UNIVERSITÄT LEIPZIG

Mainframe Internet Integration

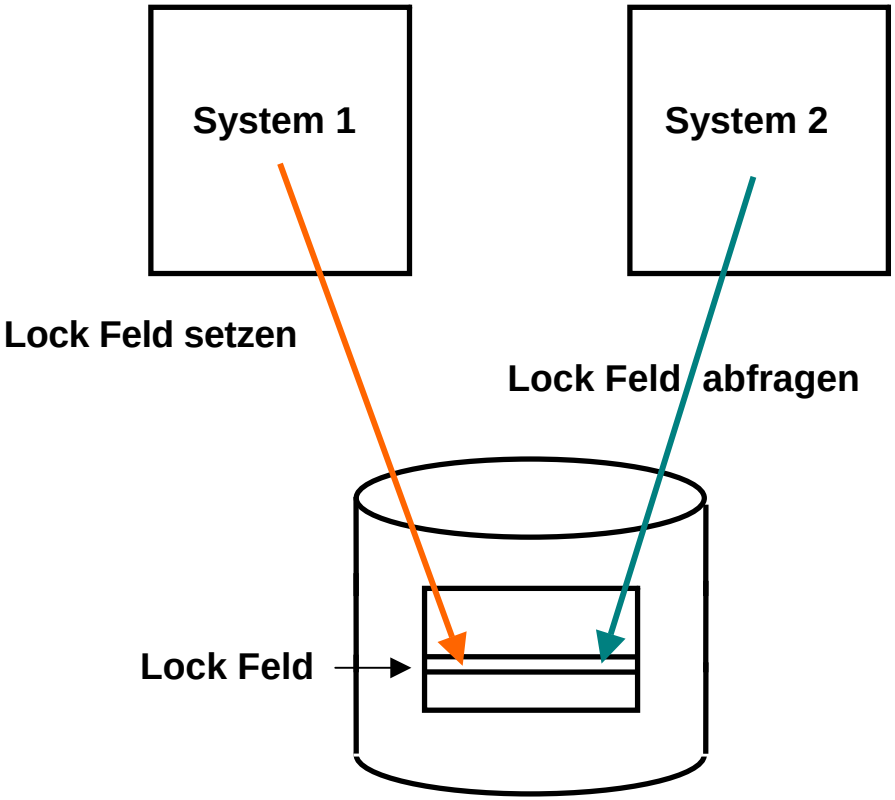
Prof. Dr. Martin Bogdan
Prof. Dr.-Ing. Wilhelm G. Spruth

SS2013

Sysplex Teil 3

Lock Strukturen

Lock Verwaltung



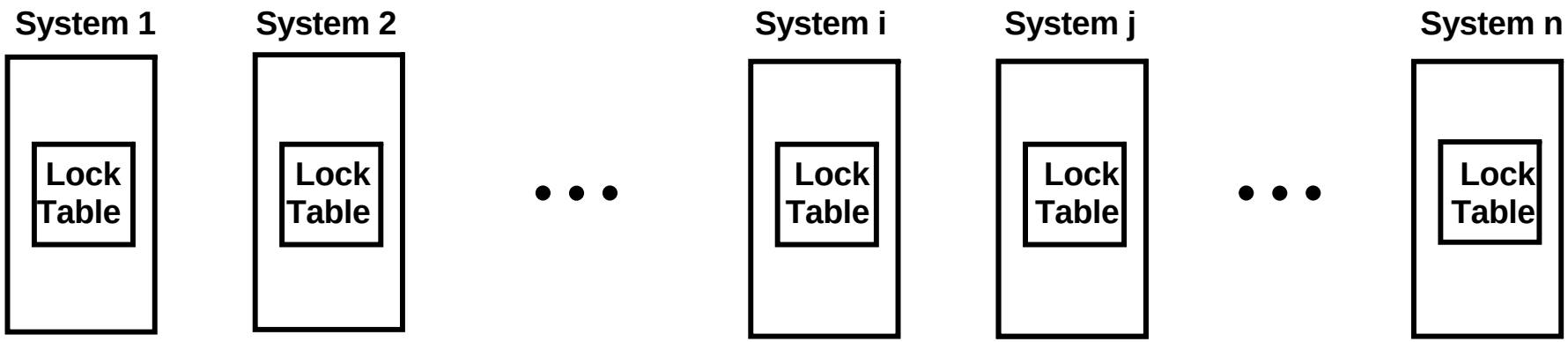
Frage: Wie speichert man die Locks ?

Einfachste Lösung: Locks mit den Daten auf dem Plattenspeicher speichern.

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 zusätzlichen Zugriffe auf den Plattenspeicher sind in Hochleistungssystemen nicht akzeptabel.



Verteilte Lock Tabelle

Decentralized Locks

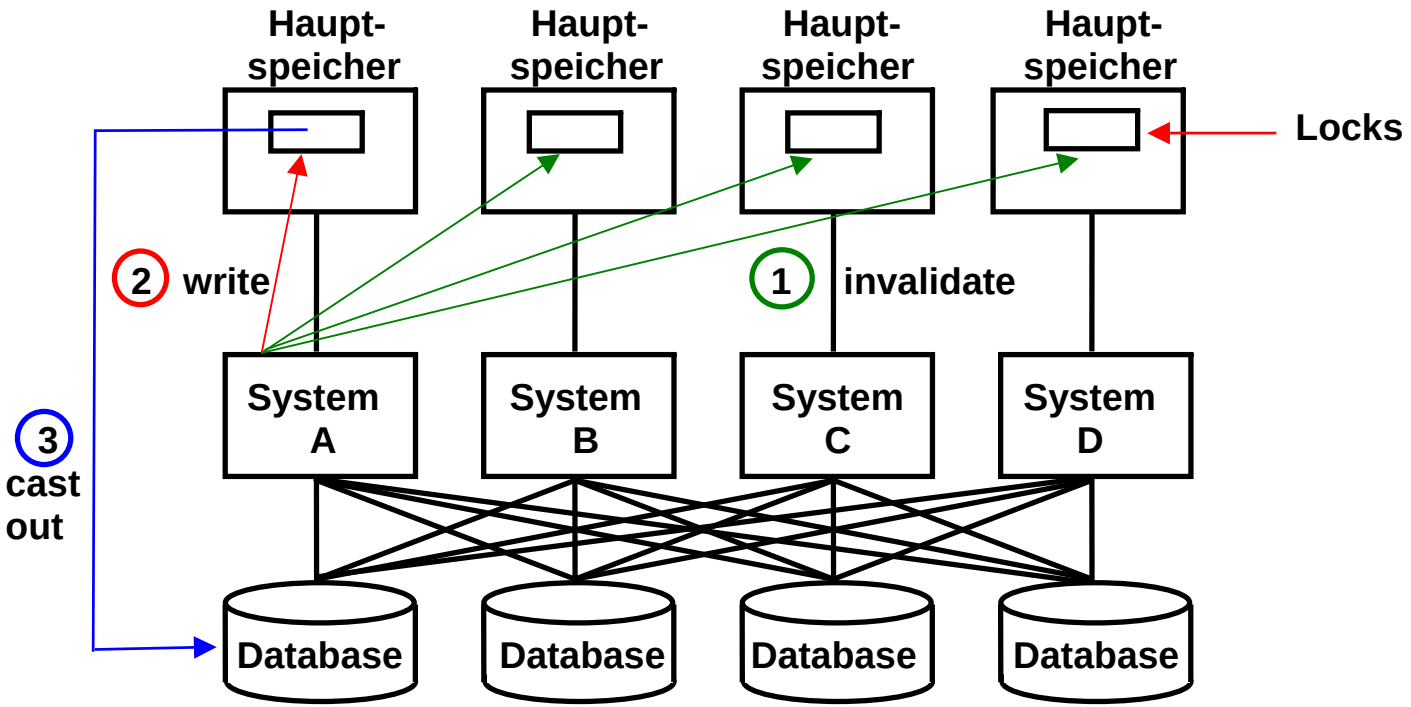
Zugriffe auf Datenbank-Tabellen mit Locks erfolgen recht häufig. Deswegen werden Lock Tabellen in der Regel im Hauptspeicher gehalten. Bei einem SMP mit nur einem Hauptspeicher ist dies unkritisch.

Bei einem Cluster mit mehreren SMPs und 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, in jedem System des Clusters die Verarbeitung der dort laufenden Transaktion auszusetzen, um ein Update der Lock Tabelle vorzunehmen. Dies erfordert einen Prozesswechsel. Der Overhead kann 20 ms CPU Verarbeitungszeit erfordern, auch dann, wenn das System an dem Zustand des Locks überhaupt nicht interessiert ist.

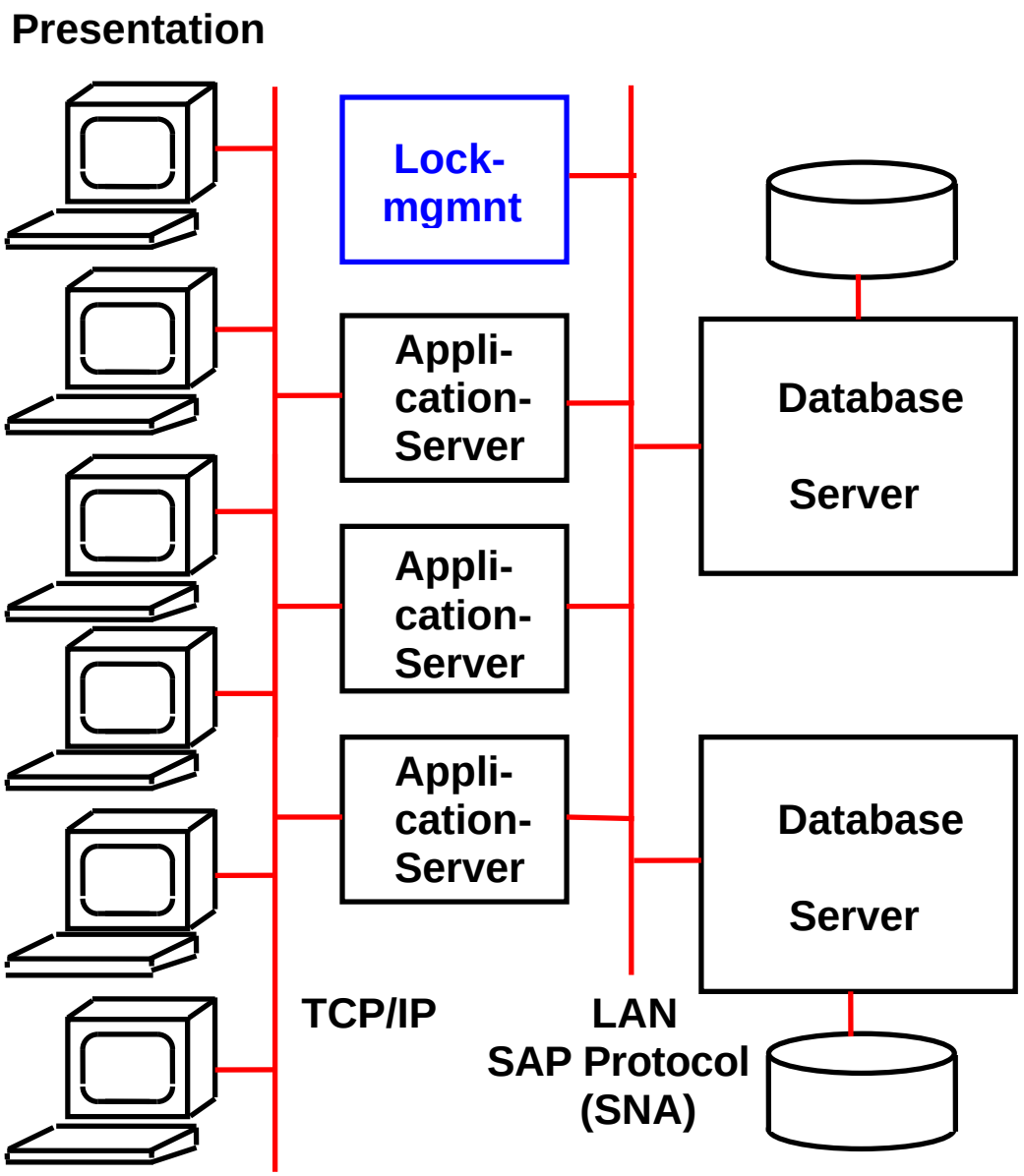
Beispiele für diese Lösung sind die VAX DBMS und VAX Rdb/VMS Datenbanksysteme. Gray/Reuter p.380



Invalidate-Broadcast Kohärenzsteuerung

Systeme A, B, C und D besitzen ein Read (S) Lock. System A möchte das Read Lock in ein Write (E) Lock umwandeln. Eine Invalidate Broadcast Nachricht an alle Knoten des Clusters benachrichtigt B, C und D, dass die Kopie des Locks nicht mehr gültig ist. Dies ist wegen des TCP/IP Overheads aufwendig, weil auch diejenigen Knoten die Nachricht erhalten und verarbeiten müssen, die an diesem spezifischen Lock gar nicht interessiert sind.

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



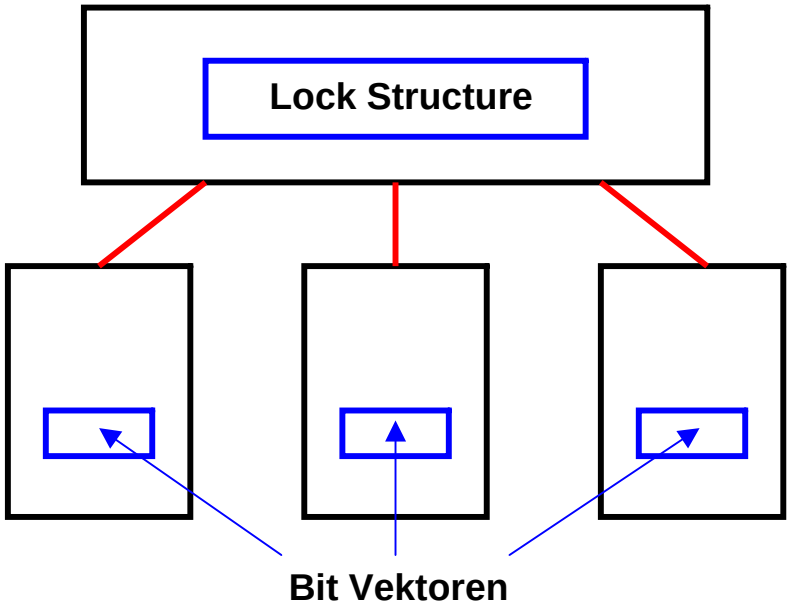
Typical SAP/R3 configuration

Dezentrale Locks arbeiten mit einer Invalidate-Broadcast Kohärenzsteuerung, die bei großen Transaktionsraten schlecht skaliert. Dieses Problem kann mit Hilfe einer 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 Lock 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.

Zur Information: Für die interne Kommunikation verwendet SAP eine Version des SNA Protokolls.



Locking mit der Coupling Facility

Die globale Sperrbehandlung (lock management) im Sysplex erfolgt in Zusammenarbeit der Coupling Facility (CF), der z/OS-Betriebssysteme sowie der auf jedem System laufenden Subsysteme (z.B. CICS, DB2), die eine globale Synchronisation benötigen.

Hierzu dient eine globale Locktabelle (Lock Structure) und lokale Bit Vektoren. Die globale Lock Tabelle ist in der Coupling Facility untergebracht, und enthält einen Eintrg für jedes zu verarbeitende Lock. Jedes an die Coupling Facility angeschlossene System enthält eine lokale Lock Tabelle, die als Bit Vektor bezeichnet wird. Alle Bit Vektoren gemeinsam enthalten eine Untermenge der in der CF Lock Structure enthaltenen Information.

Änderungen in der Lock Structure werden unmittelbar von der Coupling Facility in die betroffenen Bit Vektoren kopiert. Dies geschieht automatisch, ohne Kenntnissnahme durch das betroffene System.

Lokaler und globaler Lock Manager

Wie in der folgenden Abbildung gezeigt, läuft auf jedem Rechner ein lokaler Lock-Manager (LLM), der für die Lockanforderungen aller Transaktionen und Prozesse des jeweiligen Knotens zuständig ist.

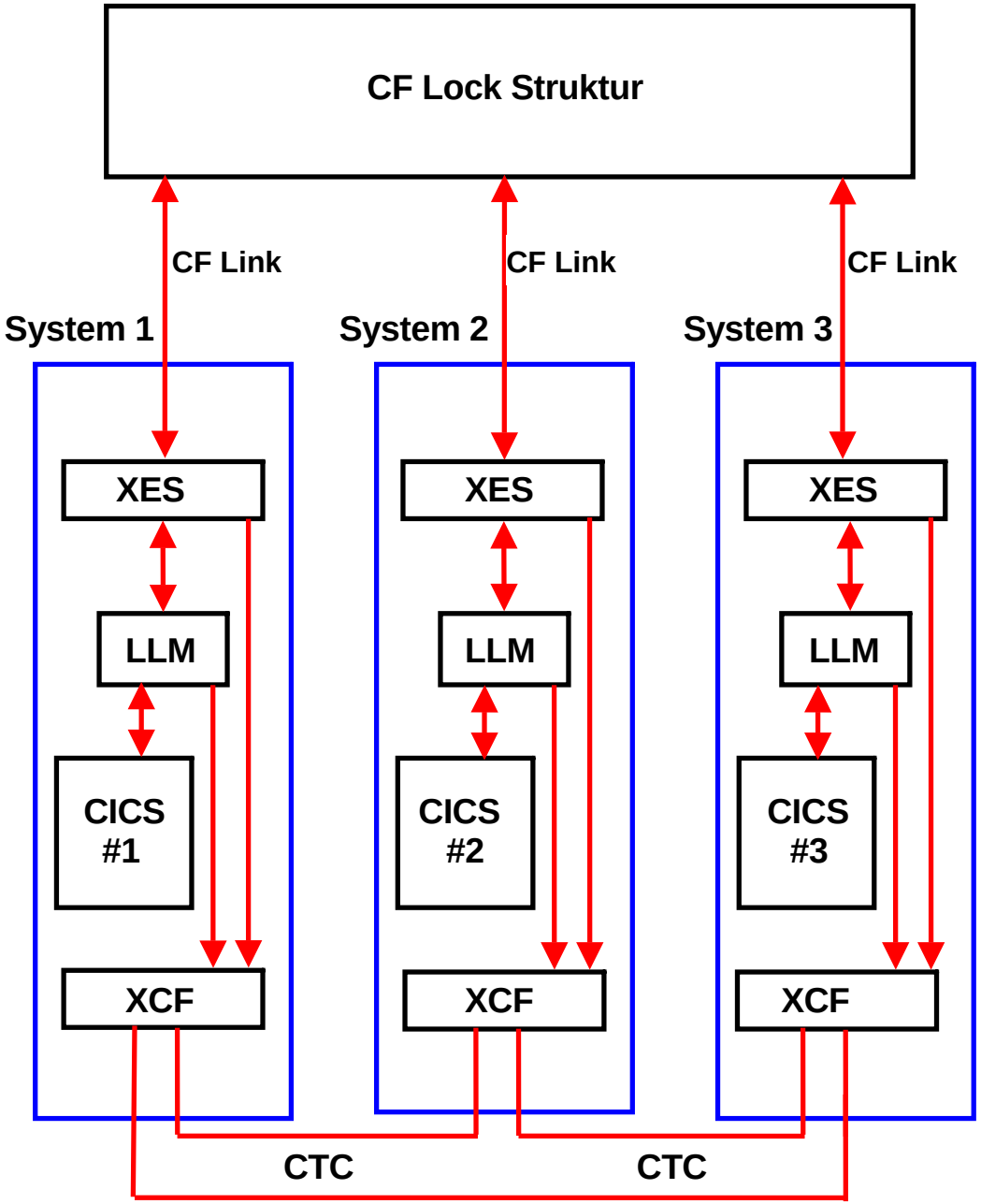
In der CF-Lock-Struktur wird eine zentrale Locktabelle verwaltet, auf die LLMs aller Knoten zugreifen, um eine schnelle systemweite Synchronisation konkurrierender Datenzugriffe zu erreichen. Diese Zugriffe erfolgen synchron über die Coupling Links und eine als „Cross System Extension Services" (XES) bezeichnete z/OS-Komponente, die u.a. spezielle Maschinenbefehle zum Setzen und Freigeben von globalen Sperren in der CF-Sperrtabelle verwendet.

Wenn ein Lock Anforderung auftritt, sind drei Alternativen möglich:

1. **Die Auflösung des Lock Anforderung kann mittels der Information im lokalen Bit Vektor und evtl. eines Lesezugriffs auf die Lock Struktur der CF bewältigt werden.**
2. **Die Auflösung der Lock Anforderung erfordert einen Schreibzugriff auf die Lock Struktur der CF. Die CF bewirkt daraufhin ein Update der Bit Vektoren in den anderen Knoten (Systemen) des Sysplex. Dies erfolgt in den anderen Knoten von diesen unbemerkt, ohne dass die dort laufenden Prozesse deshalb unterbrochen werden müssen.**
3. **Die Auflösung eines Lock Konfliktes erfordert auf einem (oder mehreren) betroffenen Knoten des Sysplex eine Aktion. In diesem Fall handelt der auslösende Rechner und der betroffene Rechner die erforderlichen Aktionen mittels einer Kommunikation über die CTC und XCF Verbindung direkt aus.**

Wichtig ist, dass die große Mehrzahl aller Lock Anforderungen durch die Alternativen 1 und 2 abgedeckt werden, und kein Aufwand für die Unterbrechung eines laufenden Prozesses erforderlich ist. Alle konfliktfreien Lockanforderungen, die typischerweise über 99% aller Locks betreffen, können über die zentrale Sperrtabelle (Lock Struktur) mit minimaler Verzögerung behandelt werden.

Nur solche Sperranforderungen, für die zwischen verschiedenen Rechner ein Konflikt auftritt, erfordern die Synchronisierung zwischen den betroffenen Rechnern. Dies bewirkt ein globaler System Lock Manager (SLM). Typischerweise sind lediglich 2 Systeme von dem Konflikt betroffen. Hierzu erfolgt ein Nachrichtenaustausch über das erwähnte CTC-Protokoll sowie über die z/OS-Komponente XCF.



Locking mit der Coupling Facility

Auf jedem System (Knoten) befindet sich ein lokaler Lock-Manager (LLM), der für die Sperrbehandlung aller seiner Prozesse (CICS in dem gezeigten Beispiel) zuständig ist.

Die XES (Cross System Extension Services) Komponente des z/OS Kernels ist für die Verbindung zur Lock Struktur in der Coupling Facility zuständig. XES benutzt hierfür die Coupling Link Prozessoren.

Zusätzlich können die Systeme (Knoten) des Sysplex über das CTC Protokoll und die XCF Komponente des z/OS Kernels paarweise (normalerweise über den FICON Switch) direkt miteinander kommunizieren.

Hash Klasse

Locks können mit einer unterschiedlichen Granularität ausgestattet sein. Ein Lock kann sich z.B. auf eine Reihe in einer SQL Tabelle, einen Teil einer SQL Tabelle, eine ganze SQL Tabelle, mehrere SQL Tabellen, einen ganzen Plattenstapel usw. beziehen. Alle Locks haben einen eindeutigen Namen. Der Name könnte z.B. in einem Feld in einer Reihe in einer SQL Tabelle untergebracht seinsein.

Die Lock Struktur der CF enthält eine globale Tabelle. Jedes aktive Lock ist durch einen Eintrag in der Lock Struktur Tabelle gekennzeichnet. Frage: Wie wird einem bestimmten Lock Namen ein eindeutiger Eintrag in der Lock Tabelle zugeordnet.

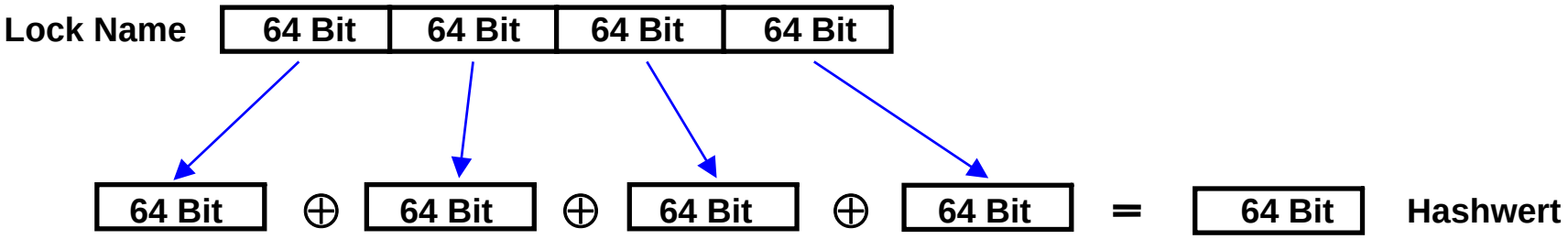
Das Problem ist, dass Lock Namen sehr lang sein können. Zum Beispiel haben Locks in IMS-Datenbanken bis zu 19 Bytes (152 Bits) lange Namen. Da eine Locktabelle mit 2^{152} Einträgen für den Hauptspeicher viel zu groß (und sehr dünn belegt) wäre, erfolgt eine Abbildung in eine Hash-Klassen-Bezeichnung von z.B. 20 Bit. Der Adressraum für die Lock-Einträge kann somit über eine CF-Locktabelle mit $n = 2^{20} = 1\,048\,576$ Einträgen abgedeckt werden.

Jeder Knoten (System) des Sysplex benutzt den gleichen Hash Algorithmus um den bis zu 152 Bit langen Namen in einen 20 Bit Hash Wert zu übersetzen.

Da die n Einträge Hash-Klassen repräsentieren, können „unechte Konflikte“ auftreten, wenn zwei Locknamen zufällig in dieselbe Hash-Klasse abgebildet werden. Die Häufigkeit solcher Kollisionen ist eine Funktion der Locktabellengröße. Diese kann vom Systemprogrammierer dynamisch verändert werden, um die Menge der falschen Konflikte zu minimieren. Eine Faustregel besagt, dass nicht mehr als 1% der Lockanforderungen zu Konflikten führen sollte.

Was ist Hashing ?

Ein einfaches Beispiel: Der Lock Name hat eine Länge von 256 Bit. Der Hash Wert soll eine Länge von 64 Bit haben.



Beispiel: Name (hier 256 Bit) in 4 Teile gleicher Länge (hier 64 Bit) zerlegen

Teile mit Exclusive Oder (Symbol ⊕) verknüpfen. Das Ergebnis ist ein 64 Bit Hash Wert.

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

Der Hash Algorithmus (hier eine einfache Exclusive Oder Verknüpfung) soll sicherstellen, dass für beliebige Nachrichten alle Bitmuster des Hashwertes mit möglichst gleicher Wahrscheinlichkeit vorkommen. Es existiert eine sehr große Auswahl an unterschiedlichen Hash Algorithmen.

Größe der CF Lock Tabelle

Unechte Konflikte treten auf, wenn zwei Locknamen in dieselbe Hash-Klasse abgebildet werden. Die Anzahl der unechten Konflikte sollte möglichst klein gehalten werden. Wie groß muss die Lock Tabelle konfiguriert werden ?

Schauen wir uns ein Beispiel an.

Nehmen wir einen Sysplex an, der 10 000 Transaktionen pro Sekunde verarbeitet, mit einer durchschnittlichen Verarbeitungsdauer (Antwortzeit) von 0,5 Sekunden pro Transaktion. In jedem Augenblick verarbeitet der Sysplex 5 000 Transaktionen gleichzeitig. Nehmen wir gleichzeitig an, jede Transaktion benötigt durchschnittlich 20 Locks. Die durchschnittliche Anzahl aller aktiven Locks beträgt damit 100 000.

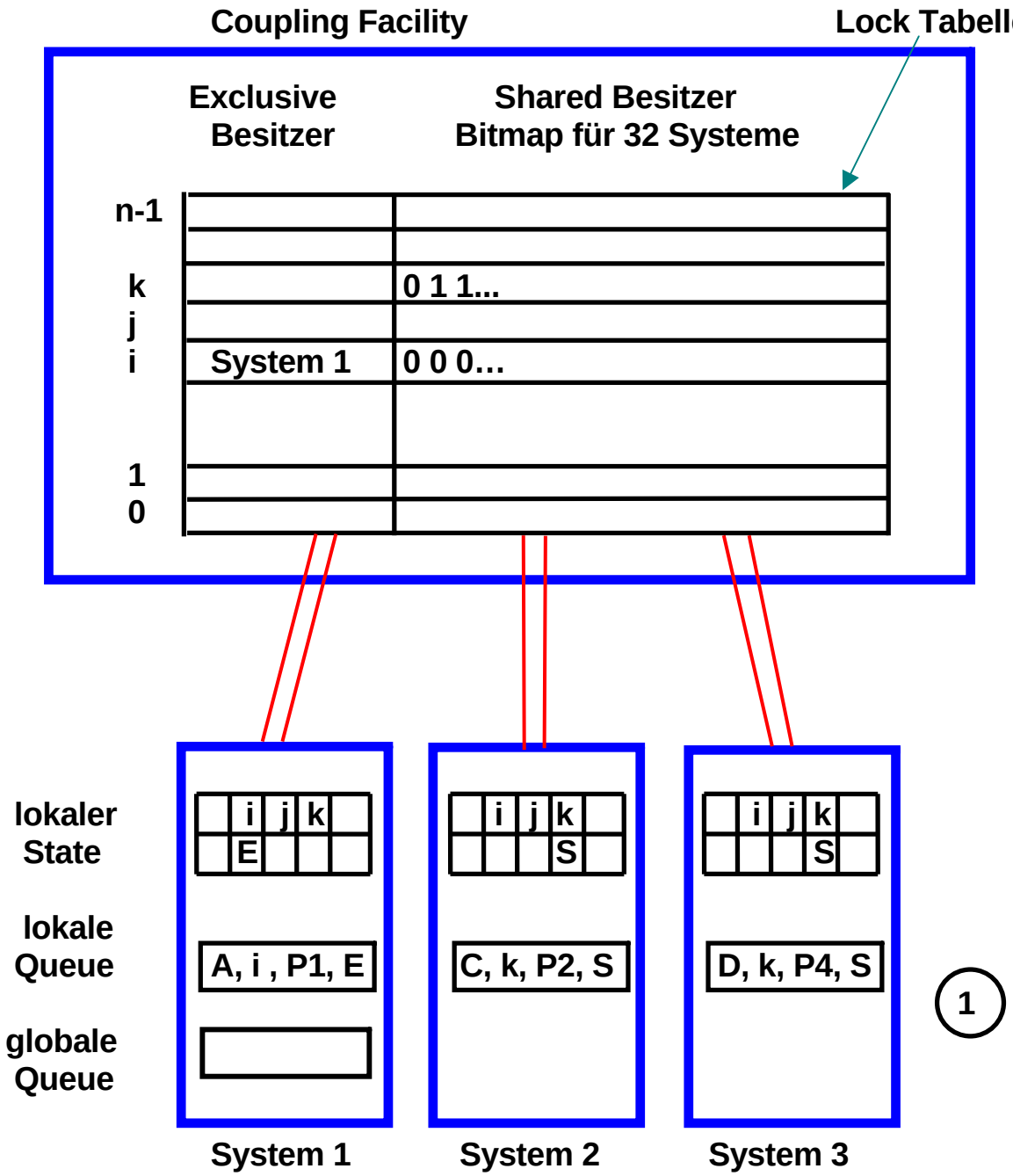
Wenn die Lock Tabelle 20 000 000 Einträge hat, beträgt die Wahrscheinlichkeit etwa 0,5 % , dass 2 Locks zufällig in der gleichen Hash Klasse abgebildet werden.

Nutzung der CF Lock Tabelle

Die folgende Abbildung verdeutlicht für eine Beispielkonfiguration mit drei Systemen (Knoten) den Aufbau der globalen Locktabelle in der CF sowie der lokalen Locktabellen der LLMs.

Die zentrale Lockinformation wird nicht auf Ebene einzelner Transaktionen, sondern nur für ganze Systeme geführt. Das Lockprotokoll unterscheidet dabei wie üblich Leselocks (Shared, S) , welche pro Objekt gleichzeitig an mehrere Systeme gewährt werden können, und exklusive Schreiblocks (E). Die CF-Locktabelle (oberer Teil in der Abbildung) verwendet hierzu pro Lockeintrag (Hash-Klasse) zwei Felder: Für den Fall eines exklusiv gesetzten Locks (EXC) enthält das erste Feld die Knoten-Adresse des Besitzers.

Das zweite Feld enthält einen Bit-Vektor mit 32 Bit-Länge. Jedes Bit dieses Vektors zeigt an, welcher der (maximal möglichen) 32 Knoten (Systeme) eines Sysplex eine Lese-Lock erworben hat (Shared, S = 1) bzw. ob kein solches vorliegt (S = 0). Das Beispiel weist somit aus, dass System 1 exklusiver Besitzer von allen Objekten der Hash-Klasse i ist und dass Systeme 2 und 3 Leseberechtigungen für alle Objekte der Hash-Klasse k besitzen.



Nutzung der CF Lock Tabelle

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

Frei	0
Shared	S = Shared
Exclusive	E = Exclusive

Nutzung der CF Lock Tabelle

Wie im unteren Teil der Abbildung gezeigt bestehen die lokalen Sperrtabellen der LLMs aus drei Teilen: lokale Sperrzustände (Local Lock State), objektspezifische Sperranforderungen lokaler Transaktionen (Local Queue) und globale Warteschlangen mit objektspezifischen Sperranforderungen mehrerer Rechner (Global Queue).

Im lokalen Sperrstatus führt der LLM seine Kenntnis vom Zustand der CF-Sperreinträge bezüglich der Hash-Klassen: "0" bedeutet keine Kenntnis (Eintrag wird von keinem anderen Knoten benutzt), "S" (Shared-Kenntnis, LLM hat Leserecht), "E" (exklusive Nutzung durch den eigenen Knoten) oder "Gx" (ein anderer Knoten x besitzt exklusive Rechte und kontrolliert die Sperrvergabe im Sysplex).

Im Beispiel weisen die lokalen Einträge entsprechend der CF-Sperreinträge System 1 als exklusiven Besitzer der Hash-Klasse i und Systeme 2 und 3 als leseberechtigt für Hash-Klasse k aus. Der zweite Bereich der lokalen Sperrtabellen enthält Informationen zu den spezifischen Objekten und lokal laufenden Transaktionen bzw. Prozessen, welche Sperren besitzen oder darauf warten. So bedeutet der Eintrag "(A, i, P1, E)" in System 1, dass der lokale Prozess P1 die Sperre zu Objekt A (Lock mit dem Namen A), welche zur Hash-Klasse i gehört (A hashes auf i), im exklusivem Modus besitzt. Der dritte Bereich (Global Queue) enthält nur relevant für Objekte, an denen ein systemübergreifender Sperrkonflikt in der CF erkannt wurde und der LLM zur Konfliktauflösung mit anderen LLMs zusammenarbeiten muß. Hierzu werden in dem Sperrbereich die Sperranforderungen der anderen Rechner abgelegt, um diese global synchronisiert abzuarbeiten.

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, nicht aber den potentiell 29 weiteren Systemen) direkt auszuhandeln.**