommen, bis eine Seite mit passender, dazugehöriger Nummer eingelesen wird. Hieraus ergibt sich eine gewisse Ineffizienz, wenn Seitenummern nicht fortlaufend vergeben werden. Dieser Nachteil wird jedoch durch die Geschwindigkeit des Zugriffs auf Seiteninformationen aufgewogen, die sich durch die Verwendung der Seitennummer als Index auf das Datenfeld ergibt.

5.3.3 Die Entwicklungsumgebung

Da die beschriebenen Werkzeuge eine Weiterentwicklung des bestehenden PARvis-Systems darstellen, waren die Komponenten der Entwicklungsumgebung vorgegeben. Sie sind in [1, Kapitel 3] ausführlich erläutert und sollen hier nur kurz erwähnt und kommentiert werden. PARvis nutzt die Möglichkeiten der unter dem Betriebssystems UNIX verfügbaren X Window-Bibliotheken, die die Erstellung komfortabler Bedienungsoberflächen unterstützen sowie die problemlose Programmierung von graphischen Darstellungen durch Bereithalten von Prozeduren zur Erstellung graphischer Grundelemente ermöglichen. Die Verwendung eines sogenannten Widget-Sets (hier OSF-Motif-Widget-Set), das X Window-Elemente zu komplexeren Bedienungselementen zusammengefaßt zur Verfügung stellt, erlaubte es, eine komfortable und optisch ansprechende Oberfläche zu schaffen. Als Programmiersprache schließlich wurde C gewählt, um die Möglichkeiten zur effizienten und systemnahen Programmierung zu nutzen. Die erste Ausbaustufe von PARvis nach [1] stellte zudem eine ganze Reihe von grundsätzlichen Konzepten und Konstrukten zur Verfügung, die bei der Programmierung der hier vorgestellten Werkzeuge mitbenutzt werden konnten.

Kapitel 6

Benutzung der Visualisierungsmöglichkeiten

Die im Rahmen dieser Arbeit vorgestellten Werkzeugkomponenten sollen praktische Hilfsmittel bei der Untersuchung paralleler Rechensysteme sein. Daher müssen sie - über die grundsätzlichen Konzepte hinaus - die praktische Anwendung problemlos ermöglichen. Aus diesen Gründen wird in diesem Kapitel die Benutzung der Hilfsmittel erläutert. Dabei sollen anhand eines Beispiels die unterschiedlichen Komponenten in ihrer konkreten Anwendung und Handhabung vorgestellt werden. Als Beispiel dient eine in [13] als Ergebnis einer Simulation erzeugte und untersuchte Protokolldatei. Daher werden die auftretenden Effekte innerhalb dieser Simulationsergebnisse nicht näher erklärt und stehen hinter der Behandlung der Werkzeuge zurück. Der an den Details interessierte Leser wird auf die entsprechende Arbeit verwiesen.

6.1 Einbettung in die Bedienungsoberfläche PARtools

Die Menüleiste des PARvis- (PARtools-) Hauptfensters enthält zwei Pulldown-Menüs mit den Bezeichnungen **Global Displays** und **Local Displays**. Sie enthalten die Aufruf-Auswahlpunkte sämtlicher Werkzeuge des PARvis-Visualisierungssystems. Unter **Global Displays** finden sich alle Hilfsmittel, die sich grundsätzlich, bis auf die Benutzung von Filterfunktionen, auf die gesamte Menge der darzustellenden Systemkomponenten wie Prozessoren oder Speicherseiten beziehen. Unter **Local Displays** sind diejenigen Werkzeuge erreichbar, vor deren Aufruf eine Vorauswahl bestimmter Komponenten erfolgen muß. Die dieser Aufteilung zugrundeliegende Idee ist, daß ein Anwender sich zunächst einen Überblick mit Hilfe der Hilfsmittel der ersten Kategorie verschafft, um dann anschließend ausgewählte, interessante Komponenten mit Werkzeugen der zweiten Kategorie genauer zu untersuchen. Folgerichtig findet man im **Global-Displays**-Menü

- die Speicherseiten-Übersicht unter **Page Overview**,
- die CPU-Seiten-Darstellung unter **CPU Pages**,
- die Kommunikationsdarstellung im Netzwerk unter **Net Display** und
- die Kommunikationszeitlinien unter **Comm. Timeline**.

Demgegenüber erreicht man unter dem Menü **Local Displays**

- die Tabelle der Beziehungen zwischen Seiten und Prozessoren unter **Page/CPU**,
- Zeitliniendarstellungen der Seitenaktivitäten unter **Page Timeline** und
- eine Speicherseiten-Übersicht für zuvor ausgewählte Seiten unter **Page Overview**.

Das Pulldown-Menü **Filter** enthält den Auswahlpunkt **page aspects**. Wird er betätigt, erscheint ein Dialogfenster, das grundsätzliche Einstell- und Filtermöglichkeiten für alle Werkzeuge zur Darstellung von Speicheraktivitäten bereithält.

Im folgenden wird die Bedienung der einzelnen Darstellungsmethoden erläutert. Sie besitzen einige Gemeinsamkeiten, die vorweg kurz erwähnt werden sollen. Nach Aufruf jeder der hier vorgestellten Werkzeugkomponenten erscheint ein neues X-Fenster, in dem die entsprechenden Darstellungen angezeigt werden. Jedes Fenster hat eine vom System vorgegebene Standardposition und -abmessung. Beide können entsprechend der Fenster-Oberflächen-Philosophie durch Mausoperationen verändert werden. Dabei übernehmen es die PARvis-Programmodule, die Darstellungen an die neuen Rahmenbedingungen so anzupassen, daß die Visualisierungen korrekt und proportioniert bleiben. Dies wird durch Änderungen der Abstände und Größenverhältnisse erreicht. Das gleiche gilt auch, wenn mit Hilfe des unter [1, Abschnitt 5.3.1] beschriebenen Dialogs ein anderer Zeichensatz für alle Darstellungswerkzeuge ausgewählt wird.

Jedes Visualisierungsfenster kann durch ein Popup-Menü, das beim Drücken der rechten Maustaste erscheint, wenn sich der Mauszeiger innerhalb der Fensters befindet, auf verschiedene Arten manipuliert werden. Die so abrufbaren Funktionen werden in Verbindung mit den einzelnen Werkzeugen erläutert. Zwei dieser Funktionen jedoch sind allen Fenstern gemeinsam:

- Mit **close** kann ein Visualisierungsfenster geschlossen und die Darstellung damit beendet werden.
- Mit **SnapShot** ist der Ausdruck des Fensterinhalts möglich [1, Abschnitt 5.2.9].

Beide Funktionen und die zugehörigen Menü-Auswahlpunkte sind bereits Bestandteile der ersten PARvis-Version und wurden in die neuen Komponenten integriert.

6.2 Die Benutzung der Speicherseiten-Übersicht

Gemäß der oben erläuterten Strategie, von allgemeineren Darstellungen zu spezielleren überzugehen, ist es sinnvoll, sich nach Einlesen der Protokolldaten zunächst einen Überblick über alle im System vorhandenen Speicherseiten zu verschaffen. Die Abbildung 6.1 zeigt die Seitenübersicht für das gewählte Beispiel, das 79 Speicherseiten umfaßt. Bei Veränderungen der Fensterdimensionen werden die Rechtecke abhänging vom Verhältnis der Fensterhöhe zur Fensterbreite automatisch umgeordnet, wobei ihre Form (möglichst quadratisch) erhalten bleibt. In dem hier vorgestellten Beispiel enthalten die Rechtecke (Repräsentanten der Speicherseiten) neben der Seitennummer (gekennzeichnet durch ein vorangestelltes *P*) die Zusatzinformation Manager (*MF*) der Seite. Der Manager bestimmt hiermit auch die Farbe (im Bild Graustufen) des Rechtecks, die entsprechende Zuordnung kann anhand der am rechten Rand liegenden Legende abgelesen werden. Die Abbildung der Prozessoren auf die Farben geschieht automatisch mit Hilfe eines Farbspektrums, das vom Hauptmenü des PARvis-Systems aus unter **Settings/color translation** eingestellt werden kann. Gibt es im dargestellten System sehr viele Speicherseiten, können die Seitenrechtecke sehr klein werden. Inschriften, die in diese Rechtecke dann nicht mehr hineinpassen, werden unterdrückt. In diesem Fall kann die zusätzliche Information trotzdem noch anhand der Farbzuordnung abgelesen werden. Die folgenden Informationen können für die Seiten angezeigt werden:

- Manager
- Besitzer nach der Information des Managers
- tatsächlicher Besitzer
- CPU mit Schreibberechtigung
- Anzahl lesender CPUs
- Anteil der Kommunikationszeit

Die Auswahl erfolgt über das Popup-Menü des Fensters, welches nach Drücken der rechten Maustaste erscheint, wenn sich der Mauszeiger innerhalb des Anzeigefensters befindet (Abbildung 6.2). Mit dem Auswahlpunkt **exhibit** kann man daraus ein Pulldown-Menü öffnen, aus dem einer der o.g. Punkte mit dem Mauszeiger ausgewählt werden kann. Wird die Anzahl der lesenden CPUs angezeigt, entspricht die dargestellte Zahleninformation nicht einer Prozessornummer, sondern einer Prozessoranzahl. Damit eine sinnvolle Darstellung für den Anteil der Kommunikationszeit erfolgen kann, muß zuvor eine Historienliste erstellt worden sein. Ist diese Voraussetzung erfüllt, erhält man eine Anzeige ähnlich der in Abbildung 6.3. Kleine Tortendiagramme verdeutlichen den Anteil der Kommunikationszeit, zusätzlich erfolgt eine entsprechende prozentuale Angabe. Die für die Tortendiagramme verwendeten Farben können mit Hilfe des separaten Dialogfensters ausgewählt werden, das global

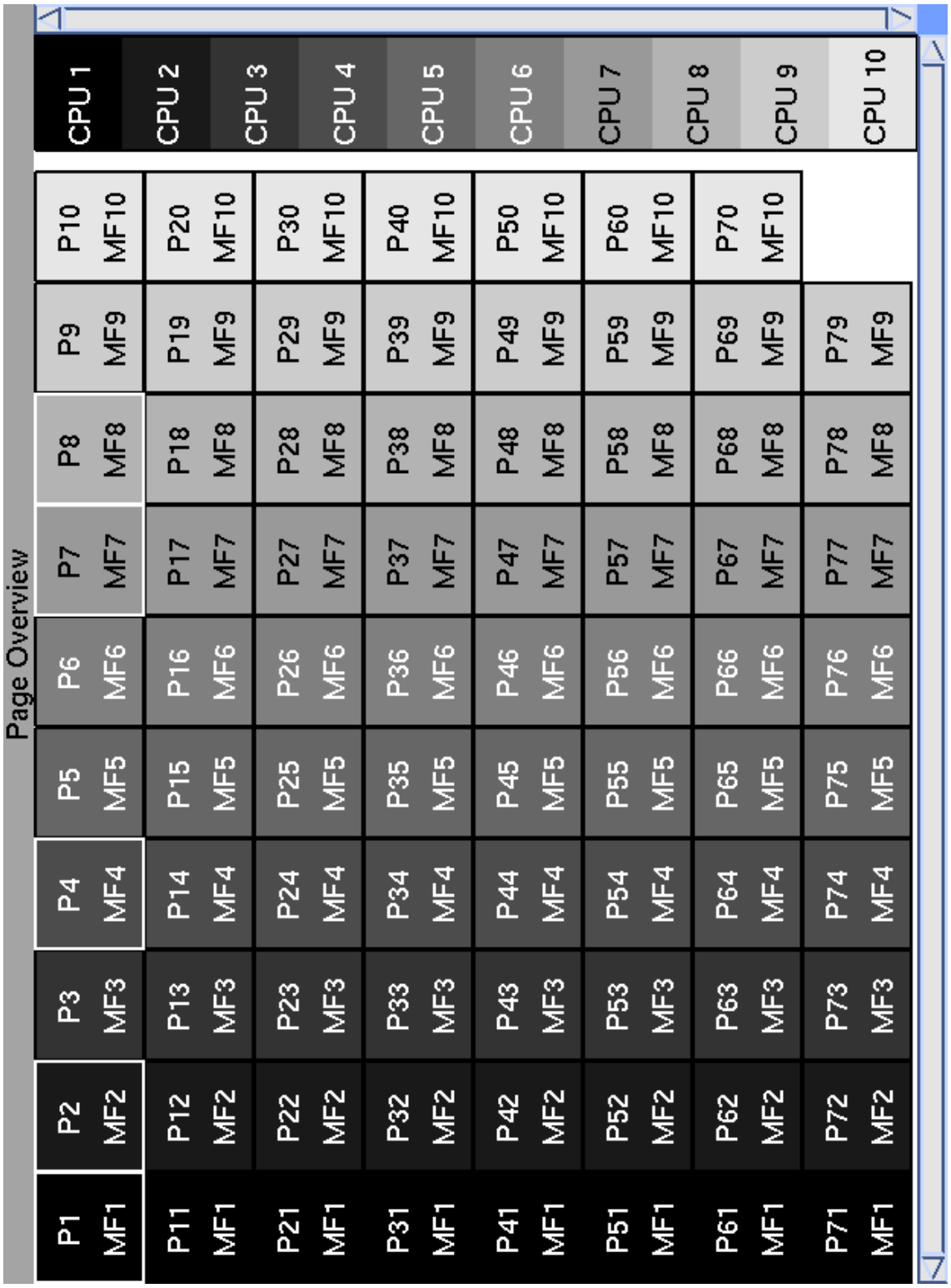


Abbildung 6.1: Die Speicherseiten-Übersicht mit Zusatzinformation: Manager

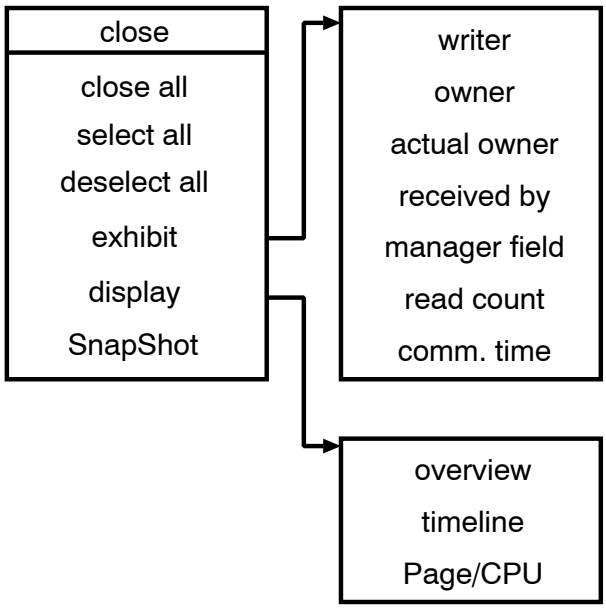


Abbildung 6.2: Popup-Menü der Seitenübersicht

für alle Speicherdarstellungen Einstellmöglichkeiten zur Verfügung stellt (Abschnitt 6.8).

Das hier erläuterte Fenster ermöglicht die Auswahl von Seiten, auf die sich weitere Werkzeuge in ihren Darstellungen beziehen bzw. beschränken können. Der Anwender wählt Speicherseiten aus, indem er den Mauszeiger in das entsprechende Rechteck bringt und die linke Maustaste drückt. Eine ausgewählte Seite wird durch eine weiße Rechteckumrandung (statt einer schwarzen) kenntlich gemacht (Abbildung 6.1). Bereits gewählte Speicherseiten können auf dieselbe Weise wieder abgewählt werden. Bewegt man die Maus bei gedrückter linker Taste, wird ein Linienrechteck aufgespannt. Die von diesem überdeckten Seitenrechtecke verändern in gleicher Weise wie oben beschrieben ihren Zustand von "gewählt" in "nicht gewählt" oder umgekehrt. Somit können größere Seitenmengen auf einfache Weise ausgewählt werden. Die grundlegenden Mausfunktionen wie Bereitstellung der Zeigerkoordinaten im Anzeigefenster und die Darstellung des Linienrechteckes sind Mechanismen, die von der ersten Ausbaustufe des PARvis-Systems übernommen werden konnten. Mit Hilfe des **select all**-Auswahlpunktes des Popup-Menüs kann der Anwender in einem Schritt alle Speicherseiten auswählen. Mit dem **deselect all**-Auswahlpunkt oder durch Drücken der mittleren Maustaste kann er alle Speicherseiten abwählen.

Möchte man die Seitenübersicht auf einen Teil der Seiten beschränken, wählt man die gewünschte Teilmenge wie oben beschrieben aus und betätigt den Popup-Menü-Auswahlpunkt **display/overview** oder den **overview**-Auswahlpunkt im Hauptmenü unter **Local Displays**. Damit wird ein weiteres Seitenübersichtsfenster erzeugt, das nur die zuvor ausgewählten Seiten darstellt. Auch in diesem Stadium

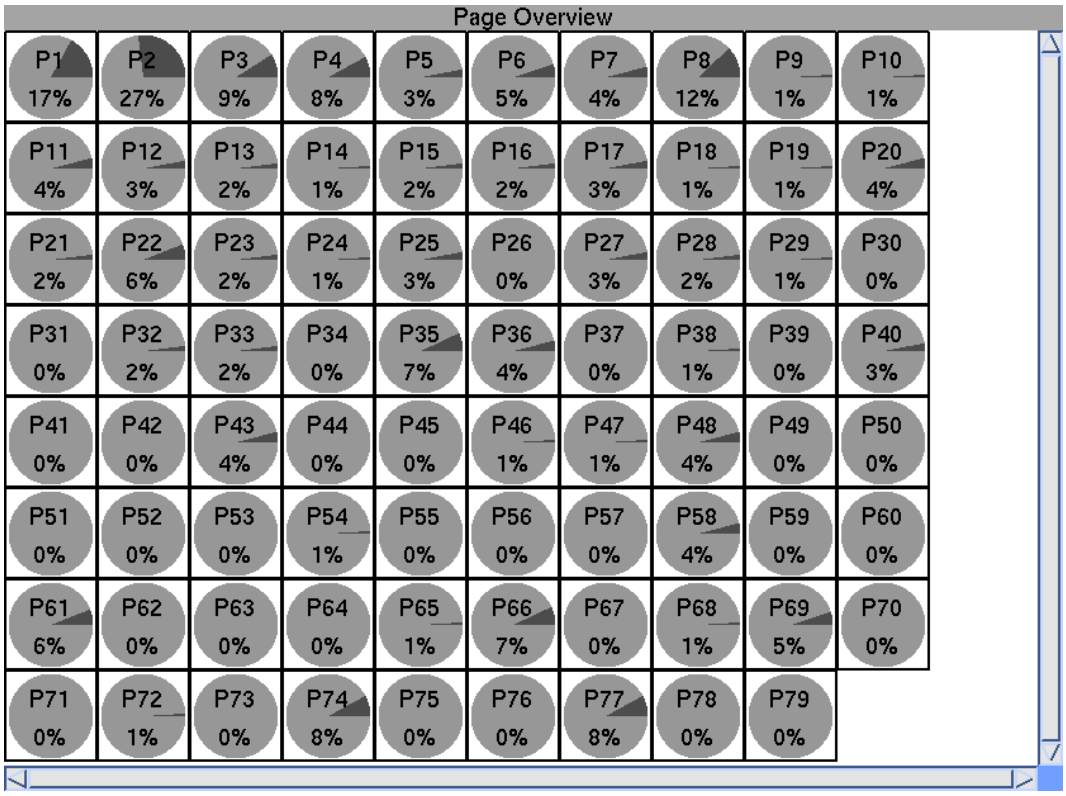


Abbildung 6.3: Die Speicherseitenübersicht mit Zusatzinformation: Anteil der Kommunikationszeit

können im ersten Überblicksfenster noch Änderungen an der Auswahl vorgenommen werden, die sofort in der Darstellung des zweiten Fensters berücksichtigt werden. Nach Drücken des Popup-Menü-Auswahlpunktes **fix actives** im zweiten Fenster kann dieses in gleicher Weise wie das erste für das normale Auswählen von Seiten verwendet werden. Auf diese Weise kann die Übersicht auf bestimmte, interessante Seiten beschränkt werden, die dann größer und deutlicher dargestellt werden. Insbesondere, wenn sehr viele Speicherseiten im zu betrachtenden Rechnersystem enthalten sind, werden die Seitenrechtecke sehr klein, und eingeschriebene Informationen können unterdrückt werden. In der reduzierten Überblicksdarstellung können diese Informationen dann wieder angezeigt werden.

Dem hier gewählten Beispiel folgend wird die Seitenauswahl zunächst zugunsten der Speicherseite 2 getroffen. Sie besitzt nach Abbildung 6.3 den größten Kommunikationszeitanteil im betrachteten Zeitintervall (27%), was darauf schließen läßt, daß mehrere Prozessoren die Seite alternierend benutzen (evtl. Prinzip der Datenlokalität verletzt). Nun soll eine Aktivitäts-Zeitliniendarstellung für die Seite 2 betrachtet werden.

6.3 Darstellung der Zeitlinien für Speicherseiten

Wurden Seiten entsprechend der gerade beschriebenen Vorgehensweise zur näheren Betrachtung ausgewählt, können Aktivitätszeitlinien für sie erzeugt werden. Dies geschieht durch das Betätigen des Auswahlpunktes **Page Timeline** im PARvis-Hauptmenü unter **Local Displays** oder direkt aus der Seitenübersicht heraus mit Hilfe der Popup-Menü-Option **display/timeline**. Für jede ausgewählte Seite wird dann ein eigenes Zeitlinienfenster wie das in Abbildung 6.4 dargestellte angezeigt. In diesem Beispiel erkennt man die Zeitskala am oberen Rand und die beiden Darstellungsbereiche, die sich darunter anschließen. Hier wurde als Zeiteinheit "Maschinenzyklen" ausgewählt. Die Balken des oberen Fensterbereichs tragen Farben und Schraffuren, die bestimmte Aktivitäten repräsentieren und in einer am rechten Fensterrand angeordneten Legende wiederzufinden sind. Die verwendeten Farben können im globalen Konfigurationsdialog ausgewählt werden. Die Aktivitätsorientierten Zeitlinien im unteren Fensterbereich tragen Farben, die den Prozessornummern oder -zahlen entsprechen und dem im vorigen Abschnitt erwähnten Farbspektrum in gleicher Weise entnommen sind. Sind die Fensterabmessungen zu klein, um alle Zeilen erkennbar wiederzugeben, werden die untersten Balkenlinien nicht mehr dargestellt. Mit einem am rechten Fensterrand angebrachten Rollbalken kann man dann den vertikalen Darstellungsausschnitt verschieben. Die Wahl des horizontalen (zeitlichen) Darstellungsausschnitts kann über die folgenden Maus-Operationen bzw. Popup-Menü-Optionen erfolgen (Abbildung 6.5):

- *Zooming*: ein durch *rechtsgerichtete* Mausbewegung bei gedrückter linker Taste erzeugtes Linienrechteck definiert in seiner horizontalen Ausdehnung den gewünschten neuen Betrachtungsausschnitt. Dieser wird beim Loslassen der Maustaste vom Fenster übernommen.

- *Verschieben*: ist der Darstellungsausschnitt auf die oben beschriebene Weise verkleinert worden, sind Verschiebungen möglich. Diese können auf unterschiedliche Arten durchgeführt werden:

  - Verschieben des am unteren Fensterrand angeordneten Rollbalkens
  - Auswahl der Option **window options/left** oder **window options/right**; dadurch wird der Darstellungsausschnitt um eine halbe Ausschnittbreite nach links bzw. recht verschoben.
  - ein durch *linksgerichtete* Mausbewegung bei gedrückter linker Taste erzeugtes Linienrechteck verschiebt den Betrachtungsausschnitt um den der horizontalen Ausdehnung entsprechenden Zeitraum nach rechts (schnelles Rechtsverschieben).

- Mit **undo zoom** kann man den letzten Zooming-Vorgang rückgängig machen (durch dynamische Organisation des Undo-Buffers können im Prinzip beliebig viele Zooming-Stufen auf diese Weise rückgängig gemacht werden).

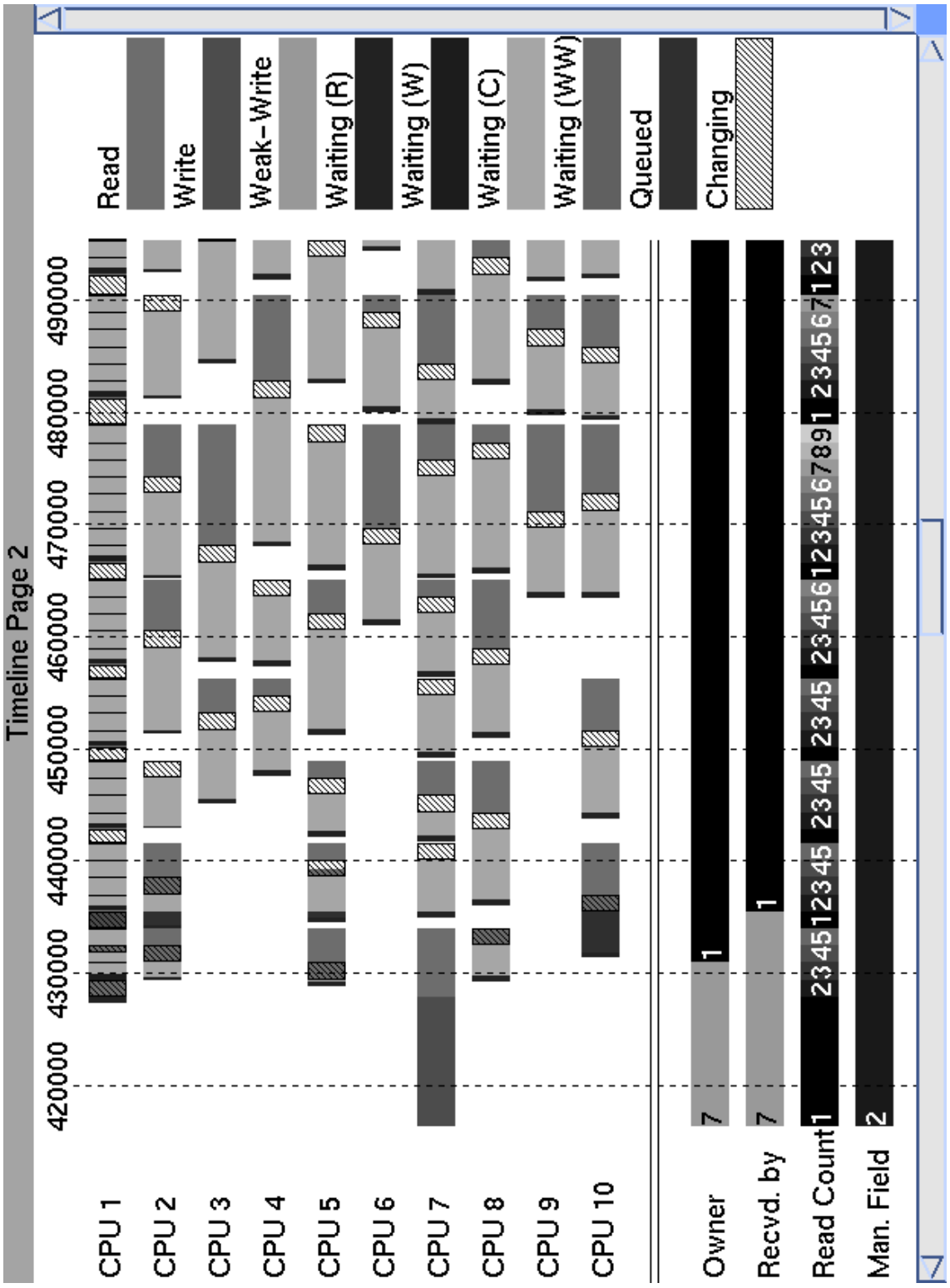


Abbildung 6.4: Eine Aktivitäts-Zeitliniendarstellung für Speicherseite 2, durch Zooming gewonnener Ausschnitt aus dem betrachteten Zeitintervall

- **window options/reset zoom** setzt den Darstellungsausschnitt auf den Stand vor der ersten Zooming-Operation zurück.

- **window options/adapt window** paßt den Betrachtungsausschnitt dem von der Historienliste definierten Zeitintervall an.

- **window options/default window** stellt den Fensterausschnitt auf eine Weise ein, die vom Hauptmenü aus unter **settings/timeline default** vorgewählt werden kann [1, Abschnitt 5.2.1]. Die dortige Einstellung definiert auch den Anfangsausschnitt, der beim Öffnen des Fensters angezeigt wird (so wie für alle Zeitliniendarstellungen in PARvis).

- **flash** (Umschaltknopf) ermöglicht die Aktivierung/Deaktivierung des Modus, in dem innerhalb eines Zeitfensters angeklickte Zeitpunkte durch senkrechte Linien markiert und in allen Zeitpunktdarstellungen angezeigt werden. Da dieser Modus für alle Speicher-Zeitliniendarstellungen gilt, wird die oben erwähnte Zeitpunktmarkierung auch in allen derartigen Darstellungen simultan durchgeführt. Das gilt auch für die Kommunikations-Zeitliniendarstellung. Jede auf der PARvis-Hauptebene durchgeführte Bewegung auf der virtuellen Zeitachse beendet diesen Modus.

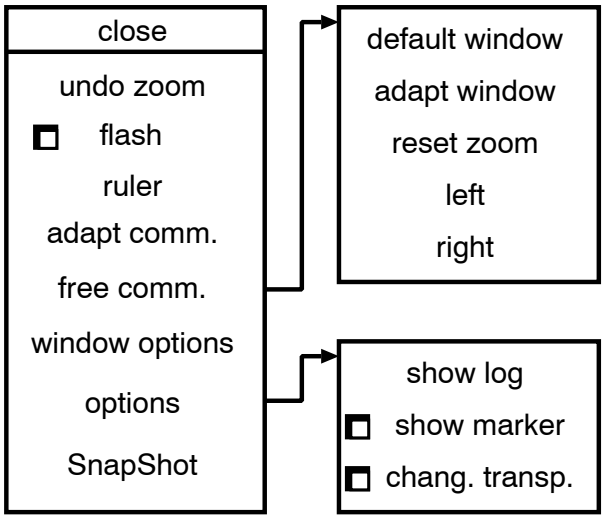


Abbildung 6.5: Popup-Menü des Aktivitäts-Zeitlinienfensters

Einige Funktionen der ersten Ausbaustufe des PARvis-Visualisierungssystems, die generell für Zeitliniendarstellungen nützlich sind, können auch in Verbindung mit dem hier beschriebenen Werkzeug aufgerufen werden. Dazu gehören ein Hilfsmittel zum exakten Ausmessen von Zeiträumen mit der Maus (**ruler**) sowie die Möglichkeit, für mit der Maus definierte Zeitabschnitte die zugehörigen Protokolldaten anzeigen zu lassen (**options/show log**). Zudem können die *Markierungen* des

PARvis-Systems in der Zeitliniendarstellung als kleine Pfeile zusammen mit ihrem zugeordneten Kommentar vermerkt werden.

Ein Changing-Zustand wird durch Schraffur symbolisiert. Mit der Umschalt-Option **options/chang. transp.** kann der Anwender bestimmen, ob zusätzlich die Farbe der beabsichtigten Aktivität hinterlegt werden soll. Der **close all**-Auswahlpunkt des Popup-Menüs der Seitenübersicht kann dazu verwendet werden, sämtliche Aktivitäts-Zeitlinienfenster in einem Schritt wieder zu schließen.

In der Abbildung 6.4 wurde ein Betrachtungsintervall gewählt, das einen Ausschnitt aus einem Zeitraum zeigt, in dem die Seite 2 oft Gegenstand von Kommunikationsvorgängen ist. Alle Prozessoren beantragen reihum Zugriff auf die Seite, rufen Page-Faults hervor und erhalten schließlich eine Kopie der Seite zugesendet, was den hohen Kommunikationsanteil von Seite 2 erklärt (Abbildung 6.3). Zudem findet ein Besitzer- (Versender-) Wechsel von CPU 7 nach CPU 1 statt (siehe *Recvd. by*-Linie in Abbildung 6.4). Parallel zur Aktivitäts-Zeitliniendarstellung soll nun eine Kommunikations-Zeitliniendarstellung zum besseren Verständnis des Kommunikationsverhaltens der Prozessoren in einem kleineren Zeitintervall, in dem der Besitzerwechsel stattfindet, betrachtet werden.

6.4 Benutzung der Kommunikations-Zeitlinien

Ein Beispielfenster für diese Zeitliniendarstellung ist in Abbildung 6.6 wiedergegeben. Deutlich sind die Sendephasen der einzelnen CPUs für verschiedene Seiten (Seitennummer durch P gekennzeichnet) an unterschiedliche Prozessoren (hinter dem kleinen Pfeil) zu erkennen. Werden die Balkenabschnitte zu klein, um beide Inschriften aufzunehmen, werden nur die übertragenen Speicherseiten angegeben, der Zielprozessor ist an der Farbe des Balkens ablesbar. Mit derselben Farbe ist die entsprechende CPU in der Beschriftung am linken Fensterrand hinterlegt. Bei sehr kurzen Balkenabschnitten wird der Not gehorchend auf jegliche Inschrift verzichtet. Die Abbildung 6.7 zeigt das Popup-Menü des Werkzeugs, alle Optionen, die die gleiche Bezeichnung haben wie im Menü des Aktivitäts-Zeitlinienfensters, haben auch hier die gleiche Funktion (siehe voriger Abschnitt). Neu sind die im folgenden beschriebenen Punkte im Untermenü **options**:

- **selected pages** (Umschaltknopf) schränkt die Darstellung auf zuvor ausgewählte Speicherseiten ein.
- **all CPUs** (Umschaltknopf) führt alle Prozessoren am linken Fensterrand auf, einschließlich derjenigen, die im Zeitintervall der Historienliste keine Sendeaktivitäten auf den zu betrachtenden Seiten durchführen. Ansonsten bleibt die Prozessorliste auf die sendeaktiven CPUs beschränkt. Diese Option ist nützlich, wenn eine vollständige Legende für die Farbzuordnung der Zielprozessoren gewünscht wird.

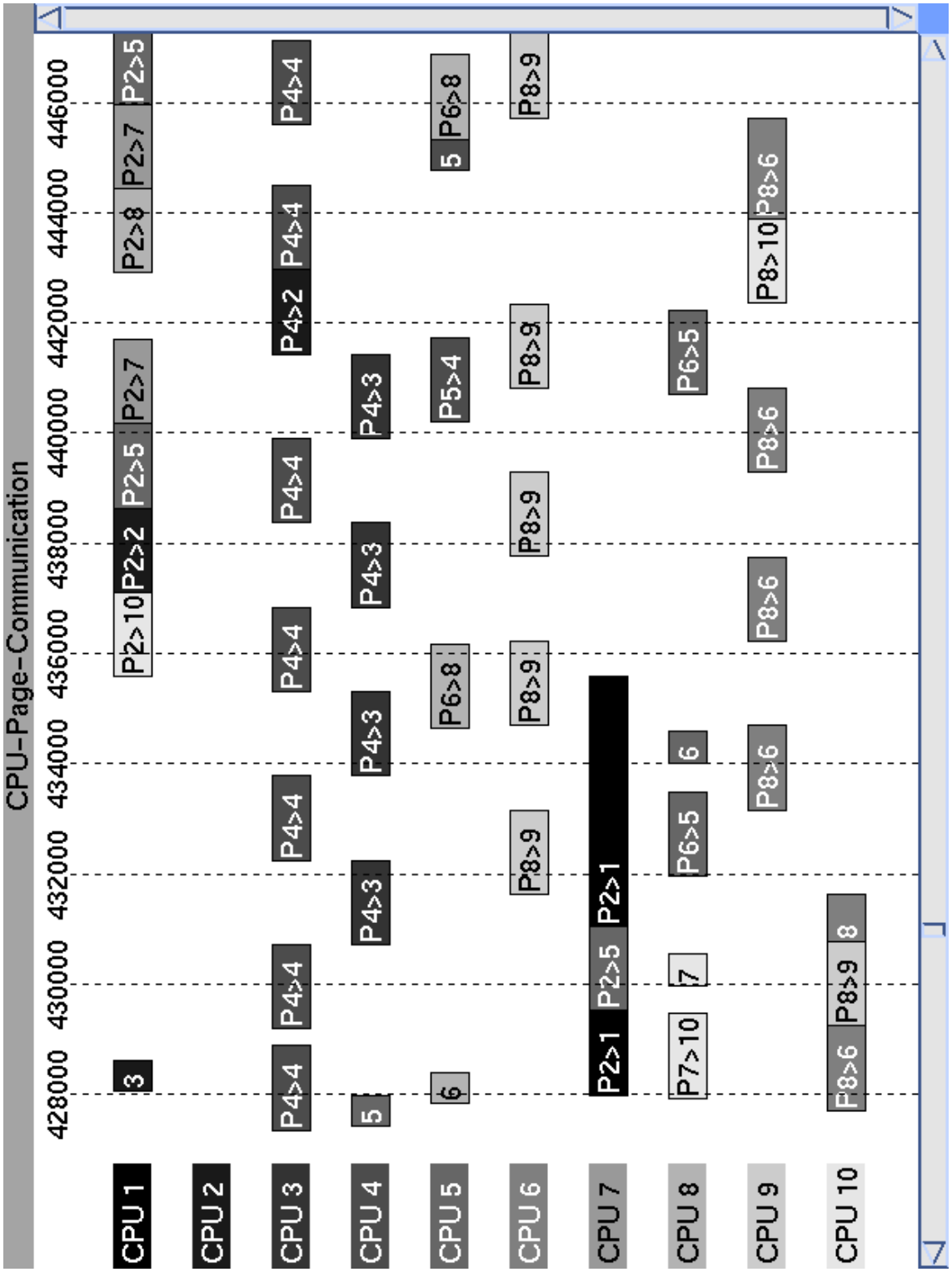


Abbildung 6.6: Kommunikations-Zeitliniendarstellung, durch Zooming gewonnener Ausschnitt

- **show links** (Umschaltknopf) veranlaßt das Einzeichnen der Linien, die am Startpunkt eines Sendebalkens beginnen und am zeitlichen Endpunkt in der Zeile des Zielprozessors enden. Abbildung 6.8 zeigt diese Linien am Beispiel (Zeitintervall wie in Abbildung 6.6, Darstellung hier jedoch auf die ersten fünf CPUs beschränkt). Diese Linien sollen die Übertragungsverbindungen deutlicher herausstellen und das Erkennen von Kommunikationsmustern erleichtern.

- **only links** (Umschaltknopf) ermöglicht es, einen Modus ein- oder auszuschalten, in dem ausschließlich diese Verbindungslinien (ohne die Balken) dargestellt werden.

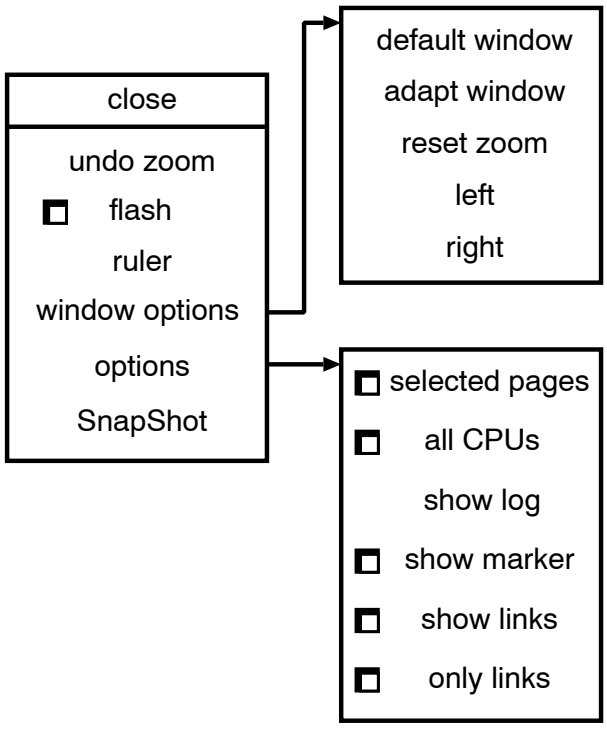


Abbildung 6.7: Popup-Menü der Kommunikations-Zeitliniendarstellung

Die Anpassung der Maßstäbe und Skalen an die eines Aktivitäts-Zeitlinienfensters geschieht durch dessen Popup-Menü. Die dortige Menü-Option **adapt comm.** bindet das (globale) Fenster für Kommunikationszeitlinien an das jeweilige Aktivitäts-Zeitlinienfenster in der Weise, daß nach o.g. Anpassungen alle Ausschnittveränderungen im zuletzt genannten Fenster vom ersteren sofort nachvollzogen werden. Es entsteht damit also eine feste Kopplung bezüglich des Betrachtungsausschnitts zwischen den beiden Zeitlinien-Werkzeugen. Diese Kopplung kann mit Hilfe der Menü-Option **free comm.** des Aktivitäts-Zeitlinienfensters wieder aufgelöst werden.

In der Abbildung 6.6 erkennt man u.a. deutlich, wie zuerst CPU 7 als Besitzer die Seite 2 an verschiedene andere Prozessoren versendet. Nach dem Besitzerwechsel zu

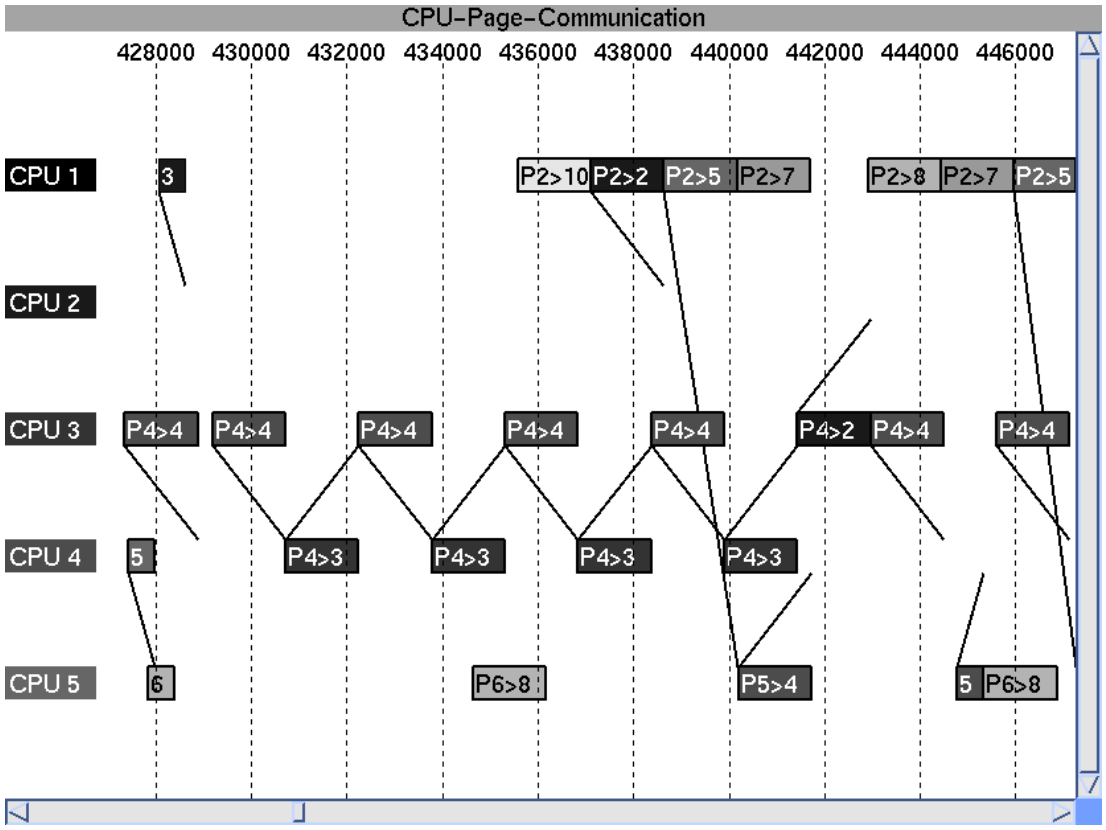


Abbildung 6.8: Kommunikations-Zeitliniendarstellung mit Linien zur besseren Verdeutlichung der Übertragungsverbindungen

CPU 1 zwischen den Maschinenzyklen Nr. 435000 und Nr. 436000 übernimmt diese dann das Versenden an die anderen CPUs. Die Liniendarstellung in Abbildung 6.8 liefert gegenüber der reinen Balkendarstellung in Bild 6.6 keine neuen Informationen. Dennoch erkennt man hier z.B. sehr deutlich den alternierenden Zugriff der Prozessoren 3 und 4 auf die Seite 4, der dazu führt, daß sich die CPUs abwechselnd die Speicherseite gegenseitig zusenden. Im folgenden wird bei der Beschreibung der Bedienung der Zeitpunktdarstellungen in den Abbildungen der Zustand bei Maschinenzyklus Nr. 430000 betrachtet. Wie aus der Abbildung 6.6 ersichtlich ist, werden hier die Seiten 2, 4, 7 und 8 versendet.

6.5 Tabellarische Darstellung der Prozessor-Seiten-Beziehungen

Unter Umständen können dem Anwender bei der Verwendung der Zeitliniendarstellungen Zeitpunkte auffallen, die er einer genaueren Untersuchung unterziehen möchte. Die nun in ihrer Anwendung beschriebene Tabelle der Beziehungen zwi-

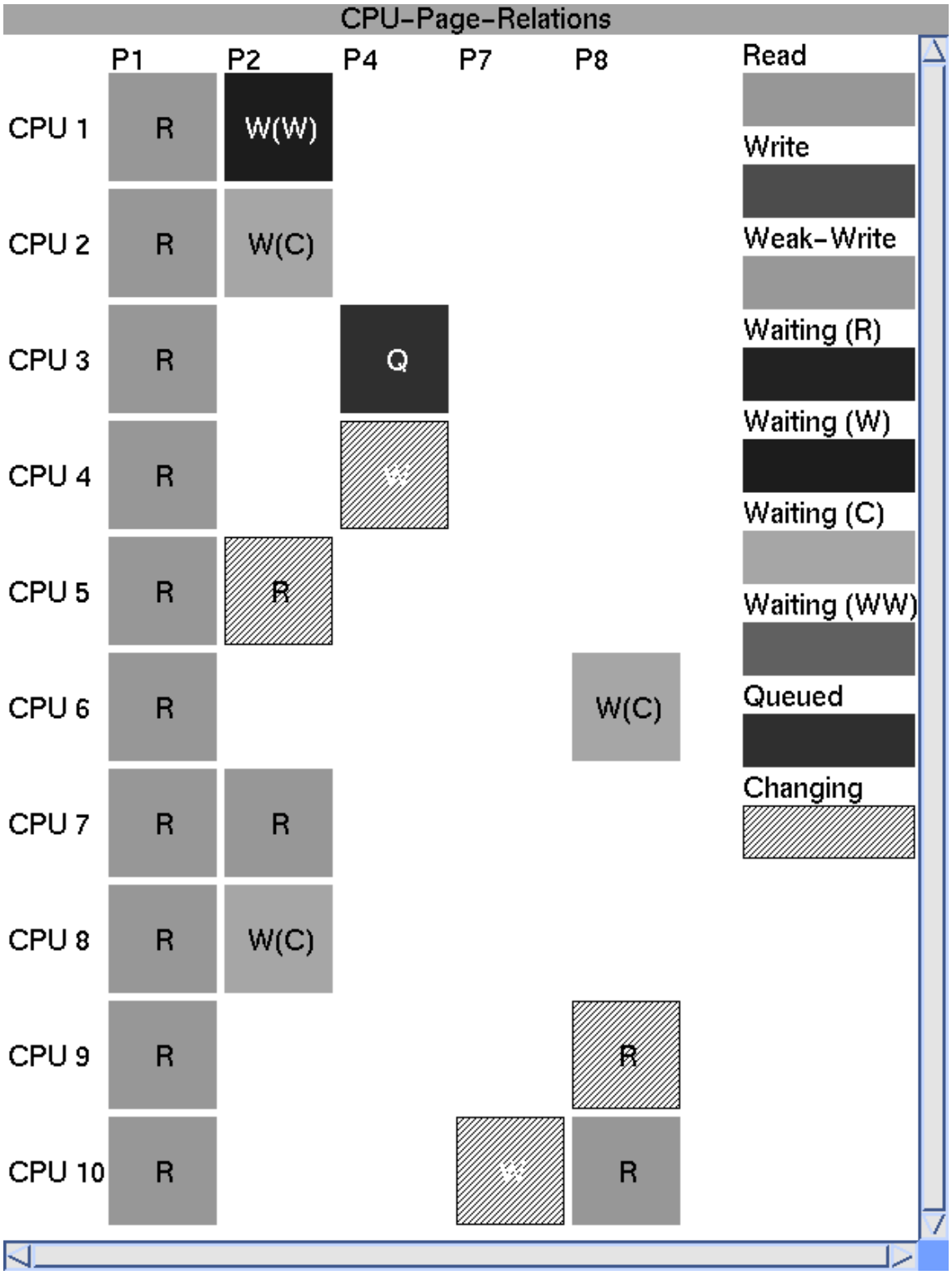


Abbildung 6.9: Prozessor-Seiten-Tabelle

schen den Prozessoren und ausgewählten Seiten kann dazu verwendet werden. Die Abbildung 6.9 zeigt die Darstellung für den o.g. Zeitpunkt des Beispiels. Zuvor wurden mit Hilfe der Seitenübersicht die Speicherseiten 1, 2, 4, 7 und 8 ausgewählt. Das hier dargestellte Fenster kann die Beziehungen dieser Seiten zu allen zehn CPUs gleichzeitig darstellen. Falls jedoch mehr Systemkomponenten zu berücksichtigen sind als das Fenster aufzunehmen in der Lage ist, werden zunächst nur die ersten, in die Darstellung hineinpassenden Komponenten wiedergegeben. Mit Hilfe der am unteren sowie am rechten Fensterrand angeordneten Rollbalken kann dann der Betrachtungsausschnitt verschoben werden, wobei sich "herausgefallene" Komponenten dann ins Blickfeld hineinschieben. Die für die Repräsentation der Aktivitäten verwendeten Farben entsprechen denen, die auch in der Aktivitäts-Zeitliniendarstellung benutzt werden und können somit im globalen Konfigurationsdialog angegeben werden. Zusätzlich zur Farbkodierung werden Abkürzungen zur Benennung der Aktivitäten verwendet, die sich unmittelbar aus der Beschriftung der Legende ableiten lassen (der Changing-Zustand ist mit der Abkürzung der beabsichtigten Aktivität beschriftet). Die Abbildung 6.10 zeigt das Popup-Menü des Werkzeugs. Da der Changing-Zustand auch hier durch Schraffur gekennzeichnet wird, kann mit Hilfe der Menü-Option **options/chang. transp.** entschieden werden, ob zusätzlich die Farbe der beabsichtigten Aktivität hinterlegt wird.

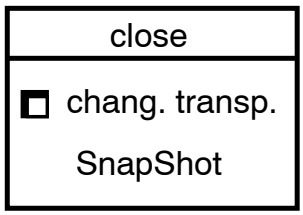


Abbildung 6.10: Popup-Menü der Prozessor-Seiten-Tabelle

In der Abbildung 6.9 erkennt man deutlich die Empfangszustände der Prozessoren 4, 5, 9 und 10 bezüglich der Seiten 4, 2, 8 und 7 (grobe Schraffur). Außerdem zeigt das Bild unterschiedliche Wartezustände verschiedener CPUs auf denselben Seiten. Alle Prozessoren besitzen Leseberechtigung auf die Seite 1.

6.6 Darstellung der CPU-orientierten Seitenaktivitäten

Auch mit einer Darstellung, die für einen Zeitpunkt für jeden Prozessor die von ihm mit verschiedenen Aktivitäten belegten Speicherseiten anzeigt, können detailliertere Betrachtungen angestellt werden, wenn interessante Zeitpunkte ermittelt worden sind. In der Abbildung 6.11 ist eine solche Darstellung für denselben Zeitpunkt wiedergegeben. Deutlich sieht man, daß jeder Prozessor durch ein Rechteck reprä-

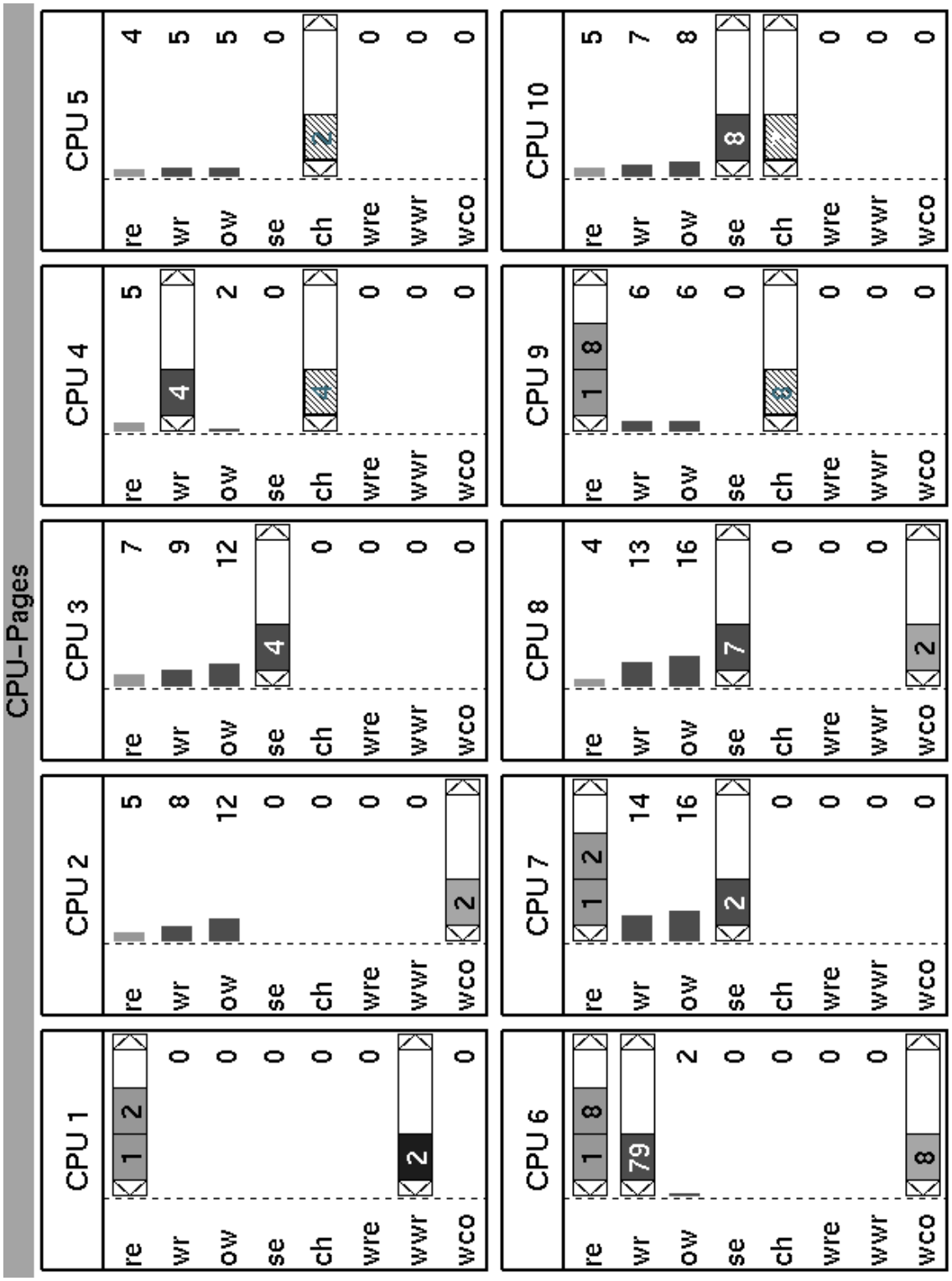


Abbildung 6.11: CPU-Darstellung mit Seitenaktivitäten

sentiert ist, das die Nummer des Prozessors trägt. Darunter folgt eine Liste der berücksichtigten Aktivitäten. Die folgenden Abkürzungen stehen dabei für zugeordnete Aktivtäten:

- **ma**: Managen (managing)
- **ow**: Besitzen nach der Information des Managers (owning)
- **re**: Lesen (reading)
- **wr**: Schreiben (writing)
- **we**: schwach-kohärentes Schreiben (weak writing)
- **ch**: Empfangen oder Invalidieren (changing)
- **se**: tatsächliches Besitzen und Versenden (sending)
- **wre**: Warten auf Lesen (waiting for reading)
- **wwr**: Warten auf Schreiben (waiting for writing)
- **wwe**: Warten auf schwach-kohärentes Schreiben (waiting for weak writing)
- **wco**: Warten auf Zusendung (waiting for communication)
- **qu**: Warten auf tatsächlichen Besitzerwechsel (queued)

Die Auswahl der interessierenden Aktivitäten erfolgt mit Hilfe des Popup-Menüs (Abbildung 6.12). Unter **show** erhält man daraus ein Pulldown-Menü, das für jeden der o.a. Punkte einen Umschaltknopf enthält, mit dem man die Darstellung der entsprechenden Aktivität ein- oder ausschalten kann. Zur Vereinfachung der Auswahl können mit Hilfe der Optionen **all acts** und **no acts** alle Aktivitätsschalter in einem Vorgang in den einen oder anderen Zustand gebracht werden.

Beim Aufruf der Werkzeugkomponente wird in jeder für eine Aktivität reservierten Zeile die Anzahl der von der CPU mit dieser Aktivität belegten Seiten durch Zahlenangabe und Balken für die relative Menge dargestellt. Nach Anklicken der Abkürzung am linken Rand des Prozessorfensters wechselt die Anzeige auf einen Rollkasten, der die Liste der entsprechenden Speicherseiten enthält, die innerhalb des Kastens durch kleine, beschriftete Rechtecke repräsentiert werden. Der Rollkasten besitzt an seinem linken und rechten Ende zwei Pfeilsymbole. Durch Anklicken dieser Markierungen kann man durch die Kette der aneinandergereihten Seitenrechtecke in Form eines gleitenden Betrachtungsausschnitts hindurchwandern, falls nicht alle Seitenangaben gleichzeitig in den Rollkasten passen. Die Farbe der Seitenrechtecke ist den Aktivitäten zugeordnet und entspricht, solange es sich um Aktivitäten handelt, die in der CPU-Seiten-Tabelle oder der Aktivitäts-Zeitliniendarstellung

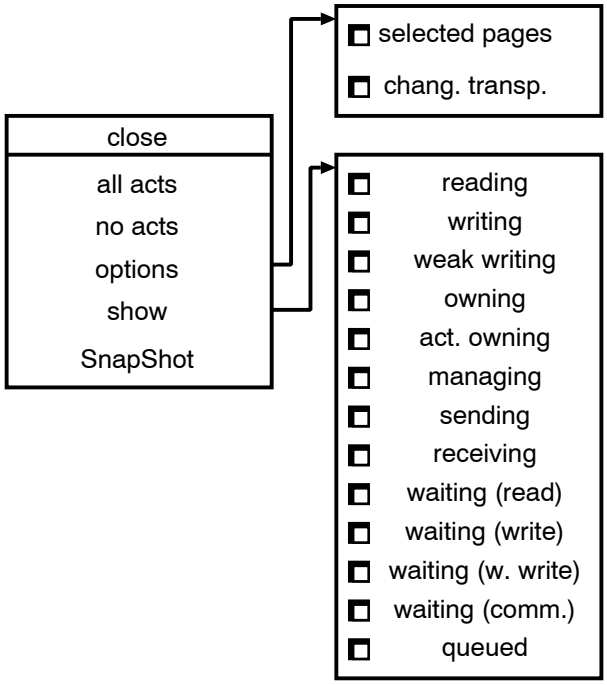


Abbildung 6.12: Popup-Menü der CPU-Darstellung mit Seitenaktivitäten

ebenfalls vorkommen, auch der dort verwendeten Farbkodierung. Die farbliche Zuordnung der anderen Aktivitäten kann jedoch ebenfalls über den globalen Konfigurationsdialog für Speicherdarstellungen angegeben werden. Der Balken für die relative Seitenanzahl und die Seitenkästchen des Rollkastens sind für die *changing* Aktion schraffiert wiedergegeben. Daher kann auch hier mit Hilfe der Menü-Option **options/chang. transp.** entschieden werden, ob zusätzlich die Farbe der beabsichtigten Aktivität hinterlegt werden soll. Die Menü-Option **options/selected pages** ermöglicht es, die Darstellung auf ausgewählte Seiten zu beschränken.

Die Kommunikationsaktivitäten, die mit den letzten Werkzeugen bereits zutage traten, können auch hier anhand der Aktivitäten sending (*se*) und changing (*ch*) abgelesen werden, da u.a. für sie je ein Rollkasten angefordert wurde (Abbildung 6.11).

6.7 Darstellung der Kommunikation im Prozessornetz

Die dritte wichtige Zeitpunktdarstellung, die Kommunikationsvorgänge darstellt und dabei die Topologie der Verbindungsstruktur zwischen den Prozessoren berücksichtigt, präsentiert sich in Form von zwei Fenstern. Zum einen wird ein Grafikfenster eröffnet, in dem das Prozessornetz in Form von Kreisen (für die Knoten) und Linien (für die Verbindungen) dargestellt wird (Abbildung 6.13). Zum anderen er-

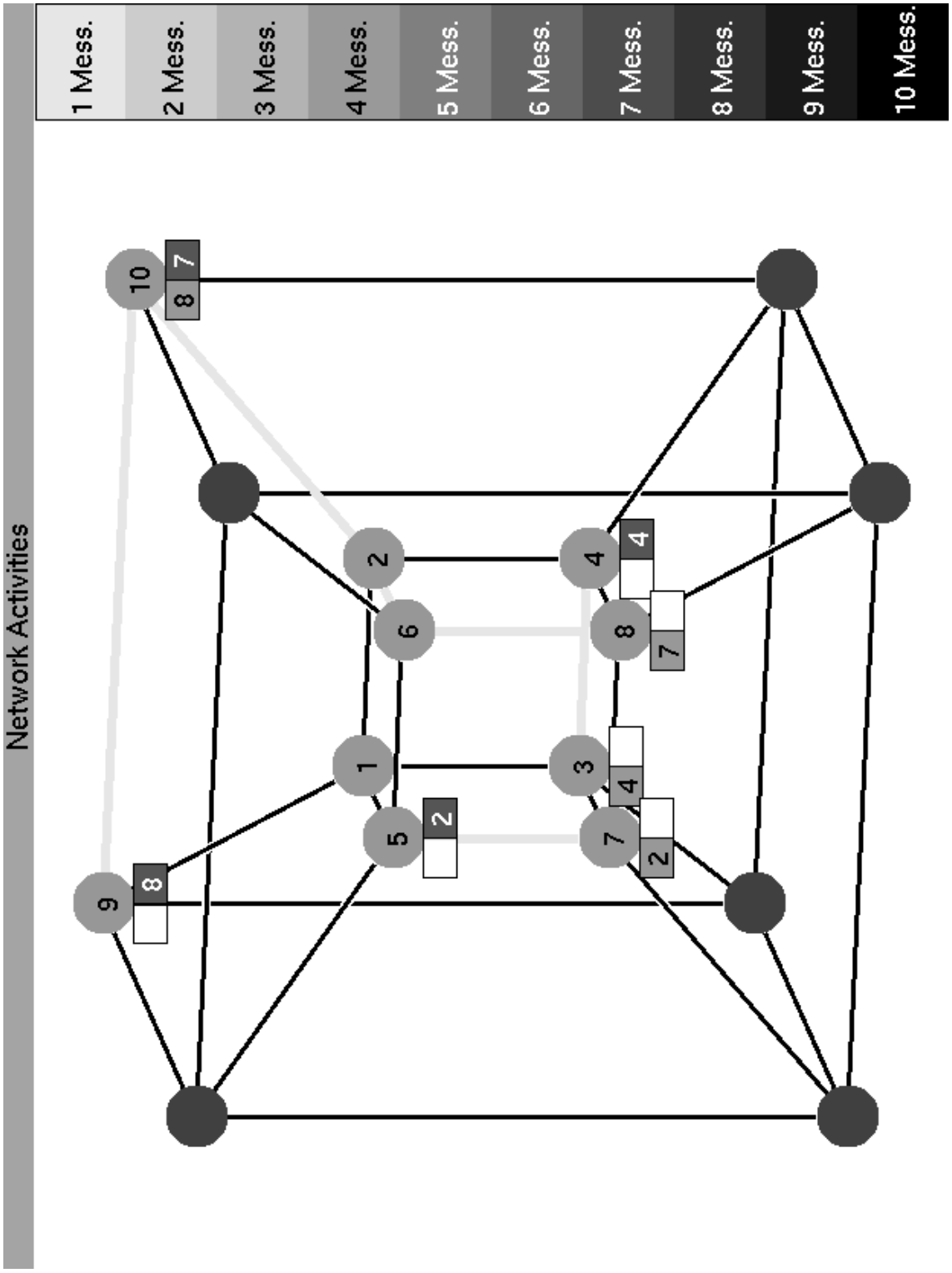


Abbildung 6.13: Prozessornetz-Darstellung am Beispiel eines 4-dim. Hypercube

scheint ein Dialogfenster, mit dem die Darstellung des Netzwerks auf vielfältige Weise beeinflußt werden kann. Dieses Fenster kann jedoch separat geschlossen werden, wenn es nicht mehr benötigt wird.

Die Hauptelemente des Grafikfensters stellen in verschiedener Form Informationen bereit, die in den folgenden Abschnitten erläutert werden.

6.7.1 Die Visualisierung der Netzverbindungen

Eine Verbindungslinie in der Darstellung des Prozessornetzes dient dazu, die Belastung der durch sie repräsentierten Netzverbindung zu verdeutlichen. Eine unbenutzte Verbindung wird als dünne schwarze Linie dargestellt. Wird eine Verbindung zur Übertragung von einer oder mehreren Speicherseiten (Nachrichten) genutzt, erscheint sie als dicke, farbige Linie. Die Linienfarbe ist dem globalen Farbspektrum des PARvis-Systems entnommen und repräsentiert die Anzahl der übertragenen Seiten (Nachrichten), so daß eine geringe Belastung durch einen eher blauen und hohe Belastung durch einen eher roten Farbton dargestellt wird ("kalte" bzw. "heiße" Verbindung). Die genaue Zuordnung kann anhand einer am rechten Fensterrand angeordneten Legende abgelesen werden.

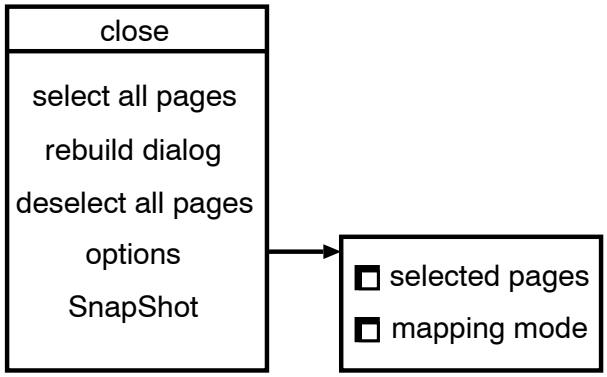


Abbildung 6.14: Popup-Menü der Prozessornetz-Darstellung

6.7.2 Die Visualisierung der Prozessorknoten

Die Kreise, die die Knoten des Prozessornetzwerkes repräsentieren, enthalten die Nummern der ihnen zugeordneten Prozessoren aus dem darzustellenden Systemlauf. Gibt es mehr Knoten in der angezeigten Topologie als CPUs im Systemlauf, werden überflüssige Kreise unbeschriftet belassen und in einer anderen Färbung wiedergegeben. Reichen die Knoten der Topologie nicht aus, alle Prozessoren des Laufs zu repräsentieren, wird der entsprechende Anteil der Prozessoren in der Darstellung ignoriert. Die Farbwerte für benutzte und unbenutzte Knoten können im globalen Konfigurationsdialog für Speicherdarstellungen angegeben werden.

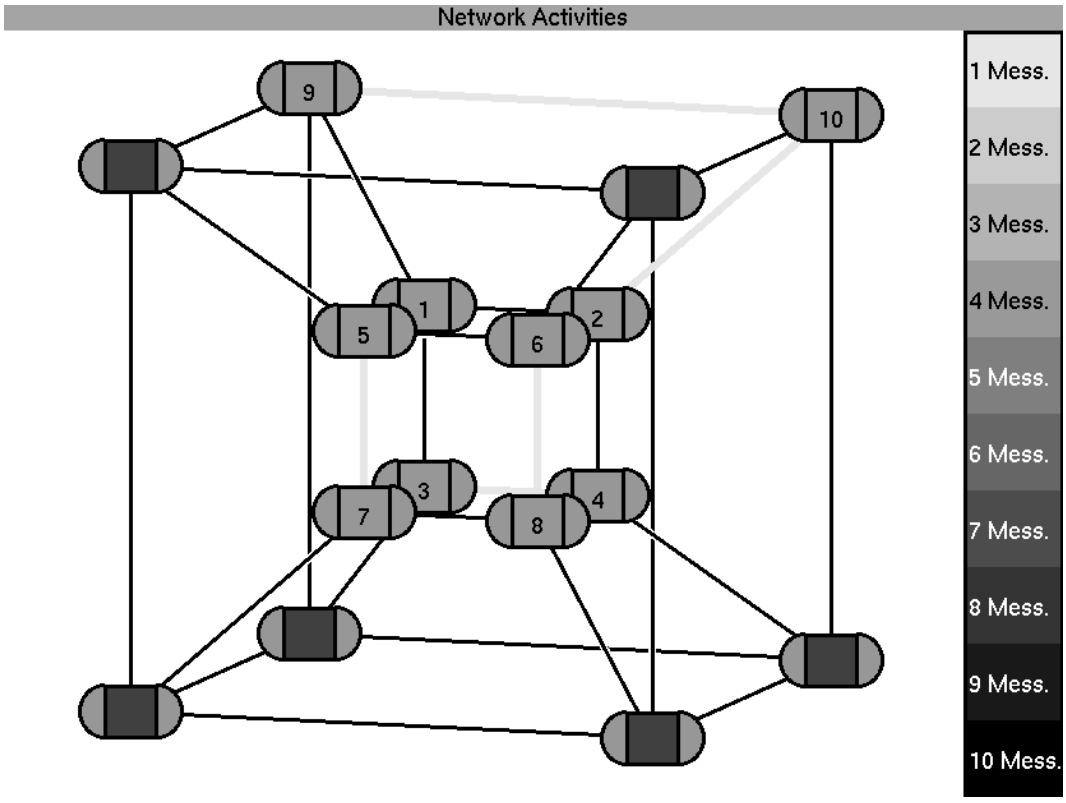


Abbildung 6.15: Prozessornetz-Darstellung im Zuordnungsmodus

6.7.2.1 Knotenzuordnung und deren Anpassung

Die Zuordnung von CPUs des Systemlaufs zu Knoten der Darstellung erfolgt grundsätzlich so, daß die CPUs in der Reihenfolge ihrer Numerierung den Knoten in der Reihenfolge ihrer internen Repräsentation zugewiesen werden. Jedoch kann diese vom System vorgegebene Zuordnung nach Betätigung des Popup-Menü-Auswahlpunktes **options/mapping mode** (Abbildung 6.14) manuell verändert werden. Durch diesen Eingriff wechselt die Anzeige in einen Modus, in dem die Knoten ein anderes Aussehen erhalten (Abbildung 6.15). Durch Anklicken der links und rechts des Knotenrechtecks angeordneten Halbkreise kann dann die Nummer der zugeordneten CPU zyklisch vermindert bzw. erhöht werden. Die der Prozessornummer Null entsprechende Einstellung symbolisiert, daß der Knoten keiner CPU des Laufs zugeordnet ist und wird durch ein unbeschriftetes Knotenrechteck dargestellt. Durch erneutes Betätigen des Umschaltknopfes **options/mapping mode** wird wieder in den normalen Anzeigemodus umgeschaltet. Dies funktioniert jedoch nur bei korrekter Zuordnung, d.h. wenn in der neuen, manuell eingestellten Zuordnung keine CPU-Nummer des Laufs mehr als einem Knoten der Netzdarstellung zugeordnet ist. Danach werden die eingebrachten Zuordnungsänderungen bei der Darstellung der Knoten- und Verbindungsinformationen berücksichtigt.

6.7.2.2 Darstellung gesendeter und empfangener Seiten

Durch Anklicken einer Knotendarstellung, die einer CPU des Systemlaufs zugeordnet ist, erscheint darunter ein kleines Doppelfenster, in dem die momentan gesendete Seite links und die momentan empfangene Seite rechts angezeigt werden (Abbildung 6.13, Knoten 10). Dabei werden zwei verschiedene Farben verwendet, die im globalen Konfigurationsdialog für Speicherdarstellungen ausgewählt werden können. Auf diese Weise können Sende- und Empfangsaktivitäten ausgewählter *Prozessoren* verfolgt werden.

Durch Betätigen des Umschaltknopfes **options/selected pages** im Popup-Menü kann man in einen Modus übergehen, in dem alle *ausgewählten* Seiten, sobald sie Gegenstand von Kommunikationsaktivitäten sind, in der oben beschriebenen Weise dargestellt werden. Die Auswahl kann neben der Möglichkeit, die die Seitenübersicht bietet, auch direkt durch Anklicken der hier (s.o.) dargestellten Seitenrechtecke erfolgen. Wird z.B. die Seite a von CPU A nach CPU B gesendet, geht im selben Moment je ein kleines Doppelfenster bei den Knoten A und B auf. Sie enthalten dann die Seite a an den entsprechenden Stellen. Auf diese Weise können Sende- und Empfangsaktivitäten ausgewählter *Seiten* verfolgt werden. Der Anwender kann mit der Menü-Option **select all** alle Speicherseiten auswählen, was zur Folge hat, daß *sämtliche* Kommunikationsaktivitäten beobachtet werden können.

6.7.3 Auswahl der Topologie und weitere Einstellmöglichkeiten

Das Dialogfenster **Network Control** (Abbildung 6.16) enthält eine Liste mit dem Namen **topology**. Sie enthält verschiedene Netzwerk-Topologien, aus denen eine gewünschte durch einfaches Anklicken ausgewählt werden kann, worauf das entsprechende Netz sofort im Grafikfenster dargestellt wird. Folgende Topologien sind in der hier beschriebenen Ausbaustufe implementiert:

- Vierdimensionaler Hypercube mit einer festen Anzahl von sechzehn Knoten
- ein- bis dreidimensionale Gitter
- ein- bis dreidimensionaler Torus, der ein- bis dreifach zyklisch sein kann

Falls ein Gitter oder ein Torus ausgewählt werden soll, müssen vorher drei Dimensionsangaben in den darunterliegenden Texteingabefenstern gemacht werden. Mit Hilfe dieser Angaben wird auch die Dimensionalität des Netzes festgelegt. Wird eine Dimension mit einer Eins besetzt, erhält man ein zweidimensionales (ebenes) Gitter (einen ebenen Torus), eine doppelte Eins-Angabe führt zur Darstellung einer Prozessorkette (eines Prozessorringes). Die gegenüber einem Gitter zusätzlichen, zyklischen Verbindungen eines Torus werden in der Darstellung als gerade weiterführende

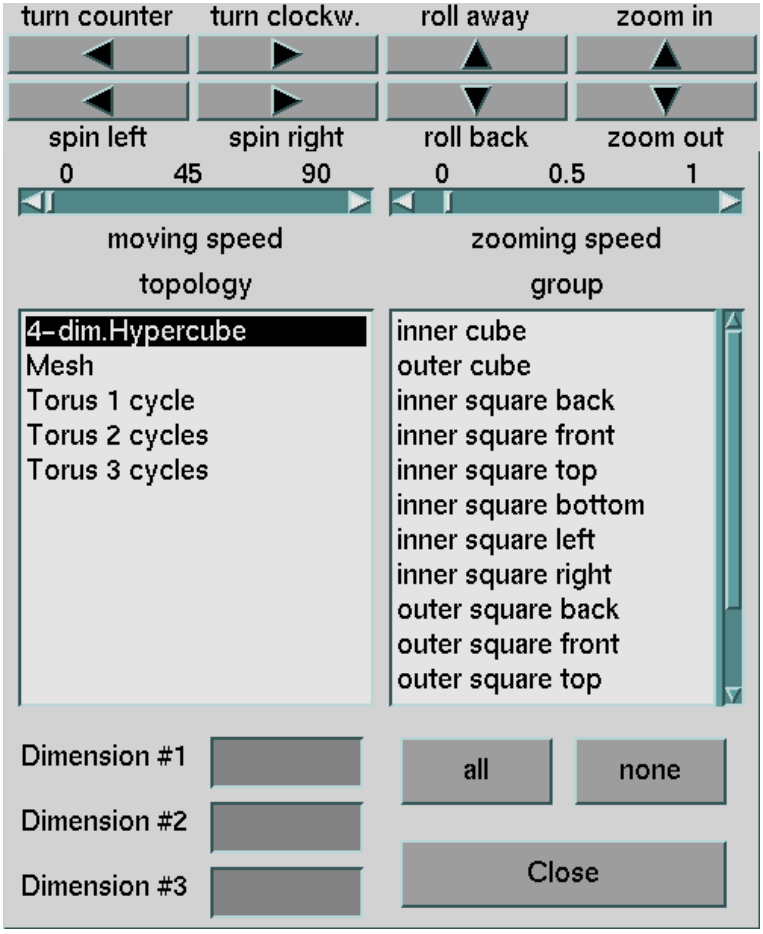


Abbildung 6.16: Dialogfenster für die Prozessornetz-Darstellung

Linien wiedergegeben, die zu einem Doppelgänger des Zielknotens (dargestellt als beschrifteter, farbiger Ring) führen (Abbildung 6.17).

Rechts neben der Liste der verfügbaren Topologien befindet sich eine weitere Liste im **Network Control**-Dialog, die nach Auswahl einer Netzstruktur eine Reihe von Teilnetzen enthält. Werden eines oder mehrere dieser Listenelemente angeklickt, wird die Netzdarstellung auf die gewählte Teiltopologie (-Kombination) reduziert. Zur leichteren Handhabung dieser Beschränkungsmöglichkeit sind unter der Liste Knöpfe zur sofortigen Auswahl aller bzw. keines der Teilnetze vorgesehen (*kein* Teilnetz bedeutet Darstellung des Gesamtnetzes).

Im oberen Teil des Dialogs finden sich Knöpfe, mit denen die als Parallelprojektion dargestellte, dreidimensionale Struktur um die drei Raumachsen gedreht werden kann. Dadurch sind unterschiedliche Perspektiven auf das Prozessornetz möglich, bei denen verschiedene Kommunikationsmuster unterschiedlich deutlich zutage treten können. Zwei Zooming-Knöpfe zum Vergrößern bzw. Verkleinern der Darstellung sind ebenfalls vorgesehen. Die darunter liegenden Schieberegler dienen zur

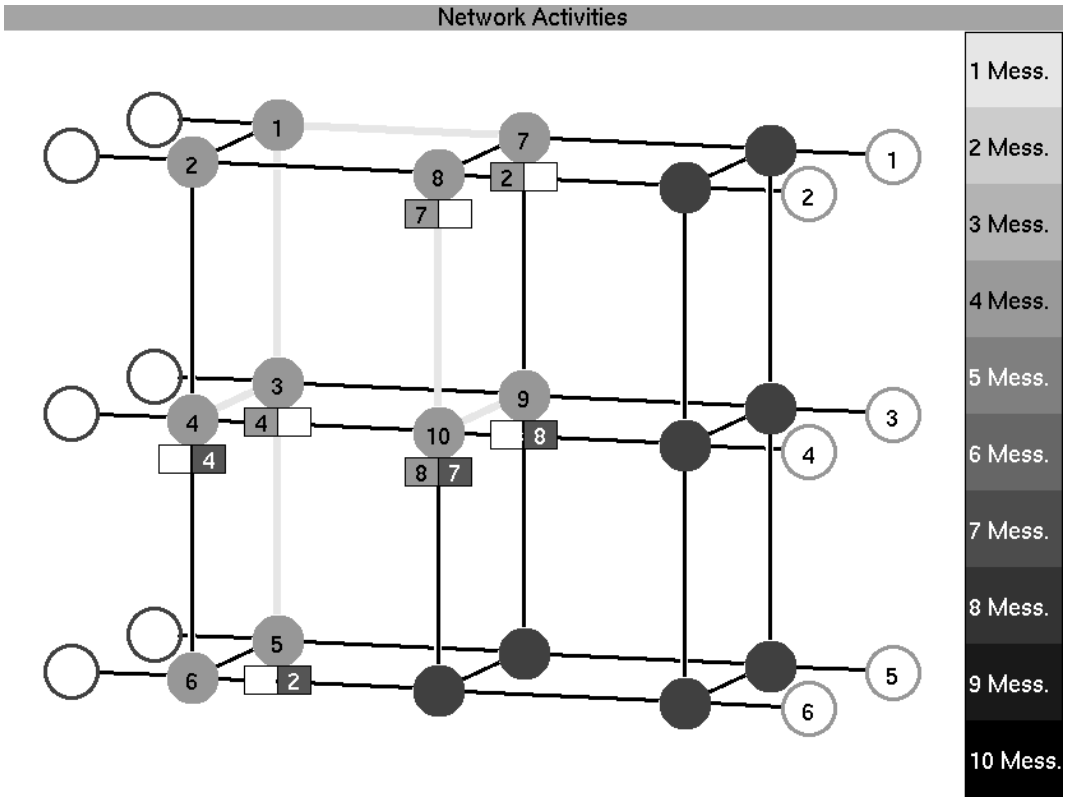


Abbildung 6.17: Darstellung eines 3x3x2-einfach zyklischen Torus

Einstellung der Schrittweite bei Dreh- und Zoomingvorgängen. Der **Close**-Knopf schließt diesen Dialog, den man dann mit Hilfe des Popup-Menüs des Grafikfensters unter dem Auswahlpunkt **rebuild dialog** wieder eröffnen kann.

In den Bildern 6.13 und 6.17 sind die Wege der nun schon aus den anderen Werkzeugen bekannten Kommunikationsvorgänge des selben Zeitpunktes deutlich abzulesen.

6.8 Einstell- und Filtermöglichkeiten für Speicherdarstellungen

Einige Eigenschaften, die für mehrere Speicherdarstellungen gelten sollen, können in einem speziellen Dialogfenster (Abbildung 6.18) eingestellt werden. Hier finden sich ebenfalls Konfigurationsmöglichkeiten für Parameter, die sich nur in jeweils einem Visualisierungsfenster auswirken, jedoch in der Regel nur selten verändert werden und mehr zur Definition der grundsätzlichen Arbeitsumgebung gehören. Der Dialog ist vom PARvis-Hauptmenü aus unter **Filter/page aspects** zu erreichen.

Der obere Teil des Dialogfensters ist nach dem Vorbild des in [1, Abschnitt 5.1.8] beschriebenen Dialogs zum Ausblenden von Prozessoren gestaltet worden und funk-

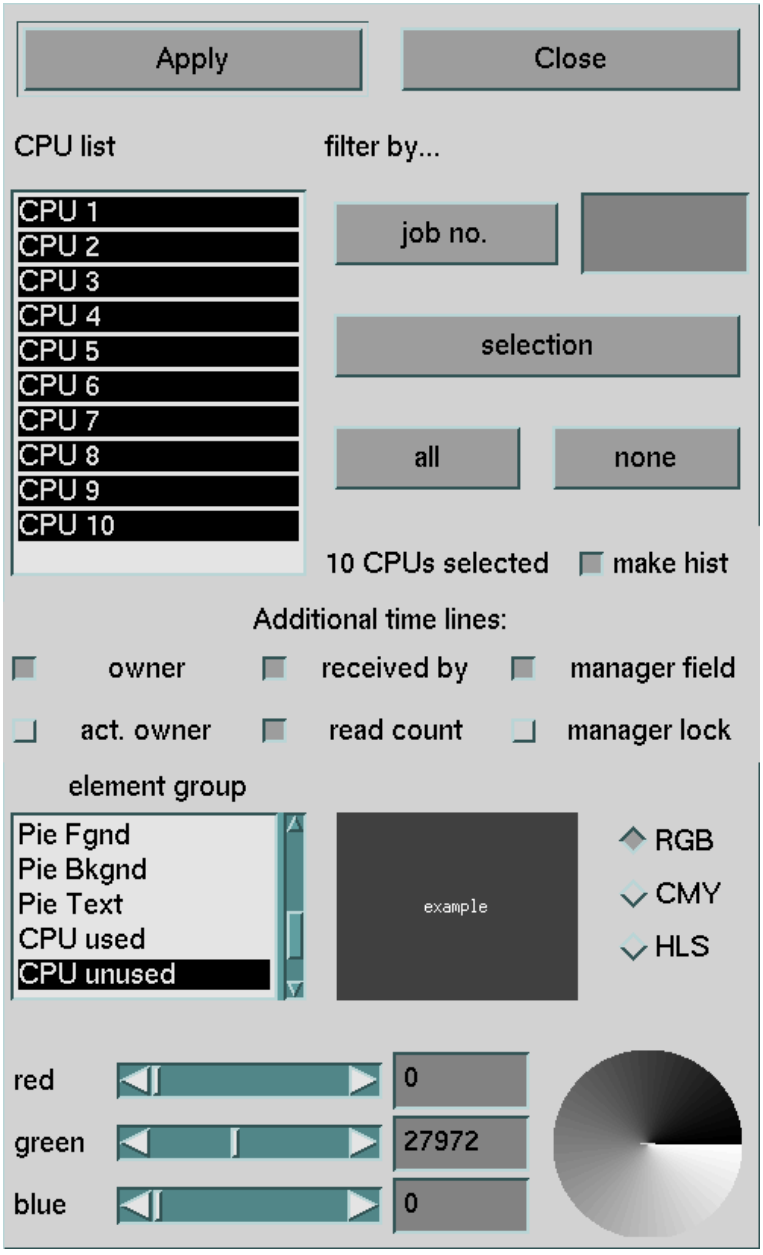


Abbildung 6.18: Konfigurationsdialog für Speicherdarstellungen

tioniert in derselben Weise. Die hier getroffenen Selektionen beziehen sich jedoch auf die Darstellungen zur Visualisierung der Speicheraktivitäten.

Der Umschaltknopf **make hist.** ermöglicht es dem Anwender, der keine Zeitlinien für Speicheraktivitäten oder Aussagen über den Kommunikationszeitanteil von Seiten benötigt, auf die Erzeugung einer Historienliste für Speicheraktivitäten zu verzichten und dadurch den Speicherbedarf des Systems zu vermindern.

Im mittleren Bereich des Dialogfensters befinden sich unter der Rubrik **Additio-**

nal time lines Umschaltknöpfe, mit denen sich bestimmte Aktivitäten auswählen lassen, für die Aktivitäts-orientierte Zeitlinien in den unter 5.2.4 und unter 6.3 beschriebenen Werkzeugkomponenten erzeugt werden sollen.

Im unteren Teil des Fensters schließlich findet sich ein Dialog zur Justierung von Farben für bestimmte Darstellungselemente; er entspricht den unter [1, Abschnitt 5.3.3] beschriebenen Dialogen in Aussehen und Funktion. Die Darstellungselemente, deren Farben hiermit ausgewählt werden können, sind in der Liste am linken Fensterrand aufgeführt. In ihr sind folgende Elemente enthalten:

- alle Aktivitäten, die Prozessoren nach dem hier zugrundeliegenden VSM-Modell auf Speicherseiten ausüben können. In allen Darstellungsfenstern, die Aktivitäten durch Symbole repräsentieren, werden die hier eingestellten Farben benutzt.
- die Komponenten der Tortendiagramme, die in der Seitenübersicht den Anteil der Kommunikationszeit der einzelnen Seiten veranschaulichen.
- die Knotensymbole des Visualisierungsfensters der Netzwerkkommunikation in den Ausprägungen "benutzt" und "unbenutzt".
- die Darstellungsrechtecke für die gesendete und die empfangene Seite in demselben Werkzeug.

Erst nach Drücken des **Apply**-Knopfes werden alle Einstellungen des **page aspects**-Fensters wirksam. **Close** schließt das Fenster. Alle Einstellungen in diesem Dialogfenster werden beim Verlassen der Visualisierungsumgebung in die PARvis-Konfigurationsdatei übernommen.

Die in diesem Kapitel beschriebenen Funktionen und Bedienungselemente sind in dem Bemühen entstanden, die prinzipiellen Möglichkeiten der einzelnen Darstellungskonzepte weitgehend auszuschöpfen. Dabei wurde darauf geachtet, auch Mechanismen vorzusehen, die verschiedene Werkzeugkomponenten funktional miteinander verknüpfen, damit die einzelnen Komponenten auch im Zusammenspiel optimal genutzt werden können.

Kapitel 7

Zusammenfassung und Ausblick

In dieser Arbeit wurden Werkzeugkomponenten vorgestellt, die dazu dienen, Vorgänge in Virtual-Shared-Memory-Systemen auf verschiedene Arten graphisch darzustellen. Diese Komponenten müssen unterschiedliche Perspektiven bezüglich der betrachteten Systeme erlauben und vielfältige Möglichkeiten zur Präzisierung der dargestellten Informationen bereitstellen. Mit dem Wunsch nach detaillierter Visualisierung war die Forderung verbunden, die Veranschaulichungen bezüglich der dargestellten Komponenten (Prozessoren, Speicherseiten) zu *filtern*. Als Ergebnis der Entwicklungsarbeiten entstanden sechs Komponenten, von denen jede einen anderen Blickwinkel auf die Speicheraktivitäten eines VSM-Systems bietet. Ausgangspunkt war das System PARvis, das zunächst eine Visualisierung von Prozeßwechselaktivitäten bot.

Zeitliniendarstellungen erlauben Einblicke in die Prozessoraktivitäten auf bestimmten Speicherseiten, wobei sowohl CPU-orientierte als auch Aktivitäts-orientierte Zeitlinien angezeigt werden können. Eine weitere Form Zeitlinien-orientierter Darstellung ermöglicht in besonderer Weise die Beobachtung der seitenbezogenen Kommunikationsaktivitäten der Prozessoren, welche direkten Aufschluß über Problemstellen und Engpässe geben, die z.B. auf mangelnder Datenlokalität beruhen. Der betrachtete zeitliche Darstellungsausschnitt kann in beiden Fällen gemäß der PARvis-Philosophie auf vielfältige Weise durch Zooming- und Verschiebefunktionen den Wünschen des Anwenders angepaßt werden. Die Möglichkeit der Kopplung beider Zeitliniendarstellungen bezüglich Maßstab und Ausschnitt eröffnet viele Bereiche des gleichzeitigen Einsatzes.

Drei *Zeitpunktdarstellungen* geben Auskunft über die Seitenaktivitäten der Prozessoren nach einer definierten Anzahl von Maschinenzyklen, wobei jeweils mehrere Seiten und CPUs in einem Bild berücksichtigt werden können. Die Visualisierungen erfolgen dabei *Seiten*-orientiert (Seitenübersicht, auch zur Auswahl bestimmter Seiten), *Prozessor*-orientiert (CPU-Seiten-Darstellung) oder in einer nach *Seiten* und *Prozessoren gleichermaßen* sortierten, matrixartigen Aktivitäts-Tabelle. Eine besondere Form der Zeitpunktdarstellung bildet die Visualisierung der Prozessorkommunikation im Netz, da hier zusätzliche Parameter (Topologie) manuell vorgegeben

werden können und dabei vielfältige Möglichkeiten der Zuordnung und Darstellung zur Verfügung stehen.

Für alle Werkzeuge gibt es Funktionen, mit denen die Darstellungen auf bestimmte Prozessoren und/oder Speicherseiten beschränkt werden können. Die Verwendung der Fenstertechnik erlaubt die gleichzeitige und verzahnte Benutzung verschiedener Darstellungshilfsmittel des PARvis-Systems. So wurde das Visualisierungssystem PARvis in einer Weise um Hilfsmittel zur Veranschaulichung von Speicheraktivitäten erweitert, daß Grundkonzepte und Bedienungsphilosophie erhalten blieben. Damit hat der Anwender ein einheitliches Werkzeug vor sich, mit dem er sich ein komplettes Bild des Untersuchungsobjektes Parallelcomputer machen kann.

Die fortschreitende Entwicklung paralleler Rechner führt zu immer höheren Prozessorzahlen und Speichergrößen. Damit wird sich früher oder später die Forderung nach Visualisierungswerkzeugen ergeben, die diesem Prozeß Rechnung tragen. Das heißt, daß Hilfsmittel für eine mehr globale Betrachtung der dynamischen Vorgänge in Mehrprozessorsystemen entwickelt werden müssen, in denen auch Methoden der Stochastik berücksichtigt werden. Diese Hilfsmittel werden bei der Anwendung dann den hier beschriebenen Werkzeugen vorgeschaltet und dienen der ersten, ungefähren Problemstellen-Lokalisierung. Die in [1] beschriebenen Statistik-Darstellungen sowie die in Abschnitt 5.2.1.2 vorgestellte Visualisierung des zeitlichen Kommunikationsanteils von Speicherseiten sind erste Schritte in diese Richtung.

Ein zweiter Aspekt bei zukünftigen Entwicklungen im Bereich der Visualisierung der Vorgänge in Mehrprozessorrechnern ist die Bereitstellung der Werkzeuge für den Anwendungsprogrammierer. Visualisierungen des Programmablaufs und der Prozeßwechsel müssen dann die entsprechenden Elemente des Programm-Quelltextes mit einbeziehen, damit der Kontrollfluß anhand der Befehlsfolge der verwendeten Programmiersprache sichtbar wird. Dies macht eine geeignete Erweiterung der Protokolldaten erforderlich.

Für Speicherdarstellungen folgt aus dieser Forderung, daß die Verbindung zur Ebene der symbolischen Datenbezeichnungen (aus dem Quellprogramm) langfristig hergestellt werden muß, damit der Programmierer die Wege der Variablen durch das System und die auf sie bezogenen Aktivitäten verfolgen kann. Dafür werden zunächst die in den Protokolldateien enthaltenen Informationen um Angaben darüber erweitert werden müssen, welche Speicherseiten welchen Datenfeldern der Anwendung zugeordnet sind. Zur Bereitstellung dieser Informationen müssen geeignete Monitore oder Instrumentierungswerkzeuge entwickelt werden, die entsprechenden Zugriff auf Compiler-Symboltabellen besitzen. Auf diese Weise kann man Werkzeuge wie PARvis aussagekräftiger und einem gößeren Anwenderkreis nutzbar machen und damit ihre Chancen erhöhen, zum Verständnis und zur effektiveren Nutzung paralleler Rechner beizutragen.

Literaturverzeichnis

[1] A. Arnold, *PARvis: Eine X-basierte Umgebung zur Visualisierung von parallelen Programmen in Multiprozessorsystemen*
Jül 2848, Forschungszentrum Jülich, Dezember 1993 (Diplomarbeit).

[2] R. Berrendorf, *Memory access in shared virtual memory*
CONPAR 92-VAPP V, Proceedings of the Second Joint International Conference on Vector and Parallel Processing, Lyon, Frankreich, September 1992, Lecture Notes in Computer Science, Springer Verlag, 785-786.

[3] A. Bode, P. Braun, *Monitoring and visualization in TOPSYS*
Performance Measurement and Visualization of Parallel Systems, Proc. Workshop on Performance Measurement and Visualization, Moravany, Tschechoslowakei, Oktober 1992, Elsevier Science Publishers B. V., 97-118.

[4] O. Brewer, J. Dongarra, D. Sorensen, *Tools to aid in the analysis of memory access patterns for FORTRAN programs*
Parallel Computing 9 (88/89), 1988, 25-35.

[5] J. M. Francioni, L. Albright, J. A. Jackson, *Debugging parallel programs using sound*
Proceedings of the ACM/ONR Workshop on Parallel and Distributed Debugging, Mai 1991, 68-75.

[6] M. T. Heath, J. E. Finger, *ParaGraph: A tool for visualizing performance of parallel programs*
ParaGraph User Guide, Oak Ridge National Laboratory, Dezember 1993.

[7] I. Ichikawa, S. Aikawa, M. Kamiko, E. Ono, T. Mohri, *Program design visualization system for object-oriented programs*
ACM Sigplan Notices 24 (4), 1989, 181-183.

[8] R. Klar, *Hardware-/Software-Monitoring*
Informatik Spektrum 8/1985, 37-40.

[9] V. Klinger, *Ein hybrider Computerbus-Monitor*
Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen, 7. ITG/GI-Fachtagung, Aachen, 1993, Springer-Verlag, 346-358.

[10] E. Leu, A. Schiper, *Execution replay: a mechanism for integrating a visualization tool with a symbolic debugger*
Parallel Processing: CONPAR 92-VAPP V, Proceedings of the Second Joint International Conference on Vector and Parallel Processing, Lyon, Frankreich, September 1992, Lecture Notes in Computer Science 634, Springer Verlag, 55-60.

[11] K. Li, *Shared virtual memory on loosely coupled multiprocessors*
PhD thesis, Yale University, Technical Report YALEU-RR-492, 1986.

[12] K. Li, P. Hudak, *Memory coherence in shared virtual memory systems*
ACM Trans. Comp. Syst. 7 (4), 1989, 321-359.

[13] M. Linn, *Untersuchungen zur Ablaufplanung bei Parallelrechnern mit virtuell gemeinsamem Speicher*
Dissertation an der RWTH-Aachen, zur Veröffentlichung vorgesehen.

[14] A. D. Malony, *Performance observability*
PhD thesis, University of Illinois, 1990, 214-230.

[15] K. Mevissen, *Untersuchungen zur Lokalität in technisch-wissenschaftlichen Programmen*
Jül 390, Forschungszentrum Jülich, März 1987 (Diplomarbeit).

[16] W. E. Nagel, *Ein verteiltes Scheduling-System für Mehrprozessorsysteme mit gemeinsamem Speicher: Untersuchungen zur Ablaufplanung von parallelen Programmen*
Jül 2850, Forschungszentrum Jülich, Dezember 1993 (Dissertation).

[17] W. E. Nagel, A. Arnold, *PARvis: Ein Werkzeug zur Visualisierung von parallelen Prozessen auf Mehrprozessorsystemen*
Proc. 7. ITG/GI Fachtagung MMB'93 (Kurzberichte und Werkzeugvorstellung), 1993, 178-187.

[18] W. E. Nagel, M. A. Linn, *Parallel programs and background load: efficiency studies with the PAR-Bench system*
Proc. 1991 International Conference on Supercomputing, 1991, 365-375.

[19] W. E. Nagel, M. A. Linn, *Benchmarking parallel programs in a multiprogramming environment: The PAR-Bench system*
Dongarra/Gentzsch, Eds: Advances in parallel computing Vol. 8: Computer benchmarks, Elsevier Science Publishers B.V., 1993, 303-322.

[20] C. M. Pancake, *Visualizing the behavior of parallel programs*
SUPERCOMPUTER, Nr. 5, September 1990, 31-37.

[21] C. M. Pancake, S. Utter, *Debugger visualizations for shared-memory multiprocessors*
High Performance Computing II, Elsevier Science Publishers B. V. 1991, 145-158.

[22] *Paragon application tools user's guide*
Intel Corporation, Oktober 1993.

[23] J. M. Stone, *A graphical representation of concurrent processes*
Proceedings of the ACM/ONR Workshop on Parallel and Distributed Debugging, Mai 1991, 226-235.

[24] B. Tourancheau, X. F. Vigouroux, M. van Riek, *The massively parallel monitoring system, a truly parallel approach to Parallel Monitoring*
Performance Measurement and Visualization of Parallel Systems, Proc. Workshop on Performance Measurement and Visualization, Moravany, Tschechoslowakei, Oktober 1992, Elsevier Science Publishers B. V., 1-17.

Dank

Die vorliegende Diplomarbeit ist am Lehrstuhl für Technische Informatik und Computerwissenschaften der Rheinisch-Westfälischen Technischen Hochschule (RWTH) Aachen entstanden. Dem Lehrstuhlinhaber Herrn Professor Dr. F. Hoßfeld danke ich herzlich dafür, daß ich die Arbeit in dem von ihm geleiteten Zentralinstitut für Angewandte Mathematik (ZAM) des Forschungszentrums Jülich (KFA) anfertigen konnte.

Herrn Dr. W. E. Nagel danke ich ganz besonders für die intensive Betreuung und die vielen fruchtbaren Anregungen. Für die zahlreichen helfenden Hinweise möchte ich auch Herrn Dr. P. Weidner herzlich danken. Herr A. Arnold hat mir viele nützliche Ratschläge in bezug auf die Programmerstellung gegeben, während Herr M. A. Linn die Anforderungen an die Visualisierungswerkzeuge formuliert und mir dabei wichtige Anregungen gegeben hat. Auch ihnen möchte ich herzlich danken.