Einführung in z/OS Enterprise Computing

Prof. Dr. Martin Bogdan
Dr. rer. nat. Paul Herrmannn
Prof. Dr.-Ing. Wilhelm G. Spruth

WS 2008/2009

Teil 13

Sysplex Coupling Facility

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

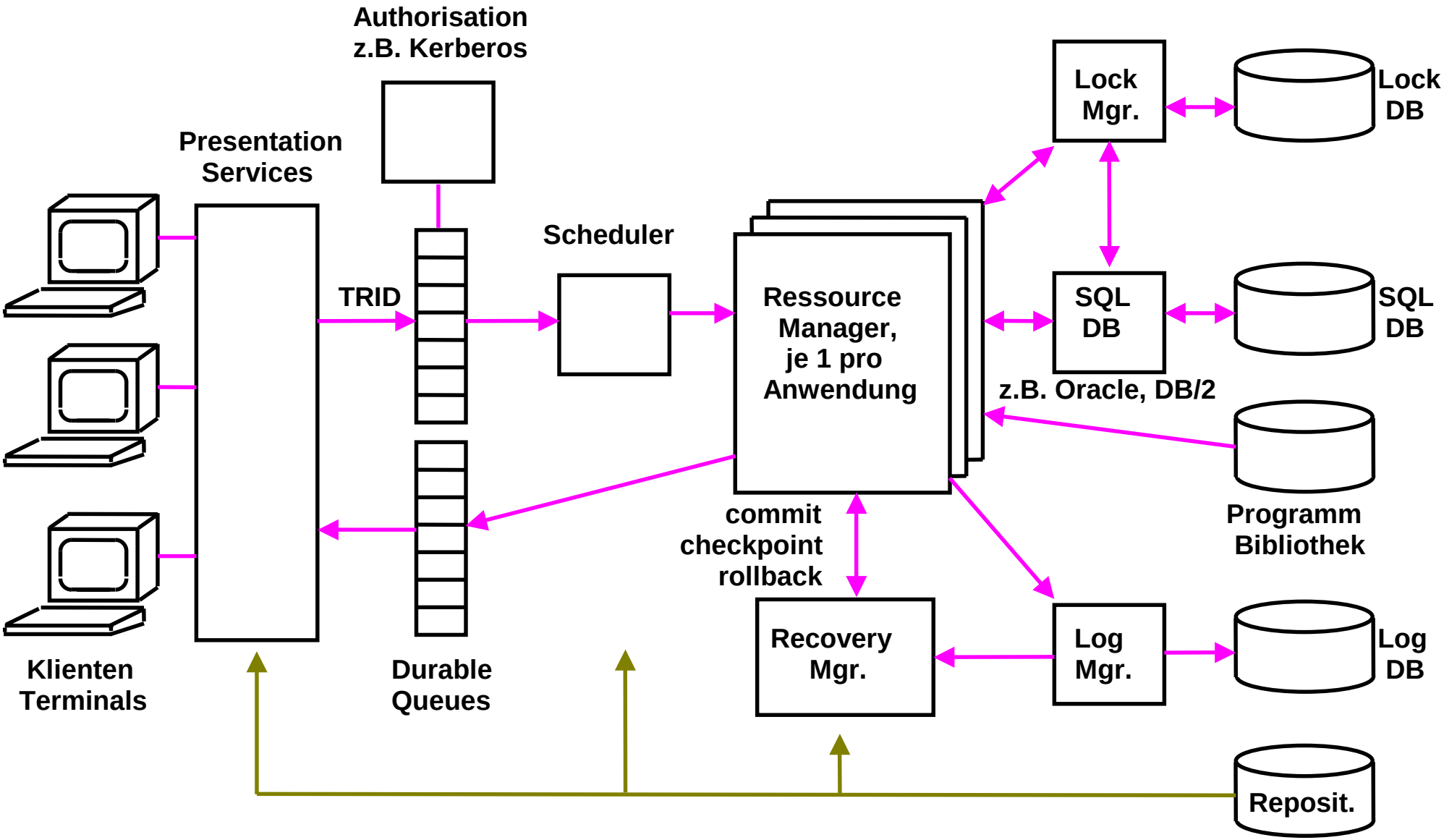
Sysplex Hardware:

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



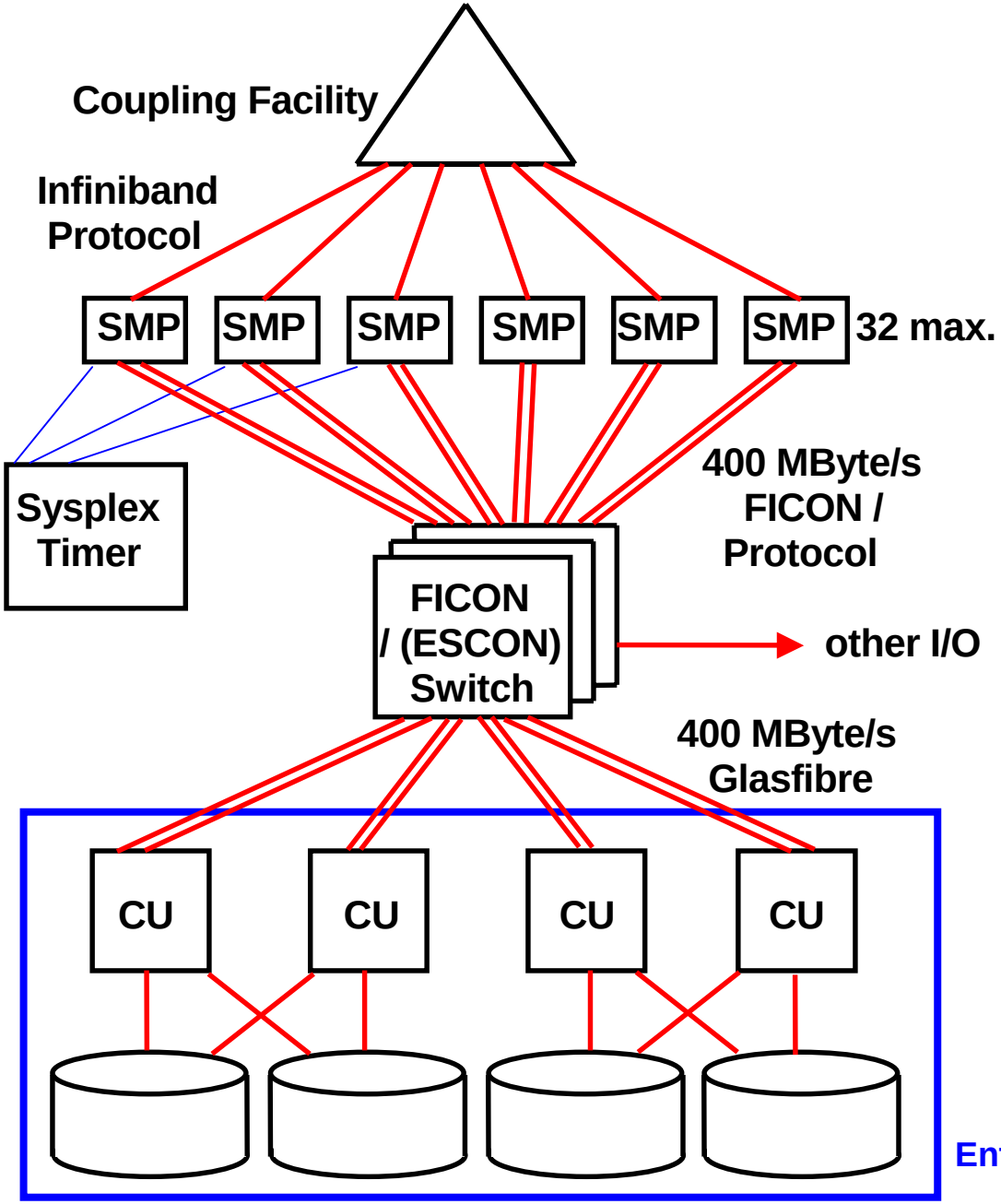
Struktur eines TP Monitors

Klienten (Arbeitsplatzrechner) werden häufig als „Terminals“ bezeichnet.

zSeries Coupling Facility

Großrechner bearbeiten
mehrere 1000 Transaktionen / Sekunde

ACID Bedingungen



Sysplex with Coupling Facility

Ein Sysplex besteht aus bis zu 32 System z (oder S/390) Rechnern, die über FICON Glasfasern und FICON Switche mit Plattenspeichern in der Form von einem oder mehreren Enterprise Storage Servern verbunden sind. Ein Enterprise Storage Server emuliert mehrere Control Units (CU).

Weiterhin sind 1 oder in der Regel 2 Coupling Facilities vorhanden.

Eine Coupling Facility ist ein regulärer System z oder S/390 Rechner, auf dem „Coupling Facility Code“ an Stelle eines Betriebssystems läuft.

Weiterhin sind 1 oder 2 Zeitgeber (Sysplex Timer) vorhanden

Parallel Sysplex Cluster Technology

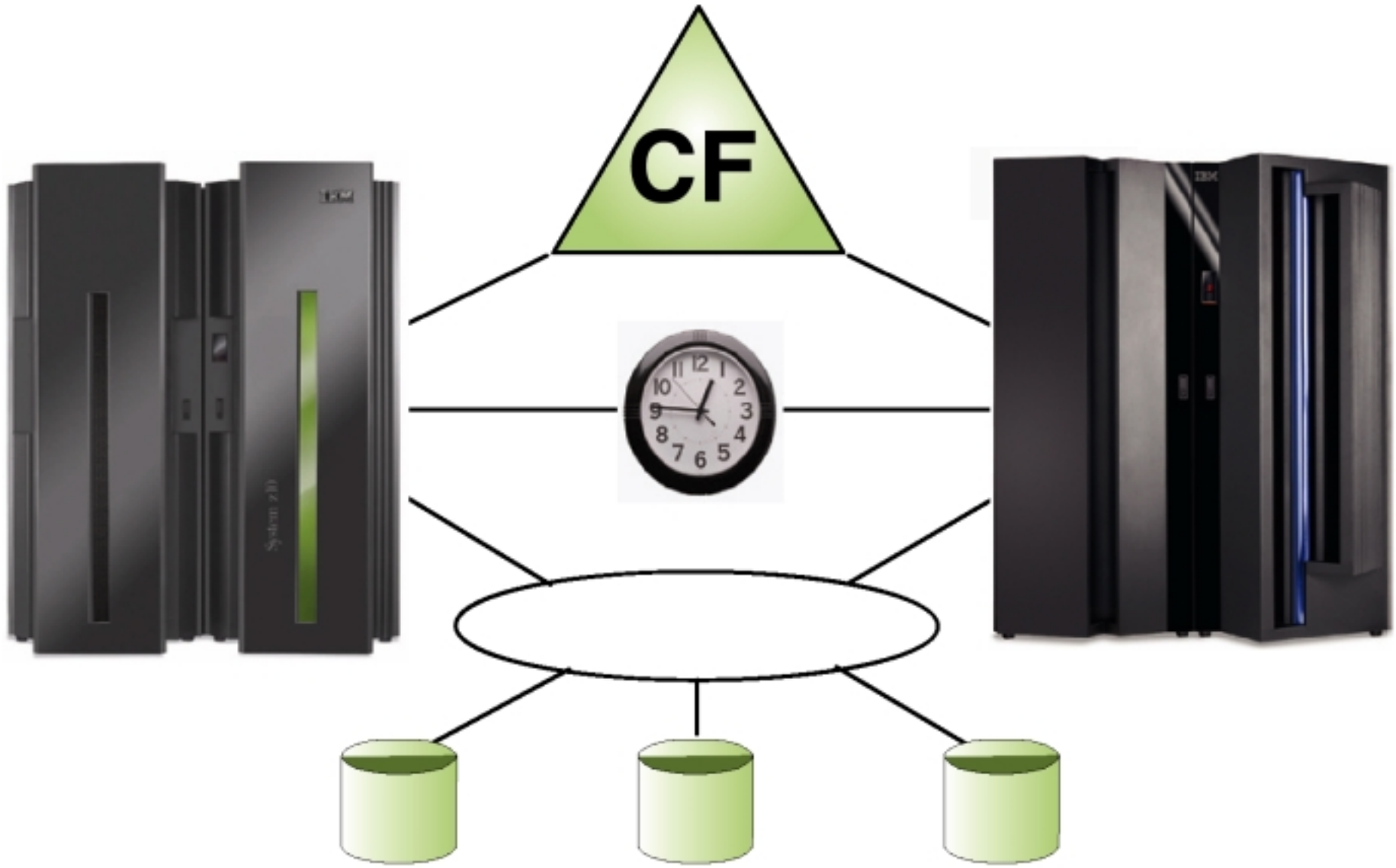
Mehrfache z/OS oder S/390 Systeme verhalten sich so, als wären sie ein einziges System (Single System Image).

Parallel Sysplex Cluster Technology Komponenten:

- **Prozessoren mit Parallel Sysplex Fähigkeiten**
- **Coupling Facility**
- **Coupling Facility Control Code (CFCC)**
- **Glasfaser Hochgeschwindigkeitsverbindungen**
- **ESCON oder FICON Switch**
- **Sysplex Timer**
- **Gemeinsam genutzte Platten (Shared DASD)**
- **System Software**
- **Subsystem Software**

Die Coupling Facility ermöglicht Data Sharing einschließlich Datenintegrität zwischen mehrfachen z/OS oder S/390 Servern

Der Sysplex Zeitgeber (Timer) stellt allen z/OS und OS/390 Instanzen eine gemeinsame Zeitbasis zur Verfügung. Dies ermöglicht korrekte Zeitstempel und Ablaufsequenzen bei Datenbank Änderungen. Dies ist besonders bei Datenbank-Recovery Operationen wichtig.



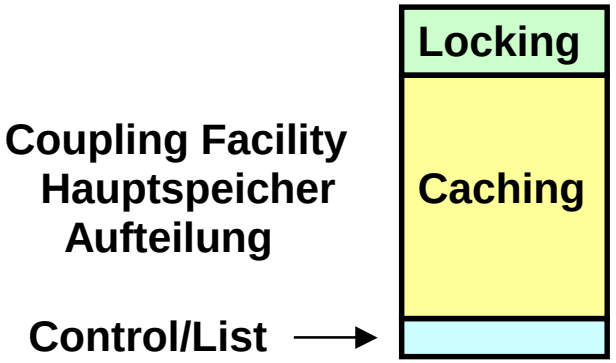
zSeries Coupling Facility

Großrechner bearbeiten mehrere 1000 Transaktionen/s unter ACID Bedingungen

Coupling Facility

Die Coupling (CF) Facility ist in Wirklichkeit ein weiterer zSeries Rechner mit spezieller Software (kein reguläres Betriebssystem). Die Aufgaben der CF sind:

- **Locking**
- **Caching**
- **Control/List Structure**



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.

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

Die Coupling Facility ist über Glasfaser Verbindungen mit einem optimierten Protokoll und spezieller Hardware Unterstützung mit den Knoten (Systemen) des Sysplex verbunden.

Locking Problem

Angenommen zwei Transaktionen, die auf unterschiedlichen Systemen des Sysplex laufen. Beide Transaktionen greifen auf die beiden zwei Variablen d1 und d2 zu.

Anfangswerte: d1 = 15, d2 = 20 .

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

Zeit ↓

Die beiden Abhängigkeiten:

Dirty Read
Eine Transaktion erhält veraltete Information

Lost update
Eine Transaktion überschreibt die Änderung einer anderen Transaktion

müssen gesteuert werden.

Das Ergebnis ist: d1 = - 5, d2 = 40 , obwohl die if-Bedingung ein negatives Ergebnis verhindern sollte.

Benutzung von Locks (Sperren)

Transaktion 1	Transaktion 2
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	GetReadLock (d1)
	read d1
	if d1 > 10
	GetWriteLock (d1)
	GetWriteLock (d2)
	sub d1, 10
	add d2, 10
end if	end if

Das Senden einer Nachricht informiert Transaktion 2, dass ihr Wissen über den Zustand der beiden Variablen d1 und d2 nicht mehr gültig ist.

Transaktion 2 muss sich neu über den Zustand der Variablen d1 informieren, ehe sie ein Update von d1 vornimmt.

Frage: Woher weiß Transaktion 1, dass Transaktion 2 ein Interesse an der Variablen d1 hat (ein shared Lock besitzt) ?

Ergebnis: d1 = + 5, d2 = 30

Two-Phase Locking
Two-Phase Transaktion

In Transaktionssystemen und Datenbanksystemen werden Locks (Sperren) benutzt, um Datenbereiche vor einem unautorisierten Zugriff zu schützen. Jedem zu schützenden Datenbereich ist ein Lock fest zugeordnet. Ein Lock ist ein Objekt welches (mindestens) über 4 Methoden und zwei Zustände S und E verfügt. Die Methode

- **GetReadLock** **reserviert S Lock (shared),**
- **GetWriteLock** **reserviert E Lock (exclusive),**
- **PromoteReadtoWrite** **bewirkt Zustandswechsel S $\rightarrow$ E,**
- **Unlock** **gibt Lock frei.**

Mehrere Transaktionen können ein S Lock für den gleichen Datenbereich (z.B. einen Datensatz) besitzen. Nur eine Transaktion kann ein E Lock für einen gegebenen Datenbereich 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.

Two Phase Locking ist nicht zu verwechseln mit dem 2-Phase Commit Protokoll der Transaktionsverarbeitung

Locking Protokoll

Vorgehensweise:

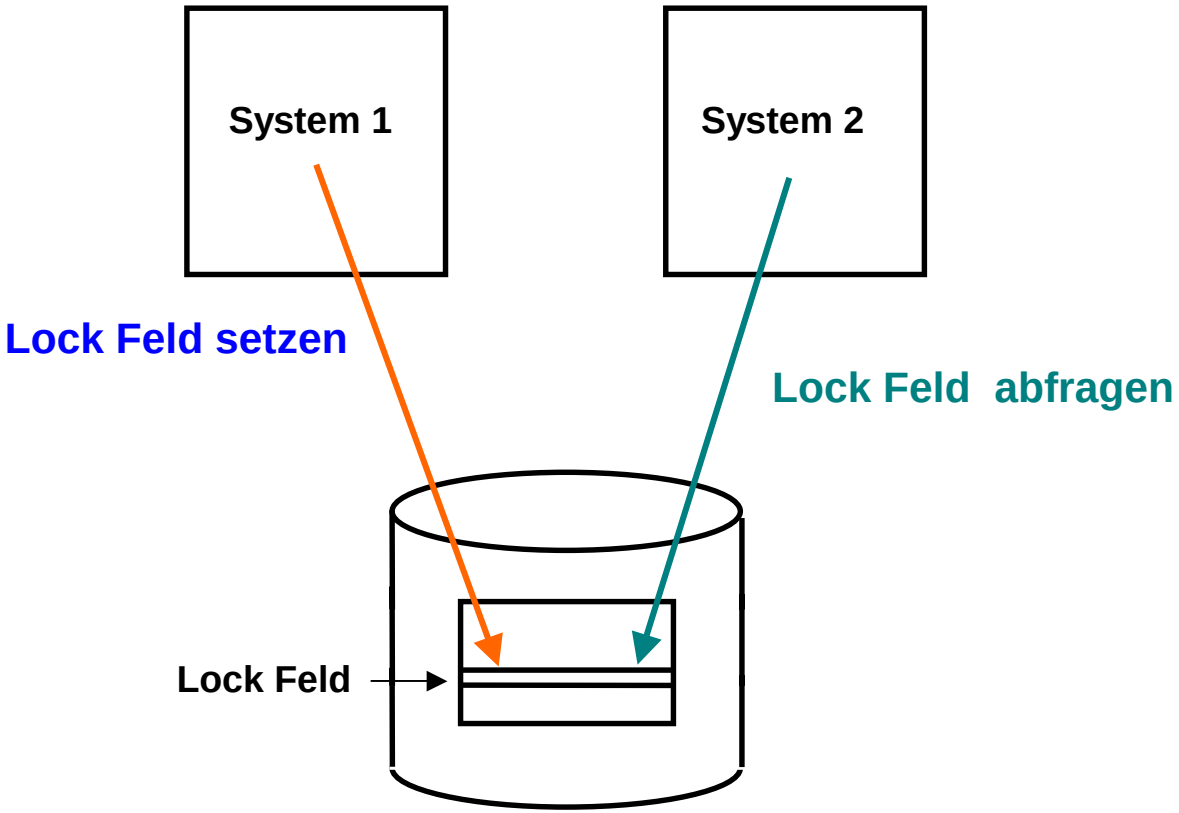
- **Shared Lock (S) erwerben vor dem erstmaligen Lesen**
- **Exclusive Lock (E) erwerben vor dem erstmaligen Schreiben**

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

Mehrere Transaktionen können das gleiche Lock im Zustand S besitzen. Nur eine Transaktion kann ein Lock im Zustand E besitzen.

Eine Transaktion kann ein Lock vom Zustand S in den Zustand E überführen. Hierzu ist es erforderlich, dass eine Nachricht an alle anderen transaktionen geschickt wird, die das gleiche Lock im Zustand S besitzen. Mittels dieser Nachricht wird das Lock für ungültig erklärt.

Lock Verwaltung

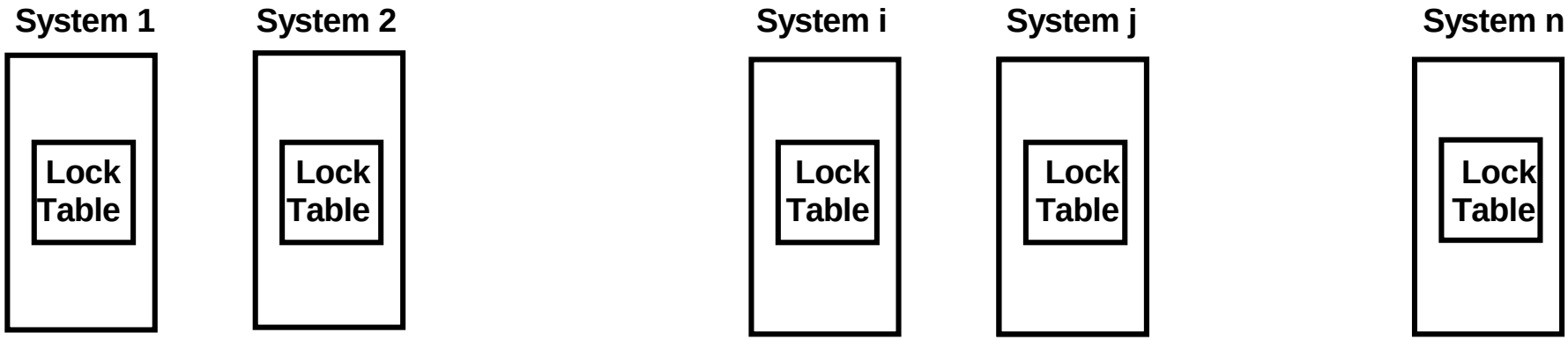


Einfachste Lösung:

Jeder zu schützende Datenbereich auf der Festplatte (z. B. eine Zeile in einer relationalen DatenbankTabelle) erhält ein zusätzliches Lock-Feld.

Bei einem Zugriff wird das Lock zunächst geprüft und dann gesetzt, ehe ein Zugriff erfolgt.

Nachteil: Die erforderlichen zuzätzlichen Zugriffe auf den Plattenspeicher sind in Hochleistungssystemen nicht akzeptabel.



Verteilte Lock Tabelle

Decentralized Locks

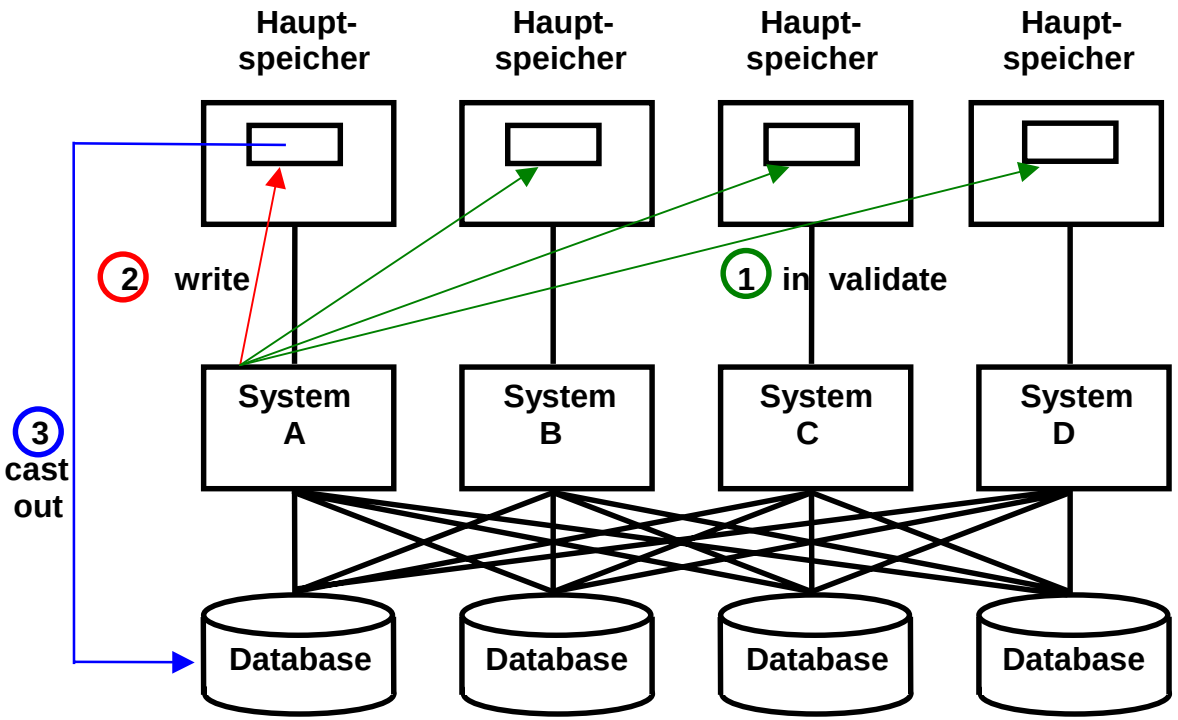
Zugriffe auf Tabellen mit Locks erfolgen recht häufig. Deswegen werden Lock Tabellen in der Regel im Hauptspeicher gehalten. Bei einem SMP mit nur einem Hauptspeicher ist dies unkritisch.

Bei einem Cluster mit mehreren Hauptspeichern sind auch mehrere Lock Tabellen in den Hauptspeichern der beteiligten Systeme vorhanden, die alle auf dem gleichen Stand gehalten werden müssen (verteilte Lock Tabelle) .

Zur Auflösung von Lock Konflikten erfolgt bei jeder Änderung in einer Lock Tabelle entweder ein Broadcast (Invalidate-Broadcast Kohärenzsteuerung) oder eine gezielte Nachricht von System i an System j.

Ersteres erfordert die Verarbeitung der laufenden Transaktion auszusetzen. Der Overhead kann 20 ms betragen,

Beispiele für diese Lösung sind die VAX DBMS und VAX Rdb/VMS Datenbanksysteme.

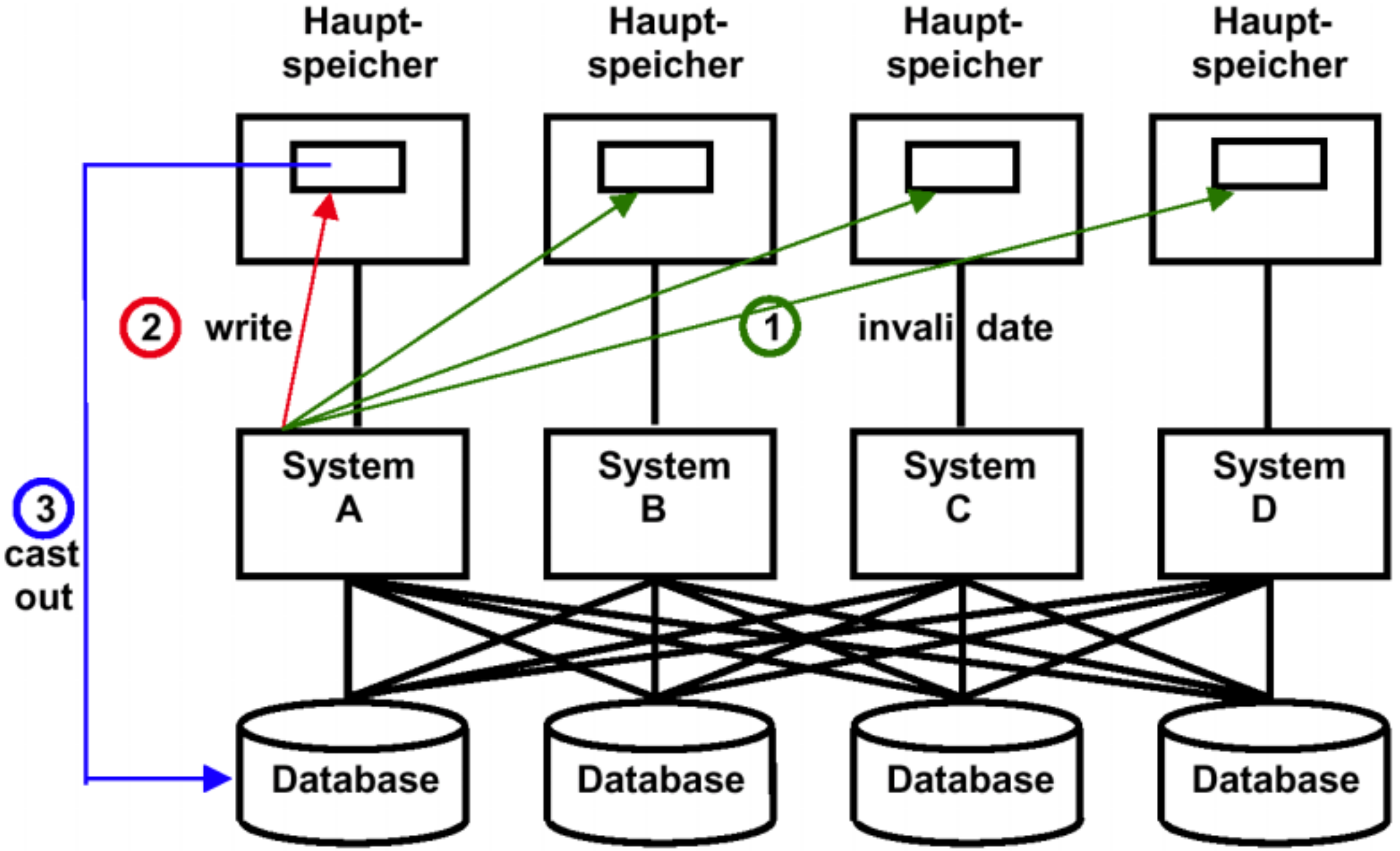


Invalidate-Broadcast Kohärenzsteuerung

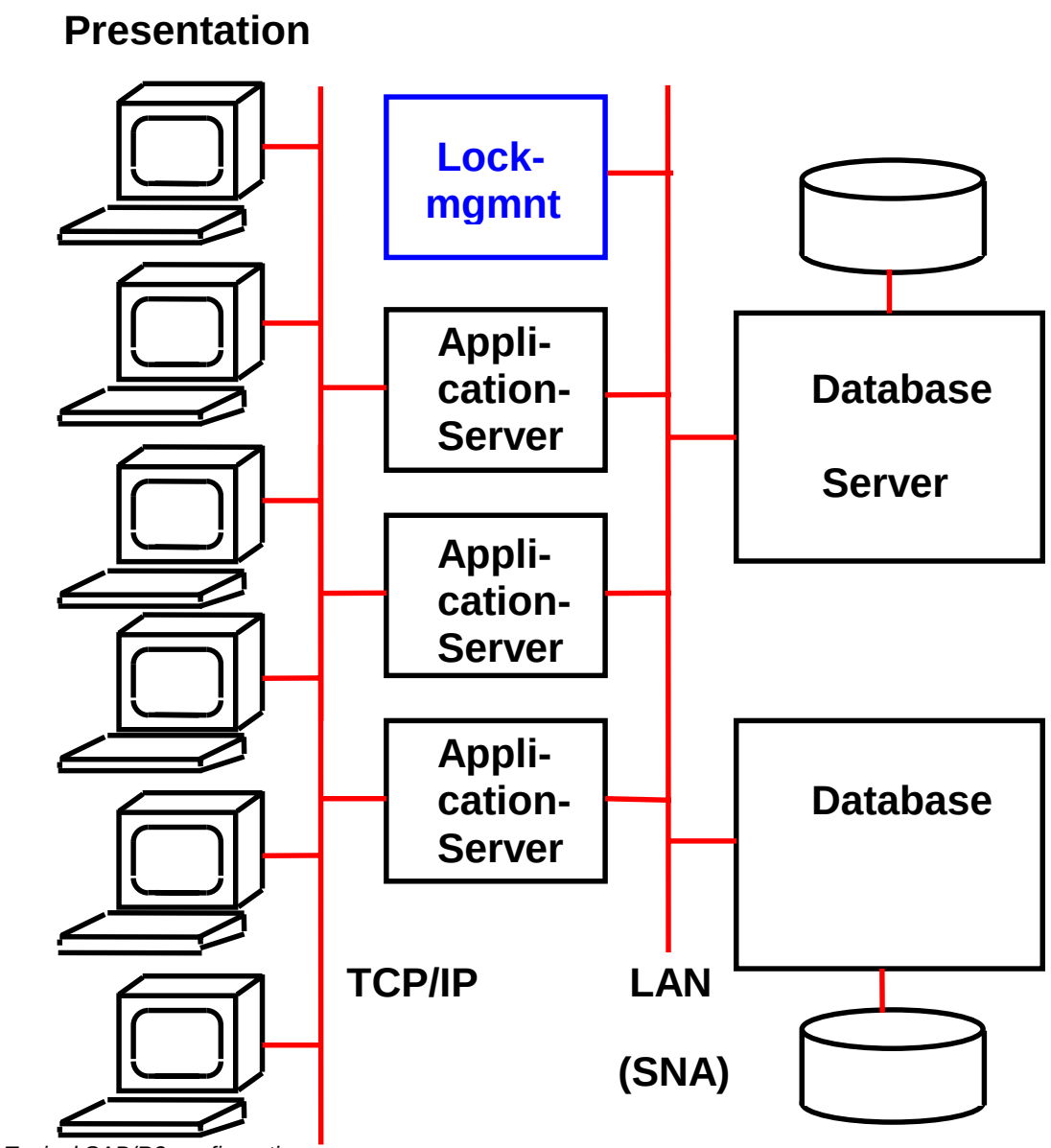
Nur System A besitzt Write Lock. B, C und D besitzen nur Read Lock. Ein Invalidate Broadcast an alle Systeme des Clusters benachrichtigt B, C und D dass die Kopie des Kocks nicht mehr gültig ist.

Beispiele für diese Lösung sind die VAX DBMS und VAX Rdb/VMS Datenbanksysteme.

Eine Faustregel besagt: You cannot build a cluster that scales, if you do not solve the locking problem .
Jim Gray, Andreas Reuter, 1993



Invalidate-Broadcast Kohärenzsteuerung



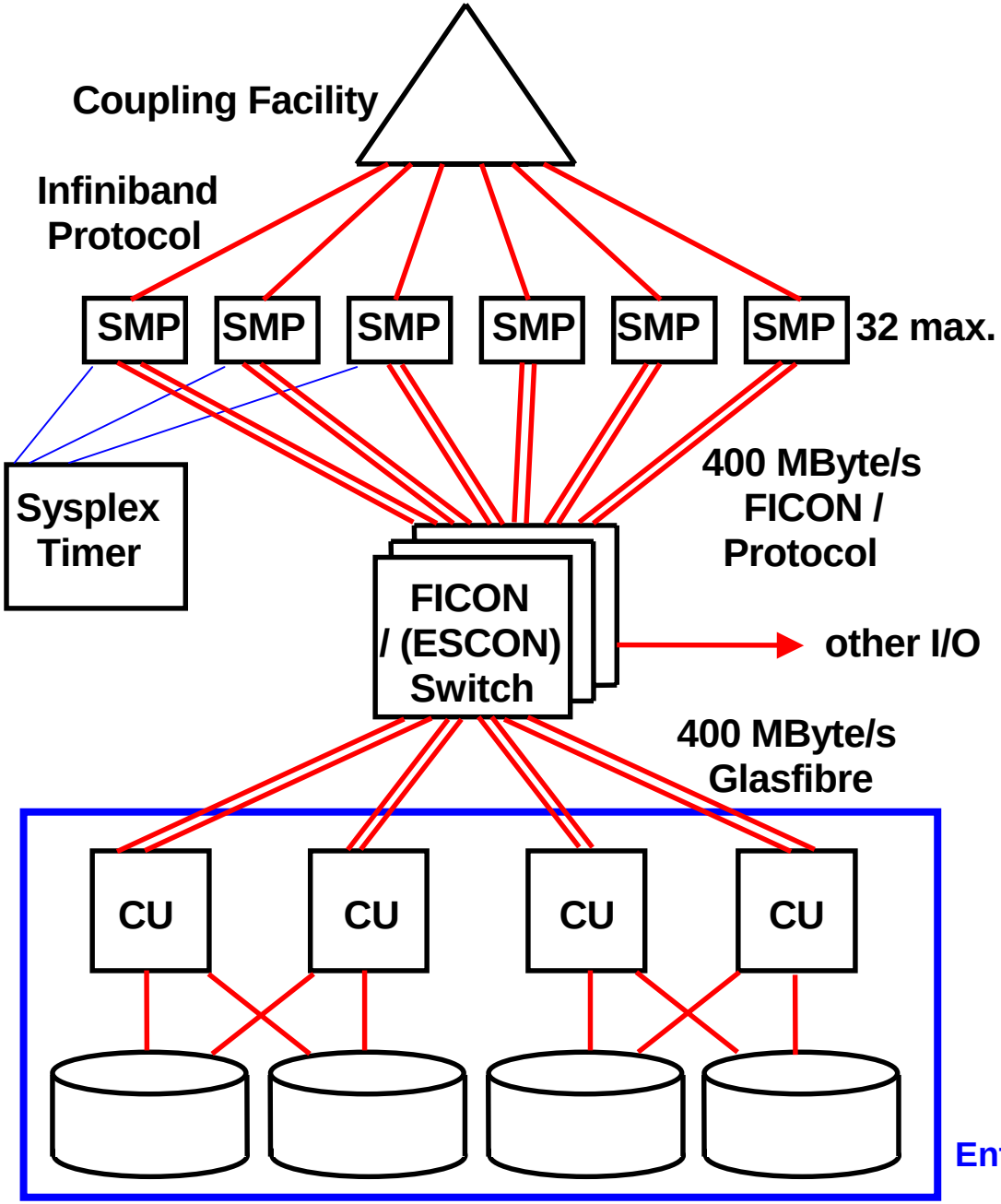
Typical SAP/R3 configuration

Dezentrale Locks arbeiten mit einer Invalidate-Broadcast Kohärenzsteuerung, die bei großen Transaktionsraten schlecht skaliert. Dieses Problem kann mit Hilfe einers zentralisierten Lockverwaltung und einem zentralen Sperrverwaltungsserver adressiert werden.

Eine typische SAP/R3 Konfiguration verwendet einen getrennten Server für das Lock management (Sperr-Server). Wenn ein Knoten ein S Lock erwerben möchte, wendet er sich an den Sperr-Server um zu erfragen, ob ein anderer Knoten ein Interesse an diesem Kock hat. Wenn ja, wird nur dieser benachrichtigt.

Nachteilig: Die Anfrage an den Sperr-Server geht über ein LAN mit dem entsprechenden TCP/IP Stack Overhead.

**“You cannot scale a transaction processing system,
if you do not solve the locking problem”
Jim Gray, Andreas Reuter, 1993**



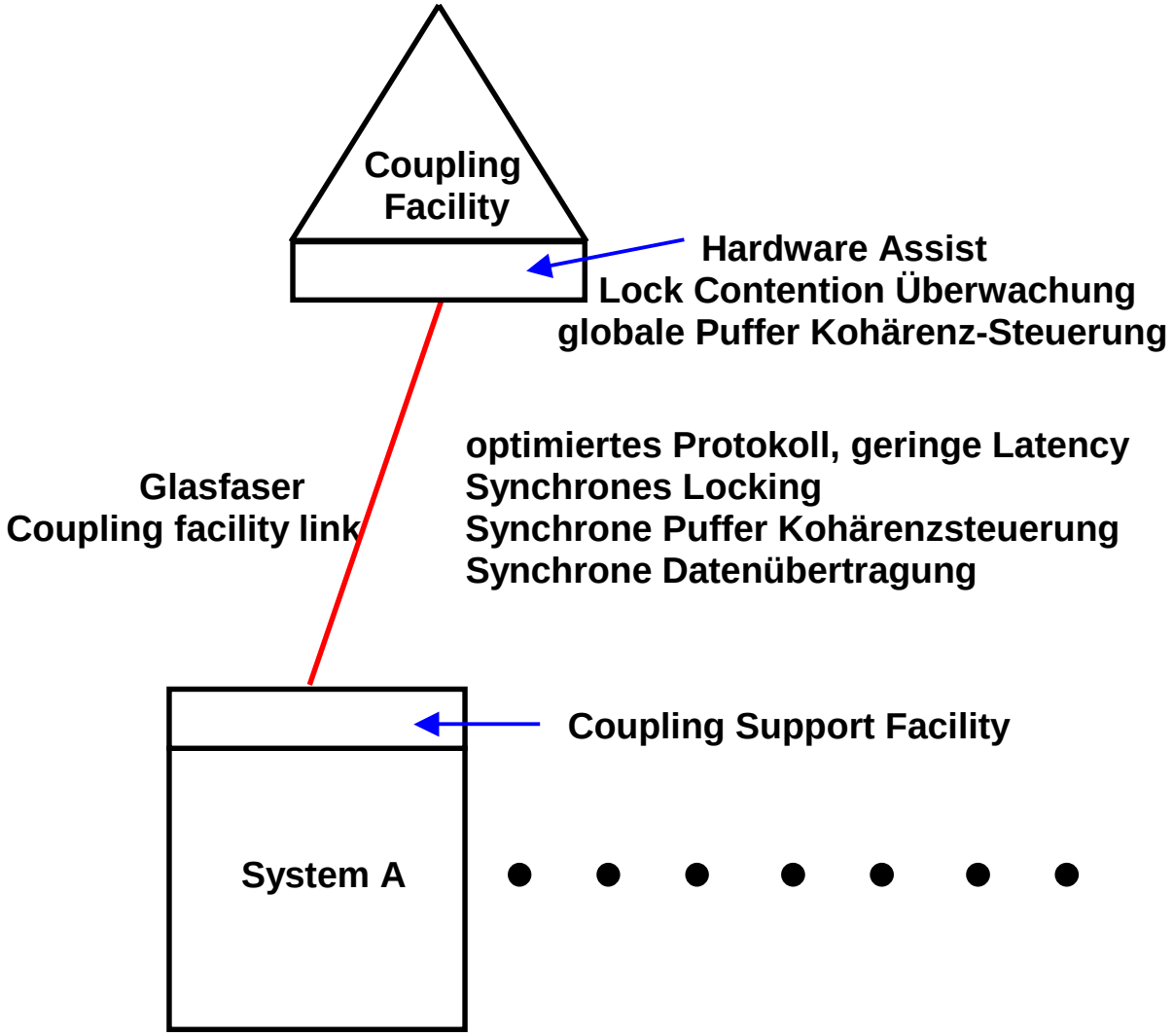
Sysplex with Coupling Facility

Ein Sysplex besteht aus bis zu 32 System z (oder S/390) Rechnern, die über FICON Glasfasern und FICON Switche mit Plattenspeichern in der Form von einem oder mehreren Enterprise Storage Servern verbunden sind. Ein Enterprise Storage Server emuliert mehrere Control Units (CU).

Weiterhin sind 1 oder in der Regel 2 Coupling Facilities vorhanden.

Eine Coupling Facility ist ein regulärer System z oder S/390 Rechner, auf dem „Coupling Facility Code“ an Stelle eines Betriebssystems läuft.

Weiterhin sind 1 oder 2 Zeitgeber (Sysplex Timer) vorhanden

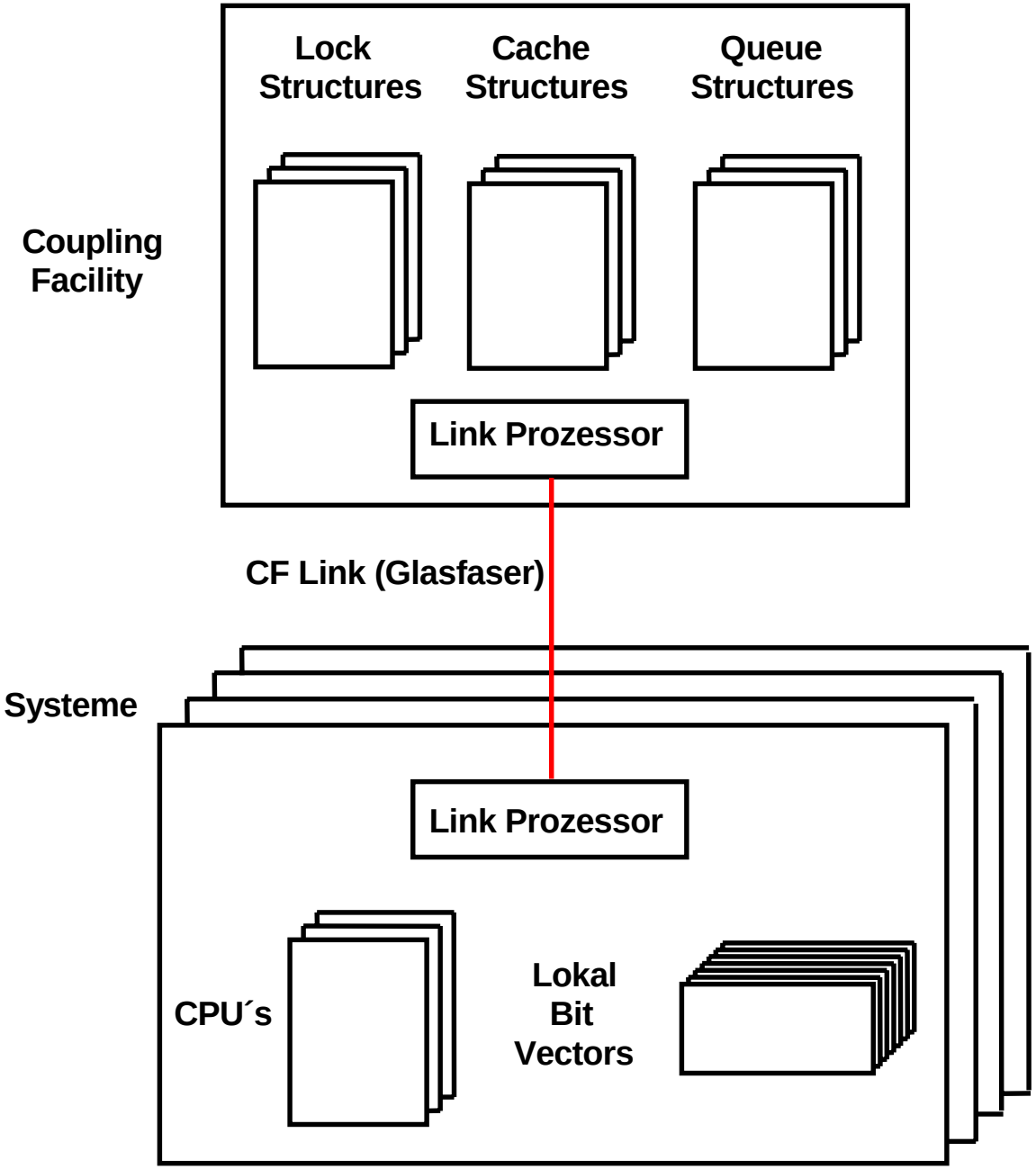


Die Coupling Facility ist durch eine Punkt-zu-Punkt Glasfaserleitung mit jedem System (Knoten) des Sysplex verbunfen (Coupling facility link).

Es wird ein spezielles Verbindung-protokoll mit besonders geringer Latency eingesesetzt .

Die CF Glasfaser Verbindung wird durch spezielle Hardware Einrichtungen und durch zusätzliche Maschinenbefehle in jedem angeschlossenen System unterstützt. Die Coupling Facility kann in einem angeschlossenen Rechner ohne Unterbrechung des laufenden Prozesses Daten in spezielle Speicherbereichen (Bit Vektoren) abändern.

Anbindung eines Systems an die Coupling Facility



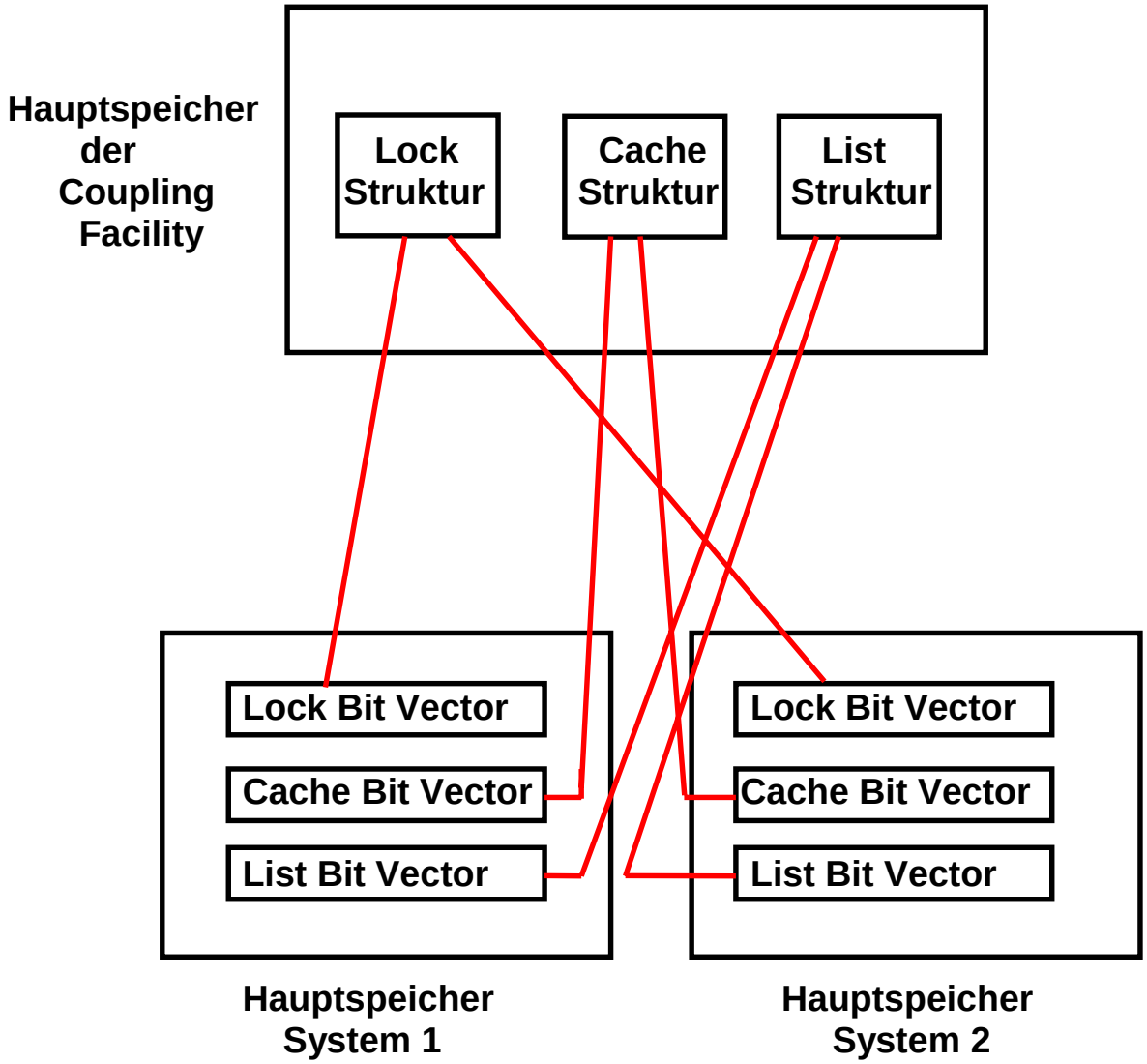
Die Coupling Facility unterhält in ihrem Hauptspeicher getrennte Strukturen für die Verwaltung von

- **Locks (Sperren)**
- **Sysplex weiter Daten Cache**
- **Listen für Sysplex-weite Aufgaben.**

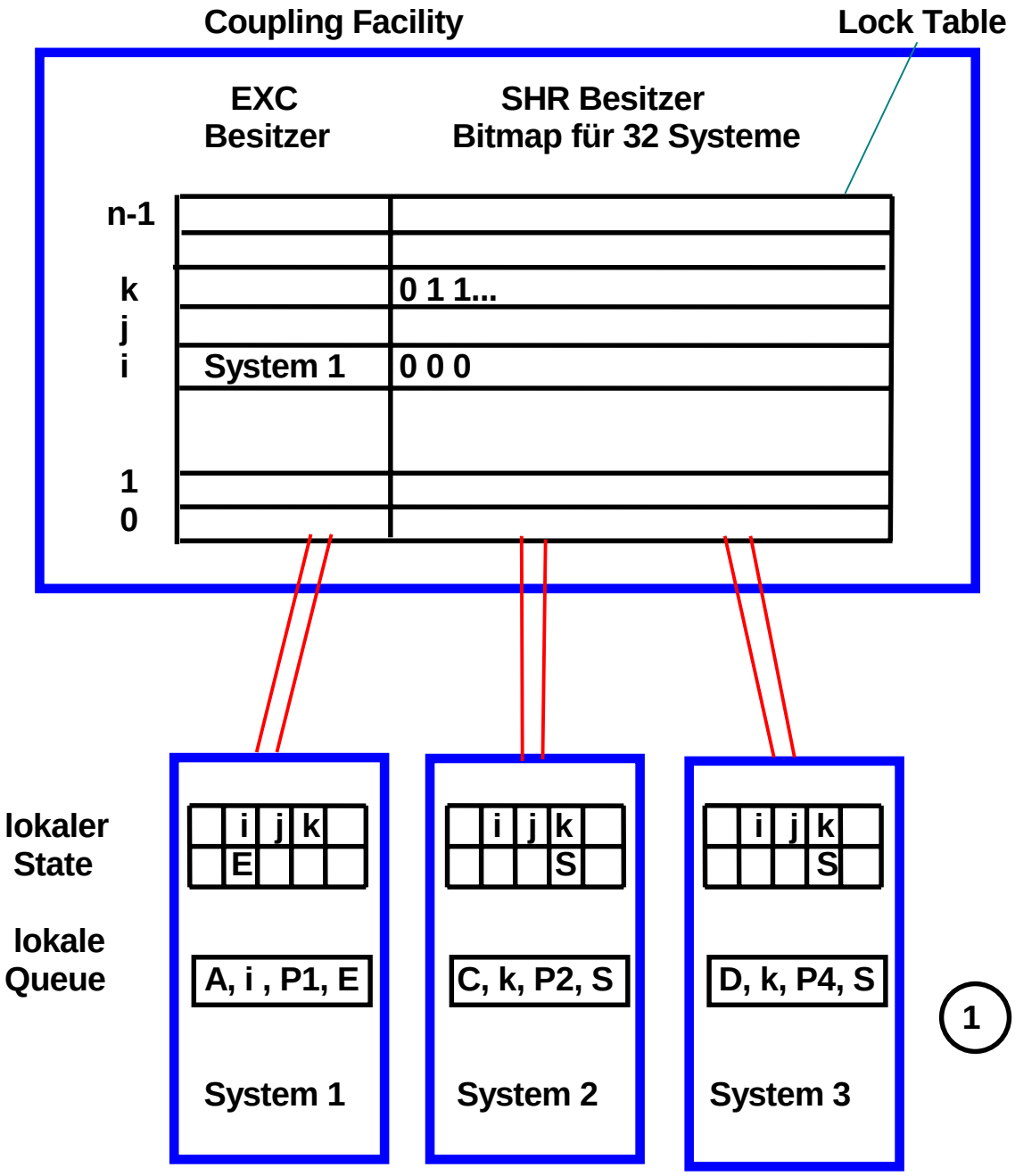
Die einzelnen Systeme des Sysplex sind mit jeder dieser Strukturen über das CF Link und die Link Prozessoren direkt verbunden.

Jedes System unterhält je 1 lokalen „Bit Vector“ für eine logische Verbindung zu jeder Struktur in der CF.

Für die Kommunikation CPU – CF sind spezifische Maschinenbefehle und zusätzliche Link Prozessor vorhanden.



Zuordnung von Bit Vektoren zu CF Strukturen



Nutzung der CF Lock Tabelle

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

Frei	0
Shared	S = SHR
Exclusive	E = EXC

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.

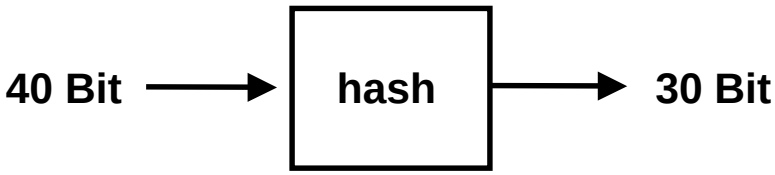
2. **Sowohl System 2 als auch System 3 wünschen für ihre jeweiligen Prozesse P2 und P4 Shared Rechte für Locks C und D, die beide in die Hash Klasse k fallen . Die CF registriert dies in der Bitmap für k und erteilt die Rechte.**

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

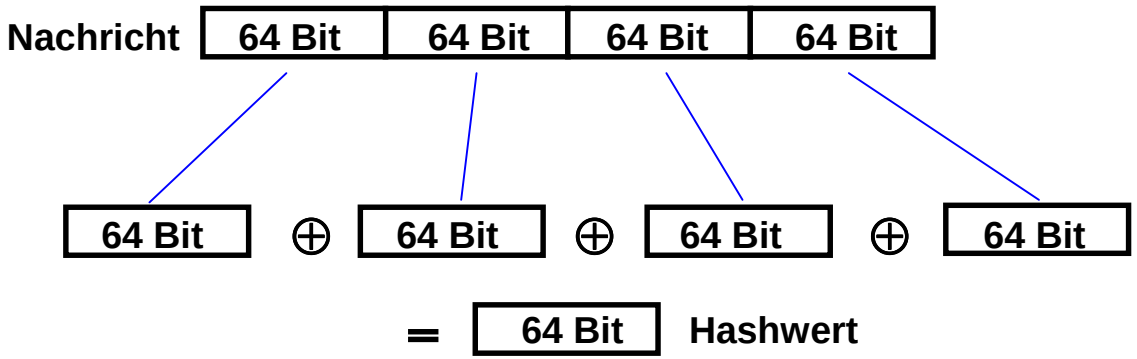
Lock Tabelle mit 10^{9} Einträgen à 8 Byte
8 GByte



IMS Lock Namen = 19 Byte = 152 Bit

Hashing

Ein einfaches Beispiel:



Beispiel: Nachricht (hier 256 Bit) in Teile gleicher Länge (hier 64 Bit) zerlegen
Teile mit Exclusive Oder verknüpfen. Ergebnis ist ein 64 Bit Hash Wert.

Problem: Es kann sein, dass zwei unterschiedliche Nachrichten den gleichen Hash Wert ergeben (Hash Synonym oder Hash Konflikt). Das Arbeiten mit Hash Werten braucht ein Verfahren, um Hash Konflikte aufzulösen.

Erwünscht: Der Hash Algorithmus (hier Exclusive Oder Verknüpfung) soll sicherstellen, dass für beliebige Nachrichten alle Bitmuster des Hashwertes mit gleicher Wahrscheinlichkeit vorkommen.

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 Konflikten

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

Beispiel:

- **10 000 Transaktionen / s**
- **0,5 s Antwortzeit**
- **Multiprogramming Level = 5 000**
- **20 Locks / aktive Transaktion**
- **100 000 aktive Locks**
- **Ziel: Falsche Konflikte ≤ 0,5 %**
- **Lock Tabelle mit 20 000 000 Einträgen**

(Little´s Gesetz:
Ankunftrate x Antwortzeit = Multiprogramming Level)

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).

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.

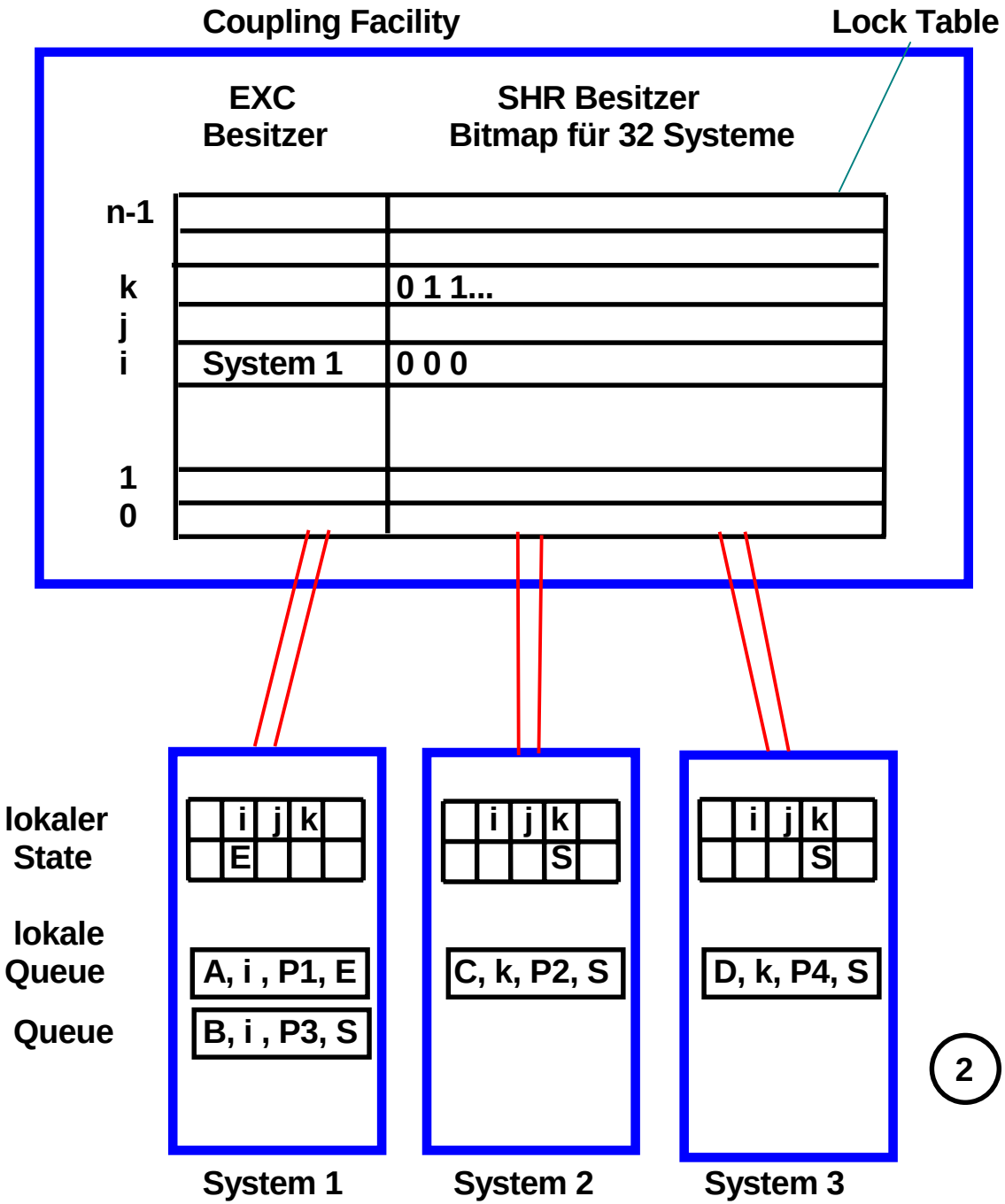
4. **Prozess P1 in System 1 möchte EXC Rechte für ein Lock in der Hash Klasse i erhalten. Anfrage an CF. Da niemand sonst Interesse hat, wird dem Request entsprochen. Im lokalen State Vektor von System 1 wird diese Berechtigung festgehalten.**

   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.

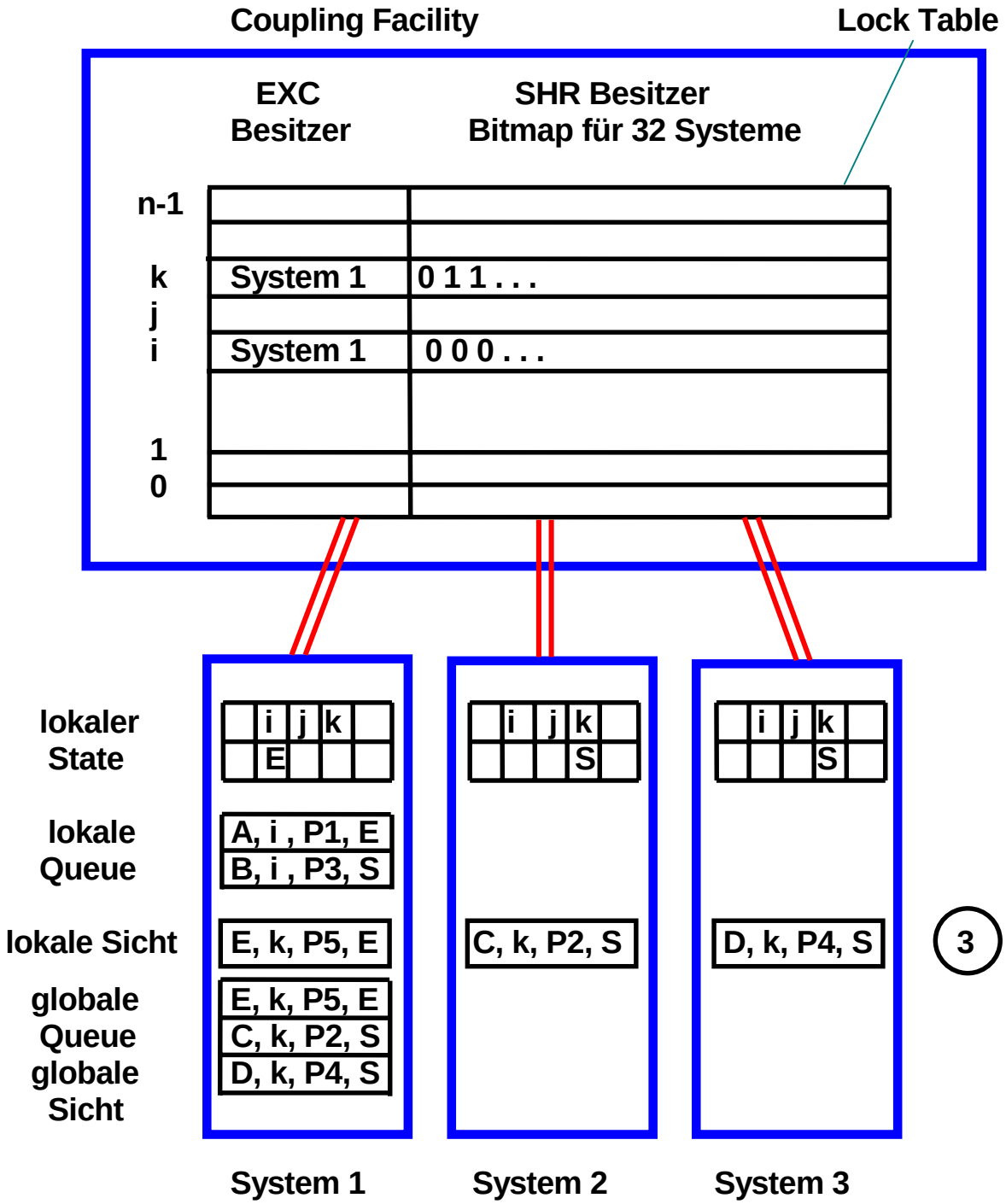
5. **Sowohl System 2 als auch System 3 wünschen für ihre jeweiligen Prozesse P2 und P4 Shared Rechte für Locks C und D, die beide in die Hash Klasse k fallen . Die CF registriert dies in der Bitmap für k und erteilt die Rechte.**

6. **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.**



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

2



System 1 fordert exclusives Lock E in Hash Klasse k an. Kein Konflikt 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.

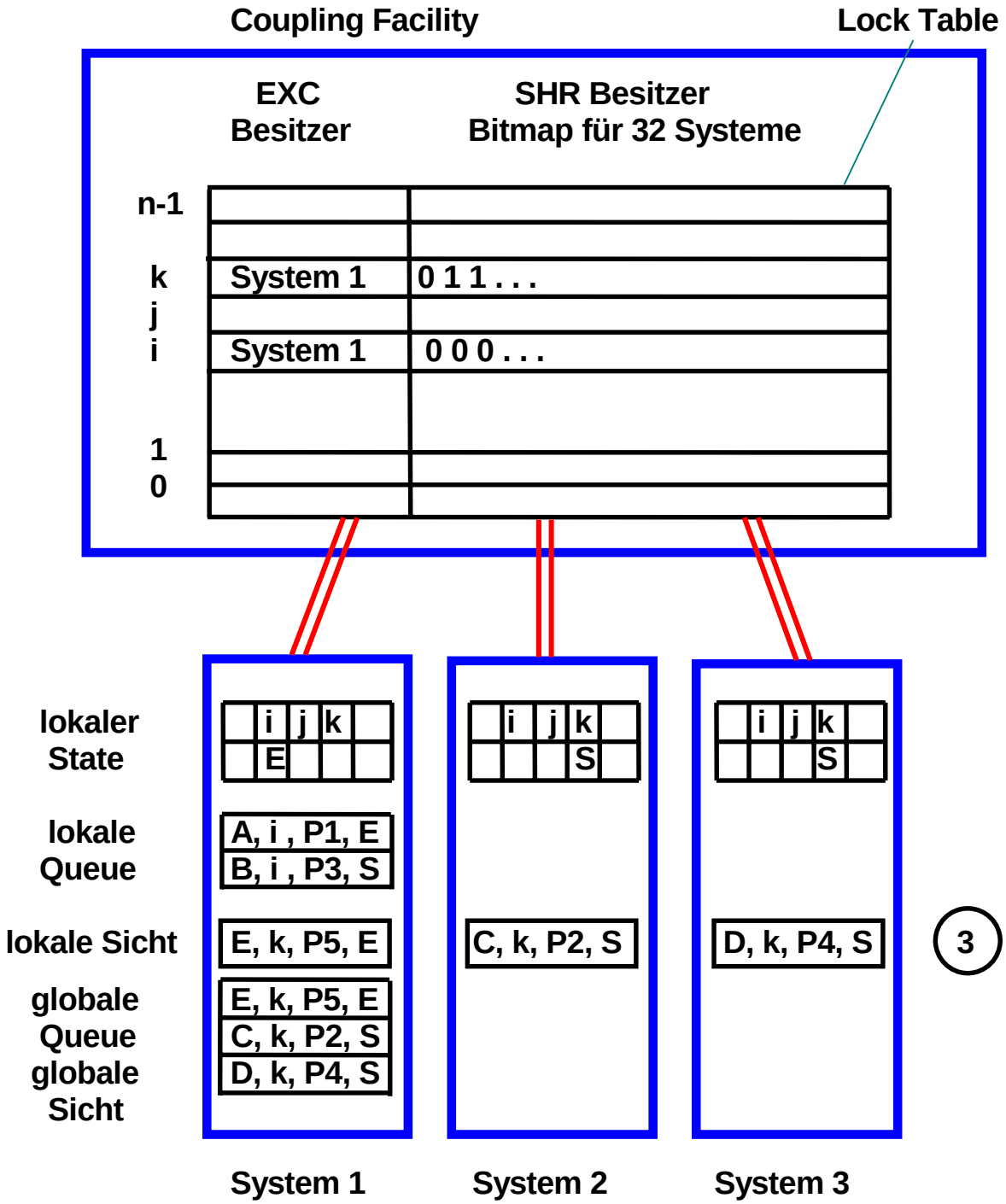
System 1 übernimmt die globale Management Verantwortung für Klasse k. Es erfrägt von Systemen 2 und 3 deren Lock Information für Klasse k. Erfolgt parallel über Hochgeschwindigkeitsverbindungen.

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

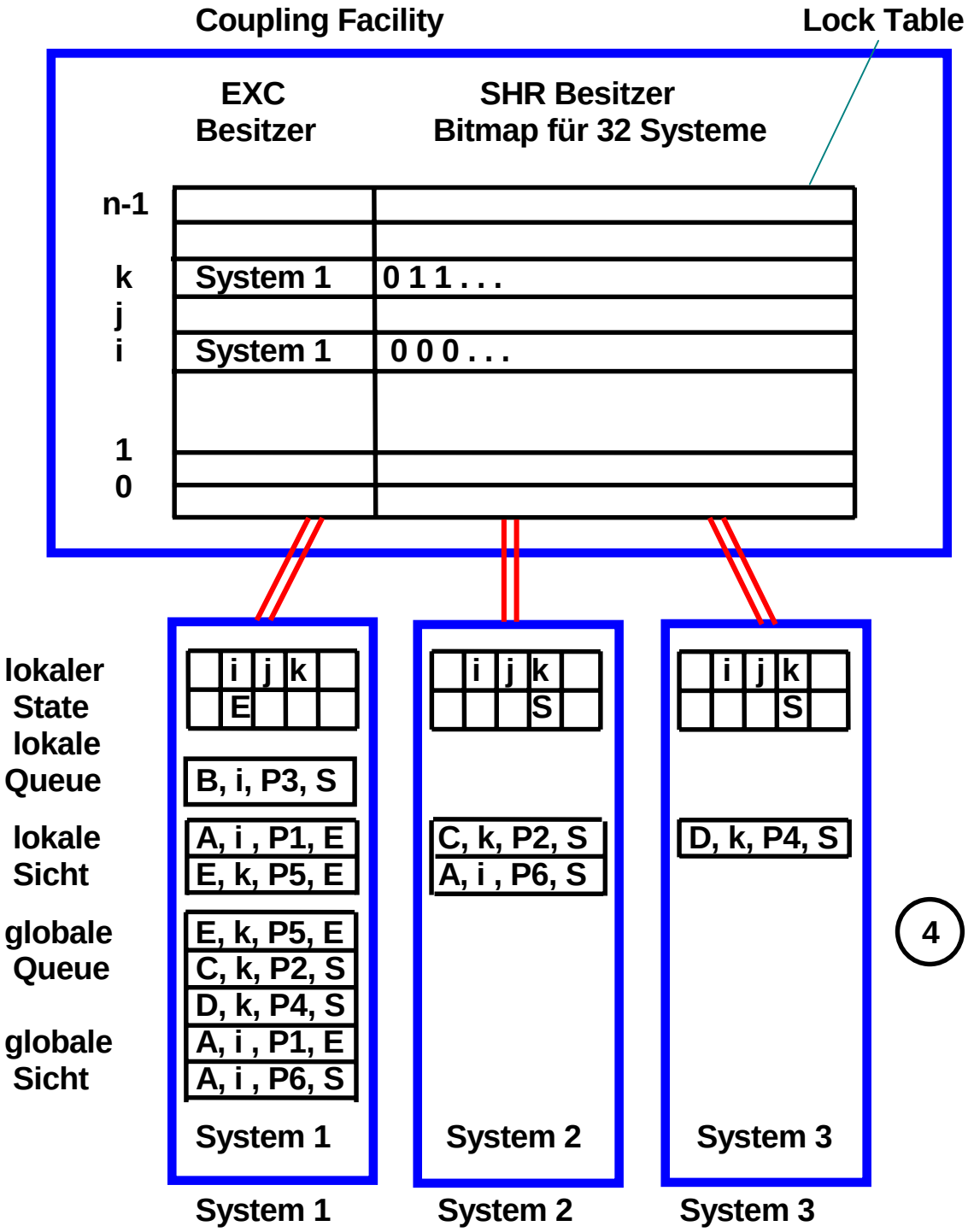
Annahme: Kein Konflickt mit den anderen Locks in Klasse k.

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



System 2 fordert shared Lock A in Hash Klasse i an. Echter Konflikt mit den anderen Locks in Klasse i. System 1 übernimmt die Auflösung des Konfliktes.

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.

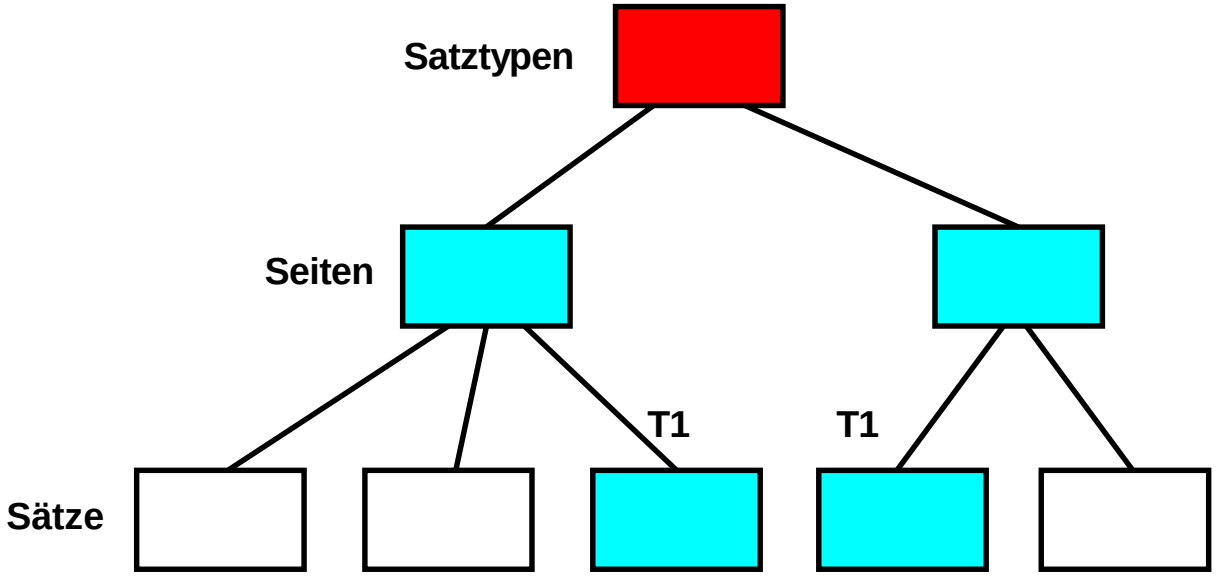
Eine Seite enthält eine Anzahl Datensätze (Records).

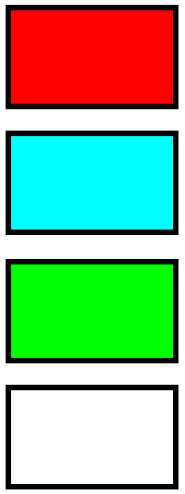
Ein feines Granulat (z.B. Satzsperren) führt zu einer geringen Anzahl von Konflikten zwischen Transaktionen und zu einem hohen Verwaltungsaufwand.

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





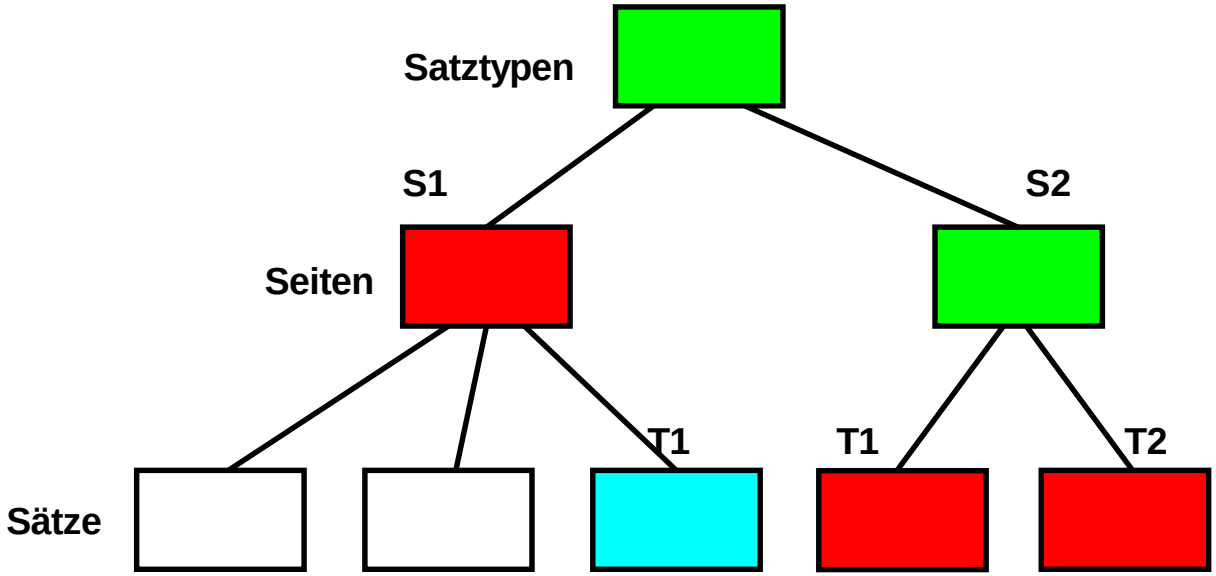
expliziert gesperrt

impliziert gesperrt

Anwartschafts-sperre

nicht referenziert

Satztyp vollständig von Transaktion 1 gesperrt



Situation nachdem Transaktion 2 einen Satz referenziert

In dem gezeigten Beispiel beinhaltet jeder Satztyp mehrere Seiten und jede Seite mehrere Datensätze.

Ursprünglich ist der gezeigte Satztyp ganz zu Gunsten von Transaktion 1 gesperrt. Nach Auflösung des Konflikts erhält der Satztyp eine Anwartschafts-sperre, d.h., mehrere seiner Elemente werden von unterschiedlichen Transaktionen benutzt. Seite S1 ist zu Gunsten von Transaktion 1 explizit gesperrt, Seite S2 erhält ebenfalls eine Anwartschaftssperre.

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.

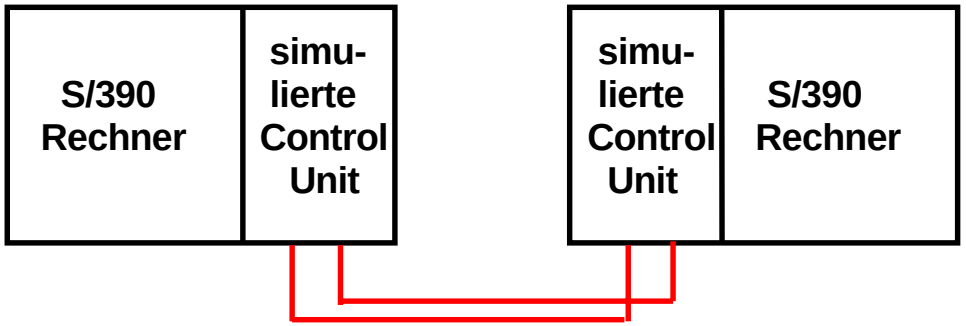
Wird ein Globales Lock angefordert, überprüft der lokale Lock Manager, ob es lokal, ohne Zugriff auf die CF, vergeben werden kann.
Hierzu dient das EHL Verfahren (Explicit Hierarchical Locking). In der Mehrzahl der Fälle ist der Zugriff auf die CF nicht erforderlich.

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.

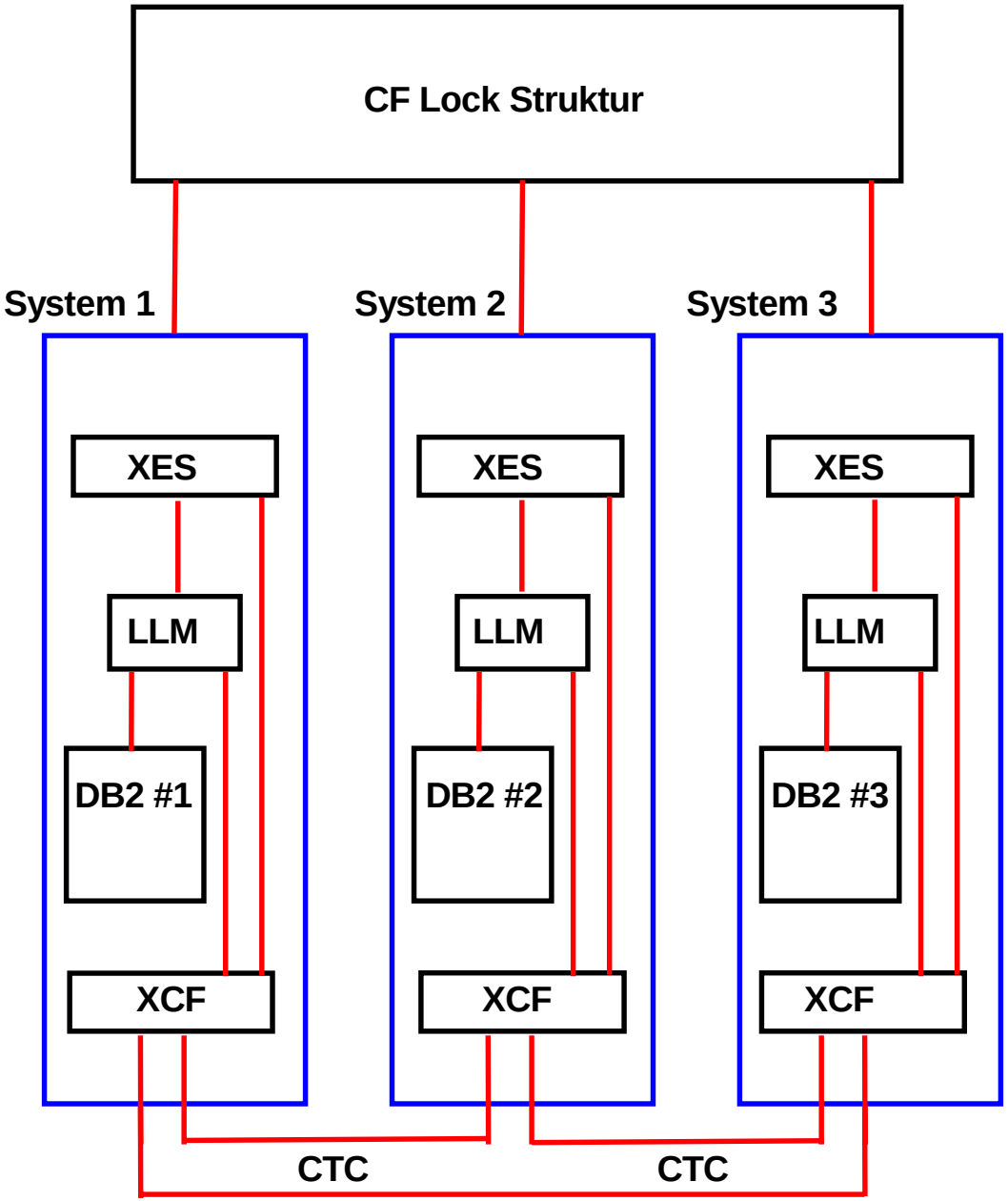
Channel- to Channel Verbindung (CTC)
Cross-System Coupling Facility (XCF)



Die Channel- to Channel Verbindung wird durch eine Hardware Einrichtung eines zSeries Systems verwirklicht, die dieses System gegenüber einem anderen zSeries System wie eine E/A Einheit erscheinen lässt.

Für eine Full Duplex Verbindung werden normalerweise zwei CTC Verbindungen eingesetzt.

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

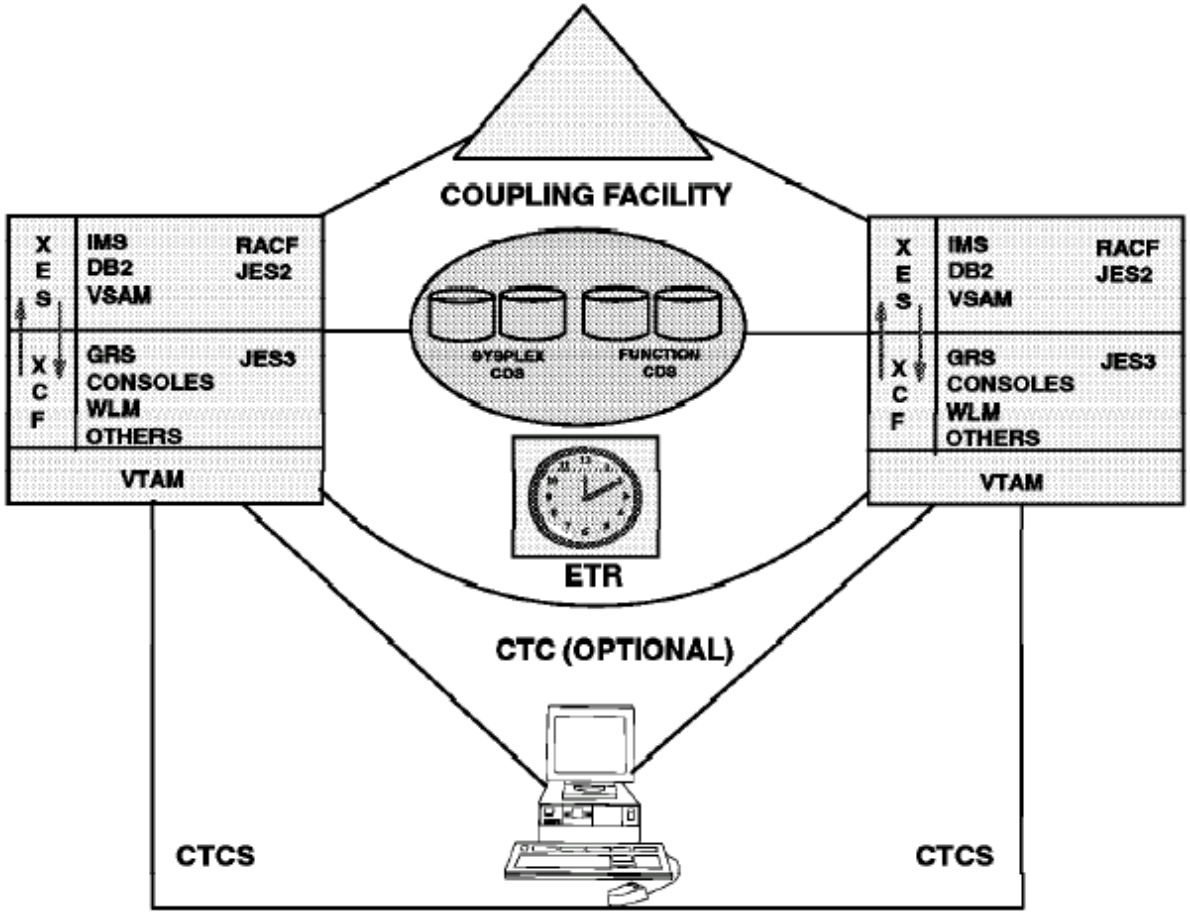


DB2 Global Locking

DB2 Global Locking uses these facilities:

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

Cross-system extended services (XES) is the z/OS component that enables communications with the coupling facility. XES is an extension to the XCF component.



XES provides the support for connection to a coupling facility. XES provides the external z/OS and OS/390 authorized programming interface and services that allow the subsystems and system components to gain access to coupling facility structures and to access and manipulate the data in each of these structures.

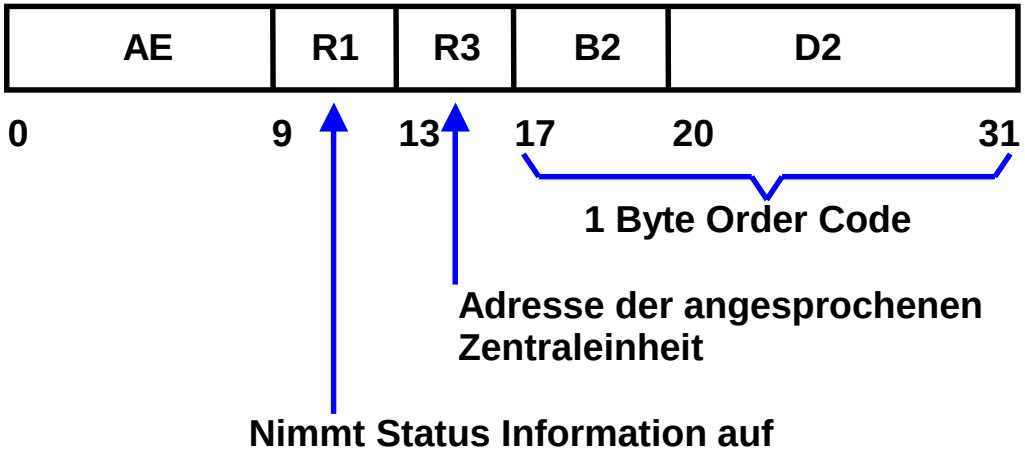
XES provides:

I. The interfaces to define protocols that allow caching of data to be shared

II. Allowances for authorized application or system component to define and manipulate lists of data items

III. Shared and exclusive locking support to an authorized application or system component to allow for user-defined protocols and contention management

Signal Processor Maschinenbefehl



An eight-bit order code and, if called for, a 32-bit parameter are transmitted to the CPU designated by the CPU address contained in the third operand. The result is indicated by the condition code and may be detailed by status assembled in bit positions 32-63 of the first-operand location.

The second-operand address is not used to address data; instead, bits 56-63 of the address contain the eight-bit order code. Bits 0-55 of the second-operand address are ignored. The order code specifies the function to be performed by the addressed CPU.

The 16-bit binary number contained in bit positions 48-63 of general register R3 forms the CPU address. Bits 0-47 of the register are ignored.

Synchrone CF Operation

Zusätzlich Elongatoren

Distanz **12 µs / km**

Link Protokoll

optimiert
Punkt zu Punkt
Kein Handshake, nur Send - Receive

CF Latency **wenige µs**

Datenübertragung

4 Kbyte, 100 Mbyte / s Link
40 µs

Pfadlänge für Verarbeitung innerhalb der Coupling Facility

Hardware / Software Tradeoff
Geschwindigkeit versus Flexibilität

Synchrone Locking Operation

4000 CF Maschinenbefehle
20 µs für 200 MIPS CF Prozessor

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 Konflikten

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

Beispiel:

- **10 000 Transaktionen / s**
- **0,5 s Antwortzeit**
- **Multiprogramming Level = 5 000**
- **20 Locks / aktive Transaktion**
- **100 000 aktive Locks**
- **Ziel: Falsche Konflikte ≤ 0,5 %**
- **Lock Tabelle mit 20 000 000 Einträgen**

(Little´s Gesetz:
Ankunftrate x Antwortzeit = Multiprogramming Level)

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).

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.

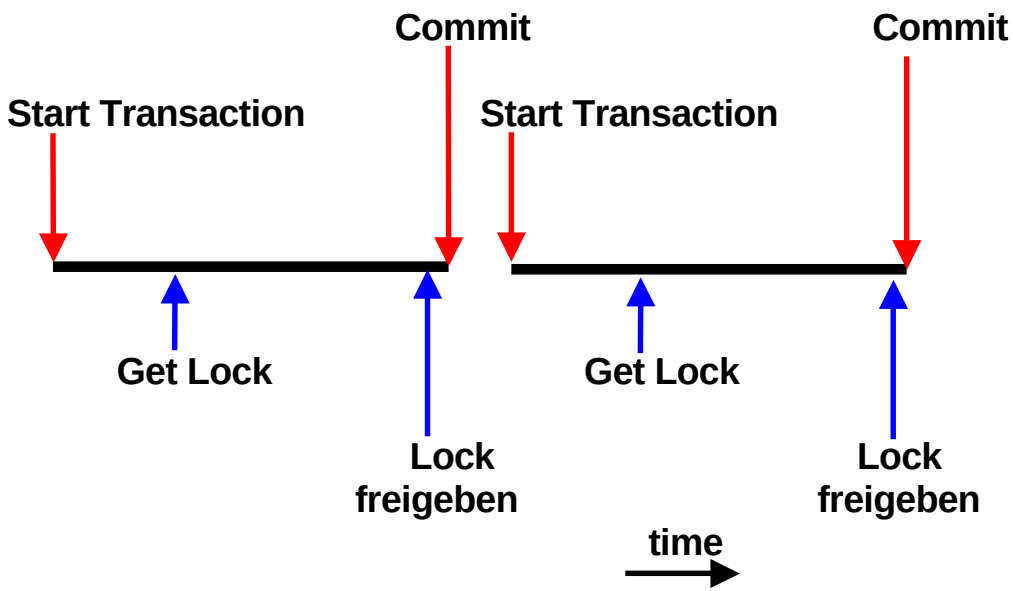
7. **Prozess P1 in System 1 möchte EXC Rechte für ein Lock in der Hash Klasse i erhalten. Anfrage an CF. Da niemand sonst Interesse hat, wird dem Request entsprochen. Im lokalen State Vektor von System 1 wird diese Berechtigung festgehalten.**

   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.

8. **Sowohl System 2 als auch System 3 wünschen für ihre jeweiligen Prozesse P2 und P4 Shared Rechte für Locks C und D, die beide in die Hash Klasse k fallen . Die CF registriert dies in der Bitmap für k und erteilt die Rechte.**

9. **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.**



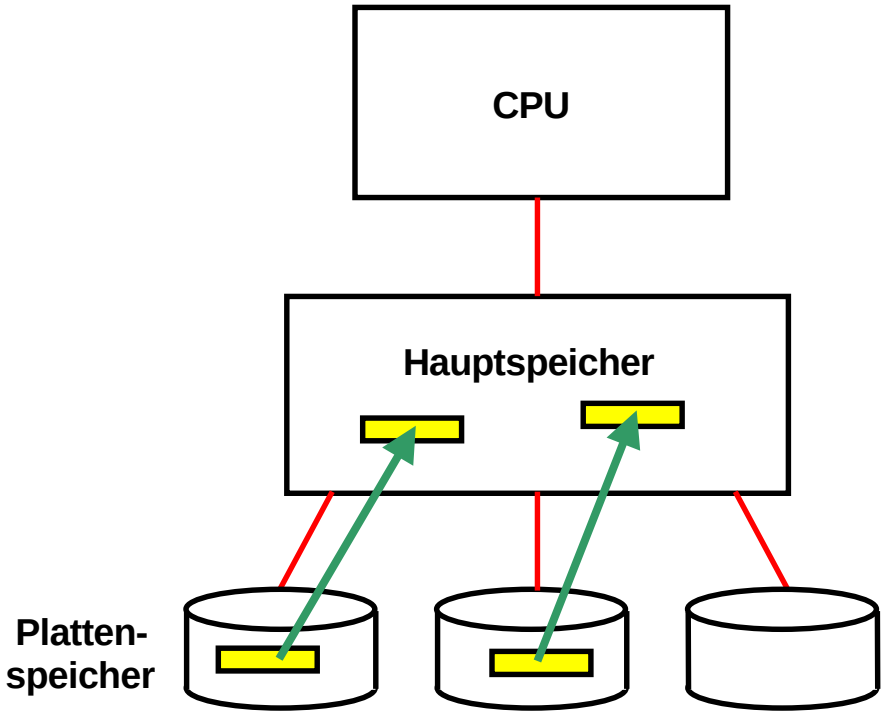
„Eager“ und „Lazy“ Locking Protokolle

Eager Protokoll Lock freigeben wenn Commit Transaction

Lazy Protokoll Lock freigeben wenn Contention

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

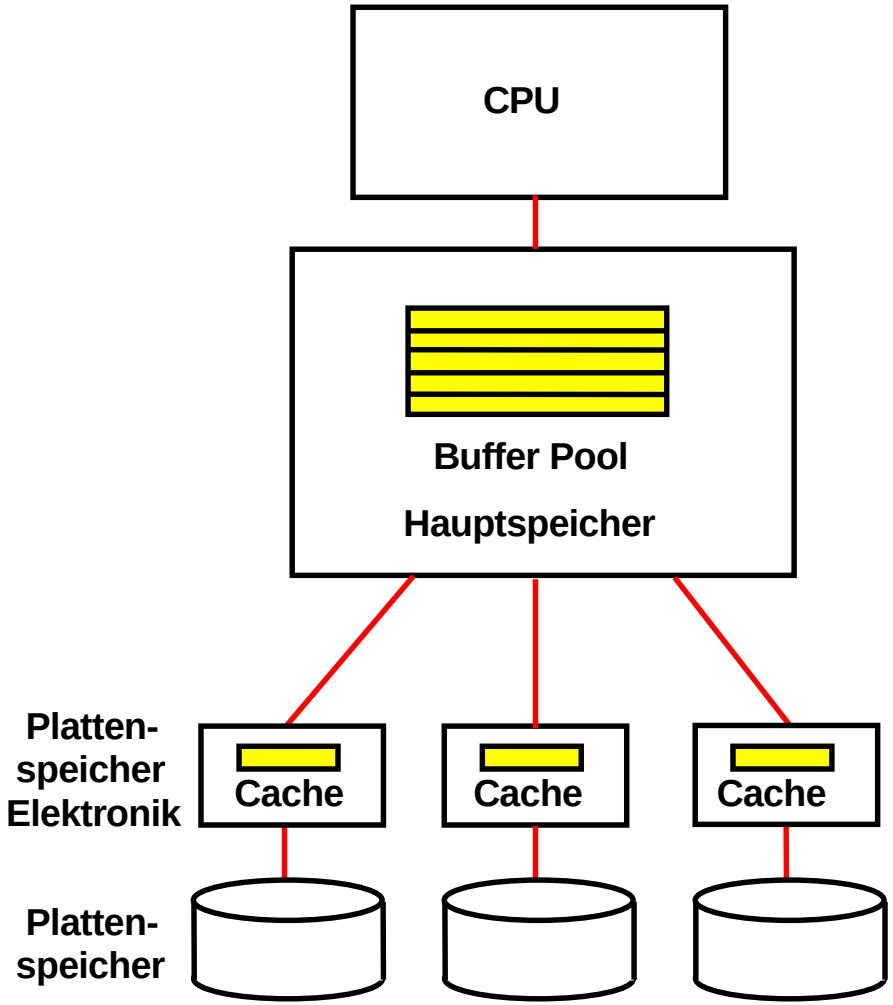
Sysplex Coupling Facility verwendet das Eager Protokoll (auch als „force-at-commit“ bezeichnet). Sharing tritt häufig auf, wenn existierende Anwendungen auf den Sysplex portiert werden.



Ein/Ausgabepuffer
I/O Buffer

In der Vergangenheit hat ein Anwendungsprogramm jeweils einzelne Datensätze (Records) vom Plattenspeicher in einen Ein/Ausgabepuffer (I/O Buffer) im Hauptspeicher gelesen und dort verarbeitet. Zur Leistungssteigerung hat man bald mehrere logische Records zu einem physischen Record zusammengefasst. Mit etwas Glück findet das Anwendungsprogramm beim nächsten Zugriff die Daten bereits im Ein/Ausgabepuffer und ein Plattenspeicherzugriff erübrigt sich.

In der Regel wird mit mehreren Dateien oder Datenbanken gleichzeitig gearbeitet, die alle ihren eigenen Ein/Ausgabepuffer benutzen. Die Menge der Ein/Ausgabepuffer wird als „Buffer Pool“ bezeichnet und vom Betriebssystem optimal verwaltet.

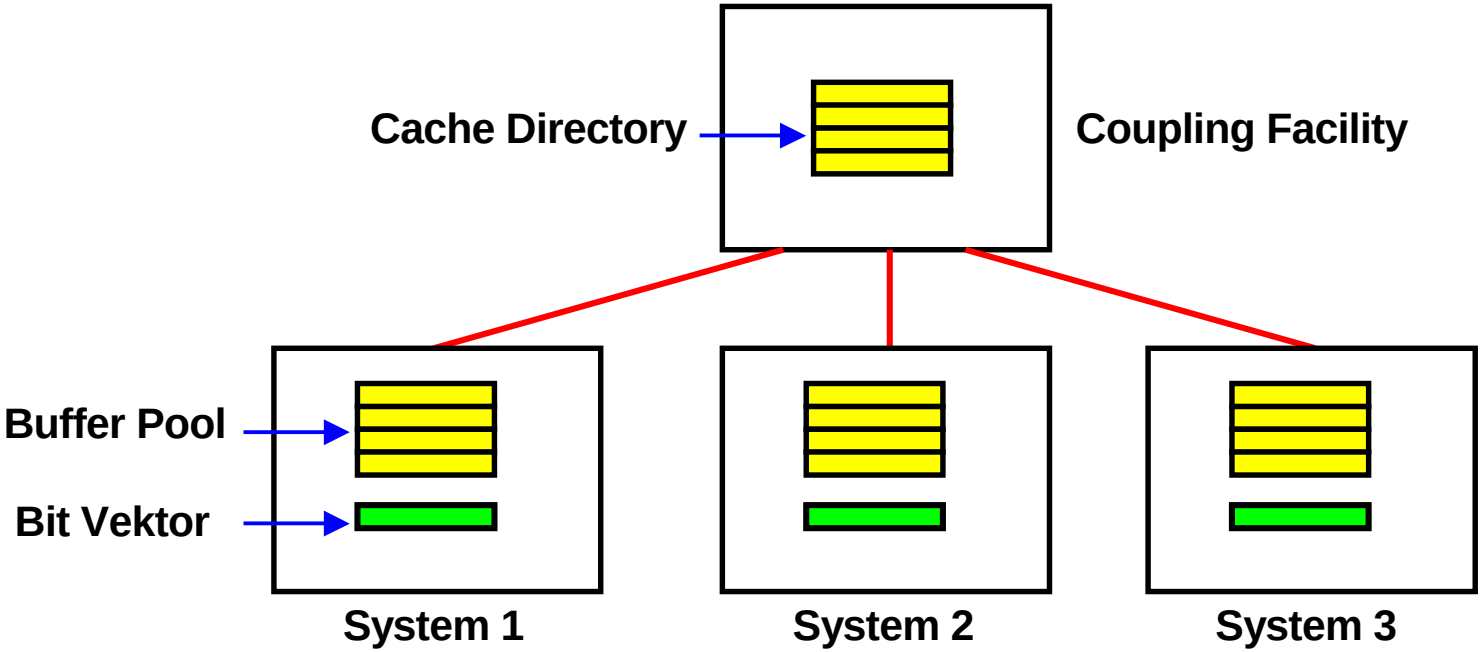


Plattenspeicher Cache und Hauptspeicher Buffer Pool

Der Buffer Pool stellt eine Art Plattenspeicher Cache im Hauptspeicher dar. Zusätzlich werden Daten in einem Plattenspeicher Cache gespeichert, der Bestandteil der Plattenspeicher Elektronik ist. Das Betriebssystem bemüht sich, den Speicherplatz im Buffer Pool und im Plattenspeicher Cache optimal zu verwalten.

Der Buffer Pool besteht aus einzelnen Puffern (Buffers), die Datenbankobjekte oder Teile einer Datei aufnehmen.

In einem Cluster ist nicht auszuschließen, dass Datensätze oder Datenbankrecords gleichzeitig in den Buffer Pools mehrerer Knoten (Systeme) abgespeichert werden.



Cache Directory in der Coupling Facility

Der Buffer Pool in jedem System enthält Blöcke (Buffer), die möglicherweise gerade bearbeitet werden. Es kann sein, dass sich in zwei unterschiedlichen Systemen Buffer mit den gleichen Datenbankrecords befinden.

Die Coupling Facility unterhält ein “Cache Directory”, in dem sich jeweils ein Eitrag für jeden Buffer in den angeschlossenen Systemen befinden. Analog zur Lock Verwaltung befinden sich außerdem in jedem System Bit Vektoren, die den Inhalt des Cache Directories teilweise replizieren.

Bei einer Änderung eines Eintrags im Cache Directory erfolgt ein automatisches Update der Bit Vektoren in allen angeschlossenen Systemen.

Coupling Facility Cache Directory

Der lokale Buffer Pool im System 1 enthält Puffer (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).

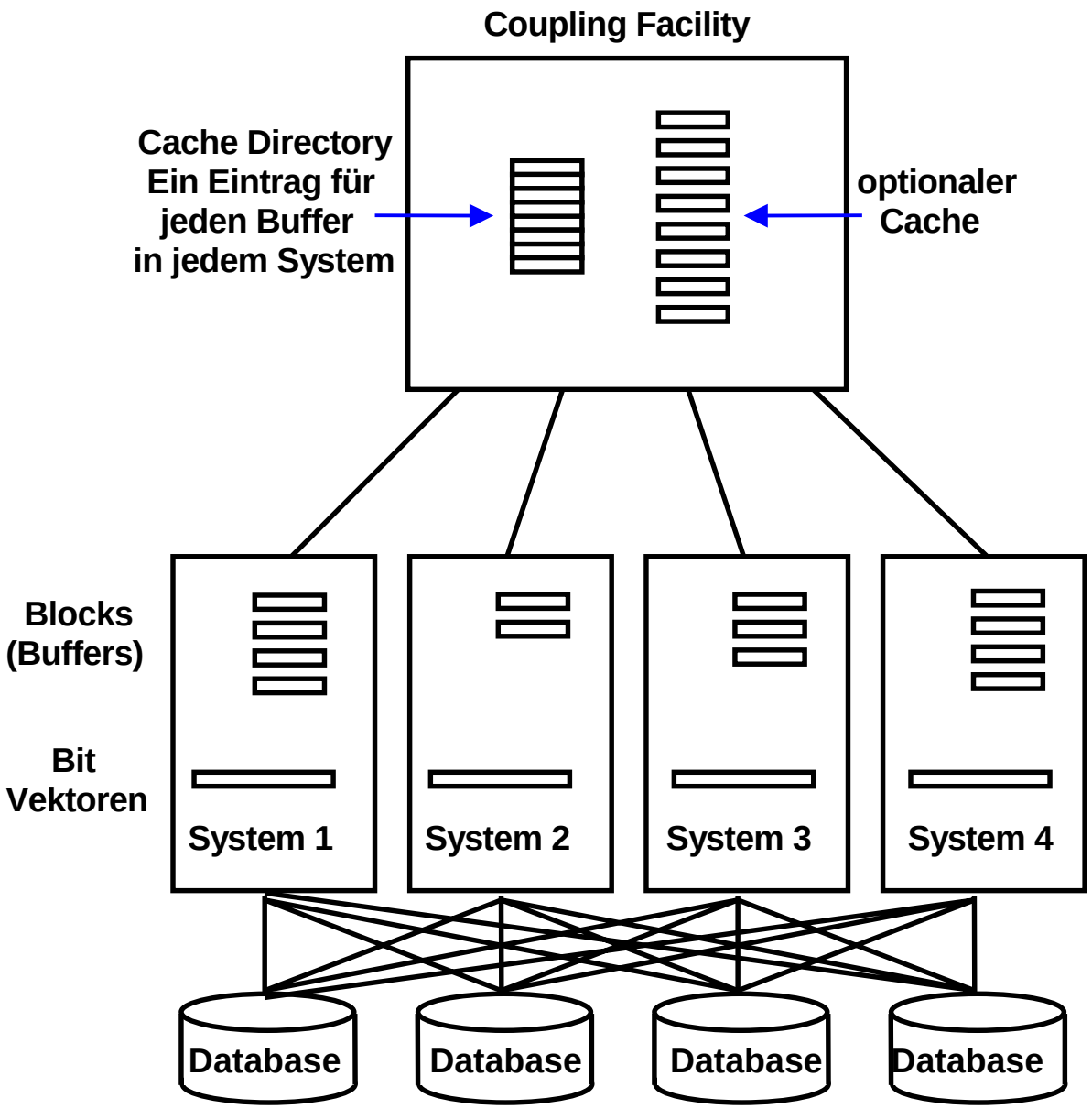
Wenn die Transaktion abgeschlossen ist (commit), werden die Locks freigegeben. Die Puffer bleiben in System 1 - evtl. werden sie demnächst wieder gebraucht.

Greift System 2 jetzt auf einen Buffer mit dem gleichen Datenbank Record zu, entsteht ein Kohärenzproblem. Die beiden Buffer in den Systemen 1 und 2 haben nicht den gleichen Inhalt.

Lösung: „Force-at-Commit“ . Bei Transaktionsabschluss erfolgt ein update des Cache Directories durch System 1.

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 !



Aller Datentransfer in 4 KByte Blöcken.

Das Cache Directory in der Coupling Facility enthält einen Eintrag für jeden Block (Buffer), der Teil eines Buffer Pools in einem der beteiligten Systeme ist.

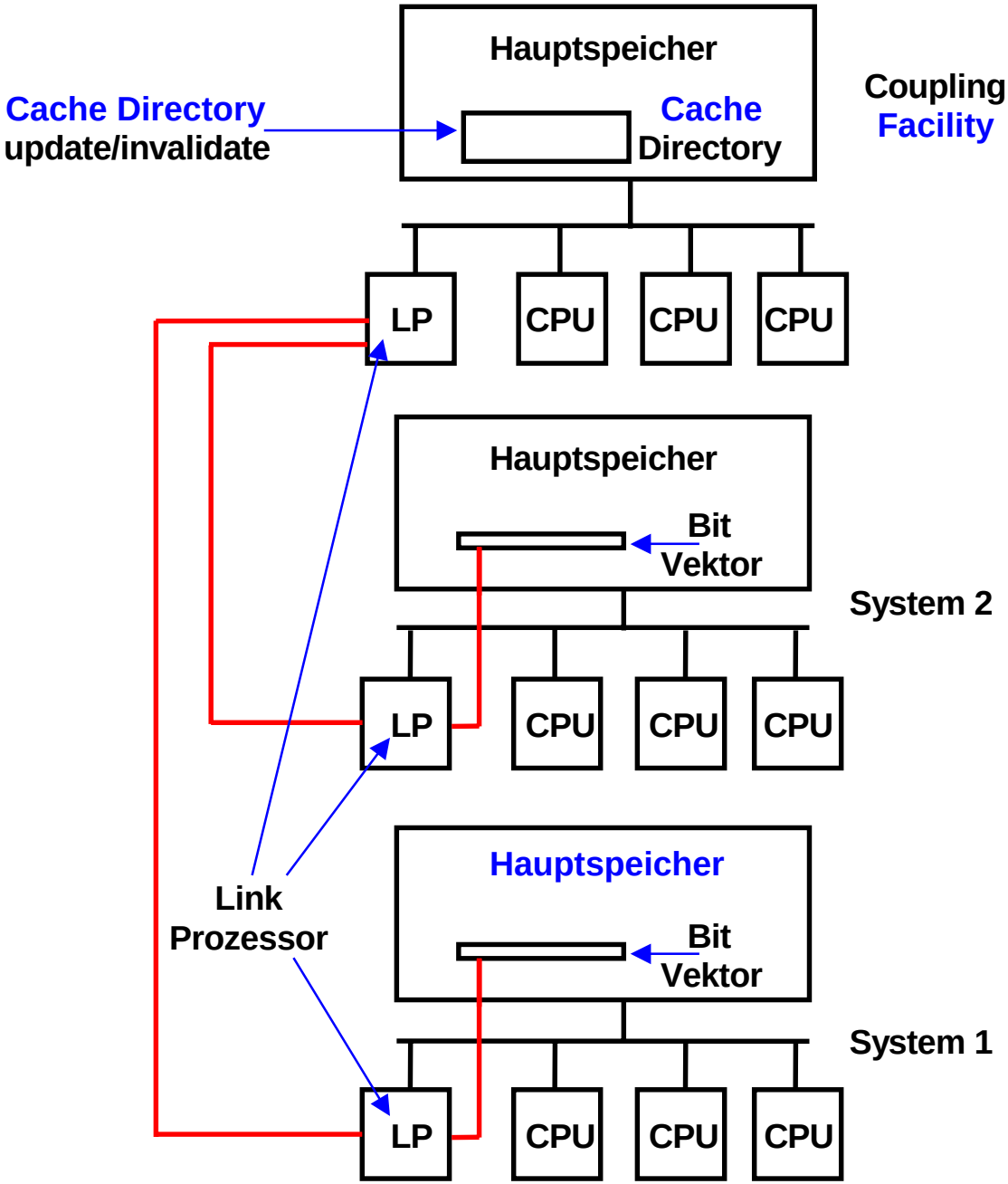
a) System 1 Read from Disk
1. Load Block from Disk
2. Register with CF Directory
3. add Bit in Bit Vector

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

einige Zeit später

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

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



Das Cache Directory in der Coupling Facility enthält einen Eintrag für jeden Block (Buffer), der Teil eines Buffer Pools in einem der beteiligten Systeme ist.

Bei Transaktionsabschluss erfolgt ein update des Cache Directories durch System 1 (Force-at-Commit).

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

Link Prozessoren haben einen Direct Memory Access auf den Hauptspeicher. Über die Link Prozessoren der Coupling Facility und die der Systeme können Bit Vektoren im Hauptspeicher abgeändert werden, ohne daß der normale Programmablauf dadurch beeinflußt wird (kein Prozesswechsel). Dies verbessert das Leistungsverhalten, da jeder Prozesswechsel eine Pfadlänge von mehreren Tausend Maschinenbefehlen erfordert.

Coupling Facility Cache.

Neben dem Cache Directory kann die Coupling Facility auch als Plattenspeicher-Cache genutzt werden. Hiervon machen einige, aber nicht alle Datenbanksysteme Gebrauch. DB2 und Adabas nutzen die Coupling Facility auch als Plattenspeicher-Cache.

DB2, IMS, Oracle und Adabas sind die wichtigsten unter z/OS eingesetzten Datenbanksysteme.

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**

CF List / Queue Strukturen

Neben dem Lock und dem Cache Management enthält die Coupling Facility Listen/Queue Strukturen, die vor allem für eine zentrale Verwaltung aller angeschlossenen Systeme eingesetzt werden.

Beispiele hierfür sind:

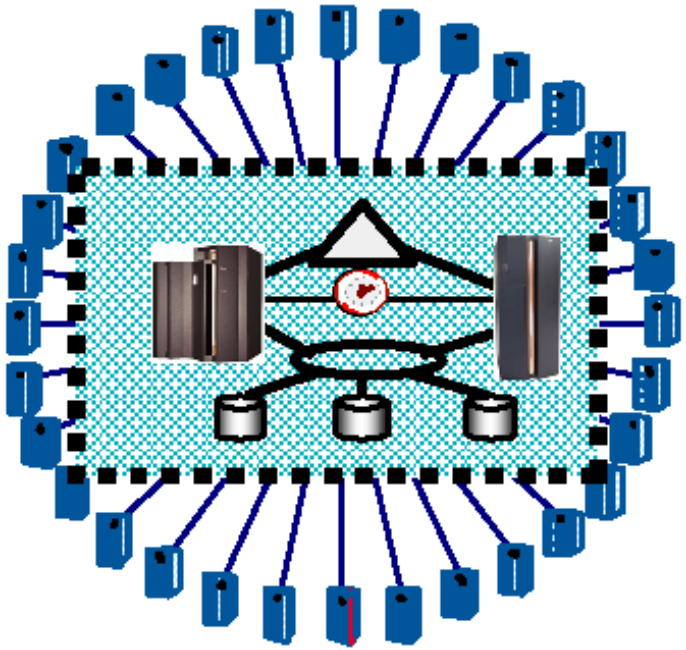
- **Clusterweite RACF Steuerung. Ein Sysplex Cluster besteht aus mehreren z/OS Instanzen. Im einfachsten Fall müsste sich ein Benutzer mit getrennten Passworten in jedes System einzeln einloggen. „Single Sign On“ ist eine Einrichtung, mit der der Benutzer mit einem einzigen Sign On Zugriffsrechte auf alle Resourcen eines Sysplex erhält. Die entsprechenden RACF Benutzerprofile werden von der Coupling Facility zentral in einer QUEUE/List Struktur verwaltet.**

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

Für einen Zugriff auf die QUEUE/List Strukturen bestehen drei Möglichkeiten:

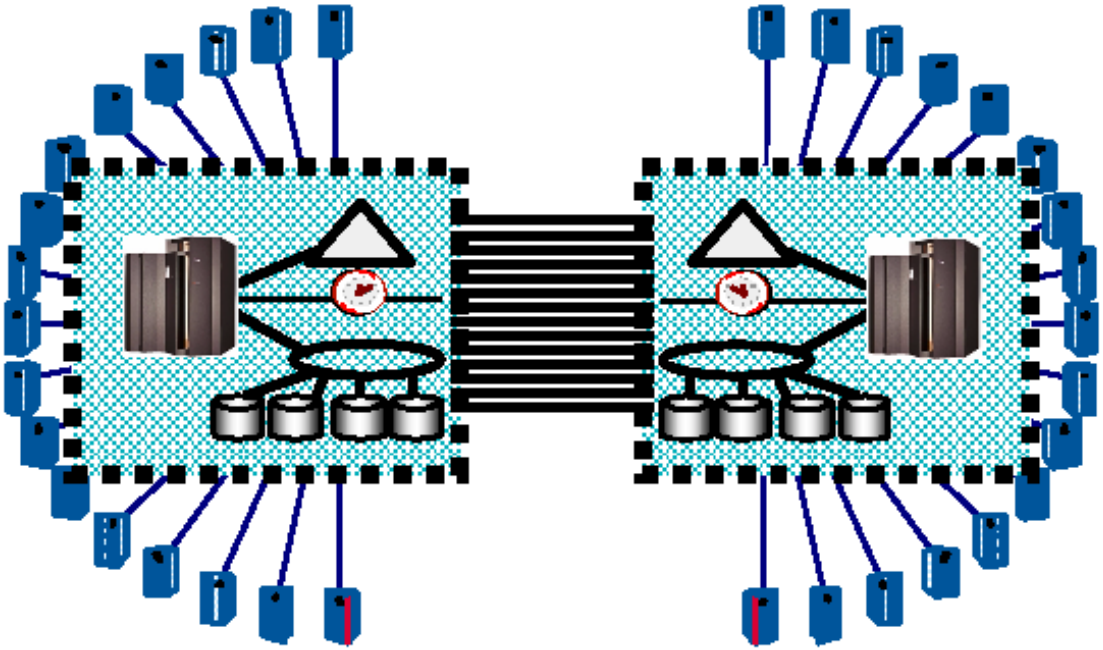
- **LIFO Queue**
- **FIFO Queue**
- **Key Sequenced**

Parallel Sysplex

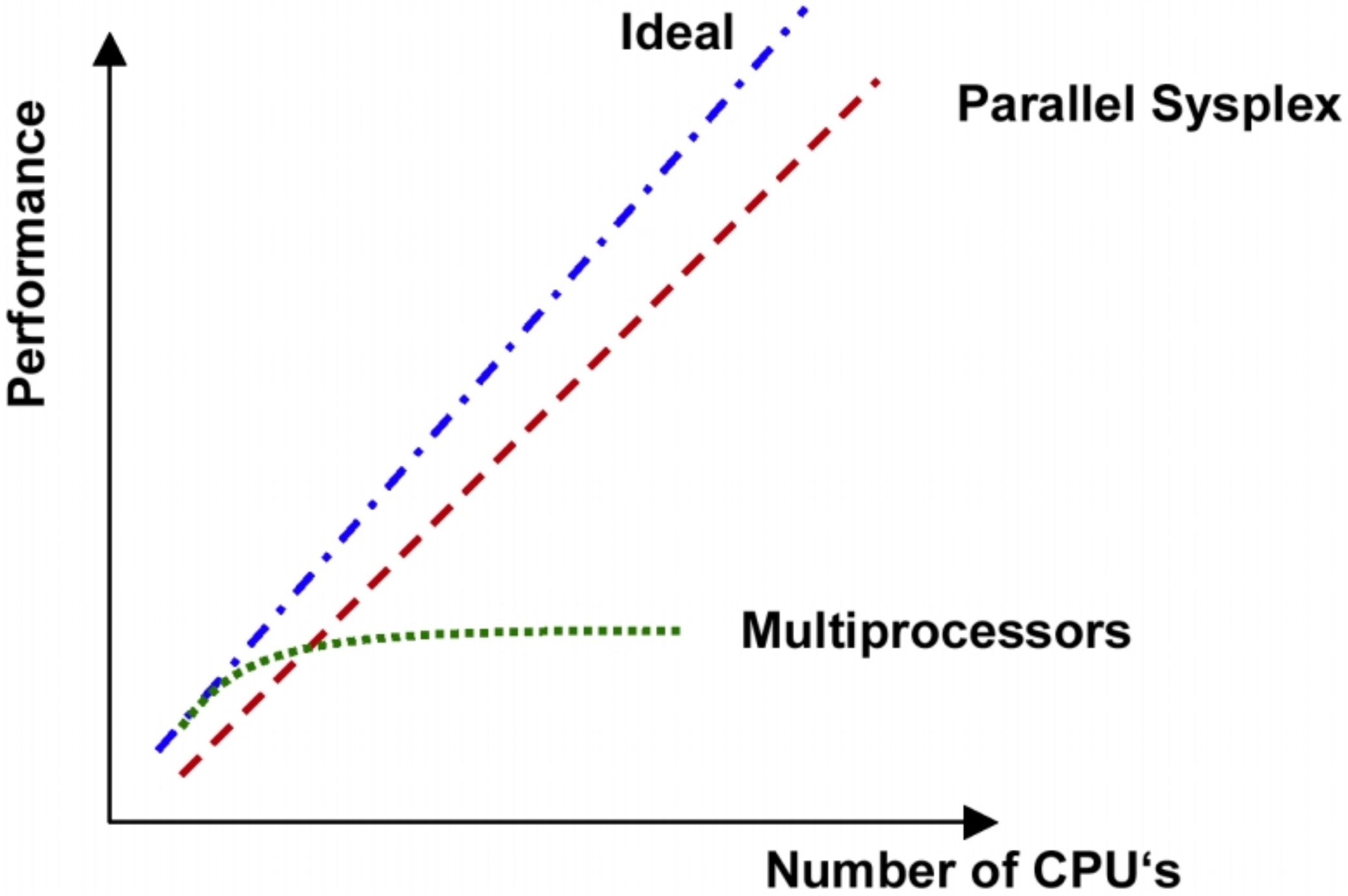


1 to 32 Systems

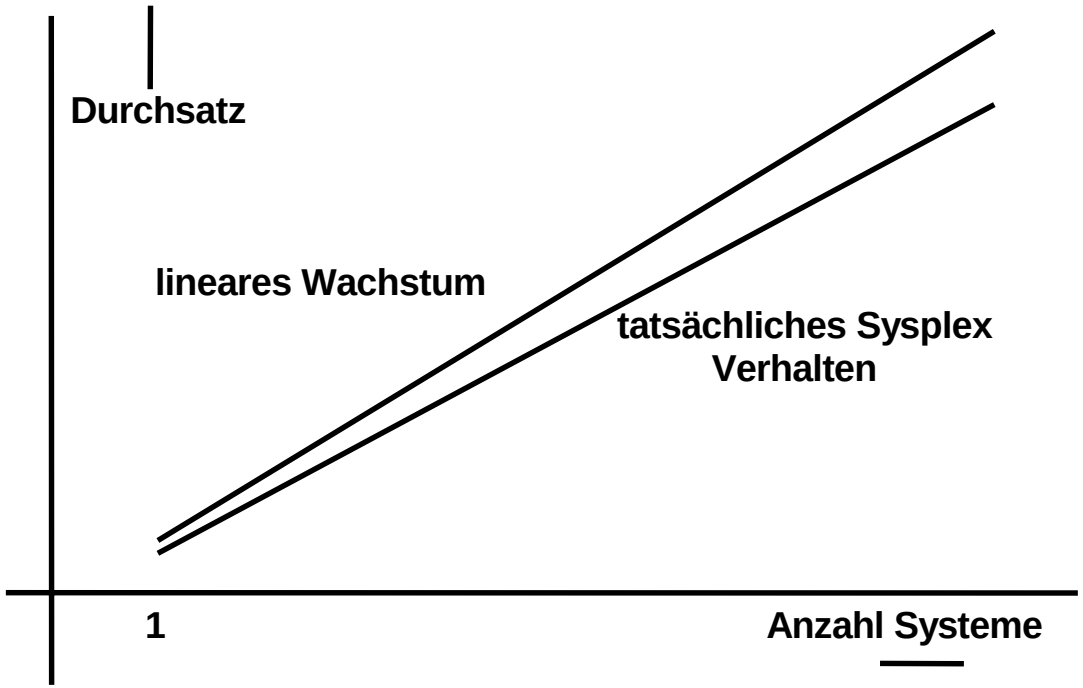
GDPS



Site 1 Site 2

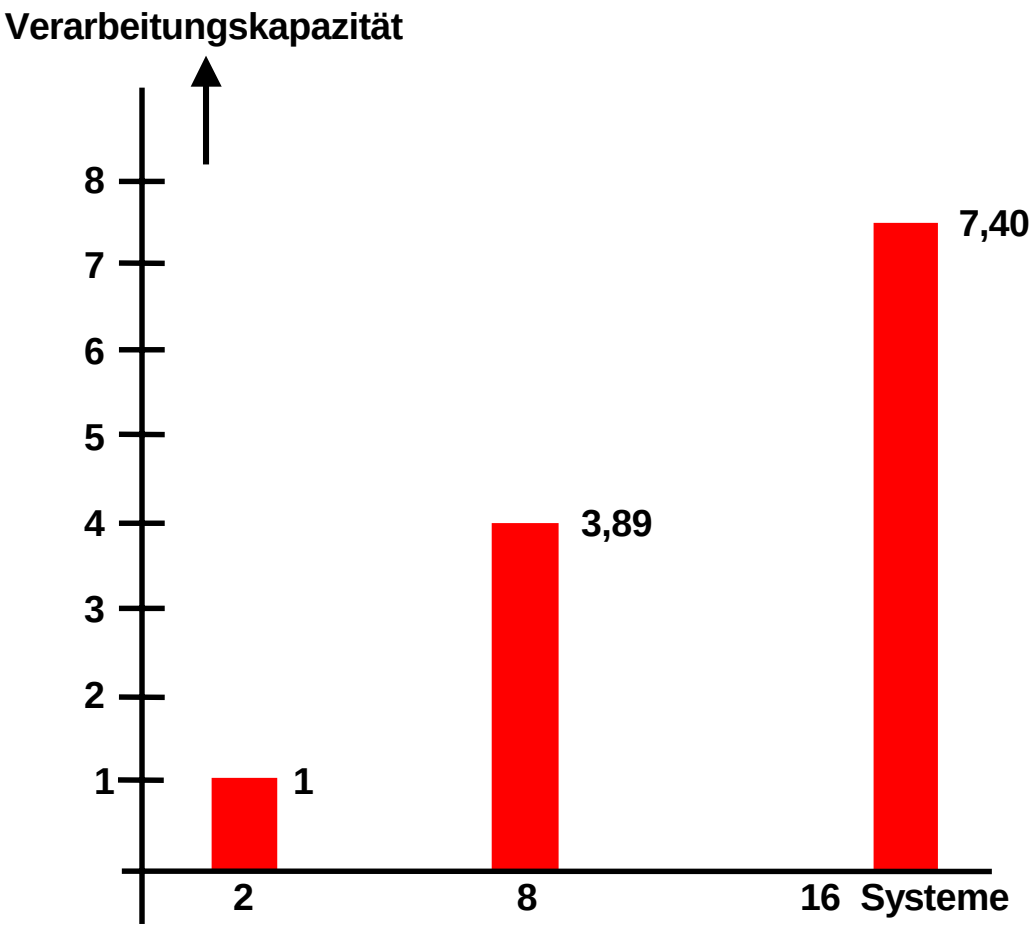

Ideal
Parallel Sysplex
Performance
Multiprocessors
Number of CPU‘s

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

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

Literatur: Coupling Facility Performance: A Real World Perspective, IBM Redbo