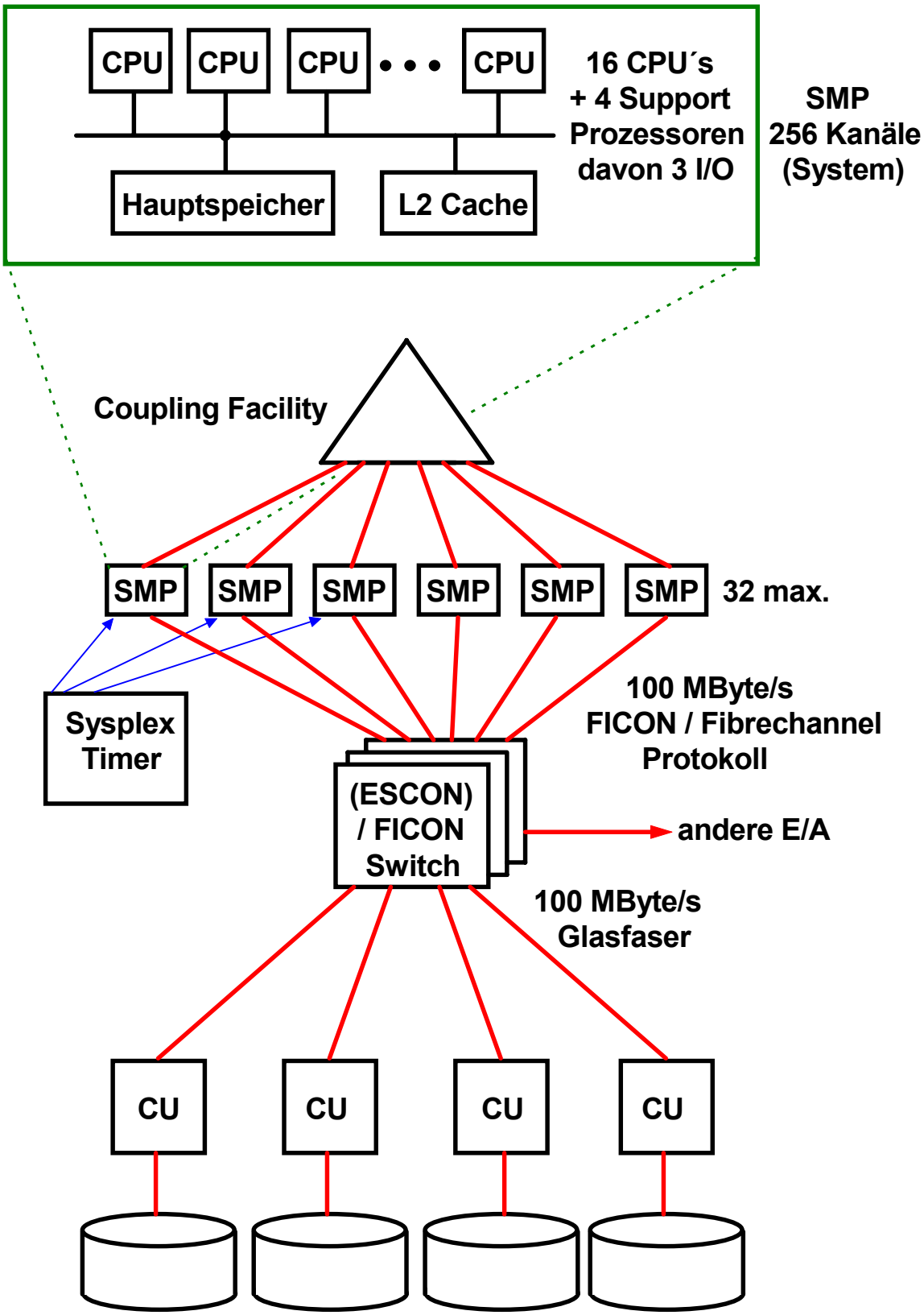
Internet Anwendungen unter OS/390

Dr. rer. nat. Paul Herrmannn
Prof. Dr.-Ing. Udo Kebschull
Prof. Dr.-Ing. Wilhelm G. Spruth

WS 2004/2005

Teil 5

Parallel Sysplex



Parallel Sysplex

Literatur

Wilhelm G. Spruth, Erhard Rahm:
Sysplex-Cluster Technologien für Hochleistungs-Datenbanken.
Datenbank-Spektrum, Heft 3, 2002, S. 16-26.

Verfügbar (download):
http://www-ti.informatik.uni-tuebingen.de/~spruth/publish.html

Sysplex Hardware:

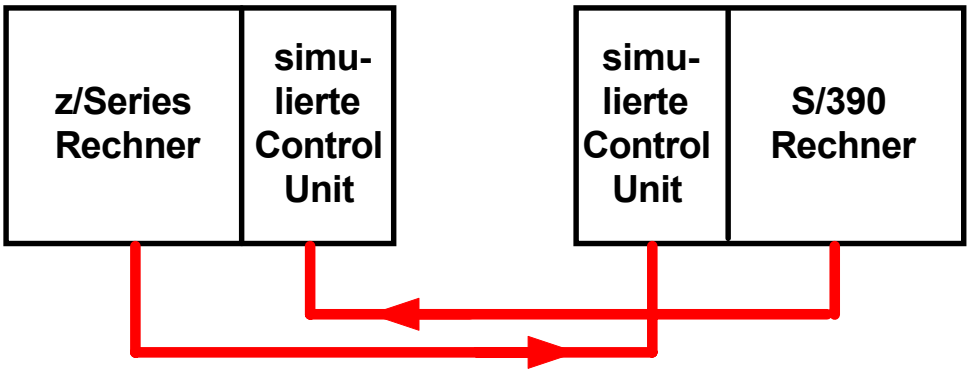
Sonderheft des IBM Journal of Research and Development, Vol. 36, No.4, July 1992.

Sysplex Software:

Sonderheft des IBM System Journal, Vol. 36, No.2, April 1997.

Verfügbar (download): //www.research.ibm.com/journal

CTC Verbindung
(Channel- to Channel)



Channel- to Channel Verbindung

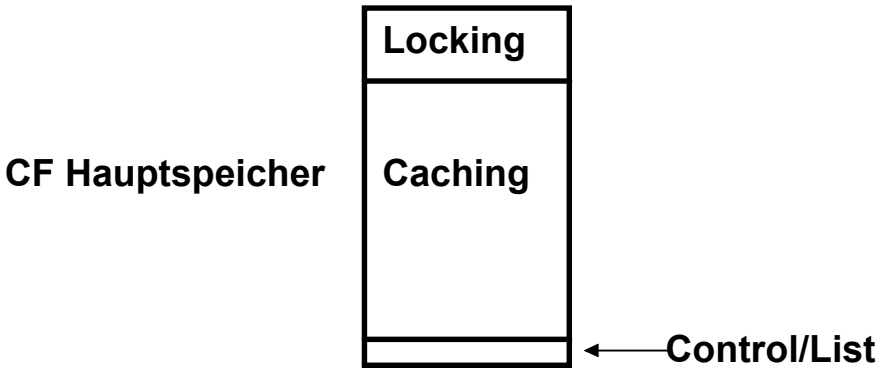
Cross-System Coupling Facility (XCF)

Die Cross-System Coupling Facility (XCF) verwendet das CTC Protokoll. Sie stellt die Coupling Services bereit, mit denen z/OS und OS/390 Systeme innerhalb eines Sysplex miteinander kommunizieren.

Coupling Facility (CF)

Die Coupling Facility ist hardwaremäßig ein S/390 Rechner mit einem eigenen minimalen Betriebssystem. Für den Sysplex Cluster übernimmt sie die folgenden Aufgaben:

- **Locking**
- **Caching**
- **Control/List Structur Management**

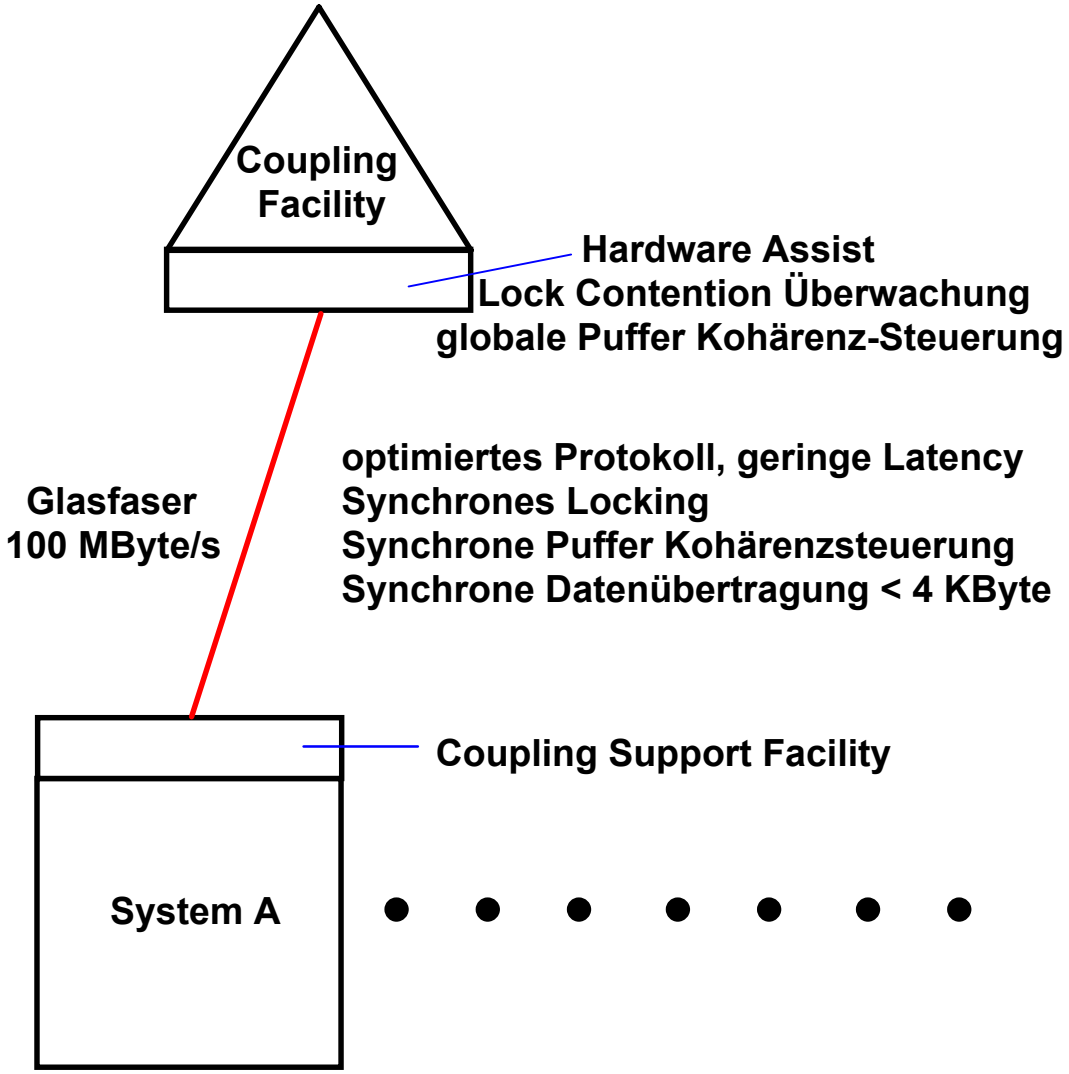


Die wichtigste Aufgabe der Coupling Facility ist ein zentrales Lock Management für die angeschlossenen Systeme. Der zentrale Lock Manager des SAP System R/3 hat in Ansätzen eine ähnliche Funktionalität.

Der größte Teil des Hauptspeichers der Coupling Facility wird als Plattenspeicher Cache genutzt. Der CF Cache dupliziert den Plattenspeicher Cache in den einzelnen Systemen. Cast out der CF Cache auf einen Plattenspeicher erfolgt über ein System.

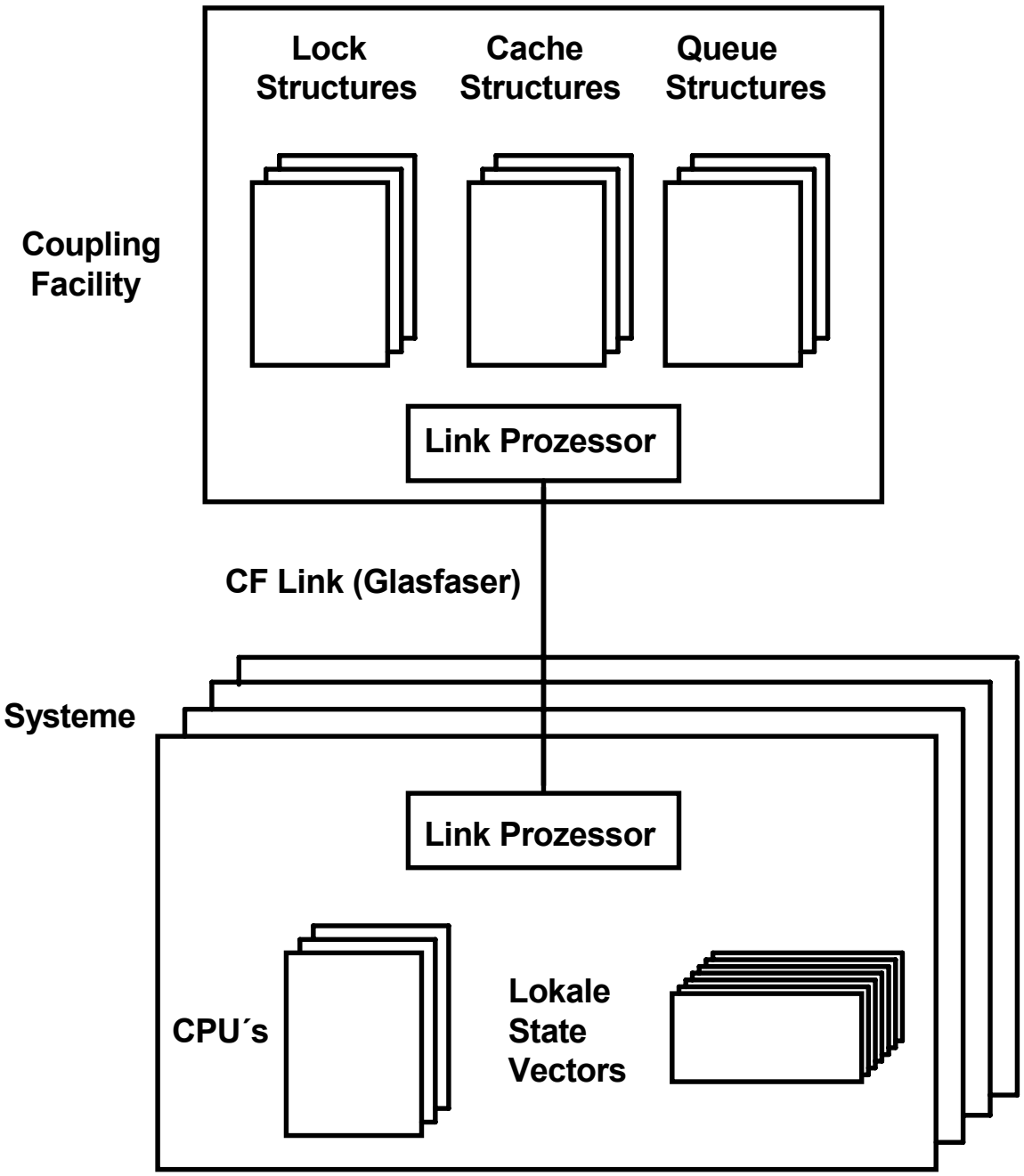
CF Cache Cross-Invalidate nur an die betroffenen Systeme

Control und List Strukturen dienen der Sysplex Cluster weiten Verwaltung. Beispiel: RACF Sicherheits Subsystem.



Anbindung eines Systems an die Coupling Facility

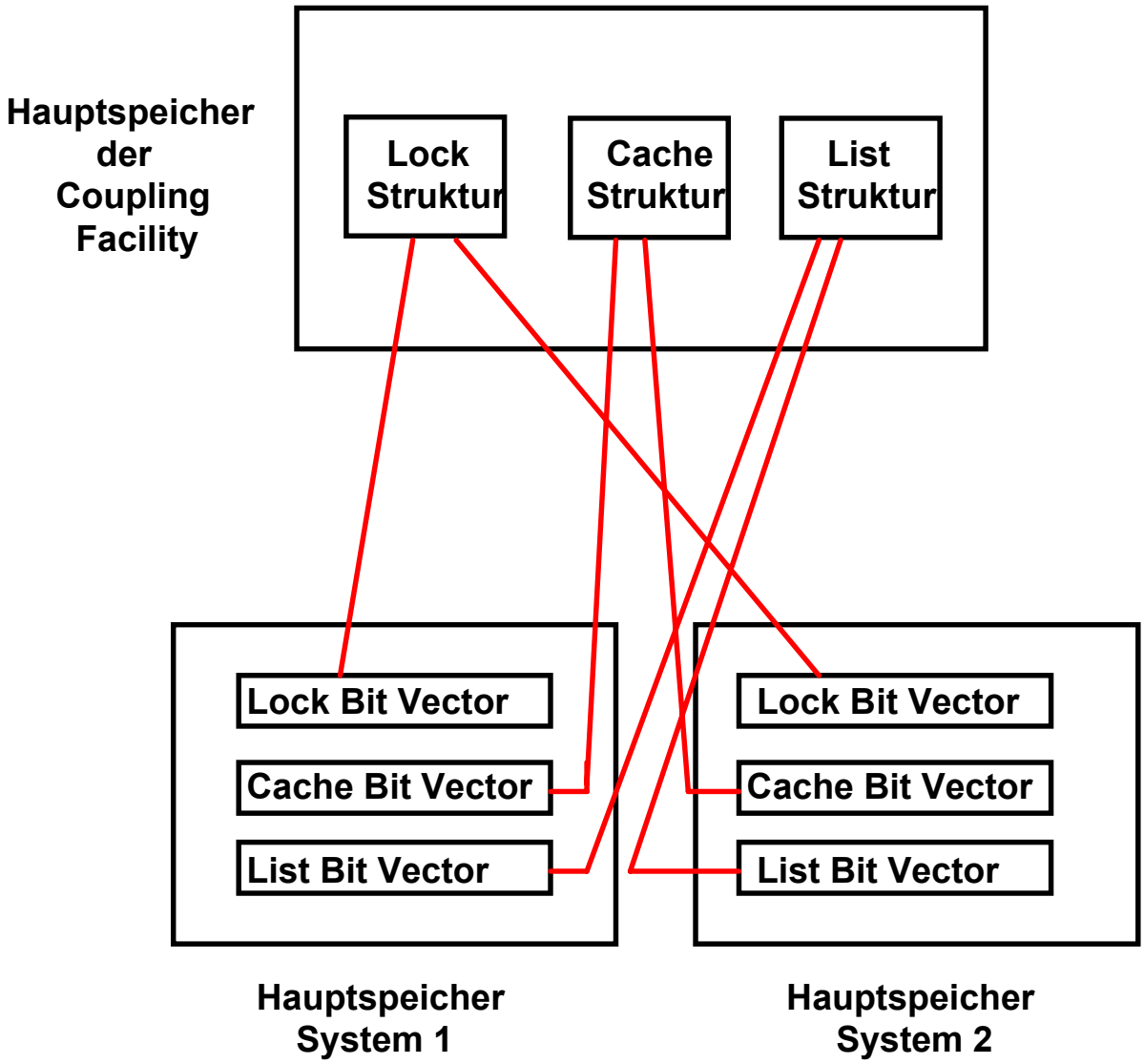
Die CF Glasfaser Verbindung wird durch spezielle Hardware Einrichtungen und durch zusätzliche Maschinenbefehle in jedem System unterstützt.



Mehrere Systeme sind mit der gleichen Struktur logisch verbunden

Je 1 lokaler State Vector für jede logische Verbindung zu einer Struktur

Spezifische Maschinenbefehle für die Kommunikation CPU - CF. Zusätzlicher Link Prozessor für die Kommunikation.



Zuordnung von Bit Vektoren zu CF Strukturen

Locking Problem

Anfangswerte: d1 = 15, d2 = 20

Transaktion 1	Transaktion 2
read d1	
if d1 > 10	
	read d1
	if d1 > 10
sub d1, 10	
	sub d1, 10
add d2, 10	
	add d2, 10

Ergebnis: d1 = - 5, d2 = 40

Benutzung von Locks (Sperren)

GetReadLock (d1)	
read d1	
if d1 > 10	
	GetReadLock (d1)
	read d1
	if d1 > 10
GetWriteLock (d1) →	*Nachricht an Transaktion 2*
GetWriteLock (d2)	
sub d1, 10	
add d2, 10	
ReleaseLocks	
	GetWriteLock (d1)
	GetWriteLock (d2)
	sub d1, 10
	add d2, 10

Ergebnis: d1 = + 5, d2 = 30

Two-Phase Locking
Two-Phase Transaktion

In Transaktionssystemen und Datenbanksystemen ist ein Lock ein Objekt welches (mindestens) über 4 Methoden verfügt:

• **GetReadLock**	**reserviert S Lock (shared)**	**SHR**
• **GetWriteLock**	**reserviert E Lock (exclusive)**	**EXC**
• **PromoteReadtoWrite**	**Wechsel S $\rightarrow$ E**	
• **Unlock**	**Lock freigeben**	

Mehrere Transaktionen können ein S Lock für das gleiche Objekt besitzen. Nur eine Transaktion kann ein E Lock für ein gegebenes Objekt besitzen. Wenn eine Transaktion ein S Lock in ein E Lock umwandelt, müssen alle anderen Besitzer des gleichen S Locks benachrichtigt werden.

Normalerweise besitzt eine Transaktion mehrere Locks.

In einer Two-Phase Transaktion finden alle Lock Aktionen zeitlich vor allen Unlock Aktionen statt. Eine Two-Phase Transaktion hat eine Wachstumsphase (growing), während der die Locks angefordert werden, und eine Schrumpf (shrink) Phase, in der die Locks wieder freigegeben werden.

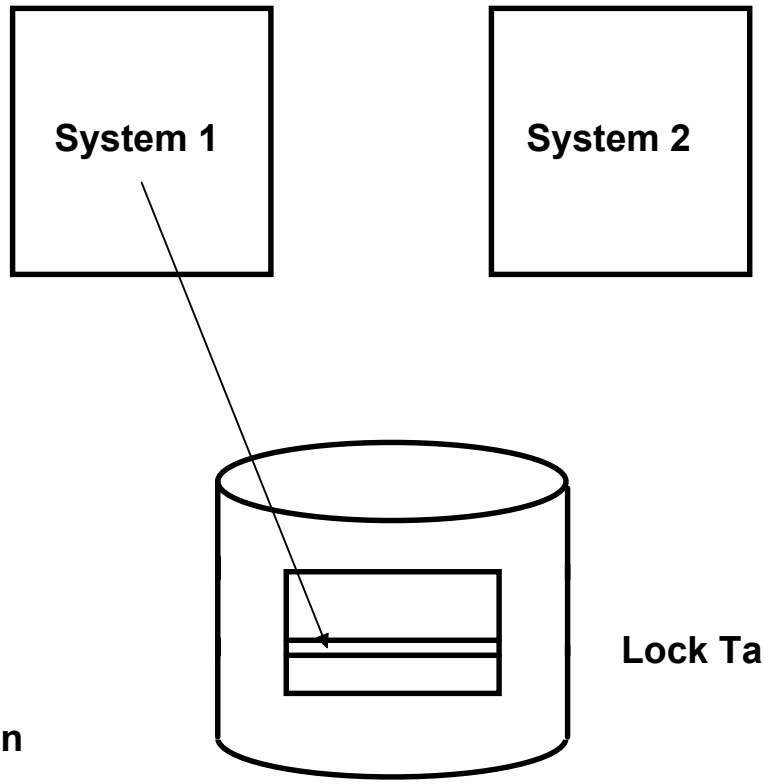
Nicht zu verwechseln mit dem 2-Pase Commit Protokoll der Transaktionsverarbeitung

Locking Protokoll

Share Lock (SHR) erwerben vor dem erstmaligen Lesen
Exclusive Lock (EXC) erwerben vor dem erstmaligen Schreiben

derzeitiger Status / Anforderung	kein	Lesen shared	Schreiben exclusive
Lesen Share	bewilligt, share-mode	bewilligt, share-mode	abgelehnt, Mitteilung über Besitzer
Schreiben Exclusive	bewilligt, exclusive mode	bewilligt, Warnung über Besitzer	abgelehnt, Mitteilung über Besitzer

Lock Verwaltung



1. Lock prüfen

2. Lock setzen

3. Datenzugriff

Im einfachsten Fall besteht die Lock Tabelle aus zusätzlichen Feldern in der Daten Tabelle

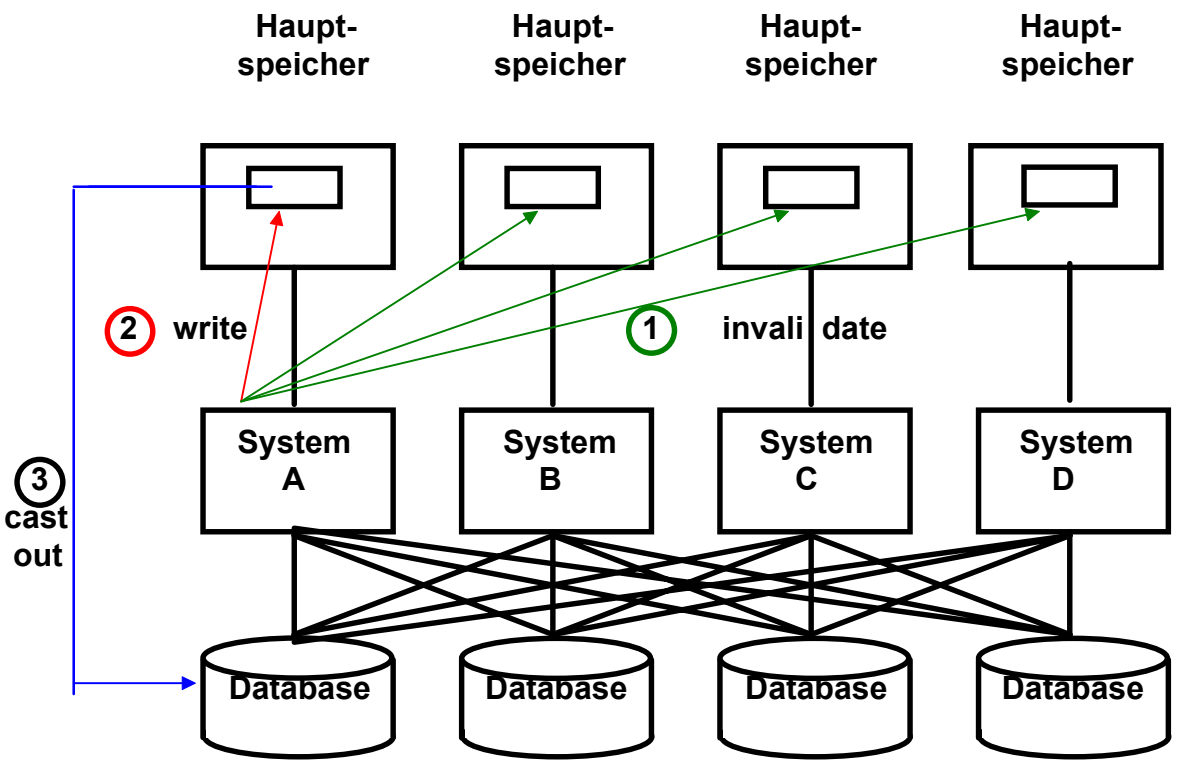
Shared Disk

Verteilte Lock Tabelle in den Hauptspeichern der beteiligten Systeme.

Zur Auflösung von Lock Konflikten Broadcast oder Nachricht von System i ⟶ System j.

Verarbeitung der laufenden Transaktion aussetzen; 20 ms Overhead.

Beispiele: VAX DBMS und VAX Rdb/VMS



Invalidate-Broadcast Kohärenzsteuerung

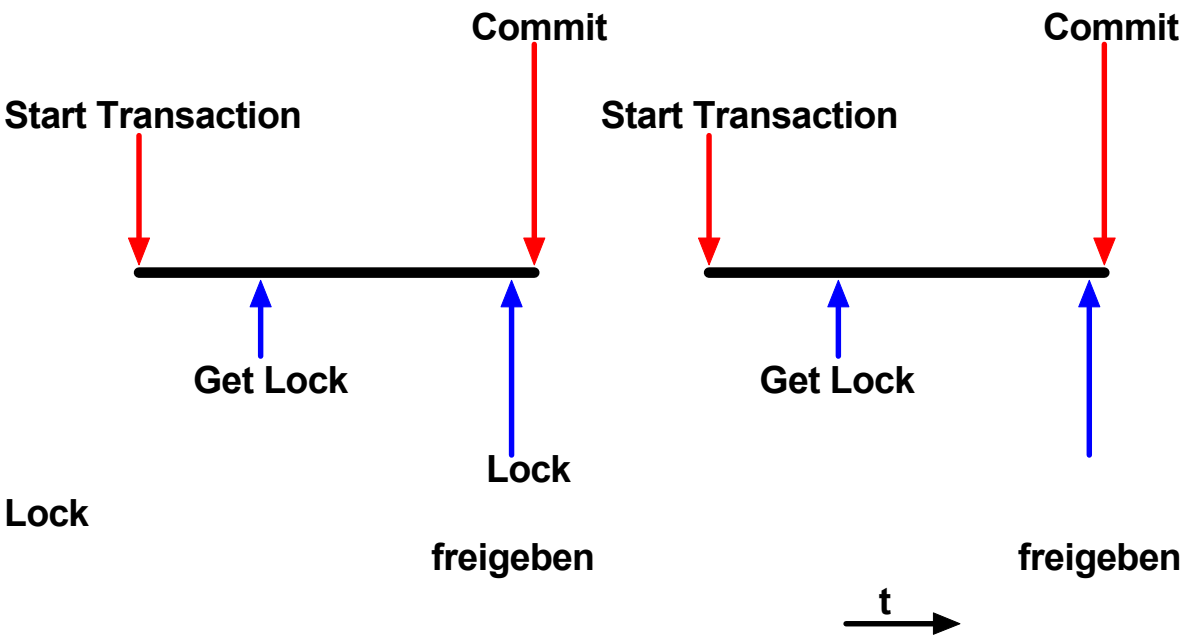
Nur System A besitzt Write Lock. B, C und D besitzen nur Read Lock. Invalidate Broadcast benachrichtigt B, C und D daß Kopie nicht mehr gültig.

Variante: **System A behält Lock für Kopie der Pufferdaten über Transaktionsgrenze hinweg, kein castout to Disk. Lock nur freigeben, wenn Contention von einem anderen System**

System B möchte schreiben, bittet A um Lock. A schreibt Buffer to Disk, B liest Buffer

Der Besitz des Locks transferiert (pings) bei Bedarf von einem System zu einem anderen.

Potentielles Ping-Pong Problem



„Eager“ und „Lazy“ Locking Protokolle

Eager Protokoll Lock freigeben wenn Commit Transaction

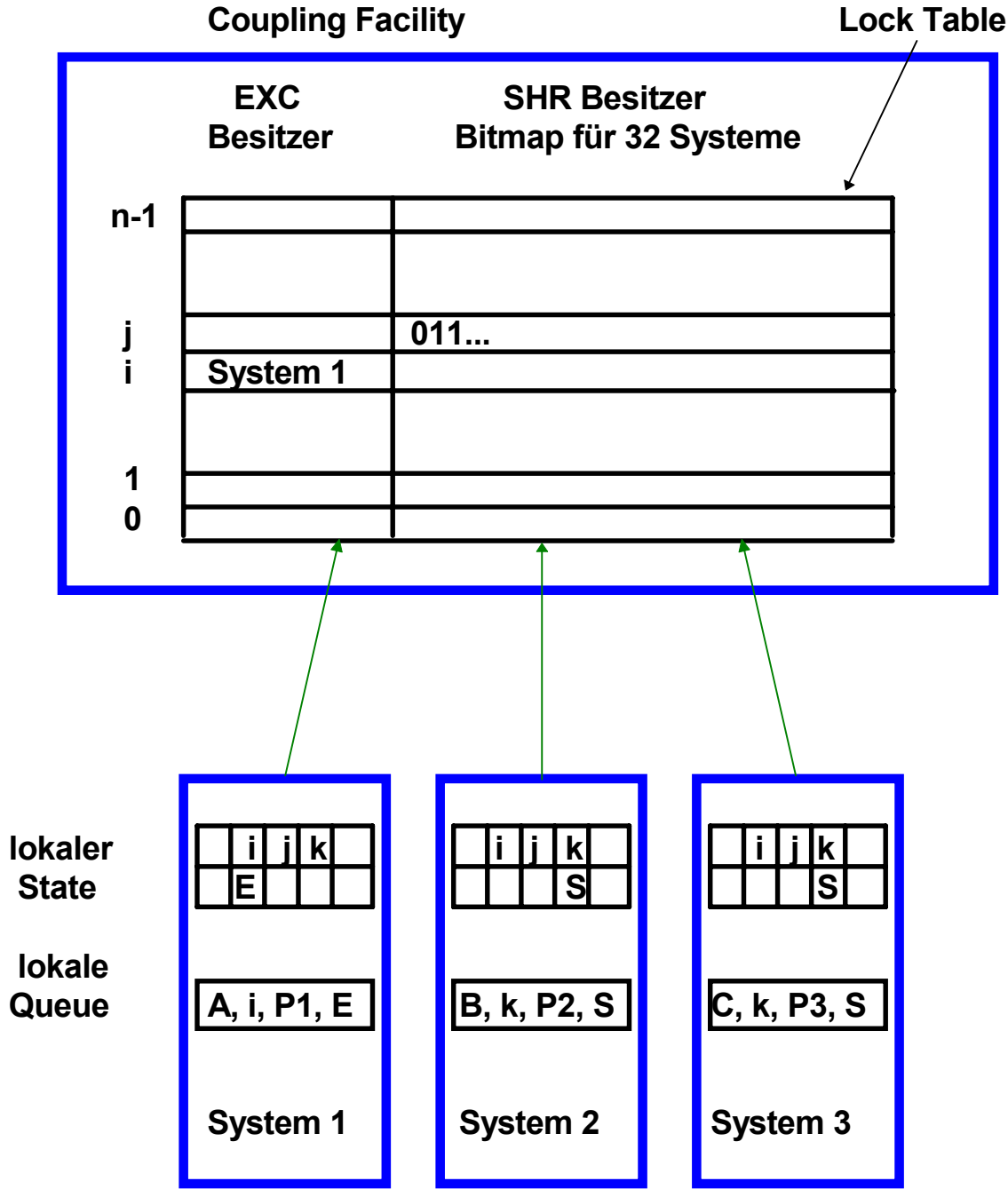
Lazy Protokoll Lock freigeben wenn Contention

Lazy Protokoll arbeitet besser, wenn Sharing selten auftritt.
Beispiel TPC-C

Sysplex Coupling Facility verwendet das Eager Protokoll (auch als „force-at-commit“ bezeichnet). Sharing tritt häufig auf, wenn existierende Anwendungen auf den Sysplex portiert werden.

Der Lock - Zustand eines Data Items kann 3 Werte haben:

Frei		0
Shared	SHR	S, = 1
Exclusive	EXC	E



Nutzung der CF Lock Tabelle

Lock Contention Steuerung

Der (symbolische) Name A eines Locks wird mit Hilfe eines Hashing Algorithmus in die Hash Klasse i abgebildet. Die Locking Tabelle enthält für jede Hash Klasse einen Eintrag.

Die Zuordnung Lock Name zu Hash Klasse erfolgt in der lokalen Queue des betreffenden Systems.

1. **Prozess P1 in System 1 möchte EXC Rechte für ein Lock in der Hash Klasse i erhalten. Anfrage an CF. Da niemand sonst Interesse hat, wird dem Request entsprochen. Im lokalen State Vektor von System 1 wird diese Berechtigung festgehalten.**

   In der lokalen Queue von System 1 wird festgehalten, daß Lock A, Hash Klasse i von dem lokalen Prozess P1 mit der Berechtigung Exclusive gehalten wird.

   Wenn Prozess P2 in System 1 ebenfalls Lock Rechte für i wünscht (möglicherweise für einen anderen Lock Namen), ist kein Zugriff auf die CF erforderlich. System 1 kann dies alleine aussortieren.

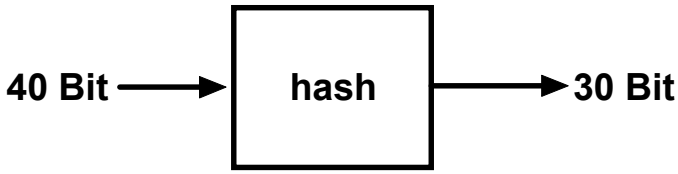
2. **Sowohl System 2 als auch System 3 wünschen für ihre jeweiligen Prozesse P2 und P3 Shared Rechte für Locks B und C, die beide in die Hash Klasse k fallen . Die CF registriert dies in der Bitmap für k und erteilt die Rechte.**

3. **Wenn jetzt System 1 Exclusive Rechte für ein Lock der Hash Klasse k will, erhält es von der CF die Bit Map der Klasse k zurück. System 1 hat jetzt die Aufgabe, weitere Maßnahmen mit den betroffenen Systemen 2 und 3 (und nur diesen) direkt auszuhandeln.**

Hashing

100 TByte Daten **10^{12} Objekte**
40 Bit Lock Namen

Lock Tabelle mit 10^{9} Einträgen à 8 Byte
8 GByte



IMS Lock Namen = 19 Byte = 152 Bit

Nutzung der Lock Tabelle in der CF

Je 1 Eintrag in der Lock Tabelle für jedes aktive Data Item.

Nur 1 System ist Besitzer (Owner), hat Schreibrechte (exclusive, E, EXC). Andere Systeme können Read Rechte (shared, S, SHR) haben.

Lock Tabellen Eintrag bezeichnet den Besitzer. Bitmap hält SHR Rechte von anderen Systemen fest.

Zugriff auf die Lock Tabelle über Software Hashing der Lock Namen. Beispiel: IMS Lock Name = 19 Bytes.

Hasching ⟶ Integer Wert ⟶ Offset für die Lock Tabelle

Kopie der Lock Tabelleneinträge in den einzelnen Systemen. Hier erfolgt die Auflösung von Synonymen.

Erteilt die CF exclusive (Schreib-) Nutzung für ein Lock, informiert dieses System alle anderen Systeme, die Share Rechte haben (und nur diese).

Für die Verwaltung von EXC Rechten zwischen unterschiedlichen Prozessen innerhalb des gleichen Systems ist nur das betroffene System zuständig. Kein Zugriff auf die CF bei Übergabe an einen anderen Prozess im gleichen System.

Mehrfache Lock Tabellen in der CF möglich.

Sehr komplexe Algorithmen, zum Teil nicht veröfentlicht. Anpassung an die einzelnen Subsysteme (z.B. CICSPlex).

System Lock Manager - SLM

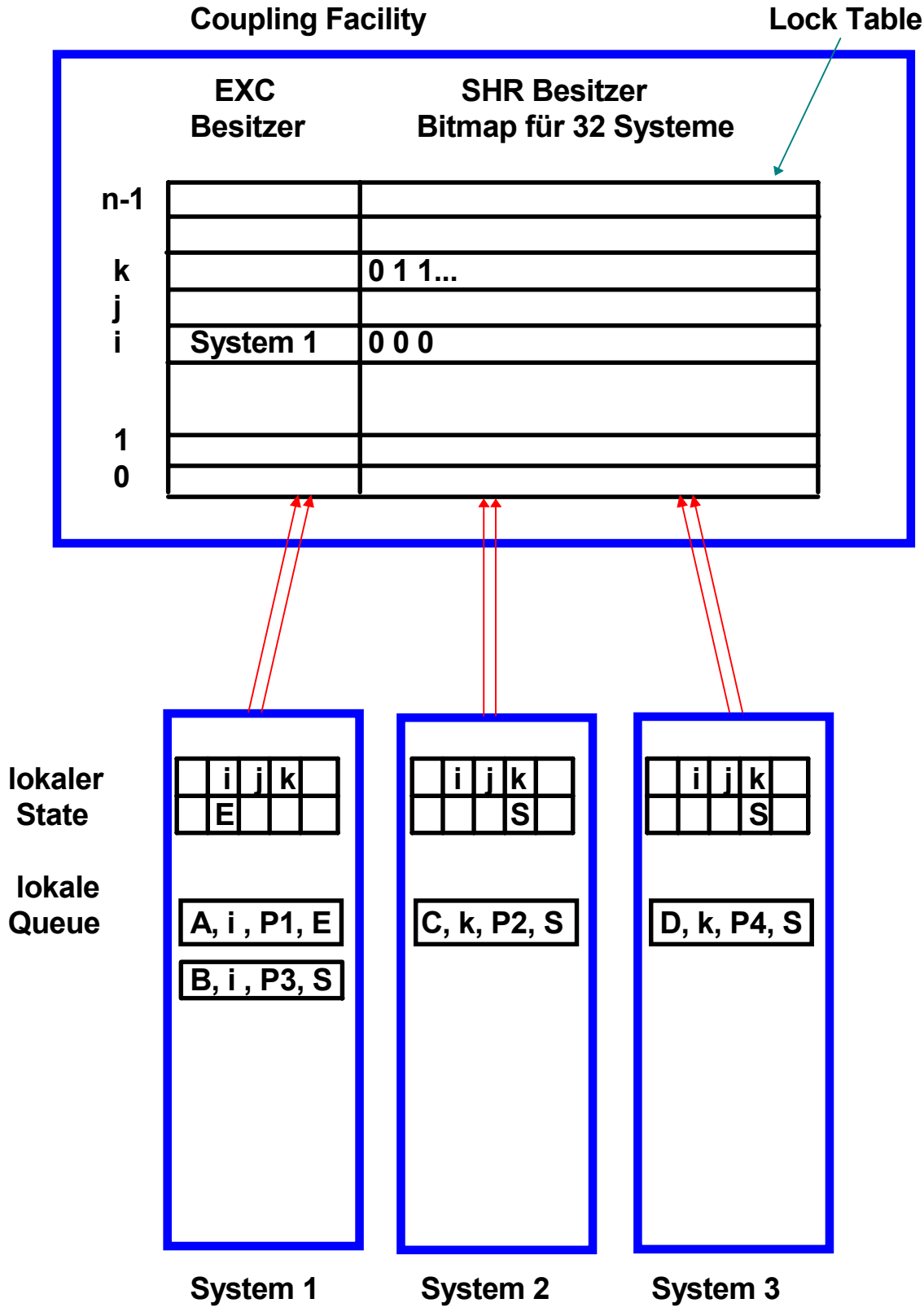
Globale Contention: Zugriff eines Systems auf ein Data Item, dessn Lock von einem anderen System gehalten wird.

SLM ist zuständig für die Auflösung von Lock Contentions

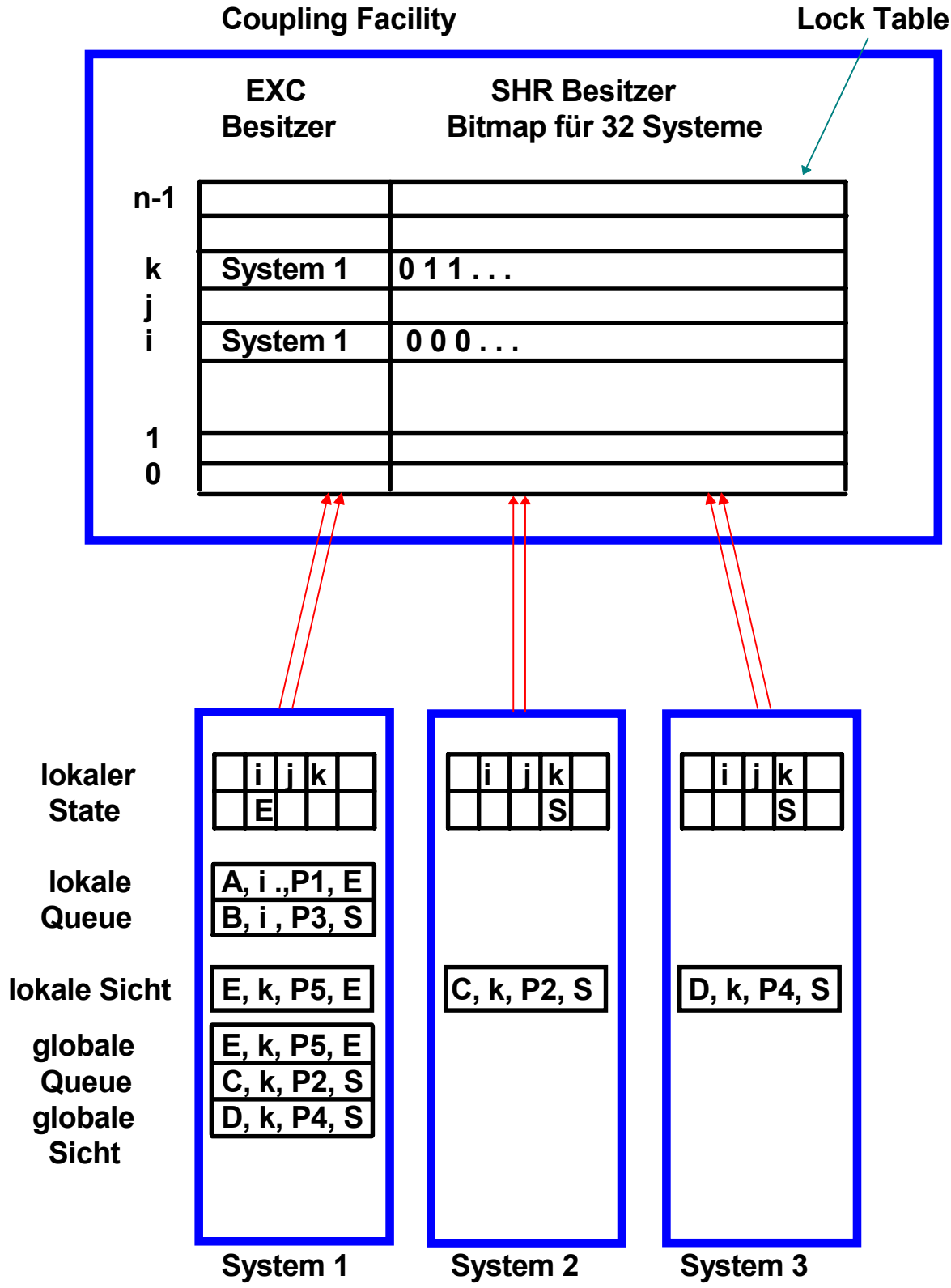
Dynamische Anpassung der durch Locks geschützten Granularität der Datenbank (möglichst groß - Kompromiss mit der Anzahl der Contentions)

Beispiel:

- **100 Transaktionen / s**
- **0,5 s Antwortzeit**
- **Multiprogramming Level = 50**
- **20 Locks / aktive Transaktion**
- **1000 aktive Locks**
- **Falsche Contention $\leq$ 0,5 %**
- **Lock Tabelle mit 200 000 Einträgen**



Wenn System 1 ein (shared) Lock B für die gleiche Hash Klasse i anfordert, ist kein Zugriff auf die CF erforderlich



System 1 fordert exclusives Lock E in Hash Klasse k an. Kein Konflickt mit den anderen Locks in Klasse k (unechter Konflikt).

Unechter Konflikt

System 1 fordert exclusives Lock E in Hash Klasse k von der Lock Tabelle der Coupling Facility an. System 2 und 3 haben ein Shared Interesse dieser Klasse. Die CF übergibt die Bit Map an System 1.

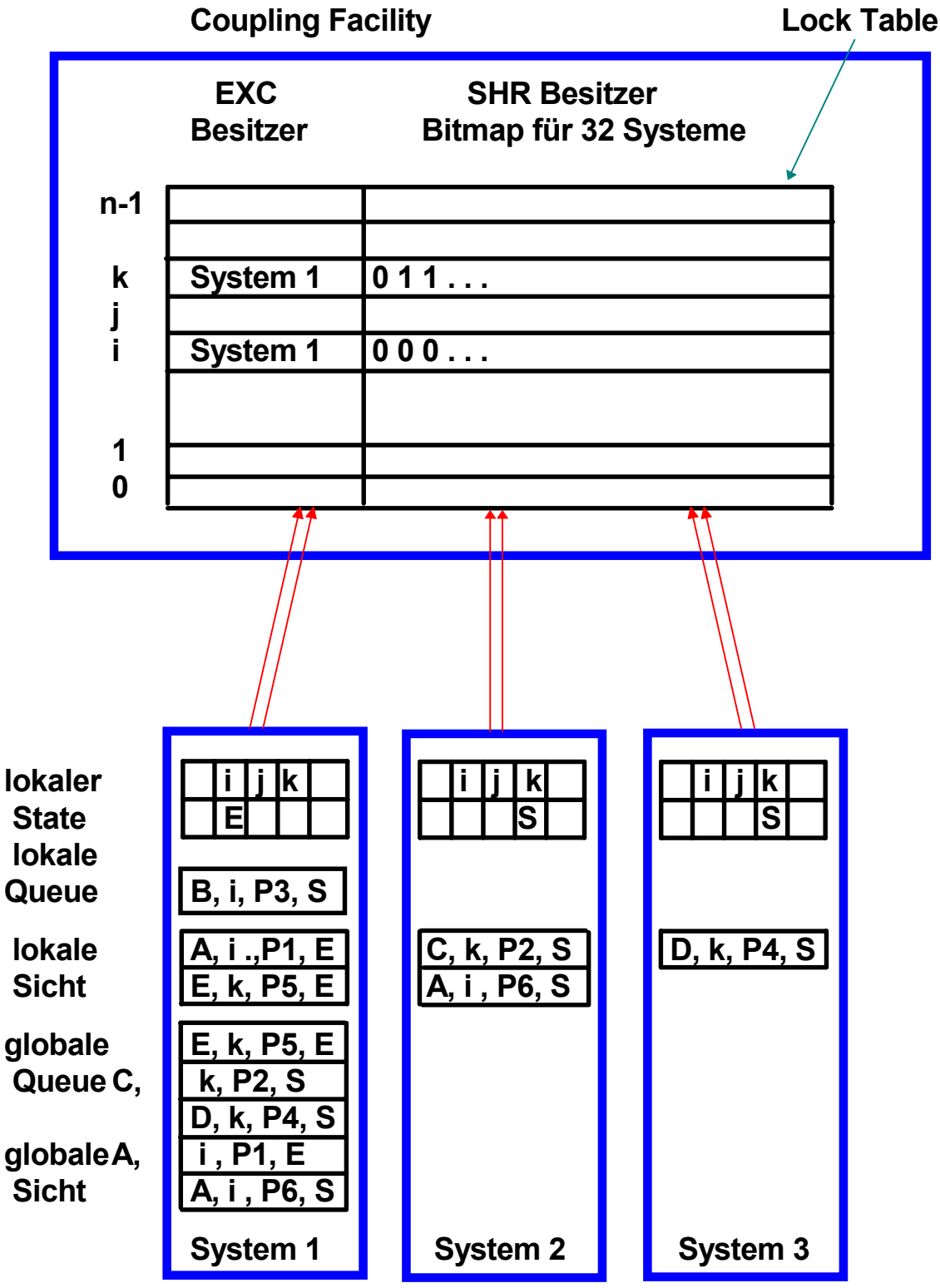
System 1 übernimmt die globale Management Verantwortung für Klasse k. Es erfrägt von Systemen 2 und 3 deren Lock Information für Klasse k. Erfolgt parallel über Hochgeschwindigkeitsverbindungen.

Nur die Systeme des Sysplex mit Locks in Klasse k sind hiervon betroffen !!! .

Annahme: Kein Konflickt mit den anderen Locks in Klasse k.

System 1 baut eine globale Queue mit allen Lock Einträgen für Klasse k auf.

Systeme 2 und 3 bewegen ihre Einträge aus ihrer lokalen Queue in ihre lokale Sicht der globalen Queue. Sie übernehmen die Verantwortung, System 1 als den globalen Manager der Klasse k bei einer Änderung des Lock Status zu benachrichtigen, z.B. Freigabe des Locks.



System 2 fordert shared Lock A in Hash Klasse i an. Echter Konflickt mit den anderen Locks in Klasse i. Wird von System 1 aufgelöst.

Echter Konflikt

System 2 fordert shared Lock A in Hash Klasse i von der Lock Tabelle der Coupling Facility an. Die CF übergibt die Bit Map an System 1 als den Besitzer von Klasse i.

System 1 hat bereits die globale Management Verantwortung für Klasse i. Es ist deshalb nicht erforderlich, andere Systeme zu benachrichtigen.

Annahme: Konflickt für Lock A mit der Anforderung von System 2. Es ist nun Aufgabe des globalen Lock Managers in System 1, den Konflikt aufzulösen. Ein möglicher Ansatz besteht darin, die Granularität des Locks zu verkleinern.

Definitionen

Ein System liest oder schreibt Blöcke zum File System (Plattenspeicher). Jeweils ein oder mehrere ganze Blöcke werden über die E/A Schnittstelle transportiert.

Seiten (Pages) sind die Einheiten, aus denen der virtuelle Adressenraum besteht (verwaltet durch den Buffer Manager).Im Falle von DB2 ist die Seitengröße gleich der Blockgröße.

Ein Slot ist die physikalische Lokation, in der ein Block auf der Plattenoberfläche abgespeichert wird.

Ein Datensatz (Record) ist die physikalische Darstellung eines Tuple. Feste oder variable Satzlänge.

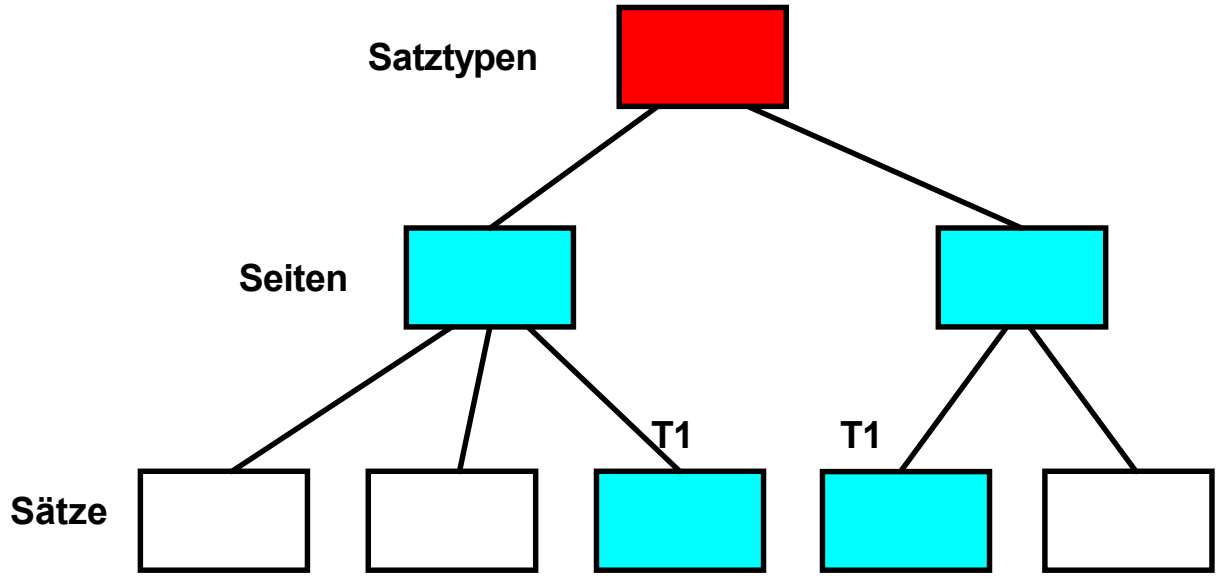
Eine Seite enthält eine Anzahl Datensätze (Records).

Ein feines Granulat (z.B. Satzsperren) führt zu einer geringen Anzahl von Konflikten zwischen Transaktionen und zu einem hohen Verwaltungsaufwand.

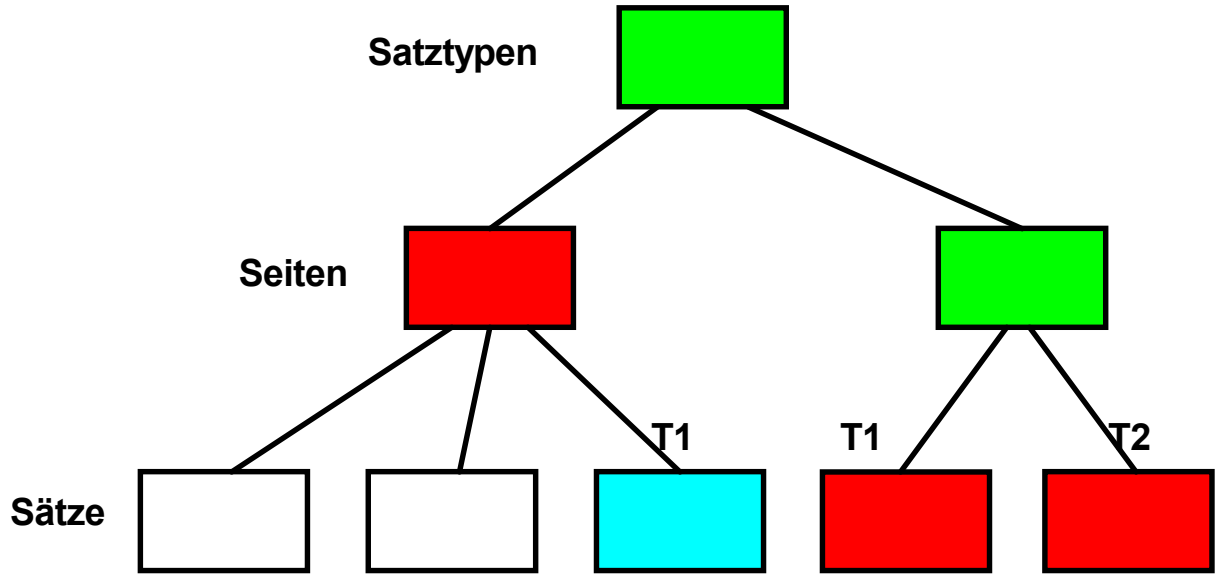
Hierarchische Sperrverfahren unterstützen 2 oder mehr Granularitäten. Für lange Transaktionen (viele Sperren) können grobe, für kurze Transaktionen feine Sperrgranulate eingesetzt werden.

Beim Sperren feiner Granulate werden die gröberen Granulate mit Anwartschaftssperren (intention locks) belegt.

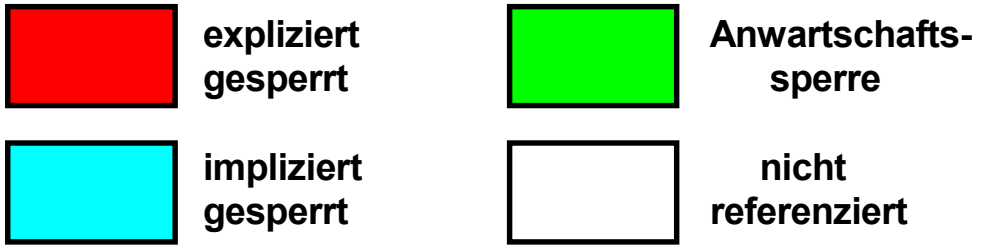
Gray, Reuter, p. 674, 751



Satztyp vollständig von Transaktion 1 gesperrt



Situation nachdem Transaktion 2 einen Satz referenziert



DB2 explizites hierarchisches Locking

Alle XES (Cross-System Extended Services) und die Lock Struktur in der CF bilden gemeinsam den Globalen Lock Manager

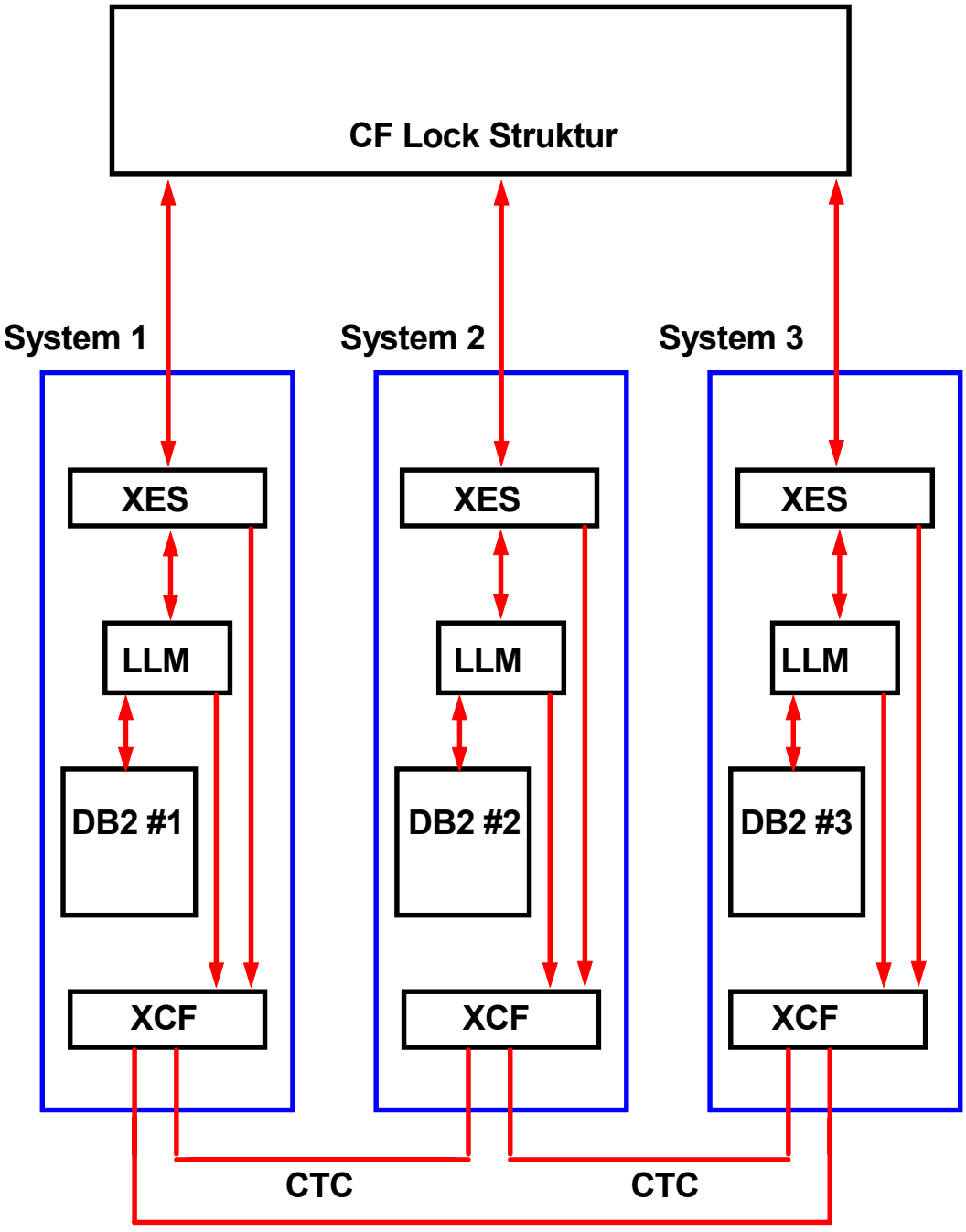
A Global Lock provides intra-DB2 and inter-DB2 Concurrency Control. A local Lock provides only intra-DB2 Concurrency Control.

Wird ein Globales Lock angefordert, überprüft der lokale Lock Manager, ob es lokal, ohne Zugriff auf die CF, vergeben werden kann. Hierzu dient das EHL Verfahren (Explicit Hierarchical Locking). In der Mehrzahl der Fälle ist der Zugriff auf die CF nicht erforderlich.

Sperren (Locking) in der CF erfolgt mit möglichst grober Granularität. Der lokale Lock Manager (LLM) verwaltet Data Items mit feinerer Granularität. Möglichst viele Lock Requests werden vom LLM abgehandelt, ohne Zugriff auf die globale Lock Struktur der CF.

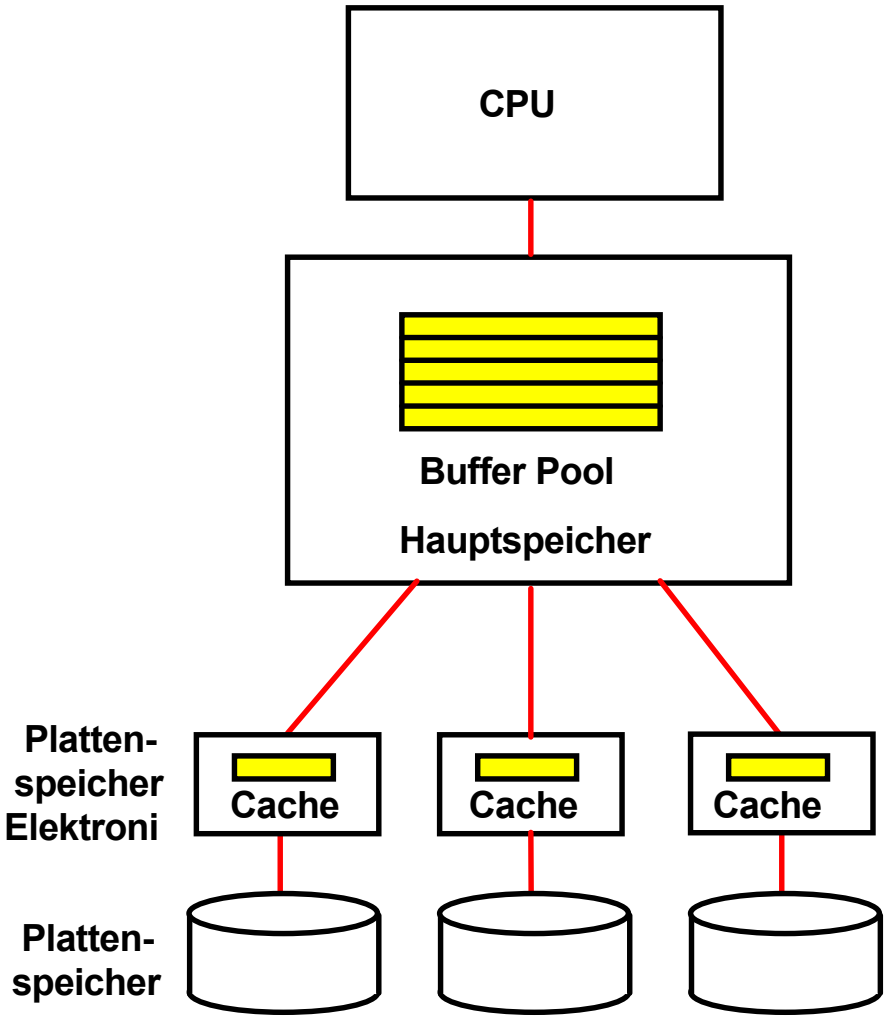
Angenommen, System 2 fordert bei der CF ein Lock an, welches derzeitig von System 1 gehalten wird. In diesem Fall benachricht die CF System 2 über die Vergabe des Globalen Locks. System 2 kann nun über die CTC (Channel-To Channel) Verbindung mit System 1 eine Herabstufung und feinere Granularität aushandeln. Die CTC Verbindung wird physikalisch über den FICON Director hergestellt.

Es besteht die Chance, daß auf einer unteren Hierarchiestufe kein Lock Konflikt besteht.



DB2 Globales Locking

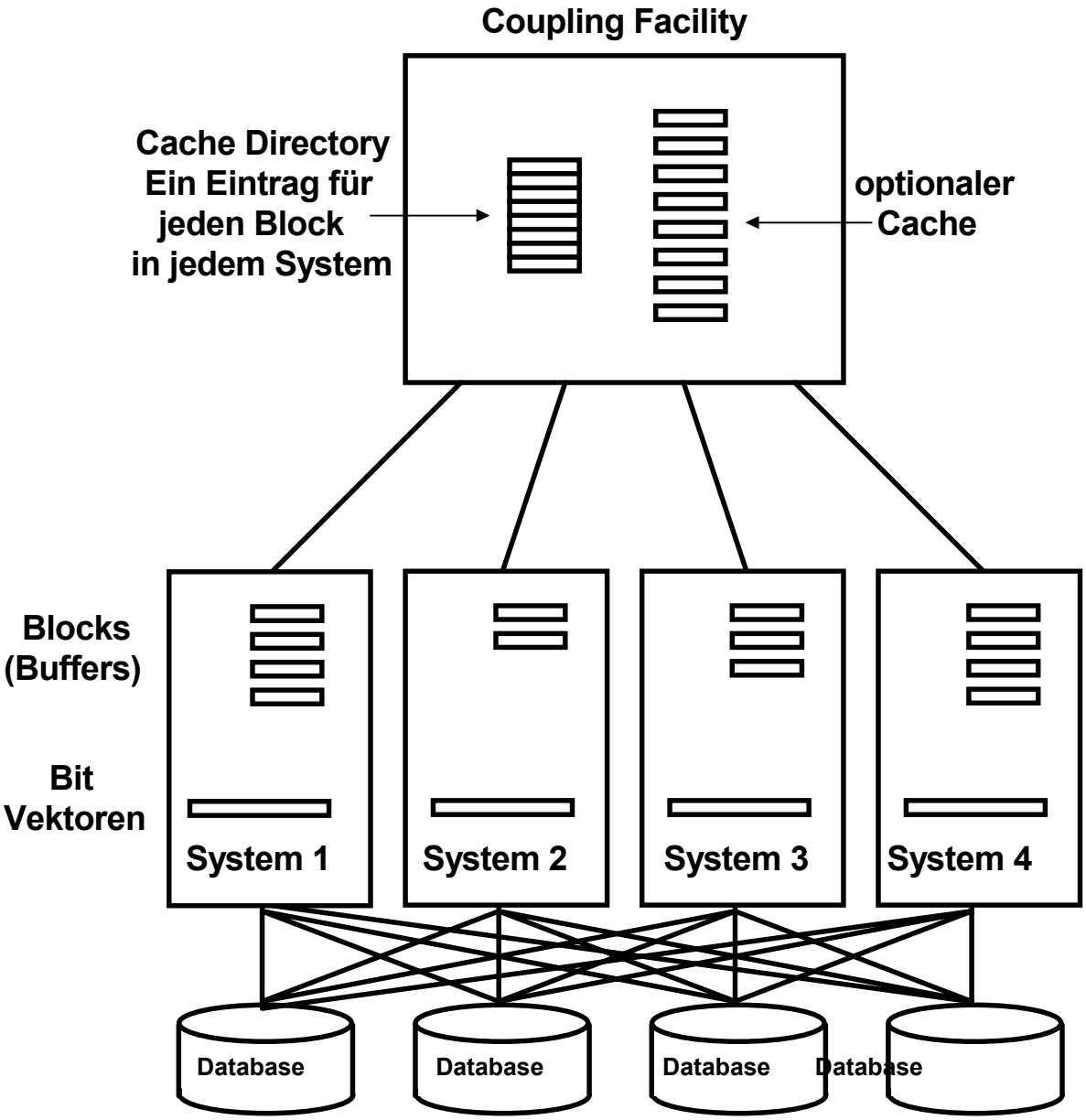
XES	**Corss System extended Services**
LLM	**Local Lock Manager, (Inter Resource Lock Manager, IRLM)**
XCF	**Cross System Coupling Facility**
CTC	**Channel to Channel Verbindung**



Plattenspeicher Cache und Hauptspeicher Buffer Pool

Ein *lokaler Cache* im Hauptspeicher eines Knotens (System) wird als *Buffer Pool* bezeichnet. Er besteht aus einzelnen Puffern (Buffers), die Datenbankobjekte aufnehmen.

Zusätzlich werden Daten in einem Plattenspeicher Cache gespeichert, der Bestandteil der Plattenspeicher Elektronik ist.



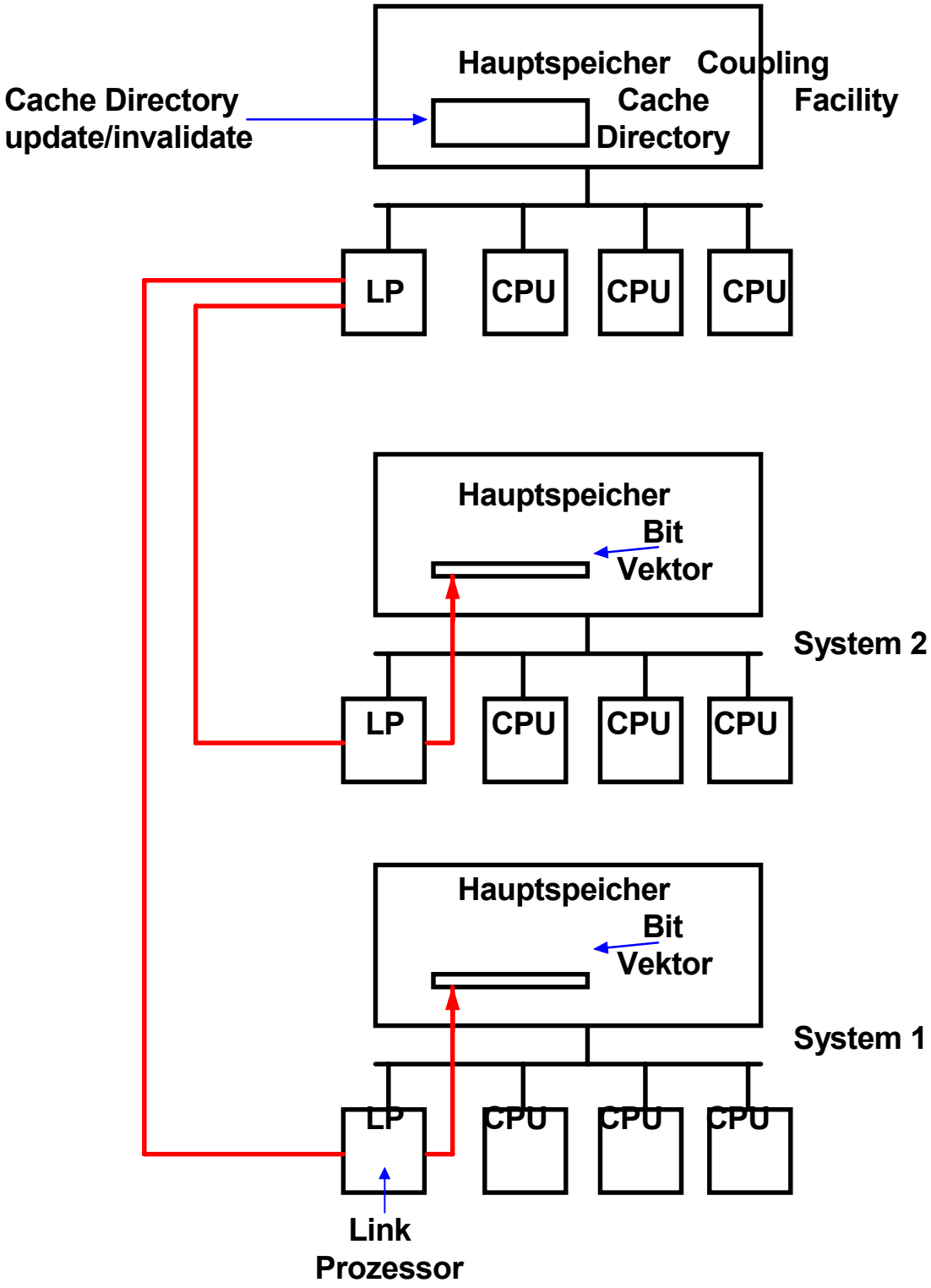
Aller Datentransfer in 4 KByte Blöcken.

System 1 Read from Disk
1. Load Block from Disk
2. Register with CF Directory
3. add Bit in Bit Vector

System 1 Write (to local Buffer)
1. Register with CF
2. CF invalidates all Bit Vectors
3. Write to local Buffer

System 2 Read from Disk
1. Load Block from Disk
2. Register with CF Directory
3. add Bit in Bit Vector

System 2 Read from Buffer
1. Read
2. detect invalid Bit in local Bit Vector
3. .....



Cache Directory Update in der Coupling Facility bewirkt, daß über die Link Prozessoren der Systeme deren Bit Vektoren im Hauptspeicher abgeändert werden, ohne daß der normale Programmablauf dadurch beeinflußt wird (kein Prozesswechsel)

Cast Out

Beispiel DB2

Daten im CF Cache haben keinen direkten Zugriff auf die Plattenspeicher

DB2 Instanzen unterhalten jeweils einen Cast-Out Thread

Cast-Out Verantwortung nach Round-Robin (oder andere) Algorithmus den einzelnen Threads zugeordnet

Cast-Out erfolgt jeweils für eine Gruppe von Seiten

z.B. LRU Algorithmus (oder besser)

CF Cache Recovery

Alle Daten befinden sich in den Puffern der einzelnen Systeme.

Im Fehlerfall hieraus Rebuild des CF Cache Inhaltes

Coupling Facility Cache

Der lokale Buffer Pool im System 1 enthält Seiten (Blöcke) mit Records, die gerade bearbeitet werden. Solange die Transaktion nicht abgeschlossen ist, verhindert der Lock Manager einen Zugriff durch ein anderes System (z.B. System 2).

Wenn die Transaktion abgeschlossen ist (commit), werden die Locks freigegeben. Die Puffer bleiben in System 1 - evtl. werden sie demnächst wieder gebraucht.

Greift System 2 jetzt auf den gleichen Block zu, entsteht ein Kohärenzproblem.

Lösung: „Force-at-Commit“ . Bei Transaktionsabschluss erfolgt ein update des Cache Directories.

(DB2 schreibt zusätzlich die Seite in den CF Cache. Dies ist ein Store-in-Cache; die CF Cache Version des Blockes kann jüngeren Datums sein als die Version auf dem Plattenspeicher. Bei anderen Anwendungen als DB2 erfolgt ggf. nur ein update des CF Cache Directories).

Die CF sendet eine „Cross-Invalidate (CI) Nachricht an alle anderen Systeme.

Die CI Nachricht ändert den lokalen State Vector innerhalb des Hauptspeichers eines jeden Systems ab. Dies geschieht durch den Link Prozessor und verursacht keine CPU Unterbrechung.

CF List / Queue Strukturen

3 Zugriffsmöglichkeiten

- **LIFO Queue**
- **FIFO Queue**
- **Key Sequenced**

Anwendungsbeispiele:

- **Clusterweite RACF Steuerung**

- **Work Load Management Instanzen tauschen periodisch Status Information aus um Transaktionen dynamisch an unterbelastete Systeme weiter zu reichen**

Sysplex Eigenschaften

WLM bewirkt eine Dynamische Steuerung des Job Mixes (systems mix of active jobs)

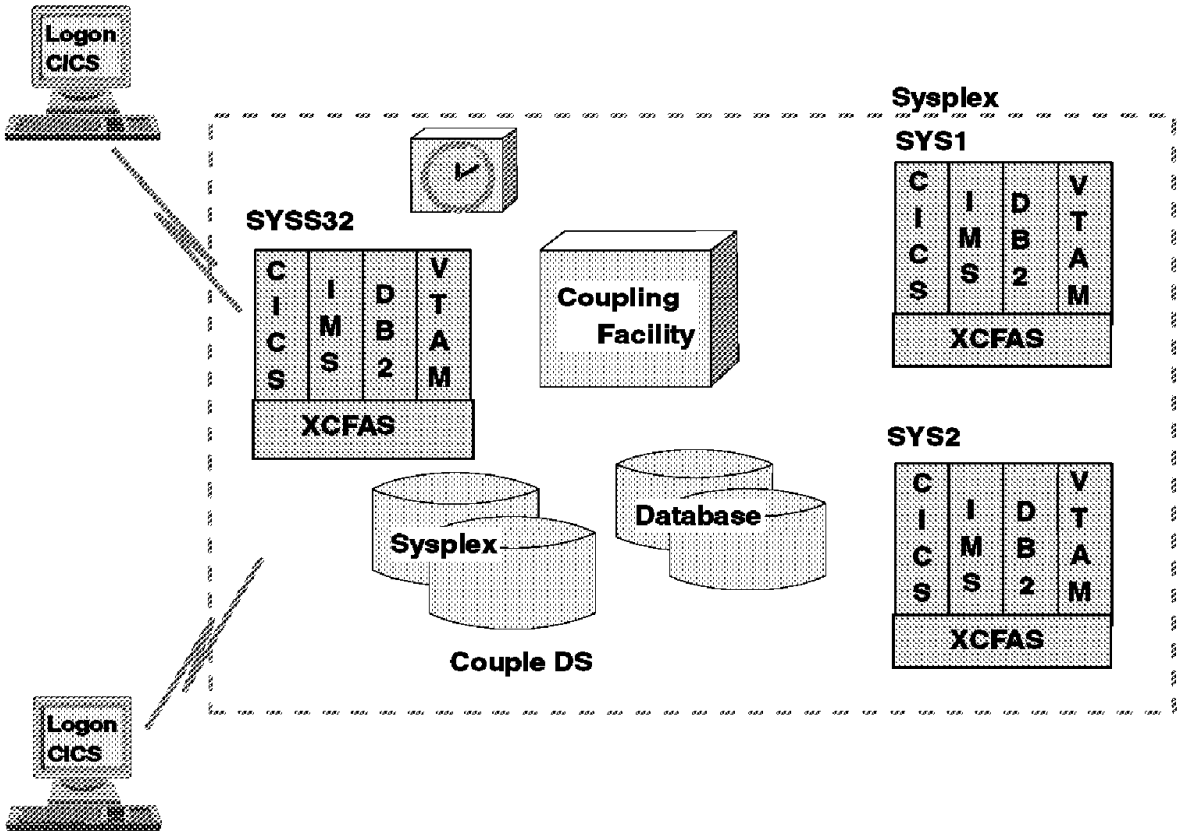
WLM und Sysplex support ist verfügbar für:

- **CICS**
- **IMS**
- **DB2**
- **USS**
- **VSAM**
- **Communication Server**
- **WebSphere**
- **SAP R/3**

Diese Subsysteme können die Coupling Facility (CF) nutzen. CF Locking und Caching sind WLM enabled. Locking geschieht auf der Record Ebene. Der Work Load Manager (WLM) ist Bestandteil jedes OS/390 Images. Kommunication über die CF.

"CICSplex" ist eine CICS Version, die auf einem Sysplex läuft.

Geographically dispersed Sysplex (GDPS).

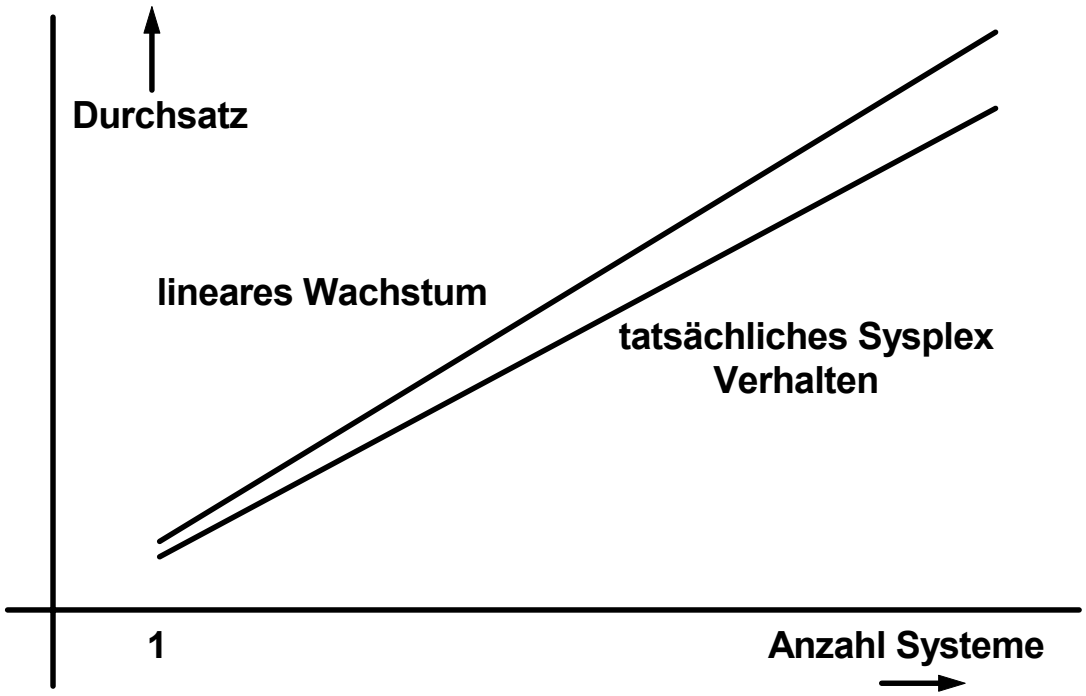


Cross Coupling Facility Address Space: XCFAS

Angenommen mehrere Instanzen einer Anwendung oder eines Subsystems auf unterschiedlichen Knoten eines Sysplex, z.B. CICS oder WebSphere. Mit Hilfe von XCFAS können die Instanzen Status Information austauschen oder miteinander kommunizieren.

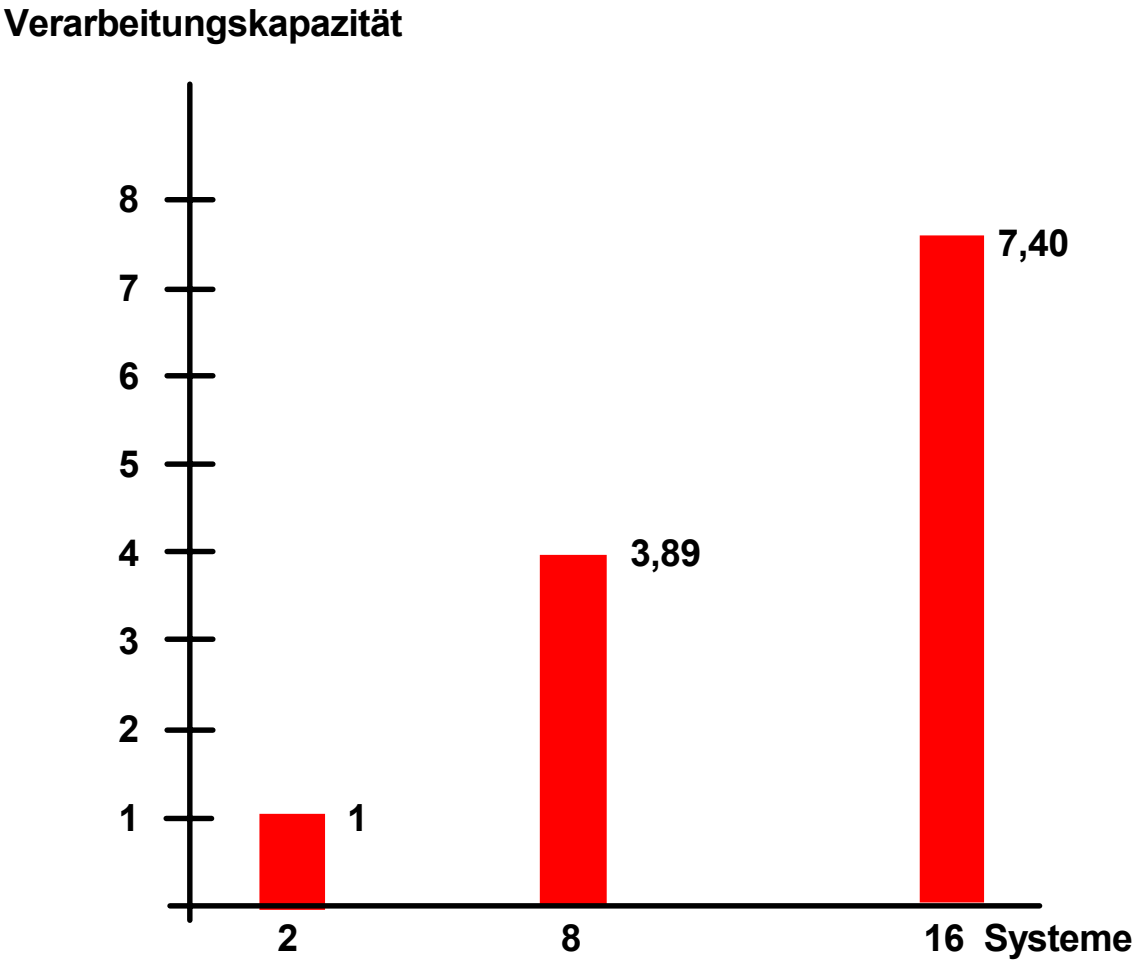
Die gemeinsam genutzten Daten befinden sich als Listen- oder Queue-Strukturen auf der Coupling Facility. Der Zugriff auf diese Daten erfolgt mit Hilfe des Cross-System Extended Services (XES) Protokolls, welches Zugriffs- und Verwaltungsdienste zur Verfügung stellt.

Sysplex Overhead



Installation	Anzahl Systeme	% Sysplex Overhead
A	4	11 %
B	3	10
C	8	9
D	2	7
E	11	10
Relational Warehouse Workload	2	13,30

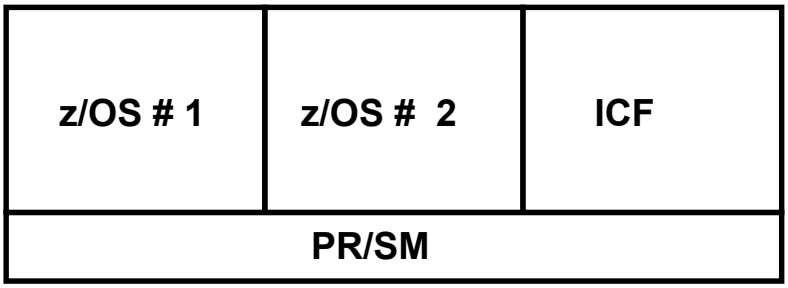
Die Sysplex Software (wenn installiert) erzeugt in jedem System zusätzlichen Overhead, selbst wenn der Sysplex nur aus einem einzigen System besteht. In jedem System wird zusätzliche CPU Kapazität benötigt um den gleichen Durchsatz zu erreichen.



Parallel Sysplex Leistungsverhalten

CICS Transaktionsmanager, CICSplex System Manager, IMS Datenbank

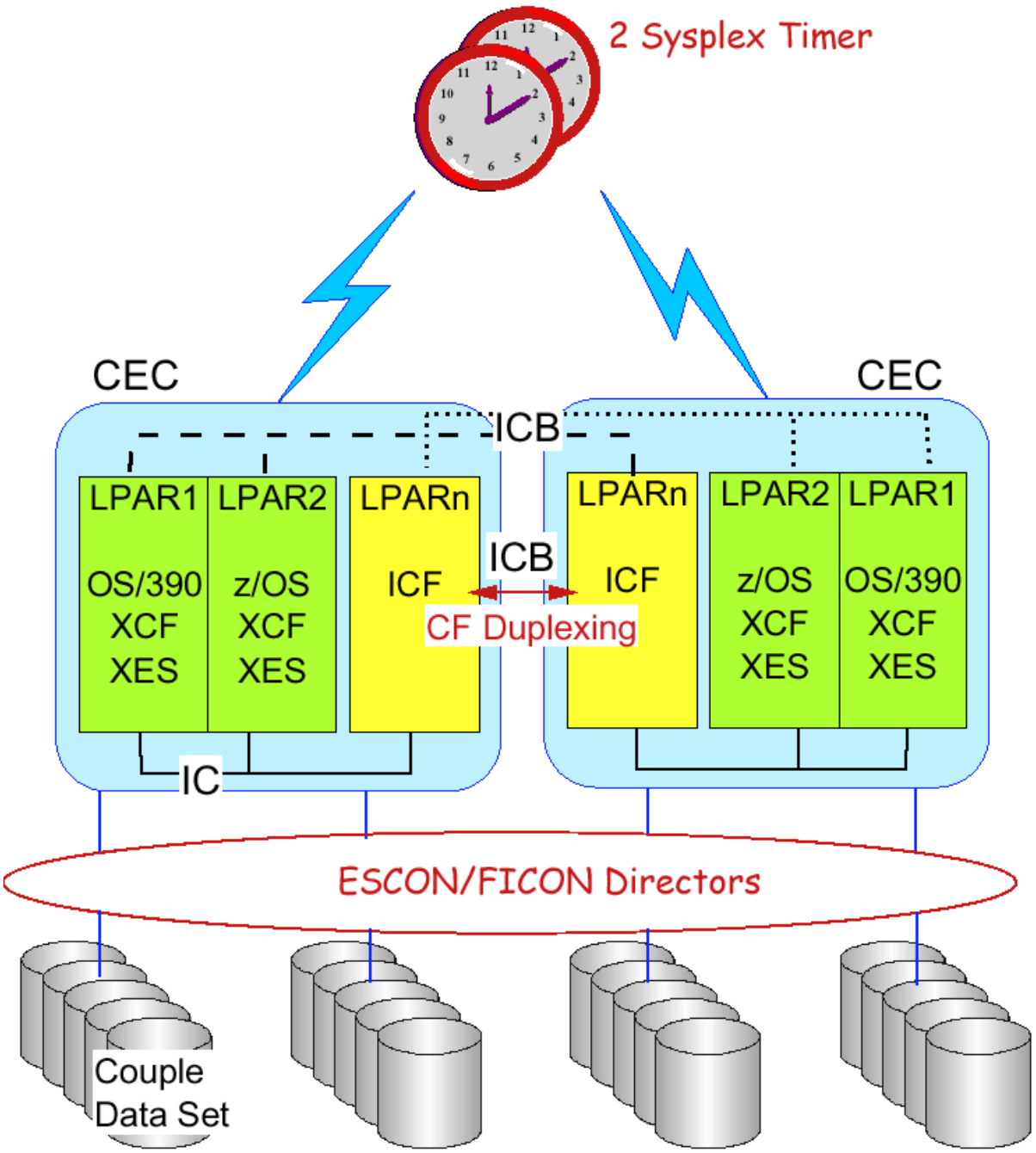
Mischung von OLTP, Reservierung, Data Warehouse und Bankanwendungen



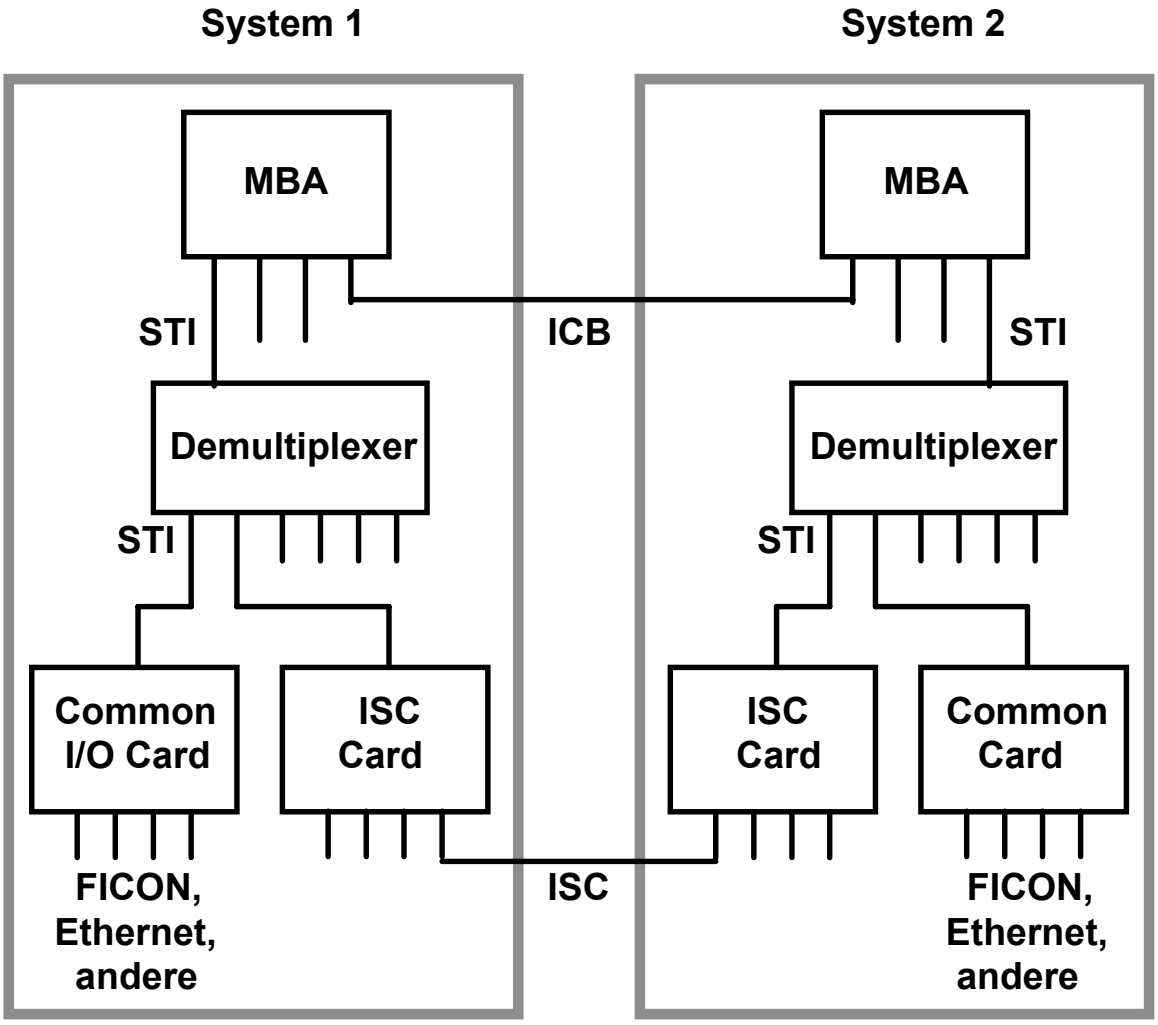
Die Coupling Facility (CF) kann auch in einer Logischen Partition (LPAR) eines zSeries Knotens (System) untergebracht werden, wobei in anderen Partitionen zwei oder mehr z/OS oder OS/390 Images laufen. Diese Art der CF wird als Internal Coupling Facility (ICF) bezeichnet. Die ICF läuft auf einer oder mehreren hierfür dedizierten CPUs eines SMP.

Die ICF stellt CF Funktionalität ohne Coupling Links zu Verfügung. Letztere werden durch PR/SM emuliert.

Aus Zuverlässigkeitsgründen sollten immer mindestens 2 CFs vorhanden sein.



Integrierte Coupling Facility



System Area Network

Von jedem I/O Port (MBA Chip) gehen 4 full duplex STI Busse zu 4 Demultiplexoren. Jeder Demultiplexor hat 6 STI full duplexBus Ausgänge. Jeder dieser Ausgänge geht zu einer I/O Card, z.B. einer Common I/O Card oder ISC Card. Jede dieser Karten hat 4 Ausgänge. Von den maximal 96 Ausgängen sind maximal 84 nutzbar für I/O Cards (z.B. FICON, Gigabit Ethernet und andere). Eine spezielle I/O Card ist die ISC Card, die es gestattet, zwei zSeries Server über eine bis zu 20 km lange Glasfaserverbindung zu koppeln.

Alternativ können zwei zSeries Server über den (elektrischen) ICB Bus gekoppelt werden; die maximale Entfernung beträgt hierbei 10 Meter.